Milton Vasconcelos da Gama Neto

O processo CRISP-DM aplicado na construção de uma solução para Análise de Risco de Crédito

Recife

2018

Milton Vasconcelos da Gama Neto

O processo CRISP-DM aplicado na construção de uma solução para Análise de Risco de Crédito

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal de Pernambuco - UFPE

Centro de Informática

Graduação em Ciência da Computação

Orientador: Professor Germano Crispim Vasconcelos

Recife

2018

Agradecimentos

Primeiro, gostaria a agradecer a minha mãe, Ywanoska, e meu pai, Pergentino. Por tudo que eles fazem por mim, por todo incentivo aos estudos, pelo apoio durante a graduação e todo carinho. Também sou grato a eles pela referência que eles são para mim.

Outra pessoa que devo muito a agradecer é minha namorada, Marília. Muito obrigado por me acompanhar nessa trajetória, pela paciência e apoio durante esse trabalho. Seu otimismo e atenção me ajudaram muito a concluir essa caminhada, com certeza você a deixou bem melhor.

Além dos meus pais, outra pessoa que me inspiro desde pequeno é meu tio Kiev. A este, devo muito a agradecer, pois sem essa referência talvez eu não ingressasse no curso de Ciência da Computação. Me espelhar nele desde pequeno me trouxe aqui. Sou muito grato, pois acredito que escolhi a profissão certa, gosto muito do que faço e me sinto realizado, mesmo com muitas metas para serem alcançadas ainda.

Meus amigos do ensino médio, que compartilham comigo vários momentos de descontração e também nas dificuldades, sempre me apoiaram. Muito obrigado também.

Sobre meus amigos, queria agradecer em especial aos que conheci na universidade. Esses que estavam comigo no dia a dia, que fizemos vários projetos juntos, foram estudos, desabafos, e toda caminhada juntos na graduação. Obrigado por compartilhar todos esses momentos.

Agradeço ao professor Germano Vasconcelos pela orientação desse trabalho.

E por fim, agradeço aos meus professores do CIn. Todos que passaram durante minha trajetória na graduação contribuíram bastante.

Resumo

As empresas coletam uma quantidade imensa de dados dos seus usuários, e procuram utilizar essa massa de dados com objetivo de aumentar os lucros, criar insights, melhorar os serviços, entre outros fins. Devido a grande quantidade de dados, se torna inviável a análise e reconhecimento de padrões ao olho nu. A Ciência dos Dados é responsável pela sistematização da análise a partir de técnicas estatísticas, computacionais e de aprendizagem de máquina. Este trabalho tem o objetivo de detalhar as etapas do processo CRISP-DM, o mais utilizado na área, e aplicá-lo em um problema de classificação para risco de crédito. O problema consiste em determinar se um cliente deve ter a aprovação do crédito ou não, buscando uma taxa alta de aprovação, mas com baixa ocorrência de inadimplência. Para isso, é utilizado o histórico de outros clientes para aprender o comportamento e criar um modelo preditivo com esses dados. Os resultados mostraram a importância da preparação dos dados, incluindo técnicas para limpeza que aumentem a qualidade dos dados para melhorar o mapeamento da entrada para saída desejada. Foi realizado o preenchimento dos dados faltantes através de aprendizagem de máquina. A implementação desta técnica foi dividida em duas etapas, a primeira para avaliação do modelo construído para predizer o valor do dado faltantes, e a segunda para preencher os dados faltantes do atributos se o modelo obter bons resultados na primeira fase. O desempenho também foi melhorado com a introdução de técnicas não-supervisionadas para identificar perfil do cliente e construir classificadores específicos para esses tipos. Foi obtido 31,7 no teste KS, considerado um resultado muito bom para esse problema.

Palavras-chaves: CRISP-DM; Ciência dos Dados; Análise de risco de crédito;

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AED	Análise Exploratória de Dados
CRISP-DM	*Cross Industry Standard Process for Data Mining*
KDD	*Knowledge Discovery in Database*
IQR	*Interquartile Range*
KS	*Kