PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UM MÉTODO PARA SEGMENTAÇÃO DE PREDITORES

ROBERTO ÂNGELO FERNANDES SANTOS

Tese de Doutorado

Universidade Federal de Pernambuco

posgraduacao@cin.ufpe.br

www.cin.ufpe.br/~posgraduacao

RECIFE, MARÇO/2010

UNIVERSIDADE FEDERAL DE PERNAMBUCO (UFPE)
CENTRO DE INFORMÁTICA (CIN)
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ROBERTO ÂNGELO FERNANDES SANTOS

UM MÉTODO PARA SEGMENTAÇÃO DE PREDITORES

Tese apresentada à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito para obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. ROBERTO SOUTO MAIOR DE BARROS, PhD

RECIFE, MARÇO/2010

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus, pela minha existência.

A toda minha família, em especial, ao meu falecido pai Augusto Ângelo e a minha mãe Sônia Santos, que me ensinaram o valor dos estudos em minha vida.

A minha esposa e filho, por sua paciência e pelo apoio.

Um muito obrigado ao meu orientador Doutor Roberto Souto Maior de Barros, que me apoiou e orientou, mesmo nos momentos mais difíceis.

Aos meus companheiros da Serasa Experian, em especial Felipe Coelho, Amanda Caputo e José Lee, que tornam minha tese menos solitária com as nossas grandes discussões.

A minha tia, que me acolheu em seu lar e me trata como se fosse seu filho.

Outro muito obrigado ao meu sempre orientador e amigo Doutor Décio Fonseca, que me apoiou em todos os meus principais momentos acadêmicos.

Meu agradecimento a todos os professores que sempre me deram incentivo. À empresa Serasa Experian que me deu oportunidade de aplicar os meus conhecimentos em Mineração de Dados.

E por fim, agradeço a todos, não citados aqui, mas que me apoiaram e me ajudaram de alguma forma.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

KDD	*Knowledge Discovery in Databa*ses
kNN	*k-Nearest Neighbours*
KS	*Kolmogorov Smirnov*
MLP	Rede Neural *Multilayer Perceptron*
OLAP	*On Line Analytical Process*
OLAM	*On Line Analytical Mining*
RNA	Redes Neurais Artificiais
ROC	*Receiver Operating Characteristics*
SVM	*Support Vector Machine*
NNTree	Segmentação utilizando o método *NNTree*
SegTree	Segmentação por *Information Gain*
RISKSEG	Segmentação utilizando o método *RISKSEG*

RESUMO

Este trabalho propõe um método para segmentação de modelos preditivos. O seu objetivo é combinar diferentes preditores, um para cada segmento dos dados, a fim de buscar modelos específicos, que tornem a resposta combinada final mais precisa. Os modelos utilizados nas segmentações podem ser de diferentes técnicas.

A combinação de modelos é uma abordagem bastante utilizada na tentativa de desenvolvimento de classificadores com melhor desempenho. Esta abordagem tem originado diversos trabalhos científicos e muitas aplicações práticas. Existem diversas maneiras de fazer a combinação de métodos preditores, as mais comuns são: *Bagging*, *Boosting* e *Stacking*. Entretanto, a segmentação dos dados é particularmente interessante porque, além de proporcionar melhores resultados na sua aplicação, possui a opção de preservar algumas das características dos diferentes tipos de classificadores utilizados. Normalmente, a segmentação dos dados baseia-se na experiência de especialistas ou utiliza métricas de separação das classes a serem preditas (entropia, por exemplo), como é feito em treinamentos de Árvores de Decisão. Alguns trabalhos também citam métodos exaustivos que testam um grande número de possibilidades.

Um novo método para combinar preditores, que segmenta uma base de dados em amostras excludentes em uma árvore de modelos é apresentado. Para o treinamento desta árvore foi desenvolvido um algoritmo que constrói uma Árvore de Decisão, a qual cada folha contém amostras excludentes e modelos treinados. A segmentação dos dados de treinamento, e consequentemente dos modelos, é feita pelas ocorrências das categorias das variáveis candidatas ao processo de divisão. O método se utiliza principalmente de um modelo fatorial para determinar quais são as variáveis mais importantes para segmentar.

Os resultados mostram que o método proposto apresenta diversas vantagens em relação a outros métodos de combinação de classificadores (inclusive a segmentação tradicional), que vão além do aumento do poder de classificação. Neste trabalho, o método é aplicado em bases de dados artificiais e de mundo real com resultados satisfatórios. Os experimentos mostram que a segmentação dos dados, utilizando o método proposto, pode ser uma importante ferramenta na construção de aplicações com modelos combinados. Estes experimentos permitem identificar muitos casos em que as condições para utilização do método são favoráveis.

ABSTRACT

This work proposes a method for the split of predictive models. The purpose is to combine different predictors, one for each data segment, to find specific models in order to make their combined result more precise. The models in this segmentation can use different techniques.

The combination of models is a widely used approach and has resulted in many scientific studies and practical applications. There are many ways of combining predictive methods, the most common are: *Bagging*, *Boosting* and Stacking. Nevertheless, data segmentation is particularly interesting because, besides providing an increase in prediction capacity for applications, there is also the possibility of maintaining some features from the techniques used. Often, data segmentation is based on experts experience or on metrics to split the data (like entropy, for example), as we can see in Decision Trees training. Some studies also mention exhaustive methods that test a large number of possibilities.

A new method for combining predictors, which split a database in exclusive samples in a tree of models is presented. For this training, an optimized algorithm was developed which builds a tree of models, such as each leaf contains exclusive samples and trained models. Data training segmentation and models are done by categories of candidate variables to participate in the division process. The method uses a factorial model to establish which variables are most important for the split.

The results show that there are several advantages among the proposed method and other classifier combination methods (including traditional segmentation), over and above the increase in predictive accuracy. In this study the method is applied in artificial and real world databases with good results. The experiments show that data segmentation using the proposed method can be an important tool to build applications with combined models. These experiments permit the identification of many cases where there are favorable conditions.

CAPÍTULO 1

INTRODUÇÃO

1.1 – MOTIVAÇÃO

A multiplicação de situações complexas e de difícil previsibilidade, bem como as reações administrativas habituais, se revelam cada vez mais ineficazes, impondo a renúncia de antigos paradigmas. A competição entre organizações, a velocidade das comunicações e o volume das informações a serem trabalhadas superam largamente a capacidade de processamento humano e mecânico, exigindo ferramentas mais adequadas nas decisões gerenciais [Dru99] [San02]. Desta forma, o sucesso de qualquer organização está condicionado à capacidade de analisar, planejar e reagir rapidamente às mudanças no ambiente ao qual ela está inserida [Dru99], tornando-se imprescindível a captação e o processamento de dados internos e externos às organizações e sua utilização no menor espaço de tempo possível.

Sabe-se que o resultado de modelos de decisão baseados em dados depende da quantidade e qualidade das informações disponíveis para modelagem, porém esta não deve ser a única preocupação, pois a melhoria das técnicas utilizadas para predição (Regressões Estatísticas e Redes Neurais, por exemplo) pode ser uma importante fonte de aumento da acurácia. Esta tendência pode ser observada por meio de diversos trabalhos práticos ou científicos elaborados por profissionais e por acadêmicos de diversas áreas como: estatística, física, administração e ciência da computação [Rud01] [ChH03] [HaB03] [AVA04] [San07]. Mas esta busca vai além do uso de apenas um modelo preditor e sinaliza para uma forte tendência do uso combinado de diversos preditores, sejam eles da mesma ou de diferentes técnicas. A literatura mostra que a combinação de técnicas por meio de diferentes métodos de amostragem ou pela combinação de preditores de naturezas distintas tem a oferecer, normalmente, soluções mais preditivas [Wol92] [LCL02] [DzZ04] [BWH05] [WiF05] [Rok05] [VCB05] [GBD06] [FER06] [IYN08].

A complexidade da combinação de preditores deve-se à quantidade de fatores envolvidos em sua construção e em seu uso. Para ilustração desta complexidade, pode-se pensar em

algumas questões como: um modelo de classificação baseado em preditores combinados é melhor do que um com apenas um classificador? Qual o melhor método de combinação para um determinado caso? Quais técnicas deverão ser combinadas? Quantos preditores são necessários para o melhor desempenho? A verdade é que não existem respostas simples e exatas para estes questionamentos e a maioria deles só podem ser respondidos a partir das restrições da aplicação e da expertise dos desenvolvedores de modelos.

O uso destes métodos de combinação tende a apresentar melhores resultados quando existe grande diversidade entre os classificadores [BWH05] [Rok05]. A metodologia de combinação de métodos de classificação consiste em utilizar diversos classificadores e criar um único, integrado, e com maior desempenho. A diversidade existe quando diferentes classificadores possuem diferentes variâncias e diferentes vieses em relação ao treinamento. A combinação destes classificadores de maneira adequada maximiza o poder de predição ou outra medida de desempenho [BWH05] [VCB05].

Quando se trata de preditores múltiplos, o maior desafio é obter o melhor conjunto de preditores individuais, de tal forma que tragam informação relevante para se criar um novo preditor baseado nas repostas destes. Basicamente, os preditores múltiplos começaram a ser estudados para tentar encontrar relações entre diversos preditores e a variável alvo [VCB05]. Diferentes preditores apresentam erros distintos, ou seja, alguns são mais eficientes para certos domínios que outros e vice-versa. Na tentativa de entender e explicar melhor essas relações, funções de combinação destes preditores começaram a ser estudadas. O resultado foi que novos métodos foram criados a partir dessas funções. Eles puderam mostrar melhor desempenho que os individuais devido à diversidade encontrada nas diferentes predições. A partir daí, essa diversidade passou a ser importante para se obter grandes melhorias na resposta final, portanto, alvo de estudos e ferramentas para se obtê-la [VCB05].

Dentre os mais diversos métodos de combinação de técnicas preditoras, uma das mais utilizadas é a segmentação. Ela propõe a divisão de um problema complexo em diversos problemas de complexidade inferior para se obter uma solução melhor para o problema. Em Mineração de Dados, a segmentação é vista como uma forma de se obter subgrupos tais que técnicas de aprendizado possam ser aplicadas, resultando em diversos classificadores especialistas, ou seja, preditores especializados em cada subgrupo gerado a partir da divisão do conjunto de dados inicial.

Existem diversas abordagens para que a segmentação dos dados seja feita, dentre elas, as técnicas subjetivas que se utilizam principalmente do conhecimento do especialista para criação

destes grupos. Eles geralmente utilizam dois parâmetros para definir a segmentação de seus modelos: o primeiro é a suficiência de informações para determinados nichos (grupos que possuem mais informações e grupos que não as possuem). O outro é a utilização de especialistas experientes, que já testaram exaustivamente diversas segmentações e que possuem experiências empíricas anteriores de sucesso [ThC02]. Chegou-se a acreditar que não existiria uma metodologia para criação de uma estrutura de segmentos sem o uso de conhecimento subjetivo [MaR05]. Entretanto, alguns autores já propuseram métodos objetivos e sistemáticos para segmentação com o foco na melhoria da classificação final, são métodos exaustivos ou semi-exaustivos de busca pela melhor estratégia de segmentação [MaR05].

Outra estratégia muito comum para segmentar, que vale a pena ser citada, é a utilização dos mesmos critérios usados na criação dos nós das Árvores de Decisão, métricas como: *Information Gain*, entropia, KS (*Kolmogorov Smirnov*), entre outros [Nev98] [Nev99] [CGM02]. Por se tratar apenas de uma adaptação, não há garantias de que o uso destas métricas resulte em um bom conjunto de modelos segmentados.

1.2 – OBJETIVOS E CONSIDERAÇÕES INICIAIS

Este trabalho propõe um método para segmentação de modelos preditores. O método foi desenvolvido para gerar decisões mais preditivas a partir da segmentação dos dados e construção de modelos especializados nesses segmentos. O método de segmentação desenvolvido utiliza um modelo fatorial para descobrir as melhores variáveis candidatas e segmentar os dados de forma a gerar um algoritmo preditor para cada um dos segmentos. Não há grandes restrições quanto aos algoritmos preditores utilizados. Dessa forma, qualquer técnica de aprendizado de máquina que trabalhe com classificações e/ou Regressões pode ser aplicada. O método proposto é a primeira e mais importante contribuição do trabalho, outras contribuições são discutidas no Capítulo 8.

O método proposto é orientado à maximização do poder preditivo, buscando os melhores atributos sem a utilização de buscas exaustivas. Ele se mostrou mais preditivo, em muitas aplicações, do que os tradicionais métodos de combinação de preditores e também em relação à forma mais tradicional de se fazer segmentação. Este método inclui, ainda, a aplicação de um algoritmo para junção de resultados, conhecido como *Stacking*, que faz a unificação dos resultados dos modelos de cada segmento. Foi utilizado também um algoritmo simples para transformação das variáveis numéricas em categóricas, através da distribuição em n classes com mesma densidade numérica e, finalmente, outro algoritmo para fazer a combinação das categorias das variáveis, duas a duas. Em Árvores de Decisão, a categorização é baseada apenas

em intervalos disjuntos e a possibilidade de utilização de dois intervalos que não sejam contínuos é pouco ou quase nada explorada. A combinação de subconjuntos permite agrupar registros que apresentam comportamentos similares em relação ao alvo definido, mesmo que estes não sejam contínuos.

Para demonstração da eficácia do método proposto foram realizadas simulações nos capítulos de estudo de caso permitindo assim uma comparação com métodos individuais de classificação (Redes Neurais, Regressão Logística e Regressão Linear) e diversos métodos de combinação (*Bagging*, *Boosting*, *NNTree*, Segmentação por *Information Gain* e *RISKSEG*) em 4 (quatro) bases geradas artificialmente, 12 (doze) bases do *UCI Repository of Machine Learning Databases* [BlM10] e em 2 (duas) bases reais cedidas pela *Serasa Experian*. Estas comparações são importantes para determinar onde o método obtém bons desempenhos, quais são os seus aspectos positivos e suas limitações. As comparações apresentadas nestes casos contêm uma diversidade de bases e métodos de combinação não encontrada em outros trabalhos na área de Mineração de Dados, principalmente nas comparações entre métodos de segmentação com *Bagging* e *Boosting.*

Para técnicas de classificação tradicionais, normalmente, a otimização limita-se à sua própria natureza, como erro quadrático médio e verossimilhança, para as Regressões Lineares e logísticas, respectivamente. Métodos de combinação permitem que outras naturezas de medidas possam ser otimizadas. Por exemplo, métricas como *KS2* [Con99] e *ROC* [Faw06] são geralmente obtidas após o treinamento de um classificador (e.g. Redes Neurais), desta forma, é possível a utilização de métodos de combinação para selecionar os classificadores que maximizem uma destas métricas, pois houve o treinamento de diversos classificadores para um mesmo objetivo. O método proposto explora esta possibilidade de otimização de métricas pouco ortodoxas, mas que já são utilizadas como avaliadoras de desempenho em trabalhos acadêmicos e aplicações de mundo real [NaP03] [AVA04]. Tal paradigma é pouco ou nada explorado em trabalhos com *Bagging* e *Boosting*, pois eles geralmente focam na apresentação de medidas de erro (normalmente erro de classificação). Para demonstrar o uso desta possibilidade, no último capítulo de experimentos foram realizados testes com a variação da métrica de otimização em duas bases reais.

Com objetivo de comparar o método proposto com um trabalho mais recente, também foi implementado um algoritmo de combinação de Redes Neurais chamado *NNTree* [Pra08]. Apesar do artigo que descreve o método *NNTree* apenas propor a utilização de Redes Neurais, foi feita uma adaptação de maneira a suportar mais dois algoritmos: Regressão Logística e

Regressão Linear. Esta implementação também permitiu a análise do método *NNTree* com outros classificadores, extrapolando assim a proposição do método pelo autor do artigo.

1.3 – ORGANIZAÇÃO DA TESE

Os parágrafos a seguir descrevem os próximos capítulos deste documento.

No **Capítulo 2** é abordado o processo de KDD, técnicas estatísticas e Inteligência Artificial mais comuns para os problemas de classificação/regressão e métodos utilizados para avaliação de desempenho. Mostra-se, desta forma, alguns dos fundamentos básicos para o entendimento da tese.

No **Capítulo 3** são apresentados os principais conceitos envolvendo combinação de métodos preditores. Neste capítulo são abordados os principais métodos de combinação de classificadores: *Bagging*, *Boosting*, *Stacking* e Segmentação.

No **Capítulo 4** são descritos os detalhes de funcionamento do método de segmentação proposto para combinação de classificadores, uma análise crítica e comparação teórica com outros métodos.

No **Capítulo 5** são realizadas as simulações que demonstram situações em que existem vantagens em utilizar o método proposto, bem como situações em que a sua utilização não é indicada, sempre comparando com preditores individuais e com outros métodos de combinação. Este estudo de caso utiliza bases pré-fabricadas, pois, desta forma, pode-se controlar alguns fatores de criação das bases, como número de casos, número de variáveis e método de construção. Os resultados destes experimentos são avaliados com os métodos estatísticos adequados e a maioria deles é discutida no Capítulo 2.

No **Capítulo 6** são efetuadas outras simulações com 12 (doze) bases do repositório *UCI* [BlM10]. Assim como no Capítulo 5, os resultados também são comparados com outros métodos. Os resultados deste capítulo também são avaliados com métodos estatísticos discutidos no Capítulo2.

No **Capítulo 7** são mostrados também resultados da aplicação do método proposto em 2 (duas) bases de dados reais utilizadas na concessão de crédito e detecção de fraude. A idéia é, principalmente, mostrar como o método proposto seria aplicado no mundo real. Os resultados destes experimentos são avaliados com métodos estatísticos e indicadores de desempenho adequados a este tipo de aplicação.

No **Capítulo 8** são feitas as considerações finais, apresentadas as contribuições, além de sugestões de trabalhos futuros e uma seção de discussão.

No **Apêndice A** são mostradas as tabelas com todos os valores dos experimentos realizados nos Capítulos 5, 6 e 7.

CAPÍTULO 2

KDD – KNOWLEDGE DISCOVERY IN DATABASE

2.1 – O PROCESSO DE KDD E MINERAÇÃO DE DADOS

A descoberta de conhecimento em base de dados (KDD) é um processo não trivial de identificação de padrões válidos nos dados. KDD é o processo de descoberta de novos conhecimentos, que não são óbvios ou de fácil identificação, como: tendências, regras de associação, probabilidades ou acontecimentos. As etapas do processo de KDD são mostradas pela Figura 2.1. Mineração de dados (*Data Mining*) é um termo genérico utilizado para métodos e técnicas computacionais visando à extração de informações úteis de um grande volume de dados.

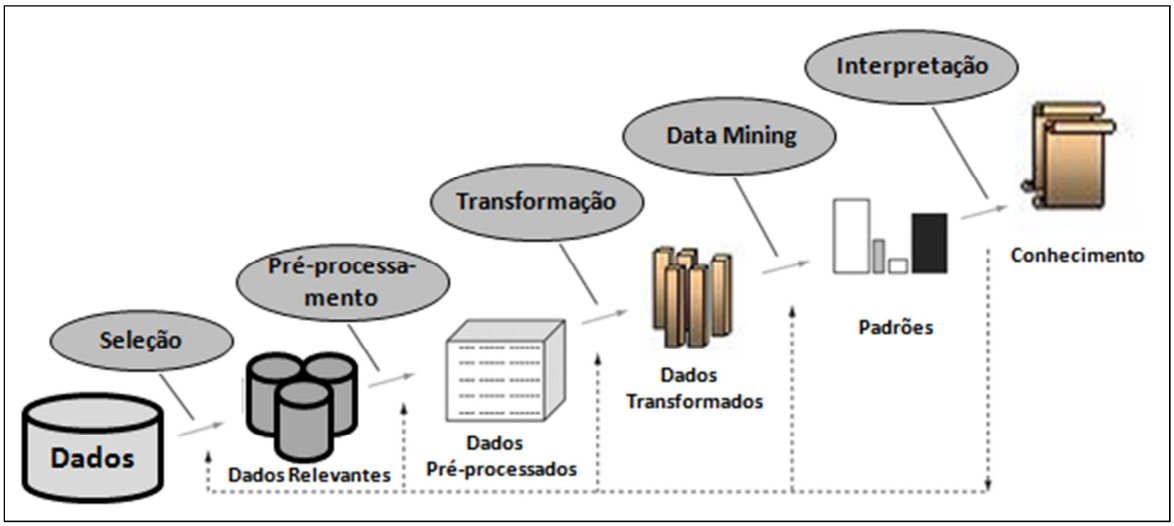


Figura 2.1 – Etapas do Processo KDD (Adaptado) [Fay96].

Seleção: nesta fase é necessária a compreensão do domínio e dos objetivos da tarefa, para obtenção dos dados necessários para o processo de KDD.

Pré-processamento: etapa em que é feita a análise dos dados para determinação do nível de ruído, ocorrência de *outliers*, inconsistência e incompletude nos dados. É uma das etapas mais demoradas e importantes do processo.

Transformação: aqui é feita a redução de dimensionalidade, combinação de atributos e toda a adequação necessária nos dados para utilização das técnicas na fase de Mineração de Dados.
Mineração de Dados: nesta etapa são aplicados os algoritmos que possibilitarão o processo de extração de conhecimento das bases de dados, por meio de tarefas de agrupamento, classificação, previsão e extração de regras.
Interpretação: na interpretação dos resultados, toda a avaliação de desempenho é feita de forma a interpretar os resultados e determinar se o resultado já é aceitável, ou se é necessário o retorno aos passos anteriores.

O KDD também pode ser explicado utilizando-se três fases principais [Fay96]. A preparação da base de dados ou pré-processamento, a Mineração de Dados e o pós-processamento. A fase de pré-processamento de dados envolveria as três primeiras etapas do processo de KDD anteriormente citadas e o pós-processamento seria equivalente à etapa de interpretação.

Quando o pós-processamento é carregado em um processo analítico exploratório, similar ao utilizado em *OLAP*, ele é chamado de *On-line Analytical Mining* (*OLAM*). A etapa de Mineração de Dados faz uso de técnicas da Inteligência Computacional e estatísticas tais como: Redes Neurais, Algoritmos Genéticos, Regressões Estatísticas, Lógica Difusa/Nebulosa e Inteligência Artificial simbólica para realizar atividades de agrupamento (identificação de grupos de indivíduos/registros que têm perfis semelhantes), regressão (estimação de valores contínuos na resposta do sistema), classificação (decisão do sistema com resposta no domínio discreto) e extração de regras (representação de relações entre variáveis do processo).

O processo de KDD possui a premissa de uma argumentação ativa, pois o usuário define o problema, seleciona os dados e as técnicas de Mineração de Dados, pesquisam automaticamente na busca por padrões, anomalias e possíveis relacionamentos nos dados que podem representar problemas ou oportunidades escondidas nestes relacionamentos.

A Mineração de Dados é considerada a principal fase do processo de KDD. Essa fase é exclusivamente responsável pelo algoritmo minerador, ou seja, o algoritmo que diante da tarefa especificada, busca extrair o conhecimento implícito e potencialmente útil dos dados. A Mineração de Dados é, na verdade, uma descoberta eficiente de informações válidas e não óbvias de uma grande coleção de dados [Big96]. Durante a fase de Mineração de Dados, necessita-se definir a técnica e o algoritmo a serem utilizados em função da tarefa proposta. A Tabela 2.1 mostra as principais tarefas de KDD e as técnicas mais utilizadas para Mineração de Dados.

Tabela 2.1 – Principais tarefas de KDD e suas técnicas de Mineração de Dados [AVL99].

Tarefas de KDD	Técnicas
Associação	Indução de Regras, Estatística e Teoria dos Conjuntos
Classificação	Algoritmos Genéricos, Redes Neurais e Árvores de Decisão
Clustering	Redes Neurais e Estatística
Predição de Séries Temporais	Redes Neurais, Lógica Nebulosa e Estatística

Uma vez escolhido o algoritmo a ser utilizado, deve-se adaptá-lo ao problema proposto e implementá-lo por intermédio de uma ferramenta existente ou pelo desenvolvimento específico de uma nova ferramenta. A partir deste ponto, inicia-se o processo de Mineração de Dados, onde serão apresentados diversos padrões, que serão interpretados para geração do conhecimento.

É importante salientar que, quando se fala em Mineração de Dados, não se considera apenas consultas complexas e elaboradas (*OLAP*), que visam ratificar uma hipótese gerada por um usuário em função dos relacionamentos existentes entre os dados, mas principalmente a descoberta de novos fatos, regularidades, restrições, padrões e relacionamentos pouco visíveis.

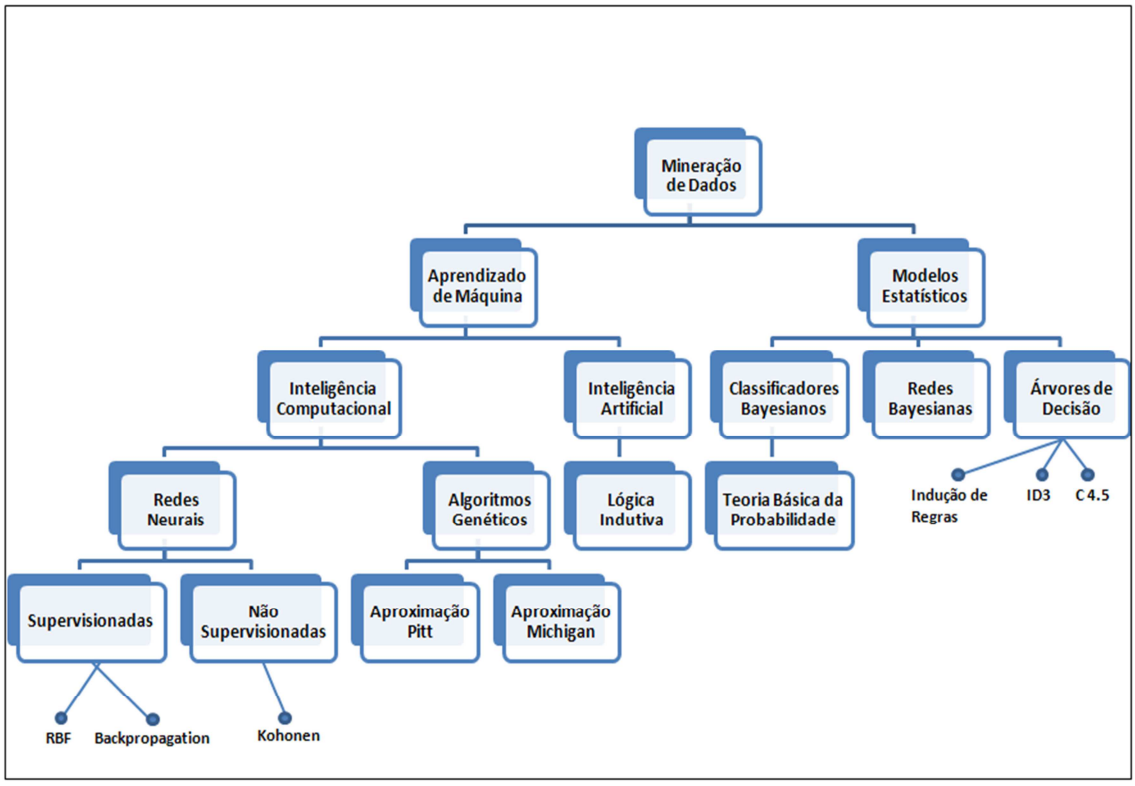


Figura 2.2 – Taxonomia das Técnicas de Mineração de Dados [AVL99].

Na mineração dos dados, o executor da tarefa pode utilizar várias ferramentas e técnicas para que o seu objetivo seja bem sucedido. Esta fase envolve diversas áreas e técnicas, além dos

principais algoritmos. A Figura 2.2 mostra uma taxonomia da fase de Mineração de Dados. Exemplos de algoritmos estão representados pelo símbolo (●), enquanto que as caixas representam áreas e técnicas.

2.2 – PRÉ-PROCESSAMENTO DE DADOS

A exploração e análise de dados são elementos importantes que facilitam a preparação dos dados. Um dos seus pré-requisitos é conseguir um bom entendimento e a uma boa visualização, o que leva quase sempre à necessidade de fazer algum uso de operações sobre os dados [Pyl99]. Tais operações podem ser realizadas utilizando técnicas estatísticas e de Mineração de Dados. Com a visualização dos dados é possível fazer uma boa avaliação, possibilitando o emprego das técnicas adequadas de tratamento dos dados.

Muitos problemas encontrados na qualidade dos dados ocorrem pela fragilidade dos critérios de verificação na entrada da informação. Em muitos casos, as aplicações não prevêem os possíveis erros na digitação ou carga de dados nos Sistemas Gerenciadores de Banco de Dados (como datas inválidas ou referência a dados não cadastrados na tabela de origem) e permitem o armazenamento de informações sem qualidade. Após a identificação da origem do erro da informação, as regras envolvidas no processo de armazenamento do dado devem ser revistas e ajustadas, ou até mesmo reescritas, para solucionar o problema.

As rotinas de pré-processamento envolvem todo processo de transformação dos dados visando torná-los legíveis para análise (exemplo normalizar um campo) e compatibilizá-los para aplicação dos algoritmos de Mineração de Dados.

De maneira geral, pode-se dizer que os motivos que levam os dados a serem transformados nas rotinas de pré-processamento são: baixo grau de correção (dados incorretos), abrangência inadequada (não atende às necessidades do problema), falta de consistência (dados conflitantes), baixo grau de completeza (dados incompletos ou nulos), incoerência com regra de negócio e precisão inadequada.

A seguir serão apresentadas algumas técnicas comuns para tratamento e transformação de dados. Para melhor entendimento, considera-se que o tratamento de dados pode ser feito por variável individualmente ou em conjunto com outras.

A *Normalização* tem o propósito de minimizar os problemas oriundos do uso de unidades e dispersões distintas entre variáveis. As variáveis podem ser normalizadas segundo a amplitude ou segundo a distribuição dos seus valores. A normalização segundo a amplitude deve ser usada

quando existem unidades diferentes ou dispersões muito heterogêneas. A normalização distribucional é interessante em situações como: remoção de distorções de valores aberrantes e obtenção de simetria. Normalizações são muito usadas para beneficiar a fase de mineração em algoritmos como: Redes Neurais, Algoritmos Genéticos, *KNN*, *clustering* entre outros [Fay96] [Pyl99] [HaK01] [WiF05].

A *Discretização* dos dados faz com que o domínio dos atributos seja reduzido, possibilitando a melhor compreensão e/ou visualização dos dados [Pyl99] [WiF05]. Também conhecida como categorização, e muito importante no pré-processamento para algumas técnicas de Mineração de Dados que trabalham apenas com dados categóricos, faz uma espécie de agrupamento, por exemplo, transformando um campo idade numérico em faixas de idade.

A técnica de *Agrupamento Outros* consiste em agrupar numa única classe as ocorrências de menor frequência. É muito utilizada para melhorar tanto a visualização das classes mais relevantes quanto o aprendizado em Mineração de Dados, valorizando as classes de maior ocorrência [Pyl99] [WiF05].

O algoritmo de *Similaridade (Matching)* visa procurar ocorrências que deveriam ser iguais e por algum problema na entrada de dados (erro de digitação, por exemplo) foram cadastrados de maneira diferente. Por intermédio de uma regulagem de precisão, pode-se dar pesos à comparação entre duas cadeias de caracteres (*strings)* e determinar se são a mesma ocorrência ou não. Desta forma, muitos campos como endereço, nome e descrições em geral podem ser beneficiados com esta técnica [San02].

A atribuição de colunas erradas ou nulas pode ser feita através da aplicação de um banco de conhecimento previamente construído para esta função. Ela aproveita campos que estão com uma boa qualidade de preenchimento e define regras para complementação de outros campos com baixo nível de preenchimento ou confiabilidade. Como exemplo, pode-se verificar que uma coluna pode determinar o preenchimento da outra com um grau de certeza muito grande, como acontece com variáveis do tipo: endereço, CEP, cidade e UF. Então, tendo a variável cidade preenchida corretamente e um banco com todas as possíveis cidades e seu respectivo estado, é possível com certo grau de acerto determinar o estado correspondente. Outra maneira de utilizar essa técnica é guardar todas as ocorrências certas ou as anomalias de um atributo e o seu valor correto, assim as ocorrências podem ser comparadas a uma base de dados e o valor correto a ser substituído. O sexo é uma variável que pode ser criada com um bom nível de certeza partindo-se apenas de um bom banco com nomes e seus respectivos sexos.

A *Codificação de dados* consiste basicamente em transformar dados discretos em numéricos. Essa codificação pode ser de valores discretos para valores binários, para o código termômetro ou outra representação numérica compatível com os algoritmos de aprendizagem [Pyl99]. Seu maior uso é para pré-processar dados que servirão de entrada para técnicas de Mineração de Dados que trabalham apenas com dados numéricos, como Redes Neurais e Regressão Logística, por exemplo.

Existem outras possibilidades que não se enquadram nas técnicas citadas e são muito usadas, mas não existe uma denominação formal conhecida. Por exemplo, às vezes é necessário fazer uma divisão de informação (campo), como no caso, da variável CEP, que pode ser agrupada em dígitos mais significativos para criar grupos mais representativos de localidades maiores. O inverso também é verdadeiro, pois muitas vezes deseja-se a junção de dois ou mais campos de maneira matemática (renda declarada + renda não declarada = renda total), interativa ou de outra forma. As possibilidades são ilimitadas porque o tratamento de dados pode ser obtido através da aplicação de diversas técnicas em sequência. Assim, torna-se difícil fazer uma classificação, visto que algumas técnicas são na verdade a especialização de outras mais genéricas ou correspondem à aplicação de duas ou mais técnicas em conjunto.

O conhecimento da potencialidade destas técnicas poderá ajudar o projetista de Mineração de Dados a manipular de maneira a extrair o máximo potencial disponível, auxiliando assim na interpretação e tratamento das anomalias, na validação de regras de negócio, na descoberta de novas características, e até mesmo na elaboração de formulários mais eficientes para entrada de dados operacionais. Nesta tese foram utilizadas algumas destas técnicas e estas são citadas nos capítulos que descrevem os experimentos.

2.3 – TÉCNICAS DE PREDIÇÃO ESTATÍSTICAS E DE INTELIGÊNCIA ARTIFICIAL

Se o ser humano é capaz de reconhecer o rosto de outra pessoa, fazer classificações, enfim, desenvolver atividades que demonstrem um grau de inteligência, a Ciência da Computação investe na máquina para tentar aproximá-la cada vez mais do modelo humano, imitando o seu comportamento no que diz respeito ao seu aprendizado, percepção, raciocínio, evolução e adaptação.

A Inteligência Computacional, disponibilizando suas técnicas, permite às máquinas utilizarem sistemas que auxiliem nas tomadas de decisões, reconhecimento de imagens, controle e automação industrial, entre outras aplicações, contribuindo para serviços mais eficientes e velozes. Nas últimas décadas, diversas aplicações simuladas e práticas [DCO96] [ABO00]

[Rud01] [ChH03] [HaB03] [KHR03] [MaM03] [AVA04] [FER06] [QiR07] vêm surgindo utilizando tais técnicas, o que torna a Inteligência Computacional um elemento cada vez mais próximo no nosso dia-dia.

Para esse estudo, considera-se relevante as funcionalidades típicas de Inteligência Computacional para classificação e regressão, em especial as que utilizam aprendizado supervisionado. Existe uma literatura com enorme variedade de técnicas de aprendizado supervisionado, dentre elas pode-se citar: *Naive Bayes* [DHS01], Redes Neurais [HSW89] [Hay98] [BCL07], Algoritmos Genéticos [RuN95] [HaK01] [WiF05], *k-Nearest Neighbors* (*kNN*) [RuN95] [DHS01] [HaK01] [WiF05], Árvores de Decisão [RuN95] [HaK01] [WiF05], Regressão Linear [Net96] [JoW98] [DHS01] e Regressão Logística [HoL89] [JoW98] .

2.3.1 – REDES NEURAIS ARTIFICIAIS

As Redes Neurais Artificiais (RNA) já estão presentes em diversos segmentos do mundo real como, por exemplo, na indústria, automatizando ou otimizando partes de processos produtivos; ou em segurança de sistemas informatizados, atuando, por exemplo, em "*firewalls* inteligentes", detectando e frustrando tentativas de invasão a redes de computadores. Mas elas também estão sendo utilizadas amplamente pelo mercado financeiro em aplicações como: medição de riscos de crédito, estratégia de cobrança, detecção de fraudes, previsão de riscos de sinistros, e até para antecipar tendências em bolsas de valores e mercadorias [FER06]. Conhecida como um aproximador universal de funções, uma Rede Neural pode efetuar operações lógicas e matemáticas, tendo como base uma somatória ponderada dos vários neurônios que recebem e processam as informações de entrada [HSW89].

Neste trabalho optou-se pelo modelo de Rede Neural *Multilayer Perceptron (MLP)* treinada com o algoritmo de *Backpropagation* [Hay98] [RuW98] e *Levenberg-Marquardt* [Hay09]. O motivo da escolha do primeiro algoritmo deve-se à sua utilização com sucesso em aplicações de problemas de classificação de padrões. O segundo algoritmo foi escolhido porque alguns autores descrevem-no como sendo mais poderoso que as técnicas convencionais de gradiente descendente [Elb03] [HaM94]. Dentre as características mais atrativas deste tipo de Rede Neural é possível destacar a excelente capacidade de generalização e a simplicidade de operação da rede.

2.3.2 – REGRESSÃO LINEAR NORMAL

Este tipo de Regressão Linear é a mais utilizada em análises de dados e geralmente é aplicada quando a variável dependente é numérica e contínua. A Regressão Linear estima o valor

esperado da variável dependente condicionado às observações das variáveis preditoras. É suposto que os erros tenham distribuição normal [Net96] [JoW98] [DHS01].

2.3.3 – REGRESSÃO LOGÍSTICA

A Regressão Logística é usada em problemas em que a variável dependente assume valores categóricos. Para este tipo de variável não é apropriado que se use a Regressão Linear convencional, uma vez que seus valores não pertencem a um espaço métrico de escala numérica e os erros de ajuste não são normalmente distribuídos. Enquanto a Regressão Linear convencional modela o valor esperado da variável resposta, condicionado aos valores das variáveis preditoras, a Regressão Logística modela a probabilidade da variável resposta assumir cada uma de suas categorias [HoL89] [JoW98].

Este trabalho, foca em um caso particular de modelo de Regressão Logística, onde a variável resposta assume valor dicotômico, ou seja, duas categorias possíveis.

2.3.4 – ÁRVORES DE DECISÃO

Uma Árvore de Decisão é um classificador que utiliza sucessivas partições no conjunto de dados e as representa de forma gráfica através de uma árvore. Esta árvore é representada por um nó raiz, que possui o conjunto completo dos dados, e ramos com outros nós que representam partições disjuntas do nó raiz. Estes outros nós podem ou não gerar mais ramos com outros nós. Os nós que não mais geram ramos são chamados folhas (ou nós terminais). Os demais são chamados nós internos. As Árvores de Decisão são criadas dividindo-se recursivamente os nós, utilizando funções dos atributos do conjunto de dados que geram dois ou mais ramos. Para maiores detalhes sobre este algoritmo recomenda-se a leitura de artigos e livros especializados [Qui86] [Qui93] [MaR05].

2.4 – MÉTODOS PARA AVALIAÇÃO DE DESEMPENHO

A estatística oferece ao processo de KDD diversas ferramentas que ajudam em suas várias fases. Para a amostragem dos dados de entrada no processo de KDD, utilizam-se técnicas estatísticas de amostragem, a fim de garantir a representatividade da massa bruta de dados. Técnicas como validação cruzada *(cross-validation)* [Pre94] são aplicadas para planejamento de experimentos auxiliando a medição do erro. Para medição de desempenho, a estatística oferece inúmeros métodos que podem ser usados para comparação de resultados entre modelos como: o teste de hipóteses com intervalo de confiança e a análise de correlação. A estatística ainda contribui com algumas técnicas de aprendizado muito utilizadas pela Mineração de Dados (algumas já

citadas). A seguir são mostradas algumas das contribuições para a análise de desempenho, incluindo as que serão utilizadas nesse trabalho.

2.4.1 – VALIDAÇÃO CRUZADA

Cross-validation (ou validação cruzada) é um método utilizado para garantir a independência estatística do conjunto de teste, aumentando a confiabilidade dos resultados. Este método consiste na divisão dos dados em *k* partes iguais (*k-fold*), onde cada parte é usada para teste e o restante (*k*-1) para treinamento. Desta forma, *k* experimentos são realizados com seus respectivos resultados. A vantagem na utilização da validação cruzada é que uma grande parcela do conjunto dos dados é usada para treinamento [(*k*-1)/*k*] e todos os dados são usados para teste [Die98] [Con99].

2.4.2 – TESTE DE HIPÓTESES

Testes de Hipóteses são procedimentos que têm por objetivo interpretar, com base em uma probabilidade, a significância de uma estatística observada em uma amostra. São formuladas hipóteses sobre o estado de variáveis em um universo amostral. Duas hipóteses excludentes são avaliadas: *Hipótese de nulidade* (H_0), a qual supõe que o valor esperado do critério do teste é nulo (e.g., que a correlação entre duas variáveis é nula, ou que os grupos não diferem). Esta é a hipótese realmente avaliada. A *Hipótese alternativa* (H_1), que refere-se ao oposto de H_0 é a hipótese aceita se H_0 for rejeitada. Dois tipos de erros estão associados às hipóteses: ERRO TIPO I: Pr (Rejeitar H_0, dado que H_0 é verdadeira) e ERRO TIPO II: *Pr* (Aceitar H_0, dado que H_0 é falsa). Existem ainda outros tipos de testes de hipóteses comumente usados para comparar os erros de classificação, são eles: teste *t-Student* para amostras pareadas e teste *t-Student* para amostras independentes [Con99]. Recentemente, os testes não paramétricos *Qui-Quadrado* de *McNemar* e *Kappa* [She97] [Die98] [Con99] têm sido usados para medir a diferença entre classificadores.

2.4.3 – KOLMOGOROV SMIRNOV

Kolmogorov e Smirnov (KS) desenvolveram um procedimento estatístico que usa a máxima distância entre duas funções como uma medida de separação entre elas [Con99] [MTW03]. O objetivo do teste é medir se as distribuições de probabilidade das duas populações são estatisticamente diferentes. Este teste é não paramétrico, pois não há suposição sobre a distribuição de probabilidade dos dados. O *KS* é geralmente utilizado para medir a aderência de dados a uma distribuição. Em sistemas decisórios, bem como neste trabalho, ele serve para medir a separabilidade entre duas distribuições a partir da função de distribuição acumulada de

cada uma delas. O *KS* mede a máxima distância entre duas funções como uma medida de separação e pondera os erros pela probabilidade de ocorrência de cada classe.

Em resumo, o teste *Kolmogorov-Smirnov* é usado para determinar se duas distribuições de probabilidade subjacentes diferem uma da outra ou se uma das distribuições de probabilidade subjacentes difere da distribuição em hipótese, em qualquer dos casos, com base em amostras finitas. Na prática, esta medida é muito utilizada para avaliação e comparação de modelos de risco baseados em respostas dicotômicas [ThC02] [May04].

2.4.4 – MEDIDA *ROC*

A medida *ROC* (*Receiver Operating Characteristics*) é muito utilizada na medição do desempenho de sistemas de classificação binária. Por se basear nas taxas de erro do tipo falso positivo ou falso negativo, essa curva sempre foi muito utilizada na medicina e análises clínicas onde metodologias e tipos de exames são julgados quanto à sua capacidade de acerto e falsos positivos. Recentemente, a medida *ROC* vem sendo muito utilizada em Mineração de Dados na comparação entre classificadores [Faw06].

Para os cálculos, considera-se um problema de classificação binária. A predição de um determinado indivíduo pode pertencer a uma das quatro possíveis situações: indivíduo de caso positivo que foi predito como positivo (verdadeiro positivo), indivíduo negativo classificado como negativo (verdadeiro negativo), ambos considerados acertos do sistema de classificação; indivíduo negativo classificado como positivo (falso positivo) e indivíduo positivo classificado como negativo (falso negativo). Após a classificação, calculam-se as contagens destas situações descritas e organizam-se os valores em uma tabela chamada matriz de confusão. Além das contagens, algumas taxas são calculadas para que se possa chegar à medida *ROC*. A Figura 2.3 ilustra uma matriz de confusão e as medidas necessárias para o cálculo da medida *ROC*.

Estas medidas geram um par de pontos (x, y) que desenhado em um gráfico pode ser comparado com outros pares de pontos de outros classificadores. Cada ponto nomeado é relativo a um classificador. A medida *ROC* é definida como a distância euclidiana do ponto até o ponto (1, 0), onde seria o ponto da classificação perfeita, ou seja, acerto total dos positivos e nenhum negativo classificado de maneira incorreta. De maneira análoga, pode-se calcular a distância do ponto à reta x=y, onde a taxa de verdadeiros positivos é igual à de falsos positivos, representando classificadores que predizem os valores baseados em aleatoriedade, ou seja, classificadores sem opinião ou inúteis.

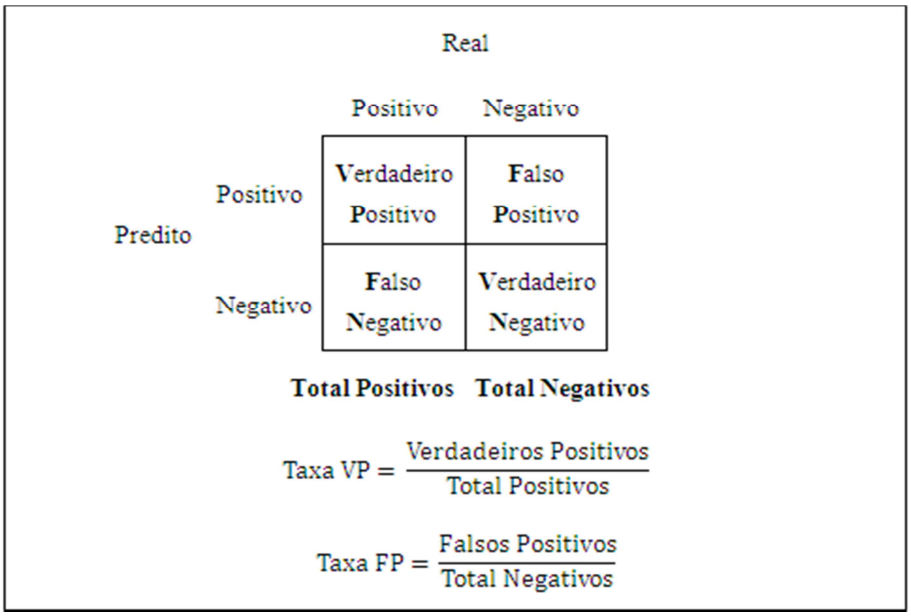


Figura 2.3 – Matriz de Confusão e métricas extraídas [Faw06].

A Figura 2.4 ilustra um exemplo de quatro classificadores (A, B, C e D), com seus respectivos pontos (x, y) desenhados no gráfico, em relação à reta $x=y$. Nota-se que o classificador D é o que mais se aproxima do ponto ótimo (1, 0), mostrando-se melhor do que os outros presentes. Classificadores abaixo da linha diagonal são considerados piores que a classificação aleatória. Estes classificadores possuem tendência a errar mais que acertar. Uma estratégia comum quando se obtém um destes classificadores é invertê-lo, ou seja, pegar sua predição e negá-la [Faw06]. Com esse processo, a negação de B passa a ser um classificador no triângulo superior definido pela diagonal.

Alguns classificadores como Árvores de Decisão, por exemplo, têm como resultado um valor predito. Este tipo de classificador gera um ponto para plotagem no plano *ROC*. Já outros classificadores como Redes Neurais, Regressões, entre outros, produzem um escore que dá uma ideia de quão próximo o indivíduo está de uma classe. Este escore pode ser visto diretamente como a probabilidade, caso se trate de regressão. Qualquer que seja a interpretação direta do escore é necessária uma nota de corte para classificar o indivíduo. Os classificadores que possuem esta natureza de construção podem gerar *n* pontos no plano *ROC*, um para cada ponto de corte escolhido. O desenho destes pontos no plano é chamada de curva *ROC*.

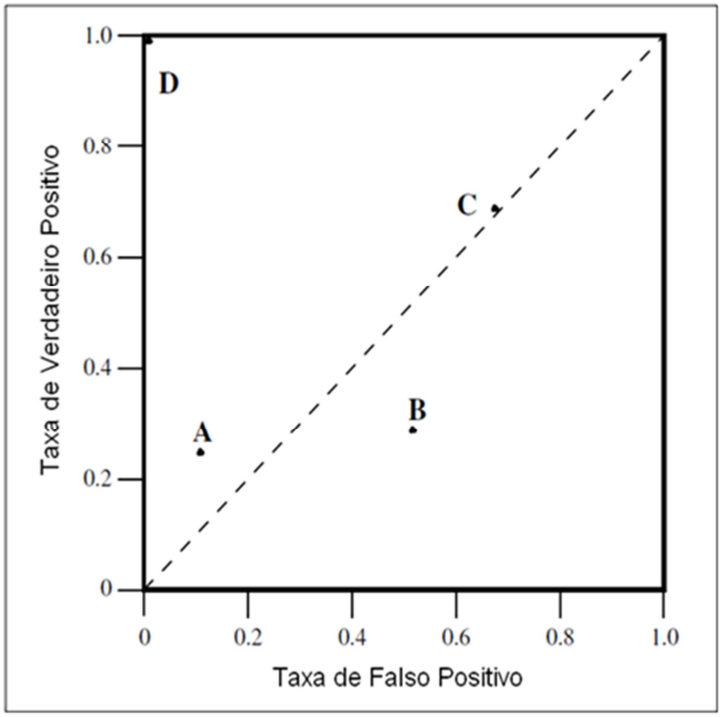


Figura 2.4 – Gráfico de pontos de quatro classificadores no plano *ROC*.

Classificadores que geram escore precisam ser comparados por intermédio de curva e não pontualmente, pois o ponto de corte (valor utilizado para separar o escore em duas classes) tem forte influência. Neste caso, a medida *ROC* utilizada para comparar os classificadores é a menor distância da curva ao ponto ótimo (1,0). A Figura 2.5 ilustra esta comparação. A curva superior, por estar mais próxima do ponto ótimo, é a curva do classificador considerado com melhor desempenho.

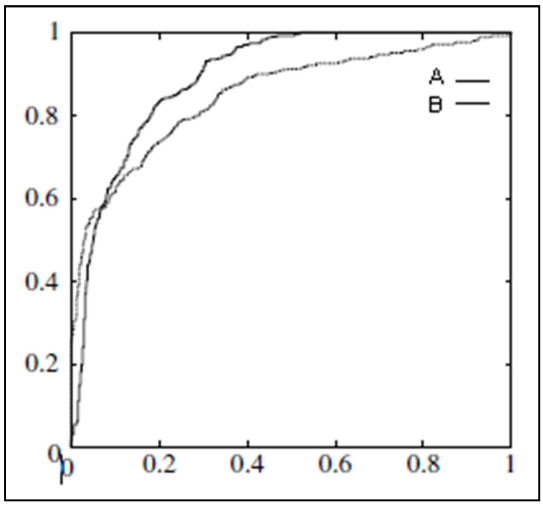


Figura 2.5 – Gráfico ilustrativo de duas curvas *ROC* para comparação.

2.5 – CONSIDERAÇÕES FINAIS

Neste capítulo são apresentados alguns dos conceitos fundamentais sobre o processo de KDD. Também são mostradas três técnicas de predição utilizadas nos experimentos e a técnica de Árvore de Decisão, que é importante para entendimento do método de segmentação proposto. Ainda foram discutidos os métodos de avaliação de desempenho necessários para compreensão do trabalho e utilizados nos Capítulos 5, 6 e 7. No próximo capítulo são descritos os principais métodos de combinação de preditores.

CAPÍTULO 3

COMBINAÇÃO DE PREDITORES

Quando necessita-se tomar decisões críticas, o natural é consultar a opinião de diversos especialistas sobre o assunto em questão. Após essas consultas, a decisão final deve ser feita com base em uma combinação das diversas opiniões dos especialistas. Fazendo uma analogia com Mineração de Dados, os especialistas seriam as diferentes técnicas de aprendizado que, combinadas, resultariam na decisão final. Aspectos como complexidade, número de exemplos e quantidade de variáveis devem ser levados em consideração para selecionar as técnicas que são mais adequadas à resolução do problema.

A combinação de métodos consiste em utilizar diversos classificadores e utilizá-los de forma integrada, a fim de melhorar o desempenho da classificação. Esta metodologia também é utilizada para estudo da robustez de alguns classificadores, ou seja, verificar o comportamento dos mesmos quando replicados em diferentes amostras de um conjunto de treinamento. Outra motivação para o uso desta técnica é combinar classificadores que apresentam diferentes tipos de comportamento e que dependem das características dos dados utilizados. Esta combinação visa contornar problemas de viés e variância que alguns classificadores podem apresentar [Rok05].

Os métodos mais conhecidos para criação de preditores múltiplos [Mit97] [WiF05] são: *Bagging* [Bre96], *Boosting* [Sch90] [Fre95] e *Stacking* [Wol92] [HaK01]. A segmentação [MaR05] também é um método de gerar múltiplos classificadores e vem se tornando cada vez mais comum nos modelos de decisão nas empresas (por exemplo: *credit e behavior scoring*). Como dito anteriormente, é consenso que esses métodos frequentemente melhoram o desempenho das decisões, porém não é fácil saber analisar que fatores individuais de cada técnica estão contribuindo para a melhoria das decisões. Estes métodos serão revisados nas próximas seções.

O uso destes métodos de combinação, normalmente, apresenta bons resultados quando existe grande diversidade entre os preditores [BWH05]. A diversidade existe quando diferentes classificadores possuem diferentes variâncias e diferentes vícios em relação ao treinamento. A combinação destes classificadores de maneira adequada maximiza o poder de predição ou outra medida de desempenho [BWH05] [Rok05].

A diversidade pode ser construída utilizando-se o mesmo classificador diversas vezes, porém alterando de certa maneira o conjunto de treinamento ou parâmetros dos classificadores. Esta construção pode ser separada em duas formas: implícita e explicita. A técnica *Bagging*, por exemplo, é dita ser implícita, pois modifica o conjunto de treinamento aleatoriamente por meio de re-amostras. Nenhuma técnica é aplicada para forçar diversidade entre os classificadores, ou seja, a diversidade pode existir apenas se os ruídos gerados em cada re-amostra forem suficientes para alterar o comportamento dos classificadores. Já o *Boosting* é um método explícito para obter classificadores com diversidade, pois altera o conjunto de treinamento de modo a obter diversidade a cada iteração.

3.1 – O PROCESSO DE CRIAÇÃO DOS CLASSIFICADORES MÚLTIPLOS

Em [RoG01] é descrito um processo de criação de classificadores múltiplos, em que grandes etapas são definidas e parte da descrição do método proposto é baseado nelas. Nesta seção é feita uma breve revisão deste processo.

A criação de classificadores combinados é feita em quatro grandes fases: determinação de vários classificadores que apresentam diversidade entre si; escolha dos classificadores que serão usados na combinação; determinação da função de combinação, por exemplo, votação, média ponderada, entre outras; e avaliação do desempenho do classificador múltiplo. Na fase final do processo, após a avaliação, pode ser necessária a volta para fases anteriores para a determinação de outros classificadores, nova função de combinação, até que se obtenha o desempenho desejado. É um ciclo análogo ao ciclo de criação de um classificador simples, onde se vê a necessidade de voltar ao início, a seleção de variáveis, até que se alcance o desempenho de interesse [RoG01] (Figura 3.1).

Figura 3.1 – Ilustração do ciclo sugerido de criação de um classificador múltiplo [RoG01].

3.1.1 – GERAÇÃO DOS CLASSIFICADORES

A primeira fase do processo é a determinação de como construir o conjunto de classificadores, de forma que apresentem diversidade entre si. *Bagging* e *Boosting* são exemplos de métodos mais utilizados para construção de tal conjunto. Cada técnica possui uma filosofia básica para a construção do conjunto de classificadores. Estas filosofias podem ser agrupadas em três tipos [BWH05]:

- *Starting Point in Hypothesis Space* (ponto de partida no espaço de hipóteses) são técnicas pertencentes à primeira categoria são as que criam diferentes classificadores modificando seus parâmetros inicias. Redes Neurais são os indutores que mais se utilizam destas técnicas. A técnica consiste em criar, por exemplo, várias Redes Neurais com diferentes pesos iniciais. Estes pesos podem ser gerados aleatoriamente (método implícito de diversidade), ou serem forçados a seguir certa regra, para que haja diversidade (explícito). Apesar de bastante utilizada, esta técnica não apresenta grandes resultados, segundo experimentos realizados [BWH05].

- *Set of Accessible Hypothesis* (conjunto de hipóteses acessíveis) é a segunda categoria e encontra-se entre as técnicas que alteram o espaço de hipóteses, ou seja, alteram o conjunto de treinamento em questão, ou ainda as arquiteturas dos classificadores. A alteração do conjunto de treinamento pode ser feita de diversas maneiras. As técnicas de *Bagging* e *Bossting* se encaixam nesta categoria, uma vez que modificam o conjunto de treinamento para cada um dos classificadores presentes no conjunto. Outra técnica que se encaixa nesta categoria é a segmentação do conjunto de treinamento em diversas partes disjuntas, utilizando regras de segmentação criadas a partir das variáveis preditoras. Cada classificador é treinado segundo conjuntos diferentes, ou seja, cada um "aprende" por meio de fontes diferentes, apresentando assim diversidade entre eles.

- *Transversal of Hypothesis Space* (transversal do espaço de hipóteses) altera a arquitetura dos classificadores mudando o algoritmo de aprendizagem de uma Rede Neural, por exemplo, ou até mesmo usando uma mistura de diferentes classificadores como Redes Neurais, Regressão Logística, Árvore de Decisão, etc. Neste caso, o conjunto de treinamento pode ser o mesmo, porém com diversas técnicas de regressão e classificação aplicadas no intuito de se obter diversidade. Diferentes pesos e penalidades podem ser aplicados de acordo com a especialidade de cada classificador.

3.1.2 – ESCOLHA DOS CLASSIFICADORES

Na segunda fase, os classificadores gerados na primeira fase são selecionados de forma a maximizar a métrica de desempenho escolhida. Em alguns casos, todos os classificadores gerados são utilizados como, por exemplo, nas técnicas de *Bagging* e *Boosting.* Eles possuem uma metodologia para a geração dos classificadores e, posteriormente, utilizam todo o conjunto gerado para formar o classificador final. Em outras metodologias, criam-se diversos classificadores e depois apenas o melhor conjunto é selecionado para compor o classificador múltiplo. O melhor conjunto é assim julgado por uma avaliação de desempenho, ou seja, o ideal é selecionar o melhor tal que maximize alguma métrica. Basicamente, existem inúmeras possibilidades de criação de subconjuntos de classificadores. Para encontrar o melhor subconjunto, podem-se testar todas as possibilidades e medir o desempenho. Obviamente que o custo computacional para esta busca é alto. Heurísticas ou algumas técnicas para restringir o número de possíveis escolhas são comumente utilizadas.

3.1.2.1 – HEURÍSTICA

Após a geração dos diversos classificadores, o conhecimento *a priori* sobre o problema pode ser empregado. Por exemplo, o número de classificadores a serem utilizados pode ser definido de antemão. Se um pesquisador conclui que o melhor conjunto é formado por quatro classificadores no máximo, a busca por este melhor conjunto se limita, evitando custos computacionais na tentativa e avaliação de conjuntos maiores que isto. Outro exemplo é a busca de um conjunto de classificadores que lidam com uma variável alvo de K classes. Se existe algum conhecimento *a priori* de quais classificadores são melhores para identificar quais classes, estes classificadores podem ser escolhidos a ponto de cobrir o máximo possível das classes com seus respectivos especialistas.

3.1.2.2 – MÉTRICAS DE DIVERSIDADE

Conforme visto anteriormente, quanto maior a diversidade entre os classificadores, maior a contribuição destes em um classificador múltiplo. Algumas métricas de diversidade dos erros

gerados por cada classificador são calculadas com o intuito de se obter o conjunto com maior diversidade. Pode ser feita uma análise, por exemplo, das variâncias e vícios encontrados por cada classificador individual. Um conjunto candidato a possuir diversidade é o conjunto composto por classificadores que apresentam variância e vícios distintos. Um conjunto de classificadores que possuem o mesmo vício de estimação e as mesmas variâncias é ruim porque acrescenta pouca ou nenhuma informação, pois este é formado por elementos muito homogêneos que apresentam respostas muito similares. Um conjunto mais heterogêneo de classificadores pode apresentar respostas muito diferentes, indicando assim que cada um tem especialidade em determinada parte do problema. Essa mistura de especialidades é que gera classificadores múltiplos mais consistentes que os simples.

3.1.2.3 – MÉTODOS DE BUSCA NÃO EXAUSTIVA

Como citado anteriormente, a busca exaustiva pelo melhor subconjunto pode ser custosa computacionalmente. Alguns métodos também focam em procura não exaustiva, pelo melhor conjunto de preditores. A investigação pode começar incluindo todos os classificadores e ir sucessivamente retirando classificadores do grupo. A cada iteração, as métricas de desempenho são calculadas e analisa-se se retirar aquele classificador é vantajoso. O processo termina quando nenhum classificador é candidato a sair, uma vez que todos os grupos com menos classificadores apresentam resultados inferiores ao grupo mais completo. Essa busca é chamada *Backward.* Outra forma possível é começar com apenas um classificador no grupo e ir adicionando sucessivamente novos e monitorar o ganho de desempenho. Essa busca é chamada *Forward*.

3.1.3 – FUNÇÃO DE COMBINAÇÃO (JUNÇÃO)

Após a etapa de escolha dos classificadores, vem a etapa da definição da função de combinação destes. É a função que relaciona as diversas “opiniões” de cada um dos classificadores. Os métodos de combinação dos classificadores mais conhecidos são [Rok05]:

- Votação: Cada classificador tem o mesmo peso (votação uniforme) e a resposta mais frequente entre eles é o resultado final para a classificação em questão. A votação pode também ser ponderada, ou seja, diferentes pesos são atribuídos para os votos;

- Combinação *Bayesiana*: é uma votação ponderada, onde o peso atribuído para cada classificador é a probabilidade *a posteriori* do classificador dado o conjunto de treinamento;

- Peso por Entropia: o peso atribuído a cada classificador é inversamente proporcional à entropia do vetor de classificação de cada um deles, ou seja, a ponderação de cada saída é feita de acordo com a medida de entropia calculada para aquele classificador;

- *Meta-combining*: métodos que utilizam resultados e aprendizados obtidos por meio de outros classificadores para se criar um novo, utilizando alguma técnica de classificação, como Árvore de Decisão, Redes Neurais, entre outras. Alguns autores chamam esta forma de combinação de *Stacking* [Wol92]. Devido à importância deste método, mais a diante ele será abordado em uma seção especial.

3.2 – ALGUMAS TÉCNICAS DE GERAÇÃO DE CONJUNTOS DE CLASSIFICADORES

3.2.1 – *BAGGING*

Bagging (*Bootstrap Aggregating*) foi proposto por Leo Breiman em 1994 com o intuito de melhorar decisões através da combinação de classificadores construídos a partir de conjuntos de amostras geradas aleatoriamente [Bre96]. Este método constrói vários classificadores a partir de diversas réplicas de um mesmo classificador e de distintos conjuntos de amostras de dados. Estas amostras são geradas a partir de um conjunto de dados de treinamento com *m* itens. Aleatoriamente são extraídos *m* exemplos com reposição a partir do conjunto original, ou seja, podendo ocorrer repetição de exemplos na amostra. Desta maneira, exemplos utilizados em uma amostra podem aparecer novamente em outra. Isto colabora para que os classificadores sejam diferentes, devido a esta variação de exemplos nas amostras.

A idéia geral é que este ruído causado pelas amostragens com reposição gere diversidade entre os classificadores e que a posterior combinação leve a um melhor desempenho. Esta combinação é feita por votação uniforme. Em outras palavras, a partir de *T* amostras B_1, B_2, ..., B_T, são construídos C_T classificadores (C_1, C_2, ..., C_T). Com a combinação destes classificadores é construído o classificador final C*, que supostamente é melhor que qualquer classificador C_i. Esta técnica além de servir como forma de construir classificadores múltiplos, pode ser usada para verificar a estabilidade de técnicas de classificação, ou seja, verificar a robustez da técnica quando uma parte da amostra de treinamento apresenta ruídos.

Conforme ilustra o Algoritmo 3.1, para o conjunto *S* de treinamento de tamanho *m* faz-se um laço (linha 1) para a geração de *T* re-amostras com substituição (linha 2). Para cada subconjunto de dados carregado em *S'* um classificador treinado é atribuído a *Ci* (linha 3). Após a geração de *T* classificadores, para determinação da classe de cada exemplo, submete-se o exemplo a todos os classificadores e aplica-se um método de combinação. Normalmente este

método é uma votação simples (linha 5), assim a classificação do exemplo é definida como o valor mais frequente.

Entrada: conjunto de treinamento S de tamanho m, classificador I, inteiro T (número de amostras do *bootstrap*).

1. Para $i = 1$ até T faça {
2 $S' =$ amostra de *bootstrap* de S (i. e., amostra com substituição)
3. $C_i = I(S')$
4. } fim para
5. $C^*(x) = \underset{y \in Y}{argmax} \sum_{i:C_i(x)=y} 1$ (o valor mais frequente predito para y)

Saída: classificador C*.

Algoritmo 3.1 – Algoritmo para *Bagging* [Bre96].

3.2.2 – *BOOSTING*

Boosting é um algoritmo de *meta-learning* criado para resolver problemas de aprendizado supervisionado [Sch90] [Fre95]. Este método faz parte da classe de algoritmos que alteram a distribuição do conjunto de treinamento, baseando-se no desempenho das classificações anteriores. Isto deve-se à característica básica do seu funcionamento, que é a geração sequencial de classificadores. A cada passagem, os pesos dos exemplos são alterados em função do sucesso de sua classificação (os exemplos que são classificados com erro têm seu peso aumentado). Por fim, após *n* passagens que são previamente definidas, é gerado um classificador final formado por um esquema de votação, sendo que o peso de cada classificador depende do seu desempenho no conjunto de treinamento em que ele foi construído. *Boosting* utiliza-se de todos os exemplos a cada passagem. Assim como o *Bagging*, o *Boosting* encontra-se na 2ª categoria dos métodos de geração de diversidade, ou seja, os métodos que alteram o conjunto de treinamento.

O problema mais comum encontrado na utilização deste método é *overfitting*. Para evitar este problema, recomenda-se que o número de iterações não seja muito alto [MaR05]. A mudança dos pesos e os critérios de paradas dependem do algoritmo utilizado. Um dos mais utilizados é o chamado *AdaBoost* [BaK98], que é mostrado no Algoritmo 3.2.

Conforme é visto no Algoritmo 3.2, o *AdaBoost* cria um conjunto de treinamento S' a partir de S dando pesos 1 para todas as instâncias (linha 1). Um laço faz a iteração T vezes para o seguinte ciclo (linha 2): primeiro gera-se um classificador I (linha 3), depois calcula-se o erro ponderado da classificação do conjunto de treinamento (linha 4), assim se o erro for maior do

que ½ tenta-se novamente com outro classificador (linha 5), senão é aplicado um peso maior para as instâncias classificadas erradas (linhas 6 e 7) e para finalizar o laço os pesos são normalizados (linha 8). Depois que iterar T vezes tem-se um classificador final (forte) ponderando as importâncias de cada classificador fraco (linha 10).

Entrada: conjunto treinamento S de tamanho m, classificador I, inteiro T (número de tentativas).

1. $S' = S$ com instância de pesos
2. Para $i = 1$ até T faça {
3. $C_i = I(S')$
4. $\epsilon_i = \frac{1}{m} \sum_{x_j \epsilon S' : C_i(x_j) \neq y_j} pesos(x)$ (erro ponderado no conjunto de treinamento)
5. Seja $\epsilon_i > 1/2$, conjunto S' para a amostra de bootstrap de S com peso 1 para qualquer instância e volta ao passo 3 (esta etapa está limitada a 25 vezes depois sai do loop).
6. $\beta_i = \epsilon_i / (1 - \epsilon_i)$
7. Para cada $x_j \epsilon S'$, se $C_i(x_j) = y_j$ então $peso(x_j) = peso(x_j) \cdot \beta_i$.
8. Normalizar os pesos das instâncias em relação ao peso total de S'.
9. }
10. $C^*(x) = \underset{y \in Y}{argmax} \sum_{i : C_i(x) = y} log \frac{1}{\beta_i}$

Saída: classificador C*.

Algoritmo 3.2 – Algoritmo AdaBoost [BaK98].

3.2.3 – SEGMENTAÇÃO

O grande objetivo da segmentação é dividir um problema complexo em diversos problemas de complexidade inferior para se obter uma solução melhor para o problema em questão. É uma forma de se obter subgrupos tais que técnicas de aprendizado possam ser aplicadas, resultando em diversos classificadores especialistas, ou seja, classificadores especializados em cada subgrupo gerado a partir da divisão do conjunto de dados inicial. Esta divisão pode ser feita de diversas formas. Existem filosofias diversas que discutem cada tipo de segmentação. Este trabalho tem um foco especial na segmentação obtida pela criação de regras que utilizam os atributos (variáveis preditoras) para dividir o espaço original em diversos subgrupos disjuntos que assumem valores diferentes para estes atributos.

Relacionando esta metodologia com o desenvolvimento de classificadores múltiplos, a segmentação da base original em subgrupos distintos e o desenvolvimento de classificadores

especialistas para cada um desses subgrupos é uma forma de gerar os diversos classificadores necessários no processo de construção do classificador múltiplo, ou seja, é o primeiro passo do processo. Esta maneira de se gerar o conjunto de classificadores está relacionada com o segundo grupo das estratégias de obtenção de diversidade de classificadores, pois, com a segmentação, altera-se o conjunto de treinamento, uma vez que para cada classificador existirá uma base de treinamento diferente. Diferentemente de técnicas como *Bagging* e *Boosting*, que geram diversidade, colocando ruídos no conjunto de treinamento ou usando diferentes pesos para as observações, respectivamente, a segmentação gera conjuntos disjuntos a partir da amostra de dados. Enquanto *Boosting* gera especialistas baseados nos erros de classificação, a segmentação gera especialistas para determinados valores de atributos, como ilustrado e formalmente apresentado a seguir:

Dado um conjunto de dados representados por: $\mathrm{Z} = (\vec{y}, \bar{X})$,

onde $\vec{y}$ é o vetor de resposta (alvo) e $\bar{X}$ matriz de atributos. [Verificar representação de X]
Sejam X_1, X_2,..., X_k subconjuntos mutuamente excludentes, de valores os quais $\bar{X}$ pode assumir.

Então, pode-se criar *k* segmentos mutuamente excludentes, a partir do conjunto original *Z*, da seguinte forma:

$\mathrm{Z}_i = (\vec{y}, \bar{X} \in X_i)$, $i = 1, ..., k$

A partir daí, são construídos *k* classificadores especialistas em seus respectivos segmentos, conforme mostra a Figura 3.2.

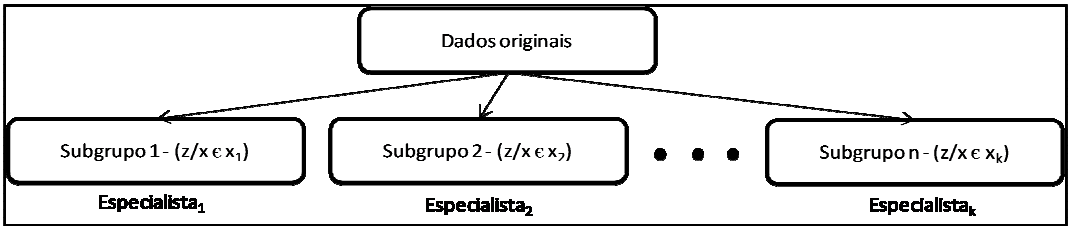


Figura 3.2 – Ilustração da divisão dos dados originais em *k* subgrupos disjuntos.

Por muito tempo as técnicas para encontrar uma estrutura de segmentação que agregasse valor limitaram-se ao conhecimento *a priori* do assunto em estudo. Estratégias subjetivas formavam a base da segmentação. Muito se usou da segmentação também para criar subgrupos de fácil interpretação, ou seja, criar classificadores específicos que por certo motivo fazia mais sentido que fossem separados. Dentre as técnicas subjetivas, também existe a divisão entre grupos que possuem várias informações (atributos) e grupos que não as possuem, ou estão com

nulos (*missing values*) [ThC02], pois este é um problema comum na modelagem de dados. Com o forte emprego destas técnicas subjetivas chegou-se a cogitar que não existiria uma metodologia para criar as estruturas de segmentação sem o uso de conhecimento subjetivo [MaR05].

Porém, alguns trabalhos sugerem métodos objetivos e algoritmos para encontrar uma estrutura de segmentação mais adequada e focada na melhoria da decisão final. Estes métodos consistem em representar a estrutura de segmentação em forma de árvores. Métodos exaustivos ou semi-exaustivos de busca pela melhor estratégia de segmentação são sugeridos [MaR05].

A busca exaustiva consiste em se testar todas as estruturas de segmentação possíveis, ou seja, todas as árvores que dividem o conjunto de dados em subgrupos. Esta estratégia passa a ser inviável quando se lida com grandes bases de dados com inúmeras variáveis. Uma alternativa seria o método semi-exaustivo por busca sequencial. Neste método uma árvore é construída de modo que o nó inicial é a base completa. Na primeira iteração, testam-se todas as regras de segmentação possíveis, ou seja, divide-se a base pelos valores ou intervalos possíveis de cada variável e sistemas de aprendizado especialistas são construídos. Utiliza-se uma função para combinar os classificadores e posteriormente é avaliado o desempenho do classificador múltiplo final. A estrutura de segmentação que apresentar melhor valor na medida de desempenho escolhida é selecionada. Na segunda iteração, cada nó terminal encontrado no primeiro passo é submetido à busca e uma nova regra de segmentação é encontrada para estes nós, como se estes fossem os nós iniciais.

Essa estratégia sempre visa melhorar o desempenho de cada nó terminal. Obviamente que este método pode não levar à estrutura ótima, pois não se testa casos em que nas primeiras iterações o ganho é pequeno, mas em divisões subsequentes poderia ser compensado com ganhos no desempenho final. Essas buscas locais pela melhoria de desempenho são feitas até que se encontre uma árvore de tamanho desejado ou não mais restem critérios de segmentação [MaR05]. Outro ponto de parada importante é o número de registros contidos no nó terminal. Se restarem poucos registros, pode não ser possível realizar o ajuste de uma Regressão Estatística ou Rede Neural, por exemplo.

Outra estratégia também proposta foi a criação de Árvores de Decisão utilizando critérios conhecidos como *Information Gain*, entropia, KS, entre outros [Nev98] [Nev99] [CGM02]. Como anteriormente discutido, as Árvores de Decisão buscam separar a população em várias populações distintas, mais homogêneas com relação ao comportamento da variável resposta. A motivação de se utilizar esta estratégia vem do fato de que populações muito distintas em

relação à variável alvo podem apresentar grandes diferenças nas estimativas dos parâmetros da regressão. Estes comportamentos distintos entre os subgrupos podem ser utilizados como fator de diversidade necessária para a criação dos classificadores múltiplos.

3.2.3.1 – VANTAGENS E DESVANTAGENS DA SEGMENTAÇÃO

Como discutido em [MaR05], as principais vantagens do uso da segmentação como forma de geração dos múltiplos classificadores são:

- Redução no volume de dados de grandes bases;
- Aumento da compreensão e interpretação dos resultados;
- Facilidade na manutenção do sistema de classificação;
- Permite computação paralela;
- Possibilita uso de diferentes técnicas de predição.

A primeira vantagem diz respeito ao mundo prático, onde bases de dados cada vez maiores são encontradas e precisam ser manipuladas e armazenadas. A segmentação reduz a dimensão da base original para bases menores, facilitando essa manipulação. Além disso, permite o cálculo em paralelo dos novos registros a serem classificados, uma vez que estes registros fazem parte de subconjuntos disjuntos.

A manutenção do sistema de classificação também é facilitada, pois, se por algum motivo for necessário treinar novamente o modelo, pode-se reestimar somente os segmentos necessários. Com isto, além da segmentação ser mais rápida, afetaria apenas alguns nichos da base de dados.

A compreensão do problema também pode ser mais fácil, uma vez que a decisão pode ser "dividida" em várias outras. Este aumento na facilidade é muito importante quando o objeto do estudo requer um foco especial na identificação de causas (variáveis) que levam a determinado alvo e não a classificação ou regressão propriamente dita. É claro que esta facilidade também é muito dependente dos métodos de predição utilizados na segmentação. Estudos na área de ciências humanas e biológicas têm interesse especial nesta característica, uma vez que precisam justificar ou propor novas teorias de causa e efeito. A segmentação também possibilita o uso de diferentes técnicas nos distintos segmentos. Por exemplo, para alguns segmentos aplica-se Regressão Linear, para outros Redes Neurais.

Apesar das diversas vantagens no uso da segmentação, o tempo de treinamento é mais alto se comparado com técnicas individuais, pois é necessário treinar diversos modelos para apenas um problema de classificação. Esta desvantagem aparece somente durante o treinamento dos segmentos, pois o tempo de submissão após o treinamento é similar ao de uma técnica

individual e menor do que os demais métodos de combinação apresentados neste trabalho. O aumento da quantidade de modelos a serem utilizados em uma classificação também aumenta a complexidade de disponibilização do modelo.

Como em todo método de combinação, não existem garantias de que a segmentação de modelos tenha resultados melhores do que a utilização de técnicas individuais ou de outros métodos de combinação.

Outro aspecto que deve ser abordado, quando utiliza-se segmentação, é quanto ao tamanho da amostra disponível, pois, como a segmentação exige a criação de subconjuntos disjuntos, esta não é uma técnica recomendada para pequenos volumes de dados, porque a divisão do conjunto de treinamento pode gerar quantidades de registros tão pequenas que não sejam suficientemente grandes para treinar classificadores com melhores desempenhos que o classificador da raiz de uma árvore de modelos.

Outra desvantagem dos métodos de segmentação, aliás, como também acontece nas técnicas de combinação de classificadores, é que, na prática, a existência de mais de um classificador implica no aumento da complexidade de disponibilização do modelo em produção. Isto acontece porque, quando o modelo precisa ser colocado em produção (rodando em tempo real, por exemplo), muitas vezes é necessário programar a sua implementação e interligação com os sistemas de informação e, quanto maior o número modelos, maior será o esforço necessário.

3.2.3.2 – *FRAMEWORK* DE SEGMENTAÇÃO

Conforme citado anteriormente, diversas técnicas para encontrar uma estrutura de segmentação podem ser aplicadas. As estruturas podem ser representadas por árvores e cada divisão representa uma regra envolvendo os valores das variáveis preditoras (atributos). A seguir é apresentado um *framework,* como proposta de algoritmo de busca sequencial por uma segmentação que apresente ganhos nos resultados de desempenho [Rok05].

Seja P o conjunto formado pelas variáveis atributos de um determinado conjunto de dados. Seja M um método de decisão de escolha entre os elementos de P.

Etapa 1 – Gera Subconjuntos

Nesta primeira etapa (Figura 3.3), seleciona-se um subconjunto de k atributos P_i‘s que pertencem ao conjunto P, pelo método de escolha M. Para cada P_i selecionado por M, o conjunto de dados é dividido em w segmentos disjuntos segundo os valores ou intervalos de

valores que P_i possa assumir. Esta divisão é arbitrária, ou seja, o valor de *w* fica a critério do especialista. Usualmente, se divide em duas partes, para que a estrutura de segmentação fique representada por uma árvore binária.

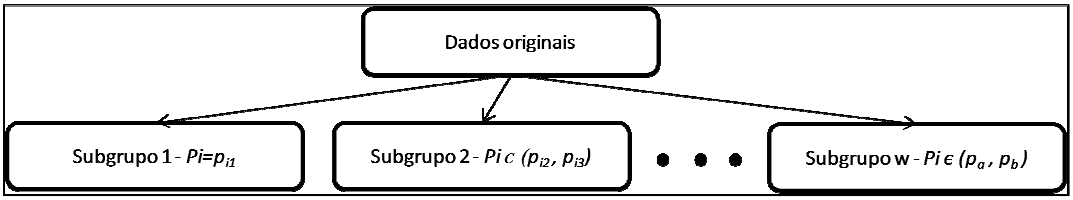


Figura 3.3 – Ilustração da divisão dos dados originais em *w* subgrupos disjuntos.

Etapa 2 – Gera Classificadores

A etapa 2 é realizada para cada iteração da etapa 1. Nesta segunda fase, sistemas especialistas são construídos para cada segmento encontrado na etapa 1. Os resultados das classificações obtidas por estes classificadores são utilizados como meta-atributos para um meta-indutor ou função de combinação *f*. Assim, um novo classificador *C* é obtido e seu desempenho é medido através de uma métrica *D*. O processo descrito é ilustrado com a Figura 3.4.

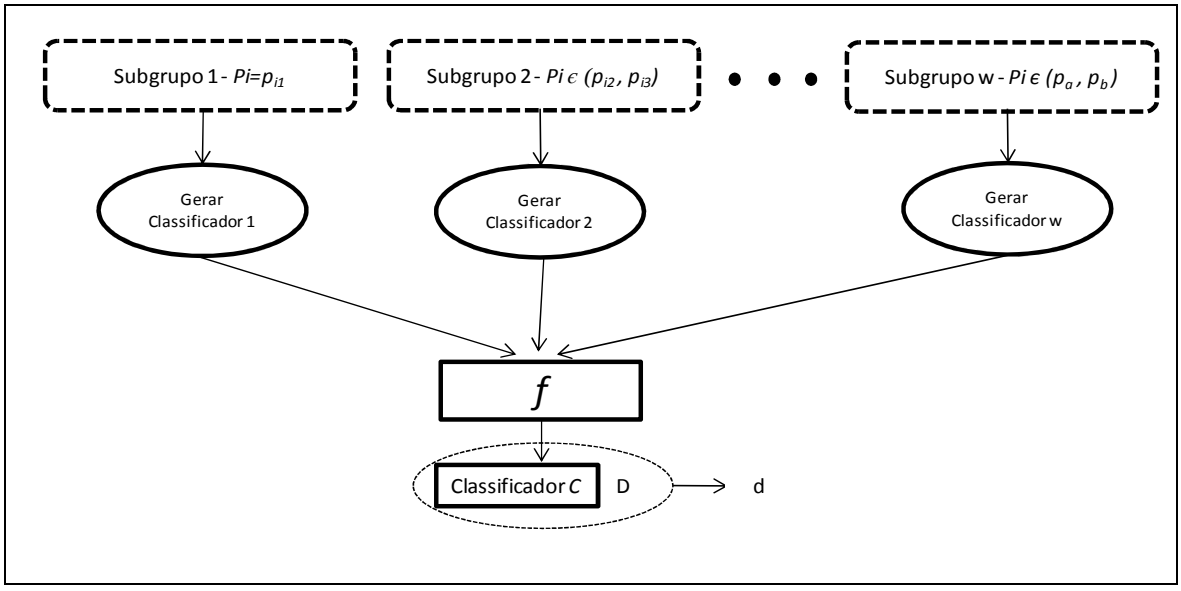


Figura 3.4 – Geração de classificadores por segmento e posterior junção.

Etapa 3 – Seleciona Melhor Segmentação

Após o término da etapa 2, tem-se *k* métricas d_j calculadas, uma para cada variável candidata à segmentação. A melhor medida encontrada mostra qual a melhor variável para ser utilizada na segmentação do conjunto original analisado. Porém, a métrica d_j nem sempre é maior que a

métrica *d* original, ou seja, nem sempre a segmentação da base original pode gerar um multi-classificador que apresente melhor desempenho que o classificador simples original [BWH05]. Neste caso, precisa-se decidir se haverá ou não a segmentação. Como a proposta deste *framework* é realizar uma busca sequencial, pode-se realizar a segmentação mesmo que esta piore o desempenho, pois seguem as buscas nos nós terminais do primeiro nível da árvore. Continuar a segmentar mesmo não obtendo melhoras é uma possível maneira de se evitar máximos locais [MaR05].

Etapa 4 – Busca Sequencial

Conforme sugere a Figura 3.5, ao final da etapa 3, tem-se uma árvore n-ária. A busca sequencial pela melhor estrutura de segmentação consiste em continuar a busca submetendo cada nó terminal da árvore a uma nova segmentação, ou seja, realizar as etapas 1, 2 e 3 para cada segmento encontrado, desde que cada um destes não atinjam o critério de parada *B*. O critério trivial é quando não mais restam atributos a serem escolhidos dentro de um segmento, ou seja, o espaço de variação das variáveis preditoras não mais existe. Outro critério é o número de indivíduos presentes em cada segmento. Se determinado nó terminal fica com número reduzido de registros, pode-se optar por parar a segmentação daquele nó. A segmentação do nó terminal é vista como uma segmentação iniciada do zero. O desempenho dos subsegmentos encontrados é comparado com o desempenho do nó terminal. Essa busca vai maximizando o desempenho localmente, por isso a importância de continuar segmentando mesmo se houver uma pequena perda local aceitável, pois no todo pode haver grande melhora.

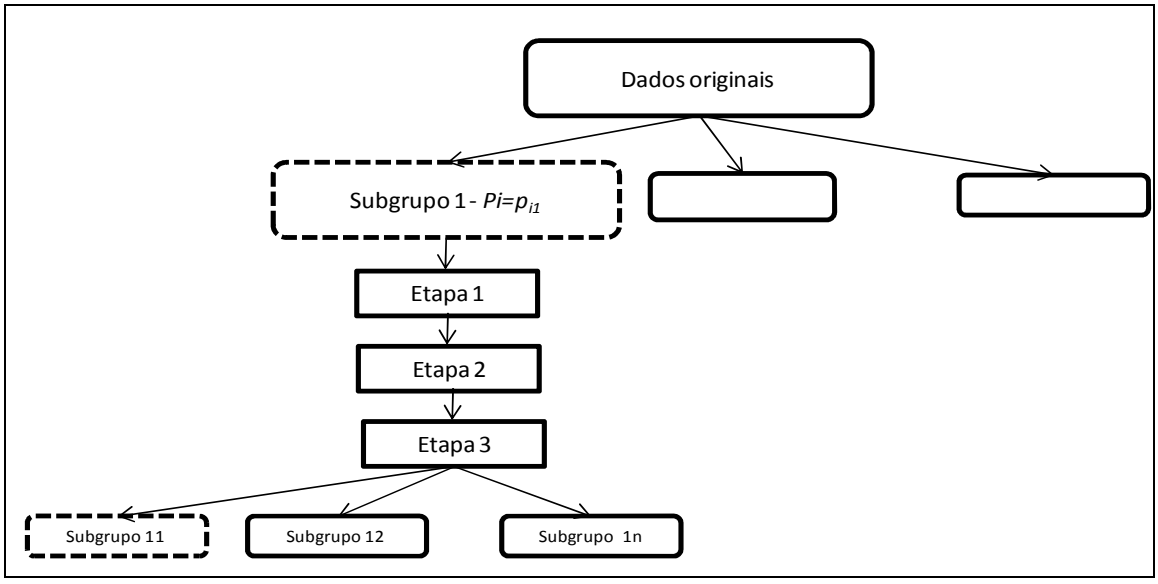


Figura 3.5 – Aplicação das etapas do método em um nó final, gerando novas folhas.

Quando a busca é encerrada o resultado obtido é uma árvore que representa a segmentação. Os nós terminais representam os conjuntos disjuntos finais aos quais se deve

aplicar a geração dos classificadores para então criar o classificador múltiplo final. Este processo pode ser observado na Figura 3.6.

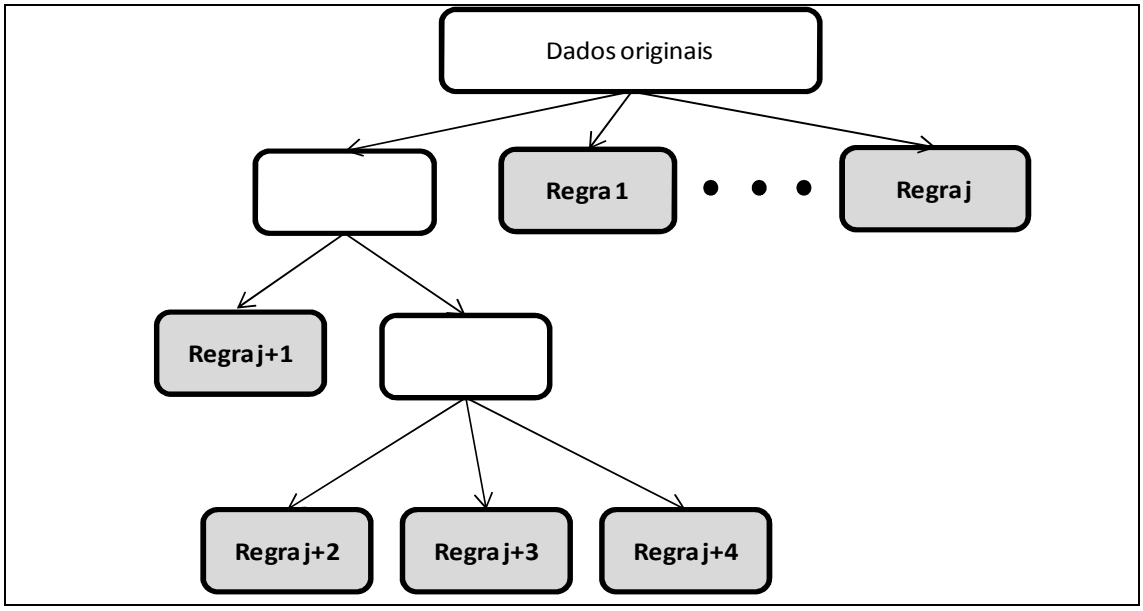


Figura 3.6 – Representação das regras de obtenção dos segmentos em uma árvore *n*-ária.

Definição dos parâmetros

Por razões didáticas, apesar da definição dos parâmetros ser o primeiro passo para a utilização do *framework*, preferiu-se formalizar e detalhar estes parâmetros *após* a descrição algorítmica do referido *framework*. A seguir estes parâmetros são definidos e contextualizados no processo.

P – Conjunto de atributos disponíveis para segmentação: a cada iteração, o nó sujeito à segmentação precisa contar com um conjunto de atributos disponíveis para serem usados como regra de quebra em segmentos distintos. Os membros de *P* geralmente são todas as variáveis preditoras contínuas e categóricas, desde que estas últimas possuam mais de uma categoria. *P* também pode ser definido como um subconjunto das variáveis preditoras, ou seja, alguns atributos podem ser excluídos da lista de candidatos a serem utilizados como regra de quebra, caso o especialista determine que estas não devam ou não possam participar do processo de segmentação.

M – Método de escolha: na primeira etapa, seleciona-se *k* membros de *P* para serem testados na segmentação do nó terminal corrente. Se a busca para achar a melhor segmentação daquele nó for exaustiva, então todos os membros de *P* são testados. Já em buscas não exaustivas, alguma regra de seleção deve ser aplicada para evitar o esforço de realizar testes de todos os membros de *P*.

N – Número de ramos: define o número de segmentos a serem produzidos pela regra de quebra. No caso de árvores binárias, tem-se que o número de ramos é sempre igual a 2.

Indutores: os classificadores gerados em cada segmento no passo 2 são construídos a partir de indutores como Regressão Logística, Redes Neurais, Árvore de Decisão, entre outros. No início do processo, precisa-se definir qual ou quais indutores serão utilizados.

f – Função de combinação ou *meta-learner*: as classificações ou escores dados pelos diversos classificadores especialistas, criados na etapa 2 de cada iteração, são utilizados como meta atributos de um *meta-learner*, que precisa ser definido para se utilizar o *framework*.

D – Métrica de desempenho: medida a ser calculada e avaliada para a determinação da melhor segmentação. Algumas métricas como acurácia, erro de classificação, erro médio quadrático, *KS2* [Con99], *ROC* [Faw06], entre outras, são as mais utilizadas na literatura [ThC02] [HaB03] [Dem06].

B – Critério de parada: a busca sequencial continua a fazer segmentações até atingir o critério de parada. Este critério pode ser formado por diversas regras, por exemplo, uma profundidade máxima para a árvore de segmentação, um número mínimo de registros por segmento, um ganho mínimo na métrica de desempenho, entre outras.

3.2.3.3 – *NNTREE* - MODELO DE APRENDIZADO HÍBRIDO COM REDES NEURAIS

Uma *NNTree* é uma árvore de decisão na qual cada nó que não é não terminal possui uma Rede Neural. Nos capítulos de comparação de métodos de combinação deste trabalho, é utilizada uma proposta relativamente recente de segmentação da *NNTree* [Pra08], que utiliza um *framework* de Redes Neurais *multilayer perceptron* para projetar uma árvore de classificação. Este método divide o conjunto de treinamento recursivamente, de maneira a arranjar em cada nó terminal um subconjunto de elementos de uma única classe e cada nó intermediário cobre um subconjunto de elementos pertencentes a mais de uma classe e possui uma Rede Neural para tentar separá-las. O processo envolve três etapas principais: divisão do nó em novos nós, verificação se os novos nós devem ser novamente divididos e atribuição dos rótulos de classe para os nós que são considerados terminais. Resumidamente, *NNTree*s são classificadores recursivos que particionam o conjunto de treinamento para tentar obter os nós pertencentes a uma única classe.

A Figura 3.7 apresenta um exemplo de uma árvore *NNTree* treinada, em que existem diversas Redes Neurais fazendo as segmentações de maneira a gerar nós terminais ou nós que conterão outras Redes Neurais. Por exemplo, MLP_1 gerou um nó terminal A_{12} que contém

somente instâncias de classe II. Os nós A_{11} e A_{13}, deram origem a duas novas Redes Neurais MLP_2 e MLP_3, que por sua vez geraram novos nós terminais (A_{21}, A_{2m} ...) e outros nós contendo novas Redes Neurais (MLP_4 e MLP_5). Para um maior aprofundamento sobre o funcionamento deste método, recomenda-se a leitura do artigo original [Pra08].

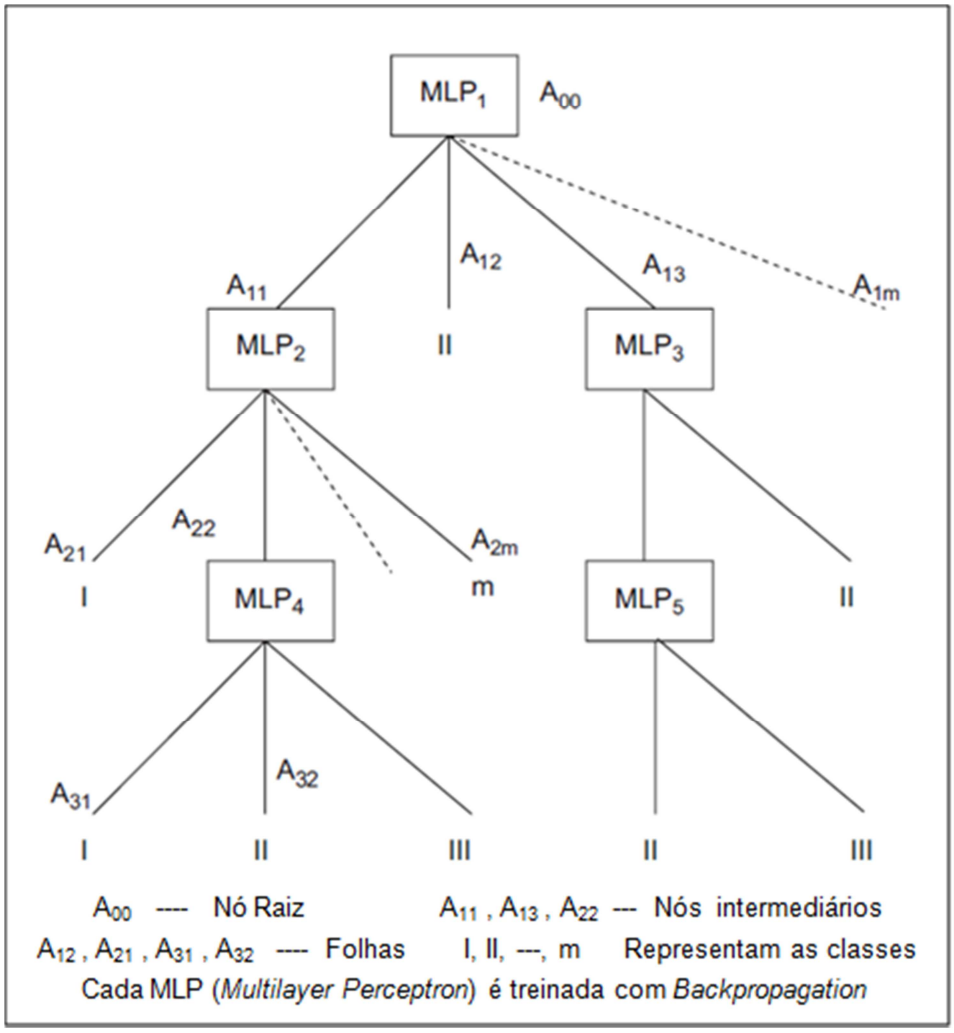


Figura 3.7 – Exemplo de estrutura de classificação utilizando *NNTree* [Pra08].

3.3 – STACKING

Idealizada inicialmente por David Wolpert [Wol92], *Stacked Generalization* ou simplesmente *Stacking* é mais uma forma de combinar múltiplas técnicas. Diferentemente de *Boosting e Bagging*, *Stacking* não é normalmente utilizado para combinar várias instâncias de uma mesma técnica e sim para fazer a integração de técnicas de naturezas diferentes, por exemplo, combinar Regressão Logística, Árvore de Decisão e Redes Neurais.

A aprendizagem é feita em dois níveis: As técnicas iniciais juntamente com seus dados de treinamento constituem o nível 0 e os resultados de cada uma das técnicas do nível 0 serão as entradas do nível 1. Ou seja, os dados de treinamento para o nível 1 são as predições das

técnicas do nível 0. Para esse modelo, o ideal é que o conjunto de treinamento usado para treinar o nível zero seja diferente do conjunto de treinamento para o treino do nível 1. Caso isso não seja possível, pode-se usar a validação cruzada para gerar diversos conjuntos de treinamento. A Figura 3.8 mostra um possível exemplo de *Stacking*. Para o nível 1, David Wolpert sugere que um simples algoritmo linear é o suficiente [Wol92]. Diversos modelos de classificadores para esse nível foram experimentados, com destaque para Análise Discriminante Linear e Árvores de Decisão [TiW99] [DzZ04].

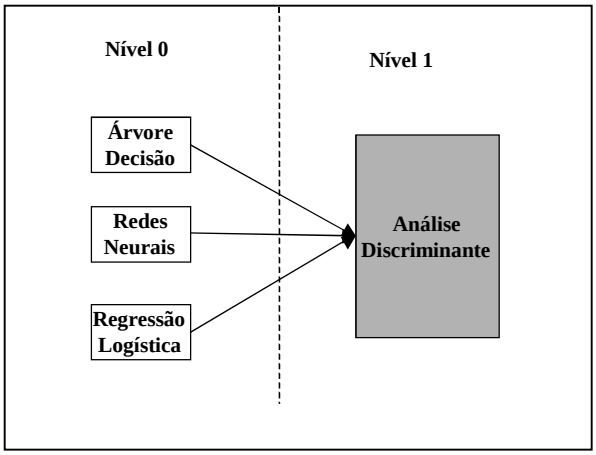


Figura 3.8 – Possível configuração de *Stacking*.

Na aquisição de conhecimento de modelos preditivos com entrada multivariada e saída univariada, há *dois princípios* básicos que são aplicados na aprendizagem supervisionada e que servem para o escalonamento das variáveis por ordem de prioridade e importância a serem utilizadas na solução do problema:

1 - Minimizar a "redundância de informação" entre as variáveis de entrada dos modelos preditivos;

2 - Maximizar a "correlação" entre as variáveis de entrada (isoladamente ou agrupadas) e a de saída.

Esses dois princípios podem ser aplicados nas decisões que serão tomadas pela combinação de sistemas inteligentes baseados nas diversas técnicas de aprendizado de máquina. Nesse sentido, a combinação deve enfatizar a complementaridade destas técnicas (princípio 1), uma vez que todas têm como objetivo o segundo princípio

Apesar de treinados com os mesmos conjuntos de exemplos, os diferentes sistemas inteligentes chegam a soluções diferentes, mesmo que idealmente sejam "aproximadores universais de funções". O fato é que, na prática, não se tem infinitos exemplos, os termos da

função implementada nos dados e nem tempo infinito para treinamento. Por isso, a solução depende tanto do tipo de função do aproximador (polinomial, senoidal, *wavelets*, etc.) quanto do algoritmo para aquisição do conhecimento, entre outros aspectos.

Conjugando os dois princípios da aprendizagem citados com a ideia de combinar técnicas, a expectativa é de que as técnicas com base em tipos de funções mais diferentes entre si produzam combinações com melhores resultados que as técnicas que têm funcionalidade semelhante. Por exemplo: a combinação de técnicas como Regressão Logística e Redes Neurais *MLP* (com função logística) devem produzir uma combinação inferior a Redes Neurais *MLP* e *k-Nearest Neighbors* (*kNN*). Enquanto na primeira combinação ambas as técnicas implementam a função logística univariada e multivariada, respectivamente, na segunda, as técnicas implementam a função logística multivariada e produção de *Voronoi Tessellation* no espaço multidimensional. Nesse segundo caso, o acréscimo de uma terceira técnica como Árvore de Decisão, com os seus hiperplanos de decisão paralelos aos eixos coordenados poderiam melhorar ainda mais o desempenho da combinação. Assim, o *Stacking* é o método de combinação que se propõe a explorar o aspecto da complementaridade funcional matemática de cada técnica de Inteligência Artificial.

3.4 – CONSIDERAÇÕES FINAIS

Neste capítulo foi abordado o estado da arte no processo de criação de classificadores múltiplos e as principais técnicas de combinação de preditores (*Bagging*, *Boosting*, *Stacking* e Segmentação). Todas as implementações dos métodos de combinações, realizadas nos experimentos deste trabalho, são comtempladas neste capítulo. No próximo capítulo o método *RISKSEG* é detalhado.

CAPÍTULO 4

O MÉTODO PROPOSTO – *RISKSEG*

Como discutido no capítulo anterior, o uso de diversos classificadores pode resultar em melhora no índice de acerto da previsão de uma variável alvo. Diversas técnicas para a construção destes múltiplos classificadores foram desenvolvidas e testadas. Cada uma delas com suas características.

No Capítulo 3, foram discutidas as diferentes filosofias e tipos de geração de conjuntos de classificadores individuais, bem como todo o processo para criação dos classificadores múltiplos. A segmentação é uma das técnicas para se criar um conjunto de preditores e é a base para o método proposto. A descrição deste método segue alguns dos padrões do *framework*, apresentado anteriormente.

Uma das principais motivações para a criação deste método é a busca por formas sistemáticas de uma prática já empregada na construção de modelos de risco, em especial, por equipes de instituições financeiras de grande porte e *bureaus* de informações. Porém, após a realização de pesquisas sobre trabalhos acadêmicos focados em segmentação e em outros métodos de combinação de preditores, foi possível também contextualizar esta prática no cenário de Mineração de Dados, mais precisamente no processo de criação de preditores múltiplos.

A criação do método *RISKSEG* é focada em estudos de risco, ou seja, em modelos cuja variável alvo é dicotômica. A resposta deste tipo de abordagem é normalmente conhecida como escore. Foram vários os motivos para a escolha deste tipo particular de problema de classificação, mas pode-se destacar a grande diversidade de possíveis métricas de avaliação de desempenho, a simplicidade na avaliação dos resultados e a sua vasta aplicabilidade. A complexidade de análise de saídas contínuas geradas a partir de uma variável dicotômica é muito menor do que quando o problema envolve mais classes. No caso de uma Regressão Logística, essa saída contínua é a estimativa da probabilidade de ocorrência para cada uma das

respostas, mas existem várias outras técnicas e abordagens que podem ser utilizadas para se chegar a este tipo de resposta, como: Análise Discriminante Linear e Redes Neurais [DCO96] [BaK98] [Rud01] [MaM03].

Para este trabalho, além da análise mais comumente conhecida, baseada em erros de classificação, também foram utilizadas o *KS2* [Con99] e a curva *ROC* [Faw06], que são métodos não-paramétricos para análise de desempenho. Estes dois, em especial, são os mais importantes e conhecidos em modelagens de riscos [ThC02] [HaB03] [Dem06]. Estas duas métricas são utilizadas principalmente para avaliação de desempenho em modelos de riscos baseados em uma resposta contínua, obtida por meio de um alvo dicotômico, permitindo que modelos sejam ajustados de forma a gerar uma saída contínua como resposta (escore ou risco). Isto as torna medidas interessantes e complementares para avaliação do método.

4.1 – FUNDAMENTAÇÃO TEÓRICA COMPLEMENTAR

Quando se analisa os efeitos de variáveis em modelos, é comum uma variável possuir um tipo de efeito isoladamente mas, quando combinada com outra, esse efeito pode ser diferente. Por exemplo, ao analisar a tendência isolada de uma variável no comportamento do risco, pode-se chegar à conclusão de que ela possui efeito negativo, ou seja, quanto maior o seu valor, maior será o risco. Porém, se esta mesma variável for analisada condicionada a outra, o seu comportamento pode ser diferente. Nesta seção serão apresentados alguns conceitos que justificam o uso da segmentação como uma alternativa para combinação de classificadores e consequentemente o aumento do poder preditivo. Para entender melhor o método de busca pelas melhores variáveis para segmentar, é importante entender um pouco melhor alguns dos conceitos necessários que justificam a sua criação.

4.1.1 – A SEGMENTAÇÃO E AS ESTRUTURAS NÃO-LINEARES

Além de possibilitar a geração de múltiplos classificadores, a segmentação pode ser entendida como uma maneira de adequar certas suposições de linearidade de alguns métodos de aprendizado ou análise estatística, como a Regressão Linear. Este método de predição pode utilizar-se de técnicas de segmentação para produzir a chamada regressão estratificada, conforme detalhado a seguir.

Com a técnica de Regressão Linear, buscam-se estimativas para parâmetros de uma equação que supostamente descreve o comportamento de uma variável dependente em relação a outras variáveis dadas. Estas equações consistem em uma combinação linear de variáveis, resultando no valor da variável dependente de interesse. Porém, a linearidade proposta pela

Regressão Linear muitas vezes não condiz com a verdadeira estrutura de dependência entre as variáveis de uma amostra. Sejam x_1, x_2 e x_3 variáveis aleatórias contínuas com densidade $f(x_i|\theta_i)$ e x_4 uma variável aleatória dicotômica que assume valores 0 (zero) ou 1 (um), de acordo com uma probabilidade p_4. Considera-se também uma variável aleatória Y que possui a seguinte relação com as demais variáveis:

$$Y = \rho + (\delta_1 x_1 + \delta_2 x_2 + \delta_3 x_3)x_4 + (\gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 x_3)(1 - x_4) + \varepsilon \qquad (1)$$

em que ε é um fator de erro aleatório com densidade $f(\varepsilon)$ e $\rho, \delta_1, \delta_2, \delta_3, \gamma_1, \gamma_2$ e γ_3 são escalares fixos.

Pode-se notar que a relação entre Y e as variáveis independentes é não-linear, pois Y não é resultado de combinação linear de x_1, x_2, x_3 e x_4. Neste caso, uma Regressão Linear não é adequada para se estimar a relação entre as variáveis "x_i's" e Y.

A relação (1) acima só não é linear devido à variável x_4. Note que esta relação pode ser reescrita condicionada ao valor de x_4:

$$Y|x_4 = \begin{cases} \rho + \delta_1 x_1 + \delta_2 x_2 + \delta_3 x_3 + \varepsilon, & \text{se } x_4 = 1 \\ \rho + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 x_3 + \varepsilon, & \text{se } x_4 = 0 \end{cases} \qquad (2)$$

Logo, Regressões Lineares aplicadas separadamente nos diferentes subgrupos ($x_4 = 1$ e $x_4 = 0$) podem ser mais adequadas quando relações como a descrita em (1) estão presentes na base de dados. Esta técnica é conhecida como regressão estratificada [Nev98] [Nev99] ou segmentação.

Portanto, a segmentação é recomendada para capturar relações entre a variável dependente e as independentes para subpopulações com perfis diferentes. Assim, os modelos especializados naquele segmento têm condições de explorar melhor as suas peculiaridades [ThC02].

Para efeito ilustrativo, as Figuras 4.1 a 4.4 mostram graficamente as diferentes relações entre duas variáveis preditoras e a variável dependente. Neste exemplo, estuda-se o risco de um determinado evento, dadas as variáveis: faixa de idade (em que a categoria 1 representa as menores idades e a categoria 10, as maiores) e sexo (M = Masculino e F = Feminino). Como mostrado na Figura 4.1, existe uma relação linear decrescente entre idade e risco. A Figura 4.2 exibe tal relação separada por sexo. Percebe-se que existe um efeito aditivo da variável sexo na relação entre a idade e o risco. As Figuras 4.3 e 4.4 mostram dois exemplos, onde a relação entre idade e o risco são afetados pelo sexo, inclusive, para a Figura 4.4, existe uma inversão do

sinal aditivo no risco, para as categorias 9 e 10 da idade. Alguns modelos não lineares (como Redes Neurais, por exemplo) ou a inclusão de termos de interação entre as variáveis podem capturar alguns destes comportamentos. Porém, mesmo com estes tipos de métodos, têm-se limitações e dificuldades de aprendizado devido ao aumento da complexidade e quantidade de variáveis envolvidas no modelo.

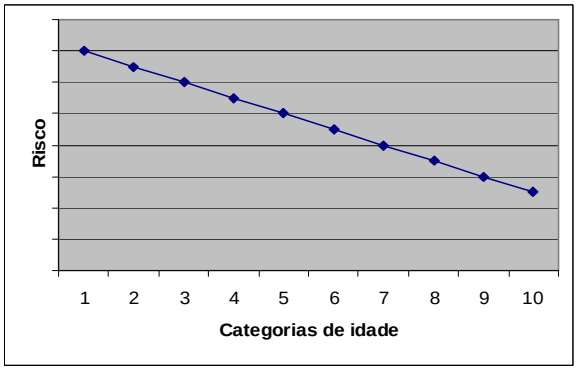


Figura 4.1 – Relação entre a variável risco e idade.

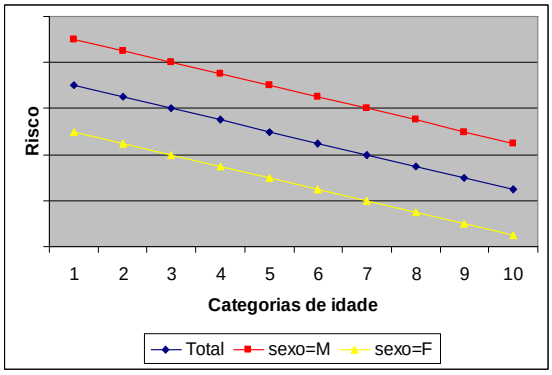


Figura 4.2 – Efeito aditivo da variável sexo na relação risco x idade.

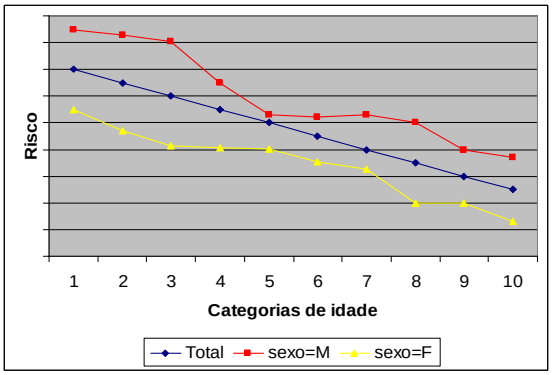


Figura 4.3 – Interação entre sexo e idade na explicação do risco.

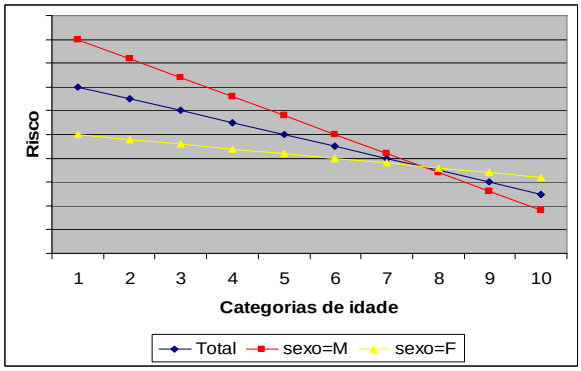


Figura 4.4 – Outra interação entre sexo e idade na explicação do risco.

Uma opção à busca de um bom modelo segmentado pode ser o aproveitamento das interações entre as variáveis através da utilização de um modelo fatorial completo, ou ao menos incluindo as combinações 2 a 2 ou 3 a 3 das variáveis. O problema é que esta abordagem acarreta em um alto custo computacional e um aumento significativo no número de variáveis que participarão da modelagem. Isto porque a análise de todas as combinações varia de acordo com o número de variáveis, que é de ordem 2^n–1, onde *n* é o número de variáveis.

A seguir é demonstrada a equivalência entre o modelo de delineamento fatorial completo e o de Regressão Linear segmentada, além de uma pequena demonstração de como modelos aninhados aumentam a razão de verossimilhança e proporcionam um aumento no desempenho.

Estes tópicos são parte da fundamentação teórica para justificar a criação e o emprego do método proposto.

4.1.1.1 – EQUIVALÊNCIA DO MODELO SEGMENTADO E FATORIAL DE ORDEM DOIS

Seja um parâmetro θ desconhecido e uma função $\tau(\theta)$ para a qual se deseja ajustar um modelo de Regressão, utilizando como variáveis regressoras X_1, X_2 e X_3 dicotômicas que assumem valores 0 (zero) ou 1 (um). Por questão de notação estatística denota-se X_i como variáveis aleatórias e x_i a realização de uma variável aleatória. Suponhamos que ao invés de um modelo único contendo as 3 (três) variáveis, deseja-se ajustar 2 (dois) modelos, um para os indivíduos em que $X_3 = 1$ e outro para $X_3 = 0$. Em outras palavras, ajustam-se modelos para dois segmentos distintos. Neste caso, têm-se dois modelos de regressão condicionados aos valores de X_3 da seguinte forma:

$$\tau(\theta/ X_3 = 1) = \mu' + \beta'_1 X_1 + \beta'_2 X_2 + e' \quad (1)$$

$$\tau(\theta/ X_3 = 0) = \mu'' + \beta''_1 X_1 + \beta''_2 X_2 + e'' \quad (2)$$

onde e' e e'' são erros aleatórios.

Pode-se escrever as equações (1) e (2) em forma de uma única equação, utilizando X_3:

$$\tau(\theta/X_3 = x_3) = x_3 (\mu' + \beta'_1 X_1 + \beta'_2 X_2 + e') + (1- x_3)(\mu'' + \beta''_1 X_1 + \beta''_2 X_2 + e'') \quad (3)$$

Nota-se (3) é equivalente a (1) caso $x_3 = 1$, pois todo o segundo termo da soma seria zerado. De mesma maneira, (3) é equivalente a (2) quando $x_3 = 0$.

$$\tau(\theta/X_3 = x_3) = x_3\mu' + x_3\beta'_1 X_1 + x_3\beta'_2 X_2 + x_3e' - x_3\mu'' - x_3\beta''_1 X_1 - x_3\beta''_2 X_2 - x_3e'' + \mu'' + \beta''_1 X_1 + \beta''_2 X_2 + e''$$

$$= \mu'' + \beta''_1 X_1 + \beta''_2 X_2 + (\mu' + e' - \mu'' - e'')x_3 + (\beta'_1 - \beta''_1) X_1 x_3 + (\beta'_2 - \beta''_2) X_2 x_3 + e'' \quad (4)$$

Nota-se que (4) tem a forma de um modelo de regressão que, além de considerar os efeitos aditivos das variáveis X_1, X_2 e X_3, possui elementos de interação de X_3 com as outras duas variáveis X_1 e X_2.

Logo, quando se segmenta a amostra em duas partes, segundo os valores de uma variável dicotômica qualquer, e ajustam-se os modelos de regressão para estes segmentos, implicitamente estamos ajustando um modelo de regressão que considera as interações das variáveis preditivas com a variável utilizada na segmentação. Pode-se também utilizar variáveis

categóricas, que não sejam dicotômicas, para isto é necessário apenas criar variáveis auxiliares dicotômicas que as representem [LoF01].

4.1.1.2 – MODELOS ANINHADOS E A RAZÃO DE VEROSSIMILHANÇA

A Regressão Linear Logística baseia-se na maximização da função de verossimilhança $L(\beta|y)$, em que y é o vetor de valores observados da variável dependente Y e β é o vetor de parâmetros da função linear que descreve a dependência entre Y e as variáveis preditoras $x_1, ..., x_n$.

Em outras palavras, procura-se encontrar valores para o vetor de parâmetros β, tais que estes sejam os mais plausíveis possíveis, dados $y, x_1, ..., x_n$, valores observados da variável dependente e das variáveis preditoras.

Após a maximização da verossimilhança em função de β, tem-se $L^*(\hat{\beta}|y)$ a máxima verossimilhança de um dado modelo $\hat{\tau}_1$.

Caso seja considerado um novo ajuste, em que uma ou mais variáveis preditoras são retiradas do modelo original, tem-se um novo modelo $\hat{\tau}_2$ dito aninhado (ou restrito) ao original $\hat{\tau}_1$ [All97]. Seja $L^*(\hat{\varphi}|y)$ a máxima verossimilhança de $\hat{\tau}_2$. Como $\hat{\tau}_1$ e $\hat{\tau}_2$ são aninhados, tem-se que a razão $\frac{L^*(\hat{\varphi}|y)}{L^*(\hat{\beta}|y)}$ é menor ou igual a um, tornando possível realizar testes de hipóteses com esta relação [CaB02].

Este teste mede o quão longe de 1 (um) e mais próxima de 0 (zero) esta razão está. Quanto mais próxima de zero a relação, maior evidência que o modelo $\hat{\tau}_1$(completo) tenha verossimilhança maior que $\hat{\tau}_2$, em outras palavras, maior a evidência de que as variáveis extras do modelo completo sejam estatisticamente significativas. Caso exista diferença significativa, o modelo completo explica melhor os dados que o mais simples, resultando em melhoria nas métricas de avaliação de desempenho.

O método de segmentação aqui abordado é baseado no delineamento fatorial completo [BoD86] e tem a vantagem de buscar uma forma otimizada de aproveitar a informação contida nas variáveis derivadas dos efeitos das principais das interações. Assim, o desempenho final do método *RISKSEG* busca aproximar-se ao máximo do modelo fatorial completo, utilizando uma quantidade reduzida de segmentações para reproduzir o efeito das principais interações. Caso estas interações realmente estejam presentes na relação da variável alvo com as preditoras, o aumento do desempenho do modelo ficará nítido nas métricas de avaliação.

4.2 – DESCRIÇÃO GERAL DO MÉTODO PROPOSTO

O método *RISKSEG* busca gerar um conjunto de modelos que, combinados, tenham melhor desempenho que os desenvolvidos a partir de apenas um preditor, utilizando toda a amostra de dados no treinamento. A busca pela melhor estrutura de segmentação normalmente é feita de forma sequencial, semi-exaustiva e recursiva, procurando a melhor segmentação para os nós terminais atuais da árvore, testando-se todas as segmentações possíveis.

Ao invés disso, no *RISKSEG*, são estudadas as possíveis interações entre as variáveis do modelo, para cada nó terminal que será segmentado. Como visto anteriormente, a segmentação dos dados em subgrupos de diferentes valores de certa variável é equivalente a adicionar fatores de interação entre esta variável e todas as demais presentes no modelo, do ponto de vista de ajuste de Regressões Estatísticas. Seguindo este raciocínio, a melhor segmentação possível seria aquela que envolve uma variável que possua tal interação com as demais.

O algoritmo que implementa a segmentação automatizada do *RISKSEG* baseia-se na construção de uma árvore de modelos feitos a partir do teste das melhores variáveis candidatas, ou seja, testam-se apenas as *n* variáveis que possuam as melhores interações com as demais, a partir de uma métrica escolhida. Desta forma, evita-se a busca exaustiva pelas melhores segmentações e pode-se aperfeiçoar o resultado final baseado em qualquer métrica desejada. A seguir, é feita uma análise do método utilizando uma Regressão Linear estatística.

Sejam *x*1, *x*2, *x*3 e *x*4 variáveis aleatórias, Y uma variável dependente (alvo), ε_i o fator de erro e $\mu_i, \beta_i, \lambda_i, \alpha_i, \omega_i, \theta_i$ são os pesos constantes das equações. Definiu-se um modelo de Regressão Linear simples:

$$Y = \mu + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \varepsilon \qquad (1)$$

Este modelo considera apenas os efeitos principais das variáveis, ou seja, nenhum termo de interação foi incluído na equação para estimação de seus parâmetros. Conforme visto anteriormente, a inclusão de termos de interação de $x_i x_j$, com Xi fixo e j variando, é equivalente a segmentar os dados por *xi* e ajustar regressões separadas para cada subgrupo. Por exemplo:

$$Y = \mu_1 + \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 + \lambda_4 x_4 + \lambda_{12} x_1 x_2 + \lambda_{13} x_1 x_3 + \lambda_{14} x_1 x_4 + \varepsilon_1 \qquad (2)$$

Neste caso, tem-se os efeitos das interações entre x_1 e as demais variáveis do modelo. Este ajuste de regressão é similar a segmentar a base entre os valores possíveis de x_1 e ajustar

diferentes regressões para os subgrupos gerados. Analogamente, pode-se definir outros três modelos:

$$Y = \mu_2 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_4 + \alpha_{12} x_1 x_2 + \alpha_{23} x_2 x_3 + \alpha_{24} x_2 x_4 + \varepsilon_2$$

$$Y = \mu_3 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + \omega_4 x_4 + \omega_{13} x_1 x_3 + \omega_{23} x_2 x_3 + \omega_{34} x_3 x_4 + \varepsilon_3 \qquad (3)$$

$$Y = \mu_4 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_{14} x_1 x_4 + \theta_{24} x_2 x_4 + \theta_{34} x_3 x_4 + \varepsilon_4$$

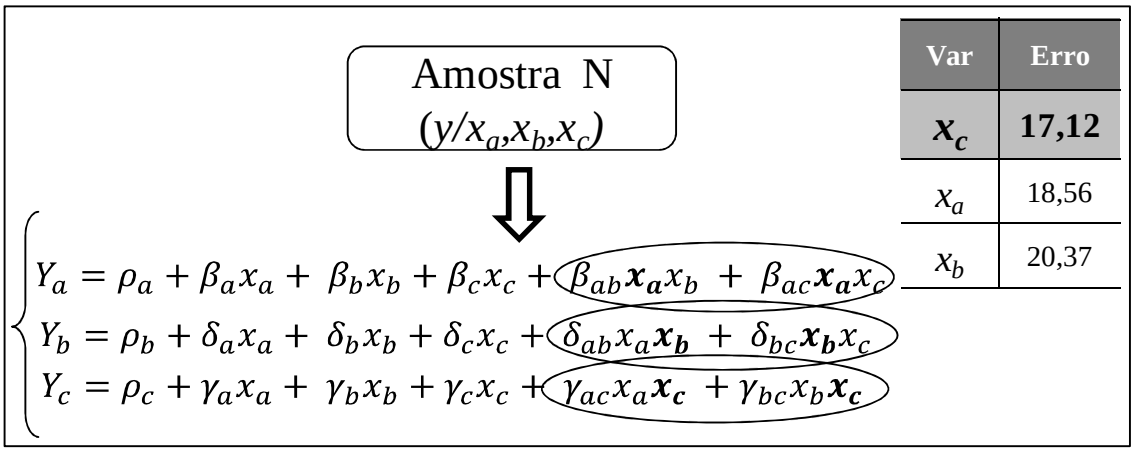
Cada um destes modelos seria, respectivamente, equivalente a segmentações por x_2, x_3 e x_4. Logo, ajustando-se as quatro regressões definidas em (2) e (3), pode-se escolher a melhor delas em termos de alguma medida de desempenho *d*. Com esta escolha, tem-se então a melhor variável para ser usada na segmentação, segundo a medida de interesse. Após a escolha da variável, é necessária a escolha da regra de quebra a ser utilizada, como por exemplo, escolher uma categoria ou um intervalo, no caso de variáveis numéricas.

Apesar do método proposto suportar a criação de árvores *n*-árias, sugere-se a utilização de árvores binárias, pois entende-se que a partir da divisão binária é possível ter resultados equivalentes às árvores *n*-árias. Desta forma, daqui por diante será utilizado este tipo de configuração nas descrições e na implementação do método para os estudos de caso.

A busca proposta no *RISKSEG* diferencia-se da busca semi-exaustiva, pois não testa todas as divisões possíveis, ou seja, todas as regras de quebra de todas as variáveis. Apesar de envolver o custo ajustando diversas regressões, não há o custo de se realizar inúmeras quebras para avaliar desempenho. O número de quebras limita-se apenas ao número de categorias das variáveis candidatas. Outro fato que deixa o método *RISKSEG* mais rápido é a possibilidade de ajuste das regressões (3) com uma amostra do conjunto de treinamento. Assim, para casos em que a amostra para treinamento é muito grande, o método gasta menos recursos de processamento computacional. No Capítulo 6 é apresentada uma comparação entre os tempos das duas abordagens.

A Figura 4.5 mostra um exemplo da otimização proposta pelo método *RISKSEG*, considerando-se 3 (três) variáveis preditivas. Como métrica de avaliação de desempenho utilizou-se a minimização de erro da Regressão Linear simples. Neste caso, considera-se uma amostra aleatória de tamanho N, 3 (três) variáveis preditivas categóricas definidas como $x_{\mathrm{a}}, x_{\mathrm{b}}$ e x_{c} , a variável alvo dicotômica y, os diferentes pesos ajustados $\beta_i, \gamma_i, \delta_{\mathrm{i}}$ do modelo, e a constante do intercepto ρ_{i}. As equações refletem apenas uma representação simplificada dos modelos,

pois, como as variáveis são categóricas, elas teriam que ser transformadas em numéricas, por exemplo, gerando uma variável dicotômica para cada categoria das variáveis.



Var	Erro
x_c	17,12
x_a	18,56
x_b	20,37

Figura 4.5 – Método de divisão RISKSEG – Escolha da variável.

Assim, para cada uma das variáveis principais, x_a, x_b e x_c, são gerados modelos de Regressão Linear simples com a interação desta variável (principal) com as demais. As partes das fórmulas envolvidas nas elipses representam as interações entre as variáveis. Como neste exemplo tem-se três variáveis principais de entrada, logo serão gerados três modelos: o primeiro para as interações de x_a (Y_a), o segundo para x_b (Y_b) e o terceiro para x_c (Y_c). À direita da Figura 4.5 tem-se uma tabela (hipotética) com os erros calculados a partir do desempenho de cada um dos modelos e ordenadas do menor para o maior. Seleciona-se então, a variável relacionada ao modelo que apresentou o menor erro. No exemplo, a variável selecionada foi x_c.

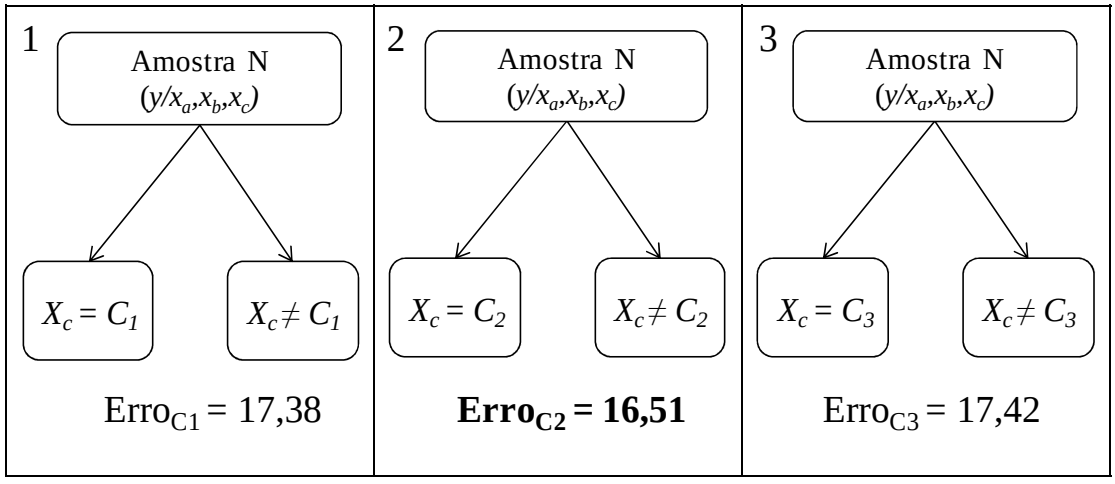


Figura 4.6 – Método de divisão *RISKSEG* – Escolha da categoria.

Na Figura 4.6 observa-se as três categorias da variável x_C representadas por C_1, C_2 e C_3. Para cada uma destas categorias é feita uma divisão da amostra N, treina-se dois modelos, um

para cada segmento, e associa-se o erro a cada uma das 3 (três) segmentações experimentada. Aquela divisão que apresentar o menor erro é selecionada (com C_2). Para este exemplo, o modelo segmentado final é obtido como resultado de:

$$Y|x_c = \begin{cases} \beta_0 + \beta_1 x_{\mathrm{a}} + \beta_2 x_{\mathrm{b}}, & \text{se } x_{\mathrm{C}} = C_2 \\ \gamma_0 + \gamma_1 x_{\mathrm{a}} + \gamma_2 x_{\mathrm{b}} + \gamma_3 x_{\mathrm{c}}, & \text{se } x_{\mathrm{C}} \neq C_2 \end{cases}$$

em que $x_{\mathrm{a}}, x_{\mathrm{b}}$ e x_{c} são as variáveis preditivas e β_n, γ_n são constantes representando os diferentes pesos ajustados por uma regressão.

O Algoritmo 4.1 ilustra o método de busca da segmentação por um nó terminal. O primeiro laço serve para escolher a melhor variável para segmentar e o segundo para selecionar a categoria desta variável que gerará a divisão dos subconjuntos. Ou seja, após selecionar a variável que apresentou o melhor resultado a partir da regressão com a interação com as demais (fatorial), testam-se todas as combinações possíveis das categorias da variável de maneira binária e, no final do processo, tem-se os dois melhores segmentos um para "*Var = Cat*" e outro para "*Var ≠ Cat*", onde *Var* e *Cat* são a variável e a categoria selecionadas, respectivamente. Após a geração desta segmentação é feito um teste para determinar se o desempenho dos dois segmentos sugeridos são mais preditivos do que o modelo único gerado no nó superior. Caso os segmentos obtenham um resultado superior ao do nó superior, a segmentação é confirmada e a divisão dos novos nós da árvore continua até que nenhuma nova divisão seja gerada, porque os modelos gerados não obtiveram um resultado melhor do que o seu nó superior ou porque o método atingiu outros critérios de parada (máxima profundidade, por exemplo), analogamente ao processo de criação de Árvores de Decisão. Os resultados finais são diversos modelos gerados a partir de subconjuntos excludentes da base disponível para treinamento. Caso a resposta final desejada seja um escore, podem ser necessários ajustes pós-treinamento para compatibilizar os resultados dos diversos segmentos. Esta junção é abordada no *framework* genérico descrito no Capítulo 3. Os possíveis métodos para esta tarefa no *RISKSEG* são mostrados na Seção 4.2.1.3.

N	– Conjunto de treinamento de um dado nó terminal
S_1, S_2	– Subconjuntos do conjunto de N gerados a partir de uma segmentação
p	– Número de variáveis de entrada do conjunto N
$X_1,X_2,...,X_p$	– Variáveis discretas de entrada do conjunto N
w	– Inteiro que recebe quantidade de categorias de uma variável discreta
C[w]	– Vetor contendo os valores das ocorrências de uma variável discreta
y	– *Variável alvo (dependente)*
fMet( )	– *Função para cálculo de desempenho (quanto maior melhor para este exemplo)*
fMerge()	– *Função para juntar as respostas dos modelos*
Var	– *Variável escolhida para teste de suas categorias*
Cat	– *Categoria de* Var escolhida
rMod,	– Respostas (alvo real + alvo predito) dos modelos fatoriais
$rMod_0$	– Melhores respostas (alvo real + alvo predito) dos modelos
$rMod_1$, $rMod_2$	– Respostas (alvo real + alvo predito) dos modelos dos segmentos candidatos
$rMod_{12}$	– Junção das respostas (alvo real + alvo predito) dos modelos dos segmentos candidatos

Entrada: *N*;

$rMod_0 \leftarrow$ ø; (Inicia melhores respostas)
Para i = 1 até p faça { (Laço para todas as variáveis de entrada)
  rMod ← Respostas da regressão y ~ $\sum_{j=1}^{P} X_j + \sum_{j=1, j\neq i}^{P} X_j X_i$, com dados de *N*; (Treina modelo com interações)
  Se *fMet*(*rMod*) > *fMet*($rMod_0$) então { (Verifica qual o melhor modelo)
    $rMod_0 \leftarrow rMod$; (Salva melhor respostas dos modelos)
    $Var \leftarrow X_i$; (Salva variável escolhida para segmentar)
  };
};

$rMod_0 \leftarrow$ ø; (Reinicia melhores respostas)
$rMod_{12} \leftarrow$ ø; (Inicia conjunto de respostas de Mod_{12})
C[w] ← Valores das categorias de *Var*; (Carrega as ocorrências de *Var* em C[w])
w ← Quantidade de categorias de *Var*; (Conta a quantidade de categorias de *Var* e carrega em w)

Para i = 1 até w faça { (Testa todas as categorias de *Var*)
  S_1 ← Subconjunto de *N*, onde *Var* = C[w]; (Gera o primeiro subconjunto para treinamento)
  S_2 ← Subconjunto de *N*, onde *Var* < > C[w]; (Gera o segundo subconjunto para treinamento)
  $rMod_1$ ← Respostas da regressão y ~ $\sum_{j=1}^{P} X_j$, com dados de S_1; (Treina modelo e carrega resultados em rMod1)
  $rMod_2$ ← Respostas da regressão y ~ $\sum_{j=1}^{P} X_j$, com dados de S_2; (Treina modelo e carrega resultados em rMod2)
  $rMod_{12} \leftarrow rMod_1 \cup rMod_2$; (Junta as respostas dos modelos segmentados)
  Se *fMet*($rMod_{12}$) > *fMet*($rMod_0$) então { (Verifica qual o melhor modelo)
    $rMod_0 \leftarrow rMod_{12}$; (Salva melhor respostas dos modelos)
    $Cat \leftarrow$ C[w]; (Salva categoria escolhida para segmentar)
  };
};

Saída: *Var e Cat*; (Melhor segmentação para o nó -- variável e categoria)

Algoritmo 4.1 – Algoritmo para escolha de uma melhor segmentação do *RISKSEG*.

4.2.1 – ANALOGIA COM *FRAMEWORK* GENÉRICO DE SEGMENTAÇÃO

O *framework* apresentado no Capítulo 3 é uma proposta genérica de apresentação do processo de segmentação de modelos em forma de árvore, que utiliza busca recursiva sequencial para escolha das divisões. O *RISKSEG* pode ser entendido como um caso particular de extensão deste *framework*, com parâmetros e características específicas.

Assim como no *framework*, o *RISKSEG* também parte do princípio da existência de um conjunto *P* de variáveis atributos disponíveis para a segmentação e uma função de escolha *M*, que decide quais elementos de *P* serão utilizados e verificados em cada iteração do método. A seguir, as etapas e definição dos parâmetros do *framework* genérico são comparadas com o *RISKSEG*. Também são mostradas melhorias e extensões.

4.2.1.1 – *RISKSEG* - ETAPA1 (GERA SUBCONJUNTOS)

Igualmente ao *framework* genérico, no primeiro passo, seleciona-se um subconjunto de *k* atributos que pertençam ao conjunto *P*, através do método de escolha *M*. No caso do *RISKSEG*, o método *M* é o processo descrito na Seção 4.2, que ajusta diversas regressões e faz uma ordenação das melhores, segundo uma métrica *D* de avaliação de modelos. A regressão de melhor desempenho indica qual ou quais variáveis devem ser escolhidas para continuar na próxima fase. Teoricamente, a regressão que apresenta melhor desempenho indica qual a melhor variável a ser utilizada na segmentação. Porém, a métrica *D* pode apresentar variâncias relevantes e os resultados obtidos para algumas regressões podem, na verdade, ser estatisticamente iguais aos resultados de outras regressões. Desta forma, não é recomendável utilizar apenas a melhor, mas sim um conjunto das *k* melhores regressões para tomada de decisão das variáveis a serem escolhidas.

Após a escolha das *k* variáveis, selecionadas por *M*, um conjunto $E=\{e_1,e_2,...e_k\}$ é construído, indicando as variáveis candidatas à segmentação. Para cada variável candidata e_i, o processo de divisão é feito. Para isto, listam-se as categorias de e_i, representadas pelo conjunto $A_i=\{a_1,a_2,...,a_c\}$ e divisões binárias são feitas nos registros do nó em questão. Cada divisão gera dois segmentos, um com a regra $e_i = a_j$ e outro com a regra $e_i \neq a_j$. Isso é feito para todo a_j, desde que cada um dos dois segmentos gerados possua uma quantidade mínima de registros especificada previamente através de um parâmetro. Este parâmetro será detalhado na Seção 4.2.1.6. O método lida também com variáveis numéricas, porém, estas precisam ser categorizadas previamente. Mais detalhes sobre a realização da categorização são abordados na Seção 4.2.1.5.

4.2.1.2 – *RISKSEG* - ETAPA 2 (GERA CLASSIFICADORES)

Nesta segunda etapa, sistemas especialistas são construídos para cada um dos segmentos gerados na etapa anterior. Apesar da formulação teórica se basear nas interações entre variáveis do ponto de vista de Regressões Estatísticas, os classificadores especialistas de cada segmento podem ser construídos utilizando-se também outras técnicas de regressão, como Redes Neurais. Isto porque a divisão da base em segmentos com mais linearidade (menos complexidade) pode facilitar o seu treinamento e melhorar seu resultado final [VAE03].

Uma importante característica do *RISKSEG* é a possibilidade do uso de diferentes técnicas de aprendizado de máquina em uma mesma árvore de modelos. As abordagens de combinações de classificadores normalmente não exploram o uso de diferentes métodos preditores para uma mesma decisão.

4.2.1.3 – *RISKSEG* - ETAPA 3 (SELECIONA MELHOR SEGMENTAÇÃO)

Esta etapa parte de uma lista de valores da métrica selecionada para verificar qual dupla de segmentação foi a mais bem sucedida. A melhor medida encontrada mostra qual a variável e qual a categoria que deve ser utilizada na segmentação do conjunto original analisado. Porém, a métrica d_j encontrada pode não ser melhor que a do conjunto original, ou seja, nem sempre a segmentação da base original pode gerar um multi-classificador que apresente melhor desempenho que o classificador simples original do nó superior [BWH05]. Neste ponto, o método proposto permite a definição de um parâmetro para o valor máximo aceitável de ganho/perda de desempenho que permita a segmentação.

Como o método *RISKSEG* também suporta a combinação de técnicas de regressão, em especial com alvos dicotômicos, existe um componente a mais, que seria o método utilizado para alinhamento e junção dos preditores. Este cuidado normalmente não é necessário quando se utiliza classificação. Mas quando se trabalha com risco de ocorrência de alguma classe, existe a preocupação, como já abordada anteriormente, de que os escores sejam compatíveis no momento da junção dos preditores, principalmente quando se utilizam técnicas diferentes. Durante o desenvolvimento do trabalho, principalmente nos estudos de caso, foram testadas algumas maneiras para esta tarefa e selecionadas duas delas.

Na junção utilizando o método *Stacking* [Wol92], cada registro tem duas colunas: uma com o escore do segmento ao qual ele pertence e outra com o escore do segmento ao qual ele não pertence. Este último, recebendo valor 0 (zero). Esses dois escores são então submetidos a

uma regressão, juntamente com a variável alvo. Assim, um novo escore é gerado para avaliação de desempenho.

A outra opção para junção dos segmentos é recalibrar os escores utilizando uma técnica similar ao *Stacking*, conhecida como *Marginal Odds* [ThC02]. Para cada um dos segmentos é gerado um novo escore, regressando o escore antigo em função da variável alvo. Depois, toma-se um desses dois segmentos como referência e calibra-se os escores do outro segmento utilizando os pesos gerados pela Regressão Logística. Esta etapa é então utilizada recursivamente para todas as duplas de segmentos encontrados na primeira etapa.

4.2.1.4 – *RISKSEG* - ETAPA 4 (BUSCA SEQUENCIAL)

Similar ao *framework* genérico, após todos os nós terminais passarem pelo processo de divisão, um novo nível da árvore pode ser iniciado, aplicando-se o processo para todos os novos nós, desde que não se tenha atingido nenhum critério de parada, como profundidade máxima da árvore, por exemplo.

4.2.1.5 – *RISKSEG* - ETAPA 5 (CATEGORIZAÇÃO DAS VARIÁVEIS NUMÉRICAS E O MÉTODO DE AGRUPAMENTO)

As atividades desta etapa não estão claramente definidas e descritas no *Framework* genérico, pois são tarefas complementares, discutidas e implementadas pelo método *RISKSEG*.

Assim como Árvores de Decisão, a segmentação pode utilizar variáveis numéricas para criar regras de divisão. No caso de Árvore de Decisão, o algoritmo encontra o ponto do intervalo numérico que melhor divide os dados em termos de uma métrica de separação das ocorrências da variável alvo. Tal métrica normalmente é baseada na entropia, ganho de informação, *Qui-Quadrado*, entre outras [MaR05]. Para o método *RISKSEG* a proposta é similar, porém deseja-se encontrar o ponto tal que a divisão represente os dois subgrupos mais distintos na relação entre a variável alvo e as variáveis preditoras, ou seja, encontrar o ponto em que a interação entre a variável selecionada e as demais é maior. Para isso, a variável numérica original é categorizada em *n* partes, gerando uma nova variável categórica que representa a variável numérica. O valor de *n* pode ser definido pelo especialista, que possui o conhecimento prévio das características e distribuição da variável ou pode ser definido experimentalmente.

Para complementar, foi implementado também um algoritmo que possibilita a construção de outras variáveis categóricas a partir da combinação de suas categorias. Essas novas variáveis representam uniões de intervalos numéricos e podem ser utilizadas no algoritmo de segmentação para terem suas categorias testadas como regra de “quebra”, caso a variável

numérica original esteja presente na lista de candidatas. O interessante é que esta metodologia permite fazer combinações de intervalos numéricos para definir os subgrupos. Enquanto que em Árvores de Decisão chega-se a regras de quebras do tipo ($X_i < a$) e ($X_i \geq a$), com a criação das *n* categorias da variável numérica e posterior geração de novas variáveis que combinam estas categorias, pode-se chegar a regras do tipo Segmento 1 = ($X_i \varepsilon A$ U $X_i \varepsilon C$) e Segmento 2 = ($X_i \varepsilon B$ U $X_i \varepsilon D$), onde *A, B, C* e *D* são intervalos disjuntos que compõem o domínio numérico onde a variável X_i está definida. Obviamente, quanto maior o número *n* de intervalos disjuntos, mais custosa fica esta busca pelo melhor agrupamento e definição da regra de divisão.

A Figura 4.7 ilustra um domínio numérico previamente categorizado em cinco partes, mas que teve categorias agrupadas em três, devido à similaridade quanto a interação com as outras variáveis. Esta categorização é uma das possíveis para representar a variável numérica original. O método de combinação de categorias também pode ser aplicado diretamente em variáveis categóricas, aumentando assim as possíveis opções para segmentação.

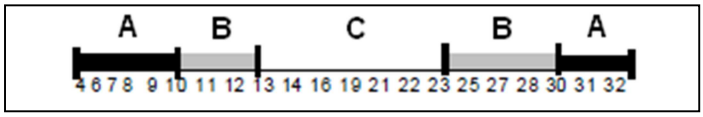


Figura 4.7 – Intervalo numérico categorizado, com agrupamento de categorias.

4.2.1.6 – *RISKSEG* - DEFINIÇÃO DOS PARÂMETROS

Os parâmetros utilizados no *RISKSEG* são em grande parte análogos aos do *framework* genérico, porém com valores particulares explícitos para o funcionamento do método. Também foram criados alguns novos para suportar o método proposto. A seguir é feita uma análise e formalização. Os parâmetros exclusivos do *RISKSEG* são precedidos da letra "*r*".

P – Variáveis candidatas selecionadas. Da mesma forma que o *framework* genérico, no início de cada iteração, é necessário um conjunto de variáveis como atributos de entrada. Desta forma, ao invés de tentar segmentar por todas as variáveis categóricas possíveis, o método deixa que o especialista escolha *a priori* quais são as variáveis elegíveis a serem utilizadas na segmentação. Com isto, um projetista experiente, por exemplo, pode simplesmente eliminar variáveis que possuam preenchimento insatisfatório, segundo sua análise. O conhecimento de possíveis interações entre as variáveis também pode ser um critério para a entrada desta variável na lista de candidatas. A restrição da entrada de variáveis através de uma pré-seleção das candidatas impede que o método faça sua escolha de forma otimizada, então, na medida do possível, recomenda-se que seja utilizado o maior número de variáveis candidatas, deixando que o método desenvolvido faça a seleção.

D – Métrica de desempenho. É a métrica de desempenho a ser calculada e avaliada para a determinação da melhor segmentação. As métricas disponíveis para escolha no método *RISKSEG* são: erro de classificação, *KS2* [Con99], *ROC* [Faw06] e *Odds Ratio* avaliada nos intervalos da distribuição dos escores [ThC02]. A utilização de métodos de combinação de preditores permite que os resultados possam ser melhorados através da escolha dos preditores que priorizem a melhoria de uma determinada métrica, mesmo que esta não possa ser utilizada diretamente no treinamento dos métodos preditores. Por exemplo, para uma análise de risco de crédito, pode-se definir que a segmentação dev*a priori*zar a detecção de uma das caudas dos escores (faixa de valores mais altos ou valores mais baixos). Assim, para o pior extremo da distribuição, pode-se praticar a rejeição dos piores clientes (potenciais) ou, no outro extremo, fazer o oferecimento de condições especiais para os melhores clientes.

M – Método de escolha. Método já descrito anteriormente, onde, após a definição inicial (manual) das variáveis *P*, é feita uma nova seleção com o método de escolha proposto, para que nem todas as variáveis de *P* precisem ser testadas de forma semi-exaustiva. Como já evidenciado, a criação deste método de busca é a principal melhoria proposta na segmentação pelo *RISKSEG*. Ele utiliza um algoritmo que escolhe as melhores variáveis a serem testadas com o uso de uma regressão contendo a interação entre a variável a ser escolhida e as demais. A seleção é feita pelo método proposto e a métrica *D* define quais são as melhores e utiliza um dos parâmetros que participa dos critérios de parada *B* para determinar o ganho mínimo estimado com a segmentação da variável.

N – Número máximo de ramos. Define o número máximo de segmentos a serem produzidos pela regra de quebra. No *RISKSEG* sugere-se a utilização de dois para que sejam geradas árvores binárias. A implementação do método neste trabalho foi realizado utilizando esta configuração.

f – Função de combinação ou *meta-learner*. Os escores gerados para cada segmento precisam ser calibrados e alinhados, para que a métrica *D* de desempenho possa ser calculada e o resultado obtido com a segmentação possa ser mensurado e comparado entre as outras possibilidades de segmentação e com o modelo do nó superior (sem divisão). No *RISKSEG* essa junção pode ser feita utilizando-se *Stacking* ou recalibração dos escores por *Marginal Odds*.

B – Critérios de parada. Os parâmetros utilizados como critérios de parada no *RISKSEG* são: número mínimo de registros presentes em um segmento; ganho mínimo ou perda máxima aceitável de desempenho em uma segmentação, em termos percentuais; e profundidade máxima da árvore de estrutura de segmentação. O parâmetro quantidade mínima de exemplos para

formação de segmento serve tanto para evitar que modelos com pouca representatividade sejam treinados, quanto para evitar que categorias sem quantidade relevante de exemplos sejam testadas. O parâmetro número máximo de níveis também ajuda no controle do tempo de execução, pois limita a profundidade da árvore. Em testes realizados com este método verificou-se que o ganho a cada novo nível é normalmente decrescente e, a partir de certo nível, ele é desprezível ou inexistente. Dependendo da aplicação que está sendo desenvolvida e com o uso contínuo do método, os especialistas podem determinar com certo grau de certeza a profundidade máxima adequada para a árvore. Esta profundidade depende principalmente do número de casos disponíveis para desenvolvimento do modelo. A taxa mínima de ganho determina o ganho mínimo ou a perda máxima, após a segmentação. Caso o ganho mínimo seja alcançado, o nó será dividido. Mesmo que haja uma leve piora nos resultados da segmentação (perda máxima), é possível que o nó seja dividido, pois existe a possibilidade de melhoria após divisões sucessivas, ou seja, aceitar perdas razoáveis evita que o método busque apenas máximos locais [MaR05].

Indutores – Conjunto de preditores. Os classificadores gerados em cada segmento na etapa 2 podem ser escolhidos entre Regressão Logística, Regressão Linear, Redes Neurais, Árvore de Decisão ou qualquer outra técnica que permita a geração de escores. Também pode ser utilizada mais de uma técnica no treinamento de uma mesma árvore. É importante salientar que, quanto maior o número de técnicas utilizadas simultaneamente, maior será o custo computacional para o seu treinamento. Então, se a massa de dados é muito grande e não há muito tempo, o melhor é a utilização de apenas uma técnica. Quando se deseja fazer a implantação de uma árvore em produção contendo vários métodos preditivos distintos, é importante observar a disponibilidade de recursos para que cada uma delas possa ser programada neste ambiente. Por exemplo, para cálculo de escores em tempo real, em uma linha de produção industrial usando ambiente *mainframe* com linguagem COBOL, métodos mais sofisticados como Redes Neurais podem ser mais complicados para realização desta tarefa, ao passo que Regressões Estatísticas e Árvores podem ser mais facilmente distribuídas.

rQtdeDivisões – Quantidade de intervalos de variáveis numéricas. Este parâmetro determina o número máximo de intervalos que serão criados na transformação automática de variáveis numéricas em variáveis categóricas. O método proposto tenta distribuir a quantidade de elementos das variáveis numéricas o mais igualmente possível em *n* intervalos categóricos. Por exemplo, se *n* é igual a 4 (quatro), tem-se quartis.

rUsaBlocos – Lista de variáveis que terão suas categorias submetidas aos testes de agrupamento para segmentação. Caso determinada variável esteja na lista *rUsaBlocos*, o

algoritmo permitirá a junção de algumas de suas categorias no processo de teste de melhor divisão. O número máximo de categorias agrupadas também é variável e é determinado pelo parâmetro *rQtdeBlocos*. Por exemplo, se *rQtdeBlocos* é igual a 3 (três), até 3 (três) categorias agrupadas podem ser testadas como regra de divisão. Apesar das vantagens discutidas em 4.2.1.5, é importante atentar para o aumento de custo computacional no treinamento, quando as variáveis selecionadas possuem muitas categorias e o valor de *rQtdeBlocos* é alto. A quantidade de agrupamentos possíveis a serem testados em uma variável é dada pela equação:

$$\binom{x}{mín(y,\lfloor\frac{x}{2}\rfloor)}+\binom{x}{mín\left(y,\lfloor\frac{x}{2}\rfloor\right)-1}+\binom{x}{mín\left(y,\lfloor\frac{x}{2}\rfloor\right)-2}+\cdots+\binom{x}{2}$$

Tal que $x \geq 4$ é o número de categorias da variável e $y \geq 2$ é o valor de *rQtdeBlocos*. Note que se $x < 4$, qualquer agrupamento é análogo a testar cada categoria contra as demais, ou seja, sem este recurso de blocagem.

rQtdeBlocos – Quantidade de blocos. Parâmetro para determinar o número máximo de categorias que poderão ser utilizadas nos testes para criar os segmentos, nas variáveis pertencentes ao parâmetro *rUsaBlocos*. São exemplos de divisões possíveis geradas (ou somente testadas), para *Qtd* e *UF*, onde *Qtd* tem domínio inteiro e *UF* tem domínio categórico com cinco categorias distintas:

Segmentação 1: segmento1 [Qtd < 1 e Qtd > 4] e segmento2 [Qtd >=1 e Qtd <=4]

Segmentação 2: segmento1 [UF ϵ {‘BA’, ‘CE’, ‘SP’}] e segmento2 [UF ϵ {‘RJ’, ‘MG’}]

rQtdeVarTeste – Quantidade de variáveis selecionadas para teste de segmentação. Este parâmetro determina a quantidade de variáveis que serão pré-selecionadas pelo método *M*. Se o valor deste parâmetro for igual a 1 (um), o método *M*, utilizará apenas a melhor variável selecionada, como mostra o algoritmo 4.1. Este parâmetro também impacta diretamente no custo computacional do treinamento, pois quanto maior o número de variáveis a serem testadas pelo processo de divisão dos segmentos, maior será o tempo de treinamento. Este tempo também será fortemente influenciado pela quantidade de categorias das variáveis pré-selecionadas pelo método *M* e se a variável pré-selecionada está na lista do parâmetro *rUsaBlocos* ou não.

rTécnicaFatorial – Técnica de predição utilizada pelo método *M*: este parâmetro determina qual técnica será utilizada para seleção das variáveis candidatas à segmentação do nó.

Nas Seções 4.1.1.1 e 4.1.1.2, foram discutidos os usos da Regressão Logística e Regressão Linear para este fim. Porém, apesar de não implementadas neste trabalho, outras técnicas como RNA e *SVM* [MLH03] podem ser testadas.

rTamValidação – Percentual do conjunto de treinamento utilizado para validação: este parâmetro determina o tamanho do conjunto de validação da técnica. Os registros da validação são utilizados tanto para cálculo da métrica de desempenho *D* quanto nas técnicas de aprendizagem de máquina, que necessitam deste tipo de conjunto em seu processo de treinamento, como é o caso de Redes Neurais, que o utiliza para mitigação do super treinamento (*overfitting*).

4.3 – AVALIAÇÃO CRÍTICA E COMPARAÇÃO COM OUTROS MÉTODOS

Algumas análises críticas já foram feitas em seções anteriores, principalmente quando foram apresentados os parâmetros do *RISKSEG*. Certamente, a vantagem mais perseguida e desejada é o incremento do poder preditivo, quando comparado com outras metodologias de construção de modelos que utilizam apenas um preditor ou uma combinação, como *Bagging* e *Boosting*. Nesta seção também são destacadas outras vantagens, neste contexto de comparações, bem como algumas limitações do método proposto.

O modelo segmentado pode ser comparado a um modelo fatorial com interações entre as variáveis preditivas. No entanto, um modelo construído a partir de uma combinação fatorial torna o número de variáveis extremamente elevado. Este pode ser um "preço" alto a se pagar, para o aproveitamento das interações. Por isto, o método propõe aproveitar este potencial utilizando segmentações de bases. Desta forma, são selecionadas somente as interações entre as variáveis que se mostrarem relevantes para o aumento do desempenho. A segmentação elimina o efeito da variável (ou da categoria) nos modelos seguintes, o que reduz a complexidade dos próximos treinamentos, pois retira-se uma das categorias do treinamento do modelo. Assim, tem-se um conjunto de modelos criados com menor número de variáveis que o modelo genérico inicial (sem segmentações) e com muito menos variáveis que o modelo fatorial completo.

A quantidade de variáveis presentes no modelo fatorial completo, quando se testa todas as interações de todos os graus, é de $(2^n - 1)$, onde n é o número original de variáveis preditivas, considerando variáveis categóricas dicotômicas. Por exemplo, as equações (1) e (2) apresentam uma comparação do ajuste teórico de um modelo com três variáveis principais (1) com um modelo fatorial de nível três (2), com mais 4 (quatro) variáveis resultantes das possíveis

combinações entre elas, onde β_y representa os pesos ajustados de um modelo linear e v_z as variáveis preditivas binárias.

$$\mathbf{Y} = \beta_1 v_1 + \beta_2 v_2 + \beta_3 v_3 \quad (1)$$

$$\mathbf{Y_{Fatorial3}} = \beta_1' v_1 + \beta_2' v_2 + \beta_3' v_3 + \beta_4 v_1 v_2 + \beta_5 v_1 v_3 + \beta_6 v_2 v_3 + \beta_7 v_1 v_2 v_3 \quad (2)$$

Se o foco da modelagem se resumir à melhoria do poder preditivo, com a adição de variáveis interativas relevantes no modelo, pode-se negligenciar outra importante necessidade de algumas aplicações de modelos de risco, que é a capacidade de explicação do escore a partir das variáveis preditivas. Em modelos de risco de crédito, um dos principais pré-requisitos para seu uso é a possibilidade de interpretar os resultados, quando solicitado [LAW08]. É muito importante saber como o modelo obteve um escore para um determinado exemplo. Normalmente, isto acontece por solicitação da parte insatisfeita com o valor do seu escore. Por exemplo, se o seu escore aponta para um alto risco de inadimplência, o detalhamento do valor pode chegar inclusive a ser feito por força de lei, onde o juiz solicita à empresa desenvolvedora do escore que explique como a pontuação foi gerada. Uma quantidade de variáveis muito grande e, principalmente, muitas combinações entre elas (interações) dificultam esta tarefa. Para os casos em que esta exigência se faz necessária, os modelos construídos com este método podem ser facilmente utilizados. Note que a segmentação, além de aumentar o poder preditivo, pode também diminuir o número de variáveis que serão utilizadas para calcular o escore, sem deixar de levá-las em consideração.

Necessitando apenas que o escore gerado seja explicado a partir do segmento sensibilizado, algumas técnicas como Árvores, Regressão Logística e Linear são particularmente mais fáceis para esta tarefa. Outras técnicas, como Redes Neurais, também podem ser utilizadas, pois diversos trabalhos publicados fornecem ferramentas para que isso seja possível [SaW07] [GEE09].

A Tabela 4.1 resume algumas comparações de métricas qualitativas entre o *RISKSEG* e alguns dos métodos de combinação e geração de classificadores múltiplos mais utilizados (*Bagging*, *Boosting* e *Stacking*). A primeira característica comparada é a utilização de diferentes técnicas de regressão/classificação no emprego desses métodos. *Bagging* e *Boosting* normalmente utilizam-se apenas de um tipo de técnica de predição em suas iterações. O *RISKSEG* e *Stacking* são métodos mais plausíveis a utilizarem diferentes técnicas preditoras, pois possuem um desenho natural para mistura de modelos especialistas. Óbvio que, se forem utilizados diferentes preditores para descoberta dos melhores modelos de nó, isto aumentará

bastante o tempo para treinamento e, possivelmente, reduzirá a capacidade explicativa, se algum dos algoritmos escolhidos não possuir esta característica (*kNN*, por exemplo).

Tabela 4.1 – Comparação entre os principais métodos de combinação de preditores.

Técnica / Métrica	Trabalha com técnicas diferentes?	Dificuldade na distribuição (produção)	Interpretação do escore	Tempo de treinamento	Tempo relativo de processamento em produção	Quantidade de modelos executados por caso
RISKSEG	Sim	Baixa*	Fácil*	Alto	Baixo	1
Bagging	Não	Alta	Complexa	Alto	Alto	Grande
Boosting	Não	Alta	Complexa	Alto	Alto	Grande
Stacking	Sim	Média	Complexa	Médio	Médio	>= 3

* Dependente da técnica de aprendizagem empregada

Outra característica qualitativa importante para comparação é a dificuldade na aplicação dos métodos em ambiente de produção em tempo real, pois a complexidade depende mais das técnicas utilizadas e da disponibilidade de geração automática de códigos por parte da ferramenta do que da implementação da árvore de modelos do método. Em ambiente com milhões de cálculos de escores, em curtos intervalos de tempo, a rápida execução é condição crítica. Neste quesito, o tempo do *RISKSEG* dependerá apenas das técnicas que foram empregadas em cada segmento, uma vez que somente um modelo é executado por vez (execução excludente). Isto porque a árvore de modelos é na verdade um conjunto de regras que designa para qual preditor cada exemplo deve ser submetido. *Bagging* e *Boosting*, por exemplo, necessitam que um exemplo seja submetido a todos os preditores e depois as suas saídas sejam consolidadas para compor o resultado final. O *Stacking* tende a ter menos execuções que estes dois métodos, uma vez que, normalmente, se utiliza um método preditor de cada técnica diferente (geralmente não são muitas) mais um preditor para fazer a junção final dos resultados intermediários. Desta forma, se os diferentes métodos de combinação utilizam as mesmas técnicas de predição, o *RISKSEG* terá um tempo médio de execução bem menor e equivalente aos modelos com apenas um preditor.

Quanto à interpretação do escore final, que neste caso é definido como sendo a capacidade que as técnicas preditoras possuem para demonstrar o quão simples é explicitar como o valor do risco foi obtido, o *RISKSEG* também é bem mais atrativo neste aspecto, uma vez que os demais métodos de combinação necessitam fazer a junção de diferentes respostas provenientes de diferentes modelos, o que torna esta tarefa não muito simples de ser mostrada matematicamente.

Apesar de todas as vantagens que a segmentação de modelos pode proporcionar, a técnica tem como resposta uma árvore de segmentação. Esta quantidade de modelos implica no

aumento da complexidade relacionada com a sua administração, implantação, homologação e monitoramento, quando comparado com a implementação de um modelo com apenas um preditor.

O tempo de treinamento para se chegar às segmentações pode ser bem grande, mesmo utilizando-se apenas um tipo de classificador (Redes Neurais, por exemplo). Isto é uma clara desvantagem em relação ao uso de apenas um classificador simples (sem nenhum tipo de combinação), mas este tempo é comparável, senão menor, aos demais métodos de combinação de classificadores. Este problema pode ser reduzido para o método proposto, pois o *RISKSEG* possui a vantagem de conseguir reduzir sensivelmente este tempo, através da manipulação de alguns de seus parâmetros exclusivos (já detalhados).

A não compatibilidade entre as escalas dos escores de diversas técnicas de regressão pode exigir um esforço extra para transformar estes escores em escalas de probabilidades ou algum tipo de compatibilização. Este conceito dificulta o uso de técnicas de regressão que não tenham como saída valores de distribuição de escores compatíveis e, principalmente, dificulta a combinação de diferentes técnicas na mesma árvore de segmentação.

Uma das grandes vantagens da segmentação é a possibilidade de usar qualquer método preditor. O *RISKSEG*, em si, não é uma técnica de regressão ou classificação e sim um método para combinar estas técnicas. A possibilidade de utilização de métodos mais adequados para determinadas situações como, por exemplo, casos em que exista grande quantidade de dados faltantes (*missing values*). Assim, pode-se simplesmente escolher técnicas que possuam robustez a este tipo de característica na base de dados, desta forma a robustez será no mínimo igual à técnica empregada pelo *RISKSEG*.

Uma outra possibilidade do *RISKSEG* é a codificação (mapeamento) dos valores nulos em variáveis, desta forma, o método pode decidir se a melhor segmentação deve ser com estas variáveis criadas para representar os nulos ou com as demais. Obviamente, esta abordagem só é válida quando a ausência de informações nas variáveis também se reflete quando os modelos estão sendo utilizados. Na prática, é muito comum, por exemplo em crédito, fazer uma segmentação manual por suficiência de dados, clientes e não clientes, por exemplo [ThC02]. Nesta situação, é comum especialistas ficarem no dilema entre ter dois modelos particionados ou um modelo só misturando registros com mais ou menos variáveis de entrada preenchidas. Quando se têm métodos de particionamento de segmentos, como o RISKSEG, o próprio algoritmo pode decidir se esta divisão é mais relevante para o aumento do desempenho.

Uma característica do *RISKSEG* é a utilização apenas de variáveis categóricas. Então, para tornar o método aplicável às variáveis categóricas e numéricas, foi utilizado um método para fazer categorização de variáveis numéricas. Essa implementação potencializa o método proposto, pois executa testes de transformações que normalmente são ignoradas em outros métodos. Este método já foi detalhado na seção anterior deste capítulo.

Uma limitação do método são as restrições com relação ao uso de amostras com pequena quantidade de elementos, pois, como uma das características do método é a divisão sucessiva da amostra em segmentos de dados disjuntos, esta redução pode gerar subgrupos com um número muito reduzido de elementos, podendo comprometer o desempenho dos classificadores. Muitos estudos foram publicados sobre o impacto causado pela redução de tamanho amostral, um deles é o artigo [Kia03]. Isto pode limitar o seu uso em problemas com poucos exemplos disponíveis para geração do escore. Esta limitação talvez possa ser contornada utilizando-se técnicas de re-amostragem, disponíveis na estatística, ou reduzindo a quantidade se segmentações. Nos capítulos de estudo de caso são utilizadas bases de dados com poucos exemplos e são feitas análises para este tipo de problema.

4.4 – CONSIDERAÇÕES FINAIS

Este capítulo descreveu em detalhes o método *RISKSEG*. Em seguida foi desenvolvida uma implementação do método proposto, que posteriormente utilizou-se nos Capítulos 5, 6 e 7 (experimentais). Os resultados dos experimentos com o *RISKSEG*, nestes capítulos, são comparados com as técnicas individuais, descritas no Capítulo 2, e com os métodos de combinação de preditores, mostrados no Capítulo 3.

CAPÍTULO 5

ESTUDO DE CASO UTILIZANDO BASES SIMULADAS

Este capítulo tem o objetivo de realizar experimentos com bases controladas, para medir o desempenho do método de segmentação proposto em comparação com outros métodos de geração de múltiplos classificadores. Os resultados são submetidos às análises estatísticas para a determinação da significância dos números encontrados. As bases utilizadas são simuladas com algumas características consideradas importantes para a análise. Estas bases foram criadas e adaptadas a partir de simulações e experimentos realizados por outros autores e publicados anteriormente [CGM02] [Kia03]. Foram feitos grupos de experimentos com quatro bases de características diferentes.

Os experimentos foram estruturados de forma que a técnica pura (Regressão Linear, Regressão Logística ou Redes Neurais) fosse utilizada como base de comparação com os demais métodos de combinação de classificadores (*Bagging*, *Boosting* e Segmentações). Nos métodos de combinação que superam significativamente o classificador base (controle) são realizados testes de significância para determinar se existe diferença entre estes. Em outras palavras, deseja-se saber se uma técnica de combinação supera a outra. Estas comparações não são disponibilizadas diretamente nas tabelas de resultados, entretanto elas são analisadas e discutidas no texto.

5.1 – AMOSTRAS E ESTRUTURAS DE DEPENDÊNCIA EXPLORADAS

Para os conjuntos de dados simulados, foram feitas 30 replicações de cada experimento (k), a fim de se calcular intervalos de confiança e testes de hipóteses. Cada replicação utilizou uma base gerada independentemente das demais. Nestes experimentos foram exploradas características quanto à quantidade de registros disponíveis no conjunto de dados e diferentes relações entre a variável alvo e as preditoras. O intuito é comparar o desempenho do método proposto em bases controladas, verificar em que casos o *RISKSEG* apresenta resultados superiores aos outros métodos, e observar a variância dos diferentes tamanhos de amostra.

Todas as bases utilizadas contêm variáveis alvo dicotômicas, pois este é um dos focos de implementação dos experimentos desenvolvidos nesta tese. Os tamanhos de amostra explorados foram 1.000, 3.000 e 5.000 registros. Este parâmetro é referenciado como *n*, neste capítulo. Para cada conjunto de dados foram definidos conjuntos de treinamento, validação e teste, na proporção 50%, 25% e 25% [Pre94], respectivamente.

A seguir, são formalizadas e descritas as formas pelas quais foram construídas as bases de dados utilizadas neste capítulo.

Base (1): Forte Interação entre Variáveis Preditoras

Neste tipo de relação entre preditoras e o alvo, espera-se que a segmentação consiga melhoras consideráveis no poder preditivo das técnicas de predição. Os conjuntos de dados utilizados para os experimentos foram gerados segundo a seguinte regra:

$$Y = \begin{cases} 1, \text{se } \dfrac{1}{1+\exp(h)} > 0{,}5 \\ 0, \text{se } \dfrac{1}{1+\exp(h)} \leq 0{,}5 \end{cases}$$

onde Y é a variável alvo dicotômica e *h* é definido da seguinte forma:

$$h = \mu_1 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_7 x_7 + \beta_8 x_8 + \beta_9 x_9 + \beta_{10} x_{10} + \gamma_1 x_1 x_6 + \gamma_1 x_1 x_8 + \gamma_2 x_2 x_8 + \gamma_3 x_2 x_9 + \gamma_4 x_5 x_7 + \gamma_5 x_6 x_7 + \alpha_1 x_4 x_5 x_6 + \alpha_2 x_8 x_9 x_{10} + \alpha_3 x_2 x_4 x_8 + \varepsilon$$

Note-se que na fórmula acima, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9 e x_{10} são variáveis aleatórias discretas que assumem valores de duas a cinco categorias, variando de uma para outra segundo funções de distribuição de massa, ε é um fator de erro onde $\varepsilon \sim N(\theta, \sigma^2)$, com μ_1, θ e σ^2 fixos escolhidos arbitrariamente.

Nota-se que foram consideradas algumas interações de ordem 2 e 3 na construção da base simulada. Um modelo linear simples não utilizaria todo o potencial da base para capturar a relação de *Y* com as variáveis preditoras, desde que estas interações não fossem consideradas no momento de determinar a forma do modelo. É possível que algumas técnicas como Redes Neurais sejam capazes de capturar a existência dessas interações sem mesmo considerá-las de antemão no planejamento do treinamento. Por este e outros motivos, foi utilizado Redes Neurais nos experimentos. Com esta técnica é possível observar e estudar o comportamento dos métodos de combinação quando aplicados a indutores já construídos para captar não-linearidades.

Base (2): Não Linear

Baseado em um artigo que propõe uma abordagem bayesiana para segmentar [CGM02], este é outro exemplo de base de teste com forte não linearidade, para verificação do ganho ao utilizar-se múltiplos classificadores. O *Y* desta base foi gerado da mesma forma que a base (1), porém com *h* definido como:

$$h = \mu_2 + 10\,\mathrm{sen}(\pi\, x_1 x_2) + 20(x_3 - 0{,}5)^2 + 10x_4 + 5x_5 + \varepsilon_1$$

onde x_1, x_2, x_3, x_4, x_5 são variáveis aleatórias com densidade $U(0,1)$, μ_2 fixo escolhido arbitrariamente, e $\varepsilon_1 \sim N(0,1)$ é o fator de ruído adicionado.

Nota-se que nesta base de dados utilizam-se funções como seno e multiplicação de variáveis, criando uma estrutura não-linear. Neste caso também se espera que existam ganhos significativos no poder de classificação quando se utilizam segmentações, que dividiram o problema complexo em pedaços com menor complexidade.

Base (3): Efeitos Aditivos (Linear)

Retirada e adaptada de um artigo de comparação de classificadores [Kia03], esta base tem o tipo de relação mais simples entre as variáveis preditoras e a variável dependente. O valor da variável alvo é afetado apenas por uma soma de fatores. No caso de variáveis dicotômicas, é a razão de chances de uma classe que é afetada por tal soma de fatores. Para os experimentos, foram geradas bases de dados com a razão de chances definida como:

$$h = \mu_3 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + \omega_4 x_4 + \varepsilon_2$$

onde x_1, x_2, x_3, x_4 são variáveis geradas da mesma forma que nas bases (1) e (2), bem como o ruído adicionado ε_2 e o intercepto μ_3.

Neste tipo de dependência simples entre alvo e preditoras, as técnicas como as Regressões Estatísticas e a Rede Neural são capazes de capturar facilmente a relação entre elas. Espera-se que técnicas de segmentação não apresentem ganhos significativos de desempenho de classificação. Desta forma, uma estrutura simples (apenas um modelo) não precisaria ser decomposto em um conjunto de modelos.

Base (4): Efeitos Quadráticos

Esta é mais uma base adaptada do artigo [Kia03], gerada com a seguinte razão de chances para a variável alvo:

$$h = \mu_4 + \sigma_1 {x_1}^2 + \sigma_2 {x_2}^2 + \sigma_3 {x_3}^2 + \sigma_4 {x_4}^2 + \varepsilon_3$$

onde x_1, x_2, x_3, x_4 são variáveis geradas da mesma forma que nas bases (1) e (2), bem como o ruído adicionado ε_3 e o intercepto μ_4.

Regressões e técnicas que não consideram o efeito quadrático das variáveis podem não capturar toda a relação entre alvo e preditoras, não atingindo níveis de desempenho desejados.

5.2 – PARÂMETROS E EXPERIMENTOS

5.2.1 – PARÂMETROS PARA AS TÉCNICAS DE PREDIÇÃO INDIVIDUAIS.

A arquitetura utilizada nas Redes Neurais treinadas nos experimentos foram *Multilayer Perceptron* (*MLP*) com uma camada intermediária de neurônios. Para a seleção da melhor configuração das Redes Neurais, foram experimentados os algoritmos de treinamento *Backpropagation* e *Levenberg-Marquardt* como método de aprendizagem, e testados 3, 10 e 20 neurônios na camada intermediária. A função de ativação utilizada foi a sigmóide logística e o erro mínimo quadrático com validação cruzada como critério de parada. Estes algoritmos de aprendizagem e o número de neurônios da camada intermediária são comumente utilizados em artigos de comparação de métodos preditores [Kia03] [AVA04] [Hay09]. Utilizou-se a taxa de aprendizado padrão do software SAS Enterprise Miner [SAS02] versão 4.0, que não pode ser alterada nesta versão. Após as experimentações com a variação dos parâmetros foi escolhida a melhor configuração para o grupo de experimentos.

Para Regressão Logística e Regressão Linear não utilizou-se *Stepwise* [DrH81] ou qualquer outro tipo de técnica para redução de dimensionalidade, pois o número de variáveis disponíveis nestas simulações é reduzido e as bases foram geradas de maneira que os alvos tivessem relações significativas com todas as variáveis envolvidas. Outra justificativa para não utilização destes métodos é manter o mesmo conjunto de variáveis em todas as técnicas.

5.2.2 – PARÂMETROS PARA OS MÉTODOS *DE COMBINAÇÃO*

Para a implementação dos algoritmos de combinação (*Bagging*, *Boosting*, *NNTree* e segmentações) foi utilizado o software SAS Base/Stat v9.1.3 SP 4 com SAS Enterprise Miner v4.0 [SAS02] e os resultados (tabelas e gráficos) foram gerados no Microsoft Excel versão 2007 Enterprise.

O método *Bagging* utiliza dois principais parâmetros: tamanho da re-amostra e quantidade de re-amostras (iterações do método). O primeiro parâmetro foi definido como o

mesmo tamanho da base original, ou seja, foram feitas re-amostragens de tamanho total, porém com reposição. Essa definição é uma das mais utilizadas na literatura quando se trata de aplicação de *Bagging* [Qui96]. O parâmetro de quantidade de re-amostras foi definido como 25, pois diversos trabalhos publicados utilizam números como 10 [Qui96], 25 [BaK98], 50 [Bra98] e 100 [FrS96], além disso, pré-experimentos realizados indicaram que após 25 iterações não ocorreram grandes alterações nos resultados para as bases utilizadas. Para o *Boosting* também foram utilizadas 25 re-amostras, pelo mesmo motivo já citado anteriormente.

Utilizou-se três métodos de segmentação e a principal diferença entre eles é a forma pela qual é feita a escolha das divisões dos segmentos. Uma das segmentações utilizadas foi o método *NNTree*, que utiliza convencionalmente uma Rede Neural *MLP* para fazer a divisão do conjunto de treinamento. Porém, nestes experimentos, além da Rede Neural, também utilizou-se Regressão Logística e Linear para cumprir a mesma função, através de uma implementação baseada em uma adaptação do método original [Pra08]. Estas Regressões também foram aplicadas nas outras duas formas de segmentação.

A segunda forma de segmentação segue a recomendação mais clássica, também aplicada em Árvores de Decisão, com a utilização de *Information Gain* [Nev98] [Nev99] [CGM02] como métrica para definição de melhor divisão.

A terceira segmentação utilizou o método proposto (*RISKSEG*). Para a profundidade máxima (número de níveis) dos métodos de segmentação utilizou-se 3 (três), pois, através de testes preliminares, verificou-se que as melhores segmentações estariam entre os níveis 1 e 3. O desempenho dos diferentes níveis é apresentado em uma tabela para a escolha do nível que apresenta melhor desempenho, ou seja, menor erro de classificação.

Para as segmentações por *Information Gain (SegTree)* e *RISKSEG*, procurou-se utilizar o máximo de semelhança possível nos parâmetros para que o objeto principal da tese fosse avaliado. Para a quantidade mínima de instâncias de uma folha utilizou-se 10%, pois entendeu-se que este valor é suficiente para os tamanhos de amostras utilizadas. O ganho mínimo para continuar a segmentação foi definido como 0 (zero), ou seja, qualquer ganho encontrado na tentativa de segmentação é suficiente para permitir que o algoritmo continue. A métrica para o cálculo deste ganho foi a diminuição do erro de classificação, pois ela também é a principal métrica das tabelas de avaliação de desempenho dos experimentos. Os métodos de segmentação utilizaram um conjunto de validação para determinar se houve ou não ganho e, se for o caso, segmentar (*rTamValidação*). Para esta medição foi usado um conjunto de validação com

tamanho de 25% do total da amostra. Este conjunto é o mesmo utilizado no treinamento da Rede Neural.

Os parâmetros utilizados para as simulações envolvendo o método de segmentação *RISKSEG* são descritos a seguir. A função de combinação ou *meta-learner* para alinhamento dos escores gerados a partir da variável alvo dicotômica, em cada segmento, foi o *Stacking* com Regressão Logística como indutor. Esta técnica foi escolhida por possuir um tempo reduzido de treinamento quando comparado, por exemplo, com Redes Neurais, e melhores desempenhos quando comparado com Regressão Linear. Para a quantidade de variáveis a serem selecionadas e posteriormente testadas na definição da melhor segmentação (*rQtdeVarTeste*), utilizou-se valor igual a 3 (três). Em outras palavras, a cada iteração, as três melhores variáveis definidas pela regressão, que realiza a seleção de variáveis (análise de interações), são testadas para definir a melhor a divisão dos conjuntos de treinamento.

Foram utilizados agrupamentos de categorias de variáveis para os testes de segmentação (*UsaBlocos*). O número máximo de categorias agrupadas (*rQtdeBlocos*) para os testes foi de 2 (dois). Apesar destes parâmetros serem característicos do *RISKSEG*, eles também foram utilizados na segmentação por *Information Gain*, pois entende-se que eles poderiam ser obtidos de forma manual (e trabalhosa) no pré-processamento de dados. Assim, permitiu-se uma comparação mais justa entre os métodos de segmentação, além de uma melhor avaliação do efeito do método de divisão de dados exclusivo do *RISKSEG*.

5.2.3 – TRATAMENTO DAS VARIÁVEIS

Todas as variáveis deste capítulo são numéricas e foram criadas para terem domínio de valores contínuos entre 0 (zero) e 1 (um), desta maneira, não é necessária nenhuma transformação extra para treinamento dos métodos de classificação. Apesar das implementações *RISKSEG* e *SegTree* suportarem o parâmetro *rQtdeDivisões* (ver seção 4.2.1.6), que faz automaticamente a categorização das variáveis numéricas para divisão dos subconjuntos (segmentar), optou-se por fazer esta transformação previamente, pois a quantidade de experimentos é grande, o que aumentaria bastante o tempo de treinamento, uma vez que esta operação teria que ser realizada várias vezes no código interno do algoritmo. Então, as variáveis numéricas foram categorizadas de modo a possuírem quatro categorias, cada uma com aproximadamente 25% dos exemplos. Estas variáveis categorizadas são utilizadas apenas para segmentação, pois, nos treinamentos das técnicas, elas são usadas em sua forma numérica, que já se encontra com valores entre 0 (zero) e 1 (um).

5.2.4 – ORGANIZAÇÃO DOS EXPERIMENTOS

Para se medir os efeitos causados no desempenho de classificação pelos diversos métodos de geração e combinação de classificadores múltiplos, os experimentos foram feitos de modo a se obter medidas pareadas das taxas de erro, ou seja, para cada unidade experimental, aplicaram-se todos os métodos de interesse para posterior comparação. Os resultados individuais de cada experimento foram organizados em tabelas de resultados contendo a taxa média de erro de cada classificador e o intervalo de confiança.

Nas tabelas de resultados, os desempenhos significativamente melhores ou piores, em relação ao método base, foram colocados em negrito, sendo que para representar uma melhora significativa foi acrescentado um asterisco no valor e para indicar uma piora de desempenho significativa, usou-se itálico nos valores. Como forma de simplificar o título das tabelas, chama-se a segmentação por *Information Gain* de *SegTree*.

Como citado anteriormente, 30 experimentos foram feitos para cada possível combinação de tamanho de amostra, estrutura de dependência e técnica de classificação. A Tabela 5.1 mostra a organização dos experimentos quanto ao número, sem levar em consideração os diferentes métodos de combinação de modelos. Apenas como uma estimativa para determinar a quantidade de modelos que foram treinados (somente) nos experimentos finais deste estudo de caso, pode-se multiplicar o número de replicações (k), pela quantidade de técnicas preditoras simples (p), pelos tamanhos de amostras (a), pelos diferentes métodos de combinação utilizados (c) e pelas diferentes combinações de parâmetros das Redes Neurais (r). Então tem-se: *kpacr* = 8.100, onde *k=30, p=3, a=3, c=5 e r=6.* Esta equação não leva em consideração o treinamento dos vários classificadores que são utilizados nos métodos de combinação, pois *Bagging* e *Boosting* têm número fixo de 25 e os demais métodos de combinação (por segmentação) possuem variação na quantidade de classificadores internos treinados, já que dependem do seu método de treinamento, características dos dados e parâmetros utilizados. Apenas para ilustrar e estimando-se por baixo, toma-se que o número médio de modelos simples utilizados durante o processo de treinamento das segmentações seja pelo menos 20, então este número multiplicado pela quantidade de métodos de combinação utilizados, que foi 8.100, tem-se pelo menos 162.000 modelos simples treinados dentro dos métodos de combinação, somente para este capítulo.

Tabela 5.1 – Grupos de experimentos.

Base	Regressão Logística			Regressão Linear			Redes Neurais		
	n=1.000	*n*=3.000	*n*=5.000	*n*=1.000	*n*=3.000	*n*=5.000	*n*=1.000	*n*=3.000	*n*=5.000
(1)	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.
(2)	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.
(3)	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.
(4)	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.	30 exp.

5.3 – SELEÇÃO DOS PARÂMETROS E ANÁLISES DOS RESULTADOS

Nesta seção, para cada base, são apresentadas três tabelas para seleção dos melhores parâmetros de cada técnica de treinamento cruzando com os método de combinação e uma tabela comparativa final com todas as técnicas e modelos escolhidos consolidadas.

As tabelas de resultados apresentadas nesta subseção contêm as médias das taxas de erro encontradas pelos experimentos, bem como os intervalos de confiança associados. Tais intervalos foram calculados a um nível de 95%. As tabelas permitem comparar os resultados entre os diferentes métodos de preditores múltiplos e observar o comportamento dos resultados de acordo com o tamanho da amostra utilizada. Valores de médias menores que a do classificador Simples são colocados em negrito e se os valores forem estatisticamente diferentes, são indicados com asterisco. Como as técnicas são aplicadas às mesmas unidades experimentais, os testes de hipóteses foram realizados considerando-se amostras pareadas. Utilizou-se o *t-Student* [Con99] de amostras pareadas com um nível de significância de 5%.

5.3.1 – RESULTADO: BASE (1) – INTERAÇÃO ENTRE VARIÁVEIS PREDITORAS.

Nas Tabelas 5.2, 5.3 e 5.4 são apresentados os erros médios para cada uma das configurações e, em negrito, são evidenciados os que obtiveram menor erro para a técnica experimentada e, consequentemente, os que foram escolhidos para a comparação entre os métodos de combinação de classificadores. Esta convenção é utilizada para seleção dos parâmetros das Redes Neurais e quantidade de níveis dos métodos de segmentação, em todos os experimentos deste capítulo.

Tabela 5.2 – Erro médio para seleção da melhor configuração da Rede Neural - Base (1).

Registros	Método	Nível	*Backpropagation*			*Levenberg-Marquardt*		
			3 neur.	10 neur.	20 neur.	3 neur.	10 neur.	20 neur.
1.000	Simples	-	10,75	10,27	**10,05**	10,81	10,47	10,48
	Boosting	-	9,84	**9,17**	9,21	9,79	9,61	9,55
	Bagging	-	**7,89**	8,25	8,19	8,12	7,95	7,93
	NNTree	1	10,53	10,15	10,01	10,65	10,37	10,25
		2	10,31	9,93	9,94	10,45	10,33	10,09
		3	10,18	**9,80**	9,90	10,34	10,30	9,99
	RISKSEG	1	11,37	10,13	**10,01**	11,41	10,36	10,23
		2	11,29	10,28	10,04	11,47	10,43	10,32
		3	11,08	10,28	10,08	11,36	10,48	10,31
	SegTree	1	10,47	**9,87**	10,33	10,80	10,25	9,97
		2	10,77	10,24	10,37	11,05	10,53	10,41
		3	10,63	10,33	10,43	11,05	10,59	10,44
3.000	Simples	-	7,95	7,89	7,56	8,11	8,07	**7,38**
	Boosting	-	7,16	7,30	**6,95**	7,27	7,54	7,18
	Bagging	-	6,65	6,44	6,30	6,64	6,49	**6,22**
	NNTree	1	7,94	7,87	7,66	8,13	8,01	**7,39**
		2	7,90	7,83	7,61	8,13	8,00	7,42
		3	7,88	7,81	7,59	8,13	8,00	7,55
	RISKSEG	1	8,03	7,66	7,56	8,00	7,96	7,72
		2	7,97	7,55	7,57	7,99	7,89	7,76
		3	7,83	**7,54**	7,55	7,99	7,87	7,68
	SegTree	1	7,52	7,90	7,68	7,78	8,03	7,90
		2	7,49	7,88	7,66	7,76	8,08	7,92
		3	**7,48**	7,90	7,65	7,76	8,08	7,87
5.000	Simples	-	7,46	6,76	**6,75**	7,54	7,22	6,88
	Boosting	-	7,21	5,94	5,94	7,10	6,05	**5,77**
	Bagging	-	6,71	**5,57**	5,62	6,46	5,67	5,72
	NNTree	1	7,18	6,72	6,68	7,31	7,19	6,75
		2	7,10	6,71	6,66	7,27	7,16	6,75
		3	7,05	6,70	**6,65**	7,25	7,14	6,68
	RISKSEG	1	7,21	6,78	6,79	7,15	6,91	6,79
		2	7,09	**6,75**	6,85	7,01	6,87	6,87
		3	7,10	6,77	6,91	7,02	6,88	6,94
	SegTree	1	7,30	**6,72**	6,78	7,24	7,11	6,82
		2	7,22	6,77	6,80	7,23	7,10	6,79
		3	7,17	6,77	6,80	7,17	7,10	6,79

Pode-se observar na Tabela 5.2, que não houve uma configuração predominante para Rede Neural. De maneira geral, *Backpropagation* apresentou menores resultados que

Levenberg-Marquardt) e os resultados dos métodos de combinação não foram expressivamente maiores do que os dos classificadores Simples. Mais adiante estes resultados serão comparados para verificar se existe melhora significativa.

Tabela 5.3 – Erro médio para seleção do melhor nível da Regressão Linear - Base (1).

Regressão Linear			
Método	Tamanho	Nível	Média
NNTree	1.000	1	18,88
		2	12,55
		3	**12,08**
	3.000	1	18,15
		2	12,29
		3	**11,72**
	5.000	1	15,99
		2	12,18
		3	**11,59**
RISKSEG	1.000	1	11,63
		2	**11,11**
		3	11,41
	3.000	1	9,10
		2	8,55
		3	**8,53**
	5.000	1	8,66
		2	8,13
		3	**8,02**
SegTree	1.000	1	**12,04**
		2	12,72
		3	12,56
	3.000	1	**11,20**
		2	11,23
		3	11,24
	5.000	1	11,07
		2	10,95
		3	**10,85**

Tabela 5.4 – Erro médio para seleção do melhor nível da Regressão Logística - Base (1).

Regressão Logística			
Método	Tamanho	Nível	Média
NNTree	1.000	1	11,58
		2	11,46
		3	**11,45**
	3.000	1	11,12
		2	11,09
		3	**11,08**
	5.000	1	10,89
		2	10,87
		3	**10,86**
RISKSEG	1.000	1	10,24
		2	**9,89**
		3	9,99
	3.000	1	8,34
		2	7,52
		3	**7,42**
	5.000	1	7,82
		2	7,22
		3	**7,03**
SegTree	1.000	1	**11,68**
		2	11,73
		3	11,71
	3.000	1	11,13
		2	11,08
		3	**11,03**
	5.000	1	10,71
		2	10,51
		3	**10,41**

Conforme as Tabelas 5.3 e 5.4, todas as segmentações para as Regressões com o método *NNTree* apresentaram melhores resultados com três níveis máximos de profundidade. Para as outras duas segmentações não houve um nível predominante.

A Tabela 5.5 apresenta os resultados obtidos pelos experimentos da primeira base estudada. Nesta tabela tem-se que a taxa de erro da Regressão Logística Simples manteve-se muito parecida para os três tamanhos amostrais. O mesmo acontece com a Regressão Linear que não teve sua taxa de erro muito alterada. Porém, as Redes Neurais apresentaram grande redução de taxa de erro à medida que o tamanho da amostra aumentou. Notadamente, as Redes Neurais obtiveram melhores resultados que as demais técnicas e se mostraram mais expostas aos efeitos

do tamanho amostral que as demais técnicas. Este comportamento também é observado e discutido em artigos que estudam os efeitos de tamanho de amostra no desempenho de Redes Neurais [Kia03]. Outro fator em que as Redes Neurais se mostraram diferentes das outras técnicas foi na aplicação de *Bagging* e *Boosting*, pois tiveram seus desempenhos melhorados com a aplicação desses métodos, independente dos tamanhos amostrais, enquanto as Regressões Logísticas e Lineares nem sempre apresentaram melhoras significativas para estes dois métodos.

Tabela 5.5 – Médias das taxas (%) de erros e intervalos de confiança para a Base (1).

Registros	Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
1.000	**Redes Neurais**	10,05 ±0,73	**7,89 ±0,58***	**9,17 ±0,67***	**9,8 ±0,71**	**10,01 ±0,69**	**9,87 ±0,74**
	Reg. Linear	18,31 ±1,8	**17,33 ±1,67***	**17,17 ±1,69***	**12,08 ±0,87***	**11,11 ±0,76***	**12,04 ±0,77***
	Reg. Logística	11,96 ±0,74	**11,71 ±0,69**	**11,87 ±0,71**	**11,45 ±0,67***	**9,89 ±0,68***	**11,68 ±0,67**
3.000	**Redes Neurais**	7,38 ±0,34	**6,22 ±0,38***	**6,95 ±0,27***	7,39 ±0,37	7,54 ±0,42	7,48 ±0,37
	Reg. Linear	16,93 ±1,21	**16,48 ±1**	**16,3 ±0,93***	**11,72 ±0,47***	**8,53 ±0,39***	**11,2 ±0,43***
	Reg. Logística	11,28 ±0,42	**11,14 ±0,41**	**11,19 ±0,41**	**11,08 ±0,45***	**7,42 ±0,35***	**11,03 ±0,41**
5.000	**Redes Neurais**	6,75 ±0,31	**5,57 ±0,27***	**5,77 ±0,41***	**6,65 ±0,32**	6,75 ±0,25	**6,72 ±0,32**
	Reg. Linear	18,44 ±1,68	**18,23 ±1,6**	**18,25 ±1,64**	**11,59 ±0,28***	**8,02 ±0,25***	**10,85 ±0,32***
	Reg. Logística	11,11 ±0,31	**10,98 ±0,27***	**11,01 ±0,28***	**10,86 ±0,3***	**7,03 ±0,24***	**10,41 ±0,35***

A Tabela 5.5 mostra que os experimentos com Redes Neurais em todos os tamanhos de bases apresentaram os métodos *Bagging* e *Boosting* como os que obtiveram os melhores e mais significativos resultados, quando comparados com o seu respectivo classificador de controle. Ainda analisando os resultados com Redes Neurais, nenhuma técnica de segmentação conseguiu resultados significativamente melhores que o classificador Simples. O método *Bagging* foi significativamente melhor quando comparado com o *Boosting* para os tamanhos de amostras 1.000 e 3.000 registros.

Na Regressão Linear, quase todos os resultados com métodos de combinação melhoram significativamente em relação aos seus respectivos classificadores Simples, destaque especial para os métodos de combinação baseados em segmentação, que melhoraram significativamente em todos os tamanhos de base (Tabela 5.5). Quando é feita uma análise para determinar qual o melhor método de combinação, o *RISKSEG* foi significativamente melhor do que todos os demais métodos experimentados com Regressão Linear, considerando-se todos os tamanhos de bases. O método *NNTree*, adaptado neste trabalho para trabalhar com Regressão Linear e Logística, não foi melhor do que o *RISKSEG*, mas apresentou bons e significativos resultados, apesar do fato de não ter sido concebido para trabalhar com estas Regressões [Pra08].

Ainda analisando a Tabela 5.5, a Regressão Logística apresentou resultados melhores para 1.000 e 3.000 registros, apenas com os métodos de segmentação *RISKSEG* e *SegTree*. Nos experimentos com 5.000 registros todos os métodos de combinação apresentaram resultados

significativamente melhores que os respectivos classificadores Simples. Na análise entre os resultados dos melhores métodos de combinação, destaque especial para o *RISKSEG* que, assim como na Regressão Linear, também apresentou resultados significativamente melhores do que os demais métodos.

Para esta base de dados verificou-se que classificadores fortes como Redes Neurais não foram impactados pela segmentação, porém, estes métodos auxiliaram as demais Regressões a atingir resultados parecidos com aqueles obtidos com as Redes Neurais. A aplicação do *RISKSEG* na Regressão Linear, que é uma técnica notoriamente mais fraca [Kia03], aproximou os resultados quando comparados com Redes Neurais. Por exemplo, nos experimentos com 5.000 registros o classificador Simples da Rede Neural (6,75%) está muito distante da Regressão Linear (18,44%), mas a aplicação do *RISKSEG* diminuiu esta diferença para 8,02%.

5.3.2 – RESULTADO: BASE (2) – NÃO LINEAR

Observa-se na Tabela 5.6, que o algoritmo *Backpropagation* apresentou menores erros médios que *Levenberg-Marquardt*, este último sendo escolhido apenas três vezes. Quanto à quantidade de neurônios na camada intermediária da Rede Neural, para a amostra com 1.000 registros, três neurônios foi a melhor configuração para todos os métodos, exceto para o *Bagging*.

Tabela 5.6 – Erro médio para seleção da melhor configuração da Rede Neural - Base (2).

Registros	Método	Nível	*Backpropagation*			*Levenberg-Marquardt*		
			3 neur.	10 neur.	20 neur.	3 neur.	10 neur.	20 neur.
1.000	Simples	-	**10,15**	10,83	10,73	10,64	11,32	11,28
	Boosting	-	9,89	10,53	19,56	**9,84**	20,01	20,19
	Bagging	-	9,31	9,84	**8,72**	8,85	8,77	8,81
	NNTree	1	9,83	10,77	10,27	10,32	11,01	10,86
		2	9,67	10,63	10,11	10,21	10,90	10,82
		3	**9,58**	10,55	10,03	10,16	10,84	10,80
	RISKSEG	1	10,37	10,89	10,73	10,83	11,72	15,16
		2	10,52	11,04	10,68	11,03	11,72	18,93
		3	**10,36**	11,15	11,09	11,35	11,43	17,23
	SegTree	1	10,68	10,93	10,69	10,75	11,17	11,28
		2	10,68	11,05	10,80	10,61	11,24	11,12
		3	**10,61**	11,05	10,80	10,77	11,28	11,11
3.000	Simples	-	9,18	**7,77**	8,23	9,04	10,29	9,22
	Boosting	-	9,36	**7,26**	7,68	8,96	8,86	17,91
	Bagging	-	8,84	**7,11**	7,20	8,29	7,29	7,43
	NNTree	1	8,90	7,85	8,36	8,84	9,84	9,14
		2	8,81	7,82	8,31	8,79	9,67	9,04
		3	8,76	**7,81**	8,28	8,76	9,57	8,98
	RISKSEG	1	9,01	**8,22**	8,50	8,92	10,09	9,11
		2	8,72	8,30	8,56	8,98	10,10	9,08
		3	8,62	8,30	8,59	8,96	10,10	9,04
	SegTree	1	9,18	**7,76**	8,35	9,09	10,13	9,39
		2	8,98	7,82	8,35	8,99	10,12	9,39
		3	8,85	7,82	8,35	8,98	10,16	9,30
5.000	Simples	-	8,78	**6,93**	7,30	8,58	8,53	8,60
	Boosting	-	9,06	6,66	6,60	8,47	7,18	**6,44**
	Bagging	-	8,55	**6,48**	6,66	8,26	6,72	6,73
	NNTree	1	8,53	**6,93**	7,30	8,42	8,48	8,44
		2	8,48	6,93	7,30	8,36	8,43	8,37
		3	8,45	6,93	7,29	8,32	8,41	8,34
	RISKSEG	1	8,33	**7,30**	7,46	8,29	8,65	7,61
		2	7,75	7,39	7,54	7,94	8,58	7,69
		3	7,44	7,39	7,54	7,95	8,61	7,69
	SegTree	1	8,62	**6,99**	7,34	8,54	8,34	9,14
		2	8,29	6,99	7,36	8,43	8,38	9,15
		3	8,17	6,99	7,36	8,45	8,38	9,15

Ainda na Tabela 5.6, pode-se ver que as amostras com 3.000 e 5.000 registros, para todos os experimentos o menor erro foi utilizando 10 neurônios, exceto para o *Boosting*, com 5.000

registros que precisou de 20 neurônios para obter seu melhor resultado. O nível de profundidade das segmentações variou apenas entre o nível 1 e o nível 3.

Conforme as Tabelas 5.7 e 5.8, quase todas as segmentações para as Regressões precisaram de três níveis para atingir o menor erro de configuração, com exceção de algumas segmentações com 1.000 registros do *RISKSEG* e *SegTree*. Pode-se observar ainda que o método *NNTree* precisa de pelo menos dois níveis de profundidade para atingir um valor médio de erro abaixo de 13%, em todos os tamanhos de amostra.

Tabela 5.7 – Erro médio para seleção do melhor nível da Regressão Linear - Base (2).

Regressão Linear			
Método	**Tamanho**	**Nível**	**Média**
NNTree	**1.000**	**1**	20,61
		2	12,30
		3	**11,84**
	3.000	**1**	21,07
		2	12,58
		3	**11,99**
	5.000	**1**	20,80
		2	11,93
		3	**11,38**
RISKSEG	**1.000**	**1**	13,72
		2	13,19
		3	**13,17**
	3.000	**1**	13,88
		2	12,72
		3	**12,28**
	5.000	**1**	13,35
		2	12,13
		3	**11,51**
SegTree	**1.000**	**1**	14,40
		2	**13,92**
		3	13,96
	3.000	**1**	13,84
		2	13,32
		3	**12,88**
	5.000	**1**	13,45
		2	12,61
		3	**12,20**

Tabela 5.8 – Erro médio para seleção do melhor nível da Regressão Logística - Base (2).

Regressão Logística			
Método	**Tamanho**	**Nível**	**Média**
NNTree	**1.000**	**1**	11,82
		2	11,41
		3	**11,18**
	3.000	**1**	11,42
		2	11,19
		3	**10,91**
	5.000	**1**	10,96
		2	10,67
		3	**10,42**
RISKSEG	**1.000**	**1**	13,37
		2	**12,91**
		3	13,07
	3.000	**1**	13,62
		2	12,81
		3	**12,25**
	5.000	**1**	13,16
		2	11,78
		3	**11,26**
SegTree	**1.000**	**1**	14,20
		2	**14,07**
		3	14,07
	3.000	**1**	13,84
		2	13,34
		3	**13,00**
	5.000	**1**	13,26
		2	12,42
		3	**12,18**

Nesta base de dados, observa-se, através da Tabela 5.9, a mesma situação encontrada na base de dados anterior, em relação aos classificadores Simples, onde as Redes Neurais foram as que mais sofreram os efeitos de tamanho amostral. As taxas de erro de Regressão Logística e Linear mantêm-se mais similares nos diferentes tamanhos amostrais. Os métodos *Bagging* e *Boosting* apresentam melhorias significativas em quase todas as amostras com a utilização de

Redes Neurais, mas nenhum resultado melhor e significativo nas Regressões. Em contrapartida, as segmentações apresentaram melhoria em todos os tamanhos amostrais, exceto a *SegTree* com 1.000 registros.

Tabela 5.9 – Médias das taxas (%) de erros e intervalos de confiança para a Base (2).

Registros	Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
1.000	Redes Neurais	10,15 ±0,7	**8,72 ±0,73***	**9,84 ±0,7**	**9,58 ±0,69**	10,36 ±0,63	10,61 ±0,61
	Reg. Linear	17,52 ±1,58	**17,36 ±1,43**	**17,4 ±1,43**	**11,84 ±0,76***	**13,17 ±0,88***	**13,92 ±1,24***
	Reg. Logística	14,61 ±0,8	**14,53 ±0,75**	**14,56 ±0,77**	**11,18 ±0,91***	**12,91 ±0,92***	**14,07 ±0,81**
3.000	Redes Neurais	7,77 ±0,3	**7,11 ±0,28***	**7,26 ±0,34***	7,81 ±0,34	*8,22 ±0,37*	**7,76 ±0,31**
	Reg. Linear	16,24 ±0,88	16,31 ±0,79	16,26 ±0,79	**11,99 ±0,43***	**12,28 ±0,45***	**12,88 ±0,53***
	Reg. Logística	14,79 ±0,46	**14,72 ±0,43**	**14,75 ±0,44**	**10,91 ±0,4***	**12,25 ±0,5***	**13 ±0,59***
5.000	Redes Neurais	6,93 ±0,27	**6,48 ±0,27***	**6,44 ±0,45***	6,93 ±0,27	*7,3 ±0,32*	*6,99 ±0,27*
	Reg. Linear	15,87 ±0,59	**15,83 ±0,63**	**15,81 ±0,61**	**11,38 ±0,39***	**11,51 ±0,53***	**12,2 ±0,56***
	Reg. Logística	14,44 ±0,38	**14,4 ±0,36**	**14,38 ±0,37**	**10,42 ±0,37***	**11,26 ±0,49***	**12,18 ±0,4***

Para todos os métodos e tamanho de amostra, notadamente, as Redes Neurais obtiveram melhores resultados e se mostraram mais expostas aos efeitos do tamanho amostral do que as demais técnicas, este último efeito é observado na maior variação do erro entre os diferentes tamanhos de amostra. Este comportamento também é observado e discutido em artigos que estudam os efeitos de tamanho de amostra no desempenho de Redes Neurais [Kia03]. Ainda analisando os resultados com Redes Neurais, nenhuma técnica de segmentação conseguiu resultados significativamente melhores que o classificador Simples. O método *Bagging* foi significativamente melhor quando comparado com o *Boosting* apenas para as amostras de 1.000 registros, para os demais tamanhos não houve diferença significativa.

Na Regressão Linear, todos os resultados com métodos de segmentação (*NNTree*, *RISKSEG* e *SegTree*) melhoraram significativamente em relação aos seus respectivos classificadores Simples. Quando é feita uma análise para determinar qual o melhor método de combinação, o *NNTree* foi significativamente melhor do que todos os demais métodos apenas com tamanho de amostra igual a 1.000 registros.

Também para a Regressão Logística, todos os métodos de segmentação apresentaram resultados melhores e significativos para os três tamanhos de amostra, exceto o *SegTree* com 1.000 registros. Destaque para o método de segmentação *NNTree*, que obteve resultados menores e significativamente diferentes quando comparados com todos os demais métodos de combinação.

Para esta base de dados, verificou-se que classificadores fortes como Redes Neurais não foram impactados pela segmentação, porém estes métodos auxiliaram as demais Regressões a

atingirem resultados mais parecidos com os obtidos usando as Redes Neurais. O *RISKSEG* permitiu ganhos em desempenho em todos os tamanhos amostrais para as técnicas de Regressão Logística e Linear. O método *NNTree* adaptado para utilizar Regressão Linear e Logística, foi melhor em quase todas as situações, quando comparado com o *RISKSEG*.

5.3.3 – RESULTADO: BASE (3) – EFEITOS ADITIVOS (LINEAR)

Pode-se observar na Tabela 5.10 que, no geral, o algoritmo *Backpropagation* apresentou menores erros médios que *Levenberg-Marquardt*, este último sendo escolhido em apenas duas configurações. Quanto à quantidade de neurônios na camada intermediária da Rede Neural, a maioria dos experimentos utilizou apenas três neurônios.

Tabela 5.10 – Erro médio para seleção da melhor configuração da Rede Neural - Base (3).

Registros	Técnica	Nível	*Backpropagation*			*Levenberg-Marquardt*		
			3 neur.	10 neur.	20 neur.	3 neur.	10 neur.	20 neur.
1.000	Simples	-	**7,29**	8,64	8,83	7,63	9,31	8,74
	Boosting	-	7,67	**7,36**	7,95	7,67	7,36	8,03
	Bagging	-	**6,83**	6,91	7,31	6,83	7,32	7,13
	NNTree	1	7,37	8,27	8,53	7,64	8,68	8,45
		2	7,37	8,16	8,36	7,60	8,53	8,43
		3	**7,36**	8,09	8,26	7,57	8,44	8,51
	RISKSEG	1	9,01	9,01	8,97	**7,77**	9,00	8,75
		2	8,68	8,72	9,08	8,05	8,62	8,82
		3	8,74	8,75	8,99	8,20	8,65	8,89
	SegTree	1	7,41	8,80	9,23	7,64	9,25	9,31
		2	**7,37**	8,71	9,29	7,77	9,33	9,35
		3	7,43	8,71	9,35	7,81	9,39	9,35
3.000	Simples	-	**6,60**	6,92	7,10	6,93	9,00	7,04
	Boosting	-	**6,47**	6,52	6,52	6,57	8,54	8,54
	Bagging	-	**6,38**	6,68	7,40	6,42	7,05	7,49
	NNTree	1	**6,76**	6,97	7,24	7,07	8,38	7,20
		2	6,80	6,99	7,21	7,10	8,07	7,10
		3	6,83	7,01	7,18	7,12	7,88	7,11
	RISKSEG	1	**6,69**	6,95	6,91	6,98	7,02	7,12
		2	6,89	7,12	6,92	7,11	7,26	6,95
		3	6,97	7,15	6,90	7,11	7,29	6,99
	SegTree	1	**6,74**	6,92	7,10	7,12	8,92	7,04
		2	6,74	6,94	7,10	7,12	8,94	7,04
		3	6,74	6,94	7,10	7,12	8,93	7,04
5.000	Simples	-	**6,66**	6,72	6,78	6,68	8,76	6,92
	Boosting	-	**6,56**	6,61	6,61	6,62	7,85	7,85
	Bagging	-	**6,51**	6,59	7,40	6,57	6,59	7,23
	NNTree	1	**6,73**	6,83	6,96	6,91	8,35	7,09
		2	6,74	6,86	6,97	6,92	8,07	6,98
		3	6,75	6,88	6,98	6,92	7,89	7,09
	RISKSEG	1	**6,77**	6,91	6,98	6,91	6,84	6,79
		2	6,80	6,98	7,00	7,03	6,90	6,83
		3	6,82	7,02	7,01	7,02	6,90	6,82
	SegTree	1	6,72	6,78	6,86	6,83	8,55	6,77
		2	**6,70**	6,80	6,84	6,83	8,57	6,72
		3	6,70	6,80	6,84	6,83	8,57	6,72

Na Tabela 5.10 observa-se que, nos métodos de segmentação, um ou dois níveis de profundidade nas segmentações foram suficientes para todos os experimentos, exceto para o *NNTree* que teve uma média ligeiramente melhor (7,37% para 7,36%) nas amostras de 1.000 registros.

Conforme as Tabelas 5.11 e 5.12, quase todas as segmentações para as Regressões precisaram de apenas um nível para atingir o menor erro de configuração de seu respectivo método, as exceções foram as segmentações *NNTree* com 1.000 e 5.000 registros na Regressão Linear.

Tabela 5.11 – Erro médio para seleção do melhor nível da Regressão Linear - Base (3).

Regressão Linear			
Método	Tamanho	Nível	Média
NNTree	1.000	1	7,95
		2	**7,87**
		3	7,89
	3.000	**1**	**7,20**
		2	7,28
		3	7,25
	5.000	1	7,40
		2	7,42
		3	**7,39**
RISKSEG	1.000	**1**	**7,33**
		2	7,49
		3	7,79
	3.000	**1**	**6,64**
		2	6,83
		3	6,92
	5.000	**1**	**6,69**
		2	6,81
		3	6,85
SegTree	1.000	**1**	**6,93**
		2	7,11
		3	7,47
	3.000	**1**	**6,47**
		2	6,68
		3	6,92
	5.000	**1**	**6,74**
		2	6,94
		3	6,98

Tabela 5.12 – Erro médio para seleção do melhor nível da Regressão Logística - Base (3).

Regressão Logística			
Método	Tamanho	Nível	Média
NNTree	1.000	**1**	**6,60**
		2	6,70
		3	6,70
	3.000	**1**	**6,39**
		2	6,40
		3	6,40
	5.000	**1**	**6,56**
		2	6,57
		3	6,57
RISKSEG	1.000	**1**	**6,45**
		2	6,63
		3	6,83
	3.000	**1**	**6,42**
		2	6,45
		3	6,44
	5.000	**1**	**6,55**
		2	6,55
		3	6,58
SegTree	1.000	**1**	**6,52**
		2	6,53
		3	6,57
	3.000	**1**	**6,34**
		2	6,36
		3	6,36
	5.000	**1**	**6,53**
		2	6,55
		3	6,56

Essas bases de dados foram construídas sem efeitos de interações e sem elementos de não-linearidade. Portanto, era esperado que as segmentações causassem menos efeitos positivos no desempenho dos classificadores, uma vez que a complexidade se limita a ruídos e a efeitos lineares aditivos na explicação da variável resposta.

Na Tabela 5.13, pode-se observar que diversos experimentos apresentaram resultados inferiores ou não significativamente melhores aos experimentos base, principalmente quando aplicados nos métodos de segmentação e *Boosting*. Estes dois métodos visam capturar estruturas mais complexas. Uma vez que não há a presença de tais estruturas, os métodos podem, inclusive, causar perda de desempenho de classificação. Neste tipo de base de dados, as segmentações podem gerar subgrupos não relevantes, pois a mesma estrutura linear se mantém nos segmentos. Através destes segmentos, têm-se indutores treinados com número inferior de registros e, por conseguinte, pode-se esperar perda de generalização.

Conforme mostra os resultados da Tabela 5.13, nos experimentos com Redes Neurais apenas o *Bagging* melhorou significativamente, porém esta melhora aconteceu para todos os tamanhos de bases. Nos experimentos com Regressão Linear os métodos *RISKSEG* e *SegTree* foram os que obtiveram melhores resultados. Ainda na Regressão Linear, o *RISKSEG* só não conseguiu melhorar significativamente em relação ao classificador Simples na amostra de 1.000 registros. O *Boosting* só conseguiu melhora significativa na amostra de 3.000 registros. Na Regressão Logística houve melhora significativa, em relação ao método de controle, apenas nos experimentos com 3.000 registros utilizando os métodos *Boosting* e *Bagging*.

Tabela 5.13 – Médias das taxas (%) de erros e intervalos de confiança para a Base (3).

Registros	Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
1.000	Redes Neurais	7,29 ±0,7	**6,83 ±0,58***	7,36 ±0,63	7,36 ±0,67	7,77 ±0,6	7,37 ±0,58
	Reg. Linear	7,95 ±0,77	**7,93 ±0,83**	8,03 ±0,86	**7,87 ±0,71**	**7,33 ±0,5**	**6,93 ±0,58***
	Reg. Logística	6,43 ±0,53	**6,33 ±0,44**	**6,31 ±0,48**	6,6 ±0,64	6,45 ±0,56	6,52 ±0,5
3.000	Redes Neurais	6,6 ±0,29	**6,38 ±0,29***	**6,47 ±0,29**	*6,76 ±0,3*	6,69 ±0,3	*6,74 ±0,31*
	Reg. Linear	7,2 ±0,45	**7,09 ±0,46**	**7,04 ±0,44***	7,2 ±0,45	**6,64 ±0,31***	**6,47 ±0,24***
	Reg. Logística	6,33 ±0,28	**6,22 ±0,28***	**6,23 ±0,27***	6,39 ±0,28	6,42 ±0,3	6,34 ±0,3
5.000	Redes Neurais	6,66 ±0,33	**6,51 ±0,29***	**6,56 ±0,3**	6,73 ±0,34	6,77 ±0,29	6,7 ±0,33
	Reg. Linear	7,4 ±0,37	**7,35 ±0,36**	**7,35 ±0,37**	7,39 ±0,35	**6,69 ±0,31***	**6,74 ±0,3***
	Reg. Logística	6,52 ±0,29	**6,47 ±0,28**	**6,5 ±0,28**	6,56 ±0,27	6,55 ±0,26	6,53 ±0,29

Nesta base de dados, o *NNTree* não apresentou melhoras significativas em nenhuma das amostras e técnicas utilizadas. Apesar de não se esperar que o método *RISKSEG* apresentasse melhoria para este tipo de base de dados, este ainda conseguiu melhorar significativamente os erros em relação ao classificador Simples em dois tamanhos de amostras para a Regressão Linear. O *SegTree*, com Regressão Linear, apresentou melhora significativa em relação ao classificador Simples nos três tamanhos de amostras.

5.3.4 – RESULTADO: BASE (4) – EFEITOS QUADRÁTICOS

Pode-se observar, na Tabela 5.14, que o algoritmo *Backpropagation* apresentou menores erros médios que *Levenberg-Marquardt* em quase todos os métodos, este último sendo escolhido apenas em três configurações. Quanto à quantidade de neurônios na camada intermediária da Rede Neural, para todos os métodos e tamanhos de amostras, três neurônios foram suficientes para gerar os menores erros médios, exceto no *Boosting* com 3.000 registros e no *RISKSEG* com 1.000 registros.

Tabela 5.14 – Erro médio para seleção da melhor configuração da Rede Neural - Base (4).

Registros	Método	Nível	*Backpropagation*			*Levenberg-Marquardt*		
			3 neur.	10 neur.	20 neur.	3 neur.	10 neur.	20 neur.
1.000	Simples	-	**7,24**	8,09	8,28	8,04	9,15	9,12
	Boosting	-	6,70	7,39	8,17	**6,56**	6,64	8,18
	Bagging	-	**6,31**	7,20	7,63	6,39	6,49	7,61
	NNTree	1	7,51	8,36	8,30	8,24	8,72	8,64
		2	7,43	8,30	8,23	8,22	8,56	8,40
		3	**7,38**	8,26	8,18	8,21	8,47	8,26
	RISKSEG	1	7,52	8,60	8,77	7,72	**7,41**	9,24
		2	8,09	9,11	8,73	8,03	7,63	9,37
		3	8,24	9,25	8,95	8,25	7,85	9,40
	SegTree	1	**7,35**	8,37	8,37	8,23	9,47	9,21
		2	7,44	8,37	8,37	8,19	9,44	9,16
		3	7,37	8,37	8,37	8,19	9,44	9,16
3.000	Simples	-	**6,56**	6,84	7,34	6,83	9,27	7,18
	Boosting	-	6,56	**6,51**	6,89	6,55	6,77	6,92
	Bagging	-	**6,51**	6,64	6,58	6,52	6,68	6,53
	NNTree	1	6,51	6,85	7,23	6,81	8,54	7,31
		2	6,47	6,83	7,13	6,79	8,16	7,30
		3	**6,45**	6,82	7,06	6,77	7,94	7,06
	RISKSEG	1	**6,82**	7,10	7,81	6,98	7,19	7,71
		2	7,08	7,29	7,89	7,04	7,44	7,78
		3	7,14	7,33	7,92	7,06	7,44	7,77
	SegTree	1	6,61	6,98	7,50	6,89	9,37	7,33
		2	**6,60**	7,01	7,52	6,99	9,25	7,34
		3	6,60	7,01	7,52	7,03	9,25	7,34
5.000	Simples	-	6,35	6,41	6,69	**6,34**	6,70	6,77
	Boosting	-	**6,18**	6,29	6,34	6,22	7,14	7,13
	Bagging	-	**6,19**	6,31	6,47	6,22	6,28	6,38
	NNTree	1	6,33	6,40	6,74	6,37	6,48	6,80
		2	6,29	6,37	6,72	6,36	6,44	6,82
		3	**6,27**	6,35	6,71	6,36	6,35	6,78
	RISKSEG	1	**6,51**	6,67	6,70	6,55	6,69	6,69
		2	6,55	6,73	6,79	6,60	6,68	6,56
		3	6,58	6,76	6,78	6,57	6,71	6,60
	SegTree	1	**6,39**	6,49	6,73	6,50	7,82	6,84
		2	6,40	6,53	6,74	6,51	7,82	6,88
		3	6,40	6,53	6,74	6,51	7,82	6,88

Conforme as Tabelas 5.15 e 5.16, quase todas as segmentações para as Regressões e para o método *NNTree* precisaram de três níveis de profundidade para atingir o menor erro de configuração. No *RISKSEG* e no *SegTree* não houve uma predominância de nenhum número máximo de níveis. Atenção apenas para o *NNTree*, que precisou de pelo menos dois níveis para ter erro menor que 10% na Regressão Linear em todos os tamanhos de amostra.

Tabela 5.15 – Erro médio para seleção do melhor nível da Regressão Linear - Base (4).

Regressão Linear			
Método	Tamanho	Nível	Média
NNTree	1.000	1	11,99
		2	8,44
		3	**8,40**
	3.000	1	11,90
		2	7,88
		3	**7,69**
	5.000	1	12,19
		2	7,80
		3	**7,57**
RISKSEG	1.000	1	**8,48**
		2	8,57
		3	8,65
	3.000	1	**7,47**
		2	7,59
		3	7,63
	5.000	1	7,41
		2	**7,23**
		3	7,27
SegTree	1.000	1	**8,61**
		2	8,80
		3	8,72
	3.000	1	8,23
		2	**8,21**
		3	8,26
	5.000	1	7,68
		2	7,59
		3	**7,51**

Tabela 5.16 – Erro médio para seleção do melhor nível da Regressão Logística - Base (4).

Regressão Logística			
Método	Tamanho	Nível	Média
NNTree	1.000	1	**7,83**
		2	7,83
		3	7,90
	3.000	1	7,71
		2	7,53
		3	**7,49**
	5.000	1	7,42
		2	7,28
		3	**7,24**
RISKSEG	1.000	1	8,15
		2	**8,12**
		3	8,15
	3.000	1	7,55
		2	7,33
		3	**7,21**
	5.000	1	7,06
		2	6,95
		3	**6,89**
SegTree	1.000	1	8,45
		2	8,21
		3	**8,11**
	3.000	1	8,39
		2	**8,38**
		3	8,39
	5.000	1	7,69
		2	7,57
		3	**7,53**

Nesta base de dados, observa-se, através da Tabela 5.17, que apenas os métodos *Boosting* e *Bagging* conseguiram melhoria significativa em relação ao classificador de controle nas amostras com 1.000 e 5.000 registros. Estes dois métodos não obtiveram resultados para as Regressões Estatísticas em nenhum tamanho de base. Entretanto, os métodos de segmentação aplicados às Regressões conseguiram melhorias em quase todos os tamanhos de amostras. Isto demonstra que, para este tipo de formação de base, de maneira geral, é relevante utilizar *Bagging* e *Boosting* para melhorar o desempenho de classificadores baseados em Redes Neurais e métodos de segmentação quando se está utilizando Regressão Linear ou Logística.

Tabela 5.17 – Médias das taxas (%) de erros e intervalos de confiança para a Base (4).

Registros	Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
1.000	Redes Neurais	7,24 ±0,47	**6,31 ±0,65***	**6,56 ±0,6***	7,38 ±0,51	7,41 ±0,52	7,35 ±0,48
	Reg. Linear	12,01 ±0,73	**11,92 ±0,72**	**12 ±0,72**	**8,4 ±0,57***	**8,48 ±0,69***	**8,61 ±0,65***
	Reg. Logística	8,36 ±0,69	8,39 ±0,71	8,37 ±0,68	**7,83 ±0,7***	**8,12 ±0,48**	**8,11 ±0,69**
3.000	Redes Neurais	6,56 ±0,3	**6,51 ±0,33**	**6,51 ±0,34**	**6,45 ±0,31**	*6,82 ±0,32*	6,6 ±0,3
	Reg. Linear	11,9 ±0,94	**11,86 ±0,91**	**11,85 ±0,94**	**7,69 ±0,41***	**7,47 ±0,38***	**8,21 ±0,42***
	Reg. Logística	8,59 ±0,35	**8,48 ±0,33***	**8,51 ±0,34**	**7,49 ±0,31***	**7,21 ±0,33***	**8,38 ±0,4***
5.000	Redes Neurais	6,34 ±0,2	**6,19 ±0,22***	**6,18 ±0,21***	**6,27 ±0,27**	*6,51 ±0,21*	6,39 ±0,25
	Reg. Linear	11,88 ±0,68	11,92 ±0,66	**11,86 ±0,66**	**7,57 ±0,3***	**7,23 ±0,24***	**7,51 ±0,32***
	Reg. Logística	8,04 ±0,23	8,04 ±0,23	8,06 ±0,24	**7,24 ±0,27***	**6,89 ±0,27***	**7,53 ±0,28***

Notadamente, as Redes Neurais obtiveram melhores resultados e se mostraram mais expostas aos efeitos do tamanho amostral que as demais técnicas, com uma melhora a cada incremento amostral. Este comportamento também é observado e discutido em artigos que estudam os efeitos de tamanho de amostra no desempenho de Redes Neurais [Kia03].

Quando é feita a análise para saber se existe diferença significativa entre os classificadores que foram superiores em relação ao controle, observa-se para os experimentos com 1.000 registros e Redes Neurais que não houve diferença significativa entre o *Bagging* e *Boosting*. Na Regressão Linear também não houve diferença entre as técnicas de segmentação (*NNTree*, *RISKSEG* e *SegTree*). Para os experimentos com 3.000 registros rodando Regressões, o método *RISKSEG* foi significativamente melhor do que o *SegTree* e não teve diferença quando comparado com o *NNTree*. Nos experimentos com 5.000 registros, não houve diferença significativa entre o *Bagging* e o *Boosting* para a Rede Neural e, nas Regressões, o *RISKSEG* foi significativamente melhor que os demais métodos de segmentação *NNTree* e *SegTree*.

5.4 – RESUMO GERAL DOS RESULTADOS

As simulações realizadas neste capítulo permitiram entender o comportamento do método *RISKSEG*, através de diversas características encontradas nas bases de dados, como tamanho amostral e estrutura de dependência entre as variáveis. Foram empregadas também diferentes técnicas de predição para que se estudasse a influência do indutor nos resultados encontrados, gerando assim uma gama de combinações possíveis entre características de dados e técnicas preditoras. Além do estudo comportamental do *RISKSEG*, foi possível também compará-lo com outros métodos utilizados em Mineração de Dados, como *NNTree*, *Bagging* e *Boosting*. Através das características das bases é possível observar suas diferenças e as situações em que cada um deles é recomendado.

Na Tabela 5.18 tem-se uma análise resumida do desempenho do *RISKSEG*. Esta análise é baseada em três informações para cada técnica: a quantidade de métodos que foram significativamente superiores ao classificador de controle (Simples), um indicador se o *RISKSEG* foi uma destas técnicas superiores e a sua posição em relação aos demais. Para este último caso foi feito o teste *t-student* para amostras pareadas, com nível de significância de 5%, para comparar os demais métodos com o *RISKSEG*, ou seja, se não houver uma diferença significativa entre os métodos eles terão a mesma posição (*rank*). Um asterisco é colocado ao lado do valor da posição pra indicar que ele está sozinho nesta posição.

Tabela 5.18 – Resumo dos resultados gerais das bases artificiais.

Base	Tamanho	Redes Neurais			Regressão Linear			Regressão Logística		
		Qtde	*RISKSEG*?	Posição	Qtde	*RISKSEG*?	Posição	Qtde	*RISKSEG*?	Posição
Base (1)	**1.000**	2	Não	-	5	Sim	1*	2	Sim	1*
	3.000	2	Não	-	4	Sim	1*	2	Sim	1*
	5.000	2	Não	-	3	Sim	1*	4	Sim	1*
Base (2)	**1.000**	1	Não	-	3	Sim	2	2	Sim	2*
	3.000	2	Não	-	3	Sim	1	3	Sim	2
	5.000	2	Não	-	3	Sim	1	3	Sim	2*
Base (3)	**1.000**	1	Não	-	1	Não	-	0	-	-
	3.000	1	Não	-	1	Sim	1	0	-	-
	5.000	1	Não	-	2	Sim	1	2	Não	-
Base (4)	**1.000**	2	Não	-	3	Sim	1	1	Não	-
	3.000	0	Não	-	3	Sim	1	4	Sim	1
	5.000	2	Não	-	3	Sim	1*	3	Sim	1*

Quando se observa os resultados nas quatro bases de dados (Tabela 5.18), o *RISKSEG* com Regressões Estatísticas conseguiu melhorar significativamente os erros em relação ao classificador Simples correspondente, em quase todos os diferentes tamanhos e bases diferentes. Até mesmo na Base (3), onde não se esperava melhoria com o método, ele conseguiu melhora em dois tamanhos de amostra com Regressão Linear. Porém com a técnica de Redes Neurais o *RISKSEG* nada melhorou em nenhuma das quatro bases.

Uma conclusão após observar os resultados encontrados nas quatro bases de dados é que os métodos de combinação são muito eficientes em bases onde não-linearidade está presente, sejam elas interações entre variáveis ou funções não-lineares das variáveis preditoras. Quando a não-linearidade presente é originada de fortes interações entre as variáveis preditoras e a técnica escolhida é uma Regressão Linear ou Logística, o *RISKSEG* apresenta resultados muito superiores a qualquer outro método de combinação. Este resultado pode ser observado na Tabela 5.18 na Base (1), onde o *RISKSEG* apresentou os melhores resultados não somente em relação ao classificador base, mas também em relação a todos os outros métodos.

Quando se observa os resultados nas bases de dados com efeitos quadráticos Base (4), o *RISKSEG* com Redes Neurais nada melhorou, uma vez que esta técnica já captura este tipo de não linearidade, ou seja, o *RISKSEG* não adiciona valor. Houve ganho nas Regressões Estatísticas, pois estas têm os seus desempenhos comprometidos quando estimam este tipo estrutura de dependência [Kia03]. Esta melhoria nas Regressões também foi observada utilizando o outro tipo de segmentação, por *Information Gain* (*SegTree*) e no método *NNTree*.

Os melhores resultados entre os experimentos que utilizam apenas o classificador Simples foram obtidos com as Redes Neurais, uma das técnicas mais aptas a capturar comportamentos não lineares nas bases de dados [HSW89] [Kia03]. Com Redes Neurais, para os métodos de *Boosting* e *Bagging*, conseguiu-se melhorias significativas em quase todas as diferentes bases e tamanhos utilizados nos experimentos.

Qualquer segmentação, por si só, já tende a dividir o problema em subproblemas de menor complexidade. Portanto, em alguns casos, o método proposto apresenta resultados similares aos de segmentações que utilizam *Information Gain*. Isto acontece, principalmente, quando a não-linearidade presente é originada de funções não lineares das variáveis preditoras. Porém, a busca do *RISKSEG* pelas melhores interações para aumento do desempenho da estimativa final obteve ganhos significativos em duas das três técnicas experimentadas.

As eficácias dos métodos de segmentação também são bastante impactadas pelo tamanho amostral disponível. Com os resultados obtidos nos experimentos, nota-se que as melhorias ficam mais evidentes quando o número de registros é maior. Este comportamento é condizente com o esperado, pois a natureza do método se baseia em diminuição de registros disponíveis para os treinamentos marginais, a cada passo. Quanto maior a árvore de segmentação, menores são os números de registros presentes nas folhas, o que pode comprometer a estrutura geral de classificação. Mesmo que uma dada estrutura de segmentação seja relevante, os ganhos obtidos podem ser prejudicados pelo baixo número de exemplos que cada subclassificador tem à sua disposição para o treinamento. Quando a mesma estrutura relevante é aplicada a uma amostra maior, estes problemas passam a não existir e o verdadeiro ganho da estrutura passa a ficar mais evidenciado.

Na Base (2), além de algumas funções não lineares presentes, como a função cosseno, existe também uma interação entre duas variáveis preditoras. Os resultados dos experimentos para esta base de dados mostraram que o *RISKSEG* foi superior a todos os classificadores de controle, ou seja, a presença de interações reforçou a justificativa de se utilizar este método para obtenção de melhoria de desempenho.

Um ambiente de simulação com o controle das características pode também ser útil quando se realiza experimentos em bases de dados reais pois, uma vez obtidos os resultados nessas bases, pode-se inferir sobre as estruturas de dependência presentes.

No próximo capítulo, bases de dados do repositório da *UCI* [BlM10] são submetidas a experimentos para comparação e estudo da eficácia de cada método explorado neste trabalho.

CAPÍTULO 6

ESTUDOS DE CASO COM BASES DO REPOSITÓRIO *UCI*

Neste capítulo são realizados experimentos com bases públicas para observação do comportamento do método proposto em bases não controladas e observar o seu desempenho quando comparado com outros métodos de combinação de classificadores. Para tal, foram escolhidas 12 (doze) bases do repositório *UCI Machine Learning Repository* [BlM10]. As bases foram escolhidas de tal forma que o alvo a ser modelado deveria ser dicotômico ou sofrer algum tipo de adaptação para possuir apenas duas classes. Isso foi necessário porque o método foi implementado para utilização em classificações dicotômicas e/ou geração de escores. Os resultados são submetidos a análises estatísticas para a determinação da significância dos números encontrados.

As simulações foram realizadas de forma que cada técnica pura seja utilizada como classificador de controle para efeito de comparação com os demais métodos de combinação de classificadores (*Bagging*, *Boosting* e Segmentações). Nos métodos de combinação que superaram significativamente o classificador base, foram realizados testes de significância para determinar se existe diferença entre estes. Com estes testes deseja-se saber se uma técnica de combinação supera a outra. Estas últimas comparações não são disponibilizadas diretamente nas tabelas de resultados, a fim de evitar poluição visual de informações, entretanto elas são analisadas e discutidas no texto.

6.1 – PARÂMETROS E EXPERIMENTOS

Nesta seção são determinados e comentados os parâmetros utilizados nos experimentos para as diversas técnicas utilizadas neste capítulo.

6.1.1 – PARÂMETROS UTILIZADOS

Neste capítulo também são utilizados como classificadores: Redes Neurais, Regressão Logística e Regressão Linear. A metodologia de escolha dos parâmetros para treinamento das Redes Neurais foi a mesma apresentada no Capítulo 5. Não foram utilizados métodos de seleção de variáveis para a Regressão Logística e Regressão Linear, portanto, estes métodos não precisaram de nenhuma metodologia de escolha de parâmetros.

Nos experimentos deste capítulo, os parâmetros utilizados pelos métodos *Bagging e Boosting* foram os mesmos descritos no Capítulo 5. Os três tipos de segmentações usadas no capítulo anterior também foram aqui experimentadas, porém com algumas adaptações em seus parâmetros. O número máximo de níveis (profundidade) foi definido como 2 (dois) para as duas primeiras bases. Este parâmetro foi determinado após alguns testes preliminares, os quais demonstraram que, a partir destes níveis, os ganhos eram pouco relevantes ou não existiam. Assim como na classificação com Árvores de Decisão [MaR05], na segmentação de modelos, quanto maiores os níveis de profundidades, maiores são as chances de *overfitting*. Então, procura-se o treinamento de modelos com poucos exemplos, para evitar uma alta variância nos conjuntos de teste. Na prática, deseja-se um número pequeno de níveis de segmentação, pois, quanto maior o número de segmentos, maiores são as dificuldades de distribuição dos modelos e maior o tempo de processamento do treinamento. Outro parâmetro importante na segmentação, que também tem alta correlação com a variância, é o número mínimo de elementos em um nó, definido diferentemente aqui como 5% do tamanho total da amostra. Este último parâmetro também exigiu alguns testes preliminares para a definição mais adequada.

Mais uma vez, os experimentos foram realizados com as implementações dos algoritmos de combinação (*Bagging*, *Boosting*, *NNTree* e segmentações) utilizando o software SAS Base/Stat v9.1.3 SP 4 com SAS Enterprise Miner v4.0 [SAS02] e os resultados (tabelas e gráficos) foram gerados no Microsoft Excel versão 2007 Enterprise.

6.1.2 – TRATAMENTO DAS VARIÁVEIS

As variáveis originalmente categóricas não sofreram qualquer tipo de agrupamento de categorias. Sendo assim, os seus valores originais foram utilizados para a codificação binária. A transformação binária utilizada foi a padrão do software SAS Enterprise Miner versão 4.0, que transforma a variável em *n* variáveis binárias com valor 0 (zero) ou 1 (um), onde *n* é o número de categorias da variável [SAS02].

Para treinamento dos classificadores, todas as variáveis numéricas deste capítulo foram normalizadas com valores contínuos entre 0 (zero) e 1 (um). Como, internamente, as implementações das segmentações *RISKSEG* e *SegTree* exigem que as variáveis sejam categóricas para dividir os subconjuntos, optou-se por não utilizar o parâmetro *rQtdeDivisões* (ver seção 4.2.1.6), que faz automaticamente (internamente) a categorização das variáveis numéricas. Esta escolha foi feita porque a quantidade de experimentos é grande, o que aumentaria bastante o tempo de treinamento, pois a operação teria que ser rodada muitas vezes para cada base de dados. Então, as variáveis numéricas foram categorizadas manualmente de modo a possuírem quatro categorias, cada uma com aproximadamente 25% dos exemplos. Estas variáveis categorizadas são utilizadas apenas para segmentação, pois nos treinamentos das técnicas elas são usadas em sua forma numérica, normalizadas entre 0 (zero) e 1 (um).

6.1.3 – ORGANIZAÇÃO DOS EXPERIMENTOS

Para os experimentos utilizando as bases de dados do repositório *UCI*, foi utilizada a técnica *K-fold Cross Validation*, onde o valor escolhido de *k* foi 10, obtendo-se assim 10 valores de erro para as realizações dos testes estatísticos. Esta forma de experimentar é muito encontrada em artigos que utilizam combinação de classificadores, como observado em [ZTD01]. As bases de dados foram divididas em 10 partes iguais, cada uma com 10% dos dados, sendo que, para cada experimento, uma das partes ficou sendo o conjunto de testes e o restante disponível para treinamento e validação. Como mostra a Figura 6.1, os conjuntos de treinamento e validação foram definidos para serem 65% e 35% do total da base, respectivamente, e, portanto, eles representam 72,2% e 27,8% dos registros disponíveis para estes fins, respectivamente.

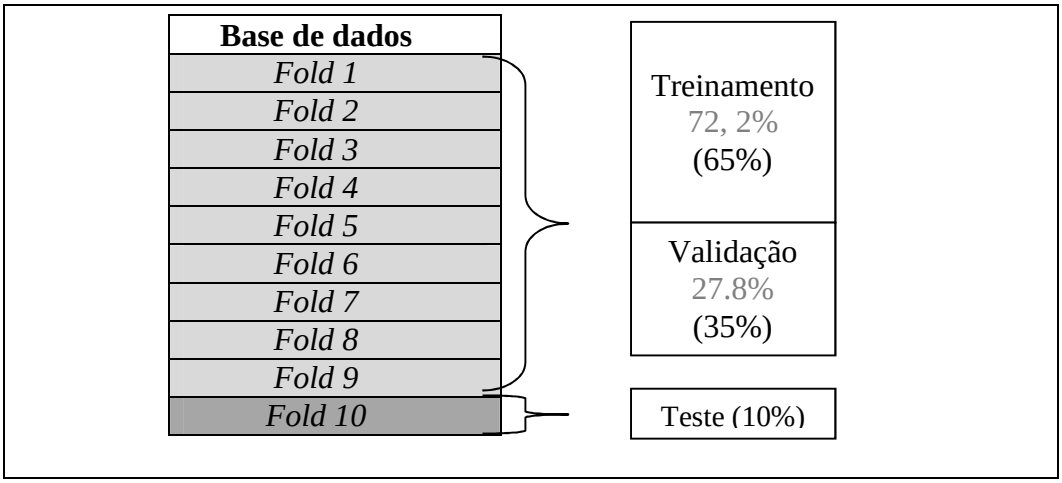


Figura 6.1 – Estrutura dos conjuntos aplicada em cada base da *UCI*.

Para as bases do repositório *UCI*, foram feitos experimentos envolvendo três técnicas: Regressão Linear, Redes Neurais e Regressão Logística. Para cada técnica, foram estudados os

comportamentos dos métodos de combinação e geração de múltiplos classificadores. Os resultados de todas as bases foram mensurados através da taxa de erro de classificação. Neste capítulo, denomina-se "*SegTree"* a segmentação por *Information Gain* e "Simples" as classificações utilizando apenas uma das técnicas puras (Redes Neurais, Regressão Linear e Regressão Logística) com classificador único.

Para a seleção dos melhores parâmetros, foram criadas, para cada base de dados, três tabelas de resultados nas seções de seleção de variáveis do capítulo. Na escolha da melhor configuração de Rede Neural, são experimentados e apresentados em uma tabela diversos parâmetros para os métodos, então é escolhida a melhor configuração (menor erro de classificação) para cada método. Em caso de empate de configurações nos métodos de segmentação, escolhe-se a configuração que possui o menor número de classificadores, ou seja, a do nível 1, os demais critérios de desempate são aplicados para todos os métodos, na seguinte ordem: menor número de neurônios na camada intermediária (mais simples) e a configuração treinada com *Backpropagation* (o método mais conhecido). Na seleção do melhor nível de profundidade dos métodos de segmentação, utilizando Regressão Linear e Regressão Logística, são criadas duas tabelas de resultados, uma para cada Regressão Estatística, com os erros médios das segmentações (*NNTree*, *RISKSEG* e *SegTree*) para cada nível. Caso exista empate, o critério utilizado para desempate é a seleção do menor número de classificadores, ou seja, escolhe-se o nível 1.

Após a seleção dos parâmetros, para cada base de dados é criada uma tabela contendo os resultados dos métodos e técnicas escolhidos. Os valores que são apresentados representam os percentuais dos erros de classificação médios. Nas análises de desempenho, utilizou-se teste *t-Student* para amostras pareadas, com nível de significância de 5%. Valores de erros médios menores que o erro do classificador Simples (controle) são colocados em negrito e valores menores e estatisticamente diferentes em relação ao experimento de controle são marcados com asterisco (*). Valores de erros médios superiores ao do experimento de controle e significativamente piores são colocados em itálico.

6.2 – ESTUDO DE CASO - *CHESS*

6.2.1 – DESCRIÇÃO DA BASE DE DADOS

Este estudo de caso utiliza uma base de dados com informações de um jogo de xadrez. Esta base contém posições e arranjos das peças em um tabuleiro de xadrez, codificadas e representadas por 36 variáveis para uso na predição e classificação da variável dicotômica y que assume valor 0 (zero) ou valor 1 (um), tal que 0 (zero) representa que as peças brancas podem vencer e 1 (um)

que elas não podem vencer. Algum detalhamento é fornecido pelo repositório *UCI* [BlM10]. Porém, pode-se encontrar maiores informações e descrições mais detalhadas das variáveis em alguns artigos que também exploraram esta base [Sha87]. A Tabela 6.1 apresenta um resumo da base de dados.

Tabela 6.1 – Características gerais da base *Chess*.

Quantidade de Variáveis	36 categóricas
Quantidade de Registros	3.196 registros
Distribuição do Alvo	1.527 registros com 0 (zero) e 1.669 registros com 1 (um)

6.2.2 – SELEÇÃO DOS PARÂMETROS

Na Tabela 6.2 são apresentados os erros médios para cada uma das configurações e em negrito são evidenciados os que obtiveram menor erro para a técnica experimentada e, consequentemente, os que foram escolhidos para a comparação entre os métodos de combinação de classificadores. Esta convenção é utilizada para seleção dos parâmetros das Redes Neurais e quantidade de níveis dos métodos de segmentação, em todos os experimentos deste capítulo.

Conforme demonstra a Tabela 6.2, não houve uma configuração predominante para Rede Neural. Para uma boa parte dos métodos, alguns dos erros de classificação foram iguais para os algoritmos de treinamento utilizados (*Backpropagation e Levenberg-Marquardt*), o que pode indicar que os dois algoritmos geraram *MLPs* com pesos semelhantes para a mesma quantidade de neurônios. Os resultados de nível 2 foram os melhores para os três tipos de segmentação. Destaque para os erros do segundo nível com 10 neurônios do *RISKSEG* que apresentou resultados bem melhores que qualquer outra configuração. A significância deste resultado será testada com os demais métodos na próxima seção.

Tabela 6.2 – Erro médio para seleção da melhor configuração da Rede Neural (*Chess*).

Método	Nível	*Backpropagation*			*Levenberg-Marquardt*		
		3	10	20	3	10	20
Bagging	-	3,63	4,57	3,51	3,51	4,63	**3,29**
Boosting	-	4,16	4,85	**3,69**	4,16	4,85	3,69
Simples	-	**4,07**	5,29	4,19	4,07	5,29	4,19
NNTree	**1**	2,93	3,15	2,97	2,93	3,15	2,97
	2	2,85	3,09	**2,81**	2,85	3,09	2,81
RISKSEG	**1**	2,47	1,41	2,25	2,47	1,56	2,25
	2	2,16	**0,35**	1,85	2,16	0,48	1,85
SegTree	**1**	3,29	2,19	4,38	3,29	2,19	4,38
	2	2,10	**1,09**	2,66	2,10	1,09	2,66

Conforme as Tabelas 6.3 e 6.4, todas as segmentações para as Regressões, exceto o método *NNTree* com Regressão Logística, apresentaram melhores resultados com um número maior de modelos, ou seja, segmentações com dois níveis de profundidade obtiveram erro

menor. A maior variação de valores de erros entre os níveis está na *NNTree*, que obteve uma melhora expressiva do primeiro para o segundo nível na Regressão Linear.

Tabela 6.3 – Erro médio para seleção do melhor nível da Regressão Linear (*Chess*).

Regressão Linear		
Método	**Nível**	**Média**
NNTree	**1**	8,08
	2	**3,18**
RISKSEG	**1**	4,04
	2	**3,25**
SegTree	**1**	5,26
	2	**2,28**

Tabela 6.4 – Erro médio para seleção do melhor nível da Regressão Logística (*Chess*).

Regressão Logística		
Método	**Nível**	**Média**
NNTree	**1**	**2,63**
	2	2,65
RISKSEG	**1**	1,06
	2	**1,02**
SegTree	**1**	2,60
	2	**2,25**

6.2.3 – ANÁLISE DOS RESULTADOS

A Tabela 6.5 apresenta os resultados obtidos pelas três técnicas de predição, cruzadas com os métodos de combinação. Os valores são as médias dos erros de classificação encontrados nos experimentos. A comparação é sempre feita a partir do classificador de controle (Simples) de cada técnica.

Tabela 6.5 – Médias das taxas de erros e intervalos de confiança para a base *Chess*.

Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
Redes Neurais	4,07 ±1,05	**3,29 ±0,75**	**3,69 ±0,68**	**2,81 ±0,59***	**0,35 ±0,14***	**1,09 ±0,42***
Reg. Linear	8,57 ±1,64	**7,13 ±1,1**	**5,48 ±0,97***	**3,18 ±0,68***	**3,25 ±0,77***	**2,28 ±0,55***
Reg. Logística	2,6 ±0,73	**2,41 ±0,55**	**2,41 ±0,62**	2,63 ±0,51	**1,02 ±0,27***	**2,25 ±0,83**

A combinação de classificadores utilizando Redes Neurais, o *Bagging* e o *Boosting* apresentaram melhores médias do que o classificador Simples, porém não foram significativas. Ainda nesta técnica, todas as três segmentações apresentaram médias melhores e significativas quando comparadas com o classificador de controle, com destaque para o método *RISKSEG* que apresentou média significativamente melhor que o segundo melhor erro (*SegTree*).

Na Regressão Linear, todos os métodos de combinação obtiveram médias de erros melhores e significativas em relação ao classificador Simples. Nesta técnica, a segmentação *SegTree* obteve o menor erro entre os métodos e também foi significativamente menor quando comparado com *NNTree* e *RISKSEG*. Ainda na Regressão Linear, apesar do *RISKSEG* obter resultados piores que o *SegTree*, este método foi melhor do que o *Bagging* e *Boosting* e não apresentou diferença significativa em relação ao *NNTree*.

Na Regressão Logística, apesar de algumas técnicas apresentarem médias de erros menores do que o classificador de controle, apenas o método *RISKSEG* obteve resultado significativamente melhor. Pode-se observar o potencial para melhoria de desempenho que a

técnica *RISKSEG* pode proporcionar nesta base de dados, pois apresentou resultados significativamente melhores para todas as técnicas e teve o menor erro em duas delas.

6.3 –ESTUDO DE CASO – GERMAN CREDIT

6.3.1 – DESCRIÇÃO DA BASE DE DADOS

Para este estudo de caso, utilizou-se uma base de dados com informações de crédito contendo 1.000 registros. Esta base é formada pela variável alvo clássica no estudo de risco de crédito. Esta variável define os credores como sendo bons ou maus pagadores (1=Bom e 2=Mau), segundo critérios envolvendo atrasos de pagamentos. Para uso nas Regressões o 2 (dois) foi transformado em 0 (zero). As variáveis são em sua maioria atributos sócio-econômicos. Ao todo são 20 variáveis preditoras, sendo que 7 (sete) delas são contínuas e 13 categóricas. Um resumo da base é apresentado na Tabela 6.6. Maiores detalhes do preenchimento das variáveis pode ser encontrado no repositório da *UCI* [BlM10].

Tabela 6.6 – Características gerais da base *German*.

Quantidade de Variáveis	7 numéricas e 13 categóricas
Quantidade de Registros	1.000 registros
Distribuição do Alvo	700 registros com 1 (um) e 300 registros com 0 (zero)

6.3.2 – SELEÇÃO DOS PARÂMETROS

Conforme demonstra a Tabela 6.7, houve muitos erros iguais entre o treinamento com *Backpropagation* e *Levenberg-Marquardt*, assim como aconteceu na base *Chess*. Esta foi uma característica observada em muitas das bases de dados do Repositório *UCI* [BlM10] utilizadas neste trabalho. No *Bagging*, o treinamento da Rede Neural com *Levenberg-Marquardt* atingiu seu melhor resultado com apenas três neurônios na camada intermediária. O treinamento com *Backpropagation* também atingiu tal resultado, mas precisou de 20 neurônios.

Tabela 6.7 – Erro médio para seleção da melhor configuração Rede Neural (*German*).

Método	Nível	*Backpropagation*			*Levenberg-Marquardt*		
		3	10	20	3	10	20
Bagging	-	25,10	24,70	24,60	**24,60**	25,00	25,30
Boosting	-	25,60	26,30	**25,40**	25,60	26,30	25,40
Simples	-	25,80	27,30	**25,20**	25,80	27,30	25,20
NNTree	**1**	28,59	29,86	**27,10**	28,59	29,86	27,10
	2	30,04	31,22	28,09	30,04	31,22	28,09
RISKSEG	**1**	27,40	26,40	**25,30**	27,40	26,40	25,30
	2	28,10	27,00	25,80	28,10	27,00	25,80
SegTree	**1**	**25,70**	27,00	25,90	25,70	27,00	25,90
	2	26,20	27,50	25,70	26,20	27,50	25,70

Nenhuma das segmentações obteve melhores resultados no segundo nível, isto pode indicar que um nível é suficiente para ter o melhor resultado para este tipo de método, nesta base. A não observância de melhores resultados no segundo nível pode ser devido à pequena quantidade de registros da base, pois ela é sempre mais reduzida para os treinamentos dos níveis mais profundos da árvore de classificadores. Claramente, pode-se observar que apenas o *Bagging* possui média de erro de classificação menor do que o classificador de controle (Simples) e na Tabela 6.10 pode-se observar que não é significativo.

Nas Tabelas 6.8 e 6.9 são apresentados os erros médios para cada uma das configurações e em negrito são apresentados os que obtiveram menor erro para a técnica experimentada e consequentemente os que foram escolhidos para a comparação entre os métodos de combinação de classificadores. Seguindo a mesma tendência dos experimentos com Redes Neurais, em todos os métodos as segmentações para as Regressões pioraram no segundo nível, exceto o *SegTree*, que teve uma pequena melhora. Como dito anteriormente, isto pode ser causado pelo tamanho reduzido da base de dados (1.000 registros).

Tabela 6.8 – Erro médio para seleção do melhor nível da Regressão Linear *(German)*.

Regressão Linear		
Método	**Nível**	**Média**
NNTree	**1**	**29,25**
	2	31,10
RISKSEG	**1**	**26,30**
	2	27,00
SegTree	**1**	26,10
	2	**26,00**

Tabela 6.9 – Erro médio para seleção do melhor nível da Regressão Logística (*German*).

Regressão Logística		
Método	**Nível**	**Média**
NNTree	**1**	**25,95**
	2	27,04
RISKSEG	**1**	**26,30**
	2	26,80
SegTree	**1**	**24,70**
	2	24,80

6.3.3 – ANÁLISE DOS RESULTADOS

A Tabela 6.10 apresenta os resultados obtidos pelas três técnicas de predição, cruzadas com os métodos de combinação. Os valores são as médias dos erros de classificação encontradas nos experimentos. A comparação é sempre feita a partir do classificador de controle (Simples) de cada técnica. Estas definições valem para todas as seções de análise de resultados dos experimentos deste capítulo.

Tabela 6.10 – Médias das taxas de erros e intervalos de confiança para a base *German*.

Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
Redes Neurais	25,2 ±1,79	**24,6 ±2,24**	25,4 ±2,39	*27,1 ±1,97*	25,3 ±1,84	25,7 ±1,2
Reg. Linear	29 ±2,31	**26,9 ±2,83**	**28,5 ±3,33**	29,25 ±3,08	**26,3 ±2**	**26 ±1,77***
Reg. Logística	25,7 ±2,08	**25 ±1,51**	**24,6 ±2,04**	25,95 ±1,94	26,3 ±2,51	**24,7 ±2,08**

Observa-se na Tabela 6.10 que apenas na Regressão Linear e com o método *SegTree* foi possível melhorar significativamente os resultados do classificador Simples. Na Rede Neural, a

segmentação *NNTree* piorou significativamente os resultados de classificação. Uma possível explicação para estes resultados é a pequena quantidade de exemplos do estudo de caso, que gera segmentos ainda menores na árvore de modelos. Por esta razão, quando a segmentação é aplicada, os resultados obtidos em cada segmento não melhoram o desempenho, pois constroem classificadores mais fracos quando comparados com o do nível superior. Nos experimentos com Redes Neurais e Regressão Logística, o classificador Simples capturou toda a informação contida nas variáveis de entrada, ou seja, nenhum dos métodos de combinação obteve uma melhora significativa. Pode-se inferir que as segmentações não apresentaram taxas de erro melhores, possivelmente pela criação de subgrupos com menos registros e com o agravante da necessidade de destinação de parte dos exemplos para formação do conjunto de validação.

6.4 –ESTUDO DE CASO – *CAR EVALUATION*

6.4.1 – DESCRIÇÃO DA BASE DE DADOS

Esta base de dados foi derivada de um modelo simples de decisão hierárquica originalmente desenvolvido para a demonstração. O modelo avalia os carros de acordo com uma determinada estrutura [BlM10]. A base é composta por 6 (seis) atributos categóricos e um alvo. As possíveis classes do alvo são: *good* (bom), *vgood* (muito bom), *acc* (aceitável) e *unacc* (inaceitável). As duas primeiras classes (*good* e *vgood*) possuem respectivamente 69 e 65 casos e foram agrupadas com a classe *acc* que possui 384 casos. O resultado foi um alvo dicotômico, sendo atribuído 1 (um) para as classes *acc*, *good*, *vgood* e 0 (zero) para a classe *unacc*. Um resumo da base é apresentado na Tabela 6.11.

Tabela 6.11 – Características gerais da base *Car*.

Quantidade de Variáveis	6 categóricas
Quantidade de Registros	1.728 registros
Distribuição do Alvo	518 registros com 1 (um) e 1.210 registros com 0 (zero)

6.4.2 – SELEÇÃO DOS PARÂMETROS

Como pode ser visto na Tabela 6.12, houve muitos erros iguais entre os treinamentos com *Backpropagation* e *Levenberg-Marquardt*, assim como aconteceu nas bases *Chess e German.* Para todos os métodos de combinação, três neurônios na camada intermediária foram suficientes para obter a melhor configuração usando Rede Neural. No *Bagging*, o treinamento da Rede Neural com *Levenberg-Marquardt* atingiu seu melhor resultado com apenas três neurônios na camada intermediária. Apenas a *NNTree* obteve seus melhores resultados no primeiro nível, porém este resultado foi muito próximo do resultado da melhor configuração para o classificador Simples da Rede Neural.

Tabela 6.12 – Erro médio para seleção da melhor configuração Rede Neural (*Car*).

Método	Nível	*Backpropagation*			*Levenberg-Marquardt*		
		3	10	20	3	10	20
Bagging	-	1,16	2,37	3,65	**1,04**	2,37	3,82
Boosting	-	**1,33**	3,65	4,57	1,33	3,65	4,57
Simples	-	**1,62**	3,07	4,75	1,62	3,07	4,75
NNTree	**1**	**1,59**	2,90	3,30	1,59	2,90	3,30
	2	1,60	2,87	3,14	1,60	2,87	3,14
RISKSEG	**1**	0,64	2,60	3,12	0,64	2,60	3,12
	2	**0,23**	2,14	2,60	0,40	2,14	2,60
SegTree	**1**	1,16	2,43	3,47	1,16	2,43	3,47
	2	**1,10**	2,26	3,47	1,10	2,26	3,47

Conforme demonstram as Tabelas 6.13 e 6.14, todos os métodos de segmentação tiveram seus resultados melhorados do primeiro para o segundo nível nas duas técnicas analisadas, exceto o *NNTree* com Regressão Logística. Esta mesma técnica com Regressão Linear obteve ganho bastante representativo entre os níveis. Ainda na Regressão Linear, a *NNTree* foi o método que obteve a pior média e a única segmentação que necessitou de pelo menos dois níveis para atingir um valor de erro menor que 5%.

Tabela 6.13 – Erro médio para seleção do melhor nível da Regressão Linear *(Car)*.

Regressão Linear		
Método	**Nível**	**Média**
NNTree	**1**	17,59
	2	**4,75**
RISKSEG	**1**	4,28
	2	**3,88**
SegTree	**1**	4,57
	2	**4,28**

Tabela 6.14 – Erro médio para seleção do melhor nível da Regressão Logística (*Car*).

Regressão Logística		
Método	**Nível**	**Média**
NNTree	**1**	**4,45**
	2	4,47
RISKSEG	**1**	3,59
	2	**1,85**
SegTree	**1**	4,34
	2	**4,22**

6.4.3 – ANÁLISE DOS RESULTADOS

Na Tabela 6.15, observa-se que as três técnicas de classificação tiveram suas taxas de erro diminuídas em relação a seus classificadores simples, com destaque especial para a Regressão Linear, que foi a mais beneficiada, com diminuição de 34,84% de erro para 3,88% no *RISKSEG*. Na Rede Neural, apesar de todos os métodos terem média de erros menores que o classificador Simples, apenas o *RISKSEG* teve uma melhora significativa em relação ao classificador Simples. Na Regressão Linear os métodos *NNTree*, *RISKSEG* e *SegTree* tiveram médias de erros menores (significativas) quando compradas com o classificador de controle. Para a Regressão Logística todos os erros das segmentações (*NNTree*, *RISKSEG* e *SegTree*) foram menores do que o classificador Simples, porém apenas o *RISKSEG* foi significativamente melhor. Nesta base de dados, o *RISKSEG* apresentou melhoras significativas em todas as três técnicas em comparação com os classificadores Simples de cada técnica. Em relação ao segundo

melhor valor de erro, o *RISKSEG* também apresentou melhora significativa na Regressão Logística e p-valor entre 5% e 6% para Redes Neurais e Regressão Linear, ou seja, muito próximo do nível de significância utilizado nos experimentos.

Tabela 6.15 – Médias das taxas de erros e intervalos de confiança para a base *Car*.

Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
Redes Neurais	1,62 ±0,58	**1,04 ±0,58**	**1,33 ±0,63**	**1,59 ±0,52**	**0,23 ±0,18***	**1,1 ±0,68**
Reg. Linear	34,84 ±2,98	35,36 ±2,55	35,71 ±3,11	**4,75 ±0,79***	**3,88 ±0,92***	**4,28 ±1,02***
Reg. Logística	4,69 ±0,91	4,69 ±0,96	4,69 ±0,83	**4,45 ±1,05**	**1,85 ±0,78***	**4,22 ±1,23**

6.5 –ESTUDO DE CASO – *MAGIC GAMMA TELESCOPE*

6.5.1 – DESCRIÇÃO DA BASE DE DADOS

Os dados desta base foram gerados para simular o registro de partículas de alta energia gama em um telescópio terrestres utilizando a técnica de imagem. O alvo é dicotômico e formado por: g (*gamma*) e h (*hadron*), transformados respectivamente em 0 (zero) e 1 (um). Maiores detalhes sobre esta base de dados pode ser encontrada no repositório da *UCI* [BlM10]. Um resumo da base é apresentado na Tabela 6.16.

Tabela 6.16 – Características gerais da base *Magic*.

Quantidade de Variáveis	10 numéricas
Quantidade de Registros	19.020 registros
Distribuição do Alvo	12.332 registros com 1 (um) e 6.688 registros com 0 (zero)

6.5.2 – SELEÇÃO DOS PARÂMETROS

Observa-se na Tabela 6.17, que o *Bagging* foi o único método que teve os seus valores de erros alterados com a mudança do algoritmo de treinamento da Rede Neural (*Backpropagation* e *Levenberg-Marquardt*). Nesta técnica, dentre as segmentações, somente o *RISKSEG* teve seu resultado melhorado no segundo nível, para todas as outras, o segundo nível aumentou o erro médio. Observou-se também, que três neurônios na camada intermediária foram suficientes para obter os melhores resultados para todos os métodos de combinação.

Tabela 6.17 – Erro médio para seleção da melhor configuração Rede Neural (*Magic*).

Método	Nível	*Backpropagation*			*Levenberg-Marquardt*		
		3	10	20	3	10	20
Bagging	-	**14,98**	15,98	16,88	15,01	16,05	16,89
Boosting	-	**15,26**	16,07	17,85	15,26	16,07	17,85
Simples	-	**15,16**	16,16	17,85	15,16	16,16	17,85
NNTree	**1**	**15,24**	15,97	16,84	15,24	15,97	16,84
	2	15,24	15,83	16,76	15,24	15,83	16,76
RISKSEG	**1**	14,17	14,53	14,99	14,17	14,53	14,99
	2	**13,51**	13,75	14,38	13,51	13,75	14,38
SegTree	**1**	**14,84**	15,96	16,69	14,84	15,96	16,69
	2	14,87	15,46	15,63	14,87	15,46	15,63

Conforme demonstram as Tabelas 6.18 e 6.19, todos os métodos de segmentação tiveram seus resultados melhorados do primeiro para o segundo nível nas duas técnicas analisadas, exceto o *NNTree* com Regressão Linear. Ainda, esta configuração obteve média de erro muito alta quando comparada com às demais segmentações nos dois níveis de profundidade.

Tabela 6.18 – Erro médio para seleção do melhor nível da Regressão Linear *(Magic)*.

Regressão Linear		
Método	**Nível**	**Média**
NNTree	**1**	**27,31**
	2	29,36
RISKSEG	**1**	17,20
	2	**16,06**
SegTree	**1**	19,71
	2	**18,73**

Tabela 6.19 – Erro médio para seleção do melhor nível da Regressão Logística (*Magic*).

Regressão Logística		
Método	**Nível**	**Média**
NNTree	**1**	18,72
	2	**18,69**
RISKSEG	**1**	16,79
	2	**15,40**
SegTree	**1**	19,45
	2	**18,29**

6.5.3 – ANÁLISE DOS RESULTADOS

A Tabela 6.20 mostra que as três técnicas de classificação tiveram suas taxas de erro diminuídas, com destaque especial para a Regressão Linear, que foi a mais beneficiada, com diminuição de 32,03% de erro para 16,06% no *RISKSEG*. Na Rede Neural, apenas as segmentações *RISKSEG* e *SegTree* obtiveram erros menores e significativos em relação ao classificador Simples. Com Regressão Linear, os métodos *NNTree*, *RISKSEG* e *SegTree* tiveram médias de erros menores (significativas) quando comparadas com o classificador de controle. Para a Regressão Logística, todos os erros das segmentações (*NNTree*, *RISKSEG* e *SegTree*) foram menores do que o classificador Simples, porém apenas o *RISKSEG* foi significativamente melhor.

Tabela 6.20 – Médias das taxas de erros e intervalos de confiança para a base *Magic*.

Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
Redes Neurais	15,16 ±0,53	**14,98 ±0,55**	15,26 ±0,53	15,24 ±0,53	**13,51 ±0,58***	**14,84 ±0,6***
Reg. Linear	32,03 ±2,81	32,62 ±2,07	32,75 ±1,93	**27,31 ±2,65***	**16,06 ±0,65***	**18,73 ±0,56***
Reg. Logística	20,98 ±0,44	**20,92 ±0,42**	**20,93 ±0,42**	**18,69 ±0,55***	**15,4 ±0,51***	**18,29 ±0,67***

É importante ressaltar que, nesta base de dados, o *RISKSEG* apresentou melhoras significativas em todas as três técnicas em comparação com os classificadores Simples de cada técnica. Também apresentou melhora significativa em todas as três técnicas, quando comparado ao método com o segundo menor erro de classificação.

6.6 –ESTUDO DE CASO – *MUSHROOM*

6.6.1 – DESCRIÇÃO DA BASE DE DADOS

Esta base de dados tem o objetivo de identificar as espécies comestíveis de cogumelos, através de suas características como: cor, formato, habitat e cheiro. São 22 variáveis categóricas de entradas e um alvo (comestível ou não). Originalmente a letra "e" representa os cogumelos comestíveis e a letra "p" os venenosos, mas elas foram transformadas, respectivamente, em 1 (um) e 0 (zero). Uma das 22 variáveis da base possui apenas uma categoria e foi descartada. Maiores detalhes sobre esta base de dados pode ser encontrada no repositório da *UCI* [BlM10]. Um resumo da base é apresentado na Tabela 6.21.

Tabela 6.21 – Características gerais da base *Mushroom*.

Quantidade de Variáveis	22 variáveis categóricas (1 com apenas uma categoria)
Quantidade de Registros	8.124 registros
Distribuição do Alvo	4.208 registros com 1 (um) e 3.916 registros com 0 (zero)

6.6.2 – SELEÇÃO DOS PARÂMETROS

Observa-se na Tabela 6.22, que o problema apresenta uma solução relativamente simples, pois, com exceção do *Bagging*, todos os métodos apresentaram configurações com erros menores do que 0,02%. Os métodos apresentaram muitos resultados iguais mesmo com a mudança do algoritmo de treinamento da Rede Neural, quantidade de neurônios na camada intermediária e níveis de profundidade da segmentação. Não houve concentração em uma quantidade de neurônios na camada intermediária e os melhores resultados para cada método foram obtidos com diferentes números.

Tabela 6.22 – Erro médio seleção da melhor configuração Rede Neural (*Mushroom*).

Método	Nível	*Backpropagation*			*Levenberg-Marquardt*		
		3	10	20	3	10	20
Bagging	-	0,14	0,14	**0,10**	0,12	0,15	0,11
Boosting	-	0,36	**0,22**	5,21	0,36	0,22	5,21
Simples	-	0,43	0,21	**0,18**	0,43	0,21	0,18
NNTree	1	0,02	0,02	0,02	0,02	0,02	0,02
	2	**0,01**	0,02	0,01	0,01	0,02	0,01
RISKSEG	1	0,02	0,14	0,02	0,02	0,14	0,02
	2	**0,01**	0,04	0,01	0,01	0,04	0,01
SegTree	1	**0,00**	0,01	0,01	0,00	0,01	0,01
	2	0,01	0,02	0,01	0,01	0,02	0,01

Conforme demonstram as Tabelas 6.23 e 6.24, todos os métodos de segmentação atingiram erros de classificação igual a zero. Isto confirma que este problema possui baixa complexidade. Observa-se que, nestes experimentos, mesmo a Regressão Linear, que vem

apresentando os piores resultados, consegue zerar os erros para todos os métodos de classificação.

Tabela 6.23 – Erro médio seleção do melhor nível da Regressão Linear (*Mushroom*).

Regressão Linear		
Método	**Nível**	**Média**
NNTree	**1**	**0,00**
	2	0,00
RISKSEG	**1**	0,00
	2	**0,00**
SegTree	**1**	0,00
	2	**0,00**

Tabela 6.24 – Erro médio seleção do melhor nível da Regressão Logística (*Mushroom*).

Regressão Logística		
Método	**Nível**	**Média**
NNTree	**1**	0,00
	2	**0,00**
RISKSEG	**1**	0,00
	2	**0,00**
SegTree	**1**	0,00
	2	**0,00**

6.6.3 – ANÁLISE DOS RESULTADOS

Na Tabela 6.25 observa-se que, nos experimentos envolvendo Rede Neural, a media de erros de todos os métodos foram menores do que o classificador Simples, exceto o *Boosting*, que apresentou valor superior de erro não significativo. Ainda analisando as Redes Neurais, nota-se que apenas o *Bagging* teve seu valor médio de erro inferior ao do classificador de controle, mas não se mostrou significante. Os classificadores Simples para Regressão Linear e Logística apresentaram erro igual a zero, o *Bagging* não piorou significativamente na Regressão Linear mesmo dando um valor médio de erro superior ao Simples e na Regressão Logística conseguiu erro médio igual a zero. Os experimentos com o *Boosting* demonstraram que para este método o erro piorou significativamente nas Regressões. Todas as segmentações com Regressões obtiveram erro médio igual a zero.

Tabela 6.25 – Médias das taxas de erros e intervalos de confiança para a base *Mushroom*.

Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
Redes Neurais	0,18 ±0,08	**0,1 ±0,07**	0,22 ±0,11	**0,01 ±0,01***	**0,01 ±0,02***	**0 ±0***
Reg. Linear	0 ±0	0 ±0	*48,2 ±1,3*	0 ±0	0 ±0	0 ±0
Reg. Logística	0 ±0	*0,05 ±0,03*	*48,2 ±1,3*	0 ±0	0 ±0	0 ±0

6.7 – ESTUDO DE CASO - *ABALONE*

6.7.1 – DESCRIÇÃO DA BASE DE DADOS

O objetivo da base *Abalone* é predizer a idade de um molusco da Califórnia tendo como base medidas físicas, como peso e diâmetro. A idade do molusco é determinada pelo número de anéis existentes no seu interior e para se contar tais anéis é preciso cortar o molusco e observá-lo no microscópio, que é uma tarefa lenta e cansativa. A idéia original básica é tentar determinar a idade sem cortar o molusco. Apesar do alvo original desta base possuir valores entre 1 e 29 (número de anéis), para este estudo de caso, o alvo foi dividido em 2 (duas) classes de forma a

manter mais ou menos 50% das ocorrências em cada. Uma classe com valor 0 (zero), para as ocorrências menores que 10 e outra com valor igual a 1 (um) para ocorrências maiores ou iguais a 10. Maiores informações sobre esta base podem ser obtidas no repositório da *UCI* [BlM10]. A Tabela 6.26 apresenta um resumo da base de dados.

Tabela 6.26 – Características gerais da base *Abalone*.

Quantidade de Variáveis	1 variável categórica e 7 variáveis numéricas
Quantidade de Registros	4.177 registros
Distribuição do Alvo	2.081 registros com 1 (um) e 2.096 registros com 0 (zero)

6.7.2 – SELEÇÃO DOS PARÂMETROS

Conforme é observado na Tabela 6.27, os métodos *SegTree* e *NNTree* precisaram de apenas um nível para atingir seus melhores resultados. O algoritmo *Levenberg-Marquardt* não gerou nenhum dos melhores resultados. Apesar das configurações dos métodos de combinação apresentarem erros menores que o método Simples, pode-se observar que todos apresentaram diferenças menores que 1%.

Tabela 6.27 – Erro médio para seleção da melhor configuração da Rede Neural (*Abalone)*.

Técnica	Nível	Backpropagation			Levenberg-Marquardt		
		3	10	20	3	10	20
Bagging	-	19,92	**19,86**	20,09	19,94	20,18	20,22
Boosting	-	**19,55**	19,68	19,55	19,99	20,90	21,14
Simples	-	19,97	**19,94**	20,30	20,18	22,82	22,82
NNTree	**1**	20,40	**20,35**	21,82	20,77	20,97	20,97
	2	20,91	21,31	21,73	20,50	21,51	21,51
RiskSeg	**1**	20,11	20,85	20,91	20,21	21,33	21,68
	2	**20,06**	20,76	20,70	20,28	21,93	22,06
SegTree	**1**	20,38	**19,97**	20,24	20,21	21,64	21,54
	2	20,38	19,97	20,24	20,21	21,64	21,54

Conforme demonstram as Tabelas 6.28 e 6.29, a maioria dos métodos de segmentação tiveram seus menores erros no segundo nível nas duas técnicas analisadas: apenas a *NNTree* com Regressão Linear precisou apenas de um nível para atingir seu menor erro. O método *SegTree* na Regressão Linear foi o que obteve a maior diferença de média de erro entre o nível 1 e o nível 2.

Tabela 6.28 – Erro médio para seleção do melhor nível da Regressão Linear (*Abalone)*.

Regressão Linear		
Técnica	**Nível**	**Média**
NNTree	**1**	**20,95**
	2	21,02
RiskSeg	**1**	21,38
	2	**20,97**
SegTree	**1**	21,91
	2	**20,57**

Tabela 6.29 – Erro médio para seleção do melhor nível da Regressão Logística (*Abalone)*.

Regressão Logística		
Técnica	**Nível**	**Média**
NNTree	**1**	20,68
	2	**20,64**
RiskSeg	**1**	20,92
	2	**20,66**
SegTree	**1**	20,88
	2	**20,61**

6.7.3 – ANÁLISE DOS RESULTADOS

Na Tabela 6.30, nos experimentos com Redes Neurais, apesar de *Boosting* e *Bagging* apresentarem médias de erros inferiores ao do classificador simples, nenhuma foi significativa. Na Regressão Linear, com exceção do método *Bagging*, que apresentou média de erro maior do que a do classificador Simples (significativa), todos os outros classificadores apresentaram melhoras significativas. Ainda na Regressão Linear, entre o *RISKSEG* com *SegTree* e *NNTree*, não houve diferença significativa, mas estes três classificadores tiveram médias menores e significativas em relação aos demais métodos de combinação. Na Regressão Logística, apenas o *SegTree* apresentou melhora significativa em relação ao classificador Simples. Entre o *RISKSEG* e o *SegTree*, a diferença do erro médio foi muito pequena e não houve diferença significativa. O *RISKSEG* apresentou um p-valor de 5,6% na comparação com o classificador Simples, ou seja ele quase alcançou o nível de significância adotado nestes experimentos.

Ainda sobre a Tabela 6.30, pode-se observar que a Regressão Linear sozinha (Simples) possui um erro muito grande, quando comparado com Redes Neurais e a Regressão Logística, mas as segmentações melhoraram muito o desempenho desta técnica (*RISKSEG*, *NNTree* e *SegTree*) e equiparam os valores dos seus erros diminuindo consideravelmente esta diferença.

Tabela 6.30 – Médias das taxas de erros e intervalos de confiança para a base *Abalone*.

Técnica	Simples	Bagging	Boosting	NNTree	RiskSeg	SegTree
Redes	19,94 ±1,22	**19,86 ±1,08**	**19,55 ±1,35**	20,35 ±0,94	20,06 ±1,19	19,97 ±0,88
Reg. Linear	31,19 ±4,87	*51,08 ±0*	**25,38 ±1,58***	**20,95 ±0,85***	**20,97 ±1,14***	**20,57 ±1,17***
Reg. Logística	21,26 ±1	21,69 ±1,29	21,33 ±1,22	**20,64 ±0,84**	**20,66 ±1,12**	**20,61 ±1,13**

6.8 – ESTUDO DE CASO - CONTRACEPTIVE METHOD CHOICE (CMC)

6.8.1 – DESCRIÇÃO DA BASE DE DADOS

Este conjunto de dados é um subconjunto do "*1987 do National Indonesia Contraceptive Prevalence Survey*". A amostra contém mulheres casadas que não estavam grávidas ou não

sabiam que estavam no momento da entrevista. O problema original é prever a escolha do método contraceptivo atual de uma mulher com base em suas características demográficas e sócio-econômicas. Para este estudo, a variável alvo foi agrupada de maneira a indicar se uma mulher deve ou não utilizar um método contraceptivo. Desta forma, o alvo foi agrupado em duas classes, 0 (zero) para representar o não uso de método contraceptivo e 1 (um) para indicar o seu uso. Maiores informações sobre esta base podem ser obtidas no repositório da *UCI* [BlM10]. A Tabela 6.31 apresenta um resumo da base de dados.

Tabela 6.31 – Características gerais da base *CMC*.

Quantidade de Variáveis	6 variáveis numéricas e 3 variáveis categóricas
Quantidade de Registros	1.473 registros
Distribuição do Alvo	629 registros com 0 (zero) e 844 registros com 1 (um)

6.8.2 – SELEÇÃO DOS PARÂMETROS

Observa-se na Tabela 6.32 que, para a maioria dos métodos, a melhor configuração foi baseada em 10 neurônios na camada intermediária. No que se refere ao algoritmo de treinamento da Rede Neural, basicamente não houve diferença no valor do erro médio, variando apenas no *Bagging*. Os métodos *RISKSEG* e *SegTree* precisaram apenas de um nível de profundidade para obter a melhor classificação. No método *SegTree* os erros médios foram iguais para o primeiro e segundo nível, possivelmente porque o algoritmo implementado só faz a divisão da folha se houver melhora de desempenho, ou seja, isto pode indicar que não houve melhora do erro em nenhuma das duas folhas do primeiro nível, desta forma árvores simplesmente não se dividem, independentemente da quantidade de níveis máximo que é utilizada como parâmetro.

Tabela 6.32 – Erro médio para seleção da melhor configuração da Rede Neural (*CMC*).

Método	Nível	*Backpropagation*			*Levenberg-Marquardt*		
		3	10	20	3	10	20
Bagging	-	27,63	26,95	28,45	**26,41**	27,49	28,38
Boosting	-	27,56	**26,61**	29,19	27,56	26,61	29,19
Simples	-	28,58	**27,83**	30,08	28,58	27,83	30,08
NNTree	1	29,01	27,76	29,42	29,01	27,76	29,42
	2	28,98	**27,52**	29,05	28,98	27,52	29,05
RISKSEG	1	29,53	28,71	**28,58**	29,53	28,71	28,58
	2	29,06	29,26	28,78	29,06	29,26	28,78
SegTree	1	28,51	**27,83**	30,08	28,51	27,83	30,08
	2	28,51	27,83	30,08	28,51	27,83	30,08

Nas Tabelas 6.33 e 6.34, pode-se observar que, nos experimentos com Regressão Logística, apenas um nível foi suficiente para obtenção dos melhores resultados em dois dos métodos. Porém, na Regressão Linear, todos os experimentos precisaram de dois níveis para os

menores erros. Isto indica que a Regressão Linear é um método mais fraco para resolução deste tipo de problema e necessita de mais níveis obtenção de melhores resultados.

Tabela 6.33 – Erro médio para seleção do melhor nível da Regressão Linear (*CMC*).

Regressão Linear		
Método	**Nível**	**Média**
NNTree	**1**	38,18
	2	**36,15**
RISKSEG	**1**	29,12
	2	**28,99**
SegTree	**1**	32,99
	2	**32,58**

Tabela 6.34 – Erro médio para seleção do melhor nível da Regressão Logística (*CMC*).

Regressão Logística		
Método	**Nível**	**Média**
NNTree	**1**	31,12
	2	**30,13**
RISKSEG	**1**	**28,72**
	2	29,33
SegTree	**1**	**32,72**
	2	32,72

6.8.3 – ANÁLISE DOS RESULTADOS

Observa-se na Tabela 6.35 que, a combinação de classificadores utilizando Redes Neurais, *Bagging*, *Boosting* e o *NNTree* obtiveram médias de erros menores que o classificador Simples, porém nenhum método melhora significantemente. Na Regressão Linear, os métodos *Boosting*, *RISKSEG* e *SegTree* apresentam médias de erros menores que o classificador de controle, porém só o *RISKSEG* foi significativamente menor. Ainda na técnica de Regressão Linear, o método *NNTree* teve uma piora significativa em relação ao controle. Na Regressão Logística, *Bagging*, *NNTree* e *RISKSEG* tiveram erros médios melhores que o controle, mas somente os dois últimos apresentaram melhoras significativas em relação ao método Simples. Apesar do *RISKSEG* possuir uma média menor, este resultado não foi significante quando comparado com o *NNTree*.

De maneira geral, como mostra a Tabela 6.35, as Redes Neurais conseguiram com o seu classificador Simples melhores resultados quando comparados com as Regressões de único classificador, mas o método *RISKSEG* conseguiu diminuir a diferença dos erros médios entre as técnicas de Regressão e Rede Neural.

Tabela 6.35 – Médias das taxas de erros e intervalos de confiança para a base *CMC*

Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
Redes Neurais	27,83 ±1,58	**26,41 ±1,65**	**26,61 ±0,79**	**27,52 ±2,31**	28,58 ±1,9	27,83 ±1,58
Reg. Linear	32,99 ±2,03	33,19 ±1,46	**32,86 ±1,76**	*36,15 ±1,2*	**28,99 ±1,47***	**32,58 ±1,92**
Reg. Logística	32,72 ±1,2	**32,58 ±1,93**	32,79 ±1,68	**30,13 ±1,75***	**28,72 ±1,85***	32,72 ±1,2

6.9 – ESTUDO DE CASO - *CONNECT4*

6.9.1 – DESCRIÇÃO DA BASE DE DADOS

Esta base de dados contém todas as possíveis posições do jogo *Connect4*, em que nenhum jogador ganhou ainda e no qual o próximo passo não é forçado. A variável alvo original que contém três classes (ganhar, perder, empate) foi transformada em ganhou (valor = 1) e não ganhou (valor = 0). Maiores informações sobre esta base podem ser obtidas no repositório da *UCI* [BlM10]. A Tabela 6.36 apresenta um resumo da base de dados.

Tabela 6.36 – Características gerais da base *Connect4.*

Quantidade de Variáveis	42 variáveis categóricas
Quantidade de Registros	67.557 registros
Distribuição do Alvo	44.473 registros com 1 (um) e 23.084 registros com 0 (zero)

6.9.2 – SELEÇÃO DOS PARÂMETROS

Como pode ser visto na Tabela 6.37, o problema apresenta melhores soluções com mais de três neurônios na camada intermediária. As médias dos métodos de combinação são relativamente próximas do classificador Simples e apenas o método *Boosting* apresentou uma diferença maior do que 1%.

Tabela 6.37 – Erro médio para seleção da melhor configuração da Rede Neural (*Connect4*)

Técnica	Nível	*Backpropagation*			*Levenberg-Marquardt*		
		3	10	20	3	10	20
Bagging	-	18,64	18,75	18,72	17,58	17,72	**17,55**
Boosting	-	19,15	**16,27**	16,75	18,91	19,24	18,90
Simples	-	19,37	**18,14**	18,34	19,96	20,01	19,87
NNTree	1	19,34	**18,21**	18,42	19,99	19,99	19,85
	2	19,32	18,27	18,46	20,02	19,98	19,86
RISKSEG	1	18,73	17,81	**17,61**	18,92	18,86	18,57
	2	19,09	18,15	18,10	18,98	18,61	18,85
SegTree	1	18,96	18,19	**18,10**	19,52	19,41	19,59
	2	19,12	18,19	18,10	19,52	19,41	19,59

Conforme é mostrado nas Tabelas 6.38 e 6.39, pode-se observar que a Regressão Linear precisou de dois níveis para atingir seus melhores resultados, enquanto que na Regressão Logística apenas o *RISKSEG* chegou ao segundo nível. Os resultados dos experimentos para as duas Regressões ficaram próximos dos 20% de erro médio, com exceção do método *NNTree* que apresentou seu melhor resultado igual a 25,50%.

Tabela 6.38 – Erro médio para seleção do melhor nível da Regressão Linear (*Connect4*).

Regressão Linear		
Método	Nível	Média
NNTree	**1**	26,81
	2	**25,50**
RISKSEG	**1**	20,59
	2	**20,25**
SegTree	**1**	20,56
	2	**20,53**

Tabela 6.39 – Erro médio p/ seleção do melhor nível da Regressão Logística (*Connect4*).

Regressão Logística		
Método	Nível	Média
NNTree	**1**	**20,99**
	2	21,00
RISKSEG	**1**	20,19
	2	**19,27**
SegTree	**1**	**20,28**
	2	20,42

6.9.3 – ANÁLISE DOS RESULTADOS

Conforme pode ser observado na Tabela 6.40, na combinação de classificadores utilizando Redes Neurais, apesar de quase todas as técnicas apresentarem médias de erros inferiores ao do classificador Simples, apenas a diferença do *Boosting* foi significativa. Nos experimentos com Regressão Linear, com exceção do *NNTree*, que apresentou piora significativa, todos os outros classificadores apresentaram melhor erro e apenas para o *Bagging* esta melhora não foi significativa, quando comparados com o classificador Simples. Ainda na análise com a Regressão Linear, entre o *RISKSEG* e o *SegTree* não houve diferença significativa, mas estes dois classificadores tiveram médias significativamente melhores em relação aos demais métodos de combinação. Na Regressão Logística, apenas o *RISKSEG* apresentou erro significativamente menor que o classificador de controle, apesar de outros classificadores apresentarem também médias de erros inferiores não significativas.

Tabela 6.40 – Médias das taxas de erros e intervalos de confiança para a base (*Connect4*).

Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
Redes Neurais	18,14 ±0,87	**17,55 ±0,85**	**16,27 ±0,62***	*18,21 ±0,91*	**17,61 ±1,29**	**18,1 ±0,83**
Reg. Linear	22,46 ±1,23	**22,06 ±1,4**	**21,89 ±1,2***	*25,5 ±1,59*	**20,25 ±0,82***	**20,53 ±0,82***
Reg. Logística	20,7 ±0,89	**20,52 ±1**	**20,48 ±0,91**	*20,99 ±0,81*	**19,27 ±0,66***	**20,28 ±0,52**

6.10 – ESTUDO DE CASO - *SOLAR FLARE*

6.10.1 – DESCRIÇÃO DA BASE DE DADOS

Esta base contém instâncias que representam características em uma região ativa do sol. O alvo contém três variáveis resposta que representam tipos de atividades solares. Para este estudo, foi gerada uma variável alvo dicotômica que identifica se existiu alguma atividade solar. Assim 1 (um) representa alguma atividade e 0 (zero) que não houve atividade. Maiores informações sobre esta base podem ser obtidas no repositório da *UCI* [BlM10]. A Tabela 6.41 apresenta um resumo da base de dados.

Tabela 6.41 – Características gerais da base *Solar Flare*.

Quantidade de Variáveis	10 variáveis categóricas
Quantidade de Registros	1.389 registros
Distribuição do Alvo	259 registros com 1 (um) e 1.130 registros com 0 (zero)

6.10.2 – SELEÇÃO DOS PARÂMETROS

Conforme demonstra a Tabela 6.42, não houve uma configuração predominante para Rede Neural e algumas configurações apresentam resultados iguais de erro de classificação para os dois algoritmos de treinamento da Rede Neural utilizados (*Backpropagation e Levenberg-Marquardt*). Isto indica que os dois algoritmos convergem para classificações semelhantes com o mesmo número de neurônios na camada intermediária.

Tabela 6.42 – Erro médio para seleção da melhor configuração da Rede Neural (*Solar)*.

Técnica	Nível	*Backpropagation*			*Levenberg-Marquardt*		
		3	10	20	3	10	20
Bagging	-	18,44	17,93	17,93	18,72	**17,86**	17,86
Boosting	-	19,59	18,43	**17,79**	19,59	18,43	17,79
Simples	-	19,44	**18,29**	18,58	19,44	18,29	18,58
NNTree	**1**	18,33	17,61	18,33	18,33	17,61	18,33
	2	18,14	**17,47**	18,26	18,14	17,47	18,26
RISKSEG	**1**	**17,64**	17,86	18,58	17,64	17,86	18,58
	2	18,00	18,01	18,15	18,00	18,01	18,15
SegTree	**1**	18,15	**17,72**	18,65	18,15	17,72	18,65
	2	18,79	17,93	18,44	18,79	17,93	18,44

Conforme mostram as Tabelas 6.43 e 6.44, o nível 1 foi o escolhido para todas as segmentações nas Regressões. Observa-se ainda que, para o método *NNTree*, não houve diferença de médias entre os níveis 1 e 2 das tabelas de erros.

Tabela 6.43 – Erro médio para seleção do melhor nível da Regressão Linear (*Solar)*.

Regressão Linear		
Método	**Nível**	**Média**
NNTree	**1**	**19,37**
	2	19,37
RISKSEG	**1**	**18,29**
	2	18,58
SegTree	**1**	**18,08**
	2	18,87

Tabela 6.44 – Erro médio para seleção do melhor nível da Regressão Logística (*Solar)*.

Regressão Logística		
Método	**Nível**	**Média**
NNTree	**1**	**18,29**
	2	18,29
RISKSEG	**1**	**18,37**
	2	18,58
SegTree	**1**	**18,37**
	2	18,73

6.10.3 – ANÁLISE DOS RESULTADOS

A Tabela 6.45 mostra que, apesar de muitos métodos obterem erros médios com valores menores do que o seu classificador de controle correspondente, nenhum conseguiu resultados significativos. Observa-se também que a diferença de médias entre os classificadores Simples das três técnicas são pequenas e não são significativamente diferentes.

Tabela 6.45 – Médias das taxas de erros e intervalos de confiança para a base (*Solar*).

Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
Redes Neurais	18,29 ±2,25	**17,86 ±2,24**	**17,79 ±2,82**	**17,47 ±2,52**	**17,64 ±1,98**	**17,72 ±2,65**
Reg. Linear	19,37 ±2,45	**18,72 ±2,1**	**19,01 ±2,13**	19,37 ±2,45	**18,29 ±2,53**	**18,08 ±2,79**
Reg. Logística	18,29 ±2,67	**18,08 ±2,89**	**17,86 ±2,74**	18,29 ±2,67	18,37 ±2,63	18,37 ±2,95

6.11 – ESTUDO DE CASO - *WINE QUALITY*

6.11.1 – DESCRIÇÃO DA BASE DE DADOS

Esta base é composta por dois conjuntos de registros, um sobre vinhos vermelhos e outro sobre vinhos brancos, ambos portugueses. Para este estudo de caso, a variável alvo, que contêm valores inteiros entre 3 (três) e 9 (nove), foi transformada em duas classes, uma para representar vinhos com maior qualidade (valores maiores que 5) e outra com qualidade mais baixa (valores menores ou iguais a 5). Estas novas classes receberam valores 1 (um) e 0 (zero), respectivamente. Também foi criada uma variável categórica que indica de o vinho é branco ou tinto. Maiores informações podem ser obtidas no repositório da *UCI* [BlM10] ou na referência [CCA09]. A Tabela 6.46 apresenta um resumo da base de dados.

Tabela 6.46 – Características gerais da base *Wine*.

Quantidade de Variáveis	11 variáveis numéricas e 1 (uma) categórica
Quantidade de Registros	6.497 registros (com a junção dos dois conjuntos de dados)
Distribuição do Alvo	2.384 registros com 0 (zero) e 4.113 registros com 1 (um)

6.11.2 – SELEÇÃO DOS PARÂMETROS

Observa-se, na Tabela 6.47, que houve muitos erros iguais entre o treinamento com *Backpropagation* e *Levenberg-Marquardt*, assim como aconteceu em outras bases. Para todos os métodos de combinação, três neurônios na camada intermediária foram suficientes para obter a melhor configuração, exceto para o *RISKSEG* que precisou de 10 neurônios. Apenas a segmentação utilizando *NNTree* obteve resultados melhores no primeiro nível, ainda assim este resultado é maior do que a melhor configuração do classificador Simples.

Tabela 6.47 – Erro médio para seleção da melhor configuração da Rede Neural (*Wine*).

Técnica	Nível	*Backpropagation*			*Levenberg-Marquardt*		
		3	10	20	3	10	20
Bagging	-	**24,18**	24,78	25,00	24,35	24,43	24,97
Boosting	-	**24,30**	25,40	25,70	24,30	25,40	25,70
Simples	-	**24,61**	25,47	25,67	24,61	25,47	25,67
NNTree	1	**26,00**	26,63	26,73	26,00	26,63	26,53
	2	26,57	27,44	27,42	26,57	27,44	27,10
RISKSEG	1	24,32	24,53	25,00	24,32	24,53	25,00
	2	24,13	**21,84**	24,83	24,13	22,14	24,83
SegTree	1	24,73	24,95	25,66	24,73	24,95	25,66
	2	**24,64**	24,64	25,26	24,64	24,64	25,26

Conforme demonstram as Tabelas 6.48 e 6.49 a maioria dos métodos de segmentação tiveram seus menores erros no segundo nível nas 2 (duas) técnicas analisadas, a exceção foi *NNTree* com Regressão Logística. A Regressão Linear com *NNTree* foi a que obteve o maior valor de erro médio, com valores muito diferentes dos demais métodos.

Tabela 6.48 – Erro médio para seleção do melhor nível da Regressão Linear (*Wine*).

Regressão Linear		
Método	**Nível**	**Média**
NNTree	**1**	35,07
	2	**32,80**
RiskSeg	**1**	25,40
	2	**24,87**
SegTree	**1**	26,27
	2	**25,58**

Tabela 6.49 – Erro médio para seleção do melhor nível da Regressão Logística (*Wine*).

Regressão Logística		
Método	**Nível**	**Média**
NNTree	**1**	**27,44**
	2	28,20
RiskSeg	**1**	25,64
	2	**25,46**
SegTree	**1**	26,03
	2	**25,72**

6.11.3 – ANÁLISE DOS RESULTADOS

Na Tabela 6.50 pode-se observar que, os métodos de combinação utilizando Regressão Logística não tiveram suas taxas de erro diminuídas em relação aos seus classificadores Simples. Destaque para o *RISKSEG* que conseguiu melhorar os resultados significativamente em relação ao classificador Simples para Redes Neurais e Regressão Linear. Na Regressão Linear, os métodos *Bagging* e *SegTree* também conseguiram uma melhora em relação ao classificador Simples. O método *NNTree* só obteve resultados piores (significativos) quando comparado com o classificador Simples de todas as técnicas. Na Regressão Linear, quando compara-se *Bagging* e *SegTree* com o *RISKSEG* não é verificada uma diferença significativa.

Tabela 6.50 – Médias das taxas de erros e intervalos de confiança para a base *Wine*.

Técnica	Simples	Bagging	Boosting	NNTree	RiskSeg	SegTree
Redes Neurais	24,61 ±1,12	**24,18 ±0,9**	**24,3 ±1**	*26 ±1,11*	**21,84 ±0,56***	24,64 ±1,07
Reg. Linear	27,67 ±1,44	**26,92 ±1,16***	**27,09 ±1,14**	*32,8 ±1,4*	**24,87 ±1,29***	**25,58 ±1,11***
Reg. Logística	25,98 ±1,48	**25,87 ±1,42**	**25,92 ±1,44**	*27,44 ±1,1*	**25,46 ±0,82**	**25,72 ±1,46**

6.12 – ESTUDO DE CASO - *ADULT*

6.12.1 – DESCRIÇÃO DA BASE DE DADOS

Este estudo de caso utiliza uma base de dados do Census de 1994. Um conjunto de dados razoavelmente limpos foram extraídos para determinar se uma pessoa ganha mais que 50.000 mil dólares por ano. A variável alvo recebe valor igual 1 (um) se a pessoas ganha mais de U$50.000 por ano e 0 (zero) se ganha um valor menor ou igual a esta quantia. Esta base possui valores ausentes em suas variáveis. Algum detalhamento sobre as variáveis preditivas é

fornecido pelo repositório *UCI* [BlM10]. Na Tabela 6.51 é apresentado um resumo da base de dados.

Tabela 6.51 – Características gerais da base *Adult*.

Quantidade de Variáveis	6 numéricas e 8 categóricas
Quantidade de Registros	48.842 registros
Distribuição do Alvo	37.155 registros com 0 (zero) e 11.687 registros com 1 (um)

6.12.2 – SELEÇÃO DOS PARÂMETROS

Conforme demonstra a Tabela 6.52, não houve uma configuração predominante para Rede Neural e algumas configurações apresentam resultados iguais de erro de classificação para os dois algoritmos de treinamento da Rede Neural utilizados (*Backpropagation e Levenberg-Marquardt*). Isto indica que os dois algoritmos convergem para classificações semelhantes com o mesmo número de neurônios na camada intermediária. Nenhuma das configurações escolhidas continha 20 neurônios na camada intermediária. Quanto à quantidade de níveis máximos de profundidade das segmentações, os métodos *RISKSEG* e *SegTree* precisaram de apenas 1 (um) nível para atingir seu melhor desempenho, enquanto que o método *NNTree* precisou de 2 (dois) níveis.

Tabela 6.52 – Erro médio para seleção da melhor configuração da Rede Neural (*Adult)*.

Técnica	Nível	*Backpropagation*			*Levenberg-Marquardt*		
		3	10	20	3	10	20
Bagging	-	14,42	14,20	14,98	14,79	**14,17**	15,18
Boosting	-	14,51	**14,33**	15,19	15,10	14,34	15,08
Simples	-	**14,89**	15,76	15,74	15,18	15,71	15,80
NNTree	1	14,14	14,60	14,35	14,14	14,51	14,25
	2	13,84	13,88	13,68	**13,60**	13,87	13,69
RISKSEG	1	**15,07**	15,50	15,94	15,23	15,68	16,05
	2	15,07	15,56	15,94	15,42	15,66	16,05
SegTree	1	**14,71**	15,41	15,80	15,34	15,26	16,04
	2	14,76	15,52	15,88	15,37	15,48	15,97

Nas Tabelas 6.53 e 6.54 pode-se observar que não houve predominância de um valor para o nível máximo de profundidade da árvore de segmentação. Apenas o método *NNTree* obteve um ganho de quase 1% do nível 1 para o nível 2, nas demais técnicas não houve ganho ou ele foi menor do que 0,25%. Para a Regressão Logística, praticamente não houve diferença de médias de erro de classificação entre o nível 1 e o nível 2 para os três métodos de combinação.

Tabela 6.53 – Erro médio para seleção do melhor nível da Regressão Linear (*Adult*).

Regressão Linear		
Método	**Nível**	**Média**
NNTree	**1**	14,62
	2	**13,68**
RISKSEG	**1**	**16,73**
	2	16,75
SegTree	**1**	16,03
	2	**15,81**

Tabela 6.54 – Erro médio para seleção do melhor nível da Regressão Logística (*Adult*).

Regressão Logística		
Método	**Nível**	**Média**
NNTree	**1**	**14,58**
	2	14,58
RISKSEG	**1**	14,97
	2	**14,96**
SegTree	**1**	**14,96**
	2	14,96

6.12.3 – ANÁLISE DOS RESULTADOS

Observa-se na Tabela 6.55, que o método *NNTree* conseguiu resultados significativamente melhores do que o classificador Simples, utilizando Redes Neurais e Regressão Linear. O método *SegTree* também conseguiu melhora significativa em relação ao classificador base na Regressão Linear. Nesta última técnica, o *RISKSEG* conseguiu média de erro mais baixa que o classificador Simples, mas não foi significativa.

Tabela 6.55 – Médias das taxas de erros e intervalos de confiança para a base (*Adult*).

Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
Redes Neurais	14,89 ±0,91	**14,17 ±0,37**	**14,33 ±0,28**	**13,6 ±1,33***	15,07 ±0,86	**14,71 ±0,96**
Reg. Linear	17,38 ±1,23	**17,28 ±1,05**	17,8 ±1,04	**13,68 ±0,83***	**16,73 ±1,25**	**15,81 ±0,78***
Reg. Logística	14,9 ±1	**14,76 ±0,93**	**14,82 ±0,9**	**14,58 ±0,97**	14,96 ±0,82	14,96 ±0,97

6.13 – ESTUDO DE CASO - *SPAMBASE*

6.13.1 – DESCRIÇÃO DA BASE DE DADOS

Esta base tem o objetivo de construir um filtro de spam personalizado de emails. Ela contém as características de emails que foram considerados spam e emails que não foram. A variável alvo tem valor 1 (um) quando o email é considerados spam e valor 0 (zero) caso contrário. Maiores informações sobre esta base podem ser obtidas no repositório da *UCI* [BlM10]. A Tabela 6.56 apresenta um resumo da base de dados.

Tabela 6.56 – Características gerais da base *Spambase*.

Quantidade de Variáveis	57 variáveis numéricas
Quantidade de Registros	4.601 registros
Distribuição do Alvo	1.813 registros com 1 (um) e 2.788 registros com 0 (zero)

6.13.2 – SELEÇÃO DOS PARÂMETROS

Como pode ser observado na Tabela 6.57, para obter o menor erro, apenas o *Bagging* precisou de 10 neurônios na camada intermediária e utilizou *Levenberg-Marquardt*, os demais métodos

utilizaram três neurônios e *Backpropagation.* Todos os métodos de segmentação precisaram de dois níveis para atingir seu menor erro de classificação.

Tabela 6.57 – Erro médio para seleção da melhor configuração da Rede Neural (*Spambase)*.

Método	Nível	*Backpropagation*			*Levenberg-Marquardt*		
		3	10	20	3	10	20
Bagging	-	5,95	5,48	5,56	5,52	**5,23**	5,64
Boosting	-	**6,65**	6,95	7,09	7,32	6,90	7,35
Simples	-	**7,06**	8,00	7,98	7,41	8,03	7,90
NNTree	1	6,89	7,17	7,13	6,99	7,07	7,01
	2	**6,81**	6,99	7,00	6,88	7,00	6,95
RISKSEG	1	7,67	7,54	7,87	7,69	8,03	7,52
	2	**7,45**	7,80	7,63	7,50	8,03	7,50
SegTree	1	7,22	7,74	7,54	7,19	7,53	7,51
	2	**7,15**	7,67	7,26	7,26	7,61	7,25

Nas Tabelas 6.58 e 6.59 pode-se observar que houve predominância de um nível como valor máximo de profundidade da árvore de segmentação. Apenas o método *NNTree* com Regressão Linear obteve seu melhor resultado no nível 2.

Tabela 6.58 – Erro médio para seleção do melhor nível da Regressão Linear (*Spambase)*.

Regressão Linear		
Método	**Nível**	**Média**
NNTree	**1**	7,04
	2	**6,45**
RiskSeg	**1**	**10,61**
	2	11,34
SegTree	**1**	**9,02**
	2	9,37

Tabela 6.59 – Erro médio para seleção do melhor nível Regressão Logística (*Spambase)*.

Regressão Logística		
Método	**Nível**	**Média**
NNTree	**1**	**6,76**
	2	6,94
RiskSeg	**1**	**8,19**
	2	8,43
SegTree	**1**	**8,67**
	2	8,69

6.13.3 – ANÁLISE DOS RESULTADOS

A Tabela 6.60 mostra que, para Redes Neurais, apenas *Bagging* e *Boosting* conseguiram resultados significativamente melhores do que o classificador Simples. Utilizando Regressão Logística, todos conseguiram médias menores que o classificador Simples e apenas o *Bagging* não foi significantemente melhor. Também na Regressão Linear, todos os métodos apresentaram médias de erro menores do que o classificador Simples, mas apenas o método *NNTree* conseguiu ser significantemente melhor.

Tabela 6.60 – Médias das taxas de erros e intervalos de confiança para a base (*Spambase)*.

Técnica	Simples	*Bagging*	*Boosting*	*NNTree*	*RISKSEG*	*SegTree*
Redes Neurais	7,06 ±0,6	**5,23 ±0,66***	**6,65 ±0,57***	**6,81 ±0,57**	7,45 ±0,71	7,15 ±0,91
Reg. Linear	24,36 ±6,83	**21,52 ±4,09**	**12,69 ±2,06***	**6,45 ±0,61***	**10,61 ±0,75***	**9,02 ±0,92***
Reg. Logística	8,69 ±0,63	**8,41 ±0,74**	**8,43 ±0,81**	**6,76 ±0,66***	**8,19 ±0,76**	**8,67 ±0,58**

6.14 - RESUMO GERAL DOS RESULTADOS E TESTES COMPLEMENTARES

Na Tabela 6.61 tem-se uma análise resumida do desempenho do *RISKSEG*. Esta análise é baseada em três informações para cada técnica: a quantidade de métodos que foram significativamente superiores ao classificador de controle (Simples), um indicador se o *RISKSEG* foi uma destas técnicas superiores e a sua posição em relação aos demais. Para este último caso foi feito o teste *t-student* para amostras pareadas, com nível de significância de 5%, a fim de comparar os demais métodos com o *RISKSEG*, ou seja, se não houver uma diferença significativa entre os métodos eles terão a mesma posição (*rank*). Um asterisco é colocado ao lado do valor da posição pra indicar que ele está sozinho nesta posição.

Pode-se observar, na Tabela 6.61, que, em boa parte das bases reais do repositório *UCI* testadas, o método *RISKSEG* obteve resultados superiores aos obtidos pelas outras técnicas experimentadas, inclusive com a utilização de Redes Neurais, que é, das técnicas utilizadas, a mais apta a capturar comportamentos não lineares nas bases de dados [HSW89] e a que obteve os melhores resultados entre os experimentos que utilizaram apenas um classificador.

Tabela 6.61 – Resultados gerais das bases da *UCI*.

Base	Redes Neurais			Regressão Linear			Regressão Logística		
	Qtde	*RISKSEG*?	Posição	Qtde	*RISKSEG*?	Posição	Qtde	*RISKSEG*?	Posição
Abalone	**0**	**-**	**-**	**4**	**Sim**	**1**	**0**	**-**	**-**
Adult	**1**	**Não**	**-**	**2**	**Não**	**-**	**0**	**-**	**-**
Car	**1**	**Sim**	**1***	**3**	**Sim**	**1**	**1**	**Sim**	**1***
Chess	**3**	**Sim**	**1***	**4**	**Sim**	**2**	**1**	**Sim**	**1***
Connect4	**1**	**Não**	**-**	**3**	**Sim**	**1**	**1**	**Sim**	**1***
CMC	**0**	**-**	**-**	**1**	**Sim**	**1***	**2**	**Sim**	**1**
German	**0**	**-**	**-**	**1**	**Não**	**-**	**0**	**-**	**-**
Magic	**2**	**Sim**	**1***	**3**	**Sim**	**1***	**3**	**Sim**	**1***
Mushroom	**3**	**Sim**	**1**	**0**	**-**	**-**	**0**	**-**	**-**
Solar Flare	**0**	**-**	**-**	**0**	**-**	**-**	**0**	**-**	**-**
Spambase	**2**	**Não**	**-**	**4**	**Sim**	**3**	**1**	**Não**	**-**
Wine	**1**	**Sim**	**1***	**3**	**Sim**	**1**	0	-	-

Uma tendência geral dos experimentos foi a melhoria dos resultados obtidos pelos métodos de combinação, quando comparados com o classificador de controle, principalmente usando Regressão Linear. Especialmente para esta última técnica, e em situações especiais quando existam limitações ou mesmo uma razão específica para seu uso (exemplo, o cliente exige que seja esta técnica), a combinação de classificadores seria uma alternativa viável para melhoria do desempenho, em especial o *RISKSEG*, que apresentou, nas bases deste capítulo, um número maior de melhores resultados, quando comparados com os demais métodos de combinação.

Observando a Tabela 6.61, o *RISKSEG* com Redes Neurais, das oito bases que obtiveram métodos de combinação com resultados melhores que o classificador Simples, ele obteve cinco primeiras posições e, em quatro destas, ele foi melhor sozinho em relação aos demais (ver asteriscos da coluna posição). Na Regressão Linear, por ser um método considerado mais fraco [Kia03], o *RISKSEG* conseguiu sua maior quantidade de melhores resultados, quando comparado com o classificador de controle (Simples), pois ele participa de oito das dez bases em que houve melhoria com métodos de combinação e obteve seis primeiras posições. Porém, nesta técnica, ele obteve o menor número de bases em que ele foi o melhor e único, com apenas duas ocorrências. Na Regressão Logística, o *RISKSEG* também apresentou um bom desempenho, pois obteve resultados melhores em cinco das seis bases em que houve métodos que superaram o classificador Simples e foi o primeiro isolado em quatro delas quando comparado com os demais métodos.

Na Tabela 6.62 são resumidas as bases de dados do capítulo fornecendo algumas informações sobre a quantidade de registros e quantidade de variáveis. Nesta tabela pode-se ver diversos tamanhos de bases e quantidade de variáveis de entrada. Para efeito de comparação, foi criada uma coluna (Equivalência 1 de *n*) que contém o total de categorias das variáveis categóricas e das numéricas categorizadas (esta última utilizada apenas no processo de segmentação) de cada base de dados. São mostradas colunas marcadas com "X" quando o *RISKSEG* obtém médias significativamente melhores que o classificador Simples.

Tabela 6.62 – Avaliação das quantidades de variáveis das Bases da *UCI*.

Base	Quantidade da Base					Média de Categorias		Sucesso (com Simples)		
	Registros	Variáveis	Numéricas	Categóricas	Equivalência 1 de *n*	Numéricas	Categóricas	RN	R. Lin	R. Log
Abalone	4.177	8	7	1	31	4,0	3,0		X	
Adult	48.842	14	6	8	122	3,2	12,9			
Car	1.728	6	0	6	21	0,0	3,5	X	X	X
Chess	3.196	36	0	36	73	0,0	2,0	X	X	X
CMC	1.473	9	6	3	30	4,0	2,0		X	X
Connect	67.557	42	0	42	126	0,0	3,0		X	X
German	1.000	20	7	13	80	4,0	4,0			
Magic	19.020	10	10	0	40	4,0	0,0	X	X	X
Mushroom	8.124	22	0	22	116	0,0	5,5	X		
Solar Flare	1.389	10	0	10	32	0,0	3,2			
Spambase	4.601	57	57	0	132	2,3	0,0		X	
Wine	6.497	12	11	1	46	4,0	2,0	X	X	

Observando-se a Tabela 6.62 não se observa nenhuma tendência do *RISKSEG* e seus resultados com relação à quantidade de variáveis ou de registros das bases de dados, pois o mesmo apresenta bons resultados para os mais diversos tamanhos de bases e quantidade de

registros. O desempenho deste método está mais ligado às características da base de dados, à relação entre as variáveis preditoras, e à técnica utilizada.

Os tempos de treinamento do método *RISKSEG* e do método exaustivo estão diretamente ligados à quantidade de variáveis e registros da base de dados, mas o tempo de treinamento do método exaustivo mostrou-se maior quanto comparado com o RISKSEG. Na Tabela 6.63 tem-se uma comparação de tempo entre o método *RISKSEG* e uma implementação de um método exaustivo que testa todas as combinações para escolher a melhor divisão dos conjuntos para treinamento do nível subsequente. Com Redes Neurais, observa-se que, para todas as bases, o *RISKSEG* obteve tempos menores, que variaram de aproximadamente 2,15 a quase 17 vezes mais rápido. Para os testes com Redes Neurais utilizou-se a mesma forma experimental e configurações já descritas neste capítulo para o *RISKSEG*.

Apesar de não apresentado aqui, na comparação dos tempos, também foram experimentados treinamentos com 10 e 20 neurônios na camada intermediária e os tempos de treinamento mostraram-se muitos lentos (em algumas bases) chegando a ser 100 vezes mais lento. Isto acontece porque, quando aumenta-se o número de neurônios, o tempo de treinamento também aumenta e, como o método *RISKSEG* utiliza menos treinamento de preditores para chegar ao seu melhor resultado, ele tende a ter um menor tempo.

Tabela 6.63 – Médias consolidadas do tempo de treinamento em segundos das bases da *UCI*.

Base	Redes Neurais		Reg. Linear		Reg. Logística	
	Exaustivo	*RISKSEG*	Exaustivo	*RISKSEG*	Exaustivo	*RISKSEG*
Abalone	221	37	47	15	55	16
Adult	1450	247	72	199	596	197
Car	80	23	11	9	52	16
Chess	229	39	29	30	309	55
Contraceptive	79	13	41	10	56	14
Connect4	3046	322	135	184	424	265
German	142	13	88	11	199	17
Magic	1002	139	77	18	79	21
Mushroom	447	208	18	98	140	130
Solar Flare	102	16	35	7	60	10
Spambase	3307	195	174	57	1105	209
Wine Quality	520	63	74	15	90	20

Ainda de acordo com a Tabela 6.63, pode-se observar que, para todas as bases, utilizando Regressão Logística, o tempo do *RISKSEG* foi inferior ao método exaustivo, variando de 1,6 a 11,7 vezes mais rápido. Na Regressão Linear, quatro bases de dados (*Adult, Chess, Connect4 e Mushroom*) do método exaustivo se mostraram mais rápidas do que com o método *RISKSEG*, ou seja, o tempo para rodar o método de seleção de variáveis do *RISKSEG* é mais custoso do

que experimentar todas as combinações para dividir o nó. Desta forma, observa-se que o método RISKSEG é mais rápido do que a utilização de uma busca exaustiva na maioria das bases e técnicas utilizadas. Pode-se concluir que a busca exaustiva é agravada com técnicas mais lentas.

A Tabela 6.64 apresenta as médias dos erros da comparação entre o método *RISKSEG* e o método de busca exaustiva. Os valores em negrito representam a menor média de erro na comparação entre os métodos e o asterisco informa que o método exaustivo foi significativamente melhor. A metodologia para escolha dos parâmetros foi a mesma abordada nos experimentos anteriores iniciais do capítulo e os resultados finais são apresentados na tabela de erros.

Tabela 6.64 – Comparação das médias dos erros e intervalos de confiança das bases da *UCI*.

Base	Redes Neurais		Reg. Linear		Reg. Logística	
	RiskSeg	*Exaustivo*	RiskSeg	*Exaustivo*	RiskSeg	*Exaustivo*
Abalone	**20,15 ±0,83**	20.35 ±0,94	21,07 ±0,99	**20.95 ±0,85**	20,66 ±1,1	**20.64 ±0,84**
Adult	**15,07 ±0,86**	16,2 ±1,12	16,73 ±1,25	**16,7 ±0,68**	**14,96 ±0,82**	16,3 ±0,8
Car	0,23 ±0,18	0,23 ±0,19	3,88 ±0,92	**2,78 ±0,58**	**1,85 ±0,78**	2,18 ±0,71
Chess	**0,35 ±0,14**	1,19 ±0,36	3,25 ±0,77	**2,6 ±0,31***	**1,02 ±0,27**	1,06 ±0,31
Connect4	17,61 ±1,29	**16,63 ±0,68**	20,25 ±0,82	**20,1 ±0,85**	**19,27 ±0,66**	19,53 ±0,94
CMC	**28,58 ±1,9**	28,65 ±2,27	**28,99 ±1,47**	30,14 ±1,27	**28,72 ±1,85**	30,48 ±1,77
German	25,3 ±1,84	**25,1 ±2,07**	26,3 ±2	**25,4 ±2,02**	26,3 ±2,51	**25,2 ±1,81**
Magic	13,51 ±0,58	**13,42 ±0,62**	16,06 ±0,65	**15,91 ±0,58**	**15,4 ±0,51**	15,54 ±0,65
Mushroom	0,01 ±0,02	**0 ±0**	0 ±0	0 ±0	0 ±0	0 ±0
Solar Flare	17,64 ±1,98	**17,51 ±1,93**	**18,29 ±2,53**	19,01 ±2,9	18,37 ±2,63	18,37 ±3,1
Spambase	7,45 ±0,71	**6,68 ±0,82**	10,61 ±0,75	**8,76 ±0,94***	8,19 ±0,76	**8,15 ±0,89**
Wine	**21,84 ±0,56**	24,61 ±1,12	**24,87 ±1,29**	27,67 ±1,44	**25,46 ±0,82**	25,98 ±1,48

Ainda na Tabela 6.64 pode-se observar que o método exaustivo obteve menores e significativas médias de erros apenas em duas bases na Regressão Linear (*Chess e Spambase*). O RISKSEG também obteve experimentos com melhora significativa em relação ao método exaustivo, como por exemplo, nas bases *Wine* e *Chess* com Redes Neurais e na base *Wine* com Regressão Linear. A partir destes resultados pode-se concluir que o método exaustivo obteve resultados (erros) bem semelhantes ao método *RISKSEG*, sendo melhor significativamente apenas para Regressão Linear em duas bases. Como o tempo de treinamento do método exaustivo pode ser muito maior, principalmente em técnicas com tempo de treinamento mais lento, como Redes Neurais com muitos neurônios na camada intermediária, recomenda-se utilizá-lo apenas em situações em que se possa esperar mais pelo tempo de treinamento. Ainda analisando os resultados das Tabelas 6.63 e 6.64, o método exaustivo é recomendado principalmente para Regressão Linear, pois o tempo de treinamento e o erro podem ser menores

do que quando se utiliza o *RISKSEG*. Entretanto, mesmo para Regressão Linear, também sugere-se a utilização do *RISKSEG*, pois em alguns casos ele pode apresentar menores erros de classificação.

A configuração de software e hardware utilizada nos experimentos para medição do tempo é mostrada na Tabela 6.65. Atenção para a utilização de um software para acelerar o processamento de *I/O*, com a criação de um drive virtual, pois sem ele os tempos seriam ainda maiores para as duas técnicas e talvez nem fosse possível terminar em tempo hábil todos os experimentos dos Capítulos 5 e 6.

Tabela 6.65 – Configuração Principal de Hardware e Software das Medições de Tempo.

Processador	Intel® Xeon® Processor X5260 (6Mb Cache, 3.33 GHz, 1333 MHz FSB)
Memória Ram	4GB DDR2 (3.25 Gb disponível)
Sistema Operacional	Windows XP - Service Pack 3
Software de Disco virtual em RAM	Dataram RAMDisk - Versão 3.5.130RC13a
Software das Técnicas	SAS Base/Stat v9.1.3 SP 4 com SAS Enterprise Miner v4.0
Hard Disk	SATA II 500 Mb (200 Mb disponível)

CAPÍTULO 7

APLICAÇÕES EM BASES REAIS

Neste capítulo é analisada e experimentada uma base de dados de risco de crédito e outra de risco de fraude, cedidas pelo *bureau* de informação *Serasa Experian*. Neste capítulo, o objetivo não é realizar comparações com diversos tipos de técnicas de combinação e nem diferentes classificadores com diversos parâmetros. O principal foco é demonstrar a aplicação do método *RISKSEG* em problemas reais que podem ser facilmente encontrados em empresas de gerenciamento de risco de crédito e fraude. Outro objetivo é a experimentação da medida *ROC* como métrica de otimização do *RISKSEG*, ao invés de somente o erro de classificação, como nos Capítulos 5 e 6.

7. 1 – DESCRIÇÃO GERAL DAS APLICAÇÕES

A idéia básica é observar o comportamento do método em aplicações reais. Para isto utiliza-se uma base de dados cedida pela *Serasa Experian*, que é a empresa possuidora do maior *bureau* de informações da América Latina. Maiores informações sobre esta organização podem ser obtidas no endereço eletrônico: www.serasa.com.br. As simulações foram feitas com informações de pessoas físicas consultadas neste *bureau*. Apesar dos experimentos serem apenas simulações, os resultados obtidos no seu desenvolvimento permitiram que os modelos aqui estudados fossem aplicados em empresas que trabalham com operações de risco de crédito para pessoa física. Os dados foram cedidos de maneira que as pessoas não possam ser identificadas, a fim de preservar suas identidades e manter o sigilo de informações. Todo e qualquer atributo (CPF e RG, por exemplo), que possa identificar diretamente um registro, foi retirado.

Em uma das aplicações, também se detalha os principais passos e resultados intermediários do método proposto. Assim, uma das árvores de modelos é escolhida para ser

montada, passo a passo. Desta forma, têm-se uma idéia mais clara de como a árvore de modelos, obtida pelo *RISKSEG*, deve ser distribuída em ambiente de produção.

Além de mostrar aplicações do método em problemas reais, também testa-se o efeito da otimização da segmentação pela métrica *ROC* Mínimo. Esta não é uma métrica normalmente utilizada no treinamento de modelos preditivos. As medidas *KS2* e *ROC* são muito empregadas para mensuração e comparação de aplicações de risco de crédito e fraude [ThC02] [May04]. Por esta razão, e também para demonstrar uma das características importantes proposta pelo *RISKSEG*, utilizou-se o *ROC* Mínimo também como medida para otimização dos experimentos. A otimização por esta métrica foi comparada com a otimização via erro de classificação.

A seguir, é dada uma pequena explanação sobre modelos para mensuração de risco de crédito e detecção de fraude, permitindo assim a familiarização com estes tipos de problemas, antes do detalhamento das aplicações.

7.1.1 – MODELOS DE RISCOS DE CRÉDITO

Os modelos de classificação e regressão aplicados às instituições financeiras, em especial na concessão de crédito, têm ganhado uma importância notável nos últimos anos. Atualmente, não existe grande ou mesmo média instituição que conceda crédito, incluindo lojas de varejo e empresas de telefonia, que não possua modelos para verificação dos riscos associados às transações realizadas. A "popularização" de modelos preditores, aliada aos bons resultados com a inserção deles nos processo de decisão de crédito, fez com que as empresas compreendessem o quão importante eles são.

O risco está presente em qualquer tipo de operação do mercado financeiro: quando um banco empresta dinheiro a um cliente, acreditando que ele devolverá com o acréscimo de juros; quando um investidor compra papéis de uma determinada empresa acreditando que eles valorizarão dentro de certo período de tempo; quando uma seguradora cobra um determinado prêmio por uma apólice de seguro, dando mais tranquilidade ao seu cliente; etc.

A estimação de riscos é uma das principais ferramentas para uso em gestão de crédito. A aplicação do método proposto teve como um de seus principais fatores motivadores o seu uso em modelos de risco de crédito. Este risco [ThC02] [May04] está relacionado a possíveis perdas quando o contratante não honra seu compromisso, ou seja, os recursos não serão mais recebidos. Por exemplo, um empréstimo é feito a um cliente que, por razões alheias, deixa de pagar as parcelas da dívida e, portanto, torna-se inadimplente. No mercado financeiro este é o mais comum dos riscos.

Estudos acadêmicos sobre modelagem quantitativa do risco de crédito foram iniciados na década de 30 com modelos univariados [Sil02] e evoluíram para modelos multivariados a partir do desenvolvimento do modelo Escore-Z [Alt68]. A modelagem de crédito consiste basicamente em "comparar" um proponente e suas informações cadastrais com outros clientes, que sejam "semelhantes" a ele no momento da abertura do contrato de crédito. É importante que as propostas utilizadas para esta comparação já possuam uma clara definição de resultado, por exemplo, se o cliente efetuou o pagamento ou não, pois somente desta forma pode-se estimar o risco desta nova solicitação.

7.1.2 – MODELOS PARA DETECÇÃO DE FRAUDE (*FRAUD SCORING*)

A fraude se tornou um grande problema para as empresas e tem sido cada vez mais difícil de ser detectada. É por isso que, derivado dos modelos de risco de crédito, novos modelos estatísticos e de inteligência artificial, normalmente com o uso de Redes Neurais, vêm sendo desenvolvidos para este tipo de finalidade. O *Fraud Scoring* é um modelo de pontuação de risco baseado em modelos estatísticos e de Inteligência Artificial. A pontuação analisa pedidos de crédito fraudulentos e não fraudulentos, para tentar determinar comportamentos de características previamente tidas confirmadas de fraude e assim determinar o risco de fraude para pedidos de créditos.

Apesar do sucesso na utilização dessa tecnologia, é importante salientar que os modelos para detecção de fraudes (*Fraud Scorings*) são apenas um componente de uma solução de controle e prevenção à fraude e devem ser empregados em conjunto com outras tecnologias, pois ela isoladamente não é uma solução antifraude completa. O emprego de outras técnicas complementares ao modelo de risco se justifica, principalmente, por uma de suas limitações, que é a incapacidade de reagir às mudanças bruscas de mercado que não tenham sido ainda refletidas no histórico dos dados utilizados para o treinamento. Para mitigar essa deficiência, normalmente são colocados, em paralelo, um ou mais modelos baseados em regras, onde os especialistas cadastram possíveis mudanças ainda não aprendidas pelo modelo. O emprego de modelos baseados em regras é necessário, mesmo fazendo-se treinamentos em curtos espaços de tempo, pois o volume de dados de determinada característica de fraude pode não ser suficientemente grande para sensibilizar ou determinar um padrão em um modelo de risco. Então, no processo de definição e revisão periódica das regras, ainda é de fundamental importância a intervenção do especialista em fraudes [San07].

7.2 – PARÂMETROS E EXPERIMENTOS

A escolha de Regressão Logística foi motivada pelo fato de ser uma das técnicas mais utilizadas para desenvolvimento de modelos de risco de crédito e fraude. Muitos trabalhos descrevem a Regressão Logística como uma das técnicas estatísticas mais utilizadas para classificação binária e das mais adequadas para problemas de escore para Crédito e Fraude [AVA04] [OHT05].

7.2.1 – PARÂMETROS UTILIZADOS

Não foi necessária uma metodologia para seleção dos parâmetros da Regressão Logística, visto que esta técnica não necessita de definição de parâmetros, salvo se estiver utilizando algum método redutor de variáveis, o que não é o caso. Em relação aos parâmetros do método *RISKSEG*, foram utilizados os mesmos do Capítulo 5. A única diferença foi o número mínimo de elementos em um nó, definido aqui como 10%, após alguns testes preliminares para a definição mais adequada a cada base.

Como a análise dos resultados deste capítulo já envolve todos os níveis de profundidade da árvore de modelos do *RISKSEG*, não foi necessária a etapa de pré-seleção do melhor nível.

7.2.2 – TRATAMENTO DAS VARIÁVEIS

As variáveis originalmente categóricas não sofreram qualquer tipo de agrupamento de categorias. Sendo assim, os seus valores originais foram utilizados para a codificação binária. A transformação binária utilizada foi a padrão do software SAS Enterprise Miner versão 4.0, que transforma a variável em *n* variáveis binárias com valor 0 (zero) ou 1 (um), onde *n* é o número de categorias da variável [SAS02].

Para treinamento dos preditores, todas as variáveis numéricas deste capítulo foram normalizadas com valores contínuos entre 0 (zero) e (um). Como, internamente, as implementações das segmentações utilizadas exigem que as variáveis sejam categóricas para dividir os subconjuntos, optou-se pela categorização das variáveis numéricas manualmente de modo a possuírem quatro categorias, cada uma com aproximadamente 25% dos elementos. Nos experimentos, esta categorização foi utilizada somente para a segmentação, pois os valores numéricos e normalizados foram utilizados nos treinamentos das Regressões Logísticas.

7.2.3 – ORGANIZAÇÃO DOS EXPERIMENTOS

Para estes experimentos também foi utilizada a técnica *k-fold Cross Validation*, com os mesmos parâmetros utilizados no capítulo anterior. Como mostra a Figura 7.1, os conjuntos de

treinamento e validação foram definidos para serem 65% e 35% do total da base, respectivamente, e, portanto, eles representam 72,2% e 27,8% dos registros disponíveis para estes fins, respectivamente.

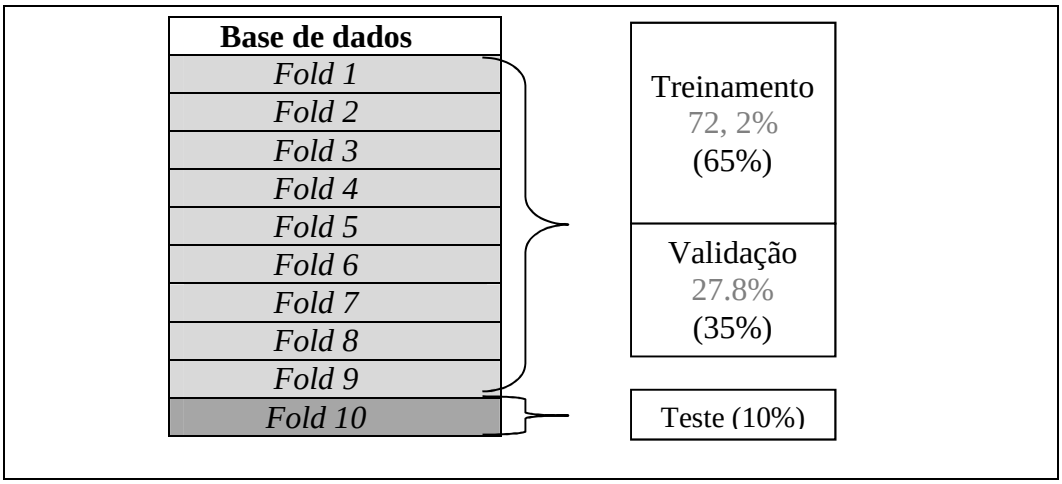


Figura 7.1 – Estrutura dos conjuntos aplicada em cada base.

Conforme comentado anteriormente, para as bases deste capítulo foi utilizada apenas a técnica de Regressão Logística. Os resultados de todas as bases foram mensurados através da taxa de erro médio de classificação e *ROC Min*.

7.3 – APLICAÇÃO – CONCESSÃO DE CRÉDITO

Para aplicações de risco, normalmente tem-se o problema modelado através de uma saída dicotômica – com duas classes. Desta forma, a resposta correspondente a cada classe pode ser conjugada por meio de uma transformação linear de uma única grandeza escalar, o escore. Esta resposta (escore) está definida no domínio contínuo de *x* a *n* (normalmente entre 0 e 100), com o *x* representando o caso mais negativo (mau cliente, por exemplo) e o *n* significando o mais positivo (bom cliente, por exemplo). Essa representação da resposta por uma grandeza escalar possibilita um monitoramento mais refinado do desempenho do modelo.

O objetivo primário de modelos de avaliação de riscos de crédito é a classificação do risco de cada nova solicitação de crédito, desta forma, pode-se saber o risco de inadimplência do cliente consultado nos próximos *y* meses. Para este estudo de caso, estabeleceu-se que seria mensurado o risco do cliente consultado ter alguma anotação negativa (deixar de pagar dívidas) no *bureau* em até 12 meses após ser consultado por alguma empresa. De forma mais explicativa, o alvo dicotômico a ser modelado, é: *Bom*, para os que não tiveram nenhuma entrada negativa no *bureau* em até 12 meses após a data referência (data da consulta) e *Mau*,

para todos os clientes que não foram classificados como *Bom*. Esta variável é chamada de CONCEITO, neste estudo de caso.

7.3.1 – DESCRIÇÃO GERAL DOS DADOS

A janela de observação utilizada para modelagem deste problema foi retirada de consultas realizadas no período de janeiro a dezembro de 2007 e observou-se o que aconteceu com cada um destes clientes um ano depois da consulta. Através de uma janela deslizante, ou seja, a contar da data da consulta (data de referência) foi verificado se após 12 meses ocorreu alguma inadimplência do consultado. Este período de avaliação chama-se de período de performance. A amostra retirada foi de 20.000 exemplos e, para avaliação de desempenho, também foi utilizado o mesmo esquema de *folds* dos outros experimentos deste capítulo. A Tabela 7.1 mostra as variáveis que foram utilizadas para o desenvolvimento deste estudo de caso.

Tabela 7.1 – Variáveis utilizadas para desenvolvimento do modelo de crédito

Atributo	Descrição	Tipo
TEMP_ULT_CON	Tempo desde a última consulta	Categórico
TEMP_PRI_CON	Tempo desde a primeira consulta	Categórico
FAIXA_CON_90	Faixa de quantidade de consultas (últimos 90 dias)	Categórico
FAIXA_RENDA	Faixa de Renda do consultado	Categórico
REGIAO_GEO	Região geográfica da residência do consultado	Categórico
CAPITAL_GEO	Verifica se cidade do consultado é capital	Categórico
FLAG_SOCIO	Verifica se o consultado é sócio de empresa	Categórico
FLAG_BANCO	Verifica se o consultado tem conta em banco	Categórico
OCUPAÇÃO	Código do tipo de ocupação do consultado	Categórico
FAIXA_TEMP_RES	Faixa de tempo de residência	Categórico
IDADE	Idade do consultado	Numérico
QTDE_EMPRE	Quantidade de empresas que o consultado é sócio	Numérico
QTDE_CARTA	Quantidade de endereços não encontrados	Numérico
TAM_FAMILIA	Quantidade de pessoas na família	Numérico
CONCEITO	Houve negativo = MAU e Não Houve = BOM	Categórico

7.3.2 – PROCESSAMENTO DAS VARIÁVEIS

De maneira geral, o principal problema encontrado com os dados foi a ausência de preenchimento. Este tipo de problema foi observado apenas em algumas colunas com informações cadastrais. Nos casos em que este problema foi encontrado, a ocorrência de nulos foi codificada como uma nova classe nos dados categóricos e preenchidos com "-1" nas variáveis numéricas. Os dados de comportamento e a variável de avaliação de desempenho (CONCEITO) estavam com um nível de qualidade excelente, considerando as dimensões importantes para Mineração de Dados [Pyl99], e, por isso, não necessitaram de maiores intervenções.

A categorização das variáveis pode ser realizada diretamente pelo método, ou no pré-processamento das variáveis, pois a segmentação é feita internamente com categorias. Porém, não necessariamente, as variáveis precisam participar dos modelos como categóricas, ou seja, uma variável numérica pode ser categorizada para participar do processo de definição de segmentos, mas pode ser utilizada diretamente, de forma numérica, no preditor no segmento da folha. Para os experimentos, as variáveis numéricas foram categorizadas apenas para a determinação dos segmentos e utilizadas no treinamento dos modelos em forma numérica.

Nos estudos de caso deste capítulo, a categorização das variáveis (para segmentação) foi feita utilizando-se os quartis da distribuição de valores de cada variável, ou seja, quatro categorias para cada variável numérica. Para as variáveis numéricas que possuíam mais de 50% dos valores iguais a zero, foram feitas apenas três categorias.

7.3.3 – ANÁLISE DOS RESULTADOS

Neste estudo de caso, foram feitas comparações entre os níveis de profundidade do método *RISKSEG* e variação da métrica de otimização. Não foram feitas comparações com outros métodos de combinação como *Bagging*, *Boosting* e outros tipos de segmentação Nestes experimentos são utilizadas Regressões Logísticas como técnica de predição, *Stacking* como função de combinação das classificações e 3 (três) como número de níveis da profundidade máxima da árvore de modelos. Quanto mais segmentações são feitas, mais interações são investigadas. E quando se exercita muitos níveis de segmentação, a árvore de modelos final tende a ser formada por um número maior de equações, o que pode tornar mais complexa a explicação do cálculo dos escores. Este requisito de explicação de escores gerados pelos modelos é um requisito normalmente requerido na definição deste tipo de aplicação (risco de crédito) [LAW08].

Nos experimentos são permitidos no máximo três níveis de segmentação. Ressalta-se que o ganho adicional a partir de um determinado nível pode não compensar o aumento na quantidade de modelos e a complexidade da árvore final. Outros parâmetros como tamanhos das amostras de treinamento, validação e teste, quantidade mínima de elementos presentes em um segmento, e mínimo ganho de desempenho para segmentar, foram os mesmos utilizados e descritos no início deste capítulo.

Para o parâmetro *D*, que define a métrica de otimização de desempenho, foram utilizados dois valores diferentes, com o objetivo de se realizar um estudo comparativo. Para isso, definiram-se dois conjuntos de experimentos: um conjunto otimizado pela medida *ROC* e outro

pela taxa de Erro de Classificação. O objetivo é verificar como a escolha deste parâmetro interfere nos resultados.

Na Tabela 7.2 são apresentadas as medidas *ROC* e Erro de Classificação obtidas em cada nível do *RISKSEG*, para comparação com o classificador simples. Estes resultados são separados, indicando a métrica de otimização escolhida para os experimentos. As diferenças significativas são indicadas por (*****) e representam a comparação de um nível com o nível anterior, e não a comparação do nível com o experimento base. Assim, fica possível observar de forma direta se o nível maior de segmentação foi melhor que o anterior. Para efeito de comparação com o nível 1, o preditor base é considerado o nível 0. Esta convenção também será adotada para o outro experimento deste capítulo.

Ainda a partir da Tabela 7.2, tem-se que o Erro de Classificação e a medida *ROC* são decrescidos significativamente com o uso da segmentação utilizando o *RISKSEG*. Quando analisados os resultados dos experimentos que utilizaram a medida *ROC* como critério de otimização, nota-se que esta medida tem seus valores melhorados, nível após nível, até o último. Essas melhoras foram estatisticamente significativas. A taxa de erro também foi melhorada, porém o resultado obtido no último nível foi similar ao do segundo nível.

Para os experimentos que utilizaram o erro como objetivo de maximização, as medidas *ROC* Mínimo e erro de classificação foram melhoradas até o segundo nível de segmentação. O último nível não alterou os resultados obtidos para *ROC* e causou uma pequena piora no Erro de Classificação. Neste caso, o ideal seria utilizar apenas dois níveis de segmentação.

Tabela 7.2 – Médias das taxas de erros e intervalos de confiança.

Métrica Otimizada	Método de Combinação	Erro de Classificação (%)	*ROC*MIN
Modelo Simples	**Simples**	29,60 ± 0,14	41,80± 0,33
***ROC* Mín.**	***RISKSEG*_1**	**29,22*** ± 0,11	**40,81*** ± 0,15
	***RISKSEG*_2**	**28,87*** ± 0,14	**40,15*** ± 0,15
	***RISKSEG*_3**	28,86 ± 0,12	**39,98*** ± 0,13
Erro de Classificação (%)	***RISKSEG*_1**	**28,31*** ± 0,07	**41,31*** ± 0,3
	***RISKSEG*_2**	**27,60*** ± 0,14	**40,53*** ± 0,15
	***RISKSEG*_3**	**28,18**** ± 0,31	40,56 ± 0,19

Estes resultados são intuitivos, pois, quando utiliza-se *ROC* como métrica de otimização, a melhoria mais expressiva deveria ser na própria medida. A contrapartida também é esperada: quando se utiliza Erro de Classificação, o ganho mais expressivo deve ser na própria medida.

Na Tabela 7.2, nota-se que o melhor valor para Erro de Classificação (27,6%) foi obtido quando se utilizou essa métrica para otimização. Já a melhor medida *ROC* encontrada (39,98%) foi obtida pelos experimentos que a utilizaram *ROC* para seus critérios de quebra e segmentação.

As Tabelas 7.3 e 7.4 confirmam estas hipóteses, pois exibem testes comparativos, nível a nível, entre os valores encontrados para cada medida, nas duas formas de otimização exploradas.

A Tabela 7.3 mostra as médias dos valores *ROC* encontrados pelos experimentos nos três níveis do *RISKSEG*. Estes valores estão separados pela métrica de otimização utilizada. Nível a nível foram realizados testes t pareados para verificar a diferença. O asterisco (*) indica que o valor *ROC* encontrado pelos experimentos maximizados por *ROC* é significativamente diferente do valor *ROC* à direita, encontrado pelos experimentos que utilizaram maximização por Erro de Classificação. Nota-se que em todos os níveis do *RISKSEG*, os valores de *ROC* são melhores nos experimentos que o utilizaram como critério.

A mesma análise pode ser feita para o Erro de Classificação na Tabela 7.4. Observa-se diferenças significativas entre os critérios de otimização escolhidos, quanto aos valores de Erro de Classificação encontrados.

Tabela 7.3 – Comparação dos resultados da medida *ROC*, por métrica de otimização.

***ROC* Mínimo**		
	Método de Otimização	
	***ROC* Mín.**	**Erro**
***RISKSEG*_1**	**40,81*** ± 0,15	41,31 ± 0,30
***RISKSEG*_2**	**40,15*** ± 0,15	40,53 ± 0,15
***RISKSEG*_3**	**39,98*** ± 0,13	40,56 ± 0,19

Tabela 7.4 – Comparação dos resultados do Erro de Classificação, por métrica de otimização.

Erro de Classificação (%)		
	Método de Otimização	
	***ROC* Mín.**	**Erro**
***RISKSEG*_1**	29,22 ± 0,11	**28,31*** ± 0,07
***RISKSEG*_2**	28,87 ± 0,14	**27,60*** ± 0,14
***RISKSEG*_3**	28,86 ± 0,12	**28,18*** ± 0,31

7.3.4 – EXEMPLO ILUSTRADO DO PROCESSO DE SEGMENTAÇÃO

Nesta seção, o funcionamento do método é demonstrado através de um exemplo que ilustra alguns dos principais passos e valores obtidos durante o treinamento. A demonstração das

comparações são feitas nível a nível, utilizando como métrica de otimização a medida *ROC*, que escolhe a melhor segmentação. Quanto menor o valor do *ROC*, melhor.

O processo de segmentação do *RISKSEG* é descrito utilizando a Figura 7.2. No nó inicial, todos os exemplos da amostra de validação são aplicados ao modelo de Regressão Logística, que foi ajustado com todo o conjunto de treinamento, e apresentou medida *ROC* igual a 0,4257. Após este cálculo, o método de seleção de variáveis é executado, ou seja, rodam-se os modelos para identificação das variáveis candidatas à geração da segmentação. Com as candidatas selecionadas, testam-se todas as combinações dois a dois (divisão binária) possíveis de segmentação das suas categorias. Para cada divisão, são treinados dois modelos, um para cada segmento. Os escores de validação calculados para cada dupla de modelos são combinados através de *Stacking* e é realizado o cálculo do *ROC* nesta junção. O valor resultante é comparado com 0,4257 (nó inicial). Haverá segmentação se existir pelo menos um valor de *ROC* menor do que o medido no nó inicial e o segmento escolhido será aquele que apresentar o melhor (menor) valor da métrica a ser otimizada.

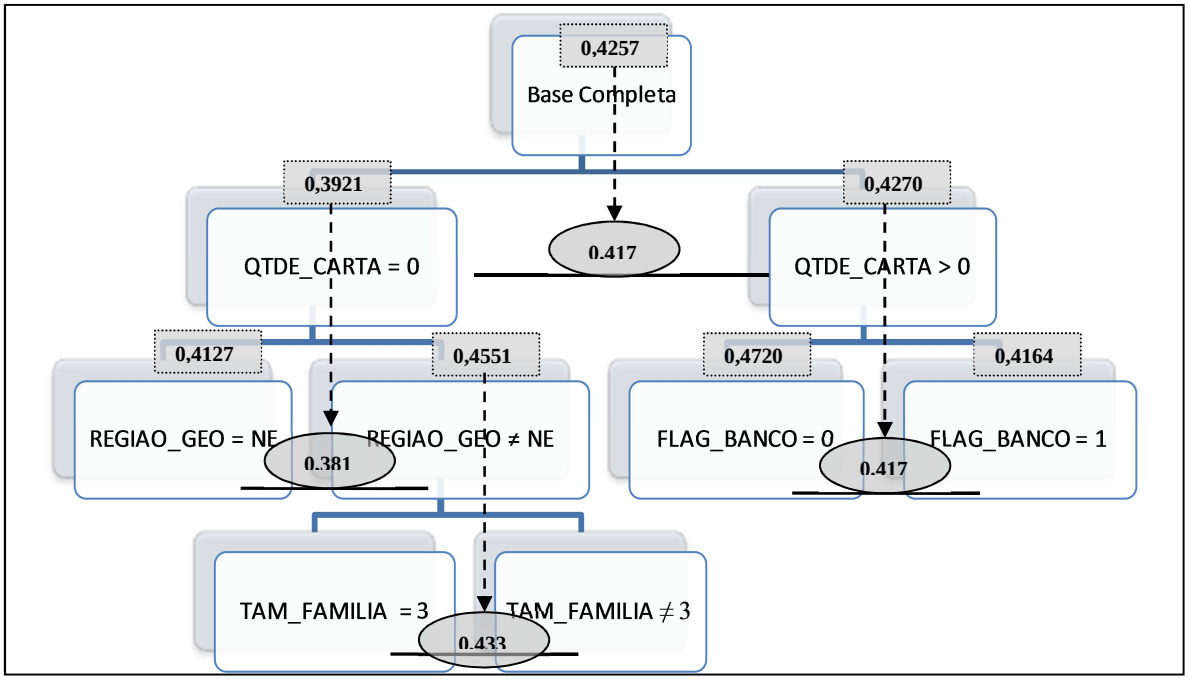


Figura 7.2 – Exemplo de uma árvore de modelos e suas medidas *ROC* na validação.

Como mostrado na Figura 7.2, uma das combinações geradas a partir da segmentação da variável QTDE_CARTA apresentou escores que, combinados, tiveram um valor de 0,417 para o *ROC*. Esta foi a melhor medida encontrada dentre todas as combinações testadas com as categorias das variáveis candidatas. A partir daí, as buscas continuam, na tentativa de melhorar o *ROC* de cada nova folha gerada. A segmentação por REGIAO_GEO e FLAG_BANCO foram as que melhor apresentaram redução de *ROC* para QTDE_CARTA = 0 e QTDE_CARTA > 0,

respectivamente, ambos no segundo nível. Dentre os quatro segmentos do nível 2, apenas um deles se dividiu para formar o terceiro nível. Em outras palavras, apesar do potencial de geração de oito novos segmentos, no terceiro nível há somente mais uma segmentação que melhora o resultado das folhas do nível anterior. A árvore de modelos final ficou com cinco preditores segmentados (nós terminais).

Ainda na Figura 7.2, observa-se que os valores de *ROC* individuais de cada segmento são representados por retângulos, enquanto que os valores de *ROC* das junções entre os segmentos são representados pelas figuras ovais entre as linhas que ligam os segmentos. A seta indica a comparação que foi feita entre o valor de *ROC* do nível superior e a melhor segmentação encontrada no processo de busca, para decidir se haveria segmentação ou não.

7.4 – APLICAÇÃO – FRAUDE NA CONCESSÃO

Assim como a aplicação anterior, além de mostrar a aplicação da técnica em um problema real, esta aplicação também testa o efeito da otimização da segmentação pela métrica *ROC* Mínimo. É importante salientar que esta não é uma métrica normalmente utilizada no treinamento de modelos preditivos, mas, em aplicações do mundo real, são muito usadas como medidas de desempenho para escolha dos melhores modelos de escore de risco.

Neste estudo de caso de fraude não detalha-se os principais passos e resultados intermediários do método proposto, para não se tornar repetitivo. Entende-se que o objetivo de mostrar o processo de segmentação para a aplicação de mundo real foi atingido na aplicação anterior deste capítulo.

O objetivo dos modelos de avaliação de risco de fraude é a classificação deste tipo de evento no momento da solicitação do crédito. Para este estudo de caso, estabeleceu-se que seria mensurado o risco do cliente realizar uma operação fraudulenta no momento da consulta. Para apuração deste evento (fraude), observou-se em um período de 6 meses após a consulta se aquela operação foi válida, ou seja, se o cliente efetuou o primeiro pagamento, se o endereço onde mora continuou o mesmo, se o titular do CPF não apareceu como óbito dias depois da operação, entre outros fatores que a instituição considera como fraude. Essa indicação de fraude ou não fraude deu origem à variável dicotômica CONCEITO.

7.4.1 – DESCRIÇÃO GERAL DOS DADOS

O período de observação utilizado deste problema foi retirado de consultas realizadas no período de janeiro a dezembro de 2008 e observou-se o que aconteceu com cada um destes

clientes seis meses após a consulta. A amostra retirada foi de 20.000 exemplos e, para avaliação de desempenho, também foi utilizado o mesmo esquema de *folds* da outra aplicação deste capítulo. A Tabela 7.5 mostra as variáveis que foram disponibilizadas para o desenvolvimento desta aplicação.

Tabela 7.5 – Variáveis utilizadas para desenvolvimento do modelo de fraude

Atributo	Descrição	Tipo
NIVEL_ESCO	Nível de Escolaridade	Categórico
IDADE	Faixa etária	Categórico
RENDA	Faixa de Renda do consultado	Categórico
ESTCIVIL	Estado Civil	Categórico
REFBANC	Verifica se o consultado tem conta em banco	Categórico
QTDE_END	Quantidade de endereços que possui	Categórico
CONSULTAS_ME	Numero de consultas de uma mesma empresa	Numérico
QT_BANCOS	Quantidade de Bancos em que possui conta	Numérico
QTDE_SOC	Quantidade de empresas que o consultado é sócio	Numérico
TEMPO_ULT_CONS	Tempo desde a última consulta	Numérico
CONS180	Quantidade de Consultas nos últimos 180 dias	Numérico
TEMPO_TELEFONE	Tempo que possui o telefone cadastrado	Numérico
CONCEITO	Houve fraude = MAU e Não Houve = BOM	Categórico

7.4.2 – PROCESSAMENTO DAS VARIÁVEIS

Assim como na aplicação de Risco de Crédito, no processamento das variáveis, o maior problema encontrado foi a ausência de preenchimento em algumas colunas com informações cadastrais. Para estes casos, o tratamento adotado também foi o mesmo.

7.4.3 – ANÁLISE DOS RESULTADOS

Assim como na aplicação de crédito, nestes experimentos são permitidos no máximo três níveis de segmentação. Outros parâmetros como tamanho das amostras de treinamento, validação e teste, quantidade mínima de elementos presentes em um segmento e mínimo ganho de desempenho para segmentar, foram os mesmos utilizados e descritos no início deste capítulo. Nestes experimentos também utilizou-se duas medidas para otimização *ROC* e Erro de Classificação.

Consolidado na Tabela 7.6, tem-se que o Erro de Classificação e a medida *ROC* são melhorados significativamente com o uso da segmentação *RISKSEG*. Nota-se que a melhoria na métrica se dá até o segundo nível e observa-se que no terceiro nível não há melhora significativa. Este resultado é observado nas duas escolhas de métrica de otimização. Os valores de Erro de Classificação melhoram até o segundo nível independentemente de ter escolhido *ROC* ou o próprio Erro de Classificação como objetivo de otimização. O mesmo ocorre com os valores de *ROC*. Então nestes casos, o ideal seria utilizar apenas dois níveis de segmentação.

Tabela 7.6 – Médias das taxas de erros e intervalos de confiança.

Métrica Otimizada	Método de Combinação	Erro de Classificação (%)	*ROC*MIN
Modelo Simples	**Simples**	11,60 ±0,79	0,1982 ± 0,16
***ROC* Mín.**	***RISKSEG*_1**	**10,37* ±0,94**	**0,1782* ± 0,01**
	RISKSEG_2	**9,45* ±0,79**	**0,1587* ± 0,01**
	***RISKSEG*_3**	9,49 ±0,44	0,1711 ± 0,01
Erro de Classificação (%)	***RISKSEG*_1**	**10,20* ± 1,04**	**0,1790* ± 0,01**
	***RISKSEG*_2**	**9,24* ± 0,84**	**0,1648* ± 0,02**
	***RISKSEG*_3**	8,84 ± 0,40	0,1533 ± 0,00

As Tabelas 7.7 e 7.8 mostram uma comparação direta do efeito da escolha do parâmetro *D* nos resultados. A Tabela 7.7 mostra os resultados de *ROC* Mínimo cruzando com qual métrica foi escolhida para otimizar. Apesar da diferença visual dos números, o teste significância não aponta diferenças, ou seja, os valores de *ROC* apresentaram melhorias equivalentes mesmo escolhendo Erro de Generalização como objetivo. Isso pode ocorrer, pois são duas medidas correlacionadas, logo, quando se melhora uma, a outra também melhora. Essa correlação entre essas duas medidas é totalmente intrínseca à base de dados em questão. A Tabela 7.8 reforça a análise, pois tem-se que as médias de Erro de Classificação são estatisticamente iguais, independentemente da métrica escolhida.

Tabela 7.7 – Comparação dos resultados da medida *ROC*, por métrica de otimização.

***ROC* Mínimo**		
	Método de Otimização	
	***ROC* Mín.**	**Erro**
***RISKSEG*_1**	**0,1782** ± 0,01	0,1790 ± 0,01
***RISKSEG*_2**	**0,1587** ± 0,01	0,1648 ± 0,02
***RISKSEG*_3**	0,1711 ± 0,01	**0,1533** ± 0,00

Tabela 7.8 – Comparação dos resultados do Erro de Classificação, por métrica de otimização.

Erro de Classificação (%)		
	Método de Otimização	
	***ROC* Mín.**	**Erro**
***RISKSEG*_1**	10,37 ±0,94	**10,20** ± 1,04
***RISKSEG*_2**	9,45 ± 0,79	**9,24** ± 0,84
***RISKSEG*_3**	9,49 ±0,44	**8,84** ± 0,40

7.5 – RESUMO GERAL DOS RESULTADOS

A aplicação do método *RISKSEG* nos dois casos reais de modelagem mostra que este pode ser aplicado para melhorar o poder preditivo, segundo a métrica objetivo. O número de níveis de segmentação necessários para capturar o máximo das relações entre as variáveis varia de acordo com a base. Na primeira aplicação, o método apresentou ganhos significativos até o terceiro nível. Já para a segunda base, dois níveis foram suficientes para atingimento do seu melhor resultado.

O método permite escolher qual o objetivo a ser otimizado, ou seja, qual métrica deve ser utilizada para avaliar a captura das relações de dependência. Em alguns casos, a complexidade do problema influi de maneira diferente nas métricas de avaliação. Por esta razão, é importante utilizar mais de uma medida para julgar se o modelo está ou não adequado, ou ainda, se o modelo pode ou não ser melhorado.

As aplicações deste capítulo mostraram que as diferentes formas de otimizar uma segmentação podem levar a atingir diferentes resultados. Isto indica que deve-se buscar uma estrutura de segmentação voltada à melhoria da métrica desejada. Por exemplo, caso o objetivo seja obter um aumento de KS2, deve-se utilizar esta métrica de otimização no método.

CAPÍTULO 8

CONCLUSÕES E TRABALHOS FUTUROS

Esta tese apresenta um método de combinação de preditores baseado na segmentação de amostras para melhorar o poder preditivo de modelos de risco. Os resultados mostram que, além de proporcionar este benefício, quando comparado com técnicas de um único preditor, o método proposto também se mostra não tão complexo de ser implementado e possui diversas vantagens em relação aos métodos tradicionais de combinação de preditores. A seguir, são discutidos alguns resultados, contribuições e trabalhos futuros.

8.1 – CONCLUSÕES

O trabalho propõe um novo método de segmentação baseado no modelo genérico de segmentação [Rok05], diferenciando-se deste, principalmente, pelo algoritmo para seleção das variáveis que são utilizadas para geração dos modelos segmentados. Como foi apresentado no Capítulo 4, o *RISKSEG* utiliza modelos com interações entre as variáveis preditoras para fazer a seleção das que serão utilizadas na segmentação. Quando comparado com métodos de segmentação com buscas mais exaustivas, o *RISKSEG* se mostra mais eficiente em relação ao tempo de treinamento, ou seja, necessita de um menor número de treinamento de modelos para decisão da divisão mais adequada.

A efetividade na forma de segmentar do método proposto é mostrada, na teoria, para modelos de Regressão Logística e Linear. Porém, com os experimentos do Capítulo 6, verificou-se que o *RISKSEG* também obteve resultados melhores e significativos quando aplicado também com Redes Neurais e comparado com o método de classificação Simples. Isto significa que, mesmo em algoritmos comprovadamente de alto poder preditivo para aplicações não lineares [Kia03], ainda assim, é possível a melhoria do poder de predição através da segmentação usando o *RISKSEG*.

O método proposto também foi comparado com os métodos de combinação clássicos *Bagging* e *Boosting*, além de outras duas formas de segmentação, uma por *information Gain* [Nev98] e outra por *NNTree* [Pra08]. O *RISKSEG* obteve melhores resultados, em muitos experimentos, quando comparado com todos os métodos de combinação utilizados, inclusive com o tradicional método de segmentação que utiliza *Information Gain* [Nev98] [Nev99] [CGM02].

O método *NNTree* é descrito originalmente para trabalhar com Redes Neurais, mas ela foi implementada de maneira a utilizar também Regressão Logística e Linear. Com esta adaptação foi possível fazer uma comparação mais completa com os demais métodos de combinação. Observou-se que esta implementação do *NNTree* apresentou bons resultados com as Regressões em algumas bases.

Os resultados mostrados nos Capítulos 5, 6 e 7 possibilitam a identificação dos pontos fortes e fracos de cada uma das técnicas avaliadas, tanto em um conjunto de bases simuladas, como em bases do repositório *UCI* [BlM10], além de em duas bases de mundo real da *Serasa Experian*. Um aspecto importante para a avaliação dos resultados e determinação destes pontos fortes e fracos foi o uso de um ambiente controlado do Capítulo 5.

No Capítulo 5, foram realizados diversos experimentos com bases simuladas de forma que as características das bases, como número de registros, relação das variáveis preditoras com a variável dependente, método de regressão e os parâmetros das técnicas, sejam totalmente definidos. Os resultados das análises deste capítulo mostraram-se bem condizentes com a teoria estudada. Os experimentos permitiram observar algumas condições nas quais o método proposto apresenta vantagens quando comparado com os demais.

Ainda no Capítulo 5, notou-se claramente a influência da presença de interações entre variáveis e o bom desempenho do método *RISKSEG*. Também observou-se que o método melhora os resultados mesmo quando a não-linearidade das interações nas variáveis é expressa indiretamente, como mostrado na segunda base. No mínimo, o *RISKSEG* se mostra com resultados análogos a qualquer outro método de segmentação, pelo menos no primeiro nível. Isto ocorre, pois, conforme discutido, qualquer segmentação é análoga à inserção de termos de interação no modelo estimado. Logo, se qualquer outro tipo de segmentação tem êxito em melhorar o poder de classificação, então se conclui que a variável escolhida por este outro método possui relação de interação com as outras variáveis e, portanto, pode ser capturada pelo *RISKSEG*.

O *RISKSEG*, por sua concepção, foca em resultados imediatos, ou seja, escolhe sempre a regra que maximize os resultados em relação ao nível anterior. Este tipo de estratégia pode não ser ótimo, pois acaba sofrendo problemas de máximos locais. É possível que uma segmentação mais fraca no início proporcione ganhos maiores após outras segmentações. Isto pode ocorrer ao se utilizar o método proposto, pois é foco deste a minimização da profundidade de árvore. Então, é possível que outros métodos de segmentação apresentem resultados melhores depois que a árvore de modelos estiver totalmente treinada. Normalmente, as características não lineares da base de dados são capturadas nos primeiros níveis da segmentação e as melhorias tendem a ser menores a cada novo nível.

Uma das limitações do *RISKSEG* refere-se diretamente ao tamanho de amostra disponível. Todos os tipos de segmentação são impactados por esta característica, uma vez que, a cada nível, menos registros estão disponíveis para os treinamentos marginais nos nós. O método proposto pode ser ainda mais impactado, pois utiliza a estimação de modelos com interações para tomada de decisão utilizando também um conjunto para validação. Se os conjuntos disponíveis para os treinamentos são muito pequenos, os conjuntos de validação, utilizados para a determinação das interações fortes, são menores ainda. Isto interfere diretamente na tomada de decisão, criando muita variância nos resultados obtidos.

Uma vantagem que pode ser destacada com a utilização de combinação de preditores através de segmentações é a característica explicativa preservada da técnica que gerou o escore. Isto é facilitado graças à característica da segmentação de rodar apenas um modelo por predição, diferentemente de *Bagging* e *Boosting* que necessitam rodar vários. Então, o detalhamento do cálculo dependerá apenas do modelo que foi sensibilizado para aquele caso específico. Sabe-se que este cálculo é mais facilmente obtido para algumas técnicas do que para outras. A Regressão Linear, por exemplo, é considerada de simples explicação, já que o seu resultado é apenas uma equação linear, porém é uma técnica com poder preditivo normalmente inferior a outras como Redes Neurais e Regressão Logística [Kia03].

Outra vantagem do *RISKSEG* é a possibilidade de aumento de desempenho da resposta de aplicações, aproveitando técnicas já conhecidas pelas equipes de modelagem, pois, como os experimentos mostram, o método proposto pode, em muitos casos, melhorar o resultado final de distintas técnicas de classificação ou regressão. Desta forma, se uma equipe utiliza apenas Regressão Linear em seus modelos, ao aplicar o método proposto, poderá obter melhores resultados e aproveitar imediatamente todo o conhecimento já existente e sem necessariamente buscar ou migrar para outras técnicas individuais ou de combinação de preditores, possivelmente mais preditivas.

A vantagem descrita anteriormente torna-se factível porque o conceito básico da segmentação baseada na divisão de conjuntos (através de variáveis) é mais simples quando comparada com outros métodos de combinação de preditores. A segmentação é apenas a divisão dos dados disponíveis para treinamento em diversos modelos especializados nas características de cada conjunto disjunto, tornando o aprendizado mais fácil de ser disseminado e sua aplicação menos complexa pelos desenvolvedores de modelos.

Como visto, o *RISKSEG* aproveita a expertise existente da equipe de modelagem e permite que a complexidade do modelo resultante esteja mais atrelada à(s) técnica(s) que está(ão) sendo utilizada(s). O número máximo de preditores também pode ser controlado, o que pode evitar que se tenha um número muito grande de modelos. Isto quer dizer que aspectos relacionados às restrições da aplicação, como tempo de resposta, tempo de desenvolvimento, explicação de resultados e desconhecimento de técnicas, podem ser facilmente ajustados.

Como já dito, o método apresenta um tempo de treinamento relativamente alto quando comparado com técnicas individuais, pois ele é resultado de treinamento de vários modelos individuais. Esta desvantagem é quase desprezível, após o treinamento dos segmentos, porque o tempo de execução (produção) é muito próximo de uma técnica individual, pois, para cada classificação, é necessário rodar apenas um modelo (o do segmento sensibilizado). O tempo de execução é também menor em relação aos outros métodos de combinação, como *Boosting*, *Bagging*, *Stacking* e *NNTree*. Existem mecanismos que também podem ser utilizados para diminuir o tempo de treinamento. Eles têm muita relação com os critérios de parada apresentados no Capítulo 4, pois, através de experimentos, observa-se que a melhoria dos resultados é cada vez menor à medida que a profundidade da árvore aumenta. Isto significa que não adianta ter um número grande de segmentos, pois o tempo de treinamento aumenta consideravelmente, sem que haja um incremento de preditividade equivalente. Além disso, o aumento da quantidade de modelos para serem utilizados também aumenta a complexidade da tarefa de disponibilização do seu uso.

Os resultados obtidos mostram que o *RISKSEG* pode ser indicado em aplicações de risco, com um ganho significativo de desempenho. As possibilidades de uso para o método são bastante flexíveis, pois este pode ser usado com técnicas mais tradicionais sem perder as suas principais características de interpretação dos resultados de cada escore e implementação em produção facilitada. O modelo final, por se tratar de *n* “equações”, pode ser visto como *n* modelos aplicados a subpopulações com características distintas. Estas subpopulações podem ser estudas separadamente e as conclusões sobre os efeitos das variáveis preditoras na variável resposta serão diferentes para cada uma. É mais simples enxergar que uma variável ‘A’ atua de

maneira diferente de acordo com os subgrupos, do que analisar um fator de interação em uma equação única que é aplicada a todo o conjunto de dados, sem segmentar. Logo, este método, além de manter as características de explicação oferecidas por métodos tradicionais, pode gerar explicações extras de grande importância para o pesquisador.

8.2 – CONTRIBUIÇÕES

A maior contribuição deste trabalho é a proposição de um novo método de segmentação que pode ser utilizado com qualquer técnica de classificação e para alguns tipos de regressão (com alvo dicotômico). Este método demonstrou, em muitos casos, melhores resultados quando comparados com métodos preditores com apenas um indutor, e mesmo com outros métodos de combinação de classificadores.

Uma contribuição inédita deste trabalho é uma comparação dos métodos de combinação de classificadores como *Bagging e Boosting*, com três outras propostas de segmentação (*SegTree*, *RISKSEG e NNTree*). Na comparação com os métodos *Boosting* e *Bagging*, o *RISKSEG* apresenta vantagens, como a versatilidade no uso de qualquer tipo de técnica de classificação e regressão, inclusive de diferentes tipos ao mesmo tempo, maior facilidade de distribuição, possibilidade de interpretação dos resultados dos escores e, em muitos casos, melhor desempenho. A utilização de *NNTree* [Pra08] é importante porque compara o método proposto com uma segmentação mais recente.

As simulações realizadas nos Capítulos 5 e 6 permitiram a avaliação de métodos individuais de classificação (Redes Neurais, Regressão Logística e Regressão Linear) e de diversos métodos de combinação (*Bagging*, *Boosting*, *NNTree*, Segmentação por *Information Gain* e *RISKSEG*) em 4 (quatro) bases geradas artificialmente e em 12 bases do repositório *UCI* [BlM10]. Os resultados destes dois capítulos podem ser utilizados para comparações com outros trabalhos.

No Capítulo 7, este trabalho explorou outras possibilidades de métricas de otimização. Nas técnicas de classificação tradicionais, normalmente, a otimização limita-se à sua própria natureza, como, por exemplo, erro quadrático médio e verossimilhança, para as Regressões Lineares e Logísticas, respectivamente. Métodos de combinação permitem que outras naturezas de medidas possam ser maximizadas. Por exemplo, métricas como *KS2* [Con99] e *ROC* [Faw06] são obtidas somente após o treinamento do método preditor. Desta forma, métodos de combinação oferecem a possibilidade de sua maximização. O método proposto explora esta possibilidade de otimização de métricas pouco ortodoxas, mas que já são utilizadas como

avaliadoras de desempenho em trabalhos acadêmicos e aplicações de mundo real [NaP03] [AVA04]. Tal paradigma é pouco ou nada explorado em trabalhos com *Bagging* e *Boosting*, pois eles geralmente focam na apresentação de medidas de erro (normalmente erro de classificação).

Também no Capítulo 7, o método *RISKSEG* foi testado em 2 (duas) bases reais e otimizado a partir de duas métricas diferentes: erro de classificação e a medida *ROC*. Este tipo de experimento não foi encontrado em nenhum outro trabalho no meio acadêmico, mas os seus resultados sugerem que este tipo de otimização pode ser utilizado em outros problemas de mundo real, que tenham suas próprias métricas de otimização.

Como dito anteriormente, foi implementado um algoritmo de segmentação utilizando um conjunto de Redes Neurais, o método *NNTree* [Pra08]. Porém, este também foi disponibilizado com duas outras Regressões Estatísticas (Logística e Linear). Portanto, uma outra contribuição é a comprovação experimental da possível extensão deste método de segmentação.

8.3 – TRABALHOS FUTUROS

Esta tese origina muitas perspectivas futuras de trabalhos e investigações. Nesta seção são destacados alguns dos possíveis estudos para complementar e ampliar as discussões, que, por razões óbvias de objetivo e escopo, não foram contemplados.

Apesar da quantidade razoável de diferentes bases de dados utilizadas nos capítulos de experimentos, ainda assim, seria recomendado aplicar o método em outros conjuntos de dados para ampliação do universo de aplicações do método proposto. Uma opção interessante seria, por exemplo, utilizar diferentes massas de dados que englobem outros tipos de classificação com variável resposta não dicotômica. Esta abordagem de alvos não dicotômicos implica em adaptações nos métodos de combinação utilizados, pois não mais seriam combinados escores e sim as classificações diretamente.

Outra oportunidade de estudos futuros é a exploração do comportamento do método quando utilizando outras técnicas de aprendizado de máquina como: *KNN* [WiF05], *SVM* [MLH03], entre outras. Elas podem ser exploradas de forma individual, como os experimentos realizados nesta tese, ou utilizando diferentes técnicas no mesmo treinamento. Desta última forma, a cada iteração, os modelos de diferentes técnicas podem ser testados para eleição da melhor maneira de dividir e formar os segmentos, dependendo apenas do ganho obtido com as diferentes combinações de categorias de variáveis e técnicas. Espera-se que a mistura de diversas técnicas possa gerar um ganho ainda maior. A desvantagem é que isso implica em um

tempo de treinamento maior, pois o teste de diferentes técnicas para cada segmento pode ser muito custoso. Então, este estudo pode também englobar a procura de uma maneira mais analítica de escolha da técnica que deve ser testada para cada treinamento e/ou segmento, de modo a tentar evitar esta busca custosa.

A forma de escolher como os segmentos serão formados é a principal característica do *RISKSEG*. Ela utiliza um modelo com a interação de uma das variáveis com as demais. Neste trabalho utiliza-se Regressão Logística para estes modelos intermediários de seleção das variáveis candidatas. Porém, uma possível investigação seria a utilização de outras técnicas de aprendizado de máquina, como Redes Neurais, para este fim. Talvez existam outras técnicas e situações específicas, onde elas sejam mais adequadas para a divisão e consequente formação dos segmentos.

Uma das vantagens da combinação de preditores é o potencial de otimização de métricas de avaliação de desempenho, através da escolha do melhor preditor para uma determinada métrica. Ou seja, é possível direcionar o desempenho da solução para métricas que, muitas vezes, não podem ser utilizadas diretamente nos métodos simples de predição, por exemplo, *KS2 e ROC*. Ainda existe, a possibilidade de usar diretamente na otimização da solução, funções de custo mais complexas com pesos ou variáveis respostas de negócio, como lucratividade. Seria particularmente interessante experimentar o método com outras formas de otimização, de maneira a cobrir necessidades de diferentes domínios de aplicação.

Neste trabalho não foram feitos experimentos comparando diferentes parâmetros de inicialização do método. Alguns testes paralelos foram realizados para determinar os valores a serem utilizados nos experimentos apresentados, porém, um estudo mais aprofundado poderá ser feito, uma vez que a quantidade de parâmetros das técnicas de aprendizado e dos métodos de combinação seria muito grande para serem completamente esgotadas neste trabalho. Futuros estudos podem ser úteis para melhorar a sensibilidade dos resultados em relação à escolha dos parâmetros. Além da observação do comportamento dos resultados, este estudo pode servir como insumo para a criação de métodos automáticos para a definição dos valores destes parâmetros. Pode-se considerar, por exemplo, características da base de dados e do domínio da aplicação. O que se deseja é o ajuste destes parâmetros direcionados aos aspectos particulares do problema, o que certamente aumentaria a eficiência do método.

Trabalhos futuros também podem focar em outro tipo de árvore de segmentação. Neste trabalho, contemplaram-se os estudos com divisões binárias da base de dados. Divisões *n*-árias devem ser exploradas para determinar se podem promover maiores ganhos, uma vez que as

estruturas de segmentação seriam mais complexas. A literatura já apresenta alguns trabalhos de busca do número ideal de folhas para divisão da base de dados [JSB05], embora em contexto diferente do tratado nesta tese.

O método proposto mostra a importância das interações entre as variáveis nos classificadores. Assim, pode-se pensar que este também poderia ser utilizado para criar variáveis derivadas de interações e inseri-las diretamente nos modelos simples (com um preditor somente), a fim de melhorar a capacidade preditiva, sem a necessidade de usar métodos de segmentação. Esta é uma possibilidade que, em trabalhos futuros, pode ser explorada e comparada com técnicas específicas para esta finalidade.

Observa-se, portanto, que há diversas opções para a ampliação e continuidade deste trabalho. As discussões deste capítulo não têm a intenção de esgotar todas as possibilidades de trabalhos futuros e sim, expor quais seriam algumas das possíveis formas de continuidade do trabalho, para que haja uma consolidação do método *RISKSEG* no meio acadêmico e para que se possa também utilizá-lo mais efetivamente nos diferentes contextos organizacionais.

8.4 – DISCUSSÕES FINAIS

A proposta inicial deste trabalho era a proposição de um método de combinação de classificadores baseado no *Stacking*, mas os resultados iniciais não foram satisfatórios e a investigação por possíveis alternativas levou à utilização de segmentação.

A tarefa de combinar classificadores possui diversas dimensões de complexidade na sua implementação. Para exemplificar, pode-se imaginar que, para uma determinada aplicação, deseja-se o melhor desempenho possível, utilizando ou não técnicas de combinação de classificadores. Para a sua execução é necessário identificar as técnicas individuais que serão utilizadas (Redes Neurais, Regressões Estatísticas, por exemplo) e selecionar quais os métodos de combinação que podem ser utilizados. Feita esta primeira identificação, é preciso experimentar e selecionar os diversos parâmetros para que se obtenha os melhores resultados. Desta maneira, tem-se não somente os parâmetros das técnicas individuais, mas também dos métodos a serem testados. Desta forma, a execução de projetos que utilizem combinação de técnicas preditoras envolverá toda a complexidade já existente de construção de modelos individuais e mais as dificuldades que envolvem os métodos de combinação de classificadores.

Os desafios para implementação experimental deste trabalho foram muitos, mas, em especial, pode-se destacar o desenvolvimento de todo código de programação dos algoritmos que foram utilizados nas comparações e o desenvolvimento do próprio método *RISKSEG*. Para

este trabalho, além da implementação de métodos tradicionais como *Bagging* e *Boosting*, também foram estudados e adaptados métodos de segmentação como o *NNTree*.

Além do esforço para desenvolvimento da plataforma com todos os métodos de combinação para suportar os experimentos, outra dificuldade apresentada foi o tempo de execução para os experimentos. Como dito anteriormente, a quantidade de modelos individuais ficou em centenas de milhares de treinamentos, contando somente os resultados finais que foram utilizados. Houve uma dificuldade muito forte, pois, em geral, a experimentação de testes resultava na espera de dias de processamento, mesmo contando com 4 (quatro) a 6 (seis) microcomputadores de última geração.

Apesar de todas as dificuldades, provar que o *RISKSEG* apresentou resultados satisfatórios, quando comparados com outros métodos de combinação de preditores (clássicos ou mais modernos), sugere que o trabalho obteve êxito na sua proposta maior de criação de um método capaz de concorrer neste nicho específico.

APÊNDICE A

VALORES DOS EXPERIMENTOS DOS CAPÍTULOS 5, 6 E 7

Este apêndice contém Tabelas com todos os valores brutos dos experimentos realizados nos Capítulos 5, 6 e 7. As Tabelas A.1 até A.24 referem-se aos experimentos do Capítulo 5, as Tabelas A.25 a A.36 são referentes ao Capítulo 6 e as demais ao capítulo seguinte.

Nos títulos das colunas das tabelas, quando são encontrados textos como "*_n*", o número *n* corresponde ao nível máximo experimentado na segmentação. Por exemplo, *RISKSEG_2*, corresponde aos experimentos com 2 (dois) níveis máximos de profundidade na segmentação com o método *RISKSEG*.

Tabela A.1 – Erros de Classificação da Base (1) para 1.000 registros.

Fold	Base 1																							
	1.000 registros																							
	Regressão Linear												Regressão Logística											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,168	0,1	0,092	0,092	0,128	0,124	0,068	0,068	0,068	0,1443	0,0888	0,0832	0,088	0,076	0,084	0,084	0,092	0,096	0,076	0,084	0,084	0,088	0,088	0,088
2	0,112	0,08	0,088	0,088	0,096	0,104	0,096	0,096	0,096	0,1489	0,0977	0,0915	0,088	0,108	0,104	0,104	0,08	0,084	0,096	0,096	0,096	0,088	0,088	0,088
3	0,172	0,108	0,096	0,096	0,164	0,164	0,108	0,144	0,144	0,2120	0,0948	0,0939	0,12	0,068	0,068	0,068	0,096	0,1	0,12	0,12	0,12	0,12	0,12	0,12
4	0,132	0,1	0,1	0,1	0,132	0,112	0,116	0,124	0,124	0,2334	0,1205	0,1137	0,12	0,1	0,1	0,1	0,12	0,12	0,12	0,12	0,12	0,12	0,12	0,12
5	0,252	0,136	0,132	0,132	0,18	0,176	0,132	0,132	0,132	0,2370	0,1178	0,1132	0,108	0,084	0,08	0,08	0,144	0,152	0,108	0,108	0,108	0,108	0,108	0,108
6	0,092	0,108	0,092	0,092	0,096	0,096	0,1	0,1	0,1	0,1283	0,1464	0,1398	0,112	0,108	0,108	0,108	0,116	0,104	0,112	0,112	0,112	0,112	0,112	0,112
7	0,168	0,132	0,104	0,104	0,172	0,168	0,12	0,12	0,12	0,2378	0,1466	0,1352	0,128	0,108	0,1	0,1	0,116	0,112	0,112	0,112	0,112	0,1023	0,1023	0,1023
8	0,228	0,14	0,088	0,088	0,22	0,22	0,124	0,124	0,124	0,205	0,1090	0,1	0,14	0,12	0,092	0,092	0,132	0,136	0,12	0,12	0,12	0,1175	0,1175	0,1175
9	0,16	0,092	0,104	0,104	0,152	0,156	0,112	0,14	0,14	0,1980	0,1179	0,1169	0,108	0,076	0,104	0,104	0,112	0,108	0,104	0,104	0,104	0,108	0,108	0,108
10	0,128	0,124	0,124	0,124	0,144	0,116	0,108	0,156	0,156	0,1435	0,1435	0,1435	0,14	0,124	0,116	0,116	0,132	0,132	0,12	0,116	0,116	0,1267	0,1267	0,1267
11	0,228	0,116	0,14	0,14	0,256	0,26	0,136	0,136	0,136	0,1414	0,1122	0,1095	0,132	0,124	0,124	0,124	0,148	0,152	0,132	0,132	0,132	0,132	0,132	0,132
12	0,192	0,144	0,128	0,128	0,176	0,184	0,156	0,156	0,156	0,2004	0,1738	0,1713	0,128	0,1	0,1	0,1	0,128	0,132	0,14	0,14	0,14	0,128	0,128	0,128
13	0,224	0,112	0,12	0,12	0,212	0,212	0,136	0,156	0,156	0,2543	0,1612	0,1522	0,148	0,104	0,096	0,096	0,128	0,14	0,148	0,16	0,16	0,1446	0,1340	0,1340
14	0,256	0,116	0,124	0,124	0,244	0,252	0,144	0,148	0,148	0,2695	0,1838	0,1780	0,148	0,088	0,108	0,108	0,14	0,14	0,14	0,148	0,148	0,1335	0,1397	0,1432
15	0,124	0,104	0,072	0,072	0,116	0,124	0,104	0,104	0,104	0,1787	0,1262	0,1226	0,112	0,096	0,088	0,088	0,116	0,108	0,116	0,116	0,116	0,1207	0,1207	0,1207
16	0,132	0,092	0,104	0,104	0,128	0,128	0,112	0,12	0,12	0,1277	0,1	0,0979	0,108	0,072	0,056	0,056	0,112	0,108	0,108	0,108	0,108	0,0939	0,0939	0,0939
17	0,144	0,124	0,112	0,112	0,128	0,12	0,108	0,108	0,108	0,1508	0,0977	0,0906	0,104	0,064	0,08	0,08	0,092	0,096	0,104	0,104	0,104	0,104	0,104	0,104
18	0,208	0,112	0,112	0,112	0,236	0,236	0,144	0,152	0,152	0,1985	0,1336	0,1380	0,124	0,12	0,12	0,12	0,128	0,128	0,124	0,124	0,124	0,124	0,124	0,124
19	0,204	0,184	0,176	0,176	0,184	0,148	0,168	0,168	0,168	0,2025	0,1454	0,1371	0,148	0,132	0,132	0,132	0,14	0,144	0,14	0,144	0,144	0,1375	0,12	0,1171
20	0,24	0,1	0,096	0,096	0,212	0,224	0,1	0,112	0,112	0,2072	0,0931	0,0899	0,104	0,108	0,076	0,076	0,1	0,104	0,096	0,096	0,096	0,104	0,104	0,104
21	0,212	0,08	0,092	0,092	0,236	0,232	0,128	0,108	0,108	0,1662	0,1034	0,1006	0,112	0,096	0,076	0,076	0,1	0,096	0,116	0,1	0,1	0,112	0,112	0,112
22	0,244	0,124	0,124	0,124	0,184	0,204	0,104	0,104	0,104	0,1418	0,1408	0,1320	0,092	0,088	0,076	0,076	0,072	0,084	0,096	0,096	0,096	0,1100	0,1056	0,1045
23	0,292	0,116	0,092	0,092	0,252	0,216	0,1	0,1	0,1	0,25	0,1308	0,1278	0,104	0,116	0,096	0,096	0,108	0,108	0,104	0,104	0,104	0,104	0,104	0,104
24	0,14	0,112	0,112	0,112	0,136	0,152	0,14	0,14	0,14	0,1795	0,1032	0,1009	0,1	0,1	0,1	0,1	0,1	0,104	0,1	0,1	0,1	0,1022	0,1022	0,1022
25	0,152	0,132	0,132	0,132	0,168	0,184	0,136	0,152	0,152	0,1930	0,1308	0,1207	0,14	0,132	0,132	0,132	0,124	0,128	0,14	0,14	0,14	0,14	0,14	0,14
26	0,188	0,116	0,112	0,112	0,172	0,172	0,12	0,12	0,12	0,2647	0,1413	0,1339	0,12	0,108	0,1	0,1	0,132	0,136	0,12	0,12	0,12	0,12	0,12	0,12
27	0,196	0,1	0,088	0,088	0,196	0,196	0,1	0,112	0,112	0,2560	0,1228	0,1191	0,092	0,092	0,092	0,092	0,112	0,112	0,092	0,092	0,092	0,092	0,092	0,092
28	0,108	0,108	0,108	0,108	0,1	0,104	0,108	0,108	0,108	0,108	0,108	0,108	0,116	0,116	0,116	0,116	0,116	0,116	0,116	0,116	0,116	0,1060	0,0948	0,0948
29	0,2	0,136	0,136	0,136	0,18	0,188	0,128	0,128	0,128	0,1091	0,1133	0,1070	0,128	0,084	0,12	0,12	0,132	0,132	0,12	0,12	0,12	0,0989	0,0989	0,0989
30	0,196	0,14	0,132	0,132	0,2	0,18	0,156	0,18	0,18	0,1766	0,1588	0,1558	0,176	0,16	0,124	0,124	0,144	0,148	0,164	0,168	0,168	0,176	0,176	0,176

Fold	Base 1																							
	1.000 registros																							
	Redes Neurais, 3 Neurônios, BPROP												Redes Neurais, 3 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,092	0,096	0,08	0,08	0,068	0,076	0,096	0,096	0,096	0,1189	0,1055	0,0978	0,076	0,076	0,076	0,076	0,076	0,06	0,076	0,076	0,076	0,076	0,076	0,076
2	0,072	0,112	0,112	0,112	0,048	0,064	0,068	0,068	0,068	0,0858	0,0867	0,0872	0,088	0,108	0,112	0,112	0,052	0,048	0,084	0,084	0,084	0,1035	0,1051	0,1061
3	0,112	0,144	0,144	0,144	0,068	0,104	0,112	0,112	0,112	0,1180	0,1103	0,1060	0,1	0,136	0,136	0,136	0,08	0,112	0,1	0,1	0,1	0,1036	0,1	0,0979
4	0,092	0,124	0,096	0,096	0,08	0,084	0,084	0,084	0,084	0,0835	0,0815	0,0804	0,092	0,092	0,092	0,092	0,088	0,088	0,092	0,092	0,092	0,092	0,092	0,092
5	0,104	0,176	0,176	0,176	0,068	0,108	0,096	0,096	0,096	0,1012	0,0964	0,0937	0,072	0,136	0,168	0,168	0,068	0,108	0,072	0,072	0,072	0,0938	0,0910	0,0895
6	0,104	0,104	0,104	0,104	0,08	0,112	0,104	0,104	0,104	0,104	0,104	0,104	0,104	0,104	0,104	0,104	0,08	0,092	0,108	0,108	0,108	0,104	0,104	0,104
7	0,12	0,124	0,124	0,124	0,076	0,116	0,1	0,1	0,1	0,1074	0,1015	0,0980	0,12	0,12	0,12	0,12	0,1	0,1	0,096	0,096	0,096	0,1099	0,1033	0,0995
8	0,1	0,088	0,128	0,128	0,06	0,08	0,096	0,096	0,096	0,0925	0,0854	0,0814	0,116	0,088	0,088	0,088	0,056	0,072	0,116	0,116	0,116	0,095	0,0872	0,0828
9	0,132	0,144	0,128	0,128	0,092	0,12	0,116	0,116	0,116	0,1246	0,1232	0,1224	0,164	0,156	0,156	0,156	0,092	0,092	0,128	0,128	0,128	0,1369	0,1285	0,1237
10	0,12	0,108	0,108	0,108	0,084	0,088	0,064	0,096	0,096	0,1172	0,1228	0,1259	0,088	0,124	0,124	0,124	0,076	0,088	0,088	0,088	0,088	0,1076	0,1076	0,1076
11	0,144	0,136	0,136	0,136	0,104	0,124	0,144	0,144	0,144	0,1195	0,1122	0,1082	0,148	0,148	0,148	0,148	0,108	0,148	0,128	0,128	0,128	0,1195	0,1122	0,1082
12	0,108	0,1	0,108	0,108	0,08	0,088	0,108	0,108	0,108	0,1163	0,1218	0,125	0,116	0,116	0,116	0,116	0,084	0,088	0,116	0,116	0,116	0,116	0,116	0,116
13	0,124	0,124	0,124	0,124	0,096	0,104	0,124	0,124	0,124	0,124	0,124	0,124	0,112	0,112	0,112	0,112	0,084	0,112	0,088	0,088	0,088	0,112	0,112	0,112
14	0,144	0,144	0,144	0,144	0,1	0,096	0,144	0,144	0,144	0,144	0,144	0,144	0,148	0,148	0,148	0,148	0,072	0,116	0,148	0,148	0,148	0,148	0,148	0,148
15	0,076	0,076	0,076	0,076	0,076	0,092	0,076	0,076	0,076	0,076	0,076	0,076	0,08	0,12	0,108	0,108	0,084	0,088	0,096	0,096	0,096	0,08	0,08	0,08
16	0,084	0,084	0,084	0,084	0,068	0,064	0,084	0,084	0,084	0,084	0,084	0,084	0,072	0,072	0,072	0,072	0,076	0,068	0,072	0,072	0,072	0,072	0,072	0,072
17	0,072	0,08	0,08	0,08	0,068	0,076	0,072	0,072	0,072	0,072	0,072	0,072	0,104	0,088	0,088	0,088	0,068	0,084	0,104	0,104	0,104	0,104	0,104	0,104
18	0,104	0,116	0,116	0,116	0,092	0,104	0,132	0,144	0,144	0,104	0,104	0,104	0,116	0,12	0,12	0,12	0,104	0,116	0,14	0,14	0,14	0,116	0,116	0,116
19	0,116	0,14	0,128	0,128	0,096	0,124	0,144	0,168	0,168	0,1225	0,1163	0,1128	0,128	0,136	0,128	0,128	0,092	0,108	0,144	0,16	0,16	0,125	0,1181	0,1142
20	0,104	0,104	0,104	0,104	0,056	0,088	0,084	0,084	0,084	0,0915	0,0862	0,0832	0,092	0,104	0,116	0,116	0,072	0,088	0,108	0,108	0,108	0,0987	0,1034	0,1060
21	0,112	0,052	0,052	0,052	0,068	0,1	0,1	0,1	0,1	0,0867	0,0844	0,0832	0,092	0,064	0,064	0,064	0,064	0,088	0,088	0,116	0,116	0,092	0,092	0,092
22	0,076	0,076	0,076	0,076	0,064	0,084	0,076	0,076	0,076	0,076	0,076	0,076	0,096	0,096	0,096	0,096	0,068	0,084	0,1	0,1	0,1	0,096	0,096	0,096
23	0,092	0,116	0,108	0,108	0,072	0,088	0,116	0,116	0,116	0,1064	0,1075	0,1081	0,092	0,092	0,092	0,092	0,068	0,12	0,104	0,104	0,104	0,092	0,092	0,092
24	0,104	0,128	0,128	0,128	0,076	0,08	0,108	0,108	0,108	0,0897	0,0887	0,0881	0,104	0,132	0,132	0,132	0,076	0,096	0,124	0,124	0,124	0,0922	0,0905	0,0896
25	0,148	0,14	0,148	0,148	0,124	0,136	0,132	0,132	0,132	0,148	0,148	0,148	0,144	0,144	0,144	0,144	0,128	0,128	0,144	0,144	0,144	0,144	0,144	0,144
26	0,128	0,112	0,116	0,116	0,08	0,104	0,132	0,132	0,132	0,1053	0,1060	0,1063	0,128	0,128	0,128	0,128	0,084	0,116	0,128	0,128	0,128	0,1102	0,1095	0,1091
27	0,108	0,108	0,108	0,108	0,068	0,108	0,128	0,152	0,152	0,1	0,0964	0,0945	0,112	0,096	0,096	0,096	0,072	0,12	0,12	0,152	0,152	0,0926	0,0912	0,0904
28	0,1	0,124	0,112	0,112	0,068	0,096	0,092	0,092	0,092	0,0963	0,0931	0,0912	0,1	0,12	0,108	0,108	0,084	0,108	0,12	0,12	0,12	0,0963	0,0931	0,0912
29	0,12	0,12	0,12	0,12	0,104	0,132	0,096	0,096	0,096	0,1243	0,1133	0,1070	0,136	0,144	0,144	0,144	0,1	0,096	0,104	0,104	0,104	0,104	0,1370	0,1226	0,1143
30	0,112	0,112	0,12	0,12	0,084	0,112	0,112	0,112	0,112	0,1194	0,1209	0,1218	0,104	0,104	0,104	0,104	0,08	0,104	0,104	0,104	0,104	0,104	0,1293	0,1281	0,1274

Tabela A.2 – Erros de Classificação da Base (1) para 1.000 registros (continuação).

Fold	Base 1																							
	1.000 registros																							
	Redes Neurais, 10 Neurônios, BPROP												Redes Neurais, 10 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,08	0,1	0,116	0,116	0,06	0,076	0,104	0,104	0,104	0,1037	0,1018	0,1007	0,08	0,124	0,132	0,132	0,06	0,092	0,124	0,124	0,124	0,1063	0,1074	0,1080
2	0,06	0,112	0,112	0,112	0,064	0,068	0,068	0,068	0,068	0,06	0,06	0,06	0,068	0,108	0,092	0,092	0,056	0,068	0,072	0,072	0,072	0,0883	0,0922	0,0944
3	0,1	0,1	0,1	0,1	0,076	0,116	0,1	0,1	0,1	0,1108	0,1017	0,0966	0,108	0,108	0,108	0,108	0,076	0,092	0,108	0,108	0,108	0,108	0,108	0,108
4	0,092	0,108	0,108	0,108	0,084	0,072	0,076	0,076	0,076	0,092	0,092	0,092	0,084	0,124	0,124	0,124	0,072	0,064	0,088	0,088	0,088	0,084	0,084	0,084
5	0,12	0,12	0,12	0,12	0,072	0,084	0,12	0,12	0,12	0,0962	0,0928	0,0909	0,08	0,08	0,08	0,08	0,068	0,108	0,08	0,08	0,08	0,0938	0,0938	0,0938
6	0,108	0,108	0,108	0,108	0,084	0,108	0,108	0,108	0,108	0,108	0,108	0,108	0,108	0,112	0,112	0,112	0,084	0,092	0,092	0,084	0,084	0,108	0,108	0,108
7	0,096	0,136	0,116	0,116	0,108	0,1	0,12	0,12	0,12	0,0971	0,0921	0,0891	0,112	0,104	0,136	0,136	0,104	0,096	0,112	0,112	0,112	0,112	0,112	0,112
8	0,084	0,084	0,084	0,084	0,076	0,096	0,084	0,084	0,084	0,084	0,084	0,084	0,088	0,088	0,088	0,088	0,068	0,08	0,088	0,088	0,088	0,088	0,088	0,088
9	0,128	0,112	0,112	0,112	0,112	0,12	0,104	0,104	0,104	0,1271	0,125	0,1237	0,124	0,116	0,104	0,104	0,092	0,116	0,136	0,124	0,124	0,1026	0,1026	0,1026
10	0,104	0,092	0,104	0,104	0,068	0,076	0,08	0,084	0,084	0,104	0,104	0,104	0,104	0,096	0,092	0,092	0,064	0,068	0,076	0,08	0,08	0,1004	0,1004	0,1004
11	0,108	0,108	0,108	0,108	0,116	0,14	0,136	0,136	0,136	0,1097	0,1052	0,1027	0,14	0,108	0,128	0,128	0,096	0,14	0,14	0,14	0,14	0,1024	0,0929	0,0876
12	0,104	0,08	0,08	0,08	0,084	0,104	0,088	0,12	0,12	0,0792	0,0698	0,0646	0,104	0,112	0,112	0,112	0,064	0,108	0,096	0,12	0,12	0,1188	0,1188	0,1188
13	0,108	0,104	0,104	0,104	0,084	0,068	0,084	0,12	0,12	0,1122	0,1122	0,1122	0,116	0,084	0,084	0,084	0,092	0,096	0,084	0,084	0,084	0,1047	0,1032	0,1024
14	0,128	0,108	0,136	0,136	0,084	0,084	0,1	0,1	0,1	0,1209	0,1194	0,1186	0,128	0,144	0,128	0,128	0,08	0,108	0,124	0,124	0,124	0,1234	0,1234	0,1234
15	0,092	0,08	0,088	0,088	0,076	0,08	0,1	0,1	0,1	0,1014	0,1038	0,1051	0,096	0,104	0,084	0,084	0,084	0,104	0,112	0,112	0,112	0,1159	0,1176	0,1185
16	0,096	0,076	0,076	0,076	0,068	0,068	0,068	0,084	0,084	0,0843	0,0793	0,0765	0,104	0,084	0,084	0,084	0,076	0,072	0,076	0,108	0,108	0,104	0,104	0,104
17	0,104	0,1	0,1	0,1	0,068	0,08	0,088	0,088	0,088	0,0792	0,0676	0,0609	0,08	0,08	0,08	0,08	0,064	0,072	0,076	0,076	0,076	0,08	0,08	0,08
18	0,104	0,116	0,116	0,116	0,1	0,088	0,108	0,14	0,14	0,104	0,104	0,104	0,104	0,112	0,112	0,112	0,1	0,088	0,104	0,104	0,104	0,104	0,104	0,104
19	0,128	0,128	0,128	0,128	0,1	0,116	0,132	0,132	0,132	0,128	0,128	0,128	0,128	0,128	0,132	0,132	0,112	0,136	0,128	0,128	0,128	0,128	0,128	0,128
20	0,076	0,076	0,076	0,076	0,064	0,076	0,076	0,076	0,076	0,076	0,076	0,076	0,096	0,092	0,092	0,092	0,056	0,084	0,096	0,096	0,096	0,0698	0,0698	0,0698
21	0,08	0,076	0,084	0,084	0,064	0,092	0,1	0,104	0,104	0,08	0,08	0,08	0,1	0,08	0,096	0,096	0,056	0,096	0,088	0,116	0,116	0,1	0,1	0,1
22	0,088	0,068	0,068	0,068	0,06	0,084	0,088	0,088	0,088	0,088	0,088	0,088	0,096	0,104	0,104	0,104	0,064	0,092	0,096	0,096	0,096	0,096	0,096	0,096
23	0,124	0,124	0,124	0,124	0,076	0,092	0,072	0,072	0,072	0,124	0,124	0,124	0,104	0,096	0,096	0,096	0,068	0,1	0,104	0,104	0,104	0,104	0,104	0,104
24	0,108	0,108	0,108	0,108	0,088	0,088	0,084	0,084	0,084	0,0947	0,0947	0,0947	0,088	0,1	0,096	0,096	0,092	0,096	0,092	0,116	0,116	0,0847	0,0851	0,0853
25	0,136	0,116	0,116	0,116	0,128	0,128	0,136	0,136	0,136	0,136	0,136	0,136	0,164	0,1	0,1	0,1	0,12	0,136	0,136	0,136	0,136	0,164	0,164	0,164
26	0,108	0,12	0,112	0,112	0,08	0,116	0,108	0,108	0,108	0,1004	0,1024	0,1035	0,128	0,12	0,112	0,112	0,084	0,104	0,12	0,12	0,12	0,1200	0,1166	0,1146
27	0,092	0,092	0,092	0,092	0,084	0,08	0,092	0,092	0,092	0,092	0,092	0,092	0,092	0,112	0,112	0,112	0,076	0,096	0,092	0,092	0,092	0,092	0,092	0,092
28	0,08	0,088	0,088	0,088	0,076	0,076	0,08	0,08	0,08	0,0843	0,0844	0,0845	0,088	0,092	0,092	0,092	0,068	0,076	0,088	0,088	0,088	0,0795	0,0810	0,0818
29	0,12	0,1	0,1	0,1	0,092	0,092	0,14	0,12	0,12	0,1319	0,1189	0,1114	0,108	0,1	0,104	0,104	0,104	0,1	0,128	0,12	0,12	0,1065	0,1003	0,0967
30	0,124	0,1	0,1	0,1	0,08	0,084	0,116	0,124	0,124	0,1343	0,1299	0,1274	0,12	0,096	0,112	0,112	0,084	0,104	0,12	0,12	0,12	0,1218	0,1209	0,1203

Fold	Base 1																							
	1.000 registros																							
	Redes Neurais, 20 Neurônios, BPROP												Redes Neurais, 20 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,084	0,104	0,108	0,108	0,064	0,08	0,104	0,104	0,104	0,084	0,084	0,084	0,088	0,088	0,092	0,092	0,076	0,088	0,088	0,088	0,088	0,0886	0,0886	0,0886
2	0,072	0,076	0,076	0,076	0,056	0,064	0,076	0,076	0,076	0,0707	0,0719	0,0726	0,08	0,08	0,08	0,08	0,056	0,064	0,08	0,08	0,08	0,08	0,08	0,08
3	0,088	0,1	0,1	0,1	0,088	0,088	0,108	0,108	0,108	0,0963	0,0862	0,0805	0,096	0,12	0,12	0,12	0,084	0,104	0,084	0,084	0,084	0,0987	0,0896	0,0845
4	0,072	0,072	0,072	0,072	0,072	0,076	0,072	0,072	0,072	0,072	0,072	0,072	0,08	0,096	0,096	0,096	0,064	0,084	0,084	0,084	0,084	0,08	0,08	0,08
5	0,124	0,12	0,108	0,108	0,096	0,108	0,124	0,124	0,124	0,1160	0,1071	0,1020	0,128	0,112	0,092	0,092	0,088	0,084	0,104	0,144	0,144	0,1012	0,0964	0,0937
6	0,096	0,112	0,112	0,112	0,072	0,068	0,096	0,096	0,096	0,096	0,096	0,096	0,112	0,112	0,112	0,112	0,064	0,08	0,112	0,112	0,112	0,112	0,112	0,112
7	0,112	0,112	0,112	0,112	0,112	0,092	0,112	0,112	0,112	0,112	0,112	0,112	0,124	0,1	0,108	0,108	0,104	0,096	0,104	0,104	0,104	0,1099	0,1033	0,0995
8	0,108	0,1	0,064	0,064	0,084	0,092	0,108	0,108	0,108	0,0875	0,0875	0,0875	0,104	0,108	0,088	0,088	0,076	0,076	0,084	0,084	0,084	0,085	0,0818	0,08
9	0,12	0,108	0,104	0,104	0,088	0,12	0,12	0,12	0,12	0,12	0,12	0,12	0,136	0,112	0,112	0,112	0,084	0,1	0,112	0,112	0,112	0,1393	0,1355	0,1334
10	0,088	0,068	0,068	0,068	0,064	0,076	0,088	0,088	0,088	0,0909	0,0909	0,0909	0,096	0,096	0,096	0,096	0,056	0,1	0,096	0,096	0,096	0,096	0,096	0,096
11	0,1	0,108	0,112	0,112	0,104	0,124	0,104	0,108	0,108	0,0853	0,0842	0,0835	0,112	0,116	0,116	0,116	0,108	0,124	0,112	0,112	0,112	0,1121	0,1017	0,0958
12	0,112	0,092	0,104	0,104	0,088	0,088	0,116	0,104	0,104	0,112	0,112	0,112	0,1	0,108	0,104	0,104	0,088	0,084	0,084	0,084	0,084	0,1	0,1	0,1
13	0,092	0,092	0,092	0,092	0,084	0,092	0,092	0,092	0,092	0,092	0,092	0,092	0,096	0,08	0,08	0,08	0,08	0,092	0,1	0,128	0,128	0,1271	0,1304	0,1322
14	0,124	0,104	0,144	0,144	0,092	0,088	0,116	0,136	0,136	0,124	0,124	0,124	0,128	0,12	0,112	0,112	0,088	0,128	0,108	0,16	0,16	0,128	0,128	0,128
15	0,092	0,116	0,112	0,112	0,072	0,088	0,108	0,108	0,108	0,1062	0,1107	0,1132	0,108	0,1	0,088	0,088	0,084	0,1	0,1	0,1	0,1	0,108	0,108	0,108
16	0,092	0,068	0,076	0,076	0,068	0,092	0,092	0,092	0,092	0,092	0,092	0,092	0,084	0,06	0,072	0,072	0,068	0,072	0,084	0,084	0,084	0,084	0,084	0,084
17	0,076	0,08	0,08	0,08	0,072	0,092	0,076	0,076	0,076	0,0818	0,0818	0,0818	0,108	0,08	0,104	0,104	0,052	0,076	0,096	0,096	0,096	0,0716	0,0657	0,0624
18	0,136	0,108	0,096	0,096	0,1	0,108	0,132	0,132	0,132	0,136	0,136	0,136	0,132	0,132	0,132	0,132	0,104	0,12	0,132	0,132	0,132	0,132	0,132	0,132
19	0,128	0,136	0,152	0,152	0,104	0,108	0,128	0,128	0,128	0,128	0,128	0,128	0,12	0,12	0,12	0,12	0,112	0,108	0,12	0,12	0,12	0,12	0,12	0,12
20	0,1	0,072	0,072	0,072	0,068	0,08	0,1	0,1	0,1	0,1	0,1	0,1	0,092	0,092	0,092	0,092	0,06	0,08	0,092	0,092	0,092	0,092	0,092	0,092
21	0,084	0,08	0,08	0,08	0,064	0,072	0,084	0,084	0,084	0,084	0,084	0,084	0,076	0,092	0,112	0,112	0,06	0,068	0,076	0,076	0,076	0,076	0,076	0,076
22	0,084	0,1	0,1	0,1	0,068	0,072	0,084	0,084	0,084	0,084	0,084	0,084	0,084	0,108	0,108	0,108	0,052	0,116	0,088	0,088	0,088	0,084	0,084	0,084
23	0,1	0,1	0,1	0,1	0,06	0,08	0,1	0,1	0,1	0,1	0,1	0,1	0,104	0,1	0,088	0,088	0,064	0,084	0,092	0,092	0,092	0,104	0,104	0,104
24	0,096	0,096	0,096	0,096	0,1	0,1	0,096	0,096	0,096	0,096	0,096	0,096	0,116	0,116	0,116	0,116	0,088	0,112	0,116	0,116	0,116	0,116	0,116	0,116
25	0,152	0,152	0,152	0,152	0,112	0,132	0,152	0,152	0,152	0,152	0,152	0,152	0,116	0,116	0,116	0,116	0,112	0,128	0,14	0,14	0,14	0,116	0,116	0,116
26	0,116	0,116	0,116	0,116	0,092	0,124	0,116	0,116	0,116	0,1127	0,1113	0,1104	0,1	0,096	0,108	0,108	0,092	0,116	0,108	0,108	0,108	0,1	0,1	0,1
27	0,076	0,096	0,1	0,1	0,076	0,092	0,076	0,076	0,076	0,076	0,076	0,076	0,116	0,108	0,108	0,108	0,084	0,116	0,116	0,116	0,116	0,116	0,116	0,116
28	0,068	0,092	0,092	0,092	0,06	0,084	0,076	0,076	0,076	0,0819	0,0827	0,0832	0,084	0,1	0,1	0,1	0,068	0,068	0,076	0,076	0,076	0,084	0,084	0,084
29	0,116	0,116	0,104	0,104	0,1	0,084	0,136	0,136	0,136	0,1065	0,1003	0,0967	0,104	0,104	0,104	0,104	0,08	0,096	0,104	0,104	0,104	0,104	0,104	0,104
30	0,108	0,108	0,108	0,108	0,076	0,1	0,108	0,108	0,108	0,108	0,108	0,108	0,1	0,096	0,12	0,12	0,084	0,1	0,1	0,112	0,112	0,1	0,1	0,1

Tabela A.3 – Erros de Classificação da Base (1) para 3.000 registros.

Fold	Base 1																							
	3.000 registros																							
	Regressão Linear												Regressão Logística											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,1946	0,0893	0,0853	0,0853	0,1853	0,1906	0,1066	0,1106	0,1106	0,2188	0,1288	0,1207	0,1053	0,0786	0,064	0,064	0,1026	0,1013	0,1093	0,1093	0,1093	0,1008	0,0983	0,0969
2	0,1906	0,096	0,088	0,088	0,1613	0,1613	0,1053	0,1053	0,1053	0,2230	0,1223	0,1150	0,116	0,0866	0,068	0,068	0,1053	0,1053	0,116	0,116	0,116	0,0973	0,0973	0,0973
3	0,1426	0,084	0,088	0,088	0,148	0,1373	0,1173	0,1093	0,1093	0,1569	0,1185	0,1157	0,1026	0,0773	0,0773	0,0773	0,0986	0,1013	0,1026	0,1026	0,1026	0,1026	0,1026	0,1026
4	0,2026	0,0946	0,0906	0,0906	0,2053	0,204	0,124	0,124	0,124	0,2879	0,1439	0,1386	0,1253	0,1026	0,0773	0,0773	0,1306	0,1266	0,1253	0,1253	0,1253	0,1253	0,1253	0,1253
5	0,1946	0,088	0,0773	0,0773	0,188	0,1933	0,1226	0,1226	0,1226	0,1207	0,1356	0,1265	0,1386	0,08	0,0746	0,0746	0,132	0,136	0,1253	0,1346	0,1346	0,1386	0,1386	0,1386
6	0,136	0,088	0,068	0,068	0,1186	0,1266	0,092	0,0906	0,0906	0,2019	0,1096	0,1037	0,0933	0,0693	0,068	0,068	0,0933	0,0933	0,0933	0,0933	0,0933	0,0932	0,0932	0,0932
7	0,148	0,0813	0,064	0,064	0,1373	0,1306	0,1	0,1	0,1	0,1211	0,1112	0,1052	0,1093	0,076	0,06	0,06	0,1066	0,112	0,1093	0,1093	0,1093	0,1093	0,1093	0,1093
8	0,1493	0,0906	0,0946	0,0946	0,1466	0,1413	0,112	0,112	0,112	0,1069	0,1040	0,0975	0,12	0,084	0,072	0,072	0,1226	0,124	0,1146	0,1146	0,1146	0,12	0,12	0,12
9	0,1466	0,1	0,092	0,092	0,1386	0,14	0,1106	0,112	0,112	0,1972	0,1322	0,1302	0,1146	0,0853	0,0893	0,0893	0,108	0,1093	0,1106	0,1066	0,1066	0,1196	0,1212	0,1216
10	0,2	0,104	0,084	0,084	0,184	0,1906	0,1293	0,1213	0,1213	0,2072	0,1239	0,1199	0,1266	0,0893	0,0933	0,0933	0,124	0,1213	0,1266	0,1266	0,1266	0,1111	0,1119	0,1104
11	0,1866	0,088	0,0866	0,0866	0,164	0,168	0,1133	0,1133	0,1133	0,2331	0,1222	0,1182	0,1173	0,076	0,076	0,076	0,1146	0,1186	0,116	0,088	0,088	0,1095	0,1095	0,1095
12	0,1653	0,0853	0,0813	0,0813	0,168	0,168	0,1066	0,12	0,12	0,1916	0,1157	0,1085	0,124	0,0813	0,0773	0,0773	0,1266	0,1266	0,1013	0,1093	0,1093	0,124	0,124	0,124
13	0,252	0,0933	0,0986	0,0986	0,228	0,2306	0,12	0,112	0,112	0,2442	0,1195	0,1149	0,112	0,0773	0,068	0,068	0,1066	0,1066	0,1133	0,1133	0,1133	0,112	0,112	0,112
14	0,1493	0,1	0,1106	0,1106	0,16	0,1653	0,0986	0,0893	0,0893	0,1493	0,1493	0,1493	0,0986	0,0893	0,0893	0,0893	0,1066	0,1013	0,096	0,1013	0,1013	0,1001	0,0990	0,0993
15	0,156	0,1186	0,1053	0,1053	0,1693	0,1626	0,1146	0,116	0,116	0,2183	0,1322	0,1262	0,112	0,092	0,0906	0,0906	0,1173	0,1133	0,1106	0,1133	0,1133	0,112	0,112	0,112
16	0,1693	0,0933	0,0906	0,0906	0,1533	0,156	0,108	0,1053	0,1053	0,1914	0,1111	0,1029	0,1106	0,0946	0,076	0,076	0,1026	0,1106	0,1026	0,1013	0,1013	0,1106	0,1106	0,1106
17	0,144	0,0893	0,092	0,092	0,1466	0,1493	0,128	0,1253	0,1253	0,1383	0,1528	0,1476	0,1213	0,1026	0,0653	0,0653	0,1266	0,1306	0,1213	0,1213	0,1213	0,1212	0,1235	0,1248
18	0,1826	0,1	0,088	0,088	0,188	0,188	0,1053	0,1213	0,1213	0,1233	0,1279	0,1209	0,1053	0,0866	0,0813	0,0813	0,1093	0,1053	0,1053	0,1186	0,1186	0,1073	0,0991	0,0982
19	0,1893	0,1013	0,1013	0,1013	0,1853	0,176	0,1293	0,1186	0,1186	0,1221	0,1213	0,1177	0,1173	0,0733	0,072	0,072	0,1146	0,112	0,12	0,1026	0,1026	0,1173	0,1173	0,1173
20	0,1013	0,0906	0,0746	0,0746	0,116	0,1146	0,1053	0,1053	0,1053	0,2164	0,1196	0,1156	0,104	0,08	0,076	0,076	0,1026	0,1026	0,1	0,1	0,1	0,1037	0,1037	0,1037
21	0,1293	0,08	0,072	0,072	0,148	0,1453	0,108	0,108	0,108	0,2284	0,1115	0,1011	0,1173	0,064	0,0626	0,0626	0,108	0,1053	0,1173	0,1173	0,1173	0,1173	0,1173	0,1173
22	0,1386	0,0906	0,088	0,088	0,1426	0,1373	0,0946	0,1093	0,1093	0,2146	0,1263	0,1206	0,0973	0,0706	0,0706	0,0706	0,1026	0,1066	0,0973	0,0973	0,0973	0,0940	0,0940	0,0940
23	0,1226	0,0733	0,076	0,076	0,1213	0,1226	0,12	0,104	0,104	0,1909	0,1275	0,1178	0,1306	0,0746	0,0746	0,0746	0,12	0,1213	0,1306	0,1306	0,1306	0,1306	0,1306	0,1306
24	0,1613	0,0866	0,0786	0,0786	0,1653	0,1666	0,092	0,1093	0,1093	0,1993	0,1116	0,1059	0,0973	0,0946	0,0813	0,0813	0,0933	0,0906	0,0906	0,0906	0,0906	0,0944	0,0944	0,0944
25	0,184	0,0813	0,096	0,096	0,176	0,1746	0,1386	0,1293	0,1293	0,1645	0,1372	0,1297	0,128	0,0893	0,076	0,076	0,1293	0,128	0,128	0,128	0,128	0,128	0,128	0,128
26	0,2453	0,0773	0,0773	0,0773	0,2306	0,1733	0,0986	0,0986	0,0986	0,1498	0,1156	0,1119	0,0973	0,088	0,08	0,08	0,1066	0,1066	0,104	0,1013	0,1013	0,0973	0,0973	0,0973
27	0,1946	0,1093	0,0853	0,0853	0,1693	0,1773	0,1213	0,1333	0,1333	0,1146	0,1225	0,1179	0,116	0,1	0,072	0,072	0,1226	0,1213	0,12	0,1186	0,1186	0,1104	0,1103	0,1083
28	0,136	0,0906	0,0853	0,0853	0,1466	0,1466	0,128	0,1306	0,1306	0,2081	0,1229	0,1181	0,124	0,092	0,08	0,08	0,1146	0,1186	0,124	0,124	0,124	0,124	0,124	0,124
29	0,184	0,0746	0,0746	0,0746	0,184	0,184	0,0973	0,0973	0,0973	0,1700	0,1060	0,1012	0,0933	0,0706	0,0666	0,0666	0,0946	0,096	0,096	0,096	0,096	0,0933	0,0933	0,0933
30	0,1813	0,0893	0,0706	0,0706	0,168	0,1666	0,1133	0,1133	0,1133	0,1338	0,1023	0,0960	0,1093	0,076	0,0746	0,0746	0,0986	0,1026	0,1106	0,1106	0,1106	0,1093	0,1093	0,1093

Fold	Base 1																							
	3.000 registros																							
	Redes Neurais, 3 Neurônios, BPROP												Redes Neurais, 3 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0866	0,076	0,076	0,076	0,0693	0,0666	0,0666	0,0666	0,0666	0,0772	0,0772	0,0772	0,0946	0,0786	0,064	0,064	0,0653	0,0666	0,0733	0,0733	0,0733	0,0846	0,0846	0,0846
2	0,068	0,0693	0,0746	0,0746	0,0613	0,076	0,076	0,0586	0,0586	0,068	0,068	0,068	0,0653	0,08	0,076	0,076	0,0613	0,0746	0,0813	0,0813	0,0813	0,0653	0,0653	0,0653
3	0,0786	0,0786	0,0786	0,0786	0,064	0,072	0,08	0,0826	0,0826	0,0786	0,0786	0,0786	0,0826	0,088	0,0906	0,0906	0,0666	0,0706	0,088	0,0866	0,0866	0,0997	0,0984	0,0977
4	0,084	0,084	0,084	0,084	0,064	0,0706	0,0733	0,0733	0,0733	0,084	0,084	0,084	0,0813	0,0813	0,0813	0,0813	0,056	0,0693	0,076	0,076	0,076	0,0813	0,0813	0,0813
5	0,0733	0,0693	0,08	0,08	0,0733	0,0693	0,0733	0,0733	0,0733	0,0733	0,0733	0,0733	0,0746	0,0746	0,0746	0,0746	0,0693	0,0733	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746
6	0,0746	0,0746	0,0746	0,0746	0,0733	0,0706	0,0733	0,0733	0,0733	0,0746	0,0746	0,0746	0,0786	0,0786	0,0786	0,0786	0,072	0,068	0,068	0,0813	0,0813	0,0786	0,0786	0,0786
7	0,072	0,072	0,072	0,072	0,0626	0,064	0,072	0,072	0,072	0,072	0,072	0,072	0,0666	0,0666	0,0666	0,0666	0,06	0,0653	0,0786	0,076	0,076	0,0666	0,0666	0,0666
8	0,096	0,096	0,096	0,096	0,072	0,0786	0,096	0,096	0,096	0,0868	0,0863	0,0860	0,1013	0,1013	0,1013	0,1013	0,068	0,0813	0,0946	0,0946	0,0946	0,1013	0,1013	0,1013
9	0,092	0,096	0,096	0,096	0,0853	0,0866	0,0813	0,0813	0,0813	0,1002	0,0968	0,0949	0,0866	0,092	0,092	0,092	0,076	0,0933	0,0813	0,0813	0,0813	0,0866	0,0866	0,0866
10	0,0986	0,0866	0,0866	0,0866	0,0666	0,0666	0,0853	0,0893	0,0893	0,0986	0,0986	0,0986	0,0973	0,096	0,0853	0,0853	0,0666	0,072	0,096	0,0973	0,0973	0,0973	0,0973	0,0973
11	0,0773	0,0786	0,0786	0,0786	0,06	0,0586	0,0666	0,0666	0,0666	0,0773	0,0773	0,0773	0,0826	0,084	0,1013	0,1013	0,0613	0,068	0,0613	0,0613	0,0613	0,0826	0,0826	0,0826
12	0,0853	0,084	0,0906	0,0906	0,0693	0,076	0,08	0,08	0,08	0,0791	0,0757	0,0738	0,0813	0,0866	0,1026	0,1026	0,068	0,0653	0,0813	0,0813	0,0813	0,0813	0,0813	0,0813
13	0,0706	0,08	0,0773	0,0773	0,0693	0,084	0,0706	0,0706	0,0706	0,0757	0,0750	0,0746	0,0706	0,0706	0,0706	0,0706	0,0653	0,0786	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706
14	0,0786	0,0946	0,0946	0,0946	0,0733	0,0813	0,0826	0,0826	0,0826	0,0837	0,0837	0,0837	0,0773	0,0773	0,0773	0,0773	0,0813	0,0733	0,0773	0,0773	0,0773	0,0773	0,0773	0,0773
15	0,08	0,072	0,072	0,072	0,072	0,0746	0,068	0,068	0,068	0,0755	0,0723	0,0705	0,076	0,076	0,076	0,076	0,08	0,0733	0,076	0,076	0,076	0,076	0,076	0,076
16	0,0813	0,0773	0,0773	0,0773	0,06	0,06	0,076	0,0826	0,0826	0,0813	0,0813	0,0813	0,096	0,08	0,072	0,072	0,056	0,064	0,072	0,08	0,08	0,096	0,096	0,096
17	0,0746	0,0866	0,0786	0,0786	0,0666	0,068	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746	0,0706	0,0706	0,0706	0,0706	0,0693	0,08	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706
18	0,068	0,068	0,068	0,068	0,0706	0,08	0,068	0,068	0,068	0,068	0,068	0,068	0,0653	0,0706	0,068	0,068	0,0666	0,08	0,076	0,0773	0,0773	0,0653	0,0653	0,0653
19	0,0653	0,0653	0,0653	0,0653	0,0653	0,0706	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,072	0,072	0,072	0,072	0,064	0,072	0,072	0,072	0,072	0,072	0,072	0,072
20	0,0906	0,0906	0,0893	0,0893	0,0746	0,0773	0,0906	0,0906	0,0906	0,0906	0,0906	0,0906	0,096	0,096	0,096	0,096	0,0733	0,0853	0,096	0,096	0,096	0,096	0,096	0,096
21	0,06	0,068	0,0546	0,0546	0,0506	0,0493	0,0573	0,0573	0,0573	0,0659	0,0656	0,0654	0,0653	0,0653	0,0706	0,0706	0,0546	0,0546	0,0573	0,0573	0,0573	0,0653	0,0653	0,0653
22	0,0693	0,0693	0,0693	0,0693	0,052	0,0773	0,0693	0,0693	0,0693	0,0693	0,0693	0,0693	0,0733	0,0733	0,0733	0,0733	0,064	0,0826	0,0866	0,0826	0,0826	0,0733	0,0733	0,0733
23	0,0826	0,1026	0,1026	0,1026	0,0653	0,0706	0,0826	0,0826	0,0826	0,0826	0,0826	0,0826	0,0773	0,0973	0,068	0,068	0,068	0,0706	0,076	0,0773	0,0773	0,0773	0,0773	0,0773
24	0,0933	0,092	0,084	0,084	0,0613	0,076	0,0733	0,0733	0,0733	0,0933	0,0933	0,0933	0,0906	0,0906	0,0906	0,0906	0,0613	0,084	0,0786	0,0786	0,0786	0,0906	0,0906	0,0906
25	0,08	0,072	0,0786	0,0786	0,0666	0,076	0,076	0,0773	0,0773	0,08	0,08	0,08	0,0906	0,0693	0,0826	0,0826	0,0613	0,0653	0,0786	0,0786	0,0786	0,0906	0,0906	0,0906
26	0,076	0,084	0,0706	0,0706	0,0653	0,0706	0,0786	0,072	0,072	0,076	0,076	0,076	0,092	0,0813	0,084	0,084	0,0586	0,0733	0,0866	0,0706	0,0706	0,092	0,092	0,092
27	0,0826	0,08	0,0826	0,0826	0,0653	0,068	0,0826	0,0826	0,0826	0,0826	0,0826	0,0826	0,076	0,076	0,076	0,076	0,068	0,068	0,076	0,076	0,076	0,076	0,076	0,076
28	0,0946	0,0946	0,0946	0,0946	0,0786	0,0826	0,0946	0,0946	0,0946	0,0946	0,0946	0,0946	0,0933	0,0933	0,0933	0,0933	0,084	0,076	0,088	0,0893	0,0893	0,0933	0,0933	0,0933
29	0,0693	0,0613	0,0733	0,0733	0,056	0,0626	0,0586	0,0586	0,0586	0,0693	0,0693	0,0693	0,0893	0,0653	0,072	0,072	0,0613	0,0693	0,0733	0,0653	0,0653	0,0893	0,0893	0,0893
30	0,0813	0,084	0,0706	0,0706	0,06	0,064	0,0626	0,0626	0,0626	0,0813	0,0813	0,0813	0,068	0,068	0,068	0,068	0,0626	0,0626	0,068	0,068	0,068	0,068	0,068	0,068

Tabela A.4 – Erros de Classificação da Base (1) para 3.000 registros (continuação).

Fold	Base 1																							
	3.000 registros																							
	Redes Neurais, 10 Neurônios, BPROP												Redes Neurais, 10 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,072	0,0706	0,0613	0,0613	0,056	0,072	0,072	0,072	0,072	0,072	0,072	0,072	0,0706	0,0706	0,0706	0,0706	0,0586	0,072	0,0706	0,0706	0,0706	0,0675	0,0679	0,0681
2	0,0653	0,0533	0,0533	0,0533	0,0613	0,0653	0,0653	0,0653	0,0653	0,0697	0,0716	0,0727	0,072	0,064	0,064	0,064	0,0626	0,0693	0,072	0,072	0,072	0,0681	0,0675	0,0673
3	0,0893	0,0853	0,0853	0,0853	0,0773	0,084	0,0946	0,1106	0,1106	0,0932	0,0943	0,0949	0,0813	0,0773	0,0853	0,0853	0,0813	0,092	0,1066	0,1066	0,1066	0,0813	0,0813	0,0813
4	0,0773	0,0786	0,072	0,072	0,0626	0,0786	0,0906	0,0906	0,0906	0,0773	0,0773	0,0773	0,0933	0,0826	0,072	0,072	0,0626	0,0706	0,0933	0,0933	0,0933	0,0913	0,0904	0,0899
5	0,0693	0,0693	0,0693	0,0693	0,0586	0,064	0,0693	0,0693	0,0693	0,0693	0,0693	0,0693	0,0666	0,0626	0,0666	0,0666	0,0573	0,0653	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666
6	0,08	0,084	0,088	0,088	0,0653	0,068	0,08	0,08	0,08	0,08	0,08	0,08	0,084	0,0866	0,0866	0,0866	0,064	0,076	0,0853	0,0946	0,0946	0,0850	0,0850	0,0850
7	0,064	0,0786	0,068	0,068	0,0666	0,0746	0,064	0,064	0,064	0,064	0,064	0,064	0,0666	0,072	0,072	0,072	0,0653	0,0733	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666
8	0,1026	0,0733	0,0733	0,0733	0,06	0,068	0,0906	0,0906	0,0906	0,0893	0,0845	0,0817	0,0773	0,0746	0,0746	0,0746	0,0586	0,0613	0,0773	0,0773	0,0773	0,0773	0,0773	0,0773
9	0,084	0,0746	0,0733	0,0733	0,0733	0,0733	0,108	0,092	0,092	0,0962	0,0939	0,0927	0,096	0,096	0,096	0,096	0,0693	0,0813	0,096	0,096	0,096	0,096	0,096	0,096
10	0,0933	0,0893	0,0893	0,0893	0,068	0,0626	0,0933	0,0933	0,0933	0,0933	0,0933	0,0933	0,084	0,084	0,084	0,084	0,0693	0,08	0,084	0,084	0,084	0,084	0,084	0,084
11	0,0866	0,0866	0,0866	0,0866	0,072	0,0786	0,0866	0,0866	0,0866	0,0866	0,0866	0,0866	0,0826	0,0826	0,0826	0,0826	0,0733	0,084	0,0826	0,0826	0,0826	0,0826	0,0826	0,0826
12	0,08	0,0786	0,072	0,072	0,064	0,0706	0,08	0,08	0,08	0,0741	0,0741	0,0741	0,076	0,084	0,0853	0,0853	0,068	0,0733	0,08	0,08	0,08	0,0675	0,0675	0,0675
13	0,08	0,0746	0,0693	0,0693	0,0613	0,0826	0,072	0,0746	0,0746	0,08	0,08	0,08	0,084	0,084	0,084	0,084	0,0613	0,0706	0,0733	0,0733	0,0733	0,084	0,084	0,084
14	0,084	0,0773	0,084	0,084	0,0733	0,0786	0,0733	0,0733	0,0733	0,084	0,084	0,084	0,0773	0,08	0,08	0,08	0,072	0,0813	0,0773	0,0773	0,0773	0,0773	0,0773	0,0773
15	0,0853	0,096	0,0853	0,0853	0,0786	0,0866	0,088	0,088	0,088	0,0853	0,0853	0,0853	0,0853	0,0973	0,0786	0,0786	0,0746	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853
16	0,0693	0,0693	0,0693	0,0693	0,052	0,068	0,0906	0,0826	0,0826	0,0693	0,0693	0,0693	0,0773	0,0653	0,0653	0,0653	0,0546	0,072	0,0826	0,0893	0,0893	0,0773	0,0773	0,0773
17	0,0813	0,0813	0,0813	0,0813	0,0653	0,0786	0,0813	0,0813	0,0813	0,0813	0,0813	0,0813	0,088	0,0853	0,0853	0,0853	0,072	0,1013	0,088	0,088	0,088	0,088	0,088	0,088
18	0,064	0,056	0,0653	0,0653	0,0546	0,0693	0,064	0,064	0,064	0,0687	0,0687	0,0687	0,0666	0,072	0,0706	0,0706	0,06	0,0773	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666
19	0,0986	0,0986	0,0986	0,0986	0,0666	0,084	0,0986	0,0986	0,0986	0,0986	0,0986	0,0986	0,1026	0,108	0,0986	0,0986	0,068	0,088	0,0986	0,0986	0,0986	0,1026	0,1026	0,1026
20	0,0893	0,0973	0,0973	0,0973	0,06	0,0586	0,076	0,076	0,076	0,0893	0,0893	0,0893	0,0826	0,084	0,084	0,084	0,068	0,0733	0,076	0,076	0,076	0,0826	0,0826	0,0826
21	0,052	0,052	0,052	0,052	0,0506	0,052	0,052	0,052	0,052	0,052	0,052	0,052	0,0586	0,0586	0,0586	0,0586	0,0493	0,0613	0,0573	0,0573	0,0573	0,0586	0,0586	0,0586
22	0,0853	0,076	0,0813	0,0813	0,0693	0,0706	0,084	0,0853	0,0853	0,0856	0,0846	0,0840	0,0933	0,0826	0,0826	0,0826	0,0626	0,0786	0,092	0,0906	0,0906	0,0915	0,0912	0,0911
23	0,076	0,0733	0,0733	0,0733	0,06	0,08	0,076	0,076	0,076	0,0636	0,0562	0,0521	0,0893	0,0813	0,0813	0,0813	0,0506	0,0746	0,0733	0,0733	0,0733	0,0893	0,0893	0,0893
24	0,0706	0,0706	0,0706	0,0706	0,0733	0,076	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706	0,084	0,084	0,084	0,084	0,0706	0,068	0,084	0,084	0,084	0,084	0,084	0,084
25	0,0773	0,084	0,084	0,084	0,064	0,08	0,0693	0,0693	0,0693	0,0773	0,0773	0,0773	0,0773	0,0733	0,0733	0,0733	0,068	0,08	0,084	0,084	0,084	0,0773	0,0773	0,0773
26	0,06	0,06	0,06	0,06	0,06	0,0773	0,06	0,06	0,06	0,06	0,06	0,06	0,08	0,084	0,0853	0,0853	0,0626	0,0666	0,08	0,08	0,08	0,08	0,08	0,08
27	0,0906	0,0773	0,0746	0,0746	0,0613	0,0773	0,0946	0,0946	0,0946	0,0906	0,0906	0,0906	0,0786	0,0813	0,08	0,08	0,0653	0,0813	0,0786	0,0786	0,0786	0,0786	0,0786	0,0786
28	0,0906	0,0906	0,0906	0,0906	0,0773	0,0853	0,0906	0,0906	0,0906	0,0906	0,0906	0,0906	0,088	0,08	0,0866	0,0866	0,084	0,076	0,0773	0,0773	0,0773	0,088	0,088	0,088
29	0,06	0,06	0,06	0,06	0,0493	0,048	0,06	0,06	0,06	0,06	0,06	0,06	0,0666	0,0666	0,0666	0,0666	0,0506	0,052	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666
30	0,0893	0,08	0,076	0,076	0,068	0,0773	0,0733	0,0733	0,0733	0,0893	0,0893	0,0893	0,0893	0,0826	0,0826	0,0826	0,0613	0,076	0,0853	0,0853	0,0853	0,0893	0,0893	0,0893

Fold	Base 1																							
	3.000 registros																							
	Redes Neurais, 20 Neurônios, BPROP												Redes Neurais, 20 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0746	0,0666	0,0666	0,0666	0,0586	0,0706	0,0746	0,0746	0,0746	0,0716	0,0661	0,0631	0,0848	0,0704	0,0704	0,0704	0,0646	0,072	0,0848	0,0848	0,0848	0,0765	0,0749	0,0732
2	0,0653	0,0693	0,0666	0,0666	0,0586	0,0626	0,072	0,072	0,072	0,0648	0,0646	0,0645	0,0700	0,0633	0,0583	0,0583	0,0604	0,0573	0,0653	0,0653	0,0653	0,0596	0,0608	0,0657
3	0,0853	0,0853	0,0853	0,0853	0,0733	0,0826	0,0786	0,0786	0,0786	0,0923	0,0925	0,0926	0,0773	0,0916	0,0916	0,0916	0,0650	0,084	0,0933	0,0933	0,0933	0,0956	0,0984	0,0978
4	0,0746	0,0773	0,08	0,08	0,0613	0,0653	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746	0,0755	0,0613	0,0781	0,0781	0,0609	0,0733	0,08	0,08	0,08	0,0755	0,0755	0,0755
5	0,0693	0,0706	0,0666	0,0666	0,06	0,0613	0,0693	0,0693	0,0693	0,0693	0,0693	0,0693	0,0696	0,0817	0,0718	0,0718	0,0674	0,072	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696
6	0,0826	0,08	0,0826	0,0826	0,0573	0,0626	0,0893	0,0893	0,0893	0,0826	0,0826	0,0826	0,0798	0,0775	0,0870	0,0870	0,0534	0,072	0,0866	0,0866	0,0866	0,0798	0,0798	0,0798
7	0,076	0,072	0,0786	0,0786	0,0613	0,076	0,068	0,068	0,068	0,076	0,076	0,076	0,0803	0,0795	0,0864	0,0864	0,0553	0,076	0,06	0,06	0,06	0,0803	0,0803	0,0803
8	0,084	0,0706	0,0733	0,0733	0,0573	0,0653	0,0906	0,0906	0,0906	0,084	0,084	0,084	0,0815	0,0808	0,0634	0,0634	0,0670	0,0626	0,0906	0,0906	0,0906	0,0815	0,0815	0,0815
9	0,0826	0,0946	0,0946	0,0946	0,0666	0,0653	0,0786	0,0786	0,0786	0,0905	0,0905	0,0905	0,0825	0,0994	0,0994	0,0994	0,0711	0,0733	0,0826	0,0826	0,0826	0,0831	0,0831	0,0831
10	0,092	0,0893	0,0786	0,0786	0,0653	0,0613	0,08	0,08	0,08	0,092	0,092	0,092	0,0986	0,0921	0,0916	0,0916	0,0663	0,0666	0,08	0,08	0,08	0,0986	0,0986	0,0986
11	0,0706	0,08	0,08	0,08	0,0706	0,08	0,0853	0,0786	0,0786	0,0821	0,0813	0,0808	0,0601	0,0911	0,0911	0,0911	0,0637	0,088	0,0601	0,0601	0,0601	0,0849	0,0907	0,0791
12	0,0733	0,0733	0,072	0,072	0,0613	0,0693	0,072	0,072	0,072	0,0733	0,0733	0,0733	0,0719	0,0647	0,0614	0,0614	0,0583	0,0733	0,0866	0,084	0,084	0,0719	0,0719	0,0719
13	0,0786	0,084	0,0826	0,0826	0,0666	0,0746	0,0813	0,0813	0,0813	0,0786	0,0786	0,0786	0,0863	0,0830	0,0783	0,0783	0,0690	0,072	0,084	0,084	0,084	0,0863	0,0863	0,0863
14	0,0733	0,08	0,088	0,088	0,072	0,0733	0,0733	0,0733	0,0733	0,0812	0,0806	0,0803	0,0690	0,0771	0,0938	0,0938	0,0744	0,0746	0,0693	0,0693	0,0693	0,0712	0,0775	0,0982
15	0,0786	0,0786	0,0786	0,0786	0,076	0,0826	0,0786	0,0786	0,0786	0,0786	0,0786	0,0786	0,0762	0,0762	0,0762	0,0762	0,0783	0,0706	0,0826	0,084	0,084	0,0762	0,0762	0,0762
16	0,072	0,068	0,0693	0,0693	0,0506	0,064	0,08	0,0853	0,0853	0,0635	0,0609	0,0594	0,0761	0,0718	0,0692	0,0692	0,0570	0,0626	0,08	0,08	0,08	0,0569	0,0492	0,0667
17	0,0786	0,08	0,08	0,08	0,068	0,068	0,0786	0,0786	0,0786	0,0781	0,0772	0,0768	0,0781	0,0752	0,0724	0,0724	0,0760	0,072	0,0781	0,0781	0,0781	0,0773	0,0706	0,0859
18	0,06	0,0706	0,0706	0,0706	0,0586	0,0706	0,06	0,06	0,06	0,0637	0,0637	0,0637	0,0645	0,0866	0,0934	0,0934	0,0451	0,0693	0,0645	0,0645	0,0645	0,0632	0,0632	0,0632
19	0,0853	0,0853	0,0853	0,0853	0,0653	0,08	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853	0,0888	0,0888	0,0888	0,0888	0,0558	0,0853	0,1013	0,1013	0,1013	0,0888	0,0888	0,0888
20	0,076	0,076	0,076	0,076	0,068	0,0693	0,076	0,076	0,076	0,076	0,076	0,076	0,0688	0,0688	0,0688	0,0688	0,0720	0,0773	0,0906	0,0906	0,0906	0,0688	0,0688	0,0688
21	0,0546	0,048	0,048	0,048	0,0533	0,048	0,0546	0,0546	0,0546	0,0546	0,0546	0,0546	0,0523	0,0444	0,0444	0,0444	0,0507	0,0573	0,0523	0,0523	0,0523	0,0523	0,0523	0,0523
22	0,0706	0,0746	0,076	0,076	0,0626	0,076	0,0866	0,0746	0,0746	0,0798	0,0798	0,0798	0,0807	0,0801	0,0843	0,0843	0,0734	0,0746	0,084	0,0866	0,0866	0,0800	0,0800	0,0800
23	0,076	0,072	0,072	0,072	0,0613	0,0653	0,076	0,076	0,076	0,076	0,076	0,076	0,0827	0,0734	0,0734	0,0734	0,0650	0,0626	0,0706	0,0706	0,0706	0,0827	0,0827	0,0827
24	0,0853	0,076	0,0786	0,0786	0,0613	0,068	0,0853	0,0853	0,0853	0,0799	0,0758	0,0735	0,0828	0,0819	0,0730	0,0730	0,0432	0,076	0,08	0,084	0,084	0,0737	0,0803	0,0751
25	0,064	0,0613	0,0586	0,0586	0,0626	0,0626	0,0733	0,0746	0,0746	0,064	0,064	0,064	0,0620	0,0617	0,0677	0,0677	0,0674	0,056	0,0620	0,0620	0,0620	0,0620	0,0620	0,0620
26	0,0813	0,0746	0,0813	0,0813	0,0613	0,0706	0,08	0,084	0,084	0,0813	0,0813	0,0813	0,0742	0,0802	0,0865	0,0865	0,0639	0,0693	0,0742	0,0742	0,0742	0,0742	0,0742	0,0742
27	0,0826	0,08	0,0773	0,0773	0,072	0,0746	0,0826	0,0826	0,0826	0,0826	0,0826	0,0826	0,0863	0,0721	0,0794	0,0794	0,0757	0,0853	0,0863	0,0863	0,0863	0,0863	0,0863	0,0863
28	0,0773	0,0893	0,0813	0,0813	0,072	0,076	0,0773	0,0773	0,0773	0,0773	0,0773	0,0773	0,0932	0,0953	0,0799	0,0799	0,0610	0,0853	0,0813	0,0813	0,0813	0,0932	0,0932	0,0932
29	0,0613	0,0586	0,0586	0,0586	0,0426	0,064	0,0573	0,0613	0,0613	0,0613	0,0613	0,0613	0,0628	0,0550	0,0550	0,0550	0,0308	0,0653	0,0826	0,0826	0,0826	0,0628	0,0628	0,0628
30	0,0826	0,0826	0,0826	0,0826	0,0626	0,0733	0,0826	0,0826	0,0826	0,0826	0,0826	0,0826	0,0905	0,0905	0,0905	0,0905	0,0524	0,068	0,0905	0,0905	0,0905	0,0905	0,0905	0,0905

Tabela A.5 – Erros de Classificação da Base (1) para 5.000 registros.

Fold	Base 1																							
	5.000 registros																							
	Regressão Linear												Regressão Logística											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,2608	0,0768	0,0664	0,0664	0,26	0,2608	0,1184	0,1096	0,1096	0,1795	0,1243	0,1183	0,1112	0,0712	0,0584	0,0584	0,1096	0,108	0,112	0,1016	0,1016	0,1112	0,1112	0,1112
2	0,1688	0,0816	0,076	0,076	0,1496	0,1624	0,1152	0,1136	0,1136	0,1836	0,1324	0,1263	0,1224	0,0784	0,0704	0,0704	0,1208	0,1192	0,116	0,1112	0,1112	0,1224	0,1224	0,1224
3	0,2	0,0832	0,076	0,076	0,1872	0,1912	0,1032	0,1032	0,1032	0,1020	0,1183	0,1116	0,1008	0,08	0,068	0,068	0,0992	0,0976	0,1	0,0984	0,0984	0,1008	0,1008	0,1008
4	0,1624	0,0904	0,0904	0,0904	0,1584	0,1552	0,1072	0,1072	0,1072	0,1156	0,1208	0,1162	0,1184	0,0896	0,0808	0,0808	0,1184	0,124	0,112	0,112	0,112	0,1184	0,1184	0,1184
5	0,1496	0,1112	0,0888	0,0888	0,1568	0,152	0,112	0,1152	0,1152	0,1355	0,1248	0,1171	0,1256	0,088	0,0784	0,0784	0,1224	0,124	0,1144	0,116	0,116	0,1256	0,1256	0,1256
6	0,156	0,0872	0,0792	0,0792	0,152	0,1544	0,1136	0,1024	0,1024	0,1426	0,1172	0,1124	0,1112	0,0792	0,0696	0,0696	0,1096	0,1112	0,1056	0,0952	0,0952	0,1112	0,1112	0,1112
7	0,1296	0,0784	0,076	0,076	0,1368	0,1336	0,0952	0,0888	0,0888	0,1516	0,1113	0,1059	0,096	0,084	0,0736	0,0736	0,0984	0,0992	0,0936	0,0984	0,0984	0,096	0,096	0,096
8	0,1728	0,0744	0,0704	0,0704	0,1744	0,1808	0,0936	0,0984	0,0984	0,2314	0,1091	0,1026	0,1	0,0784	0,0784	0,0784	0,0992	0,0968	0,0944	0,0856	0,0856	0,1	0,1	0,1
9	0,2	0,1008	0,0888	0,0888	0,1976	0,1976	0,1184	0,1128	0,1128	0,1234	0,1242	0,1216	0,1152	0,0944	0,0768	0,0768	0,1112	0,1128	0,112	0,112	0,112	0,1095	0,1095	0,1095
10	0,1944	0,0896	0,0808	0,0808	0,1688	0,1736	0,1064	0,1064	0,1064	0,2131	0,1123	0,1061	0,1112	0,072	0,064	0,064	0,1104	0,1112	0,1064	0,1064	0,1064	0,1041	0,0981	0,0950
11	0,1704	0,088	0,08	0,08	0,1888	0,1888	0,1264	0,1264	0,1264	0,2272	0,1347	0,1298	0,1192	0,0832	0,0808	0,0808	0,1168	0,1176	0,12	0,12	0,12	0,1145	0,1132	0,1130
12	0,1392	0,0984	0,0904	0,0904	0,136	0,1336	0,112	0,1168	0,1168	0,1781	0,1345	0,1281	0,1264	0,0856	0,0736	0,0736	0,1168	0,1152	0,1112	0,1104	0,1104	0,1264	0,1264	0,1264
13	0,2352	0,0816	0,076	0,076	0,24	0,2384	0,0984	0,0984	0,0984	0,1768	0,1136	0,1075	0,1008	0,0688	0,0608	0,0608	0,1	0,1	0,096	0,096	0,096	0,1008	0,1008	0,1008
14	0,3152	0,096	0,0832	0,0832	0,3128	0,316	0,1144	0,1144	0,1144	0,1131	0,1221	0,1158	0,1152	0,088	0,0808	0,0808	0,1176	0,1128	0,108	0,108	0,108	0,1152	0,1152	0,1152
15	0,1984	0,1008	0,0888	0,0888	0,1808	0,172	0,1152	0,1152	0,1152	0,1218	0,1324	0,1259	0,1104	0,0832	0,088	0,088	0,1104	0,1104	0,1096	0,1096	0,1096	0,1092	0,1092	0,1092
16	0,1632	0,0752	0,0768	0,0768	0,1704	0,168	0,1136	0,1128	0,1128	0,1067	0,1188	0,1120	0,112	0,0736	0,0728	0,0728	0,1144	0,1168	0,112	0,112	0,112	0,112	0,112	0,112
17	0,18	0,0896	0,0896	0,0896	0,164	0,16	0,1296	0,1296	0,1296	0,1136	0,1262	0,1193	0,1208	0,0864	0,0672	0,0672	0,1208	0,1216	0,1272	0,1272	0,1272	0,1111	0,1111	0,1111
18	0,1608	0,0856	0,0792	0,0792	0,164	0,1632	0,1144	0,1112	0,1112	0,1020	0,1099	0,1050	0,1144	0,0744	0,0632	0,0632	0,1112	0,112	0,1032	0,1032	0,1032	0,0971	0,0971	0,0971
19	0,2624	0,0832	0,0912	0,0912	0,264	0,2616	0,1128	0,112	0,112	0,1183	0,1346	0,1288	0,116	0,0752	0,076	0,076	0,1136	0,1144	0,1136	0,1136	0,1136	0,1123	0,1123	0,1123
20	0,1432	0,0928	0,0872	0,0872	0,1592	0,1552	0,1032	0,1032	0,1032	0,1054	0,1119	0,1068	0,1	0,0784	0,0728	0,0728	0,1056	0,104	0,1	0,1	0,1	0,1	0,1	0,1
21	0,156	0,0736	0,0672	0,0672	0,1552	0,1544	0,1064	0,1056	0,1056	0,1284	0,1224	0,1163	0,1128	0,0632	0,06	0,06	0,1056	0,108	0,0976	0,0976	0,0976	0,1115	0,1115	0,1115
22	0,1448	0,0848	0,0848	0,0848	0,1464	0,1456	0,112	0,1096	0,1096	0,1347	0,1214	0,1158	0,1104	0,0728	0,072	0,072	0,1072	0,1072	0,1064	0,1064	0,1064	0,1104	0,1104	0,1104
23	0,1472	0,0952	0,0824	0,0824	0,1584	0,1552	0,1256	0,1224	0,1224	0,1157	0,1178	0,1138	0,1224	0,096	0,0824	0,0824	0,1208	0,1224	0,12	0,1128	0,1128	0,1092	0,1092	0,1092
24	0,2856	0,1024	0,0944	0,0944	0,2632	0,2704	0,1192	0,1192	0,1192	0,1588	0,1290	0,1228	0,1136	0,0728	0,0832	0,0832	0,1152	0,1152	0,116	0,116	0,116	0,1121	0,1121	0,1121
25	0,1456	0,0888	0,0896	0,0896	0,1568	0,1568	0,1024	0,1048	0,1048	0,2449	0,1368	0,1305	0,1128	0,0728	0,072	0,072	0,1072	0,1064	0,108	0,0976	0,0976	0,1128	0,1128	0,1128
26	0,1968	0,072	0,0728	0,0728	0,1824	0,1824	0,0976	0,0976	0,0976	0,1368	0,1144	0,1090	0,092	0,076	0,0712	0,0712	0,0952	0,0984	0,0888	0,0888	0,0888	0,092	0,092	0,092
27	0,1344	0,0712	0,08	0,08	0,1344	0,1352	0,1072	0,1056	0,1056	0,2402	0,1187	0,1104	0,1088	0,064	0,064	0,064	0,104	0,1048	0,1008	0,0936	0,0936	0,1088	0,1088	0,1088
28	0,1496	0,0848	0,0816	0,0816	0,1416	0,1432	0,112	0,1056	0,1056	0,2421	0,1144	0,1087	0,1016	0,0712	0,0704	0,0704	0,1072	0,1056	0,1008	0,0952	0,0952	0,1016	0,1016	0,1016
29	0,1816	0,0792	0,0664	0,0664	0,1808	0,1816	0,1064	0,1056	0,1056	0,2434	0,1239	0,1183	0,1048	0,0752	0,068	0,068	0,0976	0,0992	0,1008	0,0992	0,0992	0,1048	0,1048	0,1048
30	0,2272	0,0816	0,0832	0,0832	0,2296	0,2328	0,1104	0,112	0,112	0,2105	0,1202	0,1143	0,1056	0,0704	0,0688	0,0688	0,1064	0,108	0,1064	0,1088	0,1088	0,1056	0,1056	0,1056

Fold	Base 1																							
	5.000 registros																							
	Redes Neurais, 3 Neurônios, BPROP												Redes Neurais, 3 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0816	0,0752	0,0664	0,0664	0,0616	0,0632	0,0728	0,0704	0,0704	0,0816	0,0816	0,0816	0,0896	0,0808	0,0768	0,0768	0,0624	0,0608	0,0648	0,0648	0,0648	0,0788	0,0772	0,0763
2	0,08	0,0576	0,0576	0,0576	0,0704	0,0776	0,0728	0,0744	0,0744	0,08	0,08	0,08	0,0768	0,064	0,0592	0,0592	0,064	0,0744	0,0736	0,0776	0,0776	0,0768	0,0768	0,0768
3	0,072	0,072	0,0792	0,0792	0,068	0,0768	0,0808	0,0784	0,0784	0,0728	0,0728	0,0728	0,0856	0,0704	0,0776	0,0776	0,068	0,08	0,0832	0,0736	0,0736	0,0856	0,0856	0,0856
4	0,0792	0,0784	0,072	0,072	0,0712	0,0776	0,0808	0,0808	0,0808	0,0716	0,0704	0,0697	0,0776	0,0776	0,0776	0,0776	0,0712	0,0784	0,0808	0,0808	0,0808	0,0776	0,0776	0,0776
5	0,0896	0,0648	0,0648	0,0648	0,076	0,08	0,0896	0,0896	0,0896	0,0726	0,0694	0,0676	0,0872	0,064	0,064	0,064	0,072	0,0792	0,0752	0,0768	0,0768	0,0706	0,0687	0,0676
6	0,0784	0,0704	0,0656	0,0656	0,0696	0,0736	0,0776	0,0776	0,0776	0,0693	0,0693	0,0693	0,0752	0,0648	0,0648	0,0648	0,0672	0,0752	0,0768	0,0768	0,0768	0,0752	0,0752	0,0752
7	0,0792	0,0792	0,0792	0,0792	0,0664	0,0696	0,072	0,072	0,072	0,0713	0,0688	0,0674	0,0944	0,0712	0,0712	0,0712	0,064	0,072	0,076	0,076	0,076	0,0753	0,0728	0,0714
8	0,0608	0,0664	0,0664	0,0664	0,06	0,0688	0,0608	0,0608	0,0608	0,0608	0,0608	0,0608	0,0608	0,0608	0,0608	0,0608	0,0584	0,0672	0,0608	0,0608	0,0608	0,0608	0,0608	0,0608
9	0,084	0,0648	0,0648	0,0648	0,0824	0,084	0,0816	0,0816	0,0816	0,084	0,084	0,084	0,0888	0,0808	0,0736	0,0736	0,0784	0,0672	0,0808	0,0824	0,0824	0,0888	0,0888	0,0888
10	0,072	0,08	0,08	0,08	0,0704	0,0744	0,0776	0,076	0,076	0,072	0,072	0,072	0,0752	0,0752	0,0752	0,0752	0,064	0,0728	0,0776	0,0736	0,0736	0,0648	0,0616	0,0598
11	0,0744	0,0816	0,0816	0,0816	0,0728	0,0808	0,0744	0,0744	0,0744	0,0749	0,0716	0,0697	0,0744	0,0672	0,0632	0,0632	0,0632	0,0856	0,0752	0,0752	0,0752	0,0744	0,0744	0,0744
12	0,072	0,0688	0,0792	0,0792	0,0696	0,0864	0,072	0,072	0,072	0,072	0,072	0,072	0,084	0,0792	0,0856	0,0856	0,064	0,0856	0,084	0,084	0,084	0,084	0,084	0,084
13	0,068	0,0672	0,0648	0,0648	0,0584	0,0648	0,0736	0,0736	0,0736	0,068	0,068	0,068	0,0576	0,0576	0,0576	0,0576	0,0608	0,0656	0,0576	0,0576	0,0576	0,0576	0,0576	0,0576
14	0,0832	0,0816	0,0824	0,0824	0,0712	0,076	0,0776	0,0776	0,0776	0,0782	0,0767	0,0758	0,084	0,0848	0,0688	0,0688	0,0744	0,0784	0,0768	0,0768	0,0768	0,084	0,084	0,084
15	0,092	0,092	0,0752	0,0752	0,0728	0,0864	0,092	0,092	0,092	0,0827	0,0832	0,0835	0,0944	0,096	0,0816	0,0816	0,0768	0,0808	0,0864	0,0864	0,0864	0,0944	0,0944	0,0944
16	0,068	0,0704	0,0704	0,0704	0,0624	0,064	0,068	0,068	0,068	0,068	0,068	0,068	0,0608	0,0608	0,0608	0,0608	0,0584	0,0544	0,064	0,064	0,064	0,0608	0,0608	0,0608
17	0,0784	0,0744	0,0864	0,0864	0,0728	0,0744	0,0688	0,0688	0,0688	0,0784	0,0784	0,0784	0,0632	0,088	0,088	0,088	0,0664	0,0696	0,0632	0,0632	0,0632	0,0632	0,0632	0,0632
18	0,0688	0,0656	0,064	0,064	0,056	0,0608	0,0624	0,0624	0,0624	0,0608	0,0579	0,0564	0,0608	0,06	0,056	0,056	0,0592	0,0624	0,0608	0,0608	0,0608	0,0583	0,0583	0,0583
19	0,0688	0,0784	0,0784	0,0784	0,0728	0,068	0,0688	0,0688	0,0688	0,0688	0,0688	0,0688	0,0744	0,0736	0,0736	0,0736	0,0656	0,0728	0,0648	0,0648	0,0648	0,0744	0,0744	0,0744
20	0,0776	0,0632	0,0648	0,0648	0,0632	0,0672	0,0776	0,0776	0,0776	0,0776	0,0776	0,0776	0,0728	0,056	0,0632	0,0632	0,064	0,0648	0,0752	0,0752	0,0752	0,0728	0,0728	0,0728
21	0,0704	0,0712	0,0736	0,0736	0,0624	0,0664	0,0616	0,056	0,056	0,0718	0,0725	0,0729	0,068	0,0656	0,0616	0,0616	0,0584	0,0616	0,0712	0,0712	0,0712	0,068	0,068	0,068
22	0,0696	0,0616	0,0656	0,0656	0,0624	0,0664	0,0704	0,0696	0,0696	0,0636	0,0601	0,0581	0,0792	0,0688	0,068	0,068	0,0608	0,064	0,0792	0,0792	0,0792	0,0711	0,0711	0,0711
23	0,0784	0,0784	0,0784	0,0784	0,0792	0,0792	0,0784	0,0784	0,0784	0,0784	0,0784	0,0784	0,0832	0,0712	0,0688	0,0688	0,072	0,0808	0,0784	0,0808	0,0808	0,0832	0,0832	0,0832
24	0,0728	0,0728	0,0728	0,0728	0,0632	0,0744	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728	0,0776	0,0784	0,0784	0,0784	0,0608	0,0872	0,076	0,076	0,076	0,0734	0,0729	0,0726
25	0,068	0,0808	0,0688	0,0688	0,0624	0,0728	0,0616	0,0584	0,0584	0,0742	0,0748	0,0752	0,0664	0,0688	0,0688	0,0688	0,0624	0,0688	0,0648	0,0616	0,0616	0,0747	0,0744	0,0743
26	0,064	0,0752	0,0712	0,0712	0,064	0,0688	0,0632	0,0632	0,0632	0,0618	0,0588	0,0571	0,0664	0,0664	0,0664	0,0664	0,06	0,0616	0,0664	0,0664	0,0664	0,0664	0,0664	0,0664
27	0,0728	0,0672	0,0544	0,0544	0,0584	0,0624	0,064	0,0568	0,0568	0,0663	0,0659	0,0657	0,072	0,0728	0,0728	0,0728	0,0552	0,064	0,0632	0,0632	0,0632	0,072	0,072	0,072
28	0,0728	0,0712	0,0712	0,0712	0,068	0,0672	0,0776	0,0744	0,0744	0,0693	0,0652	0,0629	0,0752	0,0696	0,0696	0,0696	0,0696	0,0592	0,0776	0,08	0,08	0,0752	0,0752	0,0752
29	0,0712	0,06	0,0576	0,0576	0,0632	0,0696	0,068	0,0688	0,0688	0,0587	0,0585	0,0584	0,0576	0,076	0,076	0,076	0,0592	0,0648	0,0664	0,0672	0,0672	0,0576	0,0576	0,0576
30	0,0696	0,0712	0,0712	0,0712	0,0632	0,0624	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696	0,0784	0,076	0,0744	0,0744	0,0568	0,0704	0,0712	0,0712	0,0712	0,0714	0,0714	0,0714

Tabela A.6 – Erros de Classificação da Base (1) para 5.000 registros (continuação).

Fold	Base 1																							
	5.000 registros																							
	Redes Neurais, 10 Neurônios, BPROP												Redes Neurais, 10 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0536	0,0656	0,068	0,068	0,0416	0,0464	0,0536	0,0536	0,0536	0,0536	0,0536	0,0536	0,0551	0,0569	0,0715	0,0715	0,0422	0,0540	0,068	0,0712	0,0712	0,0551	0,0551	0,0551
2	0,0616	0,0648	0,0648	0,0648	0,0472	0,0616	0,0616	0,0616	0,0616	0,0616	0,0616	0,0616	0,0619	0,0624	0,0624	0,0624	0,0525	0,0685	0,0619	0,0619	0,0619	0,0619	0,0619	0,0619
3	0,0768	0,0592	0,0608	0,0608	0,06	0,0552	0,0768	0,0768	0,0768	0,0768	0,0768	0,0768	0,0920	0,0553	0,0512	0,0512	0,0567	0,0443	0,0920	0,0920	0,0920	0,0920	0,0920	0,0920
4	0,0632	0,0784	0,0776	0,0776	0,0664	0,0744	0,0776	0,0776	0,0776	0,0632	0,0632	0,0632	0,0631	0,0894	0,0801	0,0801	0,0710	0,0691	0,0712	0,0712	0,0712	0,0790	0,0772	0,0762
5	0,0632	0,068	0,068	0,068	0,052	0,0592	0,0528	0,0632	0,0632	0,0609	0,0589	0,0577	0,0705	0,0627	0,0731	0,0731	0,0593	0,0665	0,0632	0,06	0,06	0,0705	0,0705	0,0705
6	0,0576	0,0648	0,0648	0,0648	0,0456	0,0496	0,06	0,056	0,056	0,0515	0,0515	0,0515	0,0574	0,0671	0,0671	0,0671	0,0458	0,0505	0,0574	0,0574	0,0574	0,0574	0,0574	0,0574
7	0,0832	0,0832	0,0832	0,0832	0,064	0,0704	0,0832	0,0832	0,0832	0,0832	0,0832	0,0832	0,0908	0,0908	0,0908	0,0908	0,0639	0,0645	0,0864	0,0784	0,0784	0,0823	0,0823	0,0823
8	0,0624	0,0624	0,0624	0,0624	0,0512	0,052	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624	0,0610	0,0610	0,0610	0,0610	0,0498	0,0558	0,0610	0,0610	0,0610	0,0610	0,0610	0,0610
9	0,0856	0,0712	0,0712	0,0712	0,0624	0,0656	0,0856	0,0856	0,0856	0,0856	0,0856	0,0856	0,0889	0,0711	0,0777	0,0777	0,0628	0,0708	0,0784	0,0784	0,0784	0,0889	0,0889	0,0889
10	0,0704	0,076	0,0784	0,0784	0,0648	0,0704	0,0704	0,0704	0,0704	0,0648	0,0648	0,0648	0,0612	0,0807	0,0715	0,0715	0,0596	0,0652	0,0792	0,0808	0,0808	0,0717	0,0659	0,0626
11	0,0576	0,0616	0,064	0,064	0,0496	0,0536	0,06	0,06	0,06	0,0576	0,0576	0,0576	0,0630	0,0642	0,0735	0,0735	0,0589	0,0546	0,0630	0,0630	0,0630	0,0630	0,0630	0,0630
12	0,0816	0,076	0,0752	0,0752	0,068	0,0648	0,0832	0,0832	0,0832	0,0816	0,0816	0,0816	0,0834	0,0772	0,0780	0,0780	0,0731	0,0663	0,0848	0,0856	0,0856	0,0863	0,0891	0,0908
13	0,06	0,0608	0,06	0,06	0,044	0,0576	0,0624	0,0624	0,0624	0,0627	0,0627	0,0627	0,0577	0,0544	0,0647	0,0647	0,0447	0,0550	0,0632	0,0632	0,0632	0,0607	0,0604	0,0601
14	0,068	0,0792	0,0696	0,0696	0,0624	0,0608	0,068	0,068	0,068	0,068	0,068	0,068	0,0649	0,0884	0,0665	0,0665	0,0544	0,0660	0,0744	0,0744	0,0744	0,0649	0,0649	0,0649
15	0,0752	0,0752	0,0752	0,0752	0,0712	0,0672	0,0728	0,0728	0,0728	0,0752	0,0752	0,0752	0,0719	0,0719	0,0719	0,0719	0,0736	0,0741	0,076	0,076	0,076	0,0736	0,0736	0,0736
16	0,0688	0,0648	0,0648	0,0648	0,0544	0,0576	0,0688	0,0688	0,0688	0,0688	0,0688	0,0688	0,0742	0,0639	0,0639	0,0639	0,0612	0,0481	0,0696	0,0696	0,0696	0,0742	0,0742	0,0742
17	0,0672	0,0752	0,068	0,068	0,0584	0,0608	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672	0,0588	0,0779	0,0656	0,0656	0,0609	0,0664	0,0588	0,0588	0,0588	0,0588	0,0588	0,0588
18	0,0568	0,0568	0,0568	0,0568	0,0496	0,0528	0,0568	0,0568	0,0568	0,0568	0,0568	0,0568	0,0553	0,0553	0,0553	0,0553	0,0523	0,0581	0,06	0,06	0,06	0,0553	0,0553	0,0553
19	0,0736	0,0712	0,0776	0,0776	0,0592	0,0736	0,0736	0,0736	0,0736	0,0736	0,0736	0,0736	0,0808	0,0734	0,0775	0,0775	0,0607	0,0767	0,0776	0,0776	0,0776	0,0808	0,0808	0,0808
20	0,0704	0,064	0,056	0,056	0,0584	0,0608	0,0608	0,0608	0,0608	0,0704	0,0704	0,0704	0,0780	0,0808	0,0613	0,0613	0,0551	0,0626	0,0780	0,0780	0,0780	0,0780	0,0780	0,0780
21	0,0672	0,0656	0,0688	0,0688	0,0544	0,0568	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672	0,0588	0,0636	0,0668	0,0668	0,0507	0,0557	0,0588	0,0588	0,0588	0,0588	0,0588	0,0588
22	0,0656	0,0656	0,0656	0,0656	0,06	0,0624	0,0656	0,0656	0,0656	0,0656	0,0656	0,0656	0,0718	0,0718	0,0718	0,0718	0,0664	0,0577	0,0784	0,0784	0,0784	0,0718	0,0718	0,0718
23	0,0744	0,072	0,072	0,072	0,056	0,0576	0,0568	0,0568	0,0568	0,0744	0,0744	0,0744	0,0766	0,0679	0,0679	0,0679	0,0560	0,0562	0,0766	0,0766	0,0766	0,0766	0,0766	0,0766
24	0,072	0,072	0,072	0,072	0,0584	0,0536	0,072	0,072	0,072	0,072	0,072	0,072	0,0737	0,0737	0,0737	0,0737	0,0674	0,0561	0,0737	0,0737	0,0737	0,0737	0,0737	0,0737
25	0,0736	0,0736	0,0736	0,0736	0,0528	0,0648	0,0696	0,0736	0,0736	0,0736	0,0736	0,0736	0,0664	0,0664	0,0664	0,0664	0,0523	0,0622	0,0728	0,0736	0,0736	0,0757	0,0752	0,0749
26	0,0768	0,068	0,068	0,068	0,0576	0,0584	0,0736	0,0736	0,0736	0,0768	0,0768	0,0768	0,0752	0,0740	0,0740	0,0740	0,0544	0,0627	0,0752	0,0752	0,0752	0,0752	0,0752	0,0752
27	0,0544	0,0544	0,0544	0,0544	0,0472	0,0488	0,0544	0,0544	0,0544	0,0544	0,0544	0,0544	0,0626	0,0626	0,0626	0,0626	0,0501	0,0505	0,0626	0,0626	0,0626	0,0626	0,0626	0,0626
28	0,0744	0,0672	0,068	0,068	0,0576	0,0632	0,0776	0,0816	0,0816	0,0744	0,0744	0,0744	0,0745	0,0661	0,0694	0,0694	0,0493	0,0667	0,0745	0,0745	0,0745	0,0745	0,0745	0,0745
29	0,0488	0,0568	0,0568	0,0568	0,04	0,0496	0,0568	0,0584	0,0584	0,0488	0,0488	0,0488	0,0344	0,0548	0,0548	0,0548	0,0411	0,0585	0,0624	0,064	0,064	0,0344	0,0344	0,0344
30	0,064	0,0608	0,0608	0,0608	0,0552	0,0536	0,064	0,064	0,064	0,0630	0,0619	0,0612	0,0593	0,0664	0,0664	0,0664	0,0529	0,0506	0,0576	0,0576	0,0576	0,0675	0,0647	0,0632

Fold	Base 1																							
	5.000 registros																							
	Redes Neurais, 20 Neurônios, BPROP												Redes Neurais, 20 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0531	0,0630	0,0719	0,0719	0,0468	0,0496	0,0531	0,0531	0,0531	0,0531	0,0531	0,0531	0,0530	0,0688	0,0663	0,0663	0,0516	0,0464	0,0530	0,0530	0,0530	0,0530	0,0530	0,0530
2	0,0653	0,0787	0,0787	0,0787	0,0511	0,0617	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,0675	0,0764	0,0764	0,0764	0,0511	0,0717	0,0675	0,0675	0,0675	0,0675	0,0675	0,0675
3	0,0812	0,0563	0,0618	0,0618	0,0632	0,0477	0,0812	0,0812	0,0812	0,0812	0,0812	0,0812	0,0714	0,0607	0,0602	0,0602	0,0634	0,0517	0,0714	0,0714	0,0714	0,0714	0,0714	0,0714
4	0,0629	0,0776	0,0901	0,0901	0,0696	0,0684	0,064	0,064	0,064	0,0629	0,0629	0,0629	0,0531	0,0820	0,0885	0,0885	0,0708	0,0666	0,076	0,076	0,076	0,0531	0,0531	0,0531
5	0,0591	0,0760	0,0731	0,0731	0,0622	0,0603	0,0632	0,0616	0,0616	0,0599	0,0599	0,0599	0,0488	0,0826	0,0739	0,0739	0,0664	0,0655	0,0488	0,0488	0,0488	0,0694	0,0694	0,0694
6	0,0573	0,0620	0,0620	0,0620	0,0477	0,0461	0,0573	0,0573	0,0573	0,0573	0,0573	0,0573	0,0563	0,0619	0,0619	0,0619	0,0443	0,0348	0,0576	0,0576	0,0576	0,0563	0,0563	0,0563
7	0,0904	0,0904	0,0904	0,0904	0,0621	0,0671	0,0824	0,08	0,08	0,0904	0,0904	0,0904	0,0931	0,0931	0,0931	0,0931	0,0573	0,0592	0,0896	0,0808	0,0808	0,0931	0,0931	0,0931
8	0,0522	0,0522	0,0522	0,0522	0,0457	0,0609	0,06	0,06	0,06	0,0585	0,0579	0,0576	0,0549	0,0549	0,0549	0,0549	0,0473	0,0554	0,072	0,0704	0,0704	0,0548	0,0636	0,0567
9	0,0846	0,0710	0,0769	0,0769	0,0596	0,0722	0,0846	0,0846	0,0846	0,0846	0,0846	0,0846	0,0793	0,0701	0,0821	0,0821	0,0643	0,0658	0,0792	0,0792	0,0792	0,0793	0,0793	0,0793
10	0,0597	0,0814	0,0810	0,0810	0,0731	0,0756	0,0808	0,0808	0,0808	0,0638	0,0588	0,0559	0,0544	0,0805	0,0861	0,0861	0,0831	0,0800	0,0736	0,0736	0,0736	0,0668	0,0654	0,0568
11	0,0546	0,0607	0,0691	0,0691	0,0454	0,0547	0,0546	0,0546	0,0546	0,0546	0,0546	0,0546	0,0554	0,0597	0,0781	0,0781	0,0483	0,0507	0,0648	0,0648	0,0648	0,0554	0,0554	0,0554
12	0,0900	0,0721	0,0832	0,0832	0,0731	0,0551	0,0900	0,0900	0,0900	0,0900	0,0900	0,0900	0,0905	0,0613	0,0875	0,0875	0,0659	0,0572	0,0784	0,0784	0,0784	0,0905	0,0905	0,0905
13	0,0694	0,0677	0,0502	0,0502	0,0352	0,0551	0,0584	0,0584	0,0584	0,0694	0,0694	0,0694	0,0637	0,0699	0,0409	0,0409	0,0439	0,0445	0,0637	0,0637	0,0637	0,0637	0,0637	0,0637
14	0,0668	0,0766	0,0610	0,0610	0,0602	0,0526	0,0664	0,0664	0,0664	0,0668	0,0668	0,0668	0,0751	0,0809	0,0551	0,0551	0,0550	0,0504	0,0751	0,0751	0,0751	0,0751	0,0751	0,0751
15	0,0840	0,0840	0,0840	0,0840	0,0722	0,0690	0,08	0,08	0,08	0,0840	0,0840	0,0840	0,0830	0,0830	0,0830	0,0830	0,0709	0,0862	0,0768	0,0768	0,0768	0,0830	0,0830	0,0830
16	0,0670	0,0650	0,0650	0,0650	0,0518	0,0594	0,0736	0,0736	0,0736	0,0670	0,0670	0,0670	0,0750	0,0619	0,0619	0,0619	0,0425	0,0577	0,0752	0,0752	0,0752	0,0750	0,0750	0,0750
17	0,0696	0,0709	0,0706	0,0706	0,0535	0,0582	0,072	0,072	0,072	0,0696	0,0696	0,0696	0,0777	0,0678	0,0684	0,0684	0,0509	0,0553	0,0777	0,0777	0,0777	0,0777	0,0777	0,0777
18	0,0373	0,0373	0,0373	0,0373	0,0450	0,0541	0,0373	0,0373	0,0373	0,0373	0,0373	0,0373	0,0359	0,0359	0,0359	0,0359	0,0403	0,0365	0,0584	0,0584	0,0584	0,0359	0,0359	0,0359
19	0,0607	0,0766	0,0707	0,0707	0,0543	0,0738	0,0607	0,0607	0,0607	0,0607	0,0607	0,0607	0,0593	0,0742	0,0744	0,0744	0,0496	0,0777	0,0593	0,0593	0,0593	0,0593	0,0593	0,0593
20	0,0710	0,0606	0,0709	0,0709	0,0566	0,0532	0,06	0,06	0,06	0,0688	0,0670	0,0660	0,0730	0,0552	0,0719	0,0719	0,0641	0,0469	0,0632	0,0632	0,0632	0,0616	0,0646	0,0677
21	0,0714	0,0629	0,0647	0,0647	0,0511	0,0597	0,0714	0,0714	0,0714	0,0714	0,0714	0,0714	0,0734	0,0575	0,0657	0,0657	0,0547	0,0614	0,0734	0,0734	0,0734	0,0734	0,0734	0,0734
22	0,0608	0,0608	0,0608	0,0608	0,0610	0,0559	0,0728	0,0728	0,0728	0,0608	0,0608	0,0608	0,0652	0,0652	0,0652	0,0652	0,0540	0,0528	0,0652	0,0652	0,0652	0,0652	0,0652	0,0652
23	0,0725	0,0781	0,0781	0,0781	0,0625	0,0544	0,0672	0,076	0,076	0,0623	0,0634	0,0640	0,0695	0,0743	0,0743	0,0743	0,0704	0,0548	0,0648	0,0664	0,0664	0,0700	0,0751	0,0707
24	0,0767	0,0767	0,0767	0,0767	0,0629	0,0637	0,0767	0,0767	0,0767	0,0578	0,0578	0,0578	0,0816	0,0816	0,0816	0,0816	0,0604	0,0600	0,0816	0,0816	0,0816	0,0537	0,0537	0,0537
25	0,0743	0,0743	0,0743	0,0743	0,0479	0,0706	0,0776	0,0776	0,0776	0,0697	0,0705	0,0710	0,0624	0,0624	0,0624	0,0624	0,0439	0,0659	0,0672	0,0672	0,0672	0,0759	0,0677	0,0583
26	0,0800	0,0676	0,0676	0,0676	0,0594	0,0568	0,0800	0,0800	0,0800	0,0800	0,0800	0,0800	0,0806	0,0683	0,0683	0,0683	0,0659	0,0630	0,0806	0,0806	0,0806	0,0806	0,0806	0,0806
27	0,0446	0,0446	0,0446	0,0446	0,0539	0,0528	0,0446	0,0446	0,0446	0,0446	0,0446	0,0446	0,0489	0,0489	0,0489	0,0489	0,0593	0,0464	0,052	0,052	0,052	0,0489	0,0489	0,0489
28	0,0686	0,0701	0,0659	0,0659	0,0518	0,0701	0,0736	0,0752	0,0752	0,0686	0,0686	0,0686	0,0612	0,0752	0,0717	0,0717	0,0530	0,0562	0,0612	0,0612	0,0612	0,0612	0,0612	0,0612
29	0,0507	0,0559	0,0559	0,0559	0,0406	0,0490	0,0616	0,0616	0,0616	0,0523	0,0517	0,0514	0,0522	0,0545	0,0545	0,0545	0,0553	0,0532	0,0584	0,0584	0,0584	0,0549	0,0486	0,0545
30	0,0660	0,0639	0,0639	0,0639	0,0640	0,0538	0,0660	0,0660	0,0660	0,0655	0,0655	0,0655	0,0636	0,0656	0,0656	0,0656	0,0654	0,0557	0,06	0,06	0,06	0,0638	0,0638	0,0638

Tabela A.7 – Erros de Classificação da Base (2) para 1.000 registros.

Fold	Base 2																							
	1.000 registros																							
	Regressão Linear												Regressão Logística											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,184	0,136	0,128	0,128	0,192	0,184	0,108	0,108	0,108	0,1965	0,1063	0,1067	0,136	0,14	0,128	0,128	0,132	0,132	0,136	0,136	0,136	0,1031	0,1031	0,1031
2	0,268	0,164	0,112	0,112	0,26	0,264	0,164	0,156	0,156	0,2125	0,12	0,1114	0,156	0,148	0,084	0,084	0,156	0,172	0,148	0,124	0,124	0,1225	0,1109	0,1042
3	0,148	0,156	0,156	0,156	0,156	0,156	0,152	0,152	0,152	0,1707	0,0985	0,0912	0,152	0,148	0,16	0,16	0,156	0,156	0,152	0,152	0,152	0,1113	0,0931	0,0870
4	0,176	0,116	0,108	0,108	0,172	0,172	0,092	0,112	0,112	0,2315	0,1175	0,1089	0,108	0,108	0,12	0,12	0,104	0,104	0,092	0,108	0,108	0,0966	0,0932	0,0898
5	0,172	0,196	0,18	0,18	0,176	0,18	0,188	0,172	0,172	0,2857	0,1460	0,1405	0,176	0,168	0,176	0,176	0,172	0,176	0,176	0,176	0,176	0,1301	0,1329	0,1301
6	0,148	0,128	0,16	0,16	0,148	0,152	0,148	0,148	0,148	0,1751	0,1059	0,0997	0,144	0,144	0,116	0,116	0,16	0,16	0,144	0,144	0,144	0,1091	0,0929	0,0894
7	0,112	0,124	0,132	0,132	0,112	0,112	0,124	0,132	0,132	0,112	0,112	0,112	0,116	0,108	0,108	0,108	0,112	0,116	0,124	0,152	0,152	0,0964	0,1040	0,1099
8	0,144	0,076	0,076	0,076	0,148	0,148	0,116	0,1	0,1	0,1959	0,1007	0,0951	0,136	0,08	0,08	0,08	0,128	0,124	0,136	0,136	0,136	0,0954	0,0954	0,0954
9	0,216	0,12	0,096	0,096	0,212	0,212	0,128	0,128	0,128	0,1875	0,0909	0,0871	0,144	0,124	0,124	0,124	0,14	0,136	0,152	0,152	0,152	0,0925	0,0872	0,0842
10	0,144	0,132	0,14	0,14	0,14	0,144	0,132	0,132	0,132	0,1782	0,1218	0,1179	0,144	0,132	0,14	0,14	0,148	0,148	0,132	0,132	0,132	0,1212	0,1164	0,1137
11	0,168	0,18	0,172	0,172	0,168	0,176	0,18	0,152	0,152	0,1703	0,1053	0,1048	0,164	0,148	0,152	0,152	0,16	0,172	0,168	0,172	0,172	0,1086	0,1086	0,1086
12	0,196	0,148	0,128	0,128	0,196	0,184	0,16	0,144	0,144	0,2340	0,1343	0,1251	0,164	0,136	0,148	0,148	0,152	0,144	0,156	0,148	0,148	0,1297	0,1156	0,1104
13	0,2	0,14	0,128	0,128	0,196	0,204	0,16	0,144	0,144	0,2044	0,1032	0,0953	0,132	0,136	0,132	0,132	0,128	0,128	0,144	0,144	0,144	0,1172	0,0942	0,0881
14	0,216	0,172	0,184	0,184	0,216	0,216	0,192	0,192	0,192	0,3002	0,1828	0,1708	0,184	0,16	0,164	0,164	0,172	0,176	0,18	0,18	0,18	0,1832	0,1716	0,1620
15	0,132	0,116	0,104	0,104	0,136	0,136	0,128	0,116	0,116	0,132	0,132	0,132	0,132	0,12	0,092	0,092	0,136	0,136	0,132	0,112	0,112	0,1072	0,0960	0,0896
16	0,128	0,128	0,136	0,136	0,128	0,12	0,136	0,112	0,112	0,1658	0,1053	0,1003	0,108	0,124	0,116	0,116	0,12	0,12	0,108	0,108	0,108	0,1191	0,1191	0,1191
17	0,104	0,104	0,112	0,112	0,112	0,112	0,124	0,1	0,1	0,2170	0,1263	0,1205	0,128	0,104	0,108	0,108	0,12	0,12	0,132	0,104	0,104	0,1146	0,1070	0,1054
18	0,168	0,132	0,14	0,14	0,164	0,164	0,156	0,136	0,136	0,2461	0,1379	0,1276	0,152	0,14	0,124	0,124	0,152	0,148	0,16	0,132	0,132	0,1243	0,1187	0,1124
19	0,164	0,136	0,136	0,136	0,16	0,164	0,132	0,14	0,14	0,2371	0,1320	0,1348	0,164	0,152	0,152	0,152	0,168	0,172	0,164	0,164	0,164	0,164	0,164	0,164
20	0,26	0,152	0,16	0,16	0,232	0,236	0,26	0,26	0,26	0,2311	0,1410	0,1340	0,172	0,148	0,16	0,16	0,172	0,168	0,172	0,172	0,172	0,1281	0,1172	0,1152
21	0,152	0,128	0,148	0,148	0,156	0,156	0,14	0,14	0,14	0,2879	0,1363	0,1274	0,172	0,16	0,12	0,12	0,16	0,164	0,172	0,172	0,172	0,172	0,172	0,172
22	0,184	0,14	0,148	0,148	0,184	0,192	0,16	0,16	0,16	0,1970	0,1459	0,1406	0,172	0,148	0,148	0,148	0,176	0,168	0,148	0,14	0,14	0,1256	0,1227	0,1225
23	0,144	0,14	0,14	0,14	0,144	0,148	0,132	0,132	0,132	0,2153	0,1415	0,1516	0,148	0,132	0,132	0,132	0,144	0,144	0,132	0,132	0,132	0,148	0,148	0,148
24	0,208	0,144	0,14	0,14	0,196	0,196	0,12	0,136	0,136	0,2107	0,0852	0,0779	0,156	0,144	0,148	0,148	0,148	0,152	0,12	0,148	0,148	0,0694	0,0681	0,0644
25	0,276	0,116	0,096	0,096	0,268	0,264	0,108	0,088	0,088	0,1871	0,1174	0,1114	0,132	0,108	0,092	0,092	0,148	0,14	0,132	0,132	0,132	0,0960	0,0960	0,0947
26	0,148	0,132	0,116	0,116	0,144	0,148	0,148	0,148	0,148	0,2061	0,1287	0,1237	0,124	0,108	0,136	0,136	0,124	0,12	0,136	0,136	0,136	0,1145	0,1119	0,1104
27	0,124	0,148	0,112	0,112	0,136	0,132	0,112	0,108	0,108	0,2036	0,1501	0,1475	0,128	0,136	0,12	0,12	0,128	0,132	0,1	0,12	0,12	0,1237	0,1349	0,1355
28	0,184	0,14	0,148	0,148	0,184	0,184	0,14	0,172	0,172	0,2297	0,1309	0,1279	0,116	0,128	0,128	0,128	0,124	0,124	0,128	0,128	0,128	0,0959	0,1051	0,1075
29	0,232	0,16	0,14	0,14	0,228	0,22	0,172	0,172	0,172	0,1795	0,1358	0,1322	0,192	0,164	0,172	0,172	0,188	0,184	0,168	0,168	0,168	0,1396	0,1376	0,1337
30	0,156	0,112	0,08	0,08	0,144	0,144	0,108	0,084	0,084	0,1820	0,0978	0,0938	0,132	0,116	0,092	0,092	0,132	0,132	0,116	0,096	0,096	0,0847	0,0833	0,0825

Fold	Base 2																							
	1.000 registros																							
	Redes Neurais, 3 Neurônios, BPROP												Redes Neurais, 3 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,128	0,096	0,096	0,096	0,084	0,088	0,112	0,084	0,084	0,0859	0,0780	0,0735	0,132	0,116	0,136	0,136	0,076	0,108	0,104	0,088	0,088	0,0958	0,0851	0,0790
2	0,128	0,12	0,088	0,088	0,092	0,112	0,128	0,128	0,128	0,0875	0,0875	0,0875	0,092	0,124	0,124	0,124	0,084	0,088	0,092	0,092	0,092	0,09	0,09	0,09
3	0,112	0,132	0,092	0,092	0,128	0,124	0,112	0,112	0,112	0,1113	0,1021	0,0969	0,112	0,12	0,12	0,12	0,116	0,104	0,112	0,112	0,112	0,0940	0,0940	0,0940
4	0,068	0,076	0,076	0,076	0,068	0,068	0,064	0,104	0,104	0,068	0,068	0,068	0,072	0,088	0,088	0,088	0,064	0,064	0,072	0,072	0,072	0,072	0,072	0,072
5	0,1	0,156	0,156	0,156	0,116	0,104	0,128	0,128	0,128	0,1	0,1	0,1	0,1	0,128	0,172	0,172	0,116	0,12	0,1	0,1	0,1	0,1	0,1	0,1
6	0,1	0,092	0,092	0,092	0,084	0,096	0,1	0,1	0,1	0,1065	0,0929	0,0850	0,1	0,1	0,1	0,1	0,08	0,08	0,1	0,1	0,1	0,1	0,1	0,1
7	0,068	0,092	0,096	0,096	0,076	0,076	0,088	0,084	0,084	0,0964	0,1003	0,1026	0,08	0,096	0,08	0,08	0,076	0,076	0,08	0,088	0,088	0,0989	0,1003	0,1011
8	0,116	0,088	0,088	0,088	0,08	0,064	0,1	0,1	0,1	0,116	0,116	0,116	0,1	0,088	0,116	0,116	0,06	0,06	0,088	0,088	0,088	0,1	0,1	0,1
9	0,096	0,092	0,112	0,112	0,088	0,08	0,084	0,084	0,084	0,095	0,0818	0,0742	0,096	0,112	0,088	0,088	0,072	0,096	0,104	0,104	0,104	0,09	0,0745	0,0657
10	0,112	0,108	0,104	0,104	0,104	0,124	0,128	0,124	0,124	0,112	0,112	0,112	0,108	0,116	0,116	0,116	0,088	0,116	0,152	0,124	0,124	0,1064	0,1075	0,1081
11	0,136	0,136	0,136	0,136	0,096	0,104	0,132	0,132	0,132	0,136	0,136	0,136	0,112	0,144	0,14	0,14	0,088	0,104	0,12	0,12	0,12	0,1061	0,1061	0,1061
12	0,1	0,116	0,116	0,116	0,096	0,088	0,108	0,108	0,108	0,1	0,1	0,1	0,088	0,088	0,088	0,088	0,104	0,14	0,104	0,104	0,104	0,088	0,088	0,088
13	0,128	0,116	0,116	0,116	0,104	0,112	0,128	0,128	0,128	0,1022	0,1022	0,1022	0,112	0,112	0,112	0,112	0,104	0,108	0,112	0,112	0,112	0,112	0,112	0,112
14	0,132	0,132	0,132	0,132	0,12	0,12	0,132	0,132	0,132	0,132	0,132	0,132	0,148	0,148	0,148	0,148	0,116	0,144	0,148	0,148	0,148	0,148	0,148	0,148
15	0,06	0,08	0,08	0,08	0,076	0,1	0,088	0,088	0,088	0,06	0,06	0,06	0,068	0,072	0,072	0,072	0,052	0,088	0,088	0,08	0,08	0,0822	0,0833	0,0839
16	0,108	0,104	0,132	0,132	0,1	0,124	0,108	0,108	0,108	0,108	0,108	0,108	0,116	0,104	0,128	0,128	0,092	0,116	0,104	0,104	0,104	0,116	0,116	0,116
17	0,088	0,092	0,076	0,076	0,072	0,08	0,08	0,096	0,096	0,0951	0,0951	0,0951	0,072	0,088	0,096	0,096	0,072	0,076	0,104	0,104	0,104	0,072	0,072	0,072
18	0,14	0,104	0,104	0,104	0,112	0,112	0,14	0,14	0,14	0,14	0,14	0,14	0,156	0,156	0,156	0,156	0,112	0,108	0,152	0,136	0,136	0,156	0,156	0,156
19	0,1	0,1	0,1	0,1	0,084	0,096	0,1	0,1	0,1	0,1	0,1	0,1	0,112	0,112	0,112	0,112	0,088	0,104	0,112	0,112	0,112	0,112	0,112	0,112
20	0,104	0,076	0,076	0,076	0,116	0,108	0,104	0,104	0,104	0,1005	0,0970	0,0951	0,128	0,108	0,076	0,076	0,092	0,096	0,128	0,128	0,128	0,1030	0,1030	0,1030
21	0,108	0,12	0,12	0,12	0,088	0,108	0,108	0,108	0,108	0,108	0,108	0,108	0,116	0,116	0,116	0,116	0,104	0,096	0,116	0,116	0,116	0,116	0,116	0,116
22	0,08	0,08	0,08	0,08	0,084	0,084	0,08	0,08	0,08	0,08	0,08	0,08	0,096	0,096	0,092	0,092	0,088	0,088	0,096	0,096	0,096	0,096	0,096	0,096
23	0,116	0,112	0,16	0,16	0,116	0,116	0,116	0,116	0,116	0,1163	0,1200	0,1221	0,108	0,108	0,108	0,108	0,108	0,104	0,112	0,112	0,112	0,1163	0,1163	0,1163
24	0,108	0,116	0,116	0,116	0,112	0,112	0,108	0,108	0,108	0,0668	0,0625	0,0599	0,112	0,108	0,116	0,116	0,108	0,104	0,112	0,112	0,112	0,0745	0,0681	0,0644
25	0,088	0,084	0,092	0,092	0,072	0,084	0,088	0,088	0,088	0,088	0,088	0,088	0,096	0,064	0,064	0,064	0,068	0,096	0,096	0,096	0,096	0,096	0,096	0,096
26	0,112	0,12	0,096	0,096	0,088	0,116	0,1	0,1	0,1	0,0966	0,0970	0,0972	0,104	0,092	0,096	0,096	0,1	0,08	0,092	0,092	0,092	0,1170	0,1175	0,1178
27	0,128	0,1	0,12	0,12	0,076	0,096	0,128	0,128	0,128	0,1134	0,1197	0,1234	0,152	0,1	0,128	0,128	0,08	0,084	0,12	0,14	0,14	0,1262	0,1292	0,1310
28	0,092	0,1	0,1	0,1	0,092	0,088	0,124	0,124	0,124	0,1111	0,1051	0,1017	0,104	0,148	0,148	0,148	0,092	0,116	0,104	0,104	0,104	0,1212	0,1212	0,1212
29	0,112	0,112	0,112	0,112	0,104	0,096	0,112	0,112	0,112	0,112	0,112	0,112	0,104	0,104	0,104	0,104	0,092	0,112	0,104	0,104	0,104	0,104	0,104	0,104
30	0,088	0,06	0,092	0,092	0,064	0,088	0,108	0,088	0,088	0,0798	0,0760	0,0739	0,096	0,092	0,068	0,068	0,064	0,076	0,096	0,096	0,096	0,0847	0,0797	0,0768

Tabela A.8 – Erros de Classificação da Base (2) para 1.000 registros (continuação).

Fold	Base 2																							
	1.000 registros																							
	Redes Neurais, 10 Neurônios, BPROP												Redes Neurais, 10 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,112	0,12	0,12	0,12	0,08	0,088	0,104	0,104	0,104	0,0810	0,0744	0,0707	0,104	0,132	0,128	0,128	0,076	0,2047	0,104	0,104	0,104	0,0761	0,0709	0,0679
2	0,116	0,112	0,112	0,112	0,1	0,132	0,116	0,116	0,116	0,116	0,116	0,116	0,116	0,124	0,124	0,124	0,116	0,1854	0,116	0,116	0,116	0,116	0,116	0,116
3	0,104	0,108	0,108	0,108	0,108	0,104	0,104	0,104	0,104	0,104	0,104	0,104	0,104	0,132	0,152	0,152	0,088	0,2069	0,104	0,104	0,104	0,0940	0,0940	0,0940
4	0,076	0,108	0,104	0,104	0,084	0,084	0,068	0,088	0,088	0,0788	0,0690	0,0633	0,104	0,108	0,108	0,108	0,06	0,1930	0,104	0,104	0,104	0,104	0,104	0,104
5	0,136	0,136	0,136	0,136	0,128	0,14	0,136	0,136	0,136	0,1301	0,1273	0,1257	0,144	0,128	0,128	0,128	0,12	0,1847	0,144	0,144	0,144	0,144	0,144	0,144
6	0,084	0,1	0,108	0,108	0,088	0,088	0,084	0,084	0,084	0,084	0,084	0,084	0,124	0,108	0,108	0,108	0,088	0,1888	0,124	0,124	0,124	0,0939	0,0873	0,0835
7	0,072	0,072	0,084	0,084	0,08	0,084	0,072	0,072	0,072	0,072	0,072	0,072	0,108	0,116	0,116	0,116	0,072	0,1955	0,108	0,108	0,108	0,1218	0,1171	0,1143
8	0,072	0,072	0,072	0,072	0,068	0,068	0,1	0,1	0,1	0,072	0,072	0,072	0,124	0,108	0,092	0,092	0,068	0,1802	0,088	0,088	0,088	0,1030	0,1030	0,1030
9	0,12	0,108	0,084	0,084	0,092	0,092	0,12	0,12	0,12	0,0875	0,08	0,0757	0,096	0,096	0,096	0,096	0,076	0,2011	0,096	0,096	0,096	0,096	0,096	0,096
10	0,1	0,1	0,1	0,1	0,1	0,104	0,1	0,1	0,1	0,1014	0,1014	0,1014	0,112	0,124	0,124	0,124	0,1	0,2157	0,148	0,148	0,148	0,1138	0,1093	0,1067
11	0,14	0,136	0,14	0,14	0,12	0,124	0,164	0,164	0,164	0,1259	0,1178	0,1132	0,132	0,172	0,164	0,164	0,096	0,2018	0,172	0,16	0,16	0,1086	0,1	0,0951
12	0,108	0,108	0,108	0,108	0,092	0,104	0,108	0,108	0,108	0,1043	0,1026	0,1016	0,128	0,104	0,112	0,112	0,072	0,2112	0,104	0,12	0,12	0,128	0,128	0,128
13	0,132	0,104	0,096	0,096	0,112	0,112	0,112	0,112	0,112	0,0997	0,0960	0,0938	0,116	0,132	0,136	0,136	0,072	0,1942	0,124	0,124	0,124	0,1147	0,1050	0,0995
14	0,152	0,116	0,124	0,124	0,136	0,132	0,152	0,152	0,152	0,152	0,152	0,152	0,128	0,136	0,144	0,144	0,12	0,1839	0,132	0,132	0,132	0,1246	0,1287	0,1310
15	0,068	0,084	0,064	0,064	0,068	0,076	0,08	0,092	0,092	0,068	0,068	0,068	0,1	0,084	0,084	0,084	0,048	0,1900	0,06	0,076	0,076	0,1	0,1	0,1
16	0,124	0,108	0,108	0,108	0,108	0,124	0,128	0,128	0,128	0,1191	0,1191	0,1191	0,116	0,12	0,12	0,12	0,108	0,1720	0,12	0,12	0,12	0,1243	0,1243	0,1243
17	0,116	0,1	0,1	0,1	0,088	0,084	0,116	0,116	0,116	0,116	0,116	0,116	0,064	0,1	0,1	0,1	0,056	0,1827	0,064	0,064	0,064	0,064	0,064	0,064
18	0,132	0,132	0,136	0,136	0,104	0,124	0,132	0,132	0,132	0,132	0,132	0,132	0,124	0,096	0,096	0,096	0,096	0,2031	0,124	0,124	0,124	0,1191	0,1191	0,1191
19	0,124	0,128	0,128	0,128	0,108	0,112	0,12	0,124	0,124	0,124	0,124	0,124	0,132	0,108	0,108	0,108	0,112	0,1800	0,096	0,096	0,096	0,132	0,132	0,132
20	0,108	0,112	0,1	0,1	0,092	0,108	0,108	0,108	0,108	0,108	0,108	0,108	0,112	0,12	0,1	0,1	0,088	0,1942	0,112	0,112	0,112	0,112	0,112	0,112
21	0,144	0,144	0,16	0,16	0,104	0,124	0,144	0,144	0,144	0,144	0,144	0,144	0,128	0,124	0,124	0,124	0,116	0,2098	0,128	0,128	0,128	0,128	0,128	0,128
22	0,116	0,104	0,104	0,104	0,1	0,112	0,108	0,108	0,108	0,1034	0,0996	0,0974	0,096	0,136	0,136	0,136	0,076	0,1538	0,096	0,096	0,096	0,0911	0,0907	0,0905
23	0,112	0,112	0,112	0,112	0,136	0,116	0,112	0,112	0,112	0,112	0,112	0,112	0,14	0,132	0,12	0,12	0,104	0,2039	0,132	0,132	0,132	0,14	0,14	0,14
24	0,12	0,12	0,144	0,144	0,1	0,144	0,096	0,096	0,096	0,12	0,12	0,12	0,132	0,128	0,128	0,128	0,088	0,1967	0,104	0,104	0,104	0,132	0,132	0,132
25	0,0871	0,0943	0,0910	0,0910	0,064	0,056	0,0871	0,0871	0,0871	0,0871	0,0871	0,0871	0,084	0,088	0,1	0,1	0,072	0,1963	0,084	0,084	0,084	0,0862	0,0854	0,0849
26	0,1235	0,1106	0,0925	0,0925	0,104	0,108	0,124	0,124	0,124	0,1246	0,1231	0,1222	0,132	0,108	0,096	0,096	0,088	0,1908	0,132	0,132	0,132	0,1246	0,1231	0,1222
27	0,1147	0,0892	0,1186	0,1186	0,08	0,1	0,1147	0,1147	0,1147	0,1159	0,1216	0,125	0,112	0,1	0,104	0,104	0,092	0,2046	0,112	0,112	0,112	0,1159	0,1216	0,125
28	0,1107	0,1092	0,1092	0,1092	0,096	0,104	0,1107	0,1107	0,1107	0,1111	0,1111	0,1111	0,108	0,12	0,12	0,12	0,104	0,2242	0,116	0,116	0,116	0,1035	0,1033	0,1031
29	0,1306	0,1306	0,1306	0,1306	0,136	0,14	0,1306	0,1306	0,1306	0,1306	0,1306	0,1306	0,128	0,128	0,128	0,128	0,108	0,2106	0,128	0,128	0,128	0,128	0,128	0,128
30	0,0398	0,0878	0,1186	0,1186	0,076	0,072	0,084	0,084	0,084	0,0598	0,0579	0,0568	0,048	0,104	0,12	0,12	0,052	0,2060	0,076	0,076	0,076	0,0623	0,0634	0,0640

Fold	Base 2																							
	1.000 registros																							
	Redes Neurais, 20 Neurônios, BPROP												Redes Neurais, 20 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,12	0,096	0,08	0,08	0,076	0,108	0,108	0,108	0,108	0,1031	0,0921	0,0859	0,136	0,088	0,132	0,132	0,068	0,1939	0,12	0,108	0,108	0,0859	0,0780	0,0735
2	0,092	0,096	0,096	0,096	0,108	0,136	0,092	0,092	0,092	0,095	0,095	0,095	0,128	0,092	0,368	0,368	0,108	0,2065	0,128	0,128	0,128	0,12	0,12	0,12
3	0,112	0,1	0,1	0,1	0,092	0,088	0,1	0,1	0,1	0,0990	0,0931	0,0898	0,112	0,1	0,1	0,1	0,096	0,1974	0,1	0,1	0,1	0,0965	0,0913	0,0884
4	0,076	0,052	0,052	0,052	0,068	0,136	0,052	0,052	0,052	0,076	0,076	0,076	0,084	0,052	0,052	0,052	0,056	0,1762	0,084	0,084	0,084	0,084	0,084	0,084
5	0,104	0,12	0,156	0,156	0,128	0,14	0,104	0,104	0,104	0,1198	0,1198	0,1198	0,116	0,116	0,116	0,116	0,128	0,1952	0,116	0,116	0,116	0,1147	0,1161	0,1168
6	0,128	0,104	0,136	0,136	0,08	0,092	0,116	0,12	0,12	0,0939	0,0855	0,0806	0,096	0,128	0,128	0,128	0,08	0,1674	0,124	0,124	0,124	0,0888	0,0873	0,0865
7	0,116	0,096	0,096	0,096	0,048	0,14	0,096	0,096	0,096	0,1091	0,1059	0,1041	0,104	0,104	0,104	0,104	0,06	0,1612	0,104	0,104	0,104	0,104	0,104	0,104
8	0,12	0,12	0,12	0,12	0,068	0,108	0,12	0,12	0,12	0,12	0,12	0,12	0,128	0,128	0,128	0,128	0,08	0,2123	0,124	0,124	0,124	0,128	0,128	0,128
9	0,092	0,092	0,092	0,092	0,076	0,124	0,092	0,092	0,092	0,092	0,092	0,092	0,108	0,108	0,108	0,108	0,088	0,1917	0,108	0,108	0,108	0,108	0,108	0,108
10	0,12	0,116	0,116	0,116	0,1	0,112	0,136	0,136	0,136	0,1089	0,1089	0,1089	0,14	0,132	0,132	0,132	0,104	0,2001	0,136	0,136	0,136	0,14	0,14	0,14
11	0,112	0,164	0,164	0,164	0,116	0,112	0,164	0,164	0,164	0,112	0,112	0,112	0,14	0,14	0,14	0,14	0,112	0,1904	0,14	0,14	0,14	0,14	0,14	0,14
12	0,12	0,104	0,108	0,108	0,108	0,115	0,144	0,144	0,144	0,0992	0,0914	0,0868	0,144	0,132	0,132	0,132	0,116	0,2015	0,144	0,144	0,144	0,144	0,144	0,144
13	0,108	0,112	0,108	0,108	0,076	0,096	0,108	0,108	0,108	0,0947	0,0887	0,0853	0,112	0,1	0,284	0,284	0,076	0,2201	0,112	0,112	0,112	0,0947	0,0905	0,0881
14	0,148	0,144	0,136	0,136	0,112	0,116	0,128	0,128	0,128	0,148	0,148	0,148	0,14	0,156	0,156	0,156	0,12	0,2062	0,14	0,14	0,14	0,14	0,14	0,14
15	0,08	0,088	0,088	0,088	0,052	0,096	0,08	0,076	0,076	0,0573	0,0573	0,0573	0,08	0,092	0,104	0,104	0,04	0,1964	0,08	0,08	0,08	0,08	0,08	0,08
16	0,116	0,116	0,116	0,116	0,088	0,096	0,116	0,116	0,116	0,1113	0,1072	0,1048	0,12	0,108	0,108	0,108	0,084	0,2050	0,132	0,108	0,108	0,1191	0,1168	0,1155
17	0,076	0,096	0,092	0,092	0,06	0,06	0,076	0,076	0,076	0,076	0,076	0,076	0,068	0,092	0,08	0,08	0,06	0,2226	0,068	0,068	0,068	0,0707	0,0719	0,0726
18	0,1	0,1	0,1	0,1	0,1	0,112	0,1	0,1	0,1	0,1	0,1	0,1	0,12	0,12	0,12	0,12	0,1	0,2204	0,12	0,12	0,12	0,12	0,12	0,12
19	0,104	0,12	0,12	0,12	0,096	0,108	0,108	0,108	0,108	0,104	0,104	0,104	0,104	0,456	0,456	0,456	0,092	0,1704	0,108	0,092	0,092	0,104	0,104	0,104
20	0,108	0,096	0,096	0,096	0,096	0,092	0,108	0,108	0,108	0,0979	0,0915	0,0878	0,12	0,116	0,432	0,432	0,096	0,1775	0,116	0,12	0,12	0,12	0,12	0,12
21	0,132	0,12	0,128	0,128	0,1	0,112	0,132	0,132	0,132	0,1362	0,1362	0,1362	0,14	0,372	0,132	0,132	0,104	0,2077	0,14	0,14	0,14	0,14	0,14	0,14
22	0,108	0,088	0,088	0,088	0,088	0,113	0,096	0,1	0,1	0,1133	0,1133	0,1133	0,108	0,084	0,216	0,216	0,068	0,2146	0,108	0,108	0,108	0,0837	0,0854	0,0863
23	0,112	0,136	0,108	0,108	0,08	0,12	0,128	0,148	0,148	0,112	0,112	0,112	0,124	0,148	0,148	0,148	0,104	0,2378	0,124	0,124	0,124	0,1262	0,1344	0,1390
24	0,116	0,14	0,1	0,1	0,104	0,136	0,104	0,112	0,112	0,116	0,116	0,116	0,112	0,108	0,084	0,084	0,092	0,2360	0,112	0,112	0,112	0,112	0,112	0,112
25	0,08	0,08	0,092	0,092	0,064	0,092	0,08	0,08	0,08	0,08	0,08	0,08	0,08	0,108	0,348	0,348	0,076	0,2005	0,08	0,08	0,08	0,0812	0,0765	0,0738
26	0,104	0,104	0,1	0,1	0,076	0,088	0,104	0,104	0,104	0,1043	0,1082	0,1104	0,108	0,1	0,092	0,092	0,076	0,2096	0,124	0,124	0,124	0,1170	0,1175	0,1178
27	0,128	0,132	0,14	0,14	0,1	0,116	0,128	0,128	0,128	0,1237	0,1273	0,1295	0,12	0,14	0,14	0,14	0,092	0,1858	0,116	0,116	0,116	0,1211	0,1254	0,1280
28	0,112	0,112	0,112	0,112	0,112	0,108	0,112	0,112	0,112	0,112	0,112	0,112	0,104	0,392	0,392	0,392	0,108	0,2157	0,104	0,104	0,104	0,104	0,104	0,104
29	0,108	0,096	0,112	0,112	0,092	0,088	0,108	0,108	0,108	0,108	0,108	0,108	0,104	0,444	0,444	0,444	0,108	0,1671	0,104	0,104	0,104	0,104	0,104	0,104
30	0,068	0,08	0,052	0,052	0,052	0,109	0,068	0,068	0,068	0,0573	0,0561	0,0554	0,084	0,092	0,304	0,304	0,052	0,1987	0,068	0,068	0,068	0,0648	0,0634	0,0625

Tabela A.9 – Erros de Classificação da Base (2) para 3.000 registros.

Fold	Base 2																							
	3.000 registros																							
	Regressão Linear												Regressão Logística											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,1453	0,1546	0,128	0,128	0,144	0,1413	0,1546	0,1546	0,1546	0,2249	0,1419	0,1337	0,168	0,1533	0,128	0,128	0,1613	0,1653	0,1533	0,1533	0,1533	0,1264	0,1257	0,1209
2	0,1733	0,1453	0,1186	0,1186	0,1613	0,1626	0,1533	0,16	0,16	0,1755	0,1297	0,1206	0,1546	0,1453	0,1373	0,1373	0,1573	0,156	0,156	0,1533	0,1533	0,1237	0,1169	0,1106
3	0,1413	0,132	0,132	0,132	0,1413	0,1413	0,1333	0,1	0,1	0,2160	0,1242	0,1191	0,14	0,128	0,132	0,132	0,1346	0,1346	0,136	0,1	0,1	0,1001	0,1062	0,1049
4	0,1546	0,1426	0,1293	0,1293	0,156	0,156	0,1373	0,136	0,136	0,2462	0,1196	0,1137	0,1453	0,1333	0,1253	0,1253	0,1453	0,144	0,1373	0,1333	0,1333	0,1028	0,1011	0,0991
5	0,156	0,136	0,136	0,136	0,1586	0,16	0,1293	0,1146	0,1146	0,2355	0,1041	0,0976	0,1333	0,1293	0,1346	0,1346	0,1333	0,1333	0,1333	0,1333	0,1333	0,1024	0,0994	0,0938
6	0,192	0,148	0,1386	0,1386	0,1866	0,184	0,156	0,16	0,16	0,2417	0,1433	0,1371	0,1613	0,1453	0,136	0,136	0,1626	0,1613	0,156	0,1613	0,1613	0,1377	0,1296	0,1264
7	0,22	0,1546	0,1586	0,1586	0,212	0,208	0,14	0,1386	0,1386	0,2045	0,1324	0,1236	0,152	0,1546	0,1586	0,1586	0,1533	0,1506	0,152	0,152	0,152	0,1202	0,1148	0,1098
8	0,1373	0,1293	0,1266	0,1266	0,144	0,144	0,1373	0,132	0,132	0,2083	0,1109	0,1055	0,1373	0,132	0,124	0,124	0,14	0,1413	0,1386	0,1306	0,1306	0,1129	0,1097	0,1045
9	0,1906	0,1426	0,1466	0,1466	0,2013	0,2	0,1346	0,1186	0,1186	0,1898	0,1322	0,1230	0,152	0,1466	0,1426	0,1426	0,14	0,1413	0,1346	0,1213	0,1213	0,1067	0,1	0,0975
10	0,1533	0,1493	0,1186	0,1186	0,1613	0,1586	0,1533	0,1533	0,1533	0,1875	0,1242	0,1195	0,1546	0,1413	0,1146	0,1146	0,156	0,1546	0,1373	0,1373	0,1373	0,1216	0,1145	0,1119
11	0,1466	0,1266	0,1253	0,1253	0,148	0,1453	0,116	0,108	0,108	0,2159	0,1172	0,1086	0,1373	0,1266	0,1266	0,1266	0,1386	0,14	0,1173	0,0986	0,0986	0,1066	0,0967	0,0925
12	0,1653	0,1453	0,136	0,136	0,168	0,168	0,144	0,144	0,144	0,2200	0,1231	0,1165	0,152	0,1453	0,132	0,132	0,152	0,152	0,148	0,148	0,148	0,1154	0,1115	0,1074
13	0,1413	0,128	0,14	0,14	0,1453	0,1466	0,1306	0,1293	0,1293	0,2128	0,1266	0,1213	0,1386	0,1266	0,132	0,132	0,144	0,1413	0,1333	0,1266	0,1266	0,1199	0,1199	0,1199
14	0,2426	0,156	0,1533	0,1533	0,232	0,2346	0,1533	0,1266	0,1266	0,2229	0,1173	0,1096	0,1426	0,1253	0,1253	0,1253	0,156	0,156	0,148	0,124	0,124	0,1072	0,1063	0,1010
15	0,1866	0,1333	0,1253	0,1253	0,1866	0,1826	0,1253	0,1226	0,1226	0,2111	0,1174	0,1135	0,1346	0,1333	0,132	0,132	0,1333	0,136	0,1266	0,12	0,12	0,1112	0,1103	0,1080
16	0,1453	0,1346	0,1253	0,1253	0,1466	0,148	0,1453	0,1386	0,1386	0,1690	0,1162	0,1093	0,1413	0,136	0,136	0,136	0,14	0,144	0,1413	0,1413	0,1413	0,0940	0,0932	0,0912
17	0,1546	0,124	0,104	0,104	0,1533	0,1533	0,1306	0,1186	0,1186	0,2020	0,1215	0,1155	0,1346	0,1213	0,104	0,104	0,14	0,14	0,128	0,1146	0,1146	0,1185	0,1136	0,1093
18	0,176	0,1346	0,1333	0,1333	0,172	0,1746	0,1293	0,1306	0,1306	0,2150	0,1226	0,1180	0,1533	0,1346	0,136	0,136	0,1493	0,1493	0,1293	0,1253	0,1253	0,1104	0,1134	0,1107
19	0,144	0,132	0,112	0,112	0,1493	0,148	0,132	0,1266	0,1266	0,2251	0,1432	0,1362	0,1533	0,1266	0,112	0,112	0,152	0,152	0,1266	0,1253	0,1253	0,1196	0,1189	0,1170
20	0,1386	0,1386	0,1053	0,1053	0,14	0,14	0,132	0,1186	0,1186	0,2217	0,1443	0,1393	0,1546	0,136	0,0986	0,0986	0,1533	0,1546	0,14	0,1213	0,1213	0,1269	0,1258	0,1248
21	0,176	0,1426	0,1293	0,1293	0,176	0,1786	0,1653	0,1573	0,1573	0,2028	0,1354	0,1303	0,168	0,1426	0,1346	0,1346	0,1666	0,1693	0,168	0,168	0,168	0,1277	0,1257	0,1227
22	0,168	0,1373	0,136	0,136	0,1706	0,1693	0,1373	0,1373	0,1373	0,2478	0,1346	0,1304	0,1386	0,1346	0,1333	0,1333	0,136	0,136	0,1346	0,1346	0,1346	0,1230	0,1200	0,1188
23	0,1546	0,152	0,144	0,144	0,16	0,1573	0,144	0,144	0,144	0,1928	0,1472	0,1435	0,1573	0,156	0,1426	0,1426	0,1533	0,152	0,1426	0,136	0,136	0,1296	0,1296	0,1296
24	0,1613	0,12	0,0906	0,0906	0,1613	0,1626	0,12	0,1106	0,1106	0,1573	0,1019	0,0980	0,132	0,1133	0,0906	0,0906	0,1333	0,1346	0,1133	0,1133	0,1133	0,1062	0,0977	0,0947
25	0,152	0,132	0,1333	0,1333	0,1533	0,152	0,132	0,132	0,132	0,1841	0,1214	0,1172	0,1386	0,1506	0,1386	0,1386	0,136	0,136	0,128	0,128	0,128	0,1040	0,1064	0,1054
26	0,1746	0,1586	0,1333	0,1333	0,1746	0,172	0,1586	0,1426	0,1426	0,2424	0,1432	0,1382	0,18	0,1546	0,1333	0,1333	0,172	0,1733	0,16	0,16	0,16	0,1300	0,1303	0,1281
27	0,1573	0,14	0,1306	0,1306	0,1546	0,156	0,14	0,14	0,14	0,2584	0,1279	0,1225	0,1613	0,1426	0,132	0,132	0,16	0,1613	0,1426	0,1426	0,1426	0,1119	0,1088	0,1075
28	0,1466	0,1493	0,108	0,108	0,152	0,152	0,1386	0,136	0,136	0,2252	0,1163	0,1100	0,156	0,14	0,136	0,136	0,1546	0,1573	0,1426	0,1386	0,1386	0,1170	0,1122	0,1068
29	0,1413	0,128	0,1146	0,1146	0,144	0,14	0,108	0,1226	0,1226	0,1978	0,1226	0,1175	0,12	0,1226	0,1133	0,1133	0,1173	0,1173	0,1106	0,1226	0,1226	0,1054	0,1067	0,1051
30	0,136	0,1173	0,104	0,104	0,1386	0,14	0,14	0,1426	0,1426	0,1670	0,1102	0,1066	0,144	0,1066	0,12	0,12	0,1426	0,14	0,136	0,136	0,136	0,0848	0,0912	0,0916

Fold	Base 2																							
	3.000 registros																							
	Redes Neurais, 3 Neurônios, BPROP												Redes Neurais, 3 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0973	0,096	0,096	0,096	0,1026	0,1053	0,1013	0,0933	0,0933	0,0865	0,0865	0,0865	0,0893	0,1	0,084	0,084	0,0946	0,1053	0,1013	0,1013	0,1013	0,0942	0,0927	0,0919
2	0,0853	0,088	0,0866	0,0866	0,084	0,1	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853	0,0826	0,0946	0,0946	0,0946	0,0746	0,088	0,0826	0,0826	0,0826	0,0826	0,0826	0,0826
3	0,0813	0,088	0,088	0,088	0,0893	0,08	0,0813	0,0813	0,0813	0,0813	0,0813	0,0813	0,0786	0,0773	0,084	0,084	0,0706	0,084	0,0786	0,0786	0,0786	0,0786	0,0786	0,0786
4	0,1053	0,0933	0,0973	0,0973	0,0946	0,104	0,1053	0,1053	0,1053	0,1053	0,1053	0,1053	0,104	0,0973	0,0893	0,0893	0,0933	0,096	0,104	0,104	0,104	0,104	0,104	0,104
5	0,0733	0,072	0,088	0,088	0,076	0,0773	0,0733	0,0733	0,0733	0,0677	0,0670	0,0666	0,0733	0,0813	0,0733	0,0733	0,076	0,0786	0,0813	0,076	0,076	0,0661	0,0652	0,0647
6	0,1186	0,1	0,096	0,096	0,104	0,112	0,1186	0,1186	0,1186	0,0990	0,0943	0,0917	0,0933	0,108	0,0933	0,0933	0,0933	0,0946	0,0933	0,0933	0,0933	0,0933	0,0933	0,0933
7	0,1	0,1053	0,0893	0,0893	0,0973	0,096	0,096	0,076	0,076	0,1	0,1	0,1	0,0946	0,1013	0,1013	0,1013	0,084	0,0933	0,0946	0,0946	0,0946	0,0946	0,0946	0,0946
8	0,1013	0,1013	0,0786	0,0786	0,0866	0,0866	0,0986	0,0986	0,0986	0,0769	0,0774	0,0776	0,0826	0,0826	0,0826	0,0826	0,0813	0,0933	0,0826	0,0826	0,0826	0,0826	0,0826	0,0826
9	0,0946	0,1053	0,1053	0,1053	0,1026	0,1013	0,0946	0,0946	0,0946	0,0946	0,0946	0,0946	0,0946	0,0906	0,0826	0,0826	0,088	0,092	0,0946	0,0946	0,0946	0,0779	0,0726	0,0696
10	0,0973	0,0866	0,0866	0,0866	0,092	0,0893	0,0973	0,0973	0,0973	0,0973	0,0973	0,0973	0,0946	0,0933	0,0933	0,0933	0,0866	0,1026	0,0933	0,0933	0,0933	0,0946	0,0946	0,0946
11	0,0826	0,0733	0,0626	0,0626	0,0786	0,076	0,0853	0,0706	0,0706	0,0863	0,0887	0,0900	0,0773	0,0773	0,0773	0,0773	0,0773	0,08	0,0773	0,0773	0,0773	0,0773	0,0773	0,0773
12	0,1106	0,1013	0,0853	0,0853	0,0986	0,1066	0,1106	0,1106	0,1106	0,0928	0,0878	0,0848	0,1066	0,0946	0,0946	0,0946	0,0946	0,092	0,1066	0,1066	0,1066	0,0878	0,0878	0,0878
13	0,1226	0,1013	0,0853	0,0853	0,0733	0,0813	0,1	0,1106	0,1106	0,0937	0,0920	0,0911	0,1293	0,0906	0,108	0,108	0,0746	0,084	0,112	0,0973	0,0973	0,0920	0,0920	0,0921
14	0,112	0,0853	0,0946	0,0946	0,0973	0,1053	0,1173	0,1026	0,1026	0,0821	0,0757	0,0721	0,0906	0,0826	0,0826	0,0826	0,0866	0,0893	0,0906	0,0906	0,0906	0,0863	0,0837	0,0822
15	0,096	0,096	0,0893	0,0893	0,092	0,104	0,0933	0,0786	0,0786	0,0891	0,0850	0,0826	0,096	0,0906	0,0906	0,0906	0,0866	0,0933	0,0906	0,0906	0,0906	0,096	0,096	0,096
16	0,0813	0,0866	0,076	0,076	0,08	0,0853	0,092	0,092	0,092	0,0782	0,0738	0,0713	0,084	0,1	0,0893	0,0893	0,0746	0,0813	0,088	0,0813	0,0813	0,0757	0,0696	0,0660
17	0,0773	0,0746	0,0786	0,0786	0,0773	0,092	0,0773	0,0773	0,0773	0,0826	0,0814	0,0807	0,068	0,0813	0,0986	0,0986	0,0733	0,0893	0,068	0,068	0,068	0,0909	0,0911	0,0912
18	0,0893	0,0813	0,088	0,088	0,084	0,0853	0,088	0,0906	0,0906	0,0784	0,0776	0,0772	0,084	0,0813	0,0813	0,0813	0,0853	0,0826	0,084	0,084	0,084	0,084	0,084	0,084
19	0,096	0,0826	0,0826	0,0826	0,0853	0,0826	0,096	0,096	0,096	0,096	0,096	0,096	0,0866	0,0893	0,104	0,104	0,08	0,0786	0,0866	0,0866	0,0866	0,0866	0,0866	0,0866
20	0,096	0,0773	0,0706	0,0706	0,0733	0,1013	0,096	0,096	0,096	0,1014	0,1014	0,1014	0,0906	0,08	0,0733	0,0733	0,0773	0,0946	0,0906	0,0906	0,0906	0,1005	0,1005	0,1005
21	0,1266	0,096	0,096	0,096	0,0933	0,0933	0,1266	0,1266	0,1266	0,1266	0,1266	0,1266	0,116	0,0866	0,0933	0,0933	0,0826	0,0826	0,116	0,116	0,116	0,116	0,116	0,116
22	0,0933	0,0933	0,088	0,088	0,092	0,0906	0,0933	0,0933	0,0933	0,0933	0,0933	0,0933	0,0893	0,0893	0,0893	0,0893	0,088	0,0866	0,0893	0,0893	0,0893	0,0893	0,0893	0,0893
23	0,0946	0,0733	0,1	0,1	0,0853	0,1146	0,1013	0,1013	0,1013	0,0915	0,0919	0,0921	0,0933	0,0733	0,0866	0,0866	0,084	0,1	0,0973	0,0933	0,0933	0,0875	0,0878	0,0881
24	0,0853	0,088	0,0786	0,0786	0,0946	0,0893	0,0693	0,0693	0,0693	0,0853	0,0853	0,0853	0,0813	0,084	0,084	0,084	0,0813	0,084	0,084	0,084	0,084	0,0813	0,0813	0,0813
25	0,0866	0,092	0,092	0,092	0,0933	0,0853	0,0866	0,0866	0,0866	0,0891	0,0891	0,0891	0,0866	0,0866	0,0866	0,0866	0,0853	0,096	0,0866	0,0866	0,0866	0,0866	0,0866	0,0866
26	0,1106	0,1053	0,1013	0,1013	0,1	0,1	0,1106	0,1106	0,1106	0,1006	0,0997	0,0992	0,0933	0,1053	0,1146	0,1146	0,0893	0,0906	0,0933	0,0933	0,0933	0,0947	0,0947	0,0947
27	0,0906	0,0853	0,084	0,084	0,0853	0,096	0,0906	0,0906	0,0906	0,0906	0,0906	0,0906	0,0853	0,0853	0,0853	0,0853	0,0813	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853
28	0,076	0,0866	0,0746	0,0746	0,0813	0,0773	0,076	0,076	0,076	0,076	0,076	0,076	0,0853	0,0813	0,0853	0,0853	0,0746	0,0733	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853
29	0,092	0,1013	0,0906	0,0906	0,0826	0,0946	0,072	0,072	0,072	0,0907	0,0907	0,0907	0,0893	0,0893	0,0893	0,0893	0,0853	0,0946	0,0893	0,0893	0,0893	0,0893	0,0893	0,0893
30	0,0973	0,0853	0,0853	0,0853	0,076	0,0933	0,0933	0,0933	0,0933	0,0973	0,0973	0,0973	0,0906	0,1	0,1	0,1	0,0813	0,1	0,1	0,1	0,1	0,0906	0,0906	0,0906

Tabela A.10 – Erros de Classificação da Base (2) para 3.000 registros (continuação).

Fold	Base 2																							
	3.000 registros																							
	Redes Neurais, 10 Neurônios, BPROP												Redes Neurais, 10 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0733	0,0733	0,0733	0,0733	0,0786	0,0786	0,0733	0,0733	0,0733	0,0733	0,0733	0,0733	0,1253	0,1213	0,12	0,12	0,0706	0,0946	0,112	0,1066	0,1066	0,1069	0,1064	0,1061
2	0,068	0,068	0,068	0,068	0,0666	0,0693	0,068	0,068	0,068	0,068	0,068	0,068	0,1026	0,1026	0,1026	0,1026	0,0786	0,076	0,104	0,104	0,104	0,0944	0,0925	0,0914
3	0,072	0,072	0,072	0,072	0,0573	0,0613	0,072	0,072	0,072	0,072	0,072	0,072	0,0946	0,1106	0,1146	0,1146	0,0626	0,088	0,0946	0,0946	0,0946	0,0860	0,0834	0,0819
4	0,0786	0,088	0,0866	0,0866	0,072	0,076	0,0786	0,0786	0,0786	0,0786	0,0786	0,0786	0,1053	0,1053	0,1053	0,1053	0,0773	0,08	0,1053	0,1053	0,1053	0,1053	0,1053	0,1053
5	0,0586	0,0666	0,08	0,08	0,0573	0,0626	0,0586	0,0586	0,0586	0,0586	0,0586	0,0586	0,0786	0,0706	0,0706	0,0706	0,0613	0,0786	0,0786	0,0786	0,0786	0,0809	0,0790	0,0779
6	0,084	0,104	0,104	0,104	0,072	0,0853	0,084	0,084	0,084	0,084	0,084	0,084	0,1146	0,112	0,112	0,112	0,0786	0,1066	0,1146	0,1146	0,1146	0,1146	0,1146	0,1146
7	0,0893	0,096	0,096	0,096	0,0786	0,0893	0,0853	0,0853	0,0853	0,0951	0,0951	0,0951	0,1133	0,096	0,096	0,096	0,0773	0,0853	0,1133	0,1133	0,1133	0,0959	0,0911	0,0883
8	0,0826	0,0853	0,0986	0,0986	0,0733	0,084	0,0826	0,0826	0,0826	0,0769	0,0725	0,0700	0,1053	0,0973	0,0973	0,0973	0,0733	0,104	0,1053	0,1053	0,1053	0,0979	0,0951	0,0935
9	0,06	0,06	0,06	0,06	0,0626	0,048	0,06	0,06	0,06	0,06	0,06	0,06	0,104	0,108	0,0986	0,0986	0,0666	0,0666	0,104	0,104	0,104	0,0991	0,0944	0,0916
10	0,068	0,068	0,068	0,068	0,0586	0,06	0,068	0,068	0,068	0,068	0,068	0,068	0,1	0,0866	0,0906	0,0906	0,064	0,08	0,092	0,096	0,096	0,1	0,1	0,1
11	0,0746	0,0746	0,0746	0,0746	0,0613	0,0626	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746	0,092	0,108	0,1146	0,1146	0,0586	0,0626	0,0906	0,0906	0,0906	0,0855	0,0837	0,0827
12	0,092	0,092	0,092	0,092	0,0906	0,0826	0,092	0,092	0,092	0,092	0,092	0,092	0,1266	0,108	0,108	0,108	0,0906	0,108	0,104	0,104	0,104	0,1179	0,1109	0,1069
13	0,084	0,084	0,084	0,084	0,076	0,076	0,084	0,084	0,084	0,084	0,084	0,084	0,1146	0,1066	0,1053	0,1053	0,0786	0,116	0,1146	0,1146	0,1146	0,0971	0,0971	0,0971
14	0,0866	0,0866	0,0866	0,0866	0,0773	0,0773	0,0866	0,0866	0,0866	0,0866	0,0866	0,0866	0,108	0,104	0,104	0,104	0,0813	0,096	0,108	0,108	0,108	0,0854	0,0843	0,0836
15	0,072	0,0946	0,0946	0,0946	0,0773	0,0693	0,072	0,072	0,072	0,072	0,072	0,072	0,0786	0,1	0,1	0,1	0,0666	0,092	0,0786	0,0786	0,0786	0,0957	0,0950	0,0946
16	0,0693	0,0693	0,068	0,068	0,056	0,0586	0,0693	0,0693	0,0693	0,0693	0,0693	0,0693	0,1	0,1	0,08	0,08	0,0573	0,08	0,1	0,1	0,1	0,0940	0,0907	0,0889
17	0,068	0,0746	0,0746	0,0746	0,068	0,0693	0,068	0,068	0,068	0,068	0,068	0,068	0,0906	0,0813	0,0813	0,0813	0,072	0,0733	0,0826	0,0826	0,0826	0,0859	0,0838	0,0826
18	0,0786	0,0786	0,0786	0,0786	0,0666	0,0706	0,0706	0,088	0,088	0,0786	0,0786	0,0786	0,0893	0,0853	0,0853	0,0853	0,0733	0,08	0,0893	0,0893	0,0893	0,0893	0,0893	0,0893
19	0,084	0,084	0,084	0,084	0,08	0,084	0,084	0,084	0,084	0,084	0,084	0,084	0,1013	0,096	0,096	0,096	0,0773	0,1	0,0986	0,1106	0,1106	0,1087	0,1091	0,1093
20	0,0813	0,0933	0,0933	0,0933	0,076	0,0746	0,0813	0,0813	0,0813	0,0956	0,0942	0,0935	0,1053	0,1053	0,1053	0,1053	0,0733	0,0853	0,1053	0,1053	0,1053	0,1046	0,1056	0,1061
21	0,0866	0,0866	0,0866	0,0866	0,0773	0,076	0,0866	0,0866	0,0866	0,0866	0,0866	0,0866	0,116	0,116	0,116	0,116	0,0733	0,1013	0,116	0,116	0,116	0,116	0,116	0,116
22	0,0826	0,0893	0,0893	0,0893	0,072	0,0773	0,0826	0,0826	0,0826	0,0826	0,0826	0,0826	0,104	0,1066	0,1066	0,1066	0,068	0,0866	0,1066	0,1066	0,1066	0,0946	0,0921	0,0906
23	0,08	0,08	0,08	0,08	0,0693	0,076	0,08	0,08	0,08	0,08	0,08	0,08	0,096	0,1066	0,096	0,096	0,0773	0,1013	0,096	0,096	0,096	0,096	0,096	0,096
24	0,0813	0,084	0,084	0,084	0,0746	0,0786	0,0813	0,0813	0,0813	0,0757	0,0757	0,0757	0,108	0,104	0,1053	0,1053	0,0773	0,0946	0,108	0,108	0,108	0,1013	0,0965	0,0938
25	0,0813	0,0893	0,0893	0,0893	0,0693	0,0706	0,0893	0,0893	0,0893	0,0941	0,0927	0,0918	0,104	0,1013	0,1053	0,1053	0,076	0,0746	0,1253	0,108	0,108	0,0957	0,0891	0,0853
26	0,0853	0,096	0,096	0,096	0,0746	0,0813	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853	0,1013	0,104	0,1173	0,1173	0,0853	0,1026	0,1013	0,1013	0,1013	0,1157	0,1175	0,1184
27	0,084	0,0906	0,0906	0,0906	0,0773	0,0773	0,084	0,084	0,084	0,084	0,084	0,084	0,1053	0,1013	0,1013	0,1013	0,0773	0,0866	0,112	0,112	0,112	0,0976	0,0976	0,0976
28	0,0733	0,0733	0,0733	0,0733	0,0693	0,0586	0,0733	0,0733	0,0733	0,0733	0,0733	0,0733	0,1013	0,092	0,1	0,1	0,0666	0,092	0,1066	0,0986	0,0986	0,1032	0,1029	0,1027
29	0,084	0,084	0,084	0,084	0,0773	0,0746	0,084	0,084	0,084	0,084	0,084	0,084	0,0893	0,1	0,1053	0,1053	0,08	0,0853	0,084	0,084	0,084	0,0899	0,0899	0,0899
30	0,068	0,0786	0,0786	0,0786	0,0653	0,0666	0,068	0,068	0,068	0,068	0,068	0,068	0,1106	0,0893	0,0893	0,0893	0,0653	0,0786	0,088	0,1	0,1	0,0965	0,0912	0,0882

Fold	Base 2																							
	3.000 registros																							
	Redes Neurais, 20 Neurônios, BPROP												Redes Neurais, 20 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,088	0,088	0,088	0,088	0,0826	0,0786	0,088	0,088	0,088	0,088	0,088	0,088	0,1	0,1026	0,1093	0,1093	0,0746	0,1066	0,1	0,1	0,1	0,1	0,1	0,1
2	0,0906	0,0906	0,0906	0,0906	0,0693	0,072	0,0906	0,0906	0,0906	0,0906	0,0906	0,0906	0,1053	0,1146	0,1133	0,1133	0,072	0,0623	0,0986	0,1066	0,1066	0,0877	0,0828	0,0799
3	0,0653	0,0813	0,0813	0,0813	0,0626	0,064	0,0653	0,0653	0,0653	0,0728	0,0738	0,0743	0,0866	0,1026	0,1053	0,1053	0,0746	0,0973	0,1026	0,1026	0,1026	0,0778	0,0762	0,0753
4	0,08	0,08	0,08	0,08	0,08	0,088	0,08	0,08	0,08	0,08	0,08	0,08	0,0986	0,096	0,096	0,096	0,0826	0,0843	0,0986	0,0986	0,0986	0,1003	0,1003	0,1003
5	0,0653	0,0653	0,0653	0,0653	0,068	0,068	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,072	0,072	0,072	0,072	0,068	0,0552	0,072	0,072	0,072	0,072	0,072	0,072
6	0,1013	0,1	0,1	0,1	0,072	0,088	0,1013	0,1013	0,1013	0,1013	0,1013	0,1013	0,1053	0,1133	0,112	0,112	0,0813	0,0794	0,1053	0,1053	0,1053	0,1053	0,1053	0,1053
7	0,0826	0,096	0,096	0,096	0,0813	0,0973	0,0826	0,0826	0,0826	0,0843	0,0843	0,0843	0,1013	0,0946	0,0946	0,0946	0,0813	0,1066	0,0973	0,0973	0,0973	0,0926	0,0893	0,0873
8	0,084	0,0906	0,0906	0,0906	0,0706	0,0813	0,1013	0,1013	0,1013	0,0828	0,0780	0,0752	0,0813	0,0813	0,0813	0,0813	0,0773	0,0893	0,0813	0,0813	0,0813	0,0895	0,0890	0,0887
9	0,0693	0,084	0,084	0,084	0,0613	0,068	0,0693	0,0693	0,0693	0,0693	0,0693	0,0693	0,0853	0,096	0,08	0,08	0,0653	0,0902	0,0853	0,0853	0,0853	0,0923	0,0894	0,0877
10	0,0733	0,0746	0,0746	0,0746	0,0626	0,0746	0,0733	0,0733	0,0733	0,0733	0,0733	0,0733	0,0866	0,084	0,084	0,084	0,0733	0,0933	0,084	0,084	0,084	0,0866	0,0866	0,0866
11	0,084	0,08	0,08	0,08	0,0533	0,056	0,0773	0,0773	0,0773	0,0779	0,0700	0,0655	0,108	0,104	0,0826	0,0826	0,064	0,0902	0,108	0,108	0,108	0,0999	0,0998	0,0998
12	0,0986	0,0986	0,0986	0,0986	0,0866	0,096	0,0986	0,0986	0,0986	0,0986	0,0986	0,0986	0,0973	0,0973	0,0973	0,0973	0,096	0,0649	0,0973	0,0973	0,0973	0,0973	0,0973	0,0973
13	0,0893	0,0866	0,0986	0,0986	0,0733	0,0733	0,0893	0,0893	0,0893	0,0893	0,0893	0,0893	0,0973	0,0973	0,0973	0,0973	0,076	0,0973	0,0973	0,0973	0,0973	0,0920	0,0920	0,0920
14	0,08	0,08	0,08	0,08	0,076	0,0706	0,08	0,08	0,08	0,0720	0,0720	0,0720	0,08	0,096	0,108	0,108	0,0746	0,076	0,08	0,08	0,08	0,0888	0,0888	0,0888
15	0,0786	0,0786	0,0786	0,0786	0,068	0,0733	0,0786	0,0786	0,0786	0,0786	0,0786	0,0786	0,0866	0,084	0,084	0,084	0,0733	0,1	0,0866	0,0866	0,0866	0,0842	0,0808	0,0789
16	0,068	0,068	0,068	0,068	0,0573	0,06	0,068	0,068	0,068	0,068	0,068	0,068	0,0933	0,064	0,064	0,064	0,0586	0,076	0,0626	0,0626	0,0626	0,0716	0,0690	0,0675
17	0,0786	0,0786	0,0786	0,0786	0,0626	0,0653	0,0786	0,0786	0,0786	0,0786	0,0786	0,0786	0,0797	0,0797	0,0797	0,0797	0,0598	0,0649	0,0797	0,0797	0,0797	0,0843	0,0843	0,0843
18	0,0733	0,092	0,092	0,092	0,076	0,0773	0,0733	0,0733	0,0733	0,0767	0,0745	0,0733	0,0651	0,0903	0,0955	0,0955	0,0711	0,0643	0,088	0,088	0,088	0,0651	0,0651	0,0651
19	0,0826	0,096	0,096	0,096	0,0773	0,0853	0,096	0,096	0,096	0,0826	0,0826	0,0826	0,0969	0,0926	0,0926	0,0926	0,0776	0,0930	0,0969	0,0969	0,0969	0,1012	0,0951	0,0916
20	0,0746	0,0746	0,0746	0,0746	0,0706	0,0733	0,0746	0,0746	0,0746	0,0824	0,0824	0,0824	0,0772	0,0772	0,0772	0,0772	0,0715	0,0623	0,0826	0,0826	0,0826	0,0840	0,0811	0,0794
21	0,0893	0,092	0,092	0,092	0,0773	0,084	0,0893	0,0893	0,0893	0,0893	0,0893	0,0893	0,0912	0,0958	0,0958	0,0958	0,0751	0,0902	0,1053	0,1053	0,1053	0,0912	0,0912	0,0912
22	0,0853	0,0853	0,0853	0,0853	0,08	0,0826	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853	0,0977	0,0977	0,0977	0,0977	0,0733	0,0790	0,1186	0,112	0,112	0,1006	0,1006	0,1006
23	0,0746	0,0853	0,0853	0,0853	0,072	0,0666	0,0853	0,0853	0,0853	0,0915	0,0954	0,0976	0,0858	0,0957	0,0957	0,0957	0,0722	0,0552	0,0893	0,0933	0,0933	0,0964	0,0977	0,0985
24	0,076	0,0733	0,0733	0,0733	0,068	0,0786	0,0733	0,0733	0,0733	0,0848	0,0810	0,0788	0,0739	0,0794	0,0794	0,0794	0,0696	0,0821	0,0893	0,084	0,084	0,0873	0,0882	0,0887
25	0,088	0,088	0,088	0,088	0,0773	0,0733	0,088	0,088	0,088	0,088	0,088	0,088	0,0851	0,0851	0,0851	0,0851	0,0712	0,0677	0,1173	0,1173	0,1173	0,0966	0,0938	0,0923
26	0,1013	0,0973	0,1053	0,1053	0,0853	0,0893	0,1013	0,1013	0,1013	0,1098	0,1077	0,1064	0,0994	0,0995	0,1049	0,1049	0,0819	0,0916	0,0994	0,0994	0,0994	0,1115	0,1095	0,1083
27	0,0853	0,0853	0,0853	0,0853	0,0733	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853	0,0889	0,0889	0,0889	0,0889	0,0887	0,0882	0,088	0,088	0,088	0,0925	0,0940	0,0949
28	0,0813	0,08	0,08	0,08	0,0613	0,0746	0,0853	0,0853	0,0853	0,0813	0,0813	0,0813	0,0839	0,0853	0,0853	0,0853	0,0545	0,0794	0,0839	0,0839	0,0839	0,0839	0,0839	0,0839
29	0,0946	0,0946	0,0946	0,0946	0,08	0,0853	0,0946	0,0946	0,0946	0,0946	0,0946	0,0946	0,0842	0,0842	0,0842	0,0842	0,0845	0,0843	0,0842	0,0842	0,0842	0,0842	0,0842	0,0842
30	0,0853	0,084	0,084	0,084	0,0733	0,0733	0,084	0,084	0,084	0,0853	0,0853	0,0853	0,0753	0,0806	0,0806	0,0806	0,0842	0,0705	0,0893	0,0893	0,0893	0,0753	0,0753	0,0753

Tabela A.11 – Erros de Classificação da Base (2) para 5.000 registros.

Fold	Base 2																							
	5.000 registros																							
	Regressão Linear												Regressão Logística											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,1584	0,1336	0,1256	0,1256	0,1536	0,1576	0,1336	0,1232	0,1232	0,1936	0,1158	0,1098	0,1368	0,1224	0,1192	0,1192	0,1368	0,1376	0,1312	0,1208	0,1208	0,1053	0,1027	0,0994
2	0,176	0,1448	0,1232	0,1232	0,172	0,1736	0,176	0,176	0,176	0,2240	0,1260	0,1188	0,1584	0,1424	0,12	0,12	0,1552	0,156	0,148	0,1296	0,1296	0,1223	0,1153	0,1104
3	0,156	0,1432	0,1192	0,1192	0,1536	0,1536	0,1384	0,1344	0,1344	0,2234	0,1323	0,1266	0,1488	0,1336	0,1176	0,1176	0,1432	0,1432	0,14	0,1328	0,1328	0,1290	0,1230	0,1193
4	0,1704	0,1272	0,1	0,1	0,1688	0,1688	0,1272	0,116	0,116	0,1997	0,1216	0,1153	0,136	0,12	0,0992	0,0992	0,1392	0,1392	0,1256	0,1136	0,1136	0,1040	0,1044	0,1018
5	0,128	0,1096	0,0976	0,0976	0,1264	0,1264	0,1176	0,0984	0,0984	0,2230	0,0939	0,0894	0,1264	0,1104	0,0992	0,0992	0,1232	0,124	0,12	0,1	0,1	0,0880	0,0821	0,0801
6	0,1616	0,156	0,1488	0,1488	0,16	0,1592	0,1592	0,1432	0,1432	0,2162	0,1439	0,1386	0,1664	0,1472	0,1288	0,1288	0,16	0,16	0,1552	0,1384	0,1384	0,1291	0,1303	0,1279
7	0,172	0,1408	0,1424	0,1424	0,176	0,1744	0,1464	0,132	0,132	0,2117	0,1335	0,1261	0,1488	0,1424	0,1184	0,1184	0,1528	0,1504	0,144	0,1304	0,1304	0,1232	0,1148	0,1114
8	0,132	0,12	0,1184	0,1184	0,1312	0,1304	0,1272	0,1216	0,1216	0,2357	0,1150	0,1101	0,1424	0,1224	0,1032	0,1032	0,1448	0,1456	0,132	0,1224	0,1224	0,0985	0,0986	0,0973
9	0,1648	0,1448	0,1352	0,1352	0,1632	0,1624	0,1272	0,1192	0,1192	0,2233	0,1037	0,0985	0,1392	0,144	0,1304	0,1304	0,1392	0,1384	0,1248	0,116	0,116	0,1015	0,0947	0,0914
10	0,1496	0,1152	0,0984	0,0984	0,1472	0,1472	0,1256	0,1152	0,1152	0,2004	0,1167	0,1123	0,1336	0,1152	0,096	0,096	0,1416	0,1408	0,12	0,12	0,12	0,0991	0,0995	0,0989
11	0,1632	0,1392	0,1384	0,1384	0,1608	0,1632	0,14	0,1312	0,1312	0,2222	0,1235	0,1156	0,1544	0,14	0,1448	0,1448	0,1544	0,1552	0,1392	0,1272	0,1272	0,1136	0,1088	0,1041
12	0,1464	0,12	0,1024	0,1024	0,1456	0,1464	0,1208	0,116	0,116	0,2123	0,0898	0,0854	0,128	0,1176	0,1056	0,1056	0,128	0,128	0,1216	0,1144	0,1144	0,0964	0,0880	0,0840
13	0,1256	0,1192	0,1152	0,1152	0,124	0,1256	0,1272	0,1144	0,1144	0,1908	0,1207	0,1151	0,1344	0,1152	0,0976	0,0976	0,1344	0,1352	0,128	0,1144	0,1144	0,1093	0,1056	0,1032
14	0,1768	0,1432	0,1368	0,1368	0,18	0,1784	0,1408	0,1336	0,1336	0,1943	0,1278	0,1233	0,1544	0,14	0,14	0,14	0,1568	0,1544	0,14	0,1368	0,1368	0,1150	0,1135	0,1120
15	0,1568	0,1456	0,1232	0,1232	0,1624	0,1592	0,1256	0,1208	0,1208	0,2224	0,1221	0,1197	0,1632	0,1528	0,1208	0,1208	0,1608	0,1624	0,124	0,12	0,12	0,1152	0,1181	0,1166
16	0,1736	0,1272	0,0976	0,0976	0,1704	0,172	0,1184	0,1064	0,1064	0,2129	0,1163	0,1112	0,1368	0,1104	0,092	0,092	0,132	0,1312	0,1104	0,1016	0,1016	0,0991	0,1013	0,0994
17	0,1784	0,1456	0,1232	0,1232	0,176	0,1776	0,1456	0,1328	0,1328	0,2163	0,1304	0,1237	0,148	0,1448	0,1216	0,1216	0,1472	0,1464	0,1448	0,128	0,128	0,1231	0,1231	0,1231
18	0,1552	0,1328	0,1216	0,1216	0,1552	0,1544	0,1488	0,1488	0,1488	0,2157	0,1286	0,1254	0,1536	0,1368	0,1232	0,1232	0,1528	0,1528	0,148	0,1248	0,1248	0,1150	0,1150	0,1150
19	0,1624	0,1512	0,1344	0,1344	0,1576	0,156	0,1568	0,1368	0,1368	0,1979	0,1189	0,1141	0,1624	0,148	0,1392	0,1392	0,1632	0,1624	0,1536	0,1536	0,1536	0,1150	0,1089	0,1062
20	0,1608	0,1432	0,1168	0,1168	0,1576	0,1576	0,1424	0,1296	0,1296	0,1826	0,1240	0,1179	0,1576	0,1392	0,1152	0,1152	0,1528	0,1544	0,1376	0,1312	0,1312	0,1094	0,1057	0,1036
21	0,1944	0,1424	0,1336	0,1336	0,2	0,1984	0,1344	0,1312	0,1312	0,2188	0,1181	0,1101	0,1416	0,1384	0,1328	0,1328	0,14	0,1376	0,1352	0,1256	0,1256	0,1089	0,1027	0,0981
22	0,1464	0,1176	0,1136	0,1136	0,1432	0,1448	0,1176	0,1152	0,1152	0,2018	0,1198	0,1171	0,1344	0,1168	0,116	0,116	0,1328	0,132	0,1168	0,1184	0,1184	0,1011	0,1049	0,1054
23	0,1712	0,1232	0,1216	0,1216	0,1704	0,1688	0,1392	0,128	0,128	0,2143	0,1201	0,1131	0,1416	0,14	0,1376	0,1376	0,1392	0,1392	0,1368	0,1368	0,1368	0,1165	0,1135	0,1079
24	0,1504	0,1368	0,136	0,136	0,1528	0,1512	0,1272	0,1192	0,1192	0,2021	0,1062	0,1026	0,148	0,1376	0,1296	0,1296	0,1464	0,1456	0,1296	0,1176	0,1176	0,1085	0,1029	0,1000
25	0,1832	0,1264	0,1264	0,1264	0,1896	0,1864	0,1384	0,1328	0,1328	0,2049	0,1195	0,1136	0,144	0,128	0,1304	0,1304	0,144	0,1464	0,1408	0,1288	0,1288	0,1118	0,1060	0,1030
26	0,1408	0,1352	0,1016	0,1016	0,1384	0,1392	0,1152	0,1192	0,1192	0,1788	0,1060	0,0995	0,1304	0,1328	0,1032	0,1032	0,1368	0,1336	0,1184	0,1184	0,1184	0,0959	0,0940	0,0901
27	0,1488	0,1264	0,1136	0,1136	0,152	0,1512	0,1328	0,1232	0,1232	0,2326	0,1244	0,1199	0,1488	0,1264	0,1256	0,1256	0,148	0,148	0,1296	0,1296	0,1296	0,1068	0,1092	0,1080
28	0,1704	0,1224	0,1168	0,1168	0,1728	0,172	0,1224	0,1136	0,1136	0,2072	0,1220	0,1155	0,1352	0,1216	0,116	0,116	0,1304	0,1288	0,1256	0,1256	0,1256	0,1088	0,1020	0,0998
29	0,14	0,1304	0,1328	0,1328	0,1408	0,1408	0,1312	0,1296	0,1296	0,1850	0,1226	0,1179	0,1416	0,1288	0,1048	0,1048	0,1408	0,1424	0,1288	0,1264	0,1264	0,1137	0,1116	0,1092
30	0,148	0,1408	0,1248	0,1248	0,1464	0,1472	0,1312	0,12	0,12	0,1748	0,1129	0,1071	0,1368	0,1328	0,1072	0,1072	0,1432	0,1432	0,1296	0,1216	0,1216	0,1038	0,1000	0,0970

Fold	Base 2																							
	5.000 registros																							
	Redes Neurais, 3 Neurônios, BPROP												Redes Neurais, 3 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0888	0,0728	0,0656	0,0656	0,0848	0,0776	0,0888	0,0888	0,0888	0,0787	0,0800	0,0807	0,0816	0,0784	0,0896	0,0896	0,0776	0,0824	0,0816	0,0816	0,0816	0,0792	0,0811	0,0822
2	0,1	0,0872	0,0752	0,0752	0,1	0,0976	0,1	0,1	0,1	0,0850	0,0844	0,0841	0,1008	0,0864	0,0824	0,0824	0,096	0,092	0,096	0,0928	0,0928	0,0850	0,0819	0,0801
3	0,1	0,0952	0,08	0,08	0,0912	0,0928	0,0848	0,0968	0,0968	0,1	0,1	0,1	0,092	0,0864	0,0776	0,0776	0,0904	0,0944	0,092	0,092	0,092	0,092	0,092	0,092
4	0,0816	0,0784	0,0688	0,0688	0,0792	0,0848	0,0816	0,0816	0,0816	0,0807	0,0768	0,0745	0,0848	0,0752	0,0752	0,0752	0,0776	0,0728	0,0848	0,0848	0,0848	0,0807	0,0796	0,0790
5	0,0696	0,0672	0,0624	0,0624	0,06	0,0864	0,0624	0,0656	0,0656	0,0637	0,0637	0,0637	0,0656	0,0736	0,0768	0,0768	0,0672	0,0856	0,0664	0,0648	0,0648	0,0656	0,0656	0,0656
6	0,1056	0,0976	0,088	0,088	0,104	0,108	0,1056	0,1056	0,1056	0,1056	0,1056	0,1056	0,1056	0,0856	0,0704	0,0704	0,1072	0,1048	0,1056	0,1056	0,1056	0,1056	0,1056	0,1056
7	0,0952	0,0848	0,0928	0,0928	0,0928	0,0968	0,0952	0,0952	0,0952	0,0900	0,0887	0,0880	0,0984	0,0952	0,0952	0,0952	0,0848	0,0952	0,0984	0,0984	0,0984	0,0925	0,0920	0,0918
8	0,076	0,0776	0,0688	0,0688	0,0728	0,0936	0,0848	0,0808	0,0808	0,0690	0,0670	0,0658	0,08	0,0792	0,0648	0,0648	0,0776	0,0896	0,08	0,08	0,08	0,0785	0,0764	0,0752
9	0,0888	0,0816	0,0776	0,0776	0,08	0,0888	0,0888	0,0888	0,0888	0,0730	0,0687	0,0662	0,08	0,0904	0,0768	0,0768	0,0752	0,0848	0,08	0,08	0,08	0,0741	0,0702	0,0680
10	0,0896	0,0704	0,0704	0,0704	0,0768	0,0792	0,0896	0,0896	0,0896	0,0896	0,0896	0,0896	0,0688	0,0824	0,0936	0,0936	0,0688	0,0648	0,0688	0,0688	0,0688	0,0761	0,0761	0,0761
11	0,096	0,0904	0,0768	0,0768	0,0952	0,1144	0,0952	0,0768	0,0768	0,0935	0,0949	0,0957	0,0896	0,0736	0,0736	0,0736	0,092	0,0912	0,0952	0,0888	0,0888	0,0940	0,0953	0,0960
12	0,0816	0,0704	0,0624	0,0624	0,0784	0,0712	0,0792	0,0752	0,0752	0,0816	0,0816	0,0816	0,0752	0,072	0,0632	0,0632	0,072	0,0752	0,0872	0,0824	0,0824	0,0752	0,0752	0,0752
13	0,092	0,0792	0,0736	0,0736	0,0904	0,0912	0,0848	0,0752	0,0752	0,0795	0,0796	0,0797	0,0808	0,08	0,0784	0,0784	0,0832	0,084	0,0792	0,0792	0,0792	0,0808	0,0808	0,0808
14	0,108	0,1024	0,0752	0,0752	0,0976	0,1056	0,0912	0,0912	0,0912	0,108	0,108	0,108	0,0936	0,0944	0,0864	0,0864	0,096	0,0976	0,092	0,092	0,092	0,0912	0,0887	0,0872
15	0,0872	0,0808	0,0728	0,0728	0,0824	0,084	0,0944	0,08	0,08	0,0793	0,0789	0,0788	0,0808	0,076	0,0792	0,0792	0,0744	0,08	0,0936	0,092	0,092	0,0783	0,0771	0,0765
16	0,0832	0,0824	0,0736	0,0736	0,0752	0,0808	0,076	0,076	0,076	0,0831	0,0875	0,0900	0,0752	0,0744	0,0768	0,0768	0,072	0,0752	0,0824	0,0824	0,0824	0,0851	0,0882	0,0900
17	0,0976	0,0872	0,0784	0,0784	0,0944	0,0968	0,0784	0,068	0,068	0,0976	0,0976	0,0976	0,0936	0,0768	0,0656	0,0656	0,0928	0,0968	0,08	0,0704	0,0704	0,0936	0,0936	0,0936
18	0,0928	0,0744	0,0896	0,0896	0,0776	0,0816	0,1152	0,088	0,088	0,0843	0,0819	0,0806	0,0784	0,0904	0,0736	0,0736	0,0744	0,08	0,0784	0,0784	0,0784	0,0753	0,0736	0,0727
19	0,1024	0,1032	0,0992	0,0992	0,1024	0,1	0,1064	0,0912	0,0912	0,1001	0,1006	0,1009	0,0984	0,0968	0,0944	0,0944	0,0976	0,096	0,0984	0,0984	0,0984	0,1011	0,1013	0,1014
20	0,088	0,0824	0,0776	0,0776	0,0832	0,0872	0,088	0,088	0,088	0,088	0,088	0,088	0,08	0,0864	0,0864	0,0864	0,08	0,0784	0,08	0,08	0,08	0,08	0,08	0,08
21	0,0936	0,0824	0,0912	0,0912	0,092	0,0912	0,0936	0,0936	0,0936	0,0936	0,0936	0,0936	0,088	0,0904	0,0904	0,0904	0,0856	0,0808	0,088	0,088	0,088	0,088	0,088	0,088
22	0,096	0,0824	0,0776	0,0776	0,0904	0,0872	0,0824	0,0824	0,0824	0,096	0,096	0,096	0,0912	0,0872	0,0744	0,0744	0,0896	0,0864	0,0872	0,0872	0,0872	0,0846	0,0846	0,0846
23	0,0856	0,0784	0,0736	0,0736	0,0816	0,092	0,0856	0,0856	0,0856	0,0856	0,0856	0,0856	0,0792	0,0776	0,08	0,08	0,08	0,0848	0,0792	0,0792	0,0792	0,0792	0,0792	0,0792
24	0,0944	0,096	0,0968	0,0968	0,0888	0,0968	0,0944	0,0944	0,0944	0,0944	0,0944	0,0944	0,088	0,084	0,0784	0,0784	0,0832	0,0808	0,088	0,088	0,088	0,088	0,088	0,088
25	0,0896	0,08	0,0752	0,0752	0,0848	0,0952	0,0896	0,0896	0,0896	0,0881	0,0846	0,0827	0,0856	0,088	0,088	0,088	0,0784	0,0776	0,0856	0,0856	0,0856	0,0866	0,0832	0,0813
26	0,0912	0,084	0,064	0,064	0,084	0,0888	0,0744	0,0704	0,0704	0,0779	0,0795	0,0804	0,084	0,0744	0,0696	0,0696	0,0792	0,0816	0,084	0,084	0,084	0,084	0,084	0,084
27	0,0944	0,0808	0,072	0,072	0,0824	0,0888	0,0816	0,0856	0,0856	0,0753	0,0696	0,0664	0,0928	0,0848	0,0848	0,0848	0,0736	0,0752	0,0848	0,0824	0,0824	0,0763	0,0714	0,0686
28	0,0896	0,0776	0,0712	0,0712	0,0736	0,0864	0,0768	0,0704	0,0704	0,0969	0,0969	0,0969	0,0864	0,0768	0,0704	0,0704	0,0744	0,076	0,072	0,0664	0,0664	0,0796	0,0787	0,0782
29	0,0904	0,0816	0,0856	0,0856	0,0888	0,0912	0,0808	0,0744	0,0744	0,0904	0,0904	0,0904	0,0872	0,0872	0,0776	0,0776	0,0952	0,0928	0,0872	0,0872	0,0872	0,0872	0,0872	0,0872
30	0,0864	0,0896	0,088	0,088	0,08	0,0832	0,0888	0,0888	0,0888	0,0761	0,0761	0,0761	0,0872	0,0816	0,0896	0,0896	0,0832	0,084	0,0872	0,0872	0,0872	0,0872	0,0872	0,0872

Tabela A.12 – Erros de Classificação da Base (2) para 5.000 registros (continuação).

Fold	Base 2											
	5.000 registros											
	Redes Neurais, 10 Neurônios, BPROP											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0672	0,0696	0,0696	0,0696	0,0616	0,0616	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672
2	0,0672	0,0672	0,0672	0,0672	0,0648	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672
3	0,068	0,0752	0,0752	0,0752	0,0664	0,0656	0,068	0,068	0,068	0,068	0,068	0,068
4	0,068	0,068	0,068	0,068	0,0568	0,0624	0,068	0,068	0,068	0,068	0,068	0,068
5	0,0488	0,0584	0,0624	0,0624	0,0512	0,0552	0,0488	0,0488	0,0488	0,0488	0,0488	0,0488
6	0,076	0,076	0,076	0,076	0,072	0,0712	0,076	0,076	0,076	0,076	0,076	0,076
7	0,0736	0,0952	0,0952	0,0952	0,0744	0,0752	0,0736	0,0736	0,0736	0,0736	0,0736	0,0736
8	0,0664	0,0664	0,0664	0,0664	0,0528	0,0584	0,0664	0,0664	0,0664	0,0664	0,0664	0,0664
9	0,0808	0,0816	0,0816	0,0816	0,068	0,0704	0,0808	0,0808	0,0808	0,0808	0,0808	0,0808
10	0,068	0,068	0,068	0,068	0,0584	0,0576	0,068	0,068	0,068	0,068	0,068	0,068
11	0,068	0,068	0,068	0,068	0,0576	0,0608	0,068	0,068	0,068	0,068	0,068	0,068
12	0,0592	0,0592	0,0592	0,0592	0,056	0,0584	0,0592	0,0592	0,0592	0,0592	0,0592	0,0592
13	0,0728	0,0728	0,0728	0,0728	0,0696	0,0672	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728
14	0,0768	0,0784	0,0784	0,0784	0,0744	0,08	0,0768	0,0768	0,0768	0,0768	0,0768	0,0768
15	0,0736	0,0856	0,0856	0,0856	0,0736	0,0784	0,076	0,076	0,076	0,0738	0,0735	0,0733
16	0,0752	0,0752	0,0752	0,0752	0,0744	0,0776	0,0752	0,0752	0,0752	0,0752	0,0752	0,0752
17	0,0712	0,0744	0,0744	0,0744	0,0656	0,0648	0,0712	0,0712	0,0712	0,0712	0,0712	0,0712
18	0,0704	0,068	0,0728	0,0728	0,0592	0,06	0,0704	0,0704	0,0704	0,0704	0,0704	0,0704
19	0,0848	0,0848	0,0848	0,0848	0,0808	0,0816	0,0848	0,0848	0,0848	0,0848	0,0848	0,0848
20	0,068	0,068	0,068	0,068	0,0624	0,0672	0,068	0,068	0,068	0,068	0,068	0,068
21	0,0728	0,0744	0,0856	0,0856	0,0672	0,068	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728
22	0,0648	0,0648	0,0648	0,0648	0,0616	0,0656	0,0648	0,0648	0,0648	0,0648	0,0648	0,0648
23	0,0696	0,0832	0,0832	0,0832	0,0704	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696
24	0,0568	0,068	0,0704	0,0704	0,0592	0,0576	0,0568	0,0568	0,0568	0,0568	0,0568	0,0568
25	0,0616	0,0616	0,0616	0,0616	0,0568	0,0608	0,0616	0,0616	0,0616	0,0616	0,0616	0,0616
26	0,0576	0,0656	0,0696	0,0696	0,0616	0,0624	0,0728	0,0728	0,0728	0,0576	0,0576	0,0576
27	0,0712	0,0784	0,0784	0,0784	0,06	0,064	0,0712	0,0712	0,0712	0,0712	0,0712	0,0712
28	0,0624	0,0648	0,0648	0,0648	0,0584	0,0632	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624
29	0,0824	0,0824	0,0824	0,0824	0,0728	0,0736	0,0824	0,0824	0,0824	0,0824	0,0824	0,0824
30	0,076	0,088	0,088	0,088	0,0752	0,072	0,076	0,076	0,076	0,076	0,076	0,076

Fold	Base 2											
	5.000 registros											
	Redes Neurais, 20 Neurônios, BPROP											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0688	0,0688	0,0688	0,0688	0,0648	0,0701	0,0688	0,0688	0,0688	0,0688	0,0688	0,0688
2	0,0712	0,0712	0,0712	0,0712	0,0648	0,0713	0,0712	0,0712	0,0712	0,0712	0,0712	0,0712
3	0,0712	0,0712	0,0712	0,0712	0,0704	0,0736	0,0712	0,0712	0,0712	0,0712	0,0712	0,0712
4	0,0672	0,0672	0,0672	0,0672	0,0616	0,0609	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672
5	0,0568	0,0632	0,0632	0,0632	0,0464	0,0581	0,0584	0,0656	0,0656	0,0568	0,0568	0,0568
6	0,0928	0,0928	0,0928	0,0928	0,076	0,0664	0,0928	0,0928	0,0928	0,0928	0,0928	0,0928
7	0,0856	0,092	0,1024	0,1024	0,0728	0,0748	0,0856	0,0856	0,0856	0,0856	0,0856	0,0856
8	0,056	0,056	0,056	0,056	0,0568	0,0582	0,056	0,056	0,056	0,056	0,056	0,056
9	0,0808	0,0808	0,0808	0,0808	0,072	0,0812	0,0808	0,0808	0,0808	0,0808	0,0808	0,0808
10	0,0656	0,0656	0,0656	0,0656	0,0584	0,0613	0,0688	0,0688	0,0688	0,0661	0,0645	0,0636
11	0,0712	0,0712	0,0712	0,0712	0,06	0,0691	0,0712	0,0712	0,0712	0,0712	0,0712	0,0712
12	0,068	0,068	0,068	0,068	0,0648	0,0636	0,068	0,068	0,068	0,068	0,068	0,068
13	0,0784	0,0784	0,0784	0,0784	0,0736	0,0678	0,0784	0,0784	0,0784	0,0784	0,0784	0,0784
14	0,0816	0,092	0,092	0,092	0,0744	0,0828	0,0816	0,0816	0,0816	0,0816	0,0816	0,0816
15	0,0832	0,0832	0,0832	0,0832	0,0744	0,0824	0,0832	0,0832	0,0832	0,0832	0,0832	0,0832
16	0,08	0,076	0,0888	0,0888	0,0712	0,0791	0,08	0,08	0,08	0,08	0,08	0,08
17	0,0696	0,0752	0,0752	0,0752	0,068	0,0532	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696
18	0,0736	0,0736	0,0736	0,0736	0,0664	0,0483	0,0736	0,0736	0,0736	0,0736	0,0736	0,0736
19	0,0784	0,084	0,084	0,084	0,0768	0,0713	0,0784	0,0784	0,0784	0,0784	0,0784	0,0784
20	0,072	0,072	0,072	0,072	0,0648	0,0701	0,072	0,072	0,072	0,072	0,072	0,072
21	0,0776	0,08	0,08	0,08	0,0688	0,0629	0,0776	0,0776	0,0776	0,0776	0,0776	0,0776
22	0,0752	0,0784	0,0784	0,0784	0,0656	0,0524	0,0752	0,0752	0,0752	0,0752	0,0752	0,0752
23	0,072	0,0776	0,0776	0,0776	0,072	0,0543	0,072	0,072	0,072	0,072	0,072	0,072
24	0,0648	0,0648	0,0648	0,0648	0,064	0,0555	0,0648	0,0648	0,0648	0,0648	0,0648	0,0648
25	0,0592	0,0592	0,0592	0,0592	0,0584	0,0629	0,0592	0,0592	0,0592	0,0592	0,0592	0,0592
26	0,0672	0,0672	0,0672	0,0672	0,0592	0,0528	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672
27	0,0616	0,0616	0,0616	0,0616	0,0592	0,0613	0,0616	0,0616	0,0616	0,0616	0,0616	0,0616
28	0,068	0,0736	0,0736	0,0736	0,0632	0,0622	0,0744	0,0744	0,0744	0,068	0,068	0,068
29	0,0784	0,0784	0,0784	0,0784	0,0752	0,0711	0,0784	0,0784	0,0784	0,0784	0,0784	0,0784
30	0,0944	0,0944	0,0944	0,0944	0,0728	0,0784	0,0944	0,0944	0,0944	0,0944	0,0944	0,0944

Fold	Base 2											
	5.000 registros											
	Redes Neurais, 10 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,068	0,0784	0,0784	0,0784	0,06	0,0712	0,0736	0,0736	0,0736	0,068	0,068	0,068
2	0,0848	0,1008	0,1008	0,1008	0,0616	0,0696	0,0848	0,0848	0,0848	0,0848	0,0848	0,0848
3	0,088	0,0928	0,0928	0,0928	0,0664	0,0712	0,088	0,088	0,088	0,088	0,088	0,088
4	0,0808	0,0808	0,0808	0,0808	0,0576	0,06	0,0808	0,0808	0,0808	0,0808	0,0808	0,0808
5	0,0792	0,0888	0,0816	0,0816	0,056	0,0672	0,0728	0,0832	0,0832	0,0792	0,0792	0,0792
6	0,0856	0,0856	0,0856	0,0856	0,0808	0,0776	0,0856	0,0856	0,0856	0,0856	0,0856	0,0856
7	0,0944	0,0912	0,0912	0,0912	0,0768	0,0808	0,0944	0,0944	0,0944	0,0944	0,0944	0,0944
8	0,0904	0,0816	0,0816	0,0816	0,0624	0,0704	0,064	0,064	0,064	0,0840	0,0823	0,0812
9	0,0824	0,0824	0,0824	0,0824	0,076	0,0736	0,0824	0,0824	0,0824	0,0824	0,0824	0,0824
10	0,0856	0,0928	0,0928	0,0928	0,0584	0,0648	0,0856	0,0856	0,0856	0,0856	0,0856	0,0856
11	0,092	0,0752	0,0752	0,0752	0,0648	0,0664	0,0968	0,0968	0,0968	0,092	0,092	0,092
12	0,0632	0,0632	0,0632	0,0632	0,0592	0,06	0,0632	0,0632	0,0632	0,0632	0,0632	0,0632
13	0,0944	0,0944	0,0944	0,0944	0,0744	0,08	0,0832	0,0832	0,0832	0,0944	0,0944	0,0944
14	0,084	0,084	0,084	0,084	0,0776	0,0744	0,084	0,084	0,084	0,084	0,084	0,084
15	0,104	0,092	0,092	0,092	0,068	0,0784	0,0976	0,0976	0,0976	0,0972	0,0971	0,0970
16	0,084	0,0856	0,0856	0,0856	0,0736	0,0856	0,084	0,084	0,084	0,084	0,084	0,084
17	0,0984	0,1048	0,1048	0,1048	0,0648	0,0656	0,0856	0,0856	0,0856	0,0896	0,0867	0,0850
18	0,072	0,08	0,08	0,08	0,0648	0,0696	0,072	0,072	0,072	0,072	0,072	0,072
19	0,1064	0,1024	0,1024	0,1024	0,0776	0,0728	0,0936	0,0936	0,0936	0,1051	0,1031	0,1020
20	0,088	0,0872	0,0856	0,0856	0,0632	0,0664	0,088	0,088	0,088	0,088	0,088	0,088
21	0,1016	0,1024	0,0936	0,0936	0,0728	0,0752	0,112	0,112	0,112	0,0998	0,0936	0,0900
22	0,0824	0,0824	0,0824	0,0824	0,072	0,0648	0,0824	0,0824	0,0824	0,0824	0,0824	0,0824
23	0,08	0,08	0,08	0,08	0,0696	0,0744	0,08	0,08	0,08	0,08	0,08	0,08
24	0,0696	0,0736	0,0736	0,0736	0,0632	0,0672	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696
25	0,0712	0,0872	0,0872	0,0872	0,0576	0,0648	0,0712	0,0712	0,0712	0,0712	0,0712	0,0712
26	0,0792	0,0672	0,0672	0,0672	0,06	0,0672	0,0792	0,0792	0,0792	0,0792	0,0792	0,0792
27	0,0792	0,092	0,0824	0,0824	0,0568	0,0648	0,0784	0,0784	0,0784	0,0883	0,0885	0,0886
28	0,0744	0,0864	0,0928	0,0928	0,0664	0,0672	0,0744	0,0744	0,0744	0,0756	0,0737	0,0726
29	0,108	0,0904	0,0904	0,0904	0,0792	0,0968	0,108	0,108	0,108	0,108	0,108	0,108
30	0,088	0,088	0,088	0,088	0,0744	0,0848	0,088	0,088	0,088	0,088	0,088	0,088

Fold	Base 2											
	5.000 registros											
	Redes Neurais, 20 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0604	0,0604	0,0604	0,0604	0,0593	0,0620	0,0604	0,0604	0,0604	0,0604	0,0604	0,0604
2	0,0718	0,0718	0,0718	0,0718	0,0645	0,0703	0,0976	0,0976	0,0976	0,0916	0,0892	0,0879
3	0,0705	0,0705	0,0705	0,0705	0,0652	0,0723	0,1	0,1	0,1	0,1067	0,1062	0,1059
4	0,0681	0,0681	0,0681	0,0681	0,0694	0,0556	0,0681	0,0681	0,0681	0,0681	0,0681	0,0681
5	0,0498	0,0686	0,0686	0,0686	0,0308	0,0505	0,076	0,0816	0,0816	0,0713	0,0680	0,0661
6	0,1046	0,1046	0,1046	0,1046	0,0758	0,0683	0,1046	0,1046	0,1046	0,1046	0,1046	0,1046
7	0,0912	0,0835	0,0902	0,0902	0,0789	0,0766	0,0912	0,0912	0,0912	0,0912	0,0912	0,0912
8	0,0418	0,0418	0,0418	0,0418	0,0606	0,0563	0,0418	0,0418	0,0418	0,0770	0,0757	0,0749
9	0,0851	0,0851	0,0851	0,0851	0,0784	0,0795	0,0851	0,0851	0,0851	0,0851	0,0851	0,0851
10	0,0557	0,0740	0,0740	0,0740	0,0575	0,0617	0,0557	0,0557	0,0557	0,0557	0,0557	0,0557
11	0,0786	0,0786	0,0786	0,0786	0,0704	0,0606	0,0786	0,0786	0,0786	0,0985	0,0931	0,0899
12	0,0785	0,0785	0,0785	0,0785	0,0733	0,0582	0,0785	0,0785	0,0785	0,0785	0,0785	0,0785
13	0,0776	0,0776	0,0776	0,0776	0,0688	0,0692	0,0776	0,0776	0,0776	0,0904	0,0904	0,0904
14	0,0786	0,0844	0,0844	0,0844	0,0737	0,0865	0,0786	0,0786	0,0786	0,0961	0,0961	0,0961
15	0,0887	0,0887	0,0887	0,0887	0,0755	0,0873	0,0887	0,0887	0,0887	0,0832	0,0822	0,0816
16	0,0793	0,0865	0,1023	0,1023	0,0626	0,0924	0,0793	0,0793	0,0793	0,0793	0,0793	0,0793
17	0,0752	0,0775	0,0775	0,0775	0,0708	0,0498	0,0816	0,0816	0,0816	0,0791	0,0776	0,0767
18	0,0745	0,0745	0,0745	0,0745	0,0614	0,0482	0,0745	0,0745	0,0745	0,0745	0,0745	0,0745
19	0,0759	0,0904	0,0904	0,0904	0,0751	0,0683	0,104	0,1024	0,1024	0,1046	0,1006	0,0983
20	0,0750	0,0750	0,0750	0,0750	0,0633	0,0591	0,0750	0,0750	0,0750	0,0750	0,0750	0,0750
21	0,0756	0,0793	0,0793	0,0793	0,0712	0,0616	0,0756	0,0756	0,0756	0,0793	0,0793	0,0793
22	0,0782	0,0892	0,0892	0,0892	0,0729	0,0433	0,0904	0,0904	0,0904	0,0782	0,0782	0,0782
23	0,0601	0,0722	0,0722	0,0722	0,0728	0,0485	0,0601	0,0601	0,0601	0,0601	0,0601	0,0601
24	0,0628	0,0628	0,0628	0,0628	0,0694	0,0502	0,0628	0,0628	0,0628	0,0628	0,0628	0,0628
25	0,0490	0,0490	0,0490	0,0490	0,0566	0,0673	0,0936	0,0936	0,0936	0,0490	0,0490	0,0490
26	0,0666	0,0666	0,0666	0,0666	0,0657	0,0498	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666
27	0,0684	0,0684	0,0684	0,0684	0,0637	0,0599	0,0768	0,0768	0,0768	0,0684	0,0684	0,0684
28	0,0623	0,0814	0,0814	0,0814	0,0635	0,0702	0,0623	0,0623	0,0623	0,0623	0,0623	0,0623
29	0,0761	0,0761	0,0761	0,0761	0,0783	0,0763	0,0761	0,0761	0,0761	0,0761	0,0761	0,0761
30	0,0979	0,0979	0,0979	0,0979	0,0667	0,0715	0,1072	0,1072	0,1072	0,0979	0,0979	0,0979

Tabela A.13 – Erros de Classificação da Base (3) para 1.000 registros.

Fold	Base 3 – 1.000 registros – Regressão Linear												Regressão Logística											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,108	0,064	0,084	0,084	0,096	0,092	0,072	0,08	0,08	0,0770	0,0955	0,0955	0,08	0,08	0,076	0,076	0,076	0,076	0,08	0,08	0,08	0,08	0,08	0,08
2	0,06	0,064	0,052	0,052	0,064	0,068	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,052	0,064	0,064	0,056	0,056	0,06	0,064	0,064	0,0572	0,0572	0,0572
3	0,076	0,056	0,056	0,056	0,076	0,076	0,064	0,084	0,084	0,0772	0,0823	0,0819	0,056	0,056	0,056	0,056	0,048	0,048	0,056	0,056	0,056	0,056	0,056	0,056
4	0,084	0,076	0,06	0,06	0,076	0,08	0,052	0,048	0,048	0,0780	0,0895	0,0895	0,052	0,06	0,072	0,072	0,052	0,048	0,048	0,048	0,048	0,052	0,052	0,052
5	0,1	0,08	0,072	0,072	0,104	0,104	0,072	0,072	0,072	0,0864	0,0767	0,0769	0,06	0,068	0,068	0,068	0,068	0,068	0,06	0,06	0,06	0,06	0,06	0,06
6	0,068	0,088	0,088	0,088	0,064	0,06	0,092	0,088	0,088	0,068	0,068	0,068	0,064	0,068	0,072	0,072	0,06	0,064	0,056	0,064	0,064	0,064	0,064	0,064
7	0,14	0,068	0,072	0,072	0,152	0,164	0,056	0,064	0,064	0,0766	0,0971	0,0974	0,068	0,064	0,068	0,068	0,06	0,06	0,068	0,068	0,068	0,0779	0,0850	0,0876
8	0,088	0,088	0,088	0,088	0,08	0,076	0,096	0,088	0,088	0,0770	0,0607	0,0600	0,072	0,072	0,076	0,076	0,06	0,06	0,072	0,072	0,072	0,072	0,072	0,072
9	0,076	0,064	0,06	0,06	0,076	0,092	0,076	0,064	0,064	0,0860	0,0635	0,0609	0,06	0,056	0,06	0,06	0,064	0,064	0,06	0,06	0,06	0,0542	0,0635	0,0609
10	0,088	0,076	0,088	0,088	0,08	0,08	0,076	0,08	0,08	0,0749	0,0904	0,0966	0,06	0,06	0,06	0,06	0,064	0,06	0,068	0,068	0,068	0,06	0,06	0,06
11	0,064	0,068	0,068	0,068	0,068	0,068	0,064	0,064	0,064	0,064	0,064	0,064	0,06	0,068	0,076	0,076	0,068	0,068	0,064	0,064	0,064	0,0731	0,0731	0,0731
12	0,064	0,064	0,064	0,064	0,076	0,072	0,064	0,064	0,064	0,064	0,064	0,064	0,072	0,068	0,068	0,068	0,076	0,068	0,06	0,06	0,06	0,0699	0,0699	0,0699
13	0,052	0,06	0,064	0,064	0,056	0,056	0,076	0,068	0,068	0,0889	0,0676	0,0691	0,044	0,044	0,044	0,044	0,048	0,044	0,044	0,044	0,044	0,044	0,044	0,044
14	0,124	0,104	0,116	0,116	0,128	0,12	0,108	0,096	0,096	0,124	0,124	0,124	0,112	0,12	0,108	0,108	0,1	0,1	0,104	0,104	0,104	0,1312	0,1312	0,1312
15	0,084	0,068	0,072	0,072	0,072	0,072	0,048	0,052	0,052	0,0855	0,0654	0,0665	0,06	0,056	0,06	0,06	0,06	0,06	0,068	0,06	0,06	0,0497	0,0548	0,0548
16	0,08	0,088	0,076	0,076	0,068	0,068	0,068	0,088	0,088	0,08	0,08	0,08	0,092	0,072	0,076	0,076	0,076	0,084	0,092	0,092	0,092	0,092	0,092	0,092
17	0,084	0,072	0,08	0,08	0,092	0,092	0,092	0,076	0,076	0,0855	0,086	0,0864	0,076	0,08	0,08	0,08	0,076	0,076	0,076	0,076	0,076	0,076	0,076	0,076
18	0,084	0,084	0,084	0,084	0,08	0,072	0,076	0,096	0,096	0,0811	0,1031	0,1031	0,064	0,068	0,072	0,072	0,064	0,068	0,064	0,064	0,064	0,064	0,064	0,064
19	0,044	0,064	0,064	0,064	0,048	0,048	0,044	0,044	0,044	0,0775	0,0632	0,0614	0,052	0,052	0,052	0,052	0,056	0,052	0,068	0,068	0,068	0,0524	0,0527	0,0529
20	0,084	0,072	0,088	0,088	0,068	0,072	0,064	0,072	0,072	0,0760	0,0736	0,0708	0,06	0,076	0,068	0,068	0,06	0,06	0,064	0,06	0,06	0,0600	0,0600	0,0600
21	0,072	0,076	0,084	0,084	0,072	0,076	0,072	0,072	0,072	0,072	0,072	0,072	0,064	0,068	0,072	0,072	0,068	0,068	0,076	0,076	0,076	0,064	0,064	0,064
22	0,084	0,06	0,056	0,056	0,08	0,084	0,06	0,08	0,08	0,0857	0,1340	0,1362	0,068	0,06	0,06	0,06	0,064	0,068	0,068	0,068	0,068	0,075	0,075	0,075
23	0,104	0,1	0,1	0,1	0,128	0,128	0,08	0,076	0,076	0,104	0,104	0,104	0,088	0,076	0,072	0,072	0,084	0,088	0,08	0,088	0,088	0,0928	0,0912	0,0903
24	0,04	0,06	0,052	0,052	0,044	0,044	0,028	0,028	0,028	0,0906	0,0441	0,0466	0,032	0,028	0,032	0,032	0,032	0,032	0,028	0,028	0,028	0,0320	0,0421	0,0421
25	0,076	0,092	0,1	0,1	0,064	0,076	0,08	0,088	0,088	0,076	0,076	0,076	0,06	0,06	0,06	0,06	0,064	0,064	0,068	0,068	0,068	0,06	0,06	0,06
26	0,052	0,04	0,048	0,048	0,056	0,056	0,052	0,052	0,052	0,052	0,052	0,052	0,048	0,056	0,052	0,052	0,052	0,052	0,056	0,052	0,052	0,048	0,048	0,048
27	0,084	0,072	0,072	0,072	0,084	0,084	0,068	0,068	0,068	0,084	0,084	0,084	0,068	0,084	0,08	0,08	0,068	0,064	0,072	0,072	0,072	0,0748	0,0748	0,0748
28	0,068	0,068	0,068	0,068	0,072	0,072	0,068	0,068	0,068	0,068	0,068	0,068	0,056	0,052	0,06	0,06	0,056	0,052	0,056	0,056	0,056	0,0677	0,0677	0,0677
29	0,068	0,068	0,068	0,068	0,072	0,072	0,068	0,068	0,068	0,068	0,068	0,068	0,056	0,048	0,06	0,06	0,056	0,052	0,056	0,056	0,056	0,056	0,056	0,056
30	0,084	0,096	0,104	0,104	0,084	0,084	0,084	0,084	0,084	0,084	0,084	0,084	0,064	0,064	0,064	0,064	0,064	0,068	0,064	0,064	0,064	0,064	0,064	0,064

Fold	Base 3 – 1.000 registros – Redes Neurais, 3 Neurônios, BPROP												Redes Neurais, 3 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,1004	0,1004	0,1004	0,1004		0,0651	0,112	0,1	0,1	0,1051	0,1056	0,1060	0,084	0,104	0,096	0,096	0,088	0,08	0,12	0,12	0,12	0,1159	0,1138	0,1125
2	0,0879	0,0746	0,0746	0,0746	0,0851	0,0924	0,0879	0,0879	0,0879	0,0681	0,0640	0,0615	0,064	0,104	0,104	0,104	0,072	0,072	0,092	0,084	0,084	0,0653	0,0661	0,0665
3	0,0542	0,0850	0,0751	0,0751	0,0623	0,0854	0,0542	0,0542	0,0542	0,0775	0,0775	0,0775	0,092	0,04	0,076	0,076	0,052	0,08	0,1	0,124	0,124	0,0775	0,0783	0,0787
4	0,1086	0,1086	0,1086	0,1086	0,0703	0,0777	0,1086	0,1086	0,1086	0,1086	0,1086	0,1086	0,064	0,096	0,096	0,096	0,048	0,052	0,072	0,072	0,072	0,064	0,064	0,064
5	0,0604	0,0944	0,0803	0,0803	0,0753	0,0796	0,0604	0,0604	0,0604	0,0712	0,0708	0,0706	0,068	0,076	0,076	0,076	0,076	0,076	0,068	0,068	0,068	0,0738	0,0767	0,0784
6	0,0903	0,0824	0,0730	0,0730	0,0617	0,0917	0,0903	0,0903	0,0903	0,0903	0,0903	0,0903	0,088	0,1	0,104	0,104	0,064	0,08	0,088	0,088	0,088	0,0849	0,0854	0,0857
7	0,0943	0,0808	0,0808	0,0808	0,0655	0,0781	0,0943	0,0943	0,0943	0,0943	0,0943	0,0943	0,052	0,084	0,084	0,084	0,064	0,064	0,052	0,052	0,052	0,0672	0,0708	0,0730
8	0,1080	0,1342	0,1161	0,1161	0,0673	0,0715	0,088	0,088	0,088	0,0806	0,0748	0,0714	0,1	0,076	0,076	0,076	0,08	0,072	0,084	0,084	0,084	0,0779	0,0708	0,0665
9	0,0807	0,0704	0,0720	0,0720	0,0770	0,0712	0,0807	0,0807	0,0807	0,0569	0,0553	0,0543	0,056	0,076	0,076	0,076	0,064	0,06	0,056	0,056	0,056	0,0542	0,0532	0,0527
10	0,1123	0,0862	0,0862	0,0862	0,0852	0,0779	0,1123	0,1123	0,1123	0,1123	0,1123	0,1123	0,076	0,076	0,076	0,076	0,076	0,08	0,076	0,076	0,076	0,076	0,076	0,076
11	0,0870	0,1094	0,1094	0,1094	0,0707	0,0812	0,0870	0,0870	0,0870	0,0870	0,0870	0,0870	0,06	0,06	0,06	0,06	0,068	0,064	0,06	0,06	0,06	0,06	0,06	0,06
12	0,0945	0,0879	0,0879	0,0879	0,0752	0,0813	0,0945	0,0945	0,0945	0,0945	0,0945	0,0945	0,064	0,072	0,072	0,072	0,068	0,096	0,072	0,096	0,096	0,064	0,064	0,064
13	0,0954	0,0993	0,0993	0,0993	0,0633	0,0774	0,076	0,076	0,076	0,0596	0,0614	0,0626	0,064	0,096	0,096	0,096	0,052	0,08	0,064	0,064	0,064	0,0731	0,0737	0,0741
14	0,1202	0,1089	0,1089	0,1089	0,0859	0,0765	0,116	0,116	0,116	0,1202	0,1202	0,1202	0,124	0,092	0,092	0,092	0,108	0,1	0,116	0,116	0,116	0,1417	0,1464	0,1493
15	0,0768	0,0788	0,0788	0,0788	0,0782	0,0979	0,076	0,076	0,076	0,0690	0,0717	0,0733	0,064	0,084	0,084	0,084	0,056	0,08	0,076	0,076	0,076	0,0662	0,0654	0,0648
16	0,0820	0,0934	0,0934	0,0934	0,0824	0,0808	0,0820	0,0820	0,0820	0,0820	0,0820	0,0820	0,08	0,06	0,092	0,092	0,072	0,088	0,076	0,076	0,076	0,0833	0,0769	0,0730
17	0,0788	0,0679	0,0679	0,0679	0,0796	0,0701	0,0788	0,0788	0,0788	0,0788	0,0788	0,0788	0,064	0,064	0,064	0,064	0,08	0,072	0,064	0,064	0,064	0,064	0,064	0,064
18	0,0673	0,1010	0,1004	0,1004	0,0835	0,0796	0,0673	0,0673	0,0673	0,0968	0,0968	0,0968	0,084	0,084	0,084	0,084	0,068	0,108	0,084	0,084	0,084	0,084	0,084	0,084
19	0,0493	0,0456	0,0456	0,0456	0,0683	0,0651	0,0493	0,0493	0,0493	0,0552	0,0548	0,0546	0,06	0,06	0,06	0,06	0,056	0,06	0,06	0,06	0,06	0,0635	0,0632	0,0631
20	0,0859	0,1018	0,1033	0,1033	0,0888	0,0774	0,0859	0,0859	0,0859	0,0652	0,0678	0,0693	0,06	0,076	0,076	0,076	0,064	0,072	0,06	0,06	0,06	0,0757	0,0813	0,0847
21	0,0869	0,1036	0,0974	0,0974	0,0866	0,0682	0,0869	0,0869	0,0869	0,0869	0,0869	0,0869	0,06	0,084	0,104	0,104	0,064	0,064	0,06	0,06	0,06	0,06	0,06	0,06
22	0,0732	0,1000	0,0926	0,0926	0,0722	0,0747	0,0732	0,0732	0,0732	0,0732	0,0732	0,0732	0,06	0,06	0,06	0,06	0,068	0,068	0,06	0,06	0,06	0,06	0,06	0,06
23	0,1080	0,1196	0,1148	0,1148	0,0760	0,0747	0,1	0,1	0,1	0,0956	0,0933	0,0919	0,152	0,092	0,084	0,084	0,092	0,116	0,096	0,096	0,096	0,1092	0,1037	0,1003
24	0,0549	0,0551	0,0508	0,0508	0,0782	0,0756	0,0549	0,0549	0,0549	0,0549	0,0549	0,0549	0,04	0,044	0,04	0,04	0,032	0,04	0,04	0,04	0,04	0,0374	0,0361	0,0353
25	0,1067	0,0980	0,0905	0,0905	0,0775	0,0736	0,1067	0,1067	0,1067	0,1067	0,1067	0,1067	0,08	0,088	0,088	0,088	0,1	0,092	0,084	0,084	0,084	0,0926	0,0952	0,0968
26	0,0612	0,0619	0,0445	0,0445	0,0697	0,0862	0,0612	0,0612	0,0612	0,0666	0,07	0,072	0,076	0,076	0,076	0,076	0,04	0,064	0,076	0,076	0,076	0,0613	0,054	0,0496
27	0,0970	0,0826	0,0808	0,0808	0,0846	0,0641	0,0970	0,0970	0,0970	0,0935	0,0963	0,0980	0,068	0,068	0,068	0,068	0,08	0,092	0,068	0,068	0,068	0,068	0,068	0,068
28	0,0941	0,0946	0,0922	0,0922	0,0696	0,0667	0,084	0,084	0,084	0,0903	0,0895	0,0889	0,108	0,08	0,08	0,08	0,056	0,084	0,084	0,084	0,084	0,1045	0,1026	0,1014
29	0,0477	0,0797	0,0797	0,0797	0,076	0,0882	0,072	0,072	0,072	0,0477	0,0477	0,0477	0,084	0,06	0,06	0,06	0,064	0,068	0,092	0,092	0,092	0,0666	0,0666	0,0666
30	0,0960	0,0960	0,0960	0,0960	0,076	0,0658	0,0960	0,0960	0,0960	0,0960	0,0960	0,0960	0,092	0,1	0,112	0,112	0,076	0,076	0,092	0,092	0,092	0,0989	0,0983	0,0980

Tabela A.14 – Erros de Classificação da Base (3) para 1.000 registros (continuação).

Fold	Base 3																							
	1.000 registros																							
	Redes Neurais, 10 Neurônios, BPROP												Redes Neurais, 10 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,1	0,1	0,1	0,1	0,088	0,092	0,1	0,1	0,1	0,1	0,1	0,1	0,0918	0,0918	0,0918	0,0918	0,096	0,0762	0,104	0,104	0,104	0,1185	0,1178	0,1174
2	0,084	0,072	0,072	0,072	0,064	0,052	0,092	0,092	0,092	0,0817	0,0743	0,0698	0,0866	0,0613	0,0613	0,0613	0,072	0,0525	0,088	0,088	0,088	0,0926	0,0888	0,0865
3	0,064	0,088	0,068	0,068	0,048	0,06	0,064	0,064	0,064	0,0748	0,0763	0,0771	0,0668	0,0923	0,0547	0,0547	0,056	0,0676	0,088	0,088	0,088	0,0775	0,0783	0,0787
4	0,104	0,104	0,104	0,104	0,068	0,064	0,104	0,104	0,104	0,0788	0,0746	0,0721	0,1159	0,1159	0,1159	0,1159	0,068	0,0860	0,084	0,084	0,084	0,0763	0,0727	0,0706
5	0,064	0,088	0,076	0,076	0,06	0,048	0,064	0,064	0,064	0,0606	0,0590	0,0580	0,0652	0,0887	0,0808	0,0808	0,064	0,0610	0,0652	0,0652	0,0652	0,0652	0,0652	0,0652
6	0,088	0,088	0,08	0,08	0,08	0,084	0,1	0,1	0,1	0,0904	0,0895	0,0890	0,0759	0,0896	0,0877	0,0877	0,092	0,0785	0,108	0,108	0,108	0,0904	0,0895	0,0890
7	0,092	0,076	0,076	0,076	0,064	0,076	0,1	0,08	0,08	0,0940	0,0910	0,0892	0,0899	0,0588	0,0588	0,0588	0,076	0,0853	0,0899	0,0899	0,0899	0,0833	0,0829	0,0827
8	0,108	0,128	0,124	0,124	0,072	0,08	0,096	0,096	0,096	0,0806	0,0748	0,0714	0,1114	0,1350	0,1300	0,1300	0,088	0,0483	0,116	0,104	0,104	0,0887	0,0809	0,0762
9	0,076	0,072	0,068	0,068	0,06	0,068	0,096	0,096	0,096	0,0650	0,0614	0,0593	0,0833	0,0747	0,0691	0,0691	0,064	0,0645	0,0833	0,0833	0,0833	0,0677	0,0635	0,0609
10	0,1	0,092	0,092	0,092	0,076	0,08	0,1	0,1	0,1	0,1	0,1	0,1	0,0969	0,0829	0,0829	0,0829	0,092	0,0839	0,0969	0,0969	0,0969	0,1011	0,1018	0,1022
11	0,092	0,108	0,108	0,108	0,072	0,056	0,088	0,088	0,088	0,0921	0,0881	0,0856	0,0902	0,1123	0,1123	0,1123	0,08	0,0704	0,076	0,076	0,076	0,0867	0,0840	0,0823
12	0,088	0,088	0,088	0,088	0,068	0,072	0,088	0,088	0,088	0,088	0,088	0,088	0,0828	0,1002	0,1002	0,1002	0,068	0,0705	0,0828	0,0828	0,0828	0,0880	0,0862	0,0851
13	0,096	0,096	0,096	0,096	0,06	0,064	0,076	0,076	0,076	0,0731	0,0635	0,0576	0,1093	0,0964	0,0964	0,0964	0,064	0,0623	0,1093	0,1093	0,1093	0,0677	0,0594	0,0543
14	0,124	0,104	0,104	0,104	0,116	0,116	0,124	0,124	0,124	0,1312	0,1367	0,1399	0,1163	0,1072	0,1072	0,1072	0,1	0,0580	0,12	0,112	0,112	0,1417	0,1484	0,1524
15	0,08	0,088	0,088	0,088	0,052	0,064	0,076	0,076	0,076	0,0745	0,0759	0,0767	0,0913	0,0865	0,0865	0,0865	0,052	0,0795	0,0913	0,0913	0,0913	0,0607	0,0611	0,0614
16	0,088	0,1	0,1	0,1	0,068	0,072	0,088	0,088	0,088	0,0698	0,0647	0,0616	0,0915	0,1046	0,1046	0,1046	0,08	0,0755	0,108	0,108	0,108	0,0887	0,0850	0,0827
17	0,08	0,064	0,064	0,064	0,06	0,068	0,068	0,068	0,068	0,072	0,076	0,0784	0,0797	0,0584	0,0584	0,0584	0,064	0,0401	0,0797	0,0797	0,0797	0,08	0,08	0,08
18	0,076	0,108	0,104	0,104	0,08	0,092	0,104	0,096	0,096	0,1020	0,1070	0,1099	0,0811	0,1035	0,0946	0,0946	0,076	0,0616	0,0811	0,0811	0,0811	0,0994	0,1031	0,1052
19	0,048	0,052	0,052	0,052	0,06	0,06	0,048	0,048	0,048	0,0580	0,0590	0,0597	0,0397	0,0567	0,0567	0,0567	0,052	0,0659	0,0397	0,0397	0,0397	0,0718	0,0759	0,0784
20	0,088	0,096	0,1	0,1	0,064	0,072	0,084	0,084	0,084	0,0626	0,0581	0,0554	0,0829	0,0994	0,0998	0,0998	0,072	0,0706	0,0829	0,0829	0,0829	0,0600	0,0562	0,0539
21	0,08	0,1	0,088	0,088	0,068	0,068	0,08	0,08	0,08	0,0821	0,0789	0,0769	0,0750	0,1008	0,0852	0,0852	0,076	0,1135	0,088	0,088	0,088	0,0651	0,0657	0,0661
22	0,08	0,108	0,092	0,092	0,068	0,072	0,096	0,096	0,096	0,0972	0,1106	0,1189	0,0816	0,1114	0,0821	0,0821	0,068	0,0787	0,0816	0,0816	0,0816	0,1194	0,1276	0,1327
23	0,112	0,112	0,116	0,116	0,088	0,104	0,1	0,1	0,1	0,1038	0,0995	0,0969	0,1192	0,1141	0,1176	0,1176	0,112	0,0775	0,1192	0,1192	0,1192	0,1192	0,1192	0,1192
24	0,056	0,048	0,052	0,052	0,028	0,044	0,056	0,056	0,056	0,0454	0,0454	0,0454	0,0533	0,0441	0,0503	0,0503	0,028	0,1100	0,076	0,076	0,076	0,0588	0,0481	0,0418
25	0,112	0,104	0,092	0,092	0,096	0,1	0,112	0,112	0,112	0,0926	0,0909	0,0897	0,1167	0,1013	0,0918	0,0918	0,096	0,0838	0,1167	0,1167	0,1167	0,1167	0,1167	0,1167
26	0,064	0,056	0,052	0,052	0,056	0,044	0,064	0,064	0,064	0,064	0,064	0,064	0,0679	0,0560	0,0543	0,0543	0,052	0,0927	0,0679	0,0679	0,0679	0,0533	0,048	0,0448
27	0,096	0,084	0,08	0,08	0,088	0,092	0,096	0,096	0,096	0,0989	0,0983	0,0980	0,0993	0,0880	0,0717	0,0717	0,088	0,0865	0,112	0,112	0,112	0,1016	0,1004	0,0996
28	0,096	0,1	0,1	0,1	0,072	0,096	0,096	0,096	0,096	0,0960	0,0895	0,0854	0,0948	0,0872	0,1020	0,1020	0,068	0,0897	0,0948	0,0948	0,0948	0,1045	0,0960	0,0907
29	0,056	0,08	0,08	0,08	0,064	0,064	0,08	0,08	0,08	0,0506	0,052	0,0528	0,0665	0,0860	0,0860	0,0860	0,06	0,0692	0,072	0,116	0,116	0,0666	0,064	0,0624
30	0,1	0,1	0,1	0,1	0,064	0,084	0,1	0,1	0,1	0,1	0,1	0,1	0,0937	0,0937	0,0937	0,0937	0,072	0,0918	0,0937	0,0937	0,0937	0,0937	0,0937	0,0937

Fold	Base 3																							
	1.000 registros																							
	Redes Neurais, 20 Neurônios, BPROP												Redes Neurais, 20 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,096	0,112	0,112	0,112	0,104	0,104	0,124	0,124	0,124	0,0889	0,0934	0,0962	0,0883	0,1026	0,1026	0,1026	0,0990	0,1155	0,1302	0,1267	0,1267	0,0887	0,0927	0,0826
2	0,076	0,076	0,076	0,076	0,072	0,076	0,076	0,076	0,076	0,0844	0,0847	0,0848	0,0820	0,0820	0,0820	0,0820	0,0789	0,0747	0,0820	0,0820	0,0820	0,0801	0,0661	0,0960
3	0,088	0,06	0,06	0,06	0,064	0,06	0,088	0,088	0,088	0,0828	0,0823	0,0819	0,0857	0,0569	0,0569	0,0569	0,0596	0,0459	0,0857	0,0857	0,0857	0,0796	0,0799	0,0870
4	0,076	0,072	0,072	0,072	0,08	0,092	0,084	0,084	0,084	0,0661	0,0652	0,0648	0,0674	0,0668	0,0668	0,0668	0,0850	0,0927	0,0835	0,0835	0,0835	0,0643	0,0631	0,0656
5	0,072	0,068	0,088	0,088	0,06	0,068	0,072	0,072	0,072	0,0738	0,0748	0,0753	0,0730	0,0743	0,0900	0,0900	0,0600	0,0689	0,0730	0,0730	0,0730	0,0720	0,0726	0,0761
6	0,092	0,1	0,112	0,112	0,068	0,084	0,092	0,092	0,092	0,092	0,092	0,092	0,0983	0,0845	0,1152	0,1152	0,0637	0,0918	0,0983	0,0983	0,0983	0,0983	0,0983	0,0983
7	0,104	0,104	0,104	0,104	0,064	0,076	0,104	0,104	0,104	0,0860	0,0829	0,0811	0,1018	0,1018	0,1018	0,1018	0,0577	0,0764	0,1018	0,1018	0,1018	0,0943	0,0891	0,0738
8	0,108	0,116	0,116	0,116	0,076	0,08	0,096	0,108	0,108	0,0833	0,0769	0,0730	0,0940	0,1127	0,1127	0,1127	0,0691	0,0790	0,0826	0,0975	0,0975	0,0880	0,0862	0,0729
9	0,072	0,064	0,076	0,076	0,06	0,056	0,08	0,08	0,08	0,0758	0,0696	0,0658	0,0752	0,0562	0,0730	0,0730	0,0594	0,0495	0,0904	0,0904	0,0904	0,0739	0,0676	0,0683
10	0,1	0,1	0,104	0,104	0,064	0,084	0,108	0,108	0,108	0,1011	0,1018	0,1022	0,0969	0,1061	0,0988	0,0988	0,0696	0,0899	0,1184	0,1184	0,1184	0,0956	0,0942	0,1042
11	0,096	0,092	0,1	0,1	0,068	0,088	0,096	0,096	0,096	0,0867	0,0840	0,0823	0,0998	0,0887	0,1060	0,1060	0,0650	0,0946	0,0998	0,0998	0,0998	0,0811	0,0764	0,0843
12	0,104	0,128	0,12	0,12	0,084	0,08	0,104	0,104	0,104	0,1010	0,0996	0,0987	0,1015	0,1226	0,1086	0,1086	0,0784	0,0836	0,1015	0,1015	0,1015	0,1049	0,0945	0,1059
13	0,084	0,1	0,104	0,104	0,076	0,08	0,084	0,084	0,084	0,0704	0,0614	0,0560	0,0860	0,0945	0,0987	0,0987	0,0751	0,0764	0,0860	0,0860	0,0860	0,0743	0,0564	0,0632
14	0,12	0,136	0,12	0,12	0,104	0,112	0,112	0,116	0,116	0,1391	0,1484	0,1539	0,1161	0,1412	0,1183	0,1183	0,0992	0,1062	0,1146	0,1236	0,1236	0,1403	0,1467	0,1568
15	0,076	0,088	0,088	0,088	0,064	0,048	0,076	0,076	0,076	0,0828	0,0822	0,0819	0,0695	0,0867	0,0867	0,0867	0,0686	0,0507	0,0695	0,0695	0,0695	0,0841	0,0805	0,0847
16	0,096	0,076	0,076	0,076	0,08	0,088	0,096	0,096	0,096	0,0887	0,0829	0,0795	0,0992	0,0810	0,0810	0,0810	0,0743	0,0967	0,0992	0,0992	0,0992	0,0765	0,0849	0,0838
17	0,064	0,072	0,08	0,08	0,068	0,06	0,064	0,064	0,064	0,0826	0,084	0,0848	0,0667	0,0674	0,0672	0,0672	0,0599	0,0655	0,0667	0,0667	0,0667	0,0705	0,0859	0,0891
18	0,092	0,092	0,092	0,092	0,088	0,104	0,092	0,092	0,092	0,092	0,092	0,092	0,0860	0,0860	0,0860	0,0860	0,0807	0,1023	0,0860	0,0860	0,0860	0,0860	0,0860	0,0860
19	0,052	0,068	0,072	0,072	0,064	0,088	0,068	0,072	0,072	0,0662	0,0654	0,0648	0,0509	0,0701	0,0687	0,0687	0,0619	0,0905	0,0645	0,0670	0,0670	0,0657	0,0714	0,0739
20	0,076	0,092	0,092	0,092	0,072	0,072	0,092	0,092	0,092	0,0574	0,0542	0,0523	0,0707	0,0810	0,0810	0,0810	0,0695	0,0869	0,1072	0,0954	0,0954	0,0499	0,0568	0,0437
21	0,092	0,104	0,1	0,1	0,068	0,08	0,104	0,104	0,104	0,0708	0,0701	0,0697	0,0909	0,1043	0,1071	0,1071	0,0737	0,0683	0,1025	0,1025	0,1025	0,0739	0,0773	0,0778
22	0,092	0,092	0,092	0,092	0,068	0,068	0,092	0,092	0,092	0,092	0,092	0,092	0,0794	0,0978	0,0978	0,0978	0,0715	0,0668	0,0794	0,0794	0,0794	0,0794	0,0794	0,0794
23	0,152	0,116	0,112	0,112	0,112	0,136	0,152	0,152	0,152	0,1256	0,1161	0,1103	0,1523	0,1106	0,1159	0,1159	0,1160	0,1409	0,1523	0,1523	0,1523	0,1283	0,1132	0,0943
24	0,06	0,06	0,048	0,048	0,036	0,032	0,06	0,06	0,06	0,0454	0,0421	0,0401	0,0699	0,0534	0,0355	0,0355	0,0379	0,0352	0,0699	0,0699	0,0699	0,0411	0,0418	0,0349
25	0,112	0,096	0,096	0,096	0,088	0,084	0,112	0,112	0,112	0,112	0,112	0,112	0,1127	0,1024	0,1024	0,1024	0,0879	0,0797	0,1127	0,1127	0,1127	0,1127	0,1127	0,1127
26	0,06	0,044	0,044	0,044	0,04	0,044	0,084	0,084	0,084	0,0506	0,048	0,0464	0,0644	0,0416	0,0416	0,0416	0,0275	0,0425	0,0806	0,0806	0,0806	0,0568	0,0543	0,0531
27	0,084	0,104	0,116	0,116	0,088	0,1	0,104	0,104	0,104	0,0989	0,0983	0,0980	0,0920	0,1015	0,1094	0,1094	0,0848	0,0979	0,1016	0,1016	0,1016	0,1050	0,1047	0,1048
28	0,08	0,1	0,096	0,096	0,068	0,08	0,08	0,08	0,08	0,0903	0,0851	0,0818	0,0801	0,1016	0,0920	0,0920	0,0701	0,0848	0,0801	0,0801	0,0801	0,0767	0,0957	0,0886
29	0,08	0,088	0,088	0,088	0,072	0,08	0,08	0,08	0,08	0,0773	0,072	0,0688	0,0861	0,0799	0,0799	0,0799	0,0706	0,0802	0,0861	0,0861	0,0861	0,0636	0,0825	0,0756
30	0,092	0,072	0,068	0,068	0,072	0,08	0,092	0,092	0,092	0,0935	0,0943	0,0948	0,0830	0,0666	0,0595	0,0595	0,0627	0,0729	0,0830	0,0830	0,0830	0,1000	0,0908	0,1071

Tabela A.15 – Erros de Classificação da Base (3) para 3.000 registros.

Fold	Base 3																							
	3.000 registros																							
	Regressão Linear												Regressão Logística											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,06	0,0626	0,056	0,056	0,0613	0,0613	0,06	0,0626	0,0626	0,06	0,06	0,06	0,0626	0,0586	0,06	0,06	0,06	0,0586	0,06	0,06	0,06	0,0626	0,0626	0,0626
2	0,068	0,072	0,072	0,072	0,0666	0,064	0,0706	0,0706	0,0706	0,0750	0,0557	0,0537	0,0653	0,0626	0,064	0,064	0,064	0,064	0,064	0,064	0,064	0,0572	0,0564	0,0543
3	0,068	0,068	0,068	0,068	0,0693	0,0693	0,068	0,068	0,068	0,068	0,068	0,068	0,0626	0,0626	0,0626	0,0626	0,0613	0,0613	0,0626	0,0626	0,0626	0,0626	0,0626	0,0626
4	0,0586	0,06	0,0666	0,0666	0,06	0,06	0,06	0,06	0,06	0,0598	0,0759	0,0777	0,0493	0,052	0,052	0,052	0,0506	0,0506	0,0493	0,0493	0,0493	0,0561	0,0561	0,0561
5	0,0653	0,064	0,0653	0,0653	0,0666	0,0666	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,0573	0,0573	0,0586	0,0586	0,056	0,056	0,0626	0,0626	0,0626	0,0687	0,0687	0,0687
6	0,08	0,0826	0,0773	0,0773	0,072	0,0746	0,08	0,08	0,08	0,08	0,08	0,08	0,0693	0,0813	0,076	0,076	0,068	0,068	0,0693	0,0693	0,0693	0,0693	0,0693	0,0693
7	0,1053	0,0613	0,068	0,068	0,116	0,108	0,0693	0,068	0,068	0,0739	0,0916	0,0885	0,0626	0,0666	0,0613	0,0613	0,0626	0,0613	0,0626	0,0626	0,0626	0,0626	0,0626	0,0626
8	0,064	0,064	0,064	0,064	0,0613	0,06	0,0653	0,0706	0,0706	0,0909	0,0834	0,0827	0,064	0,064	0,064	0,064	0,064	0,064	0,064	0,064	0,064	0,064	0,064	0,064
9	0,0906	0,064	0,072	0,072	0,0853	0,0853	0,068	0,0706	0,0706	0,0744	0,0850	0,0884	0,0666	0,0693	0,0693	0,0693	0,0653	0,0653	0,0706	0,0706	0,0706	0,0686	0,0686	0,0686
10	0,0746	0,0666	0,0613	0,0613	0,076	0,0773	0,0613	0,0626	0,0626	0,0764	0,0892	0,0864	0,068	0,0613	0,056	0,056	0,064	0,064	0,0626	0,0613	0,0613	0,0703	0,0703	0,0703
11	0,064	0,0506	0,0546	0,0546	0,0653	0,064	0,06	0,0573	0,0573	0,064	0,064	0,064	0,0546	0,0533	0,0546	0,0546	0,052	0,0533	0,0546	0,0546	0,0546	0,0546	0,0546	0,0546
12	0,1	0,0546	0,0546	0,0546	0,1	0,1	0,0546	0,056	0,056	0,0821	0,0594	0,0596	0,052	0,0533	0,0546	0,0546	0,052	0,0533	0,052	0,052	0,052	0,0528	0,0528	0,0528
13	0,0653	0,0733	0,0733	0,0733	0,068	0,068	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,0666	0,0706	0,068	0,068	0,0693	0,0693	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666
14	0,0813	0,0773	0,076	0,076	0,076	0,0773	0,0706	0,0653	0,0653	0,0805	0,0680	0,0653	0,068	0,0666	0,0666	0,0666	0,0693	0,068	0,068	0,068	0,068	0,068	0,068	0,068
15	0,0746	0,0733	0,0786	0,0786	0,0706	0,0693	0,06	0,0666	0,0666	0,0746	0,0746	0,0746	0,0573	0,06	0,0653	0,0653	0,0533	0,0546	0,0573	0,0573	0,0573	0,0573	0,0573	0,0573
16	0,0573	0,0573	0,0573	0,0573	0,0586	0,0573	0,0546	0,0626	0,0626	0,0573	0,0573	0,0573	0,056	0,056	0,0573	0,0573	0,0573	0,0573	0,056	0,056	0,056	0,0523	0,0523	0,0523
17	0,0706	0,068	0,0733	0,0733	0,072	0,0693	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706	0,0693	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706
18	0,0746	0,0693	0,0733	0,0733	0,0733	0,0746	0,068	0,0706	0,0706	0,0746	0,0746	0,0746	0,0626	0,0626	0,068	0,068	0,0586	0,0586	0,0626	0,0626	0,0626	0,0626	0,0626	0,0626
19	0,0653	0,0626	0,0626	0,0626	0,068	0,0653	0,0573	0,064	0,064	0,0731	0,0761	0,0763	0,0533	0,0506	0,0546	0,0546	0,0533	0,0533	0,052	0,052	0,052	0,0533	0,0533	0,0533
20	0,0866	0,0693	0,0746	0,0746	0,0853	0,088	0,0666	0,0853	0,0853	0,0778	0,0933	0,0935	0,0653	0,0653	0,068	0,068	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653
21	0,06	0,052	0,0626	0,0626	0,0573	0,056	0,06	0,06	0,06	0,0603	0,0753	0,0753	0,0586	0,06	0,0653	0,0653	0,0573	0,0586	0,0586	0,0586	0,0586	0,0586	0,0586	0,0586
22	0,0746	0,0746	0,0746	0,0746	0,068	0,0693	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746	0,0786	0,0746	0,0746	0,076	0,076	0,08	0,08	0,08	0,0746	0,0746	0,0746
23	0,06	0,068	0,068	0,068	0,0613	0,0613	0,056	0,0586	0,0586	0,0862	0,0725	0,0715	0,0666	0,0693	0,0693	0,0693	0,06	0,0626	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666
24	0,064	0,064	0,0733	0,0733	0,064	0,0626	0,0666	0,068	0,068	0,064	0,064	0,064	0,0693	0,0666	0,068	0,068	0,0706	0,0693	0,064	0,064	0,064	0,0693	0,0693	0,0693
25	0,076	0,0666	0,072	0,072	0,0693	0,0706	0,0706	0,072	0,072	0,0922	0,0785	0,0808	0,0693	0,0706	0,0693	0,0693	0,0706	0,0693	0,072	0,0706	0,0706	0,0740	0,0757	0,0785
26	0,08	0,0653	0,0666	0,0666	0,076	0,0746	0,0626	0,064	0,064	0,0935	0,0746	0,0754	0,0666	0,0666	0,0666	0,0666	0,0653	0,0653	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666
27	0,0866	0,0893	0,0853	0,0853	0,0773	0,0773	0,0773	0,0813	0,0813	0,0698	0,0803	0,0800	0,0866	0,0853	0,084	0,084	0,0813	0,0826	0,0866	0,0866	0,0866	0,0870	0,0870	0,0870
28	0,0506	0,052	0,0533	0,0533	0,0493	0,0506	0,0506	0,0506	0,0506	0,0869	0,0585	0,0582	0,0493	0,0506	0,052	0,052	0,048	0,048	0,0466	0,052	0,052	0,0517	0,0517	0,0517
29	0,064	0,0653	0,0653	0,0653	0,0626	0,06	0,064	0,064	0,064	0,0848	0,0780	0,0751	0,0613	0,0653	0,0653	0,0653	0,0613	0,0613	0,0666	0,0706	0,0706	0,0613	0,0613	0,0613
30	0,068	0,0733	0,0773	0,0773	0,0693	0,0693	0,0613	0,068	0,068	0,068	0,068	0,068	0,0613	0,068	0,068	0,068	0,0573	0,0586	0,0613	0,0613	0,0613	0,0613	0,0613	0,0613

Fold	Base 3																							
	3.000 registros																							
	Redes Neurais, 3 Neurônios, BPROP												Redes Neurais, 3 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,052	0,052	0,052	0,052	0,0626	0,0466	0,052	0,052	0,052	0,0618	0,0655	0,0677	0,0533	0,0533	0,0533	0,0533	0,0546	0,0506	0,0533	0,0533	0,0533	0,0618	0,0655	0,0677
2	0,0706	0,076	0,076	0,076	0,0666	0,068	0,068	0,068	0,068	0,0747	0,0747	0,0747	0,076	0,0693	0,0693	0,0693	0,0626	0,0746	0,076	0,076	0,076	0,0729	0,0729	0,0729
3	0,0693	0,072	0,072	0,072	0,064	0,0626	0,0746	0,0746	0,0746	0,0693	0,0693	0,0693	0,08	0,0773	0,0773	0,0773	0,0666	0,0666	0,0813	0,0813	0,0813	0,0711	0,0678	0,0658
4	0,056	0,056	0,056	0,056	0,0493	0,0573	0,056	0,056	0,056	0,056	0,056	0,056	0,064	0,0613	0,0746	0,0746	0,0586	0,0613	0,0733	0,0733	0,0733	0,064	0,064	0,064
5	0,0586	0,0546	0,064	0,064	0,056	0,0613	0,0586	0,0586	0,0586	0,0586	0,0586	0,0586	0,0666	0,0666	0,0666	0,0666	0,0573	0,0706	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666
6	0,0786	0,072	0,072	0,072	0,0706	0,0706	0,0786	0,0786	0,0786	0,0786	0,0786	0,0786	0,0733	0,0733	0,0733	0,0733	0,072	0,0693	0,0933	0,0933	0,0933	0,0843	0,0872	0,0889
7	0,0666	0,0626	0,064	0,064	0,0626	0,0626	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666	0,0653	0,068	0,068	0,068	0,0573	0,0666	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653
8	0,0613	0,064	0,064	0,064	0,064	0,0653	0,0613	0,0613	0,0613	0,0647	0,0685	0,0708	0,0666	0,0666	0,0666	0,0666	0,0653	0,0626	0,0666	0,0666	0,0666	0,0710	0,0739	0,0757
9	0,0746	0,076	0,0786	0,0786	0,0746	0,0693	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746	0,076	0,0666	0,0653	0,0653	0,072	0,0693	0,076	0,076	0,076	0,0739	0,0749	0,0755
10	0,0746	0,0666	0,0666	0,0666	0,0693	0,0706	0,0746	0,0746	0,0746	0,0712	0,0712	0,0712	0,0746	0,0706	0,0706	0,0706	0,0666	0,068	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746
11	0,06	0,06	0,06	0,06	0,0573	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,0573	0,0613	0,0613	0,0586	0,0586	0,06	0,06	0,06	0,0536	0,0536	0,0536
12	0,0586	0,0666	0,0666	0,0666	0,0533	0,0546	0,068	0,068	0,068	0,0586	0,0586	0,0586	0,06	0,072	0,072	0,072	0,0586	0,0573	0,06	0,06	0,06	0,06	0,06	0,06
13	0,0746	0,0693	0,0693	0,0693	0,072	0,0706	0,0746	0,0746	0,0746	0,0815	0,0830	0,0840	0,0706	0,0866	0,0866	0,0866	0,072	0,0653	0,0706	0,0706	0,0706	0,0759	0,0759	0,0759
14	0,0773	0,0773	0,0773	0,0773	0,0706	0,0733	0,0773	0,0773	0,0773	0,0773	0,0773	0,0773	0,0786	0,0746	0,0746	0,0746	0,0706	0,0693	0,0786	0,0786	0,0786	0,0786	0,0786	0,0786
15	0,06	0,0613	0,068	0,068	0,056	0,056	0,06	0,06	0,06	0,06	0,06	0,06	0,0626	0,0626	0,0626	0,0626	0,06	0,0546	0,0626	0,0626	0,0626	0,0626	0,0626	0,0626
16	0,06	0,0626	0,0706	0,0706	0,0573	0,0586	0,06	0,06	0,06	0,0631	0,0620	0,0614	0,06	0,06	0,06	0,06	0,0573	0,0613	0,06	0,06	0,06	0,06	0,06	0,06
17	0,0666	0,0746	0,0746	0,0746	0,0626	0,068	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666	0,064	0,0813	0,0813	0,0813	0,0653	0,0706	0,064	0,064	0,064	0,0692	0,0706	0,0714
18	0,064	0,064	0,064	0,064	0,0626	0,0706	0,064	0,064	0,064	0,064	0,064	0,064	0,068	0,0613	0,0706	0,0706	0,064	0,0626	0,068	0,068	0,068	0,068	0,068	0,068
19	0,0506	0,0506	0,0506	0,0506	0,0546	0,052	0,052	0,052	0,052	0,0506	0,0506	0,0506	0,0626	0,0573	0,0573	0,0573	0,0546	0,0626	0,0853	0,0853	0,0853	0,0685	0,0694	0,0699
20	0,068	0,0773	0,0773	0,0773	0,068	0,0733	0,068	0,068	0,068	0,068	0,068	0,068	0,072	0,072	0,072	0,072	0,0693	0,072	0,072	0,072	0,072	0,072	0,072	0,072
21	0,0626	0,0653	0,0733	0,0733	0,0613	0,0613	0,0653	0,0653	0,0653	0,0628	0,0628	0,0628	0,064	0,0733	0,076	0,076	0,0613	0,0613	0,064	0,064	0,064	0,0692	0,0705	0,0713
22	0,0866	0,0866	0,0866	0,0866	0,0773	0,076	0,0866	0,0866	0,0866	0,0866	0,0866	0,0866	0,088	0,0866	0,0866	0,0866	0,0733	0,0746	0,088	0,088	0,088	0,088	0,088	0,088
23	0,0653	0,0653	0,0653	0,0653	0,0586	0,064	0,0653	0,0653	0,0653	0,0714	0,0732	0,0743	0,0653	0,0653	0,0653	0,0653	0,0626	0,064	0,0653	0,0653	0,0653	0,0750	0,0732	0,0720
24	0,0653	0,0653	0,0653	0,0653	0,064	0,0613	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,0746	0,0746	0,0746	0,0746	0,0653	0,0613	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746
25	0,072	0,0706	0,0773	0,0773	0,0706	0,0706	0,072	0,072	0,072	0,0758	0,0771	0,0780	0,0706	0,0693	0,0693	0,0693	0,068	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706
26	0,068	0,0653	0,0653	0,0653	0,068	0,0706	0,0693	0,0693	0,0693	0,0769	0,0773	0,0776	0,0733	0,0733	0,0733	0,0733	0,068	0,0693	0,0733	0,0733	0,0733	0,0733	0,0733	0,0733
27	0,0746	0,0813	0,0813	0,0813	0,0853	0,0813	0,0893	0,0893	0,0893	0,0751	0,0768	0,0778	0,08	0,0813	0,0813	0,0813	0,0866	0,0853	0,0813	0,0813	0,0813	0,0815	0,0817	0,0817
28	0,0586	0,0613	0,0613	0,0613	0,0493	0,0493	0,0586	0,0586	0,0586	0,0586	0,0586	0,0586	0,064	0,06	0,06	0,06	0,052	0,0546	0,064	0,064	0,064	0,064	0,064	0,064
29	0,0573	0,064	0,0706	0,0706	0,0586	0,06	0,0653	0,0653	0,0653	0,0600	0,0600	0,0601	0,0613	0,0626	0,072	0,072	0,0626	0,0613	0,0613	0,0613	0,0613	0,0664	0,0669	0,0673
30	0,068	0,0666	0,076	0,076	0,0653	0,0733	0,068	0,068	0,068	0,068	0,068	0,068	0,084	0,088	0,0906	0,0906	0,0626	0,0746	0,084	0,084	0,084	0,084	0,084	0,084

Tabela A.16 – Erros de Classificação da Base (3) para 3.000 registros (continuação).

Fold	Base 3																							
	3.000 registros																							
	Redes Neurais, 10 Neurônios, BPROP												Redes Neurais, 10 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0706	0,068	0,0866	0,0866	0,068	0,0653	0,0706	0,0706	0,0706	0,0663	0,0689	0,0704	0,0764	0,0699	0,0847	0,0847	0,0733	0,0946	0,0764	0,0764	0,0764	0,0834	0,0790	0,0764
2	0,0786	0,0786	0,0786	0,0786	0,072	0,0693	0,0786	0,0786	0,0786	0,0646	0,0593	0,0560	0,0854	0,0854	0,0854	0,0854	0,072	0,0813	0,0854	0,0854	0,0854	0,0701	0,0635	0,0594
3	0,0666	0,0613	0,0613	0,0613	0,0613	0,0653	0,0666	0,0666	0,0666	0,0701	0,0678	0,0664	0,0678	0,0657	0,0657	0,0657	0,0733	0,0826	0,0893	0,0893	0,0893	0,0802	0,0747	0,0714
4	0,06	0,0613	0,076	0,076	0,0573	0,056	0,06	0,06	0,06	0,0597	0,0615	0,0625	0,0605	0,0408	0,0746	0,0746	0,0626	0,0693	0,0605	0,0605	0,0605	0,0711	0,0667	0,0641
5	0,0653	0,068	0,068	0,068	0,064	0,0573	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,0589	0,0666	0,0666	0,0666	0,0706	0,0866	0,0589	0,0589	0,0589	0,0828	0,0802	0,0787
6	0,0813	0,0693	0,0773	0,0773	0,068	0,0693	0,0813	0,0813	0,0813	0,0843	0,0872	0,0889	0,0833	0,0734	0,0706	0,0706	0,0826	0,0906	0,084	0,084	0,084	0,0968	0,0938	0,0921
7	0,0613	0,0613	0,0613	0,0613	0,0613	0,06	0,0613	0,0613	0,0613	0,0613	0,0613	0,0613	0,0502	0,0502	0,0502	0,0502	0,068	0,092	0,1066	0,1066	0,1066	0,0949	0,0957	0,0962
8	0,068	0,068	0,068	0,068	0,0666	0,064	0,068	0,068	0,068	0,0710	0,0710	0,0710	0,0646	0,0646	0,0646	0,0646	0,0773	0,0746	0,0893	0,0893	0,0893	0,0845	0,0841	0,0838
9	0,0746	0,08	0,08	0,08	0,0773	0,076	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746	0,0657	0,0848	0,0848	0,0848	0,0746	0,0973	0,0813	0,0813	0,0813	0,0944	0,0963	0,0975
10	0,0666	0,072	0,072	0,072	0,072	0,0666	0,0706	0,076	0,076	0,0666	0,0666	0,0666	0,0604	0,0795	0,0795	0,0795	0,0693	0,0906	0,0693	0,0693	0,0693	0,0811	0,0810	0,0810
11	0,06	0,0626	0,0626	0,0626	0,0586	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,0586	0,0601	0,0601	0,0601	0,052	0,0866	0,0586	0,0586	0,0586	0,0709	0,0662	0,0633
12	0,0653	0,072	0,072	0,072	0,068	0,0546	0,0653	0,0653	0,0653	0,0655	0,0650	0,0646	0,0741	0,0699	0,0699	0,0699	0,0653	0,084	0,092	0,0866	0,0866	0,0710	0,0670	0,0646
13	0,0706	0,0706	0,0706	0,0706	0,0733	0,072	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706	0,0727	0,0727	0,0727	0,0727	0,0706	0,0733	0,0727	0,0727	0,0727	0,0769	0,0759	0,0754
14	0,0773	0,0733	0,0733	0,0733	0,068	0,0693	0,0773	0,0773	0,0773	0,0773	0,0773	0,0773	0,0816	0,0782	0,0782	0,0782	0,0746	0,08	0,1093	0,1093	0,1093	0,0956	0,0853	0,0789
15	0,0666	0,0813	0,0813	0,0813	0,0666	0,068	0,0666	0,0666	0,0666	0,0752	0,0752	0,0752	0,0598	0,0809	0,0809	0,0809	0,0706	0,0946	0,0598	0,0598	0,0598	0,0807	0,0790	0,0779
16	0,0613	0,0613	0,0613	0,0613	0,0626	0,0613	0,0613	0,0613	0,0613	0,0613	0,0613	0,0613	0,0491	0,0491	0,0491	0,0491	0,0626	0,0946	0,0491	0,0491	0,0491	0,0830	0,0763	0,0723
17	0,0813	0,0693	0,0693	0,0693	0,068	0,064	0,0813	0,0813	0,0813	0,0802	0,0824	0,0837	0,0831	0,0711	0,0711	0,0711	0,0786	0,096	0,0973	0,0973	0,0973	0,0811	0,0810	0,0809
18	0,072	0,0693	0,0693	0,0693	0,0653	0,064	0,072	0,072	0,072	0,072	0,072	0,072	0,0771	0,0748	0,0748	0,0748	0,0746	0,0733	0,0933	0,0933	0,0933	0,0921	0,0915	0,0911
19	0,0546	0,0546	0,0546	0,0546	0,0546	0,0546	0,0546	0,0546	0,0546	0,0546	0,0546	0,0546	0,0611	0,0611	0,0611	0,0611	0,064	0,0613	0,0906	0,0906	0,0906	0,0827	0,0794	0,0774
20	0,0746	0,0746	0,0746	0,0746	0,068	0,0666	0,0746	0,0746	0,0746	0,0820	0,0844	0,0859	0,0807	0,0807	0,0807	0,0807	0,076	0,0933	0,0807	0,0807	0,0807	0,0964	0,0953	0,0946
21	0,064	0,0706	0,0826	0,0826	0,0626	0,0573	0,0626	0,0626	0,0626	0,0728	0,0760	0,0780	0,0545	0,0719	0,0855	0,0855	0,0626	0,0706	0,0853	0,0853	0,0853	0,0783	0,0753	0,0735
22	0,0906	0,0906	0,0906	0,0906	0,0786	0,0786	0,0906	0,0906	0,0906	0,0906	0,0906	0,0906	0,0899	0,0899	0,0899	0,0899	0,0706	0,088	0,0946	0,1053	0,1053	0,0921	0,0901	0,0889
23	0,0693	0,0693	0,0693	0,0693	0,064	0,0653	0,0693	0,0693	0,0693	0,0693	0,0693	0,0693	0,0602	0,0602	0,0602	0,0602	0,0626	0,0746	0,0946	0,0946	0,0946	0,0888	0,0836	0,0805
24	0,0733	0,0853	0,0853	0,0853	0,0653	0,0666	0,0733	0,0733	0,0733	0,0744	0,0750	0,0754	0,0790	0,0822	0,0932	0,0932	0,0653	0,0813	0,0826	0,0826	0,0826	0,0908	0,0881	0,0865
25	0,0706	0,0706	0,0706	0,0706	0,0666	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706	0,0673	0,0673	0,0673	0,0673	0,0666	0,0933	0,0673	0,0673	0,0673	0,0859	0,0841	0,0830
26	0,0693	0,0693	0,0693	0,0693	0,072	0,0693	0,0693	0,0693	0,0693	0,0693	0,0693	0,0693	0,0743	0,0743	0,0743	0,0743	0,0746	0,0853	0,0743	0,0743	0,0743	0,0941	0,0890	0,0859
27	0,0773	0,0773	0,0773	0,0773	0,084	0,0866	0,0773	0,0773	0,0773	0,0779	0,0775	0,0772	0,0847	0,0847	0,0847	0,0847	0,0893	0,0933	0,0847	0,0847	0,0847	0,0852	0,0851	0,0851
28	0,0466	0,0466	0,0466	0,0466	0,052	0,0466	0,0466	0,0466	0,0466	0,0466	0,0466	0,0466	0,0520	0,0520	0,0520	0,0520	0,068	0,0946	0,0520	0,0520	0,0520	0,0690	0,0612	0,0565
29	0,0613	0,0613	0,0613	0,0613	0,0666	0,0586	0,0613	0,0613	0,0613	0,0613	0,0613	0,0613	0,0612	0,0612	0,0612	0,0612	0,0693	0,076	0,0612	0,0612	0,0612	0,0809	0,0780	0,0762
30	0,0746	0,0746	0,0746	0,0746	0,0706	0,0733	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746	0,0823	0,0823	0,0823	0,0823	0,072	0,108	0,0823	0,0823	0,0823	0,0795	0,0733	0,0696
Fold	**Base 3**																							
	3.000 registros																							
	Redes Neurais, 20 Neurônios, BPROP												**Redes Neurais, 20 Neurônios, LEVMAR**											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0719	0,0701	0,0756	0,0756	0,074	0,0706	0,0719	0,0719	0,0719	0,0681	0,0675	0,0672	0,0839	0,0706	0,0492	0,0492	0,0718	0,5182	0,0839	0,0839	0,0839	0,075	0,0755	0,0738
2	0,0769	0,0769	0,0769	0,0769	0,074	0,0720	0,0769	0,0769	0,0769	0,0683	0,0621	0,0583	0,0703	0,0703	0,0703	0,0703	0,0842	0,5178	0,0703	0,0703	0,0703	0,0713	0,0597	0,0474
3	0,0617	0,0617	0,0617	0,0617	0,074	0,0574	0,0617	0,0617	0,0617	0,0729	0,0699	0,0681	0,0815	0,0812	0,0812	0,0812	0,0751	0,5146	0,0815	0,0815	0,0815	0,0759	0,0675	0,0572
4	0,0530	0,0691	0,0747	0,0747	0,074	0,0637	0,0530	0,0530	0,0530	0,0658	0,0674	0,0683	0,0745	0,0966	0,0942	0,0942	0,0742	0,5213	0,0745	0,0745	0,0745	0,069	0,061	0,0583
5	0,0610	0,0689	0,0689	0,0689	0,074	0,0775	0,0610	0,0610	0,0610	0,0696	0,0703	0,0708	0,058	0,0808	0,0808	0,0808	0,078	0,5186	0,058	0,058	0,058	0,06	0,0772	0,073
6	0,0744	0,0768	0,0781	0,0781	0,074	0,0534	0,0744	0,0744	0,0744	0,0744	0,0744	0,0744	0,0852	0,0869	0,0655	0,0655	0,0766	0,5177	0,0852	0,0852	0,0852	0,0852	0,0852	0,0852
7	0,0634	0,0634	0,0634	0,0634	0,074	0,0638	0,0634	0,0634	0,0634	0,0949	0,0950	0,0951	0,0763	0,0763	0,0763	0,0763	0,0671	0,5088	0,0763	0,0763	0,0763	0,0925	0,0985	0,0929
8	0,0682	0,0682	0,0682	0,0682	0,074	0,0609	0,0682	0,0682	0,0682	0,0719	0,0739	0,0751	0,0734	0,0734	0,0734	0,0734	0,0709	0,5166	0,0734	0,0734	0,0734	0,0719	0,0736	0,0743
9	0,0787	0,0719	0,0719	0,0719	0,074	0,0560	0,0787	0,0787	0,0787	0,0787	0,0787	0,0787	0,0873	0,076	0,076	0,076	0,0841	0,5095	0,0873	0,0873	0,0873	0,0873	0,0873	0,0873
10	0,0768	0,0747	0,0747	0,0747	0,074	0,0663	0,0768	0,0768	0,0768	0,0768	0,0768	0,0768	0,0881	0,0632	0,0632	0,0632	0,074	0,5198	0,0881	0,0881	0,0881	0,0881	0,0881	0,0881
11	0,0660	0,0563	0,0563	0,0563	0,074	0,0605	0,0660	0,0660	0,0660	0,0618	0,06	0,0588	0,0739	0,0474	0,0474	0,0474	0,0757	0,5164	0,0739	0,0739	0,0739	0,0648	0,0494	0,0598
12	0,0628	0,0752	0,0752	0,0752	0,074	0,0596	0,0628	0,0628	0,0628	0,0692	0,0663	0,0646	0,0629	0,0726	0,0726	0,0726	0,0796	0,5185	0,0629	0,0629	0,0629	0,0689	0,0659	0,0682
13	0,0706	0,0706	0,0706	0,0706	0,074	0,0427	0,0706	0,0706	0,0706	0,0778	0,0781	0,0782	0,0572	0,0572	0,0572	0,0572	0,0668	0,5128	0,0572	0,0572	0,0572	0,0775	0,0776	0,0818
14	0,0718	0,0681	0,0681	0,0681	0,074	0,0504	0,0718	0,0718	0,0718	0,0731	0,0680	0,0647	0,0538	0,0514	0,0514	0,0514	0,0668	0,5065	0,0538	0,0538	0,0538	0,077	0,0659	0,0677
15	0,0721	0,0864	0,0864	0,0864	0,074	0,0730	0,0721	0,0721	0,0721	0,0721	0,0721	0,0721	0,0933	0,0967	0,0967	0,0967	0,0712	0,5118	0,0933	0,0933	0,0933	0,0933	0,0933	0,0933
16	0,0539	0,0539	0,0539	0,0539	0,074	0,0608	0,0539	0,0539	0,0539	0,0613	0,0600	0,0592	0,0548	0,0548	0,0548	0,0548	0,0728	0,5119	0,0548	0,0548	0,0548	0,07	0,0549	0,0616
17	0,0802	0,0583	0,0583	0,0583	0,074	0,0758	0,0802	0,0802	0,0802	0,0756	0,0761	0,0764	0,0791	0,0799	0,0799	0,0799	0,0694	0,5098	0,0791	0,0791	0,0791	0,0717	0,0761	0,0815
18	0,0686	0,0678	0,0678	0,0678	0,074	0,0651	0,0686	0,0686	0,0686	0,0686	0,0686	0,0686	0,0996	0,0892	0,0892	0,0892	0,0853	0,5243	0,0996	0,0996	0,0996	0,0996	0,0996	0,0996
19	0,0587	0,0587	0,0587	0,0587	0,074	0,0623	0,0587	0,0587	0,0587	0,0596	0,0620	0,0635	0,0608	0,0608	0,0608	0,0608	0,0624	0,5151	0,0608	0,0608	0,0608	0,0519	0,0559	0,0589
20	0,0825	0,0825	0,0825	0,0825	0,074	0,0754	0,0825	0,0825	0,0825	0,0829	0,0851	0,0864	0,0726	0,0726	0,0726	0,0726	0,0775	0,5205	0,0726	0,0726	0,0726	0,086	0,0853	0,0822
21	0,0625	0,0771	0,0867	0,0867	0,074	0,0700	0,0625	0,0625	0,0625	0,0792	0,0795	0,0797	0,0563	0,0729	0,069	0,0808	0,0728	0,5039	0,0563	0,0563	0,0563	0,0707	0,0664	0,078
22	0,0831	0,0831	0,0831	0,0831	0,074	0,0719	0,0831	0,0831	0,0831	0,0831	0,0831	0,0831	0,0957	0,0957	0,0957	0,0957	0,0871	0,5224	0,0957	0,0957	0,0957	0,0957	0,0957	0,0957
23	0,0701	0,0701	0,0701	0,0701	0,074	0,0578	0,0701	0,0701	0,0701	0,0695	0,0690	0,0686	0,0806	0,0806	0,0806	0,0806	0,0836	0,5138	0,0806	0,0806	0,0806	0,0565	0,0689	0,0719
24	0,0694	0,0914	0,0744	0,0744	0,074	0,0672	0,0694	0,0694	0,0694	0,0726	0,0723	0,0721	0,0477	0,0745	0,0733	0,0733	0,0663	0,5111	0,0477	0,0477	0,0477	0,0788	0,0651	0,0738
25	0,0639	0,0639	0,0639	0,0639	0,074	0,0693	0,0639	0,0639	0,0639	0,0639	0,0639	0,0639	0,0464	0,0464	0,0464	0,0464	0,0698	0,5089	0,0464	0,0464	0,0464	0,0464	0,0464	0,0464
26	0,0602	0,0602	0,0602	0,0602	0,074	0,0599	0,0602	0,0602	0,0602	0,0687	0,0698	0,0705	0,0624	0,0624	0,0624	0,0624	0,0768	0,5165	0,0624	0,0624	0,0624	0,055	0,0685	0,069
27	0,0791	0,0791	0,0791	0,0791	0,074	0,0746	0,0791	0,0791	0,0791	0,0806	0,0817	0,0823	0,089	0,089	0,089	0,089	0,0787	0,5273	0,089	0,089	0,089	0,0821	0,0823	0,0795
28	0,0386	0,0386	0,0386	0,0386	0,074	0,0684	0,0386	0,0386	0,0386	0,0626	0,0619	0,0615	0,0352	0,0352	0,0352	0,0352	0,0852	0,5119	0,0352	0,0352	0,0352	0,0641	0,0626	0,0588
29	0,0534	0,0534	0,0534	0,0534	0,074	0,0503	0,0534	0,0534	0,0534	0,0727	0,0718	0,0712	0,0369	0,0369	0,0369	0,0369	0,0733	0,523	0,0369	0,0369	0,0369	0,0762	0,0768	0,0694
30	0,0736	0,0736	0,0736	0,0736	0,074	0,0697	0,0736	0,0736	0,0736	0,0736	0,0736	0,0736	0,0836	0,0836	0,0836	0,0836	0,0707	0,523	0,0836	0,0836	0,0836	0,0836	0,0836	0,0836

Tabela A.17 – Erros de Classificação da Base (3) para 5.000 registros.

Fold	Base 3																							
	5.000 registros																							
	Regressão Linear												Regressão Logística											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0752	0,06	0,0592	0,0592	0,0752	0,0752	0,064	0,0656	0,0656	0,0705	0,0587	0,0587	0,0616	0,0608	0,0608	0,0608	0,0616	0,0616	0,0616	0,0616	0,0616	0,0572	0,0572	0,0572
2	0,06	0,0624	0,0696	0,0696	0,0624	0,0624	0,06	0,06	0,06	0,0799	0,0759	0,0761	0,0632	0,064	0,0648	0,0648	0,064	0,0632	0,0632	0,0632	0,0632	0,0632	0,0632	0,0632
3	0,0864	0,072	0,0784	0,0784	0,088	0,0896	0,072	0,0784	0,0784	0,0794	0,0938	0,0964	0,0736	0,072	0,068	0,068	0,072	0,0728	0,0736	0,0736	0,0736	0,0743	0,0743	0,0743
4	0,068	0,064	0,064	0,064	0,0688	0,0672	0,0712	0,0752	0,0752	0,0786	0,0765	0,0770	0,0664	0,0648	0,0648	0,0648	0,0656	0,0656	0,0664	0,0664	0,0664	0,0664	0,0664	0,0664
5	0,0624	0,0552	0,0624	0,0624	0,0664	0,0656	0,0656	0,0672	0,0672	0,0785	0,0720	0,0725	0,0552	0,0552	0,0552	0,0552	0,056	0,0552	0,0584	0,06	0,06	0,0552	0,0552	0,0552
6	0,0792	0,0592	0,0632	0,0632	0,084	0,0824	0,0656	0,0704	0,0704	0,0813	0,0670	0,0665	0,0608	0,0656	0,0608	0,0608	0,0616	0,0616	0,064	0,064	0,064	0,0613	0,0629	0,0631
7	0,068	0,068	0,068	0,068	0,072	0,0712	0,068	0,068	0,068	0,068	0,068	0,068	0,0688	0,0688	0,0688	0,0688	0,0688	0,0696	0,0688	0,0688	0,0688	0,0688	0,0688	0,0688
8	0,0664	0,06	0,056	0,056	0,064	0,0648	0,0664	0,0664	0,0664	0,0785	0,0722	0,0706	0,0576	0,0608	0,0624	0,0624	0,0608	0,06	0,0576	0,0576	0,0576	0,0576	0,0576	0,0576
9	0,096	0,0872	0,088	0,088	0,0912	0,0904	0,0904	0,0872	0,0872	0,0815	0,0922	0,0908	0,0824	0,0832	0,0848	0,0848	0,0832	0,0832	0,0848	0,0832	0,0832	0,0824	0,0824	0,0824
10	0,0736	0,0728	0,076	0,076	0,0744	0,072	0,0736	0,0736	0,0736	0,0762	0,0828	0,0824	0,0736	0,0728	0,0736	0,0736	0,0728	0,0736	0,0688	0,0704	0,0704	0,0721	0,0721	0,0721
11	0,0656	0,0568	0,0584	0,0584	0,0664	0,0664	0,0576	0,0576	0,0576	0,0848	0,0647	0,0619	0,0528	0,0536	0,0544	0,0544	0,0512	0,052	0,0528	0,0528	0,0528	0,0558	0,0558	0,0558
12	0,06	0,0584	0,0592	0,0592	0,0592	0,0592	0,0568	0,0608	0,0608	0,0789	0,0619	0,0615	0,0552	0,06	0,0576	0,0576	0,0552	0,056	0,0544	0,0544	0,0544	0,0555	0,0574	0,0574
13	0,0984	0,0792	0,0808	0,0808	0,0952	0,0976	0,0696	0,08	0,08	0,0725	0,0914	0,0938	0,0696	0,0752	0,0752	0,0752	0,068	0,0688	0,0728	0,0728	0,0728	0,0696	0,0696	0,0696
14	0,072	0,0648	0,068	0,068	0,068	0,0704	0,072	0,072	0,072	0,0891	0,0717	0,0700	0,0624	0,0632	0,064	0,064	0,0608	0,0608	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624
15	0,0808	0,0776	0,0768	0,0768	0,0832	0,0816	0,0768	0,0776	0,0776	0,0849	0,0834	0,0818	0,0792	0,0728	0,072	0,072	0,0776	0,0776	0,0792	0,0792	0,0792	0,0792	0,0792	0,0792
16	0,064	0,064	0,0616	0,0616	0,0616	0,0624	0,0584	0,0592	0,0592	0,0774	0,0623	0,0612	0,0584	0,0592	0,06	0,06	0,0576	0,0592	0,0584	0,0584	0,0584	0,0584	0,0584	0,0584
17	0,0648	0,0608	0,06	0,06	0,0616	0,0616	0,0592	0,06	0,06	0,0707	0,0673	0,0650	0,0592	0,0592	0,06	0,06	0,0584	0,06	0,0592	0,0576	0,0576	0,0592	0,0592	0,0592
18	0,072	0,072	0,072	0,072	0,0696	0,0696	0,072	0,072	0,072	0,0706	0,0734	0,0731	0,0616	0,0632	0,0632	0,0632	0,0632	0,0624	0,0616	0,0616	0,0616	0,0637	0,0637	0,0637
19	0,0696	0,0672	0,0656	0,0656	0,0704	0,0696	0,0632	0,0616	0,0616	0,0682	0,0678	0,0666	0,0664	0,0656	0,0656	0,0656	0,064	0,0624	0,0664	0,0664	0,0664	0,0665	0,0654	0,0647
20	0,0816	0,0768	0,0744	0,0744	0,0824	0,0832	0,064	0,0712	0,0712	0,0731	0,0810	0,0792	0,0632	0,064	0,0616	0,0616	0,0632	0,0648	0,0632	0,0632	0,0632	0,0632	0,0632	0,0632
21	0,0616	0,0552	0,0576	0,0576	0,06	0,0608	0,0568	0,0568	0,0568	0,0686	0,0722	0,0716	0,056	0,056	0,0528	0,0528	0,0544	0,0544	0,056	0,056	0,056	0,056	0,056	0,056
22	0,0752	0,0744	0,0736	0,0736	0,0752	0,076	0,0752	0,0752	0,0752	0,0722	0,0831	0,0824	0,072	0,072	0,0744	0,0744	0,072	0,072	0,072	0,072	0,072	0,0708	0,0708	0,0708
23	0,076	0,0624	0,0648	0,0648	0,0736	0,0736	0,0648	0,0704	0,0704	0,0867	0,0692	0,0692	0,0624	0,0624	0,0608	0,0608	0,0616	0,0616	0,0632	0,0632	0,0632	0,0682	0,0682	0,0682
24	0,076	0,068	0,0712	0,0712	0,0768	0,076	0,0728	0,0688	0,0688	0,0818	0,0748	0,0751	0,0648	0,064	0,068	0,068	0,0632	0,0656	0,0648	0,0648	0,0648	0,0648	0,0648	0,0648
25	0,0776	0,0616	0,0656	0,0656	0,0728	0,0736	0,0592	0,0744	0,0744	0,0819	0,0726	0,0721	0,06	0,0632	0,0632	0,0632	0,0624	0,0616	0,06	0,06	0,06	0,0647	0,0647	0,0647
26	0,0688	0,0528	0,0544	0,0544	0,0648	0,0656	0,0504	0,0536	0,0536	0,0844	0,0620	0,0629	0,0544	0,0536	0,056	0,056	0,0536	0,0536	0,0512	0,0536	0,0536	0,0572	0,0572	0,0572
27	0,0672	0,0664	0,0648	0,0648	0,068	0,0696	0,064	0,0712	0,0712	0,0734	0,0682	0,0685	0,0656	0,0656	0,0656	0,0656	0,0648	0,0648	0,0656	0,0656	0,0656	0,0656	0,0656	0,0656
28	0,08	0,0808	0,0824	0,0824	0,0792	0,0768	0,08	0,08	0,08	0,0775	0,0805	0,0837	0,0808	0,0784	0,0816	0,0816	0,0792	0,0792	0,0808	0,0808	0,0808	0,0808	0,0808	0,0808
29	0,0984	0,0792	0,0792	0,0792	0,0976	0,0992	0,0784	0,0744	0,0744	0,0718	0,0903	0,0893	0,0768	0,0752	0,076	0,076	0,0736	0,0744	0,0776	0,0792	0,0792	0,0768	0,0768	0,0768
30	0,0752	0,068	0,068	0,068	0,0728	0,072	0,0728	0,0736	0,0736	0,0821	0,0688	0,0678	0,0712	0,0712	0,0696	0,0696	0,072	0,072	0,0712	0,0712	0,0712	0,0712	0,0712	0,0712

Fold	Base 3																							
	5.000 registros																							
	Redes Neurais, 3 Neurônios, BPROP												Redes Neurais, 3 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,064	0,064	0,0648	0,0648	0,0592	0,0592	0,0648	0,0648	0,0648	0,0589	0,0589	0,0589	0,064	0,064	0,064	0,064	0,0616	0,0608	0,0664	0,0664	0,0664	0,0589	0,0554	0,0532
2	0,064	0,064	0,064	0,064	0,0632	0,0616	0,064	0,064	0,064	0,064	0,064	0,064	0,0624	0,0624	0,0624	0,0624	0,064	0,0648	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624
3	0,0728	0,0752	0,0752	0,0752	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728	0,0744	0,0816	0,0816	0,0816	0,0712	0,072	0,0744	0,0744	0,0744	0,0787	0,0787	0,0787
4	0,068	0,0672	0,0672	0,0672	0,068	0,0688	0,072	0,072	0,072	0,068	0,068	0,068	0,0664	0,0704	0,0784	0,0784	0,0696	0,068	0,0744	0,0744	0,0744	0,0736	0,0728	0,0724
5	0,0544	0,0552	0,0552	0,0552	0,056	0,056	0,0544	0,0544	0,0544	0,0544	0,0544	0,0544	0,0568	0,064	0,0664	0,0664	0,0552	0,0584	0,0568	0,0568	0,0568	0,0674	0,0692	0,0702
6	0,0632	0,072	0,072	0,072	0,064	0,064	0,0632	0,0632	0,0632	0,0657	0,0657	0,0657	0,0608	0,064	0,0672	0,0672	0,0632	0,0648	0,0608	0,0608	0,0608	0,0608	0,0608	0,0608
7	0,0736	0,0696	0,0744	0,0744	0,0696	0,0696	0,0744	0,0744	0,0744	0,0736	0,0736	0,0736	0,0664	0,0728	0,0728	0,0728	0,0712	0,0672	0,0768	0,0768	0,0768	0,0797	0,0833	0,0854
8	0,0632	0,0656	0,0656	0,0656	0,0624	0,064	0,0632	0,0632	0,0632	0,0632	0,0632	0,0632	0,0656	0,064	0,064	0,064	0,0632	0,0632	0,0656	0,0656	0,0656	0,0656	0,0656	0,0656
9	0,0888	0,0792	0,0792	0,0792	0,0848	0,0856	0,0888	0,0888	0,0888	0,0888	0,0888	0,0888	0,088	0,088	0,088	0,088	0,0864	0,0872	0,088	0,088	0,088	0,088	0,088	0,088
10	0,0736	0,0792	0,0728	0,0728	0,0712	0,072	0,0736	0,0736	0,0736	0,0736	0,0736	0,0736	0,0784	0,0728	0,08	0,08	0,0728	0,0744	0,0776	0,0784	0,0784	0,0784	0,0784	0,0784
11	0,0552	0,0576	0,0576	0,0576	0,052	0,0512	0,0584	0,0536	0,0536	0,0552	0,0552	0,0552	0,056	0,0608	0,0608	0,0608	0,0528	0,0504	0,056	0,056	0,056	0,056	0,056	0,056
12	0,0544	0,0544	0,0544	0,0544	0,056	0,056	0,0544	0,0544	0,0544	0,0544	0,0544	0,0544	0,0544	0,064	0,064	0,064	0,056	0,0544	0,0544	0,0544	0,0544	0,0544	0,0544	0,0544
13	0,0728	0,0728	0,0728	0,0728	0,0736	0,0744	0,0728	0,0728	0,0728	0,0772	0,0772	0,0772	0,0728	0,0744	0,0744	0,0744	0,0696	0,0752	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728
14	0,0608	0,0608	0,0608	0,0608	0,0632	0,064	0,0608	0,0608	0,0608	0,0608	0,0608	0,0608	0,0632	0,0632	0,0632	0,0632	0,0624	0,0648	0,0632	0,0632	0,0632	0,0632	0,0632	0,0632
15	0,0784	0,0784	0,0784	0,0784	0,08	0,08	0,0784	0,0784	0,0784	0,0784	0,0784	0,0784	0,0792	0,0768	0,0768	0,0768	0,0808	0,0776	0,084	0,084	0,084	0,0792	0,0792	0,0792
16	0,0592	0,0592	0,0632	0,0632	0,0576	0,0592	0,0592	0,0592	0,0592	0,0592	0,0592	0,0592	0,056	0,0616	0,0648	0,0648	0,0608	0,06	0,0688	0,0688	0,0688	0,056	0,056	0,056
17	0,0664	0,0576	0,0576	0,0576	0,0584	0,0576	0,0664	0,0664	0,0664	0,0689	0,0689	0,0689	0,064	0,0616	0,0616	0,0616	0,0584	0,0552	0,064	0,064	0,064	0,064	0,064	0,064
18	0,064	0,0632	0,0632	0,0632	0,06	0,0624	0,0664	0,0664	0,0664	0,064	0,064	0,064	0,064	0,064	0,064	0,064	0,0624	0,0648	0,064	0,064	0,064	0,064	0,064	0,064
19	0,0624	0,068	0,068	0,068	0,0616	0,0576	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624	0,0648	0,0632	0,0632	0,06	0,0608	0,0624	0,0624	0,0624	0,0644	0,0617	0,0602
20	0,0624	0,0688	0,0696	0,0696	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624	0,0632	0,072	0,072	0,072	0,0632	0,064	0,0648	0,0648	0,0648	0,0632	0,0632	0,0632
21	0,0528	0,0616	0,0528	0,0528	0,0568	0,0592	0,0528	0,0528	0,0528	0,0586	0,0586	0,0586	0,052	0,0568	0,0656	0,0656	0,0576	0,0584	0,052	0,052	0,052	0,0597	0,0597	0,0597
22	0,0728	0,0768	0,0808	0,0808	0,0696	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728	0,0768	0,0864	0,0864	0,0864	0,0704	0,0728	0,0808	0,0808	0,0808	0,0768	0,0768	0,0768
23	0,068	0,0624	0,0624	0,0624	0,0608	0,0656	0,068	0,068	0,068	0,0677	0,0679	0,0681	0,0672	0,0664	0,0664	0,0664	0,06	0,0656	0,0672	0,0672	0,0672	0,0687	0,0696	0,0701
24	0,0672	0,0696	0,0696	0,0696	0,0656	0,0656	0,0656	0,0656	0,0656	0,0706	0,0706	0,0706	0,0696	0,068	0,064	0,064	0,0696	0,0696	0,0704	0,0696	0,0696	0,0733	0,0752	0,0764
25	0,0608	0,0664	0,0712	0,0712	0,0592	0,0568	0,0656	0,0656	0,0656	0,0608	0,0608	0,0608	0,0616	0,0616	0,0616	0,0616	0,0624	0,0632	0,0616	0,0616	0,0616	0,0616	0,0616	0,0616
26	0,052	0,0552	0,0576	0,0576	0,0528	0,0536	0,052	0,052	0,052	0,052	0,052	0,052	0,0536	0,0536	0,0536	0,0536	0,052	0,0528	0,0536	0,0536	0,0536	0,0632	0,0653	0,0665
27	0,0656	0,0728	0,0728	0,0728	0,0664	0,068	0,0656	0,0656	0,0656	0,0656	0,0656	0,0656	0,0688	0,0744	0,0744	0,0744	0,068	0,0688	0,0688	0,0688	0,0688	0,0688	0,0688	0,0688
28	0,088	0,0856	0,088	0,088	0,08	0,0784	0,088	0,088	0,088	0,088	0,088	0,088	0,0872	0,0872	0,0872	0,0872	0,0792	0,0792	0,0872	0,0872	0,0872	0,0872	0,0872	0,0872
29	0,076	0,076	0,0768	0,0768	0,0728	0,0744	0,0792	0,0792	0,0792	0,0832	0,0870	0,0893	0,076	0,0768	0,084	0,084	0,0736	0,072	0,0776	0,0784	0,0784	0,0887	0,0887	0,0887
30	0,0728	0,0728	0,0728	0,0728	0,0728	0,0752	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728	0,0752	0,0752	0,0752	0,0736	0,0768	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728

Tabela A.18 – Erros de Classificação da Base (3) para 5.000 registros (continuação).

Fold	Base 3																							
	5.000 registros																							
	Redes Neurais, 10 Neurônios, BPROP												Redes Neurais, 10 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0624	0,0624	0,0624	0,0624	0,0584	0,06	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624	0,0584	0,0584	0,0584	0,0584	0,0525	0,0664	0,0816	0,0816	0,0816	0,0699	0,0642	0,0607
2	0,064	0,068	0,072	0,072	0,0624	0,0632	0,0656	0,0656	0,0656	0,064	0,064	0,064	0,0611	0,0749	0,0626	0,0626	0,0565	0,0928	0,0611	0,0611	0,0611	0,0896	0,0869	0,0852
3	0,0704	0,0728	0,084	0,084	0,0744	0,0736	0,0768	0,0824	0,0824	0,0704	0,0704	0,0704	0,0651	0,0687	0,0853	0,0853	0,0801	0,0912	0,0651	0,0651	0,0651	0,0956	0,0950	0,0947
4	0,0736	0,0736	0,0736	0,0736	0,0712	0,072	0,0736	0,0736	0,0736	0,0720	0,0728	0,0734	0,0704	0,0731	0,0731	0,0731	0,0688	0,0824	0,0704	0,0704	0,0704	0,0872	0,0835	0,0813
5	0,06	0,0608	0,0608	0,0608	0,0576	0,0584	0,06	0,06	0,06	0,06	0,06	0,06	0,0522	0,0625	0,0625	0,0625	0,0636	0,0664	0,0784	0,0784	0,0784	0,0793	0,0785	0,0781
6	0,0736	0,0672	0,076	0,076	0,064	0,06	0,072	0,072	0,072	0,0734	0,0729	0,0726	0,0791	0,0692	0,0763	0,0763	0,0700	0,0808	0,0791	0,0791	0,0791	0,0810	0,0783	0,0766
7	0,0744	0,0744	0,0744	0,0744	0,0696	0,072	0,0744	0,0744	0,0744	0,0744	0,0744	0,0744	0,0790	0,0790	0,0790	0,0790	0,0613	0,0936	0,0790	0,0790	0,0790	0,0862	0,0849	0,0841
8	0,0608	0,0608	0,0608	0,0608	0,0632	0,0656	0,0608	0,0608	0,0608	0,0650	0,0651	0,0651	0,0534	0,0534	0,0534	0,0534	0,0648	0,0704	0,0984	0,0984	0,0984	0,0738	0,0710	0,0692
9	0,0856	0,0856	0,0856	0,0856	0,0872	0,088	0,0888	0,0888	0,0888	0,0856	0,0856	0,0856	0,0879	0,0879	0,0879	0,0879	0,0794	0,1032	0,0888	0,0888	0,0888	0,0956	0,0950	0,0946
10	0,0736	0,0784	0,0736	0,0736	0,0728	0,0712	0,072	0,072	0,072	0,0759	0,0779	0,0792	0,0702	0,0736	0,0799	0,0799	0,0730	0,0808	0,0702	0,0702	0,0702	0,0975	0,0946	0,0928
11	0,052	0,06	0,06	0,06	0,0568	0,0576	0,052	0,052	0,052	0,052	0,052	0,052	0,0564	0,0567	0,0567	0,0567	0,0570	0,0752	0,104	0,104	0,104	0,0720	0,0680	0,0655
12	0,052	0,052	0,052	0,052	0,0544	0,0584	0,052	0,052	0,052	0,052	0,052	0,052	0,0525	0,0525	0,0525	0,0525	0,0634	0,0624	0,0848	0,0848	0,0848	0,0741	0,0706	0,0685
13	0,0744	0,0736	0,0736	0,0736	0,0712	0,0696	0,0744	0,0744	0,0744	0,0744	0,0744	0,0744	0,0827	0,0666	0,0666	0,0666	0,0688	0,092	0,0827	0,0827	0,0827	0,0978	0,0963	0,0954
14	0,0648	0,068	0,068	0,068	0,0608	0,0616	0,0648	0,0648	0,0648	0,0690	0,0700	0,0706	0,0629	0,0652	0,0652	0,0652	0,0566	0,0792	0,0872	0,0872	0,0872	0,0870	0,0816	0,0783
15	0,0776	0,0776	0,0776	0,0776	0,0792	0,0768	0,0776	0,0776	0,0776	0,0776	0,0776	0,0776	0,0758	0,0758	0,0758	0,0758	0,0837	0,084	0,0944	0,0992	0,0992	0,0904	0,0887	0,0878
16	0,06	0,072	0,072	0,072	0,0576	0,0576	0,0624	0,0624	0,0624	0,0604	0,0604	0,0604	0,0644	0,0703	0,0703	0,0703	0,0453	0,0712	0,0792	0,0712	0,0712	0,0775	0,0737	0,0713
17	0,0624	0,0632	0,0672	0,0672	0,0592	0,0592	0,064	0,0632	0,0632	0,0624	0,0624	0,0624	0,0519	0,0628	0,0625	0,0625	0,0550	0,0704	0,0519	0,0519	0,0519	0,0684	0,0648	0,0627
18	0,0608	0,0608	0,0608	0,0608	0,0632	0,0616	0,0608	0,0608	0,0608	0,0664	0,0677	0,0685	0,0610	0,0610	0,0610	0,0610	0,0635	0,068	0,0610	0,0610	0,0610	0,0859	0,0804	0,0770
19	0,0656	0,0664	0,0664	0,0664	0,0608	0,0624	0,0632	0,0632	0,0632	0,0670	0,0678	0,0683	0,0632	0,0690	0,0690	0,0690	0,0676	0,0664	0,0632	0,0632	0,0632	0,0719	0,0710	0,0705
20	0,064	0,0672	0,0672	0,0672	0,0664	0,0648	0,0648	0,0648	0,0648	0,064	0,064	0,064	0,0652	0,0603	0,0603	0,0603	0,0541	0,0688	0,0856	0,0832	0,0832	0,0878	0,0852	0,0836
21	0,0536	0,0536	0,0536	0,0536	0,056	0,0584	0,0536	0,0536	0,0536	0,0536	0,0536	0,0536	0,0552	0,0552	0,0552	0,0552	0,0624	0,0656	0,0552	0,0552	0,0552	0,0552	0,0552	0,0552
22	0,0712	0,0776	0,0776	0,0776	0,0712	0,0728	0,0712	0,0712	0,0712	0,0751	0,0751	0,0751	0,0678	0,0868	0,0868	0,0868	0,0709	0,084	0,0678	0,0678	0,0678	0,0678	0,0678	0,0678
23	0,0656	0,0728	0,0728	0,0728	0,0656	0,0656	0,0656	0,0656	0,0656	0,0677	0,0696	0,0707	0,0622	0,0739	0,0739	0,0739	0,0626	0,0752	0,0832	0,0936	0,0936	0,0622	0,0622	0,0622
24	0,0664	0,0664	0,0664	0,0664	0,0664	0,0664	0,0728	0,0728	0,0728	0,0664	0,0664	0,0664	0,0656	0,0656	0,0656	0,0656	0,0703	0,0704	0,0656	0,0656	0,0656	0,0656	0,0656	0,0656
25	0,0624	0,0624	0,0624	0,0624	0,064	0,064	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624	0,0544	0,0544	0,0544	0,0544	0,0692	0,08	0,0544	0,0544	0,0544	0,0544	0,0544	0,0544
26	0,0576	0,0592	0,0592	0,0592	0,052	0,052	0,0576	0,0576	0,0576	0,0686	0,0710	0,0724	0,0606	0,0640	0,0640	0,0640	0,0503	0,0632	0,0606	0,0606	0,0606	0,0606	0,0606	0,0606
27	0,0664	0,0664	0,0664	0,0664	0,0656	0,0648	0,0664	0,0664	0,0664	0,0664	0,0664	0,0664	0,0611	0,0611	0,0611	0,0611	0,0761	0,0808	0,0611	0,0611	0,0611	0,0611	0,0611	0,0611
28	0,0848	0,088	0,088	0,088	0,0824	0,0816	0,0864	0,0864	0,0864	0,0897	0,0910	0,0918	0,0807	0,0798	0,0798	0,0798	0,0786	0,0968	0,0807	0,0807	0,0807	0,0807	0,0807	0,0807
29	0,08	0,084	0,084	0,084	0,0736	0,0744	0,08	0,08	0,08	0,08	0,08	0,08	0,0813	0,0849	0,0849	0,0849	0,0767	0,0896	0,0813	0,0813	0,0813	0,0813	0,0813	0,0813
30	0,0768	0,0768	0,0768	0,0768	0,0704	0,0688	0,0768	0,0768	0,0768	0,0704	0,0684	0,0672	0,0840	0,0840	0,0840	0,0840	0,0735	0,0824	0,0840	0,0840	0,0840	0,0840	0,0840	0,0840

Fold	Base 3																							
	5.000 registros																							
	Redes Neurais, 20 Neurônios, BPROP												Redes Neurais, 20 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0667	0,0667	0,0667	0,0667	0,074	0,0672	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0624	0,0624	0,0624	0,0624	0,0836	0,0866	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624
2	0,0594	0,0619	0,0602	0,0602	0,074	0,0701	0,0594	0,0594	0,0594	0,0594	0,0594	0,0594	0,0504	0,0837	0,0679	0,0679	0,0721	0,0782	0,0504	0,0504	0,0504	0,0504	0,0504	0,0504
3	0,0692	0,0705	0,0824	0,0824	0,074	0,0636	0,0692	0,0692	0,0692	0,0692	0,0692	0,0692	0,0604	0,0788	0,0865	0,0865	0,0742	0,0767	0,0604	0,0604	0,0604	0,0604	0,0604	0,0604
4	0,0808	0,0730	0,0730	0,0730	0,074	0,0695	0,0808	0,0808	0,0808	0,0808	0,0808	0,0808	0,0698	0,0746	0,0746	0,0746	0,0676	0,0618	0,0698	0,0698	0,0698	0,0698	0,0698	0,0698
5	0,0590	0,0588	0,0588	0,0588	0,074	0,0632	0,0590	0,0590	0,0590	0,0690	0,0728	0,0751	0,0512	0,0605	0,0605	0,0605	0,0755	0,0744	0,0512	0,0512	0,0512	0,0662	0,0658	0,0728
6	0,0767	0,0819	0,0748	0,0748	0,074	0,0710	0,0767	0,0767	0,0767	0,0701	0,07	0,0699	0,0776	0,0719	0,0768	0,0768	0,0786	0,0901	0,0776	0,0776	0,0776	0,0814	0,0730	0,0640
7	0,0837	0,0837	0,0837	0,0837	0,074	0,0653	0,08	0,08	0,08	0,0837	0,0837	0,0837	0,0661	0,0661	0,0661	0,0661	0,0631	0,0850	0,0683	0,0683	0,0683	0,0661	0,0661	0,0661
8	0,0656	0,0656	0,0656	0,0656	0,074	0,0658	0,0656	0,0656	0,0656	0,0727	0,0722	0,0719	0,0479	0,0479	0,0479	0,0479	0,0795	0,0574	0,0479	0,0479	0,0479	0,0780	0,0615	0,0727
9	0,0833	0,0833	0,0833	0,0833	0,074	0,0573	0,088	0,0824	0,0824	0,0833	0,0833	0,0833	0,0916	0,0916	0,0916	0,0916	0,0708	0,0811	0,0916	0,0762	0,0762	0,0916	0,0916	0,0916
10	0,0743	0,0669	0,0728	0,0728	0,074	0,0557	0,0743	0,0743	0,0743	0,0743	0,0743	0,0743	0,0670	0,0716	0,0801	0,0801	0,0684	0,0614	0,0670	0,0670	0,0670	0,0670	0,0670	0,0670
11	0,0569	0,0575	0,0575	0,0575	0,074	0,0601	0,0569	0,0569	0,0569	0,0569	0,0569	0,0569	0,0574	0,0650	0,0650	0,0650	0,0746	0,0747	0,0574	0,0574	0,0574	0,0574	0,0574	0,0574
12	0,0485	0,0485	0,0485	0,0485	0,074	0,0808	0,0485	0,0485	0,0485	0,0632	0,0636	0,0638	0,0522	0,0522	0,0522	0,0522	0,0728	0,0894	0,0522	0,0522	0,0522	0,0707	0,0682	0,0669
13	0,0751	0,0790	0,0790	0,0790	0,074	0,0606	0,0751	0,0751	0,0751	0,0788	0,0799	0,0805	0,0817	0,0559	0,0559	0,0559	0,0728	0,0935	0,0817	0,0817	0,0817	0,0788	0,0755	0,0864
14	0,0646	0,0651	0,0651	0,0651	0,074	0,0606	0,0646	0,0646	0,0646	0,0658	0,0659	0,0660	0,0678	0,0709	0,0709	0,0709	0,0764	0,0837	0,0678	0,0678	0,0678	0,0720	0,0698	0,0581
15	0,0867	0,0867	0,0867	0,0867	0,074	0,0707	0,0867	0,0867	0,0867	0,0844	0,0859	0,0868	0,0675	0,0675	0,0675	0,0675	0,0720	0,0870	0,0675	0,0675	0,0675	0,0836	0,0823	0,0987
16	0,0652	0,0765	0,0765	0,0765	0,074	0,0793	0,0652	0,0652	0,0652	0,0637	0,0637	0,0637	0,0659	0,0598	0,0598	0,0598	0,0710	0,0856	0,0659	0,0659	0,0659	0,0629	0,0629	0,0629
17	0,0570	0,0630	0,0606	0,0606	0,074	0,0584	0,0570	0,0570	0,0570	0,0570	0,0570	0,0570	0,0676	0,0556	0,0611	0,0611	0,0706	0,0743	0,0676	0,0676	0,0676	0,0676	0,0676	0,0676
18	0,0591	0,0591	0,0591	0,0591	0,074	0,0640	0,0816	0,0816	0,0816	0,0702	0,0693	0,0688	0,0562	0,0562	0,0562	0,0562	0,0857	0,0637	0,0888	0,0888	0,0888	0,0561	0,0686	0,0779
19	0,0734	0,0626	0,0626	0,0626	0,074	0,0614	0,0734	0,0734	0,0734	0,0734	0,0734	0,0734	0,0752	0,0660	0,0660	0,0660	0,0672	0,0966	0,0752	0,0752	0,0752	0,0752	0,0752	0,0752
20	0,0651	0,0629	0,0629	0,0629	0,074	0,0714	0,0651	0,0651	0,0651	0,0651	0,0651	0,0651	0,0605	0,0622	0,0622	0,0622	0,0672	0,0733	0,0605	0,0605	0,0605	0,0605	0,0605	0,0605
21	0,0521	0,0521	0,0521	0,0521	0,074	0,0581	0,0521	0,0521	0,0521	0,0521	0,0521	0,0521	0,0505	0,0505	0,0505	0,0505	0,0701	0,0741	0,0505	0,0505	0,0505	0,0505	0,0505	0,0505
22	0,0757	0,0774	0,0774	0,0774	0,074	0,0616	0,0757	0,0757	0,0757	0,0826	0,0827	0,0828	0,0742	0,0822	0,0822	0,0822	0,0845	0,0860	0,0742	0,0742	0,0742	0,0744	0,0673	0,0690
23	0,0704	0,0798	0,0798	0,0798	0,074	0,0762	0,0704	0,0704	0,0704	0,0704	0,0704	0,0704	0,0603	0,0825	0,0825	0,0825	0,0707	0,0736	0,0603	0,0603	0,0603	0,0603	0,0603	0,0603
24	0,0709	0,0709	0,0709	0,0709	0,074	0,0662	0,0709	0,0709	0,0709	0,0709	0,0709	0,0709	0,0628	0,0628	0,0628	0,0628	0,0672	0,0651	0,0628	0,0628	0,0628	0,0628	0,0628	0,0628
25	0,0630	0,0630	0,0630	0,0630	0,074	0,0620	0,0630	0,0630	0,0630	0,0630	0,0630	0,0630	0,0634	0,0634	0,0634	0,0634	0,0799	0,0897	0,0634	0,0634	0,0634	0,0634	0,0634	0,0634
26	0,0510	0,0652	0,0652	0,0652	0,074	0,0680	0,0510	0,0510	0,0510	0,0510	0,0510	0,0510	0,0503	0,0615	0,0615	0,0615	0,0576	0,0853	0,0503	0,0503	0,0503	0,0503	0,0503	0,0503
27	0,0592	0,0592	0,0592	0,0592	0,074	0,0621	0,0592	0,0592	0,0592	0,0592	0,0592	0,0592	0,0585	0,0585	0,0585	0,0585	0,0576	0,0753	0,0585	0,0585	0,0585	0,0585	0,0585	0,0585
28	0,0832	0,0947	0,0947	0,0947	0,074	0,0640	0,0832	0,0832	0,0832	0,0832	0,0832	0,0832	0,0921	0,0796	0,0796	0,0796	0,0782	0,0701	0,0921	0,0921	0,0921	0,0921	0,0921	0,0921
29	0,0861	0,0792	0,0792	0,0792	0,074	0,0692	0,0861	0,0861	0,0861	0,0861	0,0861	0,0861	0,0842	0,0896	0,0896	0,0896	0,0683	0,0622	0,0842	0,0842	0,0842	0,0842	0,0842	0,0842
30	0,0780	0,0780	0,0780	0,0780	0,074	0,0680	0,0780	0,0780	0,0780	0,0704	0,0684	0,0672	0,0858	0,0858	0,0858	0,0858	0,0683	0,0907	0,0858	0,0858	0,0858	0,0774	0,0739	0,0717

Tabela A.19 – Erros de Classificação da Base (4) para 1.000 registros.

Fold	Base 4																							
	1.000 registros																							
	Regressão Linear												Regressão Logística											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,136	0,096	0,084	0,084	0,132	0,136	0,08	0,084	0,084	0,2639	0,0780	0,0806	0,092	0,084	0,084	0,084	0,088	0,092	0,092	0,092	0,092	0,0989	0,0929	0,0909
2	0,1	0,084	0,1	0,1	0,104	0,104	0,1	0,1	0,1	0,1552	0,0839	0,0810	0,092	0,088	0,088	0,088	0,096	0,096	0,092	0,092	0,092	0,092	0,092	0,092
3	0,112	0,06	0,068	0,068	0,112	0,116	0,068	0,076	0,076	0,2375	0,0727	0,0757	0,048	0,052	0,06	0,06	0,052	0,052	0,048	0,048	0,048	0,045	0,045	0,045
4	0,12	0,112	0,124	0,124	0,12	0,124	0,104	0,12	0,12	0,2097	0,1245	0,1273	0,108	0,108	0,116	0,116	0,112	0,108	0,108	0,108	0,108	0,1146	0,1210	0,1246
5	0,156	0,096	0,1	0,1	0,144	0,148	0,08	0,08	0,08	0,2468	0,0522	0,0500	0,088	0,088	0,088	0,088	0,088	0,088	0,092	0,068	0,068	0,088	0,088	0,088
6	0,152	0,088	0,096	0,096	0,148	0,152	0,104	0,092	0,092	0,2255	0,1021	0,1018	0,072	0,092	0,088	0,088	0,088	0,088	0,088	0,092	0,092	0,0802	0,0821	0,0860
7	0,148	0,08	0,076	0,076	0,152	0,144	0,084	0,088	0,088	0,2172	0,0875	0,0867	0,084	0,084	0,092	0,092	0,084	0,084	0,084	0,084	0,084	0,0888	0,0785	0,0797
8	0,116	0,052	0,044	0,044	0,12	0,12	0,052	0,056	0,056	0,2005	0,0985	0,0961	0,056	0,056	0,052	0,052	0,052	0,048	0,056	0,056	0,056	0,0601	0,0601	0,0601
9	0,136	0,068	0,1	0,1	0,144	0,136	0,088	0,088	0,088	0,2264	0,0671	0,0662	0,096	0,112	0,092	0,092	0,112	0,108	0,092	0,092	0,092	0,0737	0,0727	0,0706
10	0,108	0,076	0,076	0,076	0,108	0,1	0,084	0,084	0,084	0,1585	0,0982	0,0917	0,092	0,068	0,08	0,08	0,08	0,084	0,092	0,092	0,092	0,0585	0,0614	0,0630
11	0,084	0,056	0,048	0,048	0,1	0,1	0,092	0,088	0,088	0,2624	0,0800	0,0793	0,096	0,068	0,068	0,068	0,092	0,088	0,076	0,076	0,076	0,096	0,096	0,096
12	0,152	0,088	0,08	0,08	0,148	0,152	0,084	0,092	0,092	0,2048	0,0614	0,0589	0,076	0,072	0,076	0,076	0,072	0,072	0,092	0,092	0,092	0,0585	0,0596	0,0575
13	0,152	0,084	0,072	0,072	0,148	0,152	0,064	0,092	0,092	0,2575	0,0774	0,0799	0,056	0,064	0,072	0,072	0,048	0,052	0,056	0,056	0,056	0,0631	0,0631	0,0631
14	0,104	0,068	0,092	0,092	0,1	0,1	0,1	0,1	0,1	0,2166	0,0845	0,0868	0,08	0,092	0,08	0,08	0,076	0,068	0,108	0,096	0,096	0,0654	0,0735	0,0781
15	0,12	0,064	0,064	0,064	0,112	0,112	0,056	0,076	0,076	0,2184	0,0705	0,0671	0,064	0,064	0,064	0,064	0,06	0,06	0,064	0,064	0,064	0,0578	0,0529	0,0531
16	0,1	0,096	0,096	0,096	0,104	0,104	0,1	0,1	0,1	0,2	0,0785	0,0797	0,064	0,096	0,088	0,088	0,072	0,072	0,088	0,072	0,072	0,0716	0,075	0,0769
17	0,148	0,1	0,1	0,1	0,156	0,156	0,064	0,068	0,068	0,2375	0,0727	0,0671	0,092	0,072	0,072	0,072	0,092	0,092	0,092	0,076	0,076	0,07	0,0563	0,0542
18	0,124	0,092	0,084	0,084	0,124	0,12	0,116	0,116	0,116	0,1771	0,1027	0,1019	0,128	0,092	0,088	0,088	0,124	0,124	0,112	0,104	0,104	0,1019	0,0993	0,0991
19	0,1	0,06	0,064	0,064	0,108	0,104	0,076	0,068	0,068	0,1553	0,0766	0,0803	0,068	0,056	0,072	0,072	0,072	0,08	0,068	0,068	0,068	0,068	0,068	0,068
20	0,1	0,076	0,08	0,08	0,096	0,096	0,084	0,06	0,06	0,2454	0,0820	0,0801	0,072	0,076	0,08	0,08	0,076	0,076	0,08	0,08	0,08	0,0465	0,0553	0,0590
21	0,116	0,104	0,108	0,108	0,112	0,108	0,116	0,116	0,116	0,1551	0,0943	0,1002	0,084	0,092	0,084	0,084	0,084	0,084	0,084	0,084	0,084	0,0837	0,0943	0,1002
22	0,12	0,08	0,092	0,092	0,116	0,116	0,056	0,056	0,056	0,1629	0,075	0,0769	0,06	0,072	0,064	0,064	0,064	0,064	0,04	0,04	0,04	0,0666	0,0666	0,0666
23	0,088	0,068	0,064	0,064	0,08	0,084	0,088	0,084	0,084	0,1785	0,0973	0,0946	0,076	0,08	0,084	0,084	0,068	0,072	0,076	0,076	0,076	0,076	0,076	0,076
24	0,128	0,128	0,104	0,104	0,124	0,124	0,112	0,104	0,104	0,2129	0,0884	0,0870	0,1	0,084	0,088	0,088	0,108	0,108	0,108	0,108	0,108	0,0779	0,075	0,0763
25	0,12	0,108	0,088	0,088	0,116	0,116	0,112	0,1	0,1	0,2311	0,0695	0,0720	0,108	0,1	0,1	0,1	0,108	0,104	0,108	0,108	0,108	0,108	0,108	0,108
26	0,096	0,08	0,076	0,076	0,092	0,092	0,08	0,076	0,076	0,2171	0,0793	0,0770	0,084	0,088	0,076	0,076	0,084	0,088	0,076	0,072	0,072	0,0631	0,0631	0,0631
27	0,1	0,108	0,116	0,116	0,1	0,104	0,084	0,076	0,076	0,1	0,1	0,1	0,108	0,1	0,1	0,1	0,104	0,096	0,088	0,084	0,084	0,108	0,108	0,108
28	0,132	0,12	0,108	0,108	0,136	0,136	0,104	0,12	0,12	0,1932	0,1121	0,1129	0,12	0,096	0,096	0,096	0,116	0,116	0,12	0,12	0,12	0,12	0,12	0,12
29	0,128	0,072	0,084	0,084	0,124	0,128	0,064	0,068	0,068	0,2272	0,0738	0,0726	0,072	0,072	0,072	0,072	0,072	0,072	0,072	0,072	0,072	0,072	0,072	0,072
30	0,108	0,08	0,084	0,084	0,096	0,116	0,088	0,112	0,112	0,1818	0,0886	0,0859	0,072	0,076	0,072	0,072	0,072	0,076	0,084	0,092	0,092	0,0737	0,0737	0,0737

Fold	Base 4																							
	1.000 registros																							
	Redes Neurais, 3 Neurônios, BPROP												Redes Neurais, 3 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0852	0,0852	0,0852	0,0852	0,0725	0,0804	0,0852	0,0852	0,0852	0,0939	0,0929	0,0923	0,08	0,08	0,08	0,08	0,084	0,08	0,084	0,084	0,084	0,0989	0,1003	0,1011
2	0,0814	0,0713	0,0880	0,0880	0,0662	0,0701	0,0814	0,0814	0,0814	0,0814	0,0814	0,0814	0,076	0,072	0,076	0,076	0,068	0,06	0,076	0,076	0,076	0,076	0,076	0,076
3	0,0587	0,0781	0,0781	0,0781	0,0452	0,0552	0,0587	0,0587	0,0587	0,0575	0,0545	0,0528	0,056	0,08	0,08	0,08	0,048	0,056	0,056	0,056	0,056	0,056	0,056	0,056
4	0,0986	0,1022	0,0829	0,0829	0,0931	0,0954	0,0986	0,0986	0,0986	0,0986	0,0986	0,0986	0,1	0,092	0,084	0,084	0,092	0,088	0,096	0,1	0,1	0,1121	0,1121	0,1121
5	0,0666	0,0543	0,0543	0,0543	0,0480	0,0396	0,08	0,08	0,08	0,0737	0,0671	0,0633	0,064	0,06	0,06	0,06	0,044	0,044	0,064	0,064	0,064	0,064	0,064	0,064
6	0,0523	0,0958	0,0958	0,0958	0,0647	0,0818	0,0523	0,0523	0,0523	0,0523	0,0523	0,0523	0,064	0,092	0,092	0,092	0,068	0,076	0,064	0,064	0,064	0,0877	0,0875	0,0875
7	0,0728	0,0799	0,0901	0,0901	0,0587	0,0711	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728	0,08	0,088	0,092	0,092	0,068	0,068	0,08	0,08	0,08	0,08	0,08	0,08
8	0,0617	0,0684	0,0684	0,0684	0,0513	0,0431	0,0617	0,0617	0,0617	0,0617	0,0617	0,0617	0,056	0,068	0,068	0,068	0,04	0,044	0,056	0,056	0,056	0,056	0,056	0,056
9	0,0697	0,0938	0,0876	0,0876	0,0727	0,0621	0,088	0,088	0,088	0,0712	0,0690	0,0677	0,068	0,088	0,084	0,084	0,068	0,068	0,072	0,072	0,072	0,0610	0,0615	0,0618
10	0,0655	0,0655	0,0655	0,0655	0,0763	0,0669	0,08	0,08	0,08	0,0655	0,0655	0,0655	0,072	0,072	0,072	0,072	0,068	0,072	0,072	0,072	0,072	0,072	0,072	0,072
11	0,0606	0,0625	0,0821	0,0821	0,0637	0,0649	0,068	0,08	0,08	0,0629	0,0605	0,0590	0,056	0,076	0,08	0,08	0,064	0,06	0,056	0,056	0,056	0,056	0,056	0,056
12	0,0502	0,0502	0,0502	0,0502	0,0477	0,0305	0,0502	0,0502	0,0502	0,0585	0,0561	0,0547	0,048	0,048	0,048	0,048	0,044	0,044	0,048	0,048	0,048	0,0585	0,0561	0,0547
13	0,0450	0,0824	0,0824	0,0824	0,0276	0,0518	0,0450	0,0450	0,0450	0,0450	0,0450	0,0450	0,056	0,08	0,08	0,08	0,036	0,04	0,068	0,076	0,076	0,056	0,056	0,056
14	0,0712	0,0868	0,0868	0,0868	0,0640	0,0826	0,0712	0,0712	0,0712	0,0712	0,0712	0,0712	0,068	0,1	0,1	0,1	0,068	0,076	0,068	0,068	0,068	0,068	0,068	0,068
15	0,0654	0,0550	0,0550	0,0550	0,0458	0,0487	0,0654	0,0654	0,0654	0,0763	0,0725	0,0703	0,072	0,06	0,06	0,06	0,048	0,048	0,072	0,072	0,072	0,0657	0,0627	0,0609
16	0,0626	0,0829	0,1000	0,1000	0,0520	0,0546	0,064	0,064	0,064	0,0626	0,0626	0,0626	0,064	0,088	0,096	0,096	0,048	0,056	0,096	0,1	0,1	0,0814	0,0821	0,0825
17	0,0495	0,0659	0,0680	0,0680	0,0505	0,0594	0,08	0,104	0,104	0,065	0,0618	0,06	0,048	0,068	0,072	0,072	0,048	0,048	0,048	0,048	0,048	0,048	0,048	0,048
18	0,0801	0,0708	0,0970	0,0970	0,0929	0,0844	0,0801	0,0801	0,0801	0,0801	0,0801	0,0801	0,088	0,08	0,092	0,092	0,088	0,088	0,088	0,088	0,088	0,088	0,088	0,088
19	0,0822	0,0822	0,0822	0,0822	0,0753	0,0879	0,0822	0,0822	0,0822	0,0822	0,0822	0,0822	0,08	0,08	0,08	0,08	0,072	0,088	0,12	0,092	0,092	0,0776	0,0776	0,0776
20	0,0645	0,0794	0,0722	0,0722	0,0562	0,0553	0,0645	0,0645	0,0645	0,0645	0,0645	0,0645	0,064	0,072	0,08	0,08	0,064	0,064	0,064	0,064	0,064	0,0775	0,0801	0,0816
21	0,1026	0,1026	0,1026	0,1026	0,1041	0,1002	0,104	0,096	0,096	0,1059	0,1067	0,1072	0,104	0,104	0,104	0,104	0,112	0,108	0,104	0,104	0,104	0,1034	0,1014	0,1002
22	0,0522	0,0542	0,0863	0,0863	0,0338	0,0557	0,0522	0,0522	0,0522	0,0522	0,0522	0,0522	0,056	0,056	0,072	0,072	0,04	0,06	0,092	0,092	0,092	0,0938	0,0892	0,0867
23	0,0626	0,0535	0,0535	0,0535	0,0498	0,0740	0,0626	0,0626	0,0626	0,0841	0,0880	0,0902	0,056	0,06	0,06	0,06	0,06	0,068	0,068	0,068	0,068	0,0994	0,1048	0,1079
24	0,0691	0,0688	0,0675	0,0675	0,0864	0,0793	0,076	0,076	0,076	0,0961	0,0903	0,0870	0,076	0,068	0,064	0,064	0,084	0,072	0,084	0,084	0,084	0,1012	0,0980	0,0961
25	0,0772	0,0905	0,0905	0,0905	0,0833	0,0987	0,0772	0,0772	0,0772	0,0772	0,0772	0,0772	0,08	0,092	0,092	0,092	0,084	0,092	0,08	0,08	0,08	0,08	0,08	0,08
26	0,0739	0,0645	0,0708	0,0708	0,0538	0,0629	0,0739	0,0739	0,0739	0,0739	0,0739	0,0739	0,068	0,072	0,072	0,072	0,06	0,06	0,068	0,068	0,068	0,068	0,068	0,068
27	0,0656	0,0657	0,1076	0,1076	0,0708	0,0774	0,0656	0,0656	0,0656	0,0656	0,0656	0,0656	0,064	0,076	0,104	0,104	0,06	0,06	0,064	0,064	0,064	0,064	0,064	0,064
28	0,0714	0,0916	0,1251	0,1251	0,0740	0,0773	0,0714	0,0714	0,0714	0,0714	0,0714	0,0714	0,076	0,092	0,112	0,112	0,072	0,072	0,076	0,076	0,076	0,1134	0,1134	0,1134
29	0,0815	0,0821	0,0821	0,0821	0,0415	0,0382	0,06	0,06	0,06	0,0782	0,0782	0,0782	0,092	0,084	0,084	0,084	0,044	0,04	0,104	0,104	0,104	0,0808	0,0793	0,0784
30	0,0675	0,0675	0,0675	0,0675	0,0702	0,0603	0,06	0,06	0,06	0,0638	0,0656	0,0665	0,068	0,068	0,068	0,068	0,072	0,068	0,068	0,068	0,068	0,068	0,068	0,068

Tabela A.20 – Erros de Classificação da Base (4) para 1.000 registros (continuação).

Fold	Base 4																							
	1.000 registros																							
	Redes Neurais, 10 Neurônios, BPROP												Redes Neurais, 10 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,092	0,092	0,092	0,092	0,076	0,084	0,092	0,092	0,092	0,0812	0,0836	0,0850	0,08	0,088	0,088	0,088	0,08	0,076	0,08	0,08	0,08	0,0888	0,0892	0,0894
2	0,088	0,084	0,084	0,084	0,064	0,064	0,088	0,088	0,088	0,088	0,088	0,088	0,08	0,08	0,08	0,08	0,068	0,068	0,08	0,08	0,08	0,08	0,08	0,08
3	0,08	0,092	0,088	0,088	0,064	0,064	0,08	0,08	0,08	0,08	0,08	0,08	0,06	0,064	0,064	0,064	0,06	0,056	0,06	0,06	0,06	0,06	0,06	0,06
4	0,104	0,104	0,104	0,104	0,084	0,088	0,104	0,104	0,104	0,104	0,104	0,104	0,104	0,1	0,12	0,12	0,092	0,096	0,104	0,104	0,104	0,1170	0,1192	0,1205
5	0,068	0,092	0,12	0,12	0,044	0,052	0,096	0,096	0,096	0,0763	0,0690	0,0648	0,052	0,068	0,052	0,052	0,044	0,06	0,052	0,052	0,052	0,0916	0,0802	0,0736
6	0,08	0,092	0,092	0,092	0,076	0,064	0,08	0,08	0,08	0,0776	0,0748	0,0731	0,068	0,088	0,088	0,088	0,064	0,064	0,064	0,064	0,064	0,0701	0,0693	0,0688
7	0,112	0,08	0,08	0,08	0,092	0,092	0,14	0,14	0,14	0,112	0,112	0,112	0,076	0,076	0,076	0,076	0,068	0,064	0,104	0,104	0,104	0,076	0,076	0,076
8	0,064	0,084	0,084	0,084	0,056	0,06	0,064	0,064	0,064	0,0776	0,0802	0,0817	0,044	0,056	0,056	0,056	0,036	0,044	0,044	0,044	0,044	0,0726	0,0766	0,0789
9	0,044	0,06	0,064	0,064	0,06	0,072	0,044	0,044	0,044	0,0661	0,0652	0,0648	0,056	0,052	0,052	0,052	0,064	0,072	0,088	0,084	0,084	0,0763	0,0727	0,0706
10	0,092	0,096	0,096	0,096	0,096	0,104	0,092	0,092	0,092	0,092	0,092	0,092	0,076	0,076	0,076	0,076	0,068	0,076	0,076	0,076	0,076	0,0902	0,0842	0,0808
11	0,072	0,072	0,072	0,072	0,068	0,052	0,072	0,072	0,072	0,072	0,072	0,072	0,052	0,06	0,076	0,076	0,064	0,056	0,052	0,052	0,052	0,052	0,052	0,052
12	0,056	0,068	0,068	0,068	0,064	0,052	0,056	0,056	0,056	0,0585	0,0561	0,0547	0,036	0,056	0,056	0,056	0,04	0,032	0,036	0,036	0,036	0,0634	0,0596	0,0575
13	0,072	0,088	0,088	0,088	0,068	0,08	0,072	0,072	0,072	0,072	0,072	0,072	0,056	0,068	0,072	0,072	0,04	0,036	0,096	0,084	0,084	0,0833	0,0811	0,0799
14	0,092	0,088	0,12	0,12	0,068	0,076	0,092	0,092	0,092	0,0931	0,0882	0,0853	0,084	0,092	0,092	0,092	0,08	0,088	0,084	0,084	0,084	0,0806	0,0790	0,0781
15	0,076	0,084	0,084	0,084	0,056	0,064	0,076	0,076	0,076	0,0815	0,0745	0,0703	0,056	0,056	0,056	0,056	0,052	0,048	0,084	0,084	0,084	0,0684	0,0647	0,0625
16	0,068	0,072	0,072	0,072	0,068	0,068	0,068	0,068	0,068	0,0691	0,0732	0,0755	0,064	0,08	0,08	0,08	0,064	0,068	0,064	0,064	0,064	0,0641	0,0696	0,0727
17	0,096	0,128	0,132	0,132	0,096	0,108	0,096	0,096	0,096	0,096	0,096	0,096	0,056	0,08	0,084	0,084	0,068	0,076	0,132	0,112	0,112	0,0875	0,0781	0,0728
18	0,12	0,116	0,144	0,144	0,084	0,096	0,12	0,12	0,12	0,12	0,12	0,12	0,088	0,088	0,084	0,084	0,088	0,08	0,088	0,088	0,088	0,088	0,088	0,088
19	0,064	0,068	0,076	0,076	0,076	0,08	0,064	0,064	0,064	0,064	0,064	0,064	0,064	0,088	0,088	0,088	0,06	0,056	0,092	0,092	0,092	0,1027	0,0930	0,0875
20	0,092	0,084	0,084	0,084	0,08	0,072	0,092	0,092	0,092	0,092	0,092	0,092	0,076	0,084	0,088	0,088	0,064	0,076	0,084	0,112	0,112	0,0956	0,0992	0,1013
21	0,108	0,108	0,108	0,108	0,104	0,104	0,108	0,108	0,108	0,108	0,108	0,108	0,104	0,104	0,108	0,108	0,1	0,112	0,1	0,1	0,1	0,1157	0,1120	0,1100
22	0,08	0,096	0,088	0,088	0,072	0,064	0,092	0,092	0,092	0,0814	0,0767	0,0741	0,068	0,056	0,06	0,06	0,052	0,056	0,068	0,068	0,068	0,0814	0,0767	0,0741
23	0,076	0,068	0,096	0,096	0,064	0,072	0,084	0,084	0,084	0,0969	0,1011	0,1035	0,076	0,056	0,072	0,072	0,06	0,068	0,088	0,088	0,088	0,0943	0,0992	0,1020
24	0,084	0,092	0,096	0,096	0,072	0,068	0,084	0,084	0,084	0,0909	0,0865	0,0839	0,076	0,076	0,076	0,076	0,076	0,088	0,124	0,124	0,124	0,0857	0,0826	0,0809
25	0,08	0,076	0,104	0,104	0,064	0,06	0,092	0,092	0,092	0,08	0,08	0,08	0,08	0,084	0,084	0,084	0,072	0,072	0,116	0,116	0,116	0,0929	0,0952	0,0965
26	0,06	0,06	0,06	0,06	0,056	0,06	0,06	0,06	0,06	0,0681	0,0701	0,0712	0,064	0,072	0,08	0,08	0,064	0,06	0,064	0,064	0,064	0,0681	0,0701	0,0712
27	0,068	0,076	0,076	0,076	0,08	0,068	0,076	0,076	0,076	0,068	0,068	0,068	0,068	0,068	0,068	0,068	0,06	0,06	0,068	0,068	0,068	0,0676	0,0602	0,0559
28	0,092	0,116	0,116	0,116	0,08	0,096	0,092	0,092	0,092	0,1108	0,1121	0,1129	0,096	0,092	0,092	0,092	0,08	0,072	0,096	0,096	0,096	0,1082	0,1102	0,1114
29	0,064	0,064	0,064	0,064	0,056	0,06	0,044	0,044	0,044	0,064	0,064	0,064	0,06	0,056	0,056	0,056	0,052	0,048	0,088	0,088	0,088	0,0707	0,0682	0,0668
30	0,084	0,08	0,08	0,08	0,072	0,072	0,092	0,092	0,092	0,0663	0,0656	0,0651	0,06	0,06	0,064	0,064	0,068	0,064	0,072	0,072	0,072	0,0687	0,0691	0,0693

Fold	Base 4																							
	1.000 registros																							
	Redes Neurais, 20 Neurônios, BPROP												Redes Neurais, 20 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,092	0,08	0,08	0,08	0,092	0,108	0,076	0,076	0,076	0,0913	0,0910	0,0909	0,108	0,084	0,076	0,076	0,0999	0,0792	0,108	0,108	0,108	0,0964	0,0947	0,0938
2	0,064	0,064	0,064	0,064	0,068	0,064	0,064	0,064	0,064	0,064	0,064	0,064	0,076	0,092	0,092	0,092	0,0698	0,1059	0,076	0,076	0,076	0,076	0,076	0,076
3	0,064	0,1	0,1	0,1	0,08	0,08	0,088	0,088	0,088	0,0725	0,0725	0,0725	0,1	0,104	0,104	0,104	0,0770	0,1045	0,1	0,1	0,1	0,1	0,1	0,1
4	0,108	0,116	0,104	0,104	0,088	0,092	0,112	0,112	0,112	0,1146	0,1175	0,1191	0,108	0,128	0,128	0,128	0,0761	0,0966	0,12	0,12	0,12	0,1121	0,1157	0,1178
5	0,104	0,108	0,108	0,108	0,064	0,084	0,112	0,112	0,112	0,0839	0,0746	0,0692	0,112	0,104	0,1	0,1	0,0652	0,0908	0,112	0,112	0,112	0,0865	0,0764	0,0706
6	0,088	0,08	0,088	0,088	0,088	0,076	0,084	0,084	0,084	0,0852	0,0839	0,0832	0,084	0,076	0,076	0,076	0,0826	0,0882	0,08	0,08	0,08	0,0827	0,0802	0,0789
7	0,1	0,108	0,108	0,108	0,096	0,096	0,1	0,1	0,1	0,1	0,1	0,1	0,112	0,12	0,092	0,092	0,0893	0,0790	0,104	0,104	0,104	0,112	0,112	0,112
8	0,08	0,064	0,06	0,06	0,06	0,06	0,068	0,068	0,068	0,0776	0,0802	0,0817	0,088	0,06	0,084	0,084	0,0590	0,0810	0,088	0,088	0,088	0,0902	0,0894	0,0889
9	0,064	0,068	0,068	0,068	0,072	0,076	0,064	0,064	0,064	0,0636	0,0634	0,0633	0,056	0,048	0,056	0,056	0,0662	0,0794	0,084	0,084	0,084	0,0636	0,0634	0,0633
10	0,104	0,104	0,104	0,104	0,1	0,108	0,104	0,104	0,104	0,104	0,104	0,104	0,116	0,116	0,116	0,116	0,1091	0,0987	0,108	0,108	0,108	0,0780	0,0736	0,0712
11	0,08	0,072	0,084	0,084	0,06	0,064	0,072	0,072	0,072	0,0629	0,0625	0,0622	0,076	0,084	0,068	0,068	0,0643	0,0706	0,08	0,08	0,08	0,076	0,076	0,076
12	0,04	0,076	0,056	0,056	0,056	0,06	0,04	0,04	0,04	0,0560	0,0578	0,0589	0,056	0,068	0,08	0,08	0,0615	0,1142	0,076	0,076	0,076	0,0536	0,0526	0,0520
13	0,088	0,088	0,088	0,088	0,068	0,072	0,088	0,088	0,088	0,088	0,088	0,088	0,076	0,1	0,108	0,108	0,0705	0,0736	0,076	0,076	0,076	0,0934	0,0922	0,0915
14	0,08	0,084	0,088	0,088	0,068	0,08	0,08	0,08	0,08	0,0780	0,0753	0,0738	0,092	0,08	0,104	0,104	0,0730	0,0906	0,092	0,092	0,092	0,0831	0,0772	0,0738
15	0,072	0,076	0,076	0,076	0,048	0,064	0,072	0,072	0,072	0,0894	0,0803	0,075	0,084	0,088	0,088	0,088	0,0400	0,0855	0,084	0,084	0,084	0,0868	0,0784	0,0734
16	0,068	0,104	0,104	0,104	0,068	0,068	0,068	0,068	0,068	0,0740	0,0767	0,0783	0,072	0,088	0,088	0,088	0,0637	0,0899	0,072	0,072	0,072	0,0839	0,0839	0,0839
17	0,112	0,112	0,112	0,112	0,108	0,116	0,112	0,112	0,112	0,095	0,0836	0,0771	0,108	0,108	0,108	0,108	0,1079	0,1044	0,108	0,108	0,108	0,09	0,08	0,0742
18	0,1	0,112	0,112	0,112	0,092	0,096	0,124	0,124	0,124	0,1019	0,1080	0,1114	0,112	0,12	0,112	0,112	0,1008	0,0751	0,116	0,116	0,116	0,1092	0,1114	0,1127
19	0,072	0,088	0,088	0,088	0,084	0,076	0,072	0,072	0,072	0,072	0,072	0,072	0,084	0,088	0,092	0,092	0,0816	0,0746	0,084	0,084	0,084	0,0802	0,0766	0,0746
20	0,08	0,08	0,08	0,08	0,064	0,084	0,08	0,08	0,08	0,08	0,08	0,08	0,104	0,092	0,108	0,108	0,0706	0,0934	0,084	0,084	0,084	0,0956	0,0858	0,0801
21	0,128	0,108	0,108	0,108	0,124	0,108	0,132	0,132	0,132	0,1305	0,1227	0,1183	0,144	0,124	0,112	0,112	0,1260	0,0402	0,144	0,128	0,128	0,1256	0,1192	0,1155
22	0,08	0,084	0,092	0,092	0,064	0,08	0,08	0,08	0,08	0,0641	0,0642	0,0643	0,092	0,084	0,084	0,084	0,0691	0,0815	0,092	0,092	0,092	0,0790	0,075	0,0727
23	0,06	0,06	0,06	0,06	0,072	0,076	0,06	0,06	0,06	0,0688	0,0730	0,0754	0,08	0,092	0,092	0,092	0,0711	0,0942	0,08	0,08	0,08	0,0739	0,0767	0,0784
24	0,068	0,076	0,076	0,076	0,06	0,08	0,064	0,064	0,064	0,0857	0,0826	0,0809	0,076	0,076	0,08	0,08	0,0620	0,1129	0,08	0,08	0,08	0,0857	0,0826	0,0809
25	0,072	0,08	0,076	0,076	0,076	0,084	0,072	0,072	0,072	0,072	0,072	0,072	0,096	0,076	0,076	0,076	0,0701	0,0907	0,096	0,096	0,096	0,0829	0,0824	0,0821
26	0,076	0,08	0,072	0,072	0,056	0,068	0,076	0,076	0,076	0,0808	0,0830	0,0843	0,076	0,076	0,076	0,076	0,0551	0,1092	0,076	0,076	0,076	0,0681	0,0681	0,0681
27	0,084	0,096	0,108	0,108	0,08	0,08	0,076	0,076	0,076	0,0676	0,0656	0,0645	0,088	0,12	0,12	0,12	0,0887	0,0801	0,088	0,088	0,088	0,0626	0,0565	0,0530
28	0,088	0,104	0,096	0,096	0,092	0,1	0,088	0,088	0,088	0,0979	0,1026	0,1054	0,112	0,112	0,112	0,112	0,0877	0,0970	0,112	0,112	0,112	0,1082	0,1102	0,1114
29	0,088	0,08	0,08	0,08	0,06	0,072	0,088	0,088	0,088	0,088	0,088	0,088	0,076	0,08	0,08	0,08	0,0436	0,0659	0,076	0,076	0,076	0,0732	0,0732	0,0732
30	0,08	0,08	0,08	0,08	0,08	0,08	0,096	0,096	0,096	0,0810	0,0780	0,0762	0,072	0,084	0,1	0,1	0,0795	0,0613	0,068	0,068	0,068	0,0835	0,0797	0,0776

Tabela A.21 – Erros de Classificação da Base (4) para 3.000 registros.

Fold	Base 4																							
	3.000 registros																							
	Regressão Linear												Regressão Logística											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,16	0,0773	0,0786	0,0786	0,1546	0,1586	0,068	0,0733	0,0733	0,2312	0,0703	0,0681	0,0706	0,0653	0,0666	0,0666	0,072	0,0706	0,0666	0,0706	0,0706	0,0708	0,0703	0,0703
2	0,1066	0,0813	0,0746	0,0746	0,1053	0,104	0,088	0,0813	0,0813	0,2493	0,0720	0,0708	0,0813	0,0786	0,0853	0,0853	0,0826	0,08	0,092	0,092	0,092	0,0707	0,0720	0,0708
3	0,108	0,076	0,0746	0,0746	0,1093	0,1106	0,108	0,108	0,108	0,1917	0,0626	0,0595	0,096	0,0866	0,08	0,08	0,0933	0,0933	0,096	0,096	0,096	0,0790	0,0675	0,0633
4	0,1186	0,0733	0,0693	0,0693	0,1186	0,1186	0,0853	0,08	0,08	0,2053	0,0948	0,0934	0,0893	0,0746	0,06	0,06	0,0866	0,088	0,0866	0,084	0,084	0,0872	0,0827	0,0839
5	0,14	0,06	0,064	0,064	0,1426	0,1413	0,08	0,08	0,08	0,2283	0,0763	0,0722	0,088	0,0653	0,068	0,068	0,0866	0,0866	0,088	0,088	0,088	0,0766	0,0720	0,0720
6	0,152	0,0813	0,084	0,084	0,14	0,144	0,0986	0,1	0,1	0,2418	0,0804	0,0767	0,0946	0,08	0,0786	0,0786	0,0933	0,0946	0,096	0,0933	0,0933	0,0803	0,0719	0,0700
7	0,1026	0,0746	0,08	0,08	0,1013	0,1	0,08	0,0813	0,0813	0,1907	0,0807	0,0792	0,0813	0,0733	0,0786	0,0786	0,084	0,084	0,0733	0,0733	0,0733	0,0743	0,0743	0,0743
8	0,1293	0,0893	0,0906	0,0906	0,1266	0,1293	0,0786	0,072	0,072	0,2212	0,0911	0,0888	0,088	0,08	0,0786	0,0786	0,084	0,084	0,088	0,088	0,088	0,0801	0,0801	0,0801
9	0,1386	0,1	0,1013	0,1013	0,1373	0,136	0,1013	0,1053	0,1053	0,2240	0,0876	0,0870	0,0986	0,0933	0,092	0,092	0,0986	0,0986	0,1013	0,1026	0,1026	0,0945	0,0888	0,0880
10	0,1293	0,084	0,0826	0,0826	0,1186	0,124	0,0973	0,104	0,104	0,2068	0,1018	0,1014	0,1	0,096	0,084	0,084	0,0973	0,0973	0,0986	0,1013	0,1013	0,0931	0,0931	0,0931
11	0,112	0,0813	0,0826	0,0826	0,1106	0,1093	0,088	0,092	0,092	0,2242	0,0780	0,0757	0,0973	0,0893	0,0906	0,0906	0,0906	0,092	0,0973	0,0973	0,0973	0,0761	0,0701	0,0695
12	0,1906	0,0666	0,0693	0,0693	0,188	0,1893	0,0706	0,0813	0,0813	0,2476	0,0677	0,0645	0,0666	0,076	0,072	0,072	0,068	0,0666	0,068	0,068	0,068	0,0665	0,0665	0,0665
13	0,1	0,068	0,072	0,072	0,1	0,1	0,0786	0,0786	0,0786	0,1590	0,0799	0,0765	0,0813	0,072	0,068	0,068	0,0813	0,0826	0,0786	0,0786	0,0786	0,0740	0,0731	0,0731
14	0,108	0,072	0,0733	0,0733	0,1093	0,1053	0,068	0,068	0,068	0,2097	0,0637	0,0642	0,072	0,0733	0,0746	0,0746	0,0706	0,0693	0,0706	0,068	0,068	0,072	0,072	0,072
15	0,112	0,068	0,0706	0,0706	0,1146	0,1146	0,088	0,08	0,08	0,2207	0,0800	0,0800	0,096	0,0693	0,0693	0,0693	0,0973	0,0986	0,096	0,096	0,096	0,0783	0,0783	0,0783
16	0,1106	0,084	0,0826	0,0826	0,1106	0,1066	0,0786	0,076	0,076	0,2254	0,0925	0,0916	0,084	0,0813	0,0733	0,0733	0,084	0,084	0,084	0,084	0,084	0,0779	0,0838	0,0848
17	0,0986	0,072	0,068	0,068	0,1	0,0973	0,0786	0,0773	0,0773	0,2011	0,0660	0,0663	0,08	0,0733	0,0693	0,0693	0,0786	0,0813	0,072	0,0746	0,0746	0,0637	0,0654	0,0654
18	0,1013	0,0533	0,0546	0,0546	0,1026	0,1026	0,08	0,084	0,084	0,1995	0,0570	0,0546	0,092	0,068	0,0666	0,0666	0,0946	0,0946	0,092	0,092	0,092	0,0695	0,0606	0,0574
19	0,0786	0,0666	0,0706	0,0706	0,0773	0,0773	0,0653	0,0706	0,0706	0,1900	0,0749	0,0735	0,0733	0,0573	0,0586	0,0586	0,0733	0,0746	0,0626	0,0706	0,0706	0,0647	0,0647	0,0647
20	0,1306	0,0826	0,0866	0,0866	0,132	0,132	0,092	0,092	0,092	0,2482	0,1029	0,1010	0,1	0,084	0,0813	0,0813	0,0946	0,0946	0,1	0,1	0,1	0,0935	0,0936	0,0936
21	0,1653	0,0893	0,088	0,088	0,1706	0,1706	0,0933	0,0986	0,0986	0,2210	0,0734	0,0695	0,096	0,0813	0,0813	0,0813	0,096	0,096	0,092	0,092	0,092	0,0759	0,0759	0,0759
22	0,0946	0,08	0,0813	0,0813	0,0906	0,092	0,0826	0,068	0,068	0,2459	0,0778	0,0763	0,0773	0,076	0,0706	0,0706	0,076	0,0746	0,0746	0,0666	0,0666	0,0793	0,0793	0,0793
23	0,0866	0,088	0,0786	0,0786	0,088	0,088	0,0813	0,0826	0,0826	0,1522	0,0750	0,0737	0,0853	0,0906	0,076	0,076	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853	0,0853
24	0,116	0,08	0,08	0,08	0,116	0,1146	0,092	0,092	0,092	0,2198	0,0895	0,0874	0,0906	0,0773	0,076	0,076	0,088	0,088	0,092	0,092	0,092	0,0840	0,0847	0,0836
25	0,1066	0,0613	0,0533	0,0533	0,1093	0,1106	0,0706	0,072	0,072	0,22	0,0745	0,0728	0,0733	0,0613	0,0586	0,0586	0,0733	0,0706	0,0693	0,0693	0,0693	0,0733	0,0733	0,0733
26	0,0906	0,068	0,0746	0,0746	0,0986	0,0973	0,0733	0,0693	0,0693	0,2010	0,0761	0,0742	0,0786	0,068	0,0666	0,0666	0,076	0,0786	0,0786	0,0786	0,0786	0,0718	0,0718	0,0718
27	0,1066	0,0626	0,068	0,068	0,1066	0,104	0,064	0,068	0,068	0,2333	0,0634	0,0621	0,0746	0,06	0,056	0,056	0,0746	0,0773	0,0746	0,0746	0,0746	0,0657	0,0657	0,0657
28	0,1653	0,0666	0,0786	0,0786	0,168	0,168	0,0786	0,0813	0,0813	0,2132	0,0843	0,0836	0,088	0,0733	0,0706	0,0706	0,0866	0,0893	0,08	0,08	0,08	0,0746	0,0746	0,0746
29	0,1013	0,0586	0,064	0,064	0,1053	0,104	0,0666	0,0693	0,0693	0,1683	0,0793	0,0754	0,0813	0,0573	0,0613	0,0613	0,0773	0,0773	0,0693	0,0693	0,0693	0,0745	0,0671	0,0658
30	0,1093	0,0746	0,0786	0,0786	0,104	0,1053	0,092	0,0773	0,0773	0,2103	0,0887	0,0848	0,1013	0,0826	0,0853	0,0853	0,1	0,0986	0,092	0,088	0,088	0,0831	0,0803	0,0782

Fold	Base 4																							
	3.000 registros																							
	Redes Neurais, 3 Neurônios, BPROP												Redes Neurais, 3 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0613	0,0626	0,0626	0,0626	0,0626	0,064	0,0613	0,0613	0,0613	0,0613	0,0613	0,0613	0,0653	0,0653	0,0653	0,0653	0,0626	0,0613	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653
2	0,076	0,076	0,076	0,076	0,0746	0,0786	0,076	0,076	0,076	0,076	0,076	0,076	0,0773	0,076	0,076	0,076	0,072	0,0733	0,0773	0,0773	0,0773	0,0773	0,0773	0,0773
3	0,0586	0,0586	0,0586	0,0586	0,056	0,0546	0,0586	0,0586	0,0586	0,0586	0,0586	0,0586	0,0546	0,0653	0,0653	0,0653	0,0546	0,0546	0,0546	0,0546	0,0546	0,0605	0,0577	0,0561
4	0,0613	0,06	0,0666	0,0666	0,0586	0,0546	0,0613	0,0613	0,0613	0,0613	0,0613	0,0613	0,0586	0,0613	0,0693	0,0693	0,056	0,0573	0,0586	0,0586	0,0586	0,0586	0,0586	0,0586
5	0,056	0,0626	0,0626	0,0626	0,0533	0,0546	0,0626	0,0626	0,0626	0,056	0,056	0,056	0,056	0,0613	0,0613	0,0613	0,06	0,0573	0,068	0,068	0,068	0,056	0,056	0,056
6	0,0653	0,0706	0,068	0,068	0,0693	0,068	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,068	0,0666	0,0666	0,0666	0,0693	0,0786	0,0733	0,084	0,084	0,068	0,068	0,068
7	0,0706	0,0693	0,072	0,072	0,0693	0,0693	0,0706	0,0706	0,0706	0,0611	0,0611	0,0611	0,0706	0,0773	0,0773	0,0773	0,0706	0,068	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706
8	0,0613	0,0706	0,0746	0,0746	0,064	0,064	0,0613	0,0613	0,0613	0,0613	0,0613	0,0613	0,0666	0,0706	0,072	0,072	0,0626	0,064	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666
9	0,0826	0,088	0,088	0,088	0,084	0,084	0,0826	0,0826	0,0826	0,0826	0,0826	0,0826	0,0893	0,0906	0,0906	0,0906	0,0866	0,0813	0,0893	0,0893	0,0893	0,0893	0,0893	0,0893
10	0,0813	0,0826	0,0853	0,0853	0,08	0,0786	0,0813	0,0813	0,0813	0,0813	0,0813	0,0813	0,072	0,084	0,084	0,084	0,0826	0,08	0,072	0,072	0,072	0,0829	0,0823	0,0820
11	0,076	0,0773	0,0853	0,0853	0,0706	0,0746	0,0786	0,0786	0,0786	0,0711	0,0711	0,0711	0,076	0,076	0,076	0,076	0,072	0,076	0,0813	0,0813	0,0813	0,0794	0,0756	0,0733
12	0,072	0,0693	0,0693	0,0693	0,0733	0,0746	0,0733	0,0733	0,0733	0,072	0,072	0,072	0,0706	0,0706	0,0706	0,0706	0,0733	0,072	0,0813	0,0813	0,0813	0,0706	0,0706	0,0706
13	0,0653	0,0773	0,084	0,084	0,072	0,072	0,0653	0,0653	0,0653	0,0681	0,0676	0,0673	0,072	0,072	0,072	0,072	0,0746	0,072	0,072	0,072	0,072	0,072	0,072	0,072
14	0,0573	0,0666	0,076	0,076	0,0573	0,056	0,0573	0,0573	0,0573	0,0573	0,0573	0,0573	0,0573	0,0653	0,0653	0,0653	0,0546	0,056	0,0573	0,0573	0,0573	0,0573	0,0573	0,0573
15	0,0626	0,064	0,0626	0,0626	0,0626	0,064	0,0626	0,0626	0,0626	0,0626	0,0626	0,0626	0,0613	0,0626	0,068	0,068	0,0613	0,0666	0,0613	0,0613	0,0613	0,0690	0,0690	0,0690
16	0,072	0,068	0,068	0,068	0,0693	0,0693	0,068	0,068	0,068	0,0822	0,0822	0,0822	0,0733	0,0706	0,0706	0,0706	0,0666	0,068	0,0666	0,0666	0,0666	0,0733	0,0733	0,0733
17	0,06	0,0573	0,068	0,068	0,0586	0,0613	0,06	0,06	0,06	0,0637	0,0630	0,0626	0,06	0,0666	0,0666	0,0666	0,0573	0,0573	0,06	0,06	0,06	0,06	0,06	0,06
18	0,048	0,052	0,0613	0,0613	0,0426	0,0453	0,052	0,052	0,052	0,0438	0,0438	0,0437	0,0466	0,0546	0,0546	0,0546	0,0466	0,048	0,0466	0,0466	0,0466	0,0466	0,0466	0,0466
19	0,0546	0,0546	0,0546	0,0546	0,0546	0,056	0,0506	0,0506	0,0506	0,0546	0,0546	0,0546	0,0546	0,06	0,0653	0,0653	0,0546	0,0573	0,0546	0,0546	0,0546	0,0546	0,0546	0,0546
20	0,068	0,076	0,076	0,076	0,068	0,0733	0,068	0,068	0,068	0,068	0,068	0,068	0,0746	0,0746	0,0746	0,0746	0,0733	0,0706	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746
21	0,0746	0,0773	0,08	0,08	0,0773	0,08	0,0813	0,0773	0,0773	0,0658	0,0611	0,0583	0,0746	0,0746	0,0746	0,0746	0,076	0,0786	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746
22	0,0613	0,0693	0,0693	0,0693	0,06	0,06	0,0613	0,0613	0,0613	0,0613	0,0613	0,0613	0,06	0,0693	0,0693	0,0693	0,06	0,0626	0,06	0,06	0,06	0,06	0,06	0,06
23	0,0693	0,076	0,0906	0,0906	0,0693	0,0706	0,076	0,076	0,076	0,0693	0,0693	0,0693	0,068	0,0706	0,0706	0,0706	0,068	0,0693	0,0746	0,0746	0,0746	0,0675	0,0674	0,0673
24	0,0693	0,0773	0,0773	0,0773	0,076	0,076	0,0693	0,0693	0,0693	0,0693	0,0693	0,0693	0,0746	0,084	0,084	0,084	0,0786	0,0773	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746
25	0,06	0,056	0,0666	0,0666	0,0573	0,0546	0,06	0,06	0,06	0,06	0,06	0,06	0,0613	0,0666	0,0666	0,0666	0,056	0,056	0,064	0,0786	0,0786	0,0613	0,0613	0,0613
26	0,0626	0,0653	0,0653	0,0653	0,0573	0,06	0,0626	0,0626	0,0626	0,0626	0,0626	0,0626	0,0653	0,0653	0,0653	0,0653	0,0586	0,06	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653
27	0,06	0,06	0,06	0,06	0,0626	0,0626	0,06	0,06	0,06	0,06	0,06	0,06	0,06	0,0706	0,0706	0,0706	0,0546	0,0546	0,06	0,06	0,06	0,06	0,06	0,06
28	0,0626	0,064	0,0666	0,0666	0,0613	0,0613	0,0626	0,0626	0,0626	0,0688	0,0688	0,0688	0,0653	0,0693	0,0693	0,0693	0,0613	0,0613	0,0653	0,0653	0,0653	0,0738	0,0728	0,0723
29	0,0573	0,0573	0,0573	0,0573	0,0533	0,052	0,0573	0,0573	0,0573	0,0569	0,0543	0,0528	0,0573	0,056	0,056	0,056	0,0533	0,0493	0,0573	0,0573	0,0573	0,0573	0,0573	0,0573
30	0,0786	0,0786	0,072	0,072	0,076	0,0733	0,0746	0,0746	0,0746	0,0689	0,0670	0,0659	0,0773	0,076	0,0733	0,0733	0,076	0,0746	0,076	0,0786	0,0786	0,0773	0,0773	0,0773

Tabela A.22 – Erros de Classificação da Base (4) para 3.000 registros (continuação).

Fold	Base 4																							
	3.000 registros																							
	Redes Neurais, 10 Neurônios, BPROP												Redes Neurais, 10 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,056	0,056	0,056	0,056	0,0666	0,068	0,056	0,056	0,056	0,056	0,056	0,056	0,0570	0,0570	0,0570	0,0570	0,0666	0,0786	0,0906	0,0866	0,0866	0,0759	0,0746	0,0739
2	0,072	0,072	0,072	0,072	0,072	0,0733	0,072	0,072	0,072	0,072	0,072	0,072	0,0688	0,0688	0,0688	0,0688	0,068	0,072	0,0688	0,0688	0,0688	0,0800	0,0732	0,0693
3	0,064	0,068	0,068	0,068	0,0573	0,056	0,064	0,064	0,064	0,064	0,064	0,064	0,0769	0,0676	0,0676	0,0676	0,0586	0,06	0,0853	0,0853	0,0853	0,0731	0,0669	0,0633
4	0,0666	0,076	0,088	0,088	0,052	0,0546	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666	0,0796	0,0756	0,0893	0,0893	0,0613	0,0546	0,116	0,116	0,116	0,0864	0,0845	0,0834
5	0,06	0,06	0,06	0,06	0,0586	0,0546	0,06	0,06	0,06	0,0673	0,0665	0,0659	0,0719	0,0719	0,0719	0,0719	0,0653	0,068	0,0746	0,0746	0,0746	0,0909	0,0837	0,0795
6	0,0666	0,072	0,0933	0,0933	0,0653	0,068	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666	0,0785	0,0734	0,1012	0,1012	0,0693	0,0733	0,0785	0,0785	0,0785	0,0878	0,0829	0,0800
7	0,0746	0,0746	0,0746	0,0746	0,0693	0,0653	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746	0,0719	0,0719	0,0719	0,0719	0,0733	0,0653	0,108	0,08	0,08	0,0751	0,0741	0,0736
8	0,076	0,076	0,076	0,076	0,0746	0,0706	0,076	0,076	0,076	0,0776	0,0759	0,0749	0,0848	0,0848	0,0848	0,0848	0,0626	0,0573	0,1013	0,1013	0,1013	0,0851	0,0832	0,0821
9	0,084	0,0986	0,0986	0,0986	0,0853	0,084	0,084	0,084	0,084	0,0868	0,0869	0,0870	0,0842	0,1012	0,1012	0,1012	0,0853	0,0826	0,0842	0,0842	0,0842	0,1022	0,1001	0,0989
10	0,0786	0,0786	0,0786	0,0786	0,0853	0,08	0,0786	0,0786	0,0786	0,0786	0,0786	0,0786	0,0710	0,0710	0,0710	0,0710	0,0893	0,0893	0,0710	0,0710	0,0710	0,1119	0,1125	0,1129
11	0,088	0,0906	0,0906	0,0906	0,0773	0,0746	0,088	0,088	0,088	0,0761	0,0731	0,0714	0,0804	0,0911	0,0911	0,0911	0,072	0,076	0,0973	0,0973	0,0973	0,0895	0,0829	0,0791
12	0,0693	0,072	0,072	0,072	0,0706	0,0746	0,0826	0,092	0,092	0,0693	0,0693	0,0693	0,0718	0,0872	0,0872	0,0872	0,0746	0,0693	0,096	0,0986	0,0986	0,0791	0,0745	0,0718
13	0,0653	0,0653	0,0653	0,0653	0,0746	0,0706	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,0549	0,0549	0,0549	0,0549	0,068	0,076	0,0549	0,0549	0,0549	0,0824	0,0762	0,0726
14	0,056	0,0626	0,068	0,068	0,0586	0,056	0,072	0,072	0,072	0,056	0,056	0,056	0,0626	0,0597	0,0615	0,0615	0,0533	0,056	0,0626	0,0626	0,0626	0,0626	0,0626	0,0626
15	0,0653	0,076	0,076	0,076	0,0653	0,064	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,0673	0,0653	0,0653	0,0653	0,064	0,068	0,0673	0,0673	0,0673	0,0817	0,0788	0,0771
16	0,08	0,0733	0,0733	0,0733	0,0706	0,068	0,08	0,08	0,08	0,08	0,08	0,08	0,0695	0,0674	0,0674	0,0674	0,0693	0,0693	0,0695	0,0695	0,0695	0,0974	0,0975	0,0975
17	0,06	0,068	0,068	0,068	0,06	0,0533	0,0666	0,0666	0,0666	0,0629	0,0636	0,0640	0,0651	0,0691	0,0691	0,0691	0,0533	0,0573	0,0866	0,0866	0,0866	0,0786	0,0714	0,0673
18	0,0493	0,0493	0,0493	0,0493	0,048	0,0453	0,0493	0,0493	0,0493	0,0493	0,0493	0,0493	0,0544	0,0544	0,0544	0,0544	0,048	0,0493	0,0933	0,0933	0,0933	0,0769	0,0678	0,0626
19	0,06	0,0666	0,0746	0,0746	0,056	0,0546	0,06	0,06	0,06	0,06	0,06	0,06	0,0555	0,0560	0,0772	0,0772	0,052	0,0533	0,0555	0,0555	0,0555	0,0807	0,0743	0,0706
20	0,076	0,076	0,076	0,076	0,076	0,0733	0,076	0,076	0,076	0,076	0,076	0,076	0,0715	0,0715	0,0715	0,0715	0,0813	0,0786	0,0715	0,0715	0,0715	0,0986	0,0961	0,0946
21	0,076	0,0853	0,0853	0,0853	0,08	0,072	0,076	0,076	0,076	0,076	0,076	0,076	0,0762	0,0947	0,0947	0,0947	0,0813	0,0773	0,0933	0,0866	0,0866	0,0869	0,0827	0,0802
22	0,0626	0,0626	0,0626	0,0626	0,056	0,0573	0,0626	0,0626	0,0626	0,0626	0,0626	0,0626	0,0621	0,0621	0,0621	0,0621	0,0613	0,06	0,0906	0,0906	0,0906	0,0905	0,0860	0,0834
23	0,0666	0,0653	0,0746	0,0746	0,0706	0,0693	0,0666	0,0666	0,0666	0,0684	0,0691	0,0695	0,0652	0,0709	0,0815	0,0815	0,0706	0,0746	0,0652	0,0652	0,0652	0,0708	0,0697	0,0691
24	0,084	0,0786	0,0786	0,0786	0,0786	0,0826	0,084	0,084	0,084	0,084	0,084	0,084	0,0919	0,0745	0,0745	0,0745	0,08	0,0866	0,0919	0,0919	0,0919	0,0919	0,0919	0,0919
25	0,0666	0,0666	0,0666	0,0666	0,0533	0,0573	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666	0,0746	0,0746	0,0746	0,0746	0,056	0,0533	0,0786	0,0786	0,0786	0,0866	0,0812	0,0780
26	0,0733	0,0733	0,0733	0,0733	0,06	0,0586	0,0733	0,0733	0,0733	0,0701	0,0692	0,0687	0,0716	0,0716	0,0716	0,0716	0,068	0,0666	0,0716	0,0716	0,0716	0,0821	0,0774	0,0747
27	0,0666	0,0666	0,0666	0,0666	0,0626	0,0573	0,0666	0,0666	0,0666	0,0666	0,0666	0,0666	0,0753	0,0753	0,0753	0,0753	0,0546	0,056	0,0753	0,0753	0,0753	0,0791	0,0732	0,0698
28	0,068	0,068	0,068	0,068	0,0613	0,0653	0,068	0,068	0,068	0,068	0,068	0,068	0,0655	0,0655	0,0655	0,0655	0,0653	0,0693	0,0655	0,0655	0,0655	0,0863	0,0825	0,0803
29	0,056	0,056	0,056	0,056	0,0533	0,0533	0,056	0,056	0,056	0,056	0,056	0,056	0,0632	0,0632	0,0632	0,0632	0,0546	0,06	0,0632	0,0632	0,0632	0,0745	0,0702	0,0677
30	0,0653	0,076	0,076	0,076	0,072	0,0693	0,0706	0,0706	0,0706	0,0653	0,0653	0,0653	0,0629	0,0825	0,0825	0,0825	0,0746	0,0733	0,0629	0,0629	0,0629	0,0839	0,0809	0,0791

Fold	Base 4																							
	3.000 registros																							
	Redes Neurais, 20 Neurônios, BPROP												Redes Neurais, 20 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0693	0,0693	0,0693	0,0693	0,0793	0,0693	0,0693	0,0693	0,0693	0,0693	0,0693	0,0693	0,0712	0,0712	0,0712	0,0712	0,0649	0,0636	0,0712	0,0712	0,0712	0,0712	0,0712	0,0712
2	0,0773	0,076	0,076	0,076	0,0657	0,0786	0,0773	0,0773	0,0773	0,0741	0,0714	0,0698	0,0955	0,0659	0,0659	0,0659	0,0709	0,0798	0,0955	0,0955	0,0955	0,0810	0,0800	0,0619
3	0,064	0,064	0,064	0,064	0,0671	0,0586	0,0613	0,072	0,072	0,0639	0,0614	0,0599	0,0681	0,0681	0,0681	0,0681	0,0616	0,0646	0,0537	0,0659	0,0659	0,0619	0,0654	0,0541
4	0,0693	0,068	0,0813	0,0813	0,0485	0,0613	0,0746	0,0746	0,0746	0,0693	0,0693	0,0693	0,0683	0,0662	0,0922	0,0922	0,0504	0,0575	0,0714	0,0714	0,0714	0,0683	0,0683	0,0683
5	0,068	0,0786	0,0786	0,0786	0,0587	0,06	0,068	0,068	0,068	0,0623	0,0597	0,0582	0,0807	0,0840	0,0840	0,0840	0,0688	0,0520	0,0807	0,0807	0,0807	0,0676	0,0559	0,0582
6	0,0706	0,0893	0,0853	0,0853	0,0717	0,068	0,0706	0,0706	0,0706	0,0706	0,0706	0,0706	0,0748	0,0906	0,0847	0,0847	0,0728	0,0710	0,0748	0,0748	0,0748	0,0748	0,0748	0,0748
7	0,0746	0,0746	0,0746	0,0746	0,0711	0,0733	0,0746	0,0746	0,0746	0,0746	0,0746	0,0746	0,0665	0,0665	0,0665	0,0665	0,0630	0,0774	0,0665	0,0665	0,0665	0,0665	0,0665	0,0665
8	0,0773	0,0773	0,0773	0,0773	0,0654	0,0773	0,0773	0,0773	0,0773	0,0773	0,0773	0,0773	0,0731	0,0731	0,0731	0,0731	0,0653	0,0769	0,0731	0,0731	0,0731	0,0731	0,0731	0,0731
9	0,0906	0,0906	0,0906	0,0906	0,0761	0,088	0,1013	0,1013	0,1013	0,0860	0,0857	0,0855	0,0906	0,0906	0,0906	0,0906	0,0779	0,0794	0,0967	0,0967	0,0967	0,0864	0,0927	0,0925
10	0,0933	0,0973	0,0973	0,0973	0,0907	0,0853	0,0933	0,0933	0,0933	0,0940	0,0943	0,0945	0,0925	0,0974	0,0974	0,0974	0,0887	0,0879	0,0925	0,0925	0,0925	0,0872	0,0908	0,0923
11	0,0866	0,0906	0,0906	0,0906	0,0751	0,0746	0,0866	0,0866	0,0866	0,0753	0,0725	0,0709	0,0856	0,0887	0,0887	0,0887	0,0790	0,0757	0,0856	0,0856	0,0856	0,0686	0,0810	0,0712
12	0,0653	0,0653	0,0653	0,0653	0,0685	0,072	0,0653	0,0653	0,0653	0,0653	0,0653	0,0653	0,0542	0,0542	0,0542	0,0542	0,0688	0,0737	0,0542	0,0542	0,0542	0,0542	0,0542	0,0542
13	0,068	0,0853	0,0853	0,0853	0,0721	0,0853	0,068	0,068	0,068	0,0723	0,0682	0,0658	0,0638	0,0898	0,0898	0,0898	0,0622	0,0867	0,0638	0,0638	0,0638	0,0723	0,0700	0,0667
14	0,0653	0,0746	0,0626	0,0626	0,0537	0,0653	0,0826	0,0813	0,0813	0,0547	0,0547	0,0547	0,0788	0,0805	0,0718	0,0718	0,0591	0,0716	0,0793	0,0829	0,0829	0,0646	0,0646	0,0646
15	0,0733	0,068	0,0786	0,0786	0,0599	0,0746	0,0733	0,0733	0,0733	0,0699	0,0683	0,0674	0,0758	0,0688	0,0744	0,0744	0,0645	0,0759	0,0758	0,0758	0,0758	0,0683	0,0676	0,0617
16	0,08	0,084	0,084	0,084	0,0652	0,0706	0,08	0,08	0,08	0,08	0,08	0,08	0,0902	0,0776	0,0776	0,0776	0,0651	0,0700	0,0902	0,0902	0,0902	0,0902	0,0902	0,0902
17	0,068	0,0706	0,072	0,072	0,0749	0,0693	0,068	0,068	0,068	0,0670	0,0636	0,0616	0,0651	0,0770	0,0699	0,0699	0,0524	0,0684	0,0651	0,0651	0,0651	0,0700	0,0651	0,0581
18	0,064	0,0733	0,0866	0,0866	0,0464	0,0533	0,0733	0,0733	0,0733	0,064	0,064	0,064	0,0650	0,0699	0,0827	0,0827	0,0516	0,0597	0,0667	0,0667	0,0667	0,0650	0,0650	0,0650
19	0,068	0,0746	0,0746	0,0746	0,0544	0,056	0,068	0,068	0,068	0,068	0,068	0,068	0,0578	0,0703	0,0703	0,0703	0,0463	0,0539	0,0578	0,0578	0,0578	0,0578	0,0578	0,0578
20	0,084	0,096	0,096	0,096	0,0751	0,0733	0,084	0,084	0,084	0,084	0,084	0,084	0,0762	0,0872	0,0872	0,0872	0,0756	0,0827	0,0762	0,0762	0,0762	0,0762	0,0762	0,0762
21	0,0786	0,0813	0,0813	0,0813	0,0883	0,072	0,0786	0,0786	0,0786	0,0776	0,0734	0,0710	0,0838	0,0805	0,0805	0,0805	0,0864	0,0816	0,0838	0,0838	0,0838	0,0858	0,0748	0,0635
22	0,068	0,08	0,08	0,08	0,0531	0,0586	0,068	0,068	0,068	0,0742	0,0727	0,0718	0,0632	0,0846	0,0846	0,0846	0,0451	0,0516	0,0632	0,0632	0,0632	0,0816	0,0751	0,0672
23	0,0666	0,0773	0,0773	0,0773	0,0678	0,0666	0,0666	0,0666	0,0666	0,0675	0,0674	0,0673	0,0686	0,0784	0,0784	0,0784	0,0596	0,0728	0,0686	0,0686	0,0686	0,0655	0,0744	0,0633
24	0,0866	0,0933	0,0933	0,0933	0,0728	0,0773	0,0893	0,0893	0,0893	0,0866	0,0866	0,0866	0,0778	0,0894	0,0894	0,0894	0,0798	0,0744	0,0911	0,0911	0,0911	0,0778	0,0778	0,0778
25	0,0706	0,0706	0,0706	0,0706	0,0475	0,0626	0,068	0,068	0,068	0,0706	0,0706	0,0706	0,0604	0,0604	0,0604	0,0604	0,0569	0,0587	0,0641	0,0641	0,0641	0,0604	0,0604	0,0604
26	0,0773	0,0773	0,0773	0,0773	0,0642	0,0613	0,0773	0,0773	0,0773	0,0735	0,0711	0,0697	0,0758	0,0758	0,0758	0,0758	0,0682	0,0579	0,0758	0,0758	0,0758	0,0699	0,0703	0,0773
27	0,0666	0,08	0,08	0,08	0,0546	0,0613	0,0666	0,0666	0,0666	0,0673	0,0665	0,0659	0,0615	0,0720	0,0720	0,0720	0,0548	0,0568	0,0615	0,0615	0,0615	0,0634	0,0582	0,0670
28	0,0693	0,0733	0,0733	0,0733	0,0540	0,0693	0,0786	0,0746	0,0746	0,0693	0,0693	0,0693	0,0799	0,0810	0,0810	0,0810	0,0791	0,0658	0,0776	0,0675	0,0675	0,0799	0,0799	0,0799
29	0,0613	0,0613	0,0613	0,0613	0,0568	0,0506	0,0613	0,0613	0,0613	0,0613	0,0613	0,0613	0,0673	0,0645	0,0645	0,0645	0,0468	0,0479	0,0673	0,0673	0,0673	0,0673	0,0673	0,0673
30	0,0786	0,0813	0,0813	0,0813	0,0755	0,072	0,0786	0,0786	0,0786	0,0781	0,0754	0,0739	0,0638	0,0865	0,0865	0,0865	0,0722	0,0777	0,0638	0,0638	0,0638	0,0746	0,0793	0,0761

Tabela A.23 – Erros de Classificação da Base (4) para 5.000 registros.

Fold	Base 4																							
	5.000 registros																							
	Regressão Linear												Regressão Logística											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0984	0,0752	0,0632	0,0632	0,0976	0,0984	0,0632	0,0592	0,0592	0,1893	0,0608	0,0601	0,0744	0,0584	0,0496	0,0496	0,0744	0,0744	0,0624	0,0592	0,0592	0,0744	0,0744	0,0744
2	0,1056	0,0744	0,0744	0,0744	0,1136	0,1096	0,0784	0,0808	0,0808	0,2477	0,0788	0,0771	0,0784	0,0672	0,0728	0,0728	0,0792	0,08	0,08	0,08	0,08	0,0751	0,0733	0,0733
3	0,1592	0,0608	0,0632	0,0632	0,16	0,16	0,0576	0,0616	0,0616	0,2575	0,0632	0,0609	0,0696	0,0544	0,0632	0,0632	0,0696	0,0696	0,0624	0,0608	0,0608	0,0576	0,0536	0,0533
4	0,1184	0,0672	0,0632	0,0632	0,1192	0,12	0,072	0,0752	0,0752	0,1800	0,0807	0,0773	0,0792	0,0656	0,0648	0,0648	0,0768	0,0768	0,0704	0,0704	0,0704	0,0792	0,0792	0,0792
5	0,1552	0,0784	0,0752	0,0752	0,1536	0,1536	0,0792	0,0824	0,0824	0,1914	0,0745	0,0700	0,0848	0,0736	0,0712	0,0712	0,0816	0,0808	0,0776	0,08	0,08	0,0762	0,0690	0,0656
6	0,1344	0,0728	0,0736	0,0736	0,136	0,1352	0,0832	0,088	0,088	0,2045	0,0811	0,0805	0,0864	0,0752	0,0752	0,0752	0,0888	0,0912	0,0848	0,0824	0,0824	0,0768	0,0772	0,0772
7	0,1272	0,072	0,0744	0,0744	0,12	0,1176	0,0728	0,076	0,076	0,2061	0,0755	0,0742	0,0728	0,0656	0,0664	0,0664	0,0752	0,0752	0,072	0,072	0,072	0,0721	0,0718	0,0713
8	0,0968	0,0824	0,08	0,08	0,0968	0,096	0,0832	0,084	0,084	0,2497	0,0863	0,0835	0,088	0,084	0,08	0,08	0,088	0,088	0,088	0,088	0,088	0,0870	0,0837	0,0815
9	0,104	0,0824	0,0736	0,0736	0,1048	0,1048	0,084	0,0888	0,0888	0,2168	0,0804	0,0776	0,0864	0,0816	0,072	0,072	0,088	0,0896	0,0848	0,0848	0,0848	0,0727	0,0727	0,0727
10	0,1456	0,084	0,0808	0,0808	0,1496	0,148	0,084	0,0824	0,0824	0,2326	0,0852	0,0834	0,0808	0,0776	0,0776	0,0776	0,08	0,0784	0,0808	0,0808	0,0808	0,0807	0,0811	0,0801
11	0,0984	0,0704	0,0728	0,0728	0,0976	0,0952	0,0784	0,076	0,076	0,2235	0,0809	0,0801	0,088	0,072	0,0744	0,0744	0,0864	0,0872	0,088	0,088	0,088	0,0809	0,0802	0,0802
12	0,1416	0,0776	0,08	0,08	0,1424	0,1416	0,088	0,088	0,088	0,2337	0,0820	0,0807	0,0872	0,0752	0,0784	0,0784	0,0896	0,0904	0,084	0,084	0,084	0,0775	0,0772	0,0769
13	0,1272	0,0824	0,0808	0,0808	0,1248	0,1256	0,0792	0,0744	0,0744	0,2075	0,0744	0,0708	0,0784	0,076	0,0744	0,0744	0,0792	0,0784	0,0784	0,0784	0,0784	0,0711	0,0711	0,0711
14	0,104	0,0768	0,0752	0,0752	0,1024	0,1016	0,0704	0,0704	0,0704	0,2070	0,0749	0,0709	0,0736	0,0672	0,0656	0,0656	0,072	0,072	0,0696	0,0696	0,0696	0,0656	0,0577	0,0573
15	0,1368	0,0784	0,0744	0,0744	0,1368	0,1368	0,084	0,0824	0,0824	0,2619	0,0702	0,0694	0,088	0,0736	0,0752	0,0752	0,0896	0,0904	0,0824	0,0744	0,0744	0,0750	0,0750	0,0750
16	0,0824	0,056	0,0544	0,0544	0,0848	0,0832	0,064	0,0608	0,0608	0,1909	0,0644	0,0630	0,0776	0,0568	0,0592	0,0592	0,08	0,0792	0,0728	0,0656	0,0656	0,0663	0,0629	0,0618
17	0,104	0,0704	0,072	0,072	0,1088	0,1088	0,072	0,0704	0,0704	0,2332	0,0779	0,0755	0,08	0,0672	0,0648	0,0648	0,0784	0,0768	0,08	0,08	0,08	0,08	0,08	0,08
18	0,1272	0,0896	0,084	0,084	0,1248	0,1264	0,0912	0,0952	0,0952	0,2506	0,0956	0,0925	0,0888	0,0704	0,0664	0,0664	0,088	0,0888	0,0752	0,076	0,076	0,0818	0,0818	0,0818
19	0,1128	0,0752	0,0768	0,0768	0,1136	0,1128	0,0832	0,076	0,076	0,2060	0,0836	0,0824	0,0832	0,0744	0,072	0,072	0,084	0,0856	0,0808	0,0776	0,0776	0,0771	0,0771	0,0771
20	0,1128	0,08	0,0744	0,0744	0,1152	0,116	0,0648	0,0656	0,0656	0,2454	0,0785	0,0766	0,076	0,0808	0,0824	0,0824	0,0752	0,0752	0,0632	0,0632	0,0632	0,0701	0,0699	0,0699
21	0,1096	0,0656	0,068	0,068	0,1128	0,112	0,0872	0,0776	0,0776	0,2307	0,0717	0,0693	0,0896	0,0696	0,0672	0,0672	0,0904	0,0904	0,0888	0,0792	0,0792	0,0728	0,0687	0,0669
22	0,14	0,0688	0,0656	0,0656	0,1392	0,1352	0,0672	0,0688	0,0688	0,2234	0,0724	0,0701	0,0648	0,0632	0,0624	0,0624	0,0696	0,0704	0,0672	0,0696	0,0696	0,0631	0,0628	0,0626
23	0,0992	0,0784	0,0728	0,0728	0,1024	0,1024	0,0776	0,0704	0,0704	0,2061	0,0792	0,0771	0,0808	0,0792	0,076	0,076	0,08	0,0808	0,0808	0,0808	0,0808	0,0730	0,0730	0,0730
24	0,144	0,0728	0,0712	0,0712	0,1424	0,1408	0,072	0,0688	0,0688	0,2062	0,0866	0,0846	0,0744	0,0648	0,0656	0,0656	0,0736	0,0736	0,0712	0,0712	0,0712	0,0744	0,0744	0,0744
25	0,1168	0,084	0,0792	0,0792	0,1144	0,1144	0,092	0,0856	0,0856	0,2088	0,0992	0,0970	0,0888	0,0848	0,0792	0,0792	0,0856	0,0864	0,0904	0,0888	0,0888	0,0839	0,0839	0,0839
26	0,1072	0,0768	0,076	0,076	0,112	0,1112	0,0816	0,08	0,08	0,2272	0,0787	0,0758	0,0864	0,0752	0,0696	0,0696	0,0856	0,0856	0,0784	0,0792	0,0792	0,0784	0,0726	0,0710
27	0,1056	0,0616	0,0616	0,0616	0,1064	0,1072	0,0624	0,068	0,068	0,2100	0,0649	0,0625	0,0728	0,06	0,0608	0,0608	0,0712	0,072	0,0616	0,0688	0,0688	0,0591	0,0587	0,0576
28	0,124	0,0704	0,0712	0,0712	0,1232	0,1232	0,0792	0,0704	0,0704	0,2388	0,0751	0,0717	0,0784	0,0664	0,064	0,064	0,0776	0,0768	0,0792	0,072	0,072	0,0784	0,0784	0,0784
29	0,1112	0,0688	0,072	0,072	0,1104	0,1096	0,0728	0,0688	0,0688	0,2307	0,0859	0,0829	0,0736	0,0688	0,0704	0,0704	0,0752	0,0752	0,0752	0,0688	0,0688	0,0748	0,0717	0,0717
30	0,1144	0,0704	0,0656	0,0656	0,1112	0,1112	0,0784	0,08	0,08	0,1998	0,0748	0,0704	0,08	0,0688	0,0656	0,0656	0,0784	0,0784	0,0768	0,0768	0,0768	0,0704	0,0704	0,0704

Fold	Base 4																							
	5.000 registros																							
	Redes Neurais, 3 Neurônios, BPROP												Redes Neurais, 3 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0464	0,0488	0,0512	0,0512	0,0472	0,0472	0,0464	0,0464	0,0464	0,0449	0,0431	0,0421	0,0456	0,0496	0,052	0,052	0,0472	0,0496	0,0552	0,0552	0,0552	0,0456	0,0456	0,0456
2	0,0656	0,0704	0,0704	0,0704	0,0664	0,0656	0,0656	0,0656	0,0656	0,0691	0,0691	0,0691	0,0664	0,0664	0,0664	0,0664	0,0672	0,0648	0,0664	0,0664	0,0664	0,0664	0,0664	0,0664
3	0,0536	0,0584	0,0576	0,0576	0,0528	0,0496	0,0536	0,0536	0,0536	0,0536	0,0536	0,0536	0,052	0,06	0,0576	0,0576	0,052	0,0504	0,0632	0,0632	0,0632	0,0546	0,0521	0,0507
4	0,0632	0,0632	0,0632	0,0632	0,0672	0,0656	0,0632	0,0632	0,0632	0,0632	0,0632	0,0632	0,064	0,0712	0,0712	0,0712	0,0664	0,0672	0,064	0,0672	0,0672	0,064	0,064	0,064
5	0,0712	0,0712	0,072	0,072	0,0664	0,0656	0,0712	0,0712	0,0712	0,0651	0,0619	0,0601	0,068	0,0744	0,0728	0,0728	0,0648	0,0696	0,0784	0,0784	0,0784	0,068	0,068	0,068
6	0,0688	0,0656	0,0656	0,0656	0,0656	0,0672	0,0688	0,0688	0,0688	0,0665	0,0651	0,0643	0,0704	0,0704	0,0704	0,0704	0,0672	0,0664	0,0704	0,0704	0,0704	0,0694	0,0694	0,0694
7	0,0656	0,0648	0,068	0,068	0,064	0,0608	0,0712	0,0712	0,0712	0,0656	0,0656	0,0656	0,0672	0,0672	0,0672	0,0672	0,0656	0,0592	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672
8	0,06	0,0672	0,0672	0,0672	0,0632	0,064	0,0648	0,0704	0,0704	0,0630	0,0633	0,0635	0,0608	0,068	0,072	0,072	0,0632	0,0624	0,0608	0,0608	0,0608	0,0620	0,0620	0,0620
9	0,068	0,0664	0,0664	0,0664	0,0632	0,064	0,068	0,068	0,068	0,0657	0,0647	0,0641	0,0696	0,0656	0,0696	0,0696	0,0632	0,0648	0,0712	0,0712	0,0712	0,0696	0,0696	0,0696
10	0,0688	0,0688	0,0688	0,0688	0,0688	0,0656	0,0688	0,0688	0,0688	0,0688	0,0688	0,0688	0,0672	0,0656	0,0656	0,0656	0,0648	0,0664	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672
11	0,0576	0,0616	0,0616	0,0616	0,0624	0,0632	0,0576	0,0576	0,0576	0,0576	0,0576	0,0576	0,0584	0,068	0,0672	0,0672	0,0632	0,0616	0,0584	0,0584	0,0584	0,0584	0,0584	0,0584
12	0,0688	0,0712	0,0728	0,0728	0,0696	0,0712	0,0688	0,0688	0,0688	0,0688	0,0688	0,0688	0,0696	0,0696	0,0696	0,0696	0,0712	0,0728	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696
13	0,0656	0,064	0,064	0,064	0,0616	0,0624	0,0656	0,0656	0,0656	0,0576	0,0547	0,0530	0,064	0,0656	0,0624	0,0624	0,0664	0,0632	0,064	0,064	0,064	0,064	0,064	0,064
14	0,0616	0,0696	0,0696	0,0696	0,06	0,0608	0,0616	0,0616	0,0616	0,0616	0,0616	0,0616	0,0624	0,0696	0,0696	0,0696	0,06	0,0592	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624
15	0,0688	0,0712	0,0744	0,0744	0,0648	0,0656	0,0688	0,0688	0,0688	0,0688	0,0688	0,0688	0,0696	0,0728	0,0656	0,0656	0,0656	0,0664	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696
16	0,0496	0,0552	0,0584	0,0584	0,048	0,056	0,0512	0,0512	0,0512	0,0496	0,0496	0,0496	0,0528	0,0544	0,0624	0,0624	0,0512	0,056	0,0528	0,0528	0,0528	0,0528	0,0528	0,0528
17	0,06	0,0584	0,0584	0,0584	0,056	0,056	0,0624	0,0624	0,0624	0,06	0,06	0,06	0,0584	0,0552	0,0552	0,0552	0,056	0,0544	0,0584	0,0584	0,0584	0,0584	0,0584	0,0584
18	0,0624	0,0624	0,0624	0,0624	0,0608	0,06	0,0624	0,0624	0,0624	0,0624	0,0624	0,0624	0,0632	0,0656	0,0672	0,0672	0,0624	0,0632	0,0632	0,0632	0,0632	0,0632	0,0632	0,0632
19	0,0696	0,0728	0,0712	0,0712	0,0704	0,072	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696	0,0672	0,0736	0,0784	0,0784	0,072	0,0696	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672
20	0,06	0,0664	0,0664	0,0664	0,056	0,056	0,0648	0,0648	0,0648	0,06	0,06	0,06	0,0576	0,0632	0,0632	0,0632	0,0552	0,0568	0,0656	0,0656	0,0656	0,0576	0,0576	0,0576
21	0,0672	0,0672	0,0672	0,0672	0,0624	0,0608	0,068	0,068	0,068	0,0672	0,0672	0,0672	0,0648	0,0648	0,0648	0,0648	0,0624	0,0624	0,0632	0,0632	0,0632	0,0648	0,0648	0,0648
22	0,0568	0,064	0,0648	0,0648	0,0584	0,0584	0,0568	0,0568	0,0568	0,0568	0,0568	0,0568	0,06	0,056	0,056	0,056	0,0592	0,056	0,068	0,068	0,068	0,06	0,06	0,06
23	0,0576	0,0584	0,0584	0,0584	0,0552	0,0544	0,056	0,056	0,056	0,0576	0,0576	0,0576	0,0576	0,0648	0,0648	0,0648	0,0552	0,0552	0,0656	0,0656	0,0656	0,0576	0,0576	0,0576
24	0,0624	0,0624	0,0624	0,0624	0,0608	0,0608	0,0608	0,0608	0,0608	0,0677	0,0677	0,0677	0,064	0,0616	0,0616	0,0616	0,0624	0,0624	0,064	0,064	0,064	0,064	0,064	0,064
25	0,0808	0,076	0,0688	0,0688	0,0696	0,0704	0,0808	0,0808	0,0808	0,0808	0,0808	0,0808	0,08	0,0752	0,0752	0,0752	0,072	0,0728	0,08	0,08	0,08	0,0804	0,0813	0,0819
26	0,0728	0,068	0,068	0,068	0,0696	0,0696	0,072	0,072	0,072	0,0728	0,0728	0,0728	0,0696	0,0688	0,072	0,072	0,0696	0,072	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696
27	0,0568	0,0592	0,0592	0,0592	0,0552	0,0568	0,0568	0,0568	0,0568	0,0568	0,0568	0,0568	0,0576	0,0584	0,0584	0,0584	0,0552	0,0576	0,0608	0,0608	0,0608	0,0576	0,0576	0,0576
28	0,0608	0,0624	0,0624	0,0624	0,0632	0,0592	0,0608	0,0608	0,0608	0,0608	0,0608	0,0608	0,0624	0,0608	0,0616	0,0616	0,0616	0,0608	0,0656	0,0656	0,0656	0,0624	0,0624	0,0624
29	0,0704	0,0728	0,08	0,08	0,068	0,0672	0,0704	0,0704	0,0704	0,0704	0,0704	0,0704	0,0712	0,076	0,0792	0,0792	0,0648	0,0664	0,0712	0,0712	0,0712	0,0712	0,0712	0,0712
30	0,0648	0,0648	0,0648	0,0648	0,0592	0,0576	0,0608	0,0576	0,0576	0,0648	0,0648	0,0648	0,0624	0,0616	0,0616	0,0616	0,06	0,0568	0,0616	0,0616	0,0616	0,0594	0,0588	0,0584

Tabela A.24 – Erros de Classificação da Base (4) para 5.000 registros (continuação).

Fold	Base 4																							
	5.000 registros																							
	Redes Neurais, 10 Neurônios, BPROP												Redes Neurais, 10 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0504	0,0552	0,0552	0,0552	0,0504	0,048	0,0504	0,0504	0,0504	0,0504	0,0504	0,0504	0,0610	0,0549	0,0549	0,0549	0,0523	0,0528	0,0610	0,0610	0,0610	0,0610	0,0610	0,0610
2	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696	0,0696	0,0741	0,0733	0,0728	0,0797	0,0797	0,0797	0,0797	0,0742	0,0808	0,0797	0,0797	0,0797	0,0624	0,0762	0,0655
3	0,0584	0,0584	0,0584	0,0584	0,052	0,052	0,0608	0,0608	0,0608	0,0561	0,0540	0,0528	0,0656	0,0656	0,0656	0,0656	0,0486	0,0648	0,0656	0,0656	0,0656	0,0599	0,0584	0,0418
4	0,0648	0,0648	0,0648	0,0648	0,0696	0,0712	0,0672	0,0712	0,0712	0,0648	0,0648	0,0648	0,0556	0,0556	0,0556	0,0556	0,0811	0,0768	0,0556	0,0556	0,0556	0,0556	0,0556	0,0556
5	0,0744	0,0744	0,0744	0,0744	0,0688	0,0688	0,0744	0,0744	0,0744	0,0744	0,0744	0,0744	0,0735	0,0735	0,0735	0,0735	0,0766	0,0752	0,0735	0,0735	0,0735	0,0735	0,0735	0,0735
6	0,064	0,0712	0,0712	0,0712	0,0672	0,0648	0,064	0,064	0,064	0,064	0,064	0,064	0,0738	0,0705	0,0705	0,0705	0,0583	0,0672	0,0738	0,0738	0,0738	0,0738	0,0738	0,0738
7	0,068	0,068	0,068	0,068	0,0672	0,064	0,068	0,068	0,068	0,068	0,068	0,068	0,0647	0,0647	0,0647	0,0647	0,0581	0,0832	0,0752	0,0752	0,0752	0,0647	0,0647	0,0647
8	0,0672	0,0712	0,0712	0,0712	0,0656	0,0656	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672	0,0665	0,0725	0,0725	0,0725	0,0624	0,0784	0,0665	0,0665	0,0665	0,0665	0,0665	0,0665
9	0,0672	0,0744	0,0744	0,0744	0,0616	0,0648	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672	0,0778	0,0702	0,0702	0,0702	0,0696	0,0688	0,0778	0,0778	0,0778	0,0778	0,0778	0,0778
10	0,0704	0,0816	0,0816	0,0816	0,068	0,0704	0,0704	0,0704	0,0704	0,0695	0,0676	0,0665	0,0679	0,0890	0,0890	0,0890	0,0580	0,0696	0,0944	0,0944	0,0944	0,0683	0,0686	0,0703
11	0,064	0,0664	0,0664	0,0664	0,0624	0,0616	0,064	0,064	0,064	0,064	0,064	0,064	0,0680	0,0571	0,0571	0,0571	0,0633	0,0752	0,0896	0,0896	0,0896	0,0680	0,0680	0,0680
12	0,072	0,072	0,072	0,072	0,0712	0,0712	0,072	0,072	0,072	0,072	0,072	0,072	0,0648	0,0648	0,0648	0,0648	0,0697	0,0856	0,0912	0,0912	0,0912	0,0648	0,0648	0,0648
13	0,064	0,0592	0,0592	0,0592	0,0648	0,0656	0,064	0,064	0,064	0,0586	0,0558	0,0542	0,0633	0,0631	0,0631	0,0631	0,0637	0,0696	0,0633	0,0633	0,0633	0,0567	0,0570	0,0519
14	0,0624	0,068	0,068	0,068	0,06	0,0632	0,0704	0,0704	0,0704	0,0640	0,0640	0,0640	0,0510	0,0671	0,0671	0,0671	0,0514	0,0696	0,0510	0,0510	0,0510	0,0583	0,0583	0,0583
15	0,0656	0,072	0,0664	0,0664	0,0656	0,0632	0,0656	0,0656	0,0656	0,0656	0,0656	0,0656	0,0693	0,0772	0,0587	0,0587	0,0617	0,0752	0,0976	0,0976	0,0976	0,0693	0,0693	0,0693
16	0,056	0,0552	0,0536	0,0536	0,052	0,0512	0,0528	0,056	0,056	0,0525	0,0513	0,0506	0,0497	0,0481	0,0552	0,0552	0,0497	0,0576	0,0497	0,0497	0,0497	0,0525	0,0569	0,0384
17	0,0536	0,0536	0,0536	0,0536	0,0528	0,0552	0,0536	0,0536	0,0536	0,0536	0,0536	0,0536	0,0517	0,0517	0,0517	0,0517	0,0519	0,0568	0,0517	0,0517	0,0517	0,0517	0,0517	0,0517
18	0,064	0,0608	0,0608	0,0608	0,0592	0,0592	0,064	0,064	0,064	0,064	0,064	0,064	0,0675	0,0614	0,0614	0,0614	0,0574	0,0816	0,0675	0,0675	0,0675	0,0675	0,0675	0,0675
19	0,0696	0,0744	0,0744	0,0744	0,0768	0,0752	0,0744	0,0744	0,0744	0,0696	0,0696	0,0696	0,0630	0,0804	0,0804	0,0804	0,0754	0,0936	0,0630	0,0630	0,0630	0,0630	0,0630	0,0630
20	0,0576	0,072	0,0784	0,0784	0,0576	0,0592	0,0576	0,0576	0,0576	0,0576	0,0576	0,0576	0,0586	0,0750	0,0867	0,0867	0,0615	0,0664	0,0832	0,0832	0,0832	0,0586	0,0586	0,0586
21	0,0592	0,0632	0,0632	0,0632	0,0632	0,0648	0,0592	0,0592	0,0592	0,0592	0,0592	0,0592	0,0590	0,0638	0,0638	0,0638	0,0645	0,0704	0,0590	0,0590	0,0590	0,0590	0,0590	0,0590
22	0,0616	0,0664	0,0752	0,0752	0,0568	0,0592	0,072	0,072	0,072	0,0616	0,0609	0,0605	0,0590	0,0690	0,0687	0,0687	0,0599	0,0656	0,0808	0,0808	0,0808	0,0622	0,0641	0,0591
23	0,0568	0,0568	0,0568	0,0568	0,0528	0,0536	0,0568	0,0568	0,0568	0,0600	0,0601	0,0601	0,0550	0,0550	0,0550	0,0550	0,0547	0,0632	0,0736	0,0736	0,0736	0,0668	0,0564	0,0674
24	0,0648	0,0616	0,0616	0,0616	0,0632	0,0632	0,0648	0,0648	0,0648	0,0667	0,0683	0,0692	0,0672	0,0531	0,0531	0,0531	0,0652	0,0552	0,0672	0,0672	0,0672	0,0545	0,0568	0,0616
25	0,0744	0,0824	0,0824	0,0824	0,0736	0,0704	0,0744	0,0744	0,0744	0,0744	0,0744	0,0744	0,0796	0,0862	0,0862	0,0862	0,0690	0,0808	0,0796	0,0796	0,0796	0,0796	0,0796	0,0796
26	0,0712	0,0712	0,0712	0,0712	0,0696	0,0672	0,0712	0,0712	0,0712	0,0712	0,0712	0,0712	0,0714	0,0714	0,0714	0,0714	0,0704	0,0752	0,0714	0,0714	0,0714	0,0714	0,0714	0,0714
27	0,0576	0,0552	0,0552	0,0552	0,0592	0,0568	0,0576	0,0576	0,0576	0,0576	0,0576	0,0576	0,0512	0,0549	0,0549	0,0549	0,0542	0,0632	0,0512	0,0512	0,0512	0,0512	0,0512	0,0512
28	0,06	0,06	0,06	0,06	0,0608	0,0608	0,064	0,064	0,064	0,06	0,06	0,06	0,0537	0,0537	0,0537	0,0537	0,0605	0,0664	0,0888	0,0888	0,0888	0,0537	0,0537	0,0537
29	0,0712	0,0784	0,0784	0,0784	0,0672	0,0664	0,0712	0,0712	0,0712	0,0712	0,0712	0,0712	0,0732	0,0777	0,0777	0,0777	0,0746	0,08	0,0732	0,0732	0,0732	0,0732	0,0732	0,0732
30	0,0616	0,064	0,0736	0,0736	0,064	0,06	0,0592	0,064	0,064	0,0609	0,0588	0,0576	0,0583	0,0795	0,0763	0,0763	0,0633	0,0736	0,0583	0,0583	0,0583	0,0703	0,0466	0,0584

Fold	Base 4																							
	5.000 registros																							
	Redes Neurais, 20 Neurônios, BPROP												Redes Neurais, 20 Neurônios, LEVMAR											
	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*	Simples	*RISKSEG1*	*RISKSEG2*	*RISKSEG3*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*SegTree_3*	*NNTree_1*	*NNTree_2*	*NNTree_3*
1	0,0404	0,0492	0,0492	0,0492	0,0522	0,0521	0,0404	0,0404	0,0404	0,0404	0,0404	0,0404	0,0557	0,0466	0,0466	0,0466	0,0613	0,0505	0,0557	0,0557	0,0557	0,0557	0,0557	0,0557
2	0,0799	0,0799	0,0799	0,0799	0,0697	0,0667	0,0799	0,0799	0,0799	0,0799	0,0799	0,0799	0,0782	0,0782	0,0782	0,0782	0,0832	0,0902	0,0782	0,0782	0,0782	0,0782	0,0782	0,0782
3	0,0514	0,0514	0,0514	0,0514	0,0470	0,0491	0,0514	0,0514	0,0514	0,0514	0,0514	0,0514	0,0646	0,0646	0,0646	0,0646	0,0453	0,0684	0,0646	0,0646	0,0646	0,0646	0,0646	0,0646
4	0,0578	0,0578	0,0578	0,0578	0,0660	0,0736	0,0578	0,0578	0,0578	0,0578	0,0578	0,0578	0,0547	0,0547	0,0547	0,0547	0,0756	0,0819	0,0547	0,0547	0,0547	0,0547	0,0547	0,0547
5	0,0755	0,0755	0,0755	0,0755	0,0652	0,0712	0,0755	0,0755	0,0755	0,0755	0,0755	0,0755	0,0707	0,0707	0,0707	0,0707	0,0754	0,0635	0,0707	0,0707	0,0707	0,0707	0,0707	0,0707
6	0,0582	0,0691	0,0691	0,0691	0,0664	0,0672	0,0582	0,0582	0,0582	0,0582	0,0582	0,0582	0,0710	0,0662	0,0662	0,0662	0,0571	0,0555	0,0710	0,0710	0,0710	0,0710	0,0710	0,0710
7	0,0699	0,0699	0,0699	0,0699	0,0748	0,0647	0,0699	0,0699	0,0699	0,0699	0,0699	0,0699	0,0674	0,0674	0,0674	0,0674	0,0591	0,0848	0,0674	0,0674	0,0674	0,0674	0,0674	0,0674
8	0,0691	0,0627	0,0627	0,0627	0,0625	0,0712	0,0691	0,0691	0,0691	0,0691	0,0691	0,0691	0,0692	0,0733	0,0733	0,0733	0,0638	0,0781	0,0692	0,0692	0,0692	0,0692	0,0692	0,0692
9	0,0592	0,0724	0,0724	0,0724	0,0620	0,0704	0,0592	0,0592	0,0592	0,0592	0,0592	0,0592	0,0754	0,0786	0,0786	0,0786	0,0726	0,0685	0,0754	0,0754	0,0754	0,0754	0,0754	0,0754
10	0,0651	0,0794	0,0794	0,0794	0,0684	0,0663	0,0651	0,0651	0,0651	0,0651	0,0651	0,0651	0,0726	0,0935	0,0935	0,0935	0,0559	0,0653	0,0726	0,0726	0,0726	0,0726	0,0726	0,0726
11	0,0648	0,0685	0,0685	0,0685	0,0645	0,0692	0,0648	0,0648	0,0648	0,0648	0,0648	0,0648	0,0727	0,0615	0,0615	0,0615	0,0611	0,0849	0,0727	0,0727	0,0727	0,0727	0,0727	0,0727
12	0,0728	0,0728	0,0728	0,0728	0,0657	0,0720	0,0728	0,0728	0,0728	0,0728	0,0728	0,0728	0,0580	0,0580	0,0580	0,0580	0,0705	0,0953	0,0580	0,0580	0,0580	0,0580	0,0580	0,0580
13	0,0704	0,0594	0,0594	0,0594	0,0761	0,0664	0,0688	0,0688	0,0688	0,0704	0,0704	0,0704	0,0739	0,0588	0,0588	0,0588	0,0706	0,0661	0,0671	0,0671	0,0671	0,0739	0,0739	0,0739
14	0,0681	0,0680	0,0680	0,0680	0,0713	0,0643	0,0681	0,0681	0,0681	0,0681	0,0681	0,0681	0,0617	0,0628	0,0628	0,0628	0,0582	0,0749	0,0617	0,0617	0,0617	0,0617	0,0617	0,0617
15	0,0615	0,0722	0,0627	0,0627	0,0643	0,0687	0,0615	0,0615	0,0615	0,0615	0,0615	0,0615	0,0646	0,0760	0,0547	0,0547	0,0568	0,0805	0,0646	0,0646	0,0646	0,0646	0,0646	0,0646
16	0,0641	0,0457	0,0613	0,0613	0,0615	0,0457	0,06	0,06	0,06	0,0566	0,0554	0,0547	0,0464	0,0601	0,0448	0,0448	0,0491	0,0629	0,0616	0,0616	0,0616	0,0560	0,0592	0,0528
17	0,0609	0,0609	0,0609	0,0609	0,0623	0,0497	0,0609	0,0609	0,0609	0,0609	0,0609	0,0609	0,0617	0,0617	0,0617	0,0617	0,0600	0,0563	0,0617	0,0617	0,0617	0,0617	0,0617	0,0617
18	0,0713	0,0590	0,0590	0,0590	0,0616	0,0673	0,0713	0,0713	0,0713	0,0713	0,0713	0,0713	0,0656	0,0580	0,0580	0,0580	0,0533	0,0834	0,0656	0,0656	0,0656	0,0656	0,0656	0,0656
19	0,0672	0,0698	0,0698	0,0698	0,0792	0,0701	0,0672	0,0672	0,0672	0,0672	0,0672	0,0672	0,0611	0,0769	0,0769	0,0769	0,0827	0,0843	0,0611	0,0611	0,0611	0,0611	0,0611	0,0611
20	0,0552	0,0674	0,0795	0,0795	0,0601	0,0541	0,0552	0,0552	0,0552	0,0650	0,0610	0,0587	0,0658	0,0692	0,0849	0,0849	0,0687	0,0571	0,0658	0,0658	0,0658	0,0580	0,0685	0,0622
21	0,0600	0,0683	0,0683	0,0683	0,0679	0,0705	0,0600	0,0600	0,0600	0,0600	0,0600	0,0600	0,0570	0,0663	0,0663	0,0663	0,0621	0,0732	0,0570	0,0570	0,0570	0,0570	0,0570	0,0570
22	0,0498	0,0675	0,0759	0,0759	0,0615	0,0555	0,068	0,068	0,068	0,0498	0,0498	0,0498	0,0522	0,0750	0,0665	0,0665	0,0627	0,0684	0,0676	0,0676	0,0676	0,0522	0,0522	0,0522
23	0,0636	0,0636	0,0636	0,0636	0,0497	0,0499	0,0636	0,0636	0,0636	0,0704	0,0696	0,0692	0,0459	0,0459	0,0459	0,0459	0,0616	0,0660	0,0459	0,0459	0,0459	0,0652	0,0719	0,0706
24	0,0634	0,0629	0,0629	0,0629	0,0686	0,0632	0,0634	0,0634	0,0634	0,0667	0,0675	0,0681	0,0728	0,0510	0,0510	0,0510	0,0646	0,0450	0,0728	0,0728	0,0728	0,0701	0,0609	0,0610
25	0,0730	0,0837	0,0837	0,0837	0,0790	0,0659	0,0730	0,0730	0,0730	0,0730	0,0730	0,0730	0,0852	0,0841	0,0841	0,0841	0,0684	0,0833	0,0852	0,0852	0,0852	0,0852	0,0852	0,0852
26	0,0739	0,0739	0,0739	0,0739	0,0556	0,0627	0,0752	0,0752	0,0752	0,0739	0,0739	0,0739	0,0657	0,0657	0,0657	0,0657	0,0652	0,0777	0,0703	0,0703	0,0703	0,0657	0,0657	0,0657
27	0,0603	0,0563	0,0563	0,0563	0,0641	0,0610	0,0603	0,0603	0,0603	0,0603	0,0603	0,0603	0,0552	0,0533	0,0533	0,0533	0,0569	0,0546	0,0552	0,0552	0,0552	0,0552	0,0552	0,0552
28	0,0597	0,0597	0,0597	0,0597	0,0657	0,0650	0,0597	0,0597	0,0597	0,0597	0,0597	0,0597	0,0578	0,0578	0,0578	0,0578	0,0574	0,0614	0,0578	0,0578	0,0578	0,0578	0,0578	0,0578
29	0,0709	0,0830	0,0830	0,0830	0,0587	0,0626	0,0709	0,0709	0,0709	0,0709	0,0709	0,0709	0,0675	0,0803	0,0803	0,0803	0,0716	0,0806	0,0675	0,0675	0,0675	0,0675	0,0675	0,0675
30	0,0568	0,0770	0,0800	0,0800	0,0681	0,0641	0,06	0,064	0,064	0,0614	0,0595	0,0584	0,0515	0,0896	0,0794	0,0794	0,0611	0,0742	0,0584	0,0711	0,0711	0,0611	0,0576	0,0595

Tabela A.25 – Erros de Classificação da Base *Abalone*.

Fold	Regressão Linear								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,3453	0,1918	0,1846	0,5107	0,2254	0,1894	0,1750	0,296	0,2148
2	0,3612	0,1961	0,2033	0,5107	0,2583	0,2153	0,1937	0,3660	0,2440
3	0,2894	0,2177	0,2177	0,5107	0,2870	0,2033	0,2129	0,3891	0,2452
4	0,3165	0,2086	0,2038	0,5107	0,2470	0,2326	0,2110	0,3568	0,2172
5	0,2392	0,2248	0,2200	0,5107	0,2392	0,2320	0,2200	0,3232	0,2410
6	0,2535	0,2057	0,1866	0,5107	0,2655	0,2248	0,1818	0,3651	0,2195
7	0,2494	0,2278	0,2326	0,5107	0,2494	0,2350	0,2350	0,3764	0,2496
8	0,2057	0,2153	0,2129	0,5107	0,2129	0,2177	0,2009	0,3913	0,2390
9	0,4401	0,2200	0,2272	0,5107	0,2559	0,2320	0,2248	0,4055	0,2448
10	0,4186	0,2129	0,2177	0,5107	0,2966	0,2081	0,2009	0,2979	0,1979

Fold	Regressão Logística								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1918	0,1822	0,1750	0,1894	0,1894	0,1750	0,1750	0,1918	0,1918
2	0,2105	0,2033	0,2033	0,2200	0,2177	0,2057	0,2009	0,2105	0,2105
3	0,2129	0,2009	0,2081	0,2153	0,2033	0,2129	0,2129	0,2073	0,2153
4	0,2206	0,2278	0,2302	0,2302	0,2278	0,2158	0,2062	0,2176	0,2160
5	0,2320	0,2320	0,2153	0,2440	0,2416	0,2320	0,2320	0,2320	0,2320
6	0,1937	0,2081	0,1937	0,1961	0,1913	0,2033	0,1937	0,1873	0,1939
7	0,2278	0,2302	0,2278	0,2422	0,2350	0,2302	0,2254	0,2278	0,2278
8	0,2009	0,1985	0,2009	0,1985	0,1985	0,1866	0,1889	0,2009	0,2009
9	0,2368	0,2177	0,2224	0,2368	0,2320	0,2272	0,2272	0,2368	0,2368
10	0,1985	0,1913	0,1889	0,1961	0,1961	0,1985	0,1985	0,1985	0,1985

Fold	Redes Neurais, 3 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1872	0,2030	0,2030	0,1719	0,1688	0,1872	0,1872	0,1872	0,1872
2	0,1916	0,2016	0,2055	0,1813	0,1805	0,1916	0,1916	0,1916	0,1916
3	0,1950	0,1830	0,1830	0,2059	0,1815	0,1950	0,1950	0,1950	0,1950
4	0,2128	0,1994	0,1994	0,2080	0,1999	0,1946	0,1946	0,1985	0,1863
5	0,2239	0,2175	0,2399	0,2152	0,2354	0,2140	0,2140	0,2239	0,2239
6	0,1949	0,1949	0,1949	0,1920	0,1902	0,2173	0,2173	0,1949	0,1949
7	0,2153	0,2153	0,2153	0,2356	0,2317	0,2153	0,2153	0,2153	0,2153
8	0,1858	0,1834	0,1834	0,1941	0,1863	0,1858	0,1858	0,1858	0,1858
9	0,2083	0,2220	0,2220	0,2109	0,1953	0,2102	0,2102	0,2083	0,2083
10	0,1941	0,1941	0,1941	0,1771	0,1850	0,1941	0,1941	0,1941	0,1941

Fold	Redes Neurais, 3 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1846	0,2038	0,2038	0,1846	0,1822	0,1846	0,1846	0,1846	0,1846
2	0,1889	0,2105	0,2057	0,1818	0,1794	0,1889	0,1889	0,1889	0,1889
3	0,2009	0,1746	0,1746	0,1985	0,1889	0,2009	0,2009	0,2009	0,2009
4	0,2062	0,1966	0,1966	0,2038	0,2014	0,1918	0,1918	0,1936	0,1920
5	0,2153	0,2200	0,2320	0,2177	0,2344	0,2200	0,2200	0,2153	0,2153
6	0,1961	0,1961	0,1961	0,1913	0,1985	0,2129	0,2129	0,1961	0,1961
7	0,2182	0,2182	0,2182	0,2278	0,2374	0,2182	0,2182	0,2182	0,2182
8	0,1985	0,1842	0,1842	0,1937	0,1913	0,1985	0,1985	0,1985	0,1985
9	0,2177	0,2248	0,2248	0,2129	0,1985	0,2129	0,2129	0,2177	0,2177
10	0,1913	0,1913	0,1913	0,1818	0,1866	0,1913	0,1913	0,1913	0,1913

Fold	Redes Neurais, 10 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1852	0,2161	0,2161	0,1741	0,1757	0,1852	0,1852	0,1852	0,1852
2	0,1907	0,2139	0,2034	0,1790	0,1769	0,1907	0,1907	0,1907	0,1907
3	0,2043	0,1846	0,1846	0,2030	0,1884	0,2043	0,2043	0,2043	0,2043
4	0,2029	0,1916	0,1916	0,2040	0,2060	0,1847	0,1847	0,1833	0,1860
5	0,2202	0,2257	0,2265	0,2113	0,2272	0,2129	0,2129	0,2202	0,2202
6	0,2036	0,2036	0,2036	0,1932	0,1913	0,2054	0,2054	0,2036	0,2036
7	0,2169	0,2169	0,2169	0,2283	0,2390	0,2169	0,2169	0,2169	0,2169
8	0,2011	0,2012	0,2012	0,1914	0,1851	0,2011	0,2011	0,2011	0,2011
9	0,2172	0,2345	0,2345	0,2177	0,1943	0,2081	0,2081	0,2172	0,2172
10	0,1965	0,1965	0,1965	0,1836	0,1831	0,1965	0,1965	0,1965	0,1965

Fold	Redes Neurais, 10 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2230	0,2134	0,2134	0,1774	0,1942	0,2230	0,2230	0,2448	0,2376
2	0,2177	0,1961	0,1961	0,1818	0,1770	0,2033	0,2033	0,2297	0,2286
3	0,2129	0,1937	0,1937	0,1889	0,2009	0,2129	0,2129	0,2129	0,2129
4	0,1990	0,2062	0,2062	0,1918	0,2062	0,1990	0,1990	0,1990	0,1990
5	0,2392	0,2392	0,2392	0,2153	0,2344	0,2392	0,2392	0,2277	0,2171
6	0,2177	0,2296	0,2464	0,2081	0,2200	0,2177	0,2177	0,2100	0,2024
7	0,2829	0,2517	0,2517	0,2446	0,2398	0,2302	0,2302	0,2829	0,2829
8	0,2153	0,1889	0,1889	0,1794	0,1913	0,2081	0,2081	0,2173	0,2160
9	0,2464	0,2129	0,2129	0,2320	0,2177	0,2081	0,2081	0,2383	0,2379
10	0,2272	0,2009	0,2440	0,1985	0,2081	0,2224	0,2224	0,2246	0,2175

Fold	Redes Neurais, 20 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1887	0,2131	0,2131	0,1748	0,1749	0,1887	0,1887	0,1887	0,1887
2	0,1908	0,2102	0,2014	0,1758	0,1791	0,1908	0,1908	0,1908	0,1908
3	0,1995	0,1908	0,1908	0,2087	0,1938	0,1995	0,1995	0,1995	0,1995
4	0,2105	0,1875	0,1875	0,2154	0,1860	0,1941	0,1941	0,1832	0,1958
5	0,2143	0,2363	0,2235	0,2103	0,2321	0,2163	0,2163	0,2143	0,2143
6	0,2126	0,2126	0,2126	0,1918	0,1825	0,2148	0,2148	0,2126	0,2126
7	0,2149	0,2149	0,2149	0,2292	0,2397	0,2149	0,2149	0,2149	0,2149
8	0,2079	0,1962	0,1962	0,1948	0,1801	0,2079	0,2079	0,2079	0,2079
9	0,2061	0,2380	0,2380	0,2235	0,2023	0,1961	0,1961	0,2061	0,2061
10	0,1910	0,1910	0,1910	0,1838	0,1845	0,1910	0,1910	0,1910	0,1910

Fold	Redes Neurais, 20 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2249	0,2156	0,2156	0,1813	0,1964	0,2249	0,2249	0,2469	0,2283
2	0,2138	0,2040	0,2040	0,1854	0,1797	0,1902	0,1902	0,2326	0,2371
3	0,2066	0,1921	0,1921	0,1799	0,2000	0,2066	0,2066	0,2066	0,2066
4	0,2041	0,2163	0,1994	0,1914	0,2068	0,2041	0,2041	0,2041	0,2041
5	0,2481	0,2481	0,2481	0,2149	0,2365	0,2481	0,2481	0,2214	0,2176
6	0,2009	0,2325	0,2450	0,2087	0,2189	0,2009	0,2009	0,2129	0,2098
7	0,2893	0,2473	0,2473	0,2420	0,2513	0,2264	0,2264	0,2893	0,2893
8	0,2031	0,1760	0,1939	0,1774	0,1876	0,2065	0,2065	0,2119	0,2037
9	0,2503	0,2203	0,2159	0,2341	0,2223	0,2018	0,2018	0,2273	0,2350
10	0,2321	0,2152	0,2437	0,2069	0,2143	0,2316	0,2316	0,2177	0,2139

Tabela A.26 – Erros de Classificação da Base *Adult*.

Fold	Regressão Linear								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,164	0,169	0,16	0,168	0,171	0,157	0,159	0,1670	0,1301
2	0,164	0,161	0,163	0,169	0,17	0,169	0,166	0,1418	0,1418
3	0,202	0,202	0,202	0,2	0,177	0,174	0,169	0,1486	0,1539
4	0,18	0,17	0,169	0,177	0,191	0,179	0,175	0,1618	0,1512
5	0,163	0,138	0,138	0,142	0,154	0,135	0,133	0,1374	0,1065
6	0,197	0,197	0,197	0,189	0,191	0,173	0,169	0,1748	0,1469
7	0,145	0,145	0,145	0,158	0,165	0,145	0,145	0,1265	0,1335
8	0,198	0,158	0,168	0,19	0,214	0,16	0,158	0,1415	0,1322
9	0,152	0,16	0,16	0,164	0,17	0,152	0,152	0,1313	0,1385
10	0,173	0,173	0,173	0,171	0,177	0,159	0,155	0,1307	0,1332

Fold	Regressão Logística								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,158	0,16	0,159	0,152	0,153	0,158	0,158	0,158	0,158
2	0,153	0,158	0,158	0,15	0,149	0,153	0,153	0,153	0,153
3	0,165	0,165	0,165	0,164	0,164	0,165	0,165	0,165	0,165
4	0,17	0,16	0,16	0,167	0,169	0,17	0,17	0,1596	0,1596
5	0,117	0,127	0,127	0,122	0,123	0,12	0,12	0,117	0,117
6	0,159	0,159	0,159	0,161	0,16	0,159	0,159	0,159	0,159
7	0,13	0,13	0,13	0,128	0,129	0,131	0,131	0,13	0,13
8	0,153	0,153	0,153	0,152	0,151	0,157	0,157	0,1313	0,1313
9	0,142	0,142	0,142	0,139	0,142	0,14	0,14	0,142	0,142
10	0,143	0,143	0,143	0,141	0,142	0,143	0,143	0,143	0,143

Fold	Redes Neurais, 3 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,151	0,152	0,152	0,1398	0,1400	0,151	0,151	0,1304	0,1251
2	0,155	0,157	0,157	0,1447	0,1470	0,158	0,158	0,155	0,155
3	0,161	0,161	0,161	0,1431	0,1476	0,161	0,161	0,1547	0,1496
4	0,166	0,166	0,166	0,1482	0,1476	0,165	0,165	0,1579	0,1536
5	0,12	0,12	0,12	0,1494	0,1473	0,114	0,113	0,1102	0,1057
6	0,16	0,16	0,16	0,1441	0,1439	0,157	0,157	0,1472	0,1420
7	0,132	0,139	0,139	0,1461	0,1459	0,132	0,132	0,132	0,132
8	0,16	0,16	0,16	0,1390	0,1414	0,148	0,148	0,16	0,16
9	0,14	0,14	0,14	0,1398	0,1445	0,149	0,149	0,14	0,14
10	0,144	0,152	0,152	0,1469	0,1453	0,136	0,142	0,1268	0,1205

Fold	Redes Neurais, 3 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,15	0,148	0,148	0,148	0,15	0,15	0,15	0,1320	0,1259
2	0,165	0,163	0,163	0,154	0,162	0,16	0,16	0,165	0,165
3	0,159	0,163	0,163	0,16	0,156	0,159	0,159	0,1553	0,1477
4	0,172	0,172	0,172	0,165	0,175	0,172	0,172	0,172	0,172
5	0,115	0,124	0,124	0,117	0,119	0,12	0,12	0,1091	0,1046
6	0,16	0,16	0,16	0,155	0,16	0,16	0,16	0,1560	0,1485
7	0,142	0,143	0,144	0,132	0,135	0,142	0,145	0,1320	0,1259
8	0,151	0,155	0,165	0,157	0,156	0,167	0,167	0,1358	0,1254
9	0,151	0,145	0,153	0,145	0,152	0,151	0,151	0,1336	0,1280
10	0,153	0,15	0,15	0,146	0,145	0,153	0,153	0,1229	0,1163

Fold	Redes Neurais, 10 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,149	0,15	0,15	0,1365	0,1373	0,149	0,149	0,1359	0,1290
2	0,178	0,16	0,16	0,1427	0,1466	0,178	0,178	0,178	0,178
3	0,168	0,168	0,168	0,1406	0,1423	0,164	0,164	0,1597	0,1516
4	0,172	0,168	0,168	0,1455	0,1476	0,172	0,172	0,1551	0,1489
5	0,131	0,126	0,126	0,1469	0,1465	0,132	0,132	0,1102	0,1027
6	0,171	0,163	0,163	0,1418	0,1431	0,153	0,153	0,1583	0,1469
7	0,14	0,148	0,154	0,1457	0,1498	0,14	0,14	0,1414	0,1358
8	0,163	0,163	0,163	0,1353	0,1349	0,149	0,16	0,1478	0,1362
9	0,16	0,16	0,16	0,1414	0,1423	0,16	0,16	0,1443	0,1354
10	0,144	0,144	0,144	0,1428	0,1424	0,144	0,144	0,1290	0,1232

Fold	Redes Neurais, 10 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1470	0,1520	0,1520	0,1394	0,1434	0,1470	0,1470	0,1300	0,1268
2	0,1761	0,1595	0,1595	0,1442	0,1385	0,1761	0,1761	0,1761	0,1761
3	0,1692	0,1692	0,1692	0,1386	0,1447	0,1635	0,1635	0,1660	0,1456
4	0,1731	0,1743	0,1743	0,1498	0,1381	0,1731	0,1731	0,1542	0,1554
5	0,1411	0,1278	0,1278	0,1512	0,1512	0,1292	0,1248	0,1096	0,1106
6	0,1684	0,1590	0,1590	0,1340	0,1408	0,1603	0,1603	0,1628	0,1473
7	0,1382	0,1512	0,1493	0,1473	0,1518	0,1449	0,1449	0,1388	0,1379
8	0,1699	0,1699	0,1699	0,1382	0,1394	0,1374	0,1637	0,1408	0,1425
9	0,1634	0,1634	0,1634	0,1358	0,1419	0,1634	0,1634	0,1480	0,1331
10	0,1410	0,1410	0,1410	0,1380	0,1442	0,1410	0,1410	0,1261	0,1129

Fold	Redes Neurais, 20 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,157	0,155	0,155	0,143	0,156	0,155	0,154	0,1415	0,1344
2	0,163	0,17	0,17	0,146	0,158	0,163	0,163	0,1462	0,1386
3	0,173	0,173	0,173	0,167	0,165	0,173	0,173	0,1581	0,1520
4	0,158	0,178	0,178	0,168	0,165	0,158	0,158	0,1534	0,1489
5	0,135	0,135	0,135	0,128	0,125	0,143	0,143	0,1189	0,1125
6	0,173	0,173	0,173	0,165	0,158	0,173	0,173	0,1588	0,1511
7	0,139	0,139	0,139	0,13	0,144	0,139	0,139	0,139	0,139
8	0,161	0,161	0,161	0,158	0,158	0,161	0,162	0,1438	0,1322
9	0,161	0,156	0,156	0,146	0,146	0,161	0,169	0,1386	0,1299
10	0,154	0,154	0,154	0,147	0,144	0,154	0,154	0,1368	0,1294

Fold	Redes Neurais, 20 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1516	0,1652	0,1652	0,1425	0,1458	0,1614	0,1559	0,1419	0,1268
2	0,1594	0,1633	0,1633	0,1466	0,1486	0,1594	0,1594	0,1443	0,1408
3	0,1649	0,1649	0,1649	0,1619	0,1683	0,1669	0,1669	0,1620	0,1421
4	0,1533	0,1814	0,1814	0,1686	0,1758	0,1533	0,1533	0,1562	0,1445
5	0,1472	0,1472	0,1472	0,1328	0,1214	0,1377	0,1377	0,1212	0,1160
6	0,1697	0,1697	0,1697	0,1702	0,1565	0,1697	0,1697	0,1556	0,1472
7	0,1288	0,1288	0,1288	0,1478	0,1395	0,1288	0,1288	0,1288	0,1288
8	0,1654	0,1654	0,1654	0,1605	0,1618	0,1619	0,1621	0,1500	0,1452
9	0,1641	0,1632	0,1632	0,1415	0,1485	0,1677	0,1666	0,1244	0,1336
10	0,1549	0,1549	0,1549	0,1455	0,1414	0,1549	0,1549	0,1280	0,1312

Tabela A.27 – Erros de Classificação da Base *Car*.

	Regressão Linear								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,3895	0,0406	0,0406	0,3779	0,4011	0,0406	0,0406	0,1501	0,0410
2	0,3294	0,0462	0,0289	0,3236	0,3236	0,0867	0,0693	0,2464	0,0683
3	0,3988	0,0346	0,0462	0,4335	0,4277	0,0404	0,0404	0,1279	0,0427
4	0,2832	0,0693	0,0635	0,3179	0,3236	0,0520	0,0404	0,1925	0,0715
5	0,3930	0,0346	0,0404	0,3583	0,3872	0,0520	0,0693	0,1770	0,0446
6	0,3546	0,0406	0,0348	0,3488	0,3546	0,0406	0,0232	0,1731	0,0507
7	0,3468	0,0115	0,0115	0,3468	0,3583	0,0231	0,0231	0,1744	0,0354
8	0,4046	0,0462	0,0462	0,4046	0,4104	0,0462	0,0289	0,1722	0,0334
9	0,2774	0,0404	0,0231	0,3063	0,2658	0,0462	0,0404	0,1433	0,0421
10	0,3063	0,0635	0,0520	0,3179	0,3179	0,0289	0,0520	0,2013	0,0446

	Regressão Logística								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0523	0,0348	0,0232	0,0523	0,0581	0,0465	0,0581	0,0523	0,0523
2	0,0578	0,0289	0,0173	0,0578	0,0578	0,0520	0,0635	0,0633	0,0633
3	0,0289	0,0346	0,0462	0,0404	0,0404	0,0404	0,0404	0,0269	0,0237
4	0,0462	0,0520	0,0057	0,0578	0,0520	0,0635	0,0462	0,0462	0,0462
5	0,0809	0,0578	0,0289	0,0751	0,0693	0,0809	0,0809	0,0809	0,0809
6	0,0465	0,0290	0,0116	0,0465	0,0465	0,0290	0,0232	0,0388	0,0406
7	0,0346	0,0289	0,0173	0,0231	0,0289	0,0231	0,0231	0,0346	0,0346
8	0,0404	0,0173	0,0173	0,0346	0,0346	0,0346	0,0346	0,0404	0,0404
9	0,0462	0,0346	0,0173	0,0520	0,0520	0,0404	0,0289	0,0333	0,0374
10	0,0346	0,0404	0	0,0289	0,0289	0,0231	0,0231	0,0277	0,0272

	Redes Neurais, 3 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0	0,0058	0,0058	0,0174	0,0174	0	0	0	0
2	0,0115	0,0115	0,0057	0,0115	0,0231	0,0057	0,0057	0,0140	0,0151
3	0,0289	0,0057	0	0,0231	0,0289	0,0289	0,0289	0,0289	0,0289
4	0,0289	0,0173	0	0,0173	0,0115	0,0057	0,0057	0,0202	0,0202
5	0,0173	0,0173	0	0	0	0,0057	0	0,0173	0,0173
6	0,0232	0,0232	0	0	0	0,0232	0,0232	0,0232	0,0232
7	0,0173	0,0173	0,0057	0,0057	0,0057	0,0173	0,0173	0,0201	0,0201
8	0,0057	0,0057	0,0057	0	0	0,0057	0	0,0057	0,0057
9	0,0173	0,0057	0	0,0115	0,0173	0,0231	0,0231	0,0173	0,0173
10	0,0115	0,0057	0,0115	0,0115	0,0231	0,0057	0,0057	0,0115	0,0115

	Redes Neurais, 3 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0	0,0058	0,0058	0,0232	0,0174	0	0	0	0
2	0,0115	0,0115	0,0057	0,0115	0,0231	0,0057	0,0057	0,0140	0,0151
3	0,0289	0,0057	0	0,0289	0,0289	0,0289	0,0289	0,0289	0,0289
4	0,0289	0,0173	0	0	0,0115	0,0057	0,0057	0,0202	0,0202
5	0,0173	0	0,0057	0,0057	0,0057	0	0	0,0173	0,0173
6	0,0232	0	0	0,0058	0	0,0232	0,0232	0,0232	0,0232
7	0,0173	0,0057	0,0057	0,0057	0,0057	0,0173	0,0173	0,0201	0,0201
8	0,0057	0,0057	0,0057	0	0	0,0057	0	0,0057	0,0057
9	0,0173	0,0057	0	0,0115	0,0173	0,0231	0,0231	0,0173	0,0173
10	0,0115	0,0057	0,0115	0,0115	0,0231	0,0057	0,0057	0,0115	0,0115

	Redes Neurais, 10 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0290	0,0232	0,0116	0,0290	0,0406	0,0174	0,0174	0,0290	0,0290
2	0,0346	0,0289	0,0173	0,0520	0,0693	0,0231	0,0231	0,0387	0,0455
3	0,0404	0,0173	0,0231	0,0346	0,0404	0,0346	0,0289	0,0303	0,0285
4	0,0231	0,0231	0,0289	0,0173	0,0520	0,0231	0,0231	0,0231	0,0231
5	0,0404	0,0346	0,0173	0,0231	0,0462	0,0231	0,0404	0,0347	0,0297
6	0,0465	0,0174	0,0058	0,0290	0,0406	0,0232	0,0174	0,0424	0,0424
7	0,0115	0,0231	0,0231	0,0057	0,0057	0,0115	0,0115	0,0115	0,0115
8	0,0289	0,0289	0,0289	0,0115	0,0173	0,0289	0,0173	0,0289	0,0289
9	0,0231	0,0346	0,0346	0,0115	0,0289	0,0346	0,0173	0,0231	0,0231
10	0,0289	0,0289	0,0231	0,0231	0,0231	0,0289	0,0277	0,0248	

	Redes Neurais, 10 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0290	0,0232	0,0116	0,0290	0,0406	0,0174	0,0174	0,0290	0,0290
2	0,0346	0,0289	0,0173	0,0404	0,0693	0,0231	0,0231	0,0387	0,0455
3	0,0404	0,0173	0,0231	0,0404	0,0404	0,0346	0,0289	0,0303	0,0285
4	0,0231	0,0231	0,0289	0,0173	0,0520	0,0231	0,0231	0,0231	0,0231
5	0,0404	0,0346	0,0173	0,0231	0,0462	0,0231	0,0404	0,0347	0,0297
6	0,0465	0,0174	0,0058	0,0232	0,0406	0,0232	0,0174	0,0424	0,0424
7	0,0115	0,0231	0,0231	0,0057	0,0057	0,0115	0,0115	0,0115	0,0115
8	0,0289	0,0289	0,0289	0,0115	0,0173	0,0289	0,0173	0,0289	0,0289
9	0,0231	0,0346	0,0346	0,0231	0,0289	0,0346	0,0173	0,0231	0,0231
10	0,0289	0,0289	0,0231	0,0231	0,0231	0,0231	0,0289	0,0277	0,0248

	Redes Neurais, 20 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0697	0,0290	0,0348	0,0465	0,0523	0,0465	0,0465	0,0170	0,0120
2	0,0346	0,0520	0,0289	0,0462	0,0578	0,0346	0,0289	0,0387	0,0387
3	0,0462	0,0462	0,0289	0,0520	0,0404	0,0404	0,0289	0,0336	0,0261
4	0,0462	0,0289	0,0346	0,0404	0,0346	0,0289	0,0289	0,0462	0,0462
5	0,0867	0,0346	0,0173	0,0635	0,0924	0,0231	0,0231	0,0416	0,0416
6	0,0465	0,0232	0,0290	0,0290	0,0465	0,0348	0,0290	0,0247	0,0247
7	0,0231	0,0173	0,0231	0,0115	0,0115	0,0346	0,0462	0,0268	0,0260
8	0,0462	0,0115	0,0289	0,0231	0,0289	0,0462	0,0462	0,0462	0,0462
9	0,0404	0,0462	0,0173	0,0346	0,0404	0,0173	0,0231	0,03	0,03
10	0,0346	0,0231	0,0173	0,0173	0,0520	0,0404	0,0462	0,0243	0,0223

	Redes Neurais, 20 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0697	0,0290	0,0348	0,0523	0,0523	0,0465	0,0465	0,0170	0,0120
2	0,0346	0,0520	0,0289	0,0462	0,0578	0,0346	0,0289	0,0387	0,0387
3	0,0462	0,0462	0,0289	0,0520	0,0404	0,0404	0,0289	0,0336	0,0261
4	0,0462	0,0289	0,0346	0,0404	0,0346	0,0289	0,0289	0,0462	0,0462
5	0,0867	0,0346	0,0173	0,0578	0,0924	0,0231	0,0231	0,0416	0,0416
6	0,0465	0,0232	0,0290	0,0290	0,0465	0,0348	0,0290	0,0247	0,0247
7	0,0231	0,0173	0,0231	0,0115	0,0115	0,0346	0,0462	0,0268	0,0260
8	0,0462	0,0115	0,0289	0,0289	0,0289	0,0462	0,0462	0,0462	0,0462
9	0,0404	0,0462	0,0173	0,0346	0,0404	0,0173	0,0231	0,03	0,03
10	0,0346	0,0231	0,0173	0,0289	0,0520	0,0404	0,0462	0,0243	0,0223

Tabela A.28 – Erros de Classificação da Base *Chess*.

	Regressão Linear								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0909	0,0438	0,0219	0,0721	0,0658	0,0501	0,0125	0,0381	0,0448
2	0,0718	0,0156	0,0156	0,0718	0,0656	0,05	0,0156	0,2689	0,0315
3	0,0783	0,0470	0,0407	0,0658	0,0470	0,0532	0,0438	0,2605	0,0368
4	0,1093	0,05	0,05	0,0718	0,0468	0,05	0,0218	0,1080	0,0240
5	0,1093	0,0437	0,0125	0,0812	0,0406	0,0562	0,0156	0,2937	0,0187
6	0,0438	0,0344	0,0438	0,0470	0,0658	0,0438	0,0282	0,1407	0,0331
7	0,0437	0,0375	0,0281	0,0406	0,0312	0,0281	0,0187	0,0590	0,0302
8	0,0971	0,0532	0,0376	0,1034	0,0752	0,0815	0,0250	0,1987	0,0534
9	0,1218	0,0375	0,0375	0,0843	0,0718	0,0687	0,025	0,2148	0,0258
10	0,0906	0,0406	0,0375	0,075	0,0375	0,0437	0,0218	0,2156	0,0191

	Regressão Logística								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0313	0,0031	0,0094	0,0313	0,0282	0,0313	0,0094	0,0360	0,0360
2	0,025	0,0125	0,0093	0,0187	0,025	0,025	0,025	0,0303	0,0303
3	0,0219	0,0094	0,0094	0,0250	0,0250	0,0219	0,0094	0,0317	0,0288
4	0,0187	0,0156	0,0218	0,0187	0,0187	0,0187	0,0187	0,0187	0,0187
5	0,0156	0,0093	0,0084	0,0218	0,0156	0,0156	0,0156	0,0166	0,0166
6	0,0156	0,0062	0,0062	0,0188	0,0156	0,0156	0,0156	0,0252	0,0252
7	0,0218	0,0093	0,0093	0,0187	0,0187	0,0218	0,0218	0,0210	0,0210
8	0,0564	0,0094	0,0062	0,0470	0,0501	0,0564	0,0564	0,0405	0,0405
9	0,025	0,0093	0,0093	0,0218	0,025	0,025	0,025	0,0255	0,0255
10	0,0281	0,0218	0,0125	0,0187	0,0187	0,0281	0,0281	0,0169	0,0223

	Redes Neurais, 3 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0313	0,0188	0,0125	0,0282	0,0376	0,0250	0,0062	0,0360	0,0384
2	0,0375	0,0187	0,025	0,0375	0,05	0,0406	0,0156	0,0260	0,0249
3	0,0626	0,0344	0,0282	0,0626	0,0407	0,0626	0,0250	0,0317	0,0304
4	0,0343	0,025	0,0093	0,0312	0,0406	0,0281	0,0187	0,0190	0,0176
5	0,0343	0,0281	0,0187	0,0187	0,0343	0,025	0,025	0,0208	0,0218
6	0,0250	0,0282	0,0470	0,0250	0,0344	0,0250	0,0250	0,0315	0,0300
7	0,0562	0,0156	0,0062	0,0281	0,0406	0,0187	0,0187	0,0189	0,0175
8	0,0658	0,0282	0,0282	0,0626	0,0689	0,0438	0,0219	0,0555	0,0534
9	0,0468	0,025	0,025	0,0406	0,0343	0,0312	0,0281	0,0404	0,0387
10	0,0125	0,025	0,0156	0,0218	0,0343	0,0281	0,025	0,0125	0,0125

	Redes Neurais, 3 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0313	0,0188	0,0125	0,0313	0,0376	0,0250	0,0062	0,0360	0,0384
2	0,0375	0,0187	0,025	0,0312	0,05	0,0406	0,0156	0,0260	0,0249
3	0,0626	0,0344	0,0282	0,0532	0,0407	0,0626	0,0250	0,0317	0,0304
4	0,0343	0,025	0,0093	0,0343	0,0406	0,0281	0,0187	0,0190	0,0176
5	0,0343	0,0281	0,0187	0,0218	0,0343	0,025	0,025	0,0208	0,0218
6	0,0250	0,0282	0,0470	0,0282	0,0344	0,0250	0,0250	0,0315	0,0300
7	0,0562	0,0156	0,0062	0,025	0,0406	0,0187	0,0187	0,0189	0,0175
8	0,0658	0,0282	0,0282	0,0626	0,0689	0,0438	0,0219	0,0555	0,0534
9	0,0468	0,025	0,025	0,0406	0,0343	0,0312	0,0281	0,0404	0,0387
10	0,0125	0,025	0,0156	0,0218	0,0343	0,0281	0,025	0,0125	0,0125

	Redes Neurais, 10 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0752	0,0031	0,0062	0,0344	0,0564	0,0250	0	0,0402	0,0464
2	0,0562	0,0125	0,0032	0,0437	0,0625	0,0187	0,0187	0,0520	0,0514
3	0,0752	0,0156	0,0032	0,0689	0,0626	0,0344	0,0219	0,0338	0,032
4	0,0406	0,0312	0,0032	0,0468	0,0406	0,0218	0,0187	0,0148	0,0128
5	0,0375	0,0156	0,0032	0,0406	0,0343	0,0218	0,0062	0,0145	0,0140
6	0,0438	0,0094	0,0062	0,0344	0,0376	0,0125	0,0062	0,0252	0,0268
7	0,0406	0,0218	0	0,0343	0,0343	0,0187	0,0093	0,0232	0,0222
8	0,0815	0,0094	0,0062	0,0752	0,0846	0,0219	0,0094	0,0534	0,0486
9	0,0406	0	0	0,05	0,0406	0,0187	0,0093	0,0297	0,0274
10	0,0375	0,0218	0,0032	0,0281	0,0312	0,025	0,0093	0,0274	0,0274

	Redes Neurais, 10 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0752	0,0031	0,0062	0,0344	0,0564	0,0250	0	0,0402	0,0464
2	0,0562	0,0125	0,0032	0,0437	0,0625	0,0187	0,0187	0,0520	0,0514
3	0,0752	0,0156	0,0032	0,0721	0,0626	0,0344	0,0219	0,0338	0,032
4	0,0406	0,0312	0	0,0437	0,0406	0,0218	0,0187	0,0148	0,0128
5	0,0375	0,0156	0	0,0437	0,0343	0,0218	0,0062	0,0145	0,0140
6	0,0438	0,0094	0,0062	0,0344	0,0376	0,0125	0,0062	0,0252	0,0268
7	0,0406	0,0218	0,0032	0,0343	0,0343	0,0187	0,0093	0,0232	0,0222
8	0,0815	0,0094	0,0062	0,0752	0,0846	0,0219	0,0094	0,0534	0,0486
9	0,0406	0,0156	0,0156	0,0468	0,0406	0,0187	0,0093	0,0297	0,0274
10	0,0375	0,0218	0,0032	0,0343	0,0312	0,025	0,0093	0,0274	0,0274

	Redes Neurais, 20 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0470	0,0219	0,0250	0,0376	0,0282	0,0470	0,0470	0,0402	0,0416
2	0,0375	0,025	0,0125	0,0437	0,0406	0,0437	0,0218	0,0303	0,0299
3	0,0501	0,0282	0,0219	0,0470	0,0501	0,0532	0,0282	0,0275	0,0256
4	0,0562	0,0375	0,0375	0,025	0,0375	0,0562	0,0218	0,0317	0,0288
5	0,0343	0,0312	0,0187	0,0281	0,0281	0,0343	0,0343	0,0187	0,0203
6	0,0250	0,0125	0,0062	0,0250	0,0282	0,0250	0,0250	0,0252	0,0221
7	0,0187	0,0062	0,0093	0,0187	0,0281	0,0281	0,0156	0,0168	0,0175
8	0,0689	0,0344	0,0188	0,0658	0,0595	0,0658	0,0125	0,0512	0,0470
9	0,05	0,0093	0,0093	0,0375	0,0406	0,05	0,05	0,0319	0,0290
10	0,0312	0,0187	0,025	0,0218	0,0281	0,0343	0,0093	0,0232	0,0191

	Redes Neurais, 20 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0470	0,0219	0,0250	0,0313	0,0282	0,0470	0,0470	0,0402	0,0416
2	0,0375	0,025	0,0125	0,0406	0,0406	0,0437	0,0218	0,0303	0,0299
3	0,0501	0,0282	0,0219	0,0407	0,0501	0,0532	0,0282	0,0275	0,0256
4	0,0562	0,0375	0,0375	0,025	0,0375	0,0562	0,0218	0,0317	0,0288
5	0,0343	0,0312	0,0187	0,025	0,0281	0,0343	0,0343	0,0187	0,0203
6	0,0250	0,0125	0,0062	0,0282	0,0282	0,0250	0,0250	0,0252	0,0221
7	0,0187	0,0062	0,0093	0,0218	0,0281	0,0281	0,0156	0,0168	0,0175
8	0,0689	0,0344	0,0188	0,0595	0,0595	0,0658	0,0125	0,0512	0,0470
9	0,05	0,0093	0,0093	0,0375	0,0406	0,05	0,05	0,0319	0,0290
10	0,0312	0,0187	0,025	0,0187	0,0281	0,0343	0,0093	0,0232	0,0191

Tabela A.29 – Erros de Classificação da Base *CMC*.

	Regressão Linear								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,3197	0,2721	0,2925	0,3061	0,3061	0,3197	0,3197	0,3502	0,3522
2	0,2857	0,2585	0,2857	0,3129	0,2993	0,2857	0,2857	0,3315	0,3710
3	0,3061	0,3061	0,3265	0,3333	0,3197	0,3061	0,3061	0,3842	0,3706
4	0,3648	0,3243	0,2905	0,3581	0,3581	0,3648	0,3648	0,3846	0,3636
5	0,3673	0,2993	0,2721	0,3333	0,3537	0,3673	0,3673	0,4064	0,3656
6	0,3605	0,3197	0,3061	0,3673	0,3605	0,3605	0,3197	0,4040	0,3654
7	0,2972	0,2364	0,2364	0,3243	0,3040	0,2972	0,2972	0,3724	0,3360
8	0,3537	0,2925	0,2993	0,3401	0,3469	0,3537	0,3537	0,3604	0,3238
9	0,2925	0,2653	0,2857	0,2925	0,2857	0,2925	0,2925	0,415	0,3754
10	0,3513	0,3378	0,3040	0,3513	0,3513	0,3513	0,3513	0,4093	0,3907

	Regressão Logística								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2993	0,2585	0,2585	0,2585	0,2721	0,2993	0,2993	0,3045	0,2995
2	0,3129	0,2653	0,2857	0,3061	0,3129	0,3129	0,3129	0,2989	0,3122
3	0,3197	0,3061	0,3061	0,3265	0,3265	0,3197	0,3197	0,3349	0,3127
4	0,3243	0,2972	0,2972	0,3378	0,3310	0,3243	0,3243	0,2769	0,2479
5	0,3265	0,2993	0,3061	0,3333	0,3333	0,3265	0,3265	0,3262	0,3303
6	0,3333	0,3061	0,2925	0,3673	0,3469	0,3333	0,3333	0,3232	0,3232
7	0,3243	0,2364	0,2702	0,3175	0,3378	0,3243	0,3243	0,2755	0,2786
8	0,3537	0,2993	0,3061	0,3537	0,3537	0,3537	0,3537	0,3096	0,2753
9	0,3129	0,2653	0,2721	0,3061	0,2993	0,3129	0,3129	0,31	0,2924
10	0,3648	0,3378	0,3378	0,3513	0,3648	0,3648	0,3648	0,3523	0,3403

	Redes Neurais, 3 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2925	0,2925	0,2925	0,2789	0,2653	0,2925	0,2925	0,2944	0,2955
2	0,2721	0,2857	0,2857	0,2517	0,2789	0,2721	0,2721	0,3043	0,3167
3	0,2721	0,2721	0,2721	0,2517	0,2585	0,2721	0,2721	0,2721	0,2721
4	0,3108	0,3108	0,3108	0,2905	0,2837	0,3108	0,3108	0,3108	0,3108
5	0,3129	0,3129	0,3061	0,2857	0,3061	0,3129	0,3129	0,3155	0,2995
6	0,2517	0,2721	0,2517	0,2448	0,2653	0,2448	0,2448	0,2517	0,2517
7	0,2635	0,2635	0,2635	0,2162	0,2432	0,2635	0,2635	0,2635	0,2635
8	0,2925	0,2925	0,2925	0,2721	0,2653	0,2925	0,2925	0,2925	0,2925
9	0,2517	0,3401	0,3197	0,2585	0,2925	0,2517	0,2517	0,285	0,285
10	0,3378	0,3108	0,3108	0,3108	0,2972	0,3378	0,3378	0,3108	0,3108

	Redes Neurais, 3 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2925	0,2925	0,2925	0,2585	0,2653	0,2925	0,2925	0,2944	0,2955
2	0,2721	0,2857	0,2857	0,2517	0,2789	0,2721	0,2721	0,3043	0,3167
3	0,2721	0,2721	0,2721	0,2517	0,2585	0,2721	0,2721	0,2721	0,2721
4	0,3108	0,3108	0,3108	0,2905	0,2837	0,3108	0,3108	0,3108	0,3108
5	0,3129	0,3129	0,3061	0,2857	0,3061	0,3129	0,3129	0,3155	0,2995
6	0,2517	0,2721	0,2517	0,2448	0,2653	0,2448	0,2448	0,2517	0,2517
7	0,2635	0,2635	0,2635	0,2162	0,2432	0,2635	0,2635	0,2635	0,2635
8	0,2925	0,2925	0,2925	0,2721	0,2653	0,2925	0,2925	0,2925	0,2925
9	0,2517	0,3401	0,3197	0,2585	0,2925	0,2517	0,2517	0,285	0,285
10	0,3378	0,3108	0,3108	0,3108	0,2972	0,3378	0,3378	0,3108	0,3108

	Redes Neurais, 10 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2857	0,2585	0,2585	0,2448	0,2585	0,2857	0,2857	0,2893	0,2793
2	0,2789	0,3129	0,3401	0,2653	0,2721	0,2789	0,2789	0,2789	0,2789
3	0,2517	0,2517	0,2517	0,2653	0,2721	0,2517	0,2517	0,2517	0,2517
4	0,2567	0,3040	0,3040	0,2635	0,2837	0,2567	0,2567	0,2205	0,2066
5	0,3061	0,3129	0,3197	0,2993	0,2721	0,3061	0,3061	0,3315	0,3315
6	0,2585	0,2585	0,2585	0,2517	0,2517	0,2585	0,2585	0,2585	0,2585
7	0,2837	0,2635	0,2837	0,2635	0,25	0,2837	0,2837	0,2837	0,2837
8	0,2789	0,3061	0,3061	0,2789	0,2517	0,2789	0,2789	0,2789	0,2789
9	0,2517	0,2517	0,2517	0,2448	0,2653	0,2517	0,2517	0,2517	0,2517
10	0,3310	0,3513	0,3513	0,3175	0,2837	0,3310	0,3310	0,3310	0,3310

	Redes Neurais, 10 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2857	0,2585	0,2585	0,2448	0,2585	0,2857	0,2857	0,2893	0,2793
2	0,2789	0,3129	0,3401	0,2653	0,2721	0,2789	0,2789	0,2789	0,2789
3	0,2517	0,2517	0,2517	0,2721	0,2721	0,2517	0,2517	0,2517	0,2517
4	0,2567	0,3040	0,3040	0,2702	0,2837	0,2567	0,2567	0,2205	0,2066
5	0,3061	0,3129	0,3197	0,2993	0,2721	0,3061	0,3061	0,3315	0,3315
6	0,2585	0,2585	0,2585	0,2653	0,2517	0,2585	0,2585	0,2585	0,2585
7	0,2837	0,2635	0,2837	0,2567	0,25	0,2837	0,2837	0,2837	0,2837
8	0,2789	0,3061	0,3061	0,2993	0,2517	0,2789	0,2789	0,2789	0,2789
9	0,2517	0,2517	0,2517	0,2585	0,2653	0,2517	0,2517	0,2517	0,2517
10	0,3310	0,3513	0,3513	0,3175	0,2837	0,3310	0,3310	0,3310	0,3310

	Redes Neurais, 20 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,3469	0,2312	0,2312	0,2653	0,2380	0,3469	0,3469	0,3197	0,3198
2	0,2993	0,2585	0,2585	0,2789	0,3401	0,2993	0,2993	0,3043	0,3031
3	0,3333	0,2721	0,3061	0,2789	0,3061	0,3333	0,3333	0,3152	0,3204
4	0,2837	0,3108	0,2770	0,2770	0,2702	0,2837	0,2837	0,2461	0,2314
5	0,3197	0,2993	0,3129	0,3197	0,2925	0,3197	0,3197	0,3208	0,3171
6	0,2517	0,2993	0,2993	0,2993	0,2925	0,2517	0,2517	0,2517	0,2517
7	0,2837	0,2567	0,2702	0,2635	0,2635	0,2837	0,2837	0,2837	0,2837
8	0,3061	0,2857	0,2857	0,2517	0,2653	0,3061	0,3061	0,3045	0,2753
9	0,2653	0,3129	0,2993	0,2925	0,2857	0,2653	0,2653	0,3	0,3122
10	0,3175	0,3310	0,3378	0,3175	0,3648	0,3175	0,3175	0,2953	0,2899

	Redes Neurais, 20 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,3469	0,2312	0,2312	0,2585	0,2380	0,3469	0,3469	0,3197	0,3198
2	0,2993	0,2585	0,2585	0,2789	0,3401	0,2993	0,2993	0,3043	0,3031
3	0,3333	0,2721	0,3061	0,2789	0,3061	0,3333	0,3333	0,3152	0,3204
4	0,2837	0,3108	0,2770	0,2635	0,2702	0,2837	0,2837	0,2461	0,2314
5	0,3197	0,2993	0,3129	0,3197	0,2925	0,3197	0,3197	0,3208	0,3171
6	0,2517	0,2993	0,2993	0,2925	0,2925	0,2517	0,2517	0,2517	0,2517
7	0,2837	0,2567	0,2702	0,2635	0,2635	0,2837	0,2837	0,2837	0,2837
8	0,3061	0,2857	0,2857	0,2653	0,2653	0,3061	0,3061	0,3045	0,2753
9	0,2653	0,3129	0,2993	0,2857	0,2857	0,2653	0,2653	0,3	0,3122
10	0,3175	0,3310	0,3378	0,3310	0,3648	0,3175	0,3175	0,2953	0,2899

Tabela A.30 – Erros de Classificação da Base *Connect4*.

	Regressão Linear								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,21	0,191	0,193	0,209	0,208	0,207	0,209	0,3040	0,2613
2	0,222	0,205	0,197	0,219	0,214	0,203	0,195	0,3153	0,2772
3	0,248	0,221	0,214	0,244	0,247	0,203	0,211	0,248	0,248
4	0,21	0,204	0,213	0,207	0,207	0,201	0,19	0,2472	0,2550
5	0,242	0,215	0,195	0,226	0,229	0,228	0,234	0,2682	0,2727
6	0,231	0,235	0,218	0,221	0,223	0,215	0,215	0,231	0,231
7	0,194	0,183	0,179	0,192	0,193	0,191	0,191	0,194	0,194
8	0,201	0,192	0,195	0,188	0,191	0,197	0,197	0,2896	0,2783
9	0,246	0,218	0,22	0,257	0,237	0,203	0,208	0,2538	0,2672
10	0,242	0,195	0,201	0,243	0,24	0,208	0,203	0,3294	0,2650

	Regressão Logística								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,206	0,197	0,192	0,206	0,209	0,206	0,206	0,206	0,206
2	0,214	0,204	0,193	0,218	0,214	0,202	0,194	0,214	0,214
3	0,209	0,203	0,209	0,216	0,213	0,203	0,219	0,209	0,209
4	0,199	0,19	0,182	0,2	0,201	0,198	0,195	0,199	0,199
5	0,221	0,208	0,185	0,218	0,218	0,221	0,221	0,221	0,221
6	0,216	0,224	0,212	0,208	0,204	0,204	0,212	0,216	0,216
7	0,186	0,178	0,179	0,179	0,18	0,19	0,19	0,186	0,186
8	0,189	0,19	0,197	0,178	0,182	0,195	0,195	0,2175	0,2185
9	0,232	0,225	0,19	0,226	0,226	0,209	0,209	0,232	0,232
10	0,198	0,2	0,188	0,203	0,201	0,2	0,201	0,198	0,198

	Redes Neurais, 3 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,215	0,194	0,188	0,1754	0,188	0,175	0,175	0,215	0,215
2	0,193	0,18	0,211	0,1903	0,204	0,193	0,193	0,193	0,193
3	0,198	0,188	0,194	0,1974	0,194	0,198	0,198	0,198	0,198
4	0,185	0,167	0,167	0,1826	0,181	0,185	0,185	0,185	0,185
5	0,201	0,201	0,201	0,1893	0,209	0,201	0,201	0,201	0,201
6	0,216	0,216	0,216	0,1881	0,213	0,216	0,216	0,216	0,216
7	0,186	0,162	0,162	0,1772	0,174	0,186	0,186	0,186	0,186
8	0,163	0,194	0,194	0,1915	0,162	0,163	0,163	0,163	0,163
9	0,182	0,182	0,182	0,1832	0,192	0,182	0,182	0,182	0,182
10	0,198	0,189	0,194	0,1886	0,198	0,197	0,213	0,1952	0,1934

	Redes Neurais, 3 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,19	0,184	0,197	0,177	0,17	0,179	0,179	0,19	0,19
2	0,218	0,188	0,192	0,173	0,195	0,221	0,221	0,218	0,218
3	0,214	0,192	0,181	0,187	0,194	0,214	0,214	0,214	0,214
4	0,185	0,187	0,176	0,158	0,183	0,185	0,185	0,185	0,185
5	0,2	0,2	0,2	0,189	0,198	0,2	0,2	0,2	0,2
6	0,209	0,212	0,212	0,192	0,211	0,209	0,209	0,209	0,209
7	0,178	0,175	0,175	0,164	0,168	0,178	0,178	0,178	0,178
8	0,178	0,186	0,197	0,16	0,173	0,178	0,178	0,178	0,178
9	0,206	0,178	0,178	0,184	0,215	0,206	0,206	0,206	0,206
10	0,218	0,19	0,19	0,174	0,184	0,182	0,182	0,2206	0,2236

	Redes Neurais, 10 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,169	0,178	0,185	0,1776	0,156	0,169	0,169	0,169	0,169
2	0,188	0,179	0,179	0,1976	0,163	0,188	0,188	0,1950	0,2013
3	0,189	0,176	0,188	0,1897	0,157	0,189	0,189	0,189	0,189
4	0,175	0,169	0,169	0,1882	0,151	0,18	0,18	0,175	0,175
5	0,186	0,185	0,185	0,1922	0,168	0,186	0,186	0,186	0,186
6	0,215	0,215	0,215	0,1931	0,179	0,215	0,215	0,215	0,215
7	0,171	0,158	0,173	0,1911	0,163	0,171	0,171	0,171	0,171
8	0,168	0,168	0,168	0,1829	0,147	0,168	0,168	0,168	0,168
9	0,173	0,173	0,173	0,1833	0,17	0,173	0,173	0,173	0,173
10	0,18	0,18	0,18	0,1787	0,173	0,18	0,18	0,18	0,18

	Redes Neurais, 10 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1887	0,1784	0,1899	0,1899	0,1693	0,1775	0,1775	0,1887	0,1887
2	0,2280	0,1873	0,1869	0,1705	0,2037	0,2243	0,2243	0,2280	0,2280
3	0,2190	0,1879	0,1764	0,1854	0,1916	0,2190	0,2190	0,2190	0,2190
4	0,1759	0,1856	0,1616	0,1464	0,1823	0,1759	0,1759	0,1759	0,1759
5	0,2015	0,2015	0,2015	0,1780	0,2083	0,2015	0,2015	0,2015	0,2015
6	0,2115	0,2098	0,2098	0,2014	0,2113	0,2115	0,2115	0,2115	0,2115
7	0,1833	0,1732	0,1732	0,1727	0,1652	0,1833	0,1833	0,1833	0,1833
8	0,1855	0,1853	0,1888	0,1671	0,1729	0,1855	0,1855	0,1855	0,1855
9	0,2041	0,1768	0,1768	0,1812	0,2189	0,2041	0,2041	0,2041	0,2041
10	0,2206	0,1995	0,1953	0,1788	0,2004	0,1915	0,1915	0,2155	0,2145

	Redes Neurais, 20 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1644	0,1746	0,1880	0,1808	0,158	0,1644	0,1644	0,1644	0,1644
2	0,1863	0,1773	0,1773	0,1914	0,154	0,1863	0,1863	0,1978	0,2024
3	0,1950	0,1747	0,1873	0,2007	0,165	0,1950	0,1950	0,1950	0,1950
4	0,1640	0,1726	0,1726	0,1848	0,162	0,1808	0,1808	0,1640	0,1640
5	0,1903	0,1798	0,1798	0,1963	0,175	0,1903	0,1903	0,1903	0,1903
6	0,2250	0,2250	0,2250	0,1850	0,185	0,2250	0,2250	0,2250	0,2250
7	0,1729	0,1441	0,1669	0,1746	0,16	0,1729	0,1729	0,1729	0,1729
8	0,1566	0,1566	0,1566	0,1916	0,165	0,1566	0,1566	0,1566	0,1566
9	0,1719	0,1719	0,1719	0,1846	0,165	0,1719	0,1719	0,1719	0,1719
10	0,1841	0,1841	0,1841	0,1821	0,186	0,1841	0,1841	0,1841	0,1841

	Redes Neurais, 20 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2004	0,1903	0,2030	0,1867	0,1671	0,1820	0,1820	0,2004	0,2004
2	0,2166	0,1842	0,2088	0,1784	0,2024	0,2253	0,2253	0,2166	0,2166
3	0,2212	0,1856	0,1712	0,1926	0,1873	0,2212	0,2212	0,2212	0,2212
4	0,1849	0,1759	0,1768	0,1595	0,1754	0,1849	0,1849	0,1849	0,1849
5	0,1948	0,1948	0,1948	0,1840	0,2020	0,1948	0,1948	0,1948	0,1948
6	0,2081	0,2055	0,2055	0,1882	0,2129	0,2081	0,2081	0,2081	0,2081
7	0,1874	0,1721	0,1721	0,1658	0,1650	0,1874	0,1874	0,1874	0,1874
8	0,1860	0,1805	0,1956	0,1496	0,1759	0,1860	0,1860	0,1860	0,1860
9	0,2038	0,1728	0,1728	0,1719	0,2140	0,2038	0,2038	0,2038	0,2038
10	0,2096	0,1952	0,1836	0,1779	0,1875	0,1843	0,1843	0,2223	0,2238

Tabela A.31 – Erros de Classificação da Base *German*.

	Regressão Linear								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,29	0,23	0,24	0,27	0,31	0,25	0,25	0,344	0,3333
2	0,34	0,25	0,26	0,27	0,27	0,24	0,24	0,256	0,2333
3	0,25	0,29	0,29	0,25	0,31	0,26	0,26	0,3464	0,3376
4	0,3	0,28	0,32	0,25	0,29	0,25	0,25	0,3089	0,2945
5	0,26	0,25	0,25	0,2	0,19	0,25	0,25	0,2519	0,3116
6	0,32	0,32	0,32	0,34	0,36	0,3	0,27	0,3606	0,3402
7	0,3	0,23	0,23	0,28	0,28	0,27	0,27	0,2424	0,3231
8	0,22	0,22	0,22	0,24	0,23	0,22	0,22	0,22	0,22
9	0,33	0,27	0,27	0,35	0,36	0,33	0,33	0,32	0,3733
10	0,29	0,29	0,3	0,24	0,25	0,24	0,26	0,275	0,3428

	Regressão Logística								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,29	0,25	0,25	0,27	0,3	0,24	0,24	0,272	0,3266
2	0,22	0,22	0,22	0,22	0,2	0,22	0,22	0,22	0,22
3	0,26	0,26	0,26	0,22	0,23	0,26	0,26	0,26	0,26
4	0,24	0,27	0,32	0,25	0,25	0,24	0,24	0,3089	0,3082
5	0,24	0,24	0,24	0,23	0,21	0,23	0,23	0,2519	0,2727
6	0,3	0,34	0,34	0,27	0,28	0,3	0,3	0,3	0,3
7	0,24	0,24	0,24	0,27	0,27	0,24	0,24	0,24	0,24
8	0,21	0,21	0,21	0,23	0,21	0,21	0,21	0,21	0,21
9	0,31	0,31	0,31	0,29	0,25	0,31	0,31	0,272	0,3066
10	0,26	0,29	0,29	0,25	0,26	0,22	0,23	0,26	0,26

	Redes Neurais, 3 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,29	0,23	0,27	0,27	0,26	0,25	0,29	0,328	0,36
2	0,25	0,3	0,3	0,22	0,24	0,25	0,25	0,232	0,2333
3	0,26	0,3	0,34	0,21	0,23	0,3	0,3	0,2362	0,2467
4	0,21	0,25	0,24	0,21	0,26	0,25	0,25	0,3008	0,3287
5	0,25	0,25	0,25	0,24	0,23	0,25	0,25	0,2755	0,2987
6	0,3	0,32	0,32	0,31	0,3	0,28	0,28	0,3442	0,3442
7	0,28	0,28	0,28	0,25	0,31	0,25	0,25	0,3181	0,3414
8	0,23	0,23	0,23	0,21	0,18	0,26	0,26	0,2436	0,2436
9	0,25	0,3	0,3	0,29	0,3	0,25	0,25	0,32	0,3466
10	0,26	0,28	0,28	0,24	0,25	0,23	0,24	0,26	0,26

	Redes Neurais, 3 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,29	0,23	0,27	0,28	0,26	0,25	0,29	0,328	0,36
2	0,25	0,3	0,3	0,22	0,24	0,25	0,25	0,232	0,2333
3	0,26	0,3	0,34	0,21	0,23	0,3	0,3	0,2362	0,2467
4	0,21	0,25	0,24	0,21	0,26	0,25	0,25	0,3008	0,3287
5	0,25	0,25	0,25	0,24	0,23	0,25	0,25	0,2755	0,2987
6	0,3	0,32	0,32	0,31	0,3	0,28	0,28	0,3442	0,3442
7	0,28	0,28	0,28	0,25	0,31	0,25	0,25	0,3181	0,3414
8	0,23	0,23	0,23	0,21	0,18	0,26	0,26	0,2436	0,2436
9	0,25	0,3	0,3	0,29	0,3	0,25	0,25	0,32	0,3466
10	0,26	0,28	0,28	0,24	0,25	0,23	0,24	0,26	0,26

	Redes Neurais, 10 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,28	0,25	0,26	0,24	0,3	0,29	0,29	0,352	0,3866
2	0,27	0,26	0,25	0,23	0,25	0,23	0,23	0,288	0,288
3	0,27	0,3	0,3	0,23	0,23	0,28	0,26	0,27	0,27
4	0,26	0,25	0,26	0,21	0,25	0,25	0,25	0,3008	0,3150
5	0,28	0,26	0,26	0,24	0,24	0,28	0,28	0,28	0,28
6	0,33	0,31	0,31	0,31	0,3	0,32	0,32	0,3852	0,3852
7	0,28	0,28	0,28	0,27	0,21	0,28	0,28	0,2878	0,3170
8	0,21	0,23	0,23	0,24	0,22	0,22	0,29	0,2605	0,2898
9	0,3	0,24	0,29	0,26	0,37	0,3	0,3	0,312	0,34
10	0,25	0,26	0,26	0,24	0,26	0,25	0,25	0,25	0,25

	Redes Neurais, 10 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,28	0,25	0,26	0,26	0,3	0,29	0,29	0,352	0,3866
2	0,27	0,26	0,25	0,25	0,25	0,23	0,23	0,288	0,288
3	0,27	0,3	0,3	0,23	0,23	0,28	0,26	0,27	0,27
4	0,26	0,25	0,26	0,23	0,25	0,25	0,25	0,3008	0,3150
5	0,28	0,26	0,26	0,23	0,24	0,28	0,28	0,28	0,28
6	0,33	0,31	0,31	0,28	0,3	0,32	0,32	0,3852	0,3852
7	0,28	0,28	0,28	0,27	0,21	0,28	0,28	0,2878	0,3170
8	0,21	0,23	0,23	0,21	0,22	0,22	0,29	0,2605	0,2898
9	0,3	0,24	0,29	0,29	0,37	0,3	0,3	0,312	0,34
10	0,25	0,26	0,26	0,25	0,26	0,25	0,25	0,25	0,25

	Redes Neurais, 20 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,25	0,27	0,26	0,27	0,27	0,26	0,28	0,25	0,25
2	0,24	0,24	0,24	0,24	0,2	0,24	0,24	0,24	0,24
3	0,21	0,21	0,21	0,21	0,22	0,21	0,21	0,21	0,21
4	0,25	0,25	0,25	0,21	0,26	0,27	0,27	0,2764	0,2945
5	0,25	0,23	0,23	0,21	0,23	0,25	0,25	0,3070	0,3246
6	0,3	0,31	0,31	0,28	0,33	0,3	0,3	0,3	0,3
7	0,28	0,28	0,29	0,3	0,29	0,28	0,28	0,3030	0,3292
8	0,21	0,22	0,24	0,21	0,22	0,21	0,21	0,2521	0,2681
9	0,28	0,26	0,26	0,28	0,27	0,32	0,28	0,288	0,3066
10	0,25	0,26	0,29	0,25	0,25	0,25	0,25	0,2833	0,2857

	Redes Neurais, 20 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,25	0,27	0,26	0,25	0,27	0,26	0,28	0,25	0,25
2	0,24	0,24	0,24	0,25	0,2	0,24	0,24	0,24	0,24
3	0,21	0,21	0,21	0,21	0,22	0,21	0,21	0,21	0,21
4	0,25	0,25	0,25	0,22	0,26	0,27	0,27	0,2764	0,2945
5	0,25	0,23	0,23	0,21	0,23	0,25	0,25	0,3070	0,3246
6	0,3	0,31	0,31	0,33	0,33	0,3	0,3	0,3	0,3
7	0,28	0,28	0,29	0,29	0,29	0,28	0,28	0,3030	0,3292
8	0,21	0,22	0,24	0,21	0,22	0,21	0,21	0,2521	0,2681
9	0,28	0,26	0,26	0,31	0,27	0,32	0,28	0,288	0,3066
10	0,25	0,26	0,29	0,25	0,25	0,25	0,25	0,2833	0,2857

Tabela A.32 – Erros de Classificação da Base *Magic*.

	Regressão Linear								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,3506	0,1782	0,1671	0,3506	0,3512	0,2076	0,1971	0,2488	0,3034
2	0,3217	0,1514	0,1498	0,3338	0,3307	0,1750	0,1771	0,3072	0,3022
3	0,2087	0,1813	0,1740	0,2618	0,2644	0,1924	0,1924	0,3070	0,2708
4	0,3449	0,1803	0,1629	0,3449	0,3449	0,2008	0,1777	0,3069	0,2852
5	0,3380	0,1671	0,1472	0,3396	0,3401	0,2018	0,1945	0,2232	0,3016
6	0,3580	0,1735	0,1650	0,3580	0,3580	0,2034	0,1940	0,3352	0,2965
7	0,3191	0,1787	0,1477	0,3207	0,3238	0,1892	0,1735	0,2350	0,3041
8	0,3375	0,1782	0,1740	0,3349	0,3359	0,1987	0,1924	0,2408	0,3005
9	0,3475	0,1703	0,1656	0,3475	0,3475	0,2071	0,1945	0,3046	0,2790
10	0,2770	0,1603	0,1519	0,2697	0,2786	0,1940	0,1792	0,2218	0,2922

	Regressão Logística								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2134	0,1698	0,1572	0,2139	0,2139	0,2039	0,1940	0,194	0,1888
2	0,2008	0,1535	0,1451	0,2003	0,2008	0,1808	0,1682	0,1699	0,1654
3	0,2071	0,1729	0,1593	0,2071	0,2071	0,1945	0,1824	0,1885	0,1932
4	0,2118	0,1677	0,1409	0,2113	0,2113	0,1940	0,1777	0,1914	0,1945
5	0,2176	0,1719	0,1519	0,2145	0,2160	0,1982	0,1934	0,1862	0,1883
6	0,2166	0,1671	0,1577	0,2166	0,2160	0,1992	0,1876	0,1949	0,1919
7	0,2013	0,1671	0,1508	0,2013	0,2018	0,1845	0,1671	0,1858	0,1945
8	0,2092	0,1771	0,1661	0,2082	0,2071	0,1955	0,1919	0,1903	0,1874
9	0,2197	0,1745	0,1640	0,2181	0,2181	0,2071	0,1940	0,1893	0,1782
10	0,2003	0,1572	0,1472	0,2003	0,1997	0,1866	0,1724	0,1813	0,1863

	Redes Neurais, 3 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1566	0,1377	0,1272	0,1545	0,1561	0,1540	0,1529	0,1566	0,1566
2	0,1356	0,1277	0,1240	0,1319	0,1377	0,1293	0,1340	0,1356	0,1356
3	0,1635	0,1414	0,1414	0,1608	0,1635	0,1593	0,1614	0,1662	0,1663
4	0,1535	0,1403	0,1293	0,1487	0,1493	0,1545	0,1545	0,1535	0,1535
5	0,1498	0,1351	0,1382	0,1493	0,1550	0,1393	0,1414	0,1498	0,1498
6	0,1582	0,1503	0,1419	0,1593	0,1645	0,1608	0,1566	0,1584	0,1596
7	0,1503	0,1424	0,1398	0,1487	0,1529	0,1440	0,1440	0,1556	0,1556
8	0,1577	0,1650	0,1508	0,1598	0,1577	0,1519	0,1519	0,1577	0,1577
9	0,1514	0,1456	0,1366	0,1466	0,1472	0,1477	0,1461	0,1453	0,1441
10	0,1388	0,1314	0,1209	0,1403	0,1419	0,1424	0,1435	0,1449	0,1451

	Redes Neurais, 3 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1566	0,1377	0,1272	0,1540	0,1561	0,1540	0,1529	0,1566	0,1566
2	0,1356	0,1277	0,1240	0,1335	0,1377	0,1293	0,1340	0,1356	0,1356
3	0,1635	0,1414	0,1414	0,1603	0,1635	0,1593	0,1614	0,1662	0,1663
4	0,1535	0,1403	0,1293	0,1508	0,1493	0,1545	0,1545	0,1535	0,1535
5	0,1498	0,1351	0,1382	0,1493	0,1550	0,1393	0,1414	0,1498	0,1498
6	0,1582	0,1503	0,1419	0,1593	0,1645	0,1608	0,1566	0,1584	0,1596
7	0,1503	0,1424	0,1398	0,1487	0,1529	0,1440	0,1440	0,1556	0,1556
8	0,1577	0,1650	0,1508	0,1598	0,1577	0,1519	0,1519	0,1577	0,1577
9	0,1514	0,1456	0,1366	0,1451	0,1472	0,1477	0,1461	0,1453	0,1441
10	0,1388	0,1314	0,1209	0,1393	0,1419	0,1424	0,1435	0,1449	0,1451

	Redes Neurais, 10 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1671	0,1451	0,1424	0,1671	0,1640	0,1545	0,1461	0,166	0,1613
2	0,1466	0,1372	0,1209	0,1419	0,1456	0,1466	0,1466	0,1393	0,1364
3	0,1561	0,1451	0,1398	0,1608	0,1635	0,1650	0,1614	0,1674	0,1679
4	0,1550	0,1440	0,1388	0,1577	0,1577	0,1556	0,1550	0,1544	0,1517
5	0,1671	0,1451	0,1319	0,1661	0,1677	0,1556	0,1409	0,1553	0,1542
6	0,1740	0,1477	0,1382	0,1708	0,1703	0,1740	0,1740	0,1678	0,1672
7	0,1577	0,1498	0,1419	0,1566	0,1566	0,1535	0,1466	0,1693	0,1746
8	0,1656	0,1498	0,1472	0,1640	0,1692	0,1687	0,1556	0,1619	0,1595
9	0,1629	0,1508	0,1419	0,1545	0,1545	0,1629	0,1629	0,1556	0,1529
10	0,1629	0,1377	0,1319	0,1582	0,1577	0,1587	0,1561	0,1592	0,1568

	Redes Neurais, 10 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1671	0,1451	0,1424	0,1677	0,1640	0,1545	0,1461	0,166	0,1613
2	0,1466	0,1372	0,1209	0,1419	0,1456	0,1466	0,1466	0,1393	0,1364
3	0,1561	0,1451	0,1398	0,1619	0,1635	0,1650	0,1614	0,1674	0,1679
4	0,1550	0,1440	0,1388	0,1587	0,1577	0,1556	0,1550	0,1544	0,1517
5	0,1671	0,1451	0,1319	0,1650	0,1677	0,1556	0,1409	0,1553	0,1542
6	0,1740	0,1477	0,1382	0,1682	0,1703	0,1740	0,1740	0,1678	0,1672
7	0,1577	0,1498	0,1419	0,1582	0,1566	0,1535	0,1466	0,1693	0,1746
8	0,1656	0,1498	0,1472	0,1677	0,1692	0,1687	0,1556	0,1619	0,1595
9	0,1629	0,1508	0,1419	0,1550	0,1545	0,1629	0,1629	0,1556	0,1529
10	0,1629	0,1377	0,1319	0,1603	0,1577	0,1587	0,1561	0,1592	0,1568

	Redes Neurais, 20 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1871	0,1519	0,1388	0,1698	0,1756	0,1698	0,1482	0,1644	0,1626
2	0,1645	0,1293	0,1361	0,1572	0,1666	0,1414	0,1361	0,1486	0,1488
3	0,1761	0,1519	0,1414	0,1787	0,1840	0,1650	0,1729	0,1779	0,1798
4	0,1792	0,1466	0,1482	0,1650	0,1750	0,1698	0,1535	0,1668	0,1635
5	0,1719	0,1503	0,1382	0,1756	0,1808	0,1608	0,1582	0,1633	0,1639
6	0,1882	0,1661	0,1577	0,1792	0,1887	0,1792	0,1708	0,1747	0,1721
7	0,1745	0,1514	0,1388	0,1556	0,1635	0,1635	0,1545	0,1794	0,1814
8	0,1750	0,1550	0,1498	0,1671	0,1756	0,1782	0,1556	0,1684	0,1708
9	0,1887	0,1529	0,1508	0,1756	0,1940	0,1645	0,1561	0,1687	0,1637
10	0,1798	0,1435	0,1382	0,1635	0,1813	0,1761	0,1561	0,1711	0,1685

	Redes Neurais, 20 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,1871	0,1519	0,1388	0,1719	0,1756	0,1698	0,1482	0,1644	0,1626
2	0,1645	0,1293	0,1361	0,1519	0,1666	0,1414	0,1361	0,1486	0,1488
3	0,1761	0,1519	0,1414	0,1792	0,1840	0,1650	0,1729	0,1779	0,1798
4	0,1792	0,1466	0,1482	0,1587	0,1750	0,1698	0,1535	0,1668	0,1635
5	0,1719	0,1503	0,1382	0,1719	0,1808	0,1608	0,1582	0,1633	0,1639
6	0,1882	0,1661	0,1577	0,1819	0,1887	0,1792	0,1708	0,1747	0,1721
7	0,1745	0,1514	0,1388	0,1587	0,1635	0,1635	0,1545	0,1794	0,1814
8	0,1750	0,1550	0,1498	0,1708	0,1756	0,1782	0,1556	0,1684	0,1708
9	0,1887	0,1529	0,1508	0,1745	0,1940	0,1645	0,1561	0,1687	0,1637
10	0,1798	0,1435	0,1382	0,1687	0,1813	0,1761	0,1561	0,1711	0,1685

Tabela A.33 – Erros de Classificação da Base *Mushroom*.

	Regressão Linear								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0	0	0	0	0,4519	0	0	0	0
2	0	0	0	0	0,4618	0	0	0	0
3	0	0	0	0	0,4920	0	0	0	0
4	0	0	0	0	0,5036	0	0	0	0
5	0	0	0	0	0,4944	0	0	0	0
6	0	0	0	0	0,4692	0	0	0	0
7	0	0	0	0	0,4519	0	0	0	0
8	0	0	0	0	0,4920	0	0	0	0
9	0	0	0	0	0,5061	0	0	0	0
10	0	0	0	0	0,4969	0	0	0	0

	Regressão Logística								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0	0	0	0	0,4519	0	0	0	0
2	0	0	0	0	0,4618	0	0	0	0
3	0	0	0	0	0,4920	0	0	0	0
4	0	0	0	0,0012	0,5036	0	0	0	0
5	0	0	0	0,0012	0,4944	0	0	0	0
6	0	0	0	0	0,4692	0	0	0	0
7	0	0	0	0	0,4519	0	0	0	0
8	0	0	0	0	0,4920	0	0	0	0
9	0	0	0	0,0012	0,5061	0	0	0	0
10	0	0	0	0,0012	0,4969	0	0	0	0

	Redes Neurais, 3 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0012	0,0012	0	0,0012	0,0049	0	0,0012	0	0
2	0,0024	0	0	0	0,0036	0	0	0	0
3	0,0061	0	0	0,0012	0,0049	0	0	0,0016	0,0006
4	0,0024	0	0	0	0,0049	0	0	0	0
5	0,0061	0	0	0,0036	0,0049	0	0	0	0
6	0,0036	0	0	0,0036	0,0024	0	0	0	0
7	0,0110	0,0012	0,0012	0,0012	0,0024	0	0	0,0007	0,0005
8	0	0	0	0	0,0024	0	0	0	0
9	0,0086	0	0	0,0012	0,0024	0	0	0	0
10	0,0012	0	0	0,0012	0,0024	0	0	0	0

	Redes Neurais, 3 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0012	0,0012	0	0,0012	0,0049	0	0,0012	0	0
2	0,0024	0	0	0	0,0036	0	0	0	0
3	0,0061	0	0	0	0,0049	0	0	0,0016	0,0006
4	0,0024	0	0	0	0,0049	0	0	0	0
5	0,0061	0	0	0,0036	0,0049	0	0	0	0
6	0,0036	0	0	0,0036	0,0024	0	0	0	0
7	0,0110	0,0012	0,0012	0,0012	0,0024	0	0	0,0007	0,0005
8	0	0	0	0	0,0024	0	0	0	0
9	0,0086	0	0	0,0012	0,0024	0	0	0	0
10	0,0012	0	0	0,0012	0,0024	0	0	0	0

	Redes Neurais, 10 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0012	0,0012	0	0,0012	0	0	0	0,0007	0,0005
2	0	0	0	0	0,0012	0	0	0	0
3	0,0012	0	0	0,0012	0,0024	0	0,0012	0,0008	0,0006
4	0,0012	0,0012	0	0	0,0049	0	0	0,0008	0,0006
5	0,0024	0,0012	0	0,0024	0,0024	0	0	0	0
6	0,0049	0,0049	0,0012	0,0049	0,0061	0,0012	0,0012	0	0
7	0,0049	0,0024	0,0012	0,0012	0,0012	0	0	0	0
8	0,0012	0,0012	0,0012	0	0,0012	0	0	0	0
9	0,0012	0	0	0,0012	0,0012	0	0	0	0
10	0,0024	0,0012	0	0,0012	0,0012	0	0	0	0

	Redes Neurais, 10 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0012	0,0012	0	0,0012	0	0	0	0,0007	0,0005
2	0	0	0	0	0,0012	0	0	0	0
3	0,0012	0	0	0,0012	0,0024	0	0,0012	0,0008	0,0006
4	0,0012	0,0012	0	0	0,0049	0	0	0,0008	0,0006
5	0,0024	0,0012	0	0,0024	0,0024	0	0	0	0
6	0,0049	0,0049	0,0012	0,0049	0,0061	0,0012	0,0012	0	0
7	0,0049	0,0024	0,0012	0,0024	0,0012	0	0	0	0
8	0,0012	0,0012	0,0012	0	0,0012	0	0	0	0
9	0,0012	0	0	0,0012	0,0012	0	0	0	0
10	0,0024	0,0012	0	0,0012	0,0012	0	0	0	0

	Redes Neurais, 20 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0024	0	0	0	0,0012	0	0	0,0007	0,0005
2	0,0024	0	0	0	0	0	0	0	0
3	0,0012	0	0	0	0	0	0,0012	0	0
4	0	0	0	0	0,5036	0	0	0	0
5	0,0012	0	0	0,0024	0,0012	0	0	0	0
6	0,0036	0	0	0,0036	0,0036	0,0012	0	0	0
7	0,0036	0,0024	0	0,0012	0,0061	0	0	0,0007	0,0007
8	0	0	0	0	0	0	0	0	0
9	0,0012	0	0	0,0012	0,0036	0	0	0	0
10	0,0024	0	0,0012	0,0012	0,0012	0	0	0	0

	Redes Neurais, 20 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,0024	0	0	0,0012	0,0012	0	0	0,0007	0,0005
2	0,0024	0	0	0	0	0	0	0	0
3	0,0012	0	0	0	0	0	0,0012	0	0
4	0	0	0	0	0,5036	0	0	0	0
5	0,0012	0	0	0,0024	0,0012	0	0	0	0
6	0,0036	0	0	0,0036	0,0036	0,0012	0	0	0
7	0,0036	0,0024	0	0,0012	0,0061	0	0	0,0007	0,0007
8	0	0	0	0	0	0	0	0	0
9	0,0012	0	0	0,0012	0,0036	0	0	0	0
10	0,0024	0	0,0012	0,0012	0,0012	0	0	0	0

Tabela A.34 – Erros de Classificação da Base *Solar Flare*.

	Regressão Linear								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2391	0,2753	0,2753	0,2391	0,2318	0,2753	0,2826	0,2391	0,2391
2	0,2158	0,1726	0,1798	0,2014	0,2086	0,1726	0,1942	0,2158	0,2158
3	0,1582	0,1582	0,1582	0,1654	0,1654	0,1366	0,1654	0,1582	0,1582
4	0,2446	0,2230	0,2230	0,2374	0,2374	0,2230	0,2374	0,2446	0,2446
5	0,2302	0,1726	0,1942	0,1942	0,2230	0,1870	0,1942	0,2302	0,2302
6	0,1223	0,1366	0,1366	0,1294	0,1294	0,1294	0,1294	0,1223	0,1223
7	0,1870	0,1870	0,1870	0,1798	0,1726	0,1870	0,1870	0,1870	0,1870
8	0,1870	0,1654	0,1654	0,1726	0,1870	0,1582	0,1510	0,1870	0,1870
9	0,1582	0,1942	0,1942	0,1582	0,1654	0,2014	0,2086	0,1582	0,1582
10	0,1942	0,1438	0,1438	0,1942	0,1798	0,1366	0,1366	0,1942	0,1942

	Regressão Logística								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2536	0,2753	0,2826	0,2753	0,2608	0,2753	0,2826	0,2536	0,2536
2	0,2086	0,1726	0,1726	0,1942	0,1942	0,1726	0,2014	0,2086	0,2086
3	0,1366	0,1582	0,1510	0,1366	0,1294	0,1366	0,1366	0,1366	0,1366
4	0,2230	0,2230	0,2230	0,2302	0,2230	0,2302	0,2302	0,2230	0,2230
5	0,1798	0,1654	0,1942	0,1798	0,1798	0,1942	0,1942	0,1798	0,1798
6	0,1294	0,1294	0,1366	0,1223	0,1223	0,1294	0,1294	0,1294	0,1294
7	0,2086	0,2014	0,2014	0,1798	0,1870	0,2086	0,2086	0,2086	0,2086
8	0,1582	0,1510	0,1510	0,1582	0,1510	0,1582	0,1582	0,1582	0,1582
9	0,2014	0,2014	0,2014	0,1942	0,2014	0,2014	0,2014	0,2014	0,2014
10	0,1294	0,1582	0,1438	0,1366	0,1366	0,1294	0,1294	0,1294	0,1294

	Redes Neurais, 3 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2391	0,2391	0,2391	0,2608	0,2753	0,2391	0,2391	0,2391	0,2391
2	0,2158	0,1870	0,1726	0,1942	0,1942	0,1798	0,2014	0,1749	0,1679
3	0,1438	0,1366	0,1366	0,1366	0,1438	0,1438	0,1438	0,1438	0,1438
4	0,2302	0,1870	0,2014	0,2230	0,2517	0,2086	0,2230	0,2302	0,2302
5	0,2014	0,1870	0,2014	0,1870	0,1942	0,1870	0,2014	0,1808	0,1761
6	0,1798	0,1294	0,1294	0,1438	0,1582	0,1582	0,1223	0,1343	0,1259
7	0,2086	0,2014	0,2086	0,2014	0,1942	0,1726	0,2086	0,2014	0,2014
8	0,1870	0,1654	0,1654	0,1654	0,1726	0,1582	0,1582	0,1654	0,1679
9	0,2158	0,1870	0,1870	0,1870	0,1870	0,1870	0,1870	0,1870	0,1870
10	0,1726	0,1510	0,1798	0,1438	0,1438	0,1366	0,1366	0,1471	0,1457

	Redes Neurais, 3 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2391	0,2391	0,2391	0,2681	0,2753	0,2391	0,2391	0,2391	0,2391
2	0,2158	0,1870	0,1726	0,2014	0,1942	0,1798	0,2014	0,1749	0,1679
3	0,1438	0,1366	0,1366	0,1294	0,1438	0,1438	0,1438	0,1438	0,1438
4	0,2302	0,1870	0,2014	0,2230	0,2517	0,2086	0,2230	0,2302	0,2302
5	0,2014	0,1870	0,2014	0,1942	0,1942	0,1870	0,2014	0,1808	0,1761
6	0,1798	0,1294	0,1294	0,1438	0,1582	0,1582	0,1223	0,1343	0,1259
7	0,2014	0,1942	0,1870	0,1942	0,1942	0,1726	0,2086	0,2014	0,2014
8	0,1438	0,1654	0,1654	0,1654	0,1726	0,1582	0,1582	0,1654	0,1679
9	0,2158	0,1870	0,1870	0,2086	0,2014	0,1942	0,2086	0,2158	0,2158
10	0,1726	0,1510	0,1798	0,1438	0,1726	0,1726	0,1726	0,1471	0,1457

	Redes Neurais, 10 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2536	0,2753	0,2753	0,2391	0,2463	0,2536	0,2536	0,2536	0,2536
2	0,2086	0,2014	0,1870	0,2086	0,2158	0,1798	0,1942	0,1787	0,1705
3	0,1294	0,1294	0,1294	0,1366	0,1582	0,1294	0,1294	0,1294	0,1294
4	0,2158	0,2086	0,2302	0,2086	0,2158	0,2446	0,2446	0,2158	0,2158
5	0,1654	0,1654	0,1654	0,1798	0,1870	0,1654	0,1654	0,1771	0,1736
6	0,1582	0,1223	0,1223	0,1510	0,1726	0,1366	0,1223	0,1194	0,1158
7	0,1942	0,2014	0,2086	0,1798	0,1798	0,1798	0,2014	0,1942	0,1942
8	0,1582	0,1510	0,1510	0,1582	0,1438	0,1366	0,1366	0,1470	0,1481
9	0,1870	0,1870	0,1870	0,1942	0,1798	0,1870	0,1870	0,1870	0,1870
10	0,1582	0,1438	0,1438	0,1366	0,1582	0,1582	0,1582	0,1582	0,1582

	Redes Neurais, 10 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2536	0,2753	0,2753	0,2391	0,2463	0,2536	0,2536	0,2536	0,2536
2	0,2086	0,2014	0,1870	0,2086	0,2158	0,1798	0,1942	0,1787	0,1705
3	0,1294	0,1294	0,1294	0,1366	0,1582	0,1294	0,1294	0,1294	0,1294
4	0,2158	0,2086	0,2302	0,2086	0,2158	0,2446	0,2446	0,2158	0,2158
5	0,1654	0,1654	0,1654	0,1870	0,1870	0,1654	0,1654	0,1771	0,1736
6	0,1582	0,1223	0,1223	0,1438	0,1726	0,1366	0,1223	0,1194	0,1158
7	0,1942	0,2014	0,2086	0,1870	0,1798	0,1798	0,2014	0,1942	0,1942
8	0,1582	0,1510	0,1510	0,1438	0,1438	0,1366	0,1366	0,1470	0,1481
9	0,1870	0,1870	0,1870	0,1942	0,1798	0,1870	0,1870	0,1870	0,1870
10	0,1582	0,1438	0,1438	0,1366	0,1438	0,1582	0,1582	0,1582	0,1582

	Redes Neurais, 20 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2608	0,2608	0,2536	0,2463	0,2753	0,2608	0,2608	0,2608	0,2608
2	0,1942	0,1942	0,1942	0,1942	0,1798	0,1942	0,1942	0,1711	0,1705
3	0,1438	0,1654	0,1582	0,1223	0,1294	0,1438	0,1438	0,1438	0,1438
4	0,2230	0,2230	0,2230	0,2230	0,2014	0,2230	0,2230	0,2230	0,2230
5	0,1654	0,1942	0,1870	0,1726	0,1798	0,1942	0,1870	0,1881	0,1885
6	0,1366	0,1151	0,1151	0,1510	0,1438	0,1438	0,1151	0,1194	0,1183
7	0,1942	0,2014	0,1798	0,2086	0,2014	0,1942	0,1942	0,1942	0,1942
8	0,1870	0,1510	0,1582	0,1438	0,1366	0,1582	0,1582	0,1544	0,1530
9	0,2302	0,2086	0,2086	0,2014	0,2014	0,2302	0,2302	0,2302	0,2302
10	0,1223	0,1438	0,1366	0,1294	0,1294	0,1223	0,1366	0,1471	0,1432

	Redes Neurais, 20 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2608	0,2608	0,2536	0,2536	0,2753	0,2608	0,2608	0,2608	0,2608
2	0,1942	0,1942	0,1942	0,1942	0,1798	0,1942	0,1942	0,1711	0,1705
3	0,1438	0,1654	0,1582	0,1223	0,1294	0,1438	0,1438	0,1438	0,1438
4	0,2230	0,2230	0,2230	0,2230	0,2014	0,2230	0,2230	0,2230	0,2230
5	0,1654	0,1942	0,1870	0,1726	0,1798	0,1942	0,1870	0,1881	0,1885
6	0,1366	0,1151	0,1151	0,1438	0,1438	0,1438	0,1151	0,1194	0,1183
7	0,1942	0,2014	0,1798	0,2014	0,2014	0,1942	0,1942	0,1942	0,1942
8	0,1870	0,1510	0,1582	0,1438	0,1366	0,1582	0,1582	0,1544	0,1530
9	0,2302	0,2086	0,2086	0,2014	0,2014	0,2302	0,2302	0,2302	0,2302
10	0,1223	0,1438	0,1366	0,1294	0,1294	0,1223	0,1366	0,1471	0,1432

Tabela A.35 – Erros de Classificação da Base *Spambase*.

	Regressão Linear								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	Seg_DT_1	Seg_DT_2	*NNTree_1*	*NNTree_2*
1	0,1391	0,0978	0,0978	0,1543	0,1521	0,0782	0,0782	0,0707	0,0684
2	0,2608	0,1021	0,1	0,2391	0,1369	0,1152	0,1195	0,0802	0,083
3	0,1043	0,0934	0,1	0,163	0,0891	0,0695	0,076	0,0581	0,0569
4	0,3891	0,1	0,1173	0,2652	0,1282	0,0847	0,0826	0,055	0,0631
5	0,1521	0,1	0,1	0,1717	0,1717	0,1	0,0978	0,0762	0,0515
6	0,4043	0,1217	0,1282	0,376	0,1673	0,0869	0,0891	0,065	0,065
7	0,1456	0,1021	0,1021	0,1891	0,0782	0,0782	0,1	0,0759	0,07
8	0,2	0,1	0,1	0,1891	0,0869	0,0804	0,0782	0,0493	0,0493
9	0,3086	0,1108	0,1152	0,1891	0,1391	0,1	0,1021	0,0729	0,0715
10	0,3318	0,1323	0,1735	0,2147	0,1193	0,1084	0,1127	0,1002	0,066

	Regressão Logística								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	Seg_DT_1	Seg_DT_2	*NNTree_1*	*NNTree_2*
1	0,0891	0,0717	0,0673	0,0804	0,0826	0,0891	0,0891	0,0638	0,066
2	0,1	0,0934	0,0869	0,0913	0,0913	0,1	0,1	0,0878	0,09
3	0,076	0,0652	0,0652	0,0652	0,063	0,076	0,076	0,0526	0,0528
4	0,076	0,0673	0,0804	0,0739	0,076	0,0782	0,0804	0,0574	0,0608
5	0,0869	0,0826	0,0826	0,0847	0,0804	0,0869	0,0869	0,0635	0,0648
6	0,0934	0,1043	0,1043	0,0956	0,1021	0,0934	0,0934	0,0727	0,0711
7	0,076	0,0782	0,0869	0,0739	0,0717	0,076	0,076	0,0661	0,0694
8	0,076	0,0826	0,0826	0,0782	0,0782	0,076	0,076	0,0682	0,0682
9	0,0934	0,0934	0,0934	0,0934	0,0956	0,0934	0,0934	0,0819	0,0834
10	0,1019	0,0802	0,0932	0,1041	0,1019	0,0976	0,0976	0,0621	0,0667

	Redes Neurais, 3 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	Seg_DT_1	Seg_DT_2	*NNTree_1*	*NNTree_2*
1	0,0673	0,0891	0,0826	0,05	0,0586	0,0652	0,0586	0,0558	0,0555
2	0,076	0,076	0,076	0,0652	0,0717	0,0891	0,0891	0,076	0,076
3	0,0478	0,0478	0,0478	0,0478	0,05	0,0478	0,0478	0,0512	0,0538
4	0,063	0,063	0,063	0,0543	0,0543	0,063	0,063	0,063	0,063
5	0,0739	0,0717	0,0717	0,0586	0,0695	0,063	0,063	0,0663	0,0609
6	0,0826	0,1	0,0847	0,0521	0,0717	0,0782	0,0782	0,0781	0,0731
7	0,0695	0,076	0,076	0,0565	0,0695	0,0695	0,0695	0,0695	0,0695
8	0,0717	0,0826	0,0826	0,0543	0,0652	0,0717	0,0717	0,0717	0,0717
9	0,0782	0,076	0,076	0,0717	0,0782	0,0978	0,0978	0,0805	0,0815
10	0,0759	0,0845	0,0845	0,0845	0,0759	0,0759	0,0759	0,0759	0,0759

	Redes Neurais, 3 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	Seg_DT_1	Seg_DT_2	*NNTree_1*	*NNTree_2*
1	0,0717	0,0543	0,0543	0,0434	0,0673	0,0543	0,0608	0,0725	0,0606
2	0,0826	0,0826	0,0826	0,0673	0,076	0,0826	0,0826	0,0826	0,0826
3	0,0717	0,0782	0,0782	0,0369	0,0586	0,0739	0,0739	0,0511	0,0508
4	0,0782	0,0608	0,0608	0,0521	0,0521	0,0782	0,0782	0,0782	0,0782
5	0,0717	0,0804	0,0608	0,0543	0,0695	0,0652	0,0652	0,0639	0,0644
6	0,0695	0,0978	0,0978	0,0586	0,0804	0,0695	0,0695	0,0813	0,0732
7	0,0717	0,0891	0,0891	0,0565	0,0913	0,0717	0,0717	0,0717	0,0717
8	0,063	0,063	0,063	0,0456	0,0782	0,063	0,063	0,063	0,063
9	0,0869	0,0869	0,0869	0,0652	0,0891	0,0869	0,0869	0,074	0,0834
10	0,0737	0,0759	0,0759	0,0715	0,0694	0,0737	0,0737	0,0737	0,0737

	Redes Neurais, 10 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	Seg_DT_1	Seg_DT_2	*NNTree_1*	*NNTree_2*
1	0,0739	0,0586	0,0652	0,0413	0,0652	0,0739	0,0739	0,0651	0,0632
2	0,113	0,0934	0,0934	0,0652	0,076	0,0826	0,0826	0,0917	0,0917
3	0,0608	0,0608	0,0521	0,0478	0,0565	0,0608	0,0608	0,0526	0,0518
4	0,0608	0,0543	0,0586	0,0456	0,076	0,0739	0,0739	0,0655	0,0656
5	0,0717	0,0652	0,0869	0,0543	0,0673	0,0717	0,0717	0,0649	0,0599
6	0,0978	0,0891	0,0869	0,0717	0,0891	0,0891	0,0891	0,0781	0,0731
7	0,0782	0,0913	0,0913	0,0478	0,063	0,0717	0,0652	0,0769	0,0772
8	0,0739	0,0847	0,0847	0,0521	0,0543	0,063	0,063	0,0654	0,0654
9	0,0782	0,0739	0,0739	0,0608	0,0673	0,0934	0,0934	0,0766	0,0759
10	0,0911	0,0824	0,0867	0,0607	0,0802	0,0932	0,0932	0,0797	0,0745

	Redes Neurais, 10 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	Seg_DT_1	Seg_DT_2	*NNTree_1*	*NNTree_2*
1	0,0717	0,0646	0,0677	0,0375	0,0687	0,0717	0,0717	0,0541	0,0658
2	0,1186	0,0959	0,0959	0,0597	0,0783	0,1015	0,1015	0,0855	0,0855
3	0,068	0,078	0,049	0,0477	0,0489	0,068	0,068	0,053	0,049
4	0,0555	0,0655	0,063	0,0412	0,081	0,0666	0,0666	0,0612	0,0733
5	0,0592	0,0581	0,0825	0,05	0,0669	0,0592	0,0592	0,0627	0,0593
6	0,0992	0,0944	0,0865	0,0599	0,0809	0,0865	0,0865	0,0803	0,077
7	0,0844	0,0941	0,0941	0,055	0,0637	0,0525	0,0603	0,0848	0,0787
8	0,0757	0,0919	0,0919	0,0398	0,052	0,0702	0,0702	0,067	0,067
9	0,0794	0,0751	0,0751	0,0637	0,0627	0,0818	0,0818	0,083	0,0771
10	0,0907	0,0851	0,0963	0,0681	0,0868	0,0915	0,0915	0,0754	0,0673

	Redes Neurais, 20 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	Seg_DT_1	Seg_DT_2	*NNTree_1*	*NNTree_2*
1	0,0717	0,0695	0,0695	0,0456	0,0652	0,0717	0,0717	0,0678	0,0689
2	0,0869	0,076	0,0869	0,0565	0,0695	0,0978	0,0847	0,0904	0,0919
3	0,0521	0,0673	0,0717	0,0391	0,0543	0,0413	0,0456	0,0485	0,0489
4	0,0956	0,0826	0,0652	0,0456	0,0717	0,0804	0,0608	0,0788	0,0752
5	0,0847	0,0891	0,0847	0,0652	0,0652	0,0739	0,0739	0,0581	0,055
6	0,1108	0,0913	0,0913	0,0739	0,0804	0,0934	0,0934	0,0836	0,0771
7	0,0739	0,0869	0,0695	0,0456	0,0652	0,0739	0,0739	0,0769	0,0772
8	0,0826	0,0826	0,0826	0,0652	0,0913	0,0826	0,0826	0,0668	0,0668
9	0,0695	0,0695	0,0695	0,063	0,076	0,0695	0,0695	0,0695	0,0695
10	0,0694	0,0715	0,0715	0,0563	0,0694	0,0694	0,0694	0,0716	0,0686

	Redes Neurais, 20 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	Seg_DT_1	Seg_DT_2	*NNTree_1*	*NNTree_2*
1	0,0771	0,0728	0,0762	0,0495	0,0659	0,0771	0,0771	0,0689	0,0802
2	0,0995	0,0757	0,0835	0,0571	0,0738	0,0935	0,0789	0,0901	0,0892
3	0,0431	0,0711	0,0714	0,0379	0,0561	0,0396	0,047	0,0441	0,0477
4	0,0926	0,0732	0,0731	0,0385	0,0739	0,0803	0,0618	0,0806	0,0801
5	0,0878	0,0832	0,0813	0,0633	0,0677	0,07	0,07	0,057	0,0536
6	0,1096	0,0898	0,0898	0,0828	0,0775	0,0994	0,0994	0,0797	0,0756
7	0,0701	0,0851	0,073	0,0481	0,0783	0,0701	0,0701	0,0769	0,0791
8	0,0813	0,0813	0,0813	0,0616	0,0841	0,0813	0,0813	0,064	0,064
9	0,0549	0,0549	0,0549	0,0622	0,0764	0,0549	0,0549	0,0549	0,0549
10	0,0738	0,0645	0,0645	0,0629	0,0807	0,0738	0,0738	0,0769	0,0629

Tabela A.36 – Erros de Classificação da Base *Wine*.

	Regressão Linear								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2681	0,2588	0,2573	0,2681	0,2604	0,2681	0,2681	0,3513	0,3399
2	0,2738	0,2307	0,2261	0,2661	0,2753	0,2276	0,2353	0,3321	0,3110
3	0,2507	0,2353	0,2169	0,2476	0,2492	0,2584	0,2415	0,3419	0,3057
4	0,2989	0,2896	0,2788	0,2942	0,2973	0,2989	0,2989	0,3453	0,3380
5	0,3046	0,2523	0,2492	0,2876	0,2907	0,2661	0,2615	0,3986	0,3558
6	0,2538	0,2492	0,2507	0,2538	0,2507	0,2538	0,2538	0,3151	0,3246
7	0,2773	0,2419	0,2388	0,2711	0,2727	0,2434	0,2434	0,3435	0,3035
8	0,28	0,2769	0,2738	0,2769	0,2784	0,2830	0,2553	0,3773	0,3710
9	0,3138	0,2723	0,2661	0,2892	0,2876	0,2815	0,2538	0,3714	0,3154
10	0,2461	0,2323	0,2292	0,2369	0,2461	0,2461	0,2461	0,3298	0,3149

	Regressão Logística								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2573	0,2650	0,2557	0,2619	0,2665	0,2619	0,2604	0,2843	0,2943
2	0,2153	0,24	0,24	0,2184	0,2184	0,2153	0,2153	0,2448	0,2651
3	0,2461	0,2323	0,2384	0,2461	0,2461	0,2492	0,2492	0,2686	0,2754
4	0,2989	0,2896	0,2835	0,2989	0,2989	0,2989	0,2989	0,2975	0,2976
5	0,2584	0,2646	0,2615	0,2569	0,2553	0,2584	0,2584	0,2794	0,2910
6	0,2630	0,2553	0,2523	0,26	0,2584	0,2630	0,2630	0,2701	0,2755
7	0,2496	0,2542	0,2542	0,2434	0,2449	0,2496	0,2496	0,2576	0,2588
8	0,2830	0,2753	0,2615	0,2815	0,2815	0,28	0,2630	0,2986	0,3042
9	0,2830	0,26	0,2569	0,2769	0,28	0,2830	0,2830	0,2853	0,2982
10	0,2430	0,2276	0,2415	0,2430	0,2415	0,2430	0,2307	0,2570	0,2599

	Redes Neurais, 3 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2403	0,2326	0,2403	0,2388	0,2403	0,2403	0,2403	0,2679	0,2830
2	0,2338	0,2323	0,2430	0,2230	0,2215	0,2338	0,2338	0,2494	0,2494
3	0,2430	0,2338	0,2323	0,2446	0,2338	0,2430	0,2430	0,2607	0,2607
4	0,2711	0,2650	0,2480	0,2588	0,2650	0,2696	0,2604	0,2823	0,2882
5	0,2492	0,2446	0,2523	0,2507	0,2415	0,2492	0,2492	0,2646	0,2764
6	0,2538	0,2584	0,2446	0,2584	0,2830	0,2538	0,2538	0,2642	0,2716
7	0,2342	0,2218	0,2192	0,2295	0,2295	0,2342	0,2342	0,2376	0,2435
8	0,2753	0,2615	0,2192	0,2476	0,2707	0,2584	0,2584	0,2928	0,3033
9	0,2723	0,2430	0,2192	0,2584	0,2584	0,2569	0,2569	0,2735	0,2810
10	0,2415	0,2338	0,2153	0,2369	0,2430	0,2446	0,2230	0,2617	0,2694

	Redes Neurais, 3 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2403	0,2326	0,2403	0,2434	0,2403	0,2403	0,2403	0,2679	0,2830
2	0,2338	0,2323	0,2430	0,2230	0,2215	0,2338	0,2338	0,2494	0,2494
3	0,2430	0,2338	0,2323	0,2446	0,2338	0,2430	0,2430	0,2607	0,2607
4	0,2711	0,2650	0,2480	0,2526	0,2650	0,2696	0,2604	0,2823	0,2882
5	0,2492	0,2446	0,2523	0,2523	0,2415	0,2492	0,2492	0,2646	0,2764
6	0,2615	0,2507	0,2507	0,2615	0,2461	0,2692	0,2692	0,2665	0,2726
7	0,2157	0,2141	0,2003	0,2187	0,2187	0,2157	0,2157	0,2157	0,2157
8	0,2738	0,2738	0,2707	0,2538	0,2646	0,2738	0,2738	0,2738	0,2738
9	0,24	0,2584	0,2430	0,2476	0,2584	0,24	0,24	0,2570	0,2648
10	0,2323	0,2261	0,2323	0,2369	0,24	0,2384	0,2384	0,2617	0,2722

	Redes Neurais, 10 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2665	0,2573	0,2155	0,2526	0,2650	0,2480	0,2449	0,2890	0,3048
2	0,2215	0,2276	0,2155	0,2246	0,22	0,2215	0,2215	0,2321	0,2412
3	0,2430	0,2461	0,2155	0,2323	0,2369	0,2430	0,2430	0,2494	0,2584
4	0,2681	0,2619	0,2155	0,2681	0,2681	0,2711	0,2650	0,2917	0,2995
5	0,2707	0,2415	0,2155	0,2523	0,2646	0,2630	0,2630	0,2707	0,2707
6	0,2538	0,2584	0,2446	0,2723	0,2830	0,2538	0,2538	0,2642	0,2716
7	0,2342	0,2218	0,2155	0,2265	0,2295	0,2342	0,2342	0,2376	0,2435
8	0,2753	0,2615	0,2155	0,2476	0,2707	0,2584	0,2584	0,2928	0,3033
9	0,2723	0,2430	0,2155	0,2584	0,2584	0,2569	0,2569	0,2735	0,2810
10	0,2415	0,2338	0,2153	0,2430	0,2430	0,2446	0,2230	0,2617	0,2694

	Redes Neurais, 10 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2665	0,2573	0,2192	0,2496	0,2650	0,2480	0,2449	0,2890	0,3048
2	0,2215	0,2276	0,2192	0,2169	0,22	0,2215	0,2215	0,2321	0,2412
3	0,2430	0,2461	0,2192	0,2230	0,2369	0,2430	0,2430	0,2494	0,2584
4	0,2681	0,2619	0,2192	0,2727	0,2681	0,2711	0,2650	0,2917	0,2995
5	0,2707	0,2415	0,2192	0,2492	0,2646	0,2630	0,2630	0,2707	0,2707
6	0,2538	0,2584	0,2446	0,2584	0,2830	0,2538	0,2538	0,2642	0,2716
7	0,2342	0,2218	0,2192	0,2295	0,2295	0,2342	0,2342	0,2376	0,2435
8	0,2753	0,2615	0,2192	0,2476	0,2707	0,2584	0,2584	0,2928	0,3033
9	0,2723	0,2430	0,2192	0,2584	0,2584	0,2569	0,2569	0,2735	0,2810
10	0,2415	0,2338	0,2153	0,2369	0,2430	0,2446	0,2230	0,2617	0,2694

	Redes Neurais, 20 Neurônios, BPROP								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2496	0,2357	0,2419	0,2619	0,2742	0,2496	0,2496	0,2761	0,2953
2	0,2369	0,2323	0,2261	0,22	0,2307	0,2369	0,2369	0,2505	0,2568
3	0,2384	0,2261	0,2261	0,2261	0,2430	0,2523	0,2492	0,2505	0,2575
4	0,2788	0,2681	0,2711	0,2773	0,2773	0,2711	0,2650	0,2882	0,2948
5	0,26	0,2738	0,2815	0,2538	0,2553	0,2507	0,2615	0,2806	0,2919
6	0,26	0,2676	0,2569	0,2569	0,2584	0,2723	0,2507	0,2642	0,2687
7	0,2372	0,2295	0,2311	0,2311	0,2434	0,2357	0,2372	0,2423	0,2445
8	0,2830	0,2707	0,2615	0,2615	0,2769	0,2830	0,2830	0,2830	0,2830
9	0,2769	0,2538	0,2492	0,2615	0,2661	0,2738	0,2523	0,2759	0,2801
10	0,2461	0,2415	0,2369	0,2492	0,2446	0,24	0,24	0,2617	0,2685

	Redes Neurais, 20 Neurônios, LEVMAR								
	Simples	*RISKSEG1*	*RISKSEG2*	*Bagging*	*Boosting*	*SegTree_1*	*SegTree_2*	*NNTree_1*	*NNTree_2*
1	0,2496	0,2357	0,2419	0,2542	0,2742	0,2496	0,2496	0,2761	0,2953
2	0,2369	0,2323	0,2261	0,2230	0,2307	0,2369	0,2369	0,2505	0,2568
3	0,2384	0,2261	0,2261	0,2307	0,2430	0,2523	0,2492	0,2505	0,2575
4	0,2788	0,2681	0,2711	0,2912	0,2773	0,2711	0,2650	0,2882	0,2948
5	0,26	0,2738	0,2815	0,2492	0,2553	0,2507	0,2615	0,26	0,26
6	0,26	0,2676	0,2569	0,2538	0,2584	0,2723	0,2507	0,2642	0,2687
7	0,2372	0,2295	0,2311	0,2295	0,2434	0,2357	0,2372	0,2423	0,2445
8	0,2830	0,2707	0,2615	0,2615	0,2769	0,2830	0,2830	0,2830	0,2830
9	0,2769	0,2538	0,2492	0,2630	0,2661	0,2738	0,2523	0,2759	0,2801
10	0,2461	0,2415	0,2369	0,24	0,2446	0,24	0,24	0,2617	0,2685

Tabela A.37 – Erros Médios de Classificação da Base de Concessão de Crédito.

Experimento	Erros Médios de Classificação							
	Otimização por Erro de Classificação				Otimização por *ROC*MIN			
	Simples	*RISKSEG*1	*RISKSEG*2	*RISKSEG*3	Simples	*RISKSEG*1	*RISKSEG*2	*RISKSEG*3
1	0,2994	0,2841	0,2769	0,2789	0,2929	0,2920	0,2873	0,2859
2	0,2945	0,2829	0,2745	0,2834	0,2971	0,2921	0,2881	0,2898
3	0,3045	0,2830	0,2746	0,2807	0,2947	0,2919	0,2875	0,2895
4	0,2978	0,2831	0,2780	0,2828	0,2946	0,2905	0,2900	0,2887
5	0,2983	0,2836	0,2773	0,2749	0,2953	0,2901	0,2864	0,2915
6	0,2828	0,2816	0,2802	0,2789	0,2976	0,2941	0,2942	0,2862
7	0,2998	0,2810	0,2739	0,2817	0,2969	0,2922	0,2895	0,2882
8	0,2932	0,2832	0,2746	0,2791	0,2945	0,2915	0,2881	0,2911
9	0,2911	0,2843	0,2769	0,2846	0,2954	0,2924	0,2881	0,2868
10	0,2948	0,2845	0,2733	0,2931	0,3011	0,2958	0,2884	0,2887

Tabela A.38 – Valores Médios de *ROC*MIN da Base de Concessão de Crédito.

Experimento	Valores Médios de *ROC*MIN							
	Otimização por Erro de Classificação				Otimização por *ROC*MIN			
	Simples	*RISKSEG*1	*RISKSEG*2	*RISKSEG*3	Simples	*RISKSEG*1	*RISKSEG*2	*RISKSEG*3
1	0,4210	0,4138	0,4085	0,4064	0,4201	0,4092	0,4025	0,4003
2	0,4155	0,4096	0,4059	0,4037	0,4160	0,4089	0,4015	0,3989
3	0,4192	0,4137	0,4093	0,4007	0,4210	0,4059	0,4005	0,3993
4	0,4160	0,4197	0,4057	0,4053	0,4088	0,4039	0,3990	0,3982
5	0,4154	0,4169	0,4043	0,4037	0,4226	0,4110	0,4021	0,4020
6	0,4199	0,4071	0,4067	0,4049	0,4282	0,4109	0,4042	0,3984
7	0,4190	0,4144	0,4044	0,4103	0,4157	0,4101	0,3964	0,3961
8	0,4142	0,4060	0,4033	0,4039	0,4156	0,4059	0,4010	0,4031
9	0,4247	0,4192	0,4025	0,4106	0,4174	0,4068	0,4038	0,4000
10	0,4199	0,4102	0,4023	0,4064	0,4150	0,4081	0,4038	0,4019

Tabela A.39 – Erros Médios de Classificação da Base de Fraude.

Experimento	Erros Médios de Classificação							
	Otimização por Erro de Classificação				Otimização por *ROC*MIN			
	Simples	*RISKSEG*1	*RISKSEG*2	*RISKSEG*3	Simples	*RISKSEG*1	*RISKSEG*2	*RISKSEG*3
1	0,1006	0,1036	0,0837	0,0857	0,1006	0,1006	0,0926	0,0936
2	0,1134	0,1015	0,0955	0,0945	0,1134	0,1015	0,0955	0,0955
3	0,1095	0,0846	0,0806	0,0846	0,1095	0,1015	0,0756	0,0756
4	0,1303	0,1045	0,1005	0,1005	0,1303	0,1045	0,0975	0,0935
5	0,1383	0,1154	0,0965	0,0985	0,1383	0,1274	0,1164	0,0955
6	0,1255	0,1285	0,1205	0,1205	0,1255	0,1255	0,1066	0,1086
7	0,1005	0,0925	0,0896	0,0846	0,1005	0,0925	0,0896	0,0896
8	0,1134	0,0925	0,0836	0,0826	0,1134	0,1025	0,0995	0,0846
9	0,1234	0,1224	0,1015	0,1015	0,1234	0,1065	0,0975	0,0935
10	0,1055	0,0746	0,0726	0,0726	0,1055	0,0746	0,0746	0,0746

Tabela A.40 – Valores Médios de *ROC*MIN da Base de Fraude.

Experimento	Valores Médios de *ROC*MIN							
	Otimização por Erro de Classificação				Otimização por *ROC*MIN			
	Simples	*RISKSEG*1	*RISKSEG*2	*RISKSEG*3	Simples	*RISKSEG*1	*RISKSEG*2	*RISKSEG*3
1	0,1916	0,1771	0,1486	0,1492	0,1916	0,1722	0,1629	0,1707
2	0,1969	0,1704	0,1577	0,1609	0,1969	0,1704	0,1577	0,1595
3	0,1838	0,1483	0,1362	0,1377	0,1838	0,1559	0,1438	0,1438
4	0,2048	0,186	0,1765	0,1765	0,2048	0,186	0,1667	0,1639
5	0,2097	0,1908	0,1763	0,1801	0,2097	0,1957	0,1801	0,1663
6	0,2435	0,2513	0,2281	0,2281	0,2435	0,2265	0,1952	0,2023
7	0,1845	0,1403	0,1559	0,149	0,1845	0,1403	0,1203	0,1203
8	0,1943	0,1755	0,1442	0,1432	0,1943	0,186	0,1658	0,1461
9	0,2293	0,2228	0,205	0,205	0,2293	0,2219	0,1857	0,1889
10	0,1442	0,1279	0,1196	0,1196	0,1442	0,1279	0,1089	0,1089

REFERÊNCIAS BIBLIOGRÁFICAS

[ABO00] F.M. Azevedo, L.M. Brasil and R. C. L. Oliveira, "*Redes Neurais com Aplicações em Controle e em Sistemas Especialistas*", Visual Books, 2000.

[AVA04] P. J. L. Adeodato, G.C. Vasconcelos, A. L. Arnaud, R. A. F. Santos, R. C. L. V. Cunha, and D. S. M. P. Monteiro, "*Neural Networks vs Logistic Regression: a Comparative Study on a Large Data Set*", ICPR'04, Volume 3, pp. 355-358, 2004.

[All97] M. P. Allen, "*Understanding Regression Analysis*", Plenum Press, 1997.

[Alt68] E. I. Altman, "*Financial Ratios, Discriminant Analysis and Prediction of Corporate Bankruptcy*", In of Journal of Finance, Volume 23, pp. 586-609, 1968.

[AVL99] M. Aurélio, M. Vellasco, and C. H. Lopes, "*Descoberta de Conhecimento e Mineração de Dados*". Em Apostila, Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, ICA – Lab. Inteligência Computacional Aplicada, 1999.

[BaK98] E. Bauer, and R. Kohvi, "*An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants*", Machine Learning, Volume V, pp. 1-38, 1998.

[BCL07] A. P. Braga, A. P. L. F. Carvalho, and T. B. Ludermir, "*Redes Neurais Artificiais: Teoria e aplicações*", 2ª. Edição, LTC Editora, Rio de Janeiro, 2007.

[Bra98] L. Braiman, "*Arcing classifiers"*, In The Annals of Statistics, Volume 26, Number 3, pp. 801-824, 1998.

[Big96] J. P. Bigus, "*Data Mining with Neural Network – Solving Business Problems from Application Development to Decision Support*", McGraw-Hill, 1996.

[BlM10] C. L. Blake, and C. J. Merz, "*UCI repository of machine learning databases"*, http://www.ics.uci.edu/~mlearn/MLRepository.htm, Department of Information and Computer Science, University of California, Irvine, CA, 2010 (data do último acesso 01/02/2010).

[BoD86] G. E. P. Box, and N. R. Draper, "*Empirical Model Building and Response Surface*", John Wiley & Sons, USA, 1986.

[Bre96] L. Breiman, "*Bagging predictors*", In Machine Learning, Volume 24, Number 2, pp. 123-140, 1996.

[BWH05] G. Brown, J. Wyatt, R. Harris, and X. Yao, "*Diversity Creation Methods: A Survey and Categorisation*", In Journal of Information Fusion, Volume 6, Number 1, pp. 5-20, 2005.

[CaB02] Casella, George and R. L. Berger, *"Statistical inference"*, 2nd ed., Duxbury Press, Pacific Grove, CA, 2002.

[CCA09] P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis, "*Modeling wine preferences by data mining from physicochemical properties*", In *Decision Support Systems*, Elsevier, Volume 47, Number 4, pp. 547-553, 2009.

[CGM02] H. Chipman, E. I. George, and R. E. McCulloch, "*Bayesian Treed Models*", In *Machine Learning*, Volume 48, Numbers 1-3, pp. 299-320, 2002.

[ChH03] M.-C. Chen, and S.-H. Huang, "*Credit scoring and rejected instances reassigning through evolutionary computation techniques*", In Expert Systems with Applications, Volume 24, Number 4, pp. 433-441, 2003.

[Con99] W. J. Conover, "*Practical Nonparametric Statistics*", 3rd ed., In John Wiley & Sons, 1999.

[DCO96] V. S. Desai, J. N. Crook, and G. A. Overstreet, "*A Comparison of Neural Networks and Linear Scoring Models in the Credit Union Environment"*, In European Journal of Operational Research, Volume 95, Number 1, pp. 24-37, 1996.

[Dem06] J. Demsar, "*Statistical Comparisons of Classifiers over Multiple Data Sets*", In The Journal of Machine Learning Research, Volume 7, pp. 1-30, 2006.

[DHS01] R. O. Duda, P. E. Hart, and D. G. Stork, "*Pattern classification*", 2nd ed., John Wiley & Sons, USA, 2001.

[Die98] T. G. Dietterich, "*Approximate statistical tests for comparing supervised classification learning algorithms*", In Journal of Neural Computation, Volume 10, pp. 1895-1923, 1998.

[DrH81] N. Draper, and H. Smith, "*Applied Regression Analysis"*, 2nd ed., John Wiley & Sons, USA, 1981.

[Dru99] P. F. Drucker, "*Management Challenges for the 21st Century*", HarperBusiness, 1999.

[DzZ04] S. Dzeroski, and B. Zenko, "*Is Combining Classifiers with Stacking Better than Selecting the Best One?*" In The Journal of Machine Learning, Volume 54, Number 3, pp. 255–273, 2004.

[Elb03] M. Y. El-Bakyr, "*Feed forward neural networks modeling for K–P interactions. Chaos"*, Solutons & Fractals, Volume 18, Number 5, pp. 995–1000, 2003.

[Faw06] T. Fawcett, "*An introduction to ROC analysis*", In Journal of Pattern Recognition Letters, Volume 27, Number 8, pp. 861-874, 2006.

[Fer06] T. A. E. Ferreira, "*Uma Nova Metodologia Híbrida Inteligente para a Previsão de Séries Temporais"*, Tese de Doutorado, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2006.

[Fay96] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, "*Advances in Knowledge Discovery and Data Mining*", American Association for Artificial Intelligence Press, California, USA, 1996.

[Fre95] Y. Freund, "*Boosting a weak learning algorithm by majority*", In Journal of Information and Computation, Volume 121, Number 2, pp. 256-285, Academic Press, 1995.

[FrS96] Y. Freund, and R. E. Schapire, "*Experiments with a new Boosting algorithm*", In Proceedings of the Thirteenth International Conference on Machine Learning, pp. 148-156, 1996.

[GBD06] T. V. Gestel, B. Baesens, P. V. Dijcke, J. A. K. Suykens, J. Garcia, and T. Alderweireld, "*Linear and Non-Linear Credit Scoring by Combining Logistic Regression and Support Vector Machines*", In Journal of Credit Risk, Volume 1, Number. 4, 2006.

[GEE09] M. Green, U. Ekelund, L. Edenbrandt, J. Björk, J. L. Forberg, and M. Ohlsson, *"Exploring new possibilities for case-based explanation of artificial neural network ensembles"*, Neural Networks, Volume 22, Number 1, pp. 75-81, 2009.

[HaB03] M. Hajmeer, and I. Basheer, "*Comparison of logistic regression and neural network-based classifiers for bacterial growth"*, In Food Microbiology (Academic Press), Volume 20, Number 1, pp.43-55, 2003.

[HaK01] J.Han, and M. Kamber, "*Data Mining: Concepts and Techniques*", Morgan Kaufmann Publishers, USA, 2001.

[HaM94] M. T Hagan, and M. B. Menhaj, *"Training feed forward networks with the Marquardt algorithm"*. IEEE Trans. Neural Networks 6, pp. 861-867, 1994.

[Hay98] S. Haykin, "*Neural Networks: a comprehensive foundation*", 2nd ed., Prentice Hall, New Jersey, USA, 1998.

[Hay09] S. Haykin, "*Neural Networks and Learning Machines*", 3rd ed., Prentice Hall, NJ, USA 936pp, 2009.

[HoL89] D. W. Hosmer, and S. Lemeshow, "*Applied Logistic Regression*", John Wiley & Sons Inc, USA, 1989.

[HSW89] K. Hornik, M. Stinchcombe, and H. White, "*Multilayer feedforward networks are universal approximators*", In Journal of Neural Network, Volume 2, Number 5, pp. 359-366, 1989.

[IYN08] M. M. Islam, X. Yao, S. M. S. Nirjon, M. A. Islam, and K. Murase, "*Bagging and Boosting negatively correlated neural networks***"**, IEEE Transactions on System, Man, and Cybernetics, Part B, Volume 38, Number 3, pp.771-84, 2008.

[JoW98] R. A. Johnson, and D. W. Wichern, "*Applied Multivariate Statistical Analysis*", Prentice Hall, Upper Saddle River, New Jersey, 4th edition, 1998.

[JSB05] B. Jackson, J. D. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and T. T. Tsai, "*An algorithm for optimal partitioning of data on an interval*", IEEE Signal Processing Letter, Volume 12, Number 2, pp. 105-108, 2005.

[Kia03] M. Y. Kiang, "*A comparative assessment of classification methods*", Decision Support Systems, Volume 35, Number 4, pp. 441-454, 2003.

[KHR03] M. Karahan, D. Hakkani-Tur, G. Riccardi, and G. Tur,. "*Combining classifiers for spoken language understanding*", In Proceedings of IEEE Workshop Automatic Speech Recognition and Understanding - ASRU '03, pp. 589-594, 2003.

[LAW08] A.Lahsasna, R. N. Ainon, and T. Y. Wah, "*Intelligent Credit Scoring Model using Soft Computing Approach"*, International Conference on Computer and Communication Engineering, pp. 396-402, 2008.

[LCL02] T.-S. Lee, C.-C. Chiu, C.-J. Lu, and I.-F Chen, "*Credit scoring using the hybrid neural discriminant technique*", Expert Systems with Applications, Volume 23, Number 3, pp. 245-254, 2002.

[LoF01] J. S. Long, and J. Freese, *"Regression Models for Categorical Dependent Variables Using Stata"*. Stata Press, College Station, Texas, 2001.

[MaM03] R. Malhotra, and D. K. Malhotra, "*Evaluating consumer loans using neural networks*", In Internacional Journal of Management Science - Omega, Volume 31, Number 2, pp. 83-96, 2003.

[May04] E. Mays, "*Credit Scoring for Risk Managers: The Handbook for Lenders*", 1st ed., South-Western Educational Pub, USA, 2004.

[MaR05] O. Maimon, and L. Rokach, "*Decomposition Methodology for Knowledge Discovery and Data Mining: Theory and Applications*", Series in Machine Perception and Artificial Intelligence, Volume 61, World Scientific Publishing, 2005.

[Mit97] T. Mitchell, "*Machine Learning"*, McGraw-Hill Series in Computer Science, 1997.

[MLH03] D. Meyer, F. Leisch, and K. Hornik, "*The support vector machine under test"*, In Neurocomputing, Volume 55, Number 1-2, pp. 169-186, 2003.

[MTW03] G. Marsaglia, W. W Tsang, and J. Wang, "*Evaluating Kolmogorov's distribution*", In Journal of Statistical Software, Volume 8, Number 18, pp. 1-4, 2003.

[NaP03] S. Nargundkar, and J. L. Priestley, "*Assessment of Evaluation Methods for Binary Classification Modeling*", In Proceedings of the 2003 Decision Sciences Institute National Conference, pp. 1-6, Nov. 2003.

[Nev98] P. G. Neville, "*Growing Trees for Stratified Modeling"*, Interface 98, In Proceedings of Computing Science and Statistics, Volume 30, pp. 528-533, 1998.

[Nev99] P. G. Neville, "*Decision Trees for Predictive Modeling*, SAS Technical Report, SAS Institute, 1999.

[Net96] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, "*Applied Linear Statistical Models*", McGraw Hill, 4th Edition, 1996.

[OHT05] C. S. Ong, J. J. Huang, G. H. Tzeng, *"Building credit scoring models using genetic programming"*, Expert Systems with Applications, Volume 29, Number 1, pp 41-47, 2005.

[Pra08] Pradipta Maji, *"Efficient Design of Neural Network Tree Using A New Splitting Criterion"*, Neurocomputing, Volume 71, Número 4-6, pp. 787-800, 2008.

[Pre94] L. Prechelt, "*PROBEN1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules*", Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe, Germany, 1994.

[Pyl99] D. Pyle, "*Data Preparation for Data Mining"*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[QiR07] B. Qian, and K. Rasheed, "*Stock market prediction with multiple classifiers*", In Applied Intelligence, Volume 26, Number 1, pp. 25-33, 2007.

[Qui86] J. R. Quinlan, "*Induction of Decision Trees*", In Journal of Machine Learning, Volume 1, Number 1, pp. 81-106, USA, 1986.

[Qui93] J. R. Quinlan, "*C4.5: Programs for Machine Learning*", Morgan Kaufmann Publishers Inc., USA, 1993.

[Qui96] J. R. Quinlan, "*Bagging, Boosting and C4.5*", In Proceedings of the Thirteenth National Conference on Artificial Intelligence, pp. 725-730, AAAI Press, USA, 1996.

[RoG01] F. Roli, and G. Giancinto, "*An Approach to the Automatic Design of Multiple Classifier Systems*", In Proceedings of Pattern Recognition Letters**,** Volume 22, Number 11, pp. 25-33, 2001.

[Rok05] L. Rokach, "*Ensemble Methods for Classifiers"*, Data Mining and Knowledge Discovery Handbook, Part VI, pp. 957-980, Springer US, 2005.

[Rud01] O. P. Rud, "*Data Mining Cookbook: Modeling Data for Marketing, Risk and Customer Relationship Management*", John Wiley & Sons, New York, 2001.

[RuN95] J. R. Stuart, and P. Norvig, "*Artificial Intelligence: A Modern Approach*", Prentice Hall, 1995.

[RuW98] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "*Learning internal representations by error propagation"*, Parallel distributed processing: explorations in the microstructure of cognition, Volume 1: foundations, pp. 318-362, MIT Press, 1998.

[San02] R. A. F. Santos, "*Metodologia e uso de técnicas de exploração e análise de dados na construção de data warehouse*", Dissertação de mestrado, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2002.

[San07] R. A. F. Santos, "*Uso de Redes Neurais artificiais na detecção de fraudes*", Revista Tecnologia de Crédito, Volume 63, Number 11, pp. 65-72, São Paulo, 2007.

[SAS02] SAS Institute Inc., "*Finding the solution to data mining: A map of the features and components of SAS Enterprise Miner software*", http://www.sas.com/, SAS Institute Inc., North Carolina, USA, 2002 (data do último acesso 30/03/2009).

[SaW07] E. W. Saad, and D. C. Wunsch II, *"Neural network explanation using inversion"*, Neural Networks, Volume 20, Number 1, pp. 78-93, 2007.

[Sch90] R. E. Schapire, "*The Strength of Weak Learnability*", In Journal of Machine Learning, Volume 5, Number 2, pp. 197-227, 1990.

[Sil02] J. P. Silva, "*Gestão e análise do risco de crédito*". Terceira edição, Atlas Editora, São Paulo, 2002.

[Sha87] A.D. Shapiro, *"Structured Induction in Expert Systems"*, Addison-Wesley, USA, 1987.

[She97] D. J. Sheskin, "*Handbook of parametric and nonparametric statistical procedures*", Chapman and Hall/CRC, New York, 1997.

[ThC02] L. C. Thomas, D. Edelman, and J. Crook, "*Credit Scoring and its Applications*", Society for Industrial and Applied Mathematics, Philadelphia, USA, 2002.

[TiW99] K. M. Ting, and I. H. Witten, *"Issues in stacked generalization"*, In Journal of Artificial Intelligence Research, Volume 10, Number 1, pp. 271–289, 1999.

[VAE03] R. Vilalta, M.-K. Achari, and C. F. Eick, "*Class Decomposition Via Clustering:A New Framework For Low-Variance Classifiers*", Third IEEE International Conference on Data Mining (ICDM 2003), pp. 673–676, 2003.

[VCB05] R. Vilalta, C. Carrier, and P. Brazdil, Meta-Learning: *"Concepts and Techniques, The Data Mining and Knowledge Discovery Handbook"*, Springer US, pp. 731-748, 2005.

[WiF05] I. H. Witten, and E. Frank, "*Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations***"**, Morgan Kaufmann Publishers Inc., San Francisco, 2005.

[Wol92] D. H. Wolpert, *"Stacked Generalization"*, In Neural Networks, Volume 5, Number 2, pp. 241-259, 1992.

[ZTD01] B. Zenko, L. Todorovski, and S. Dzeroski, *"A comparison of stacking with MDTs to Bagging, Boosting, and other stacking methods"*, Proceedings of ECML/PKDD01 Workshop: Integrating Aspects of Data Mining, pp. 163-175, 2001.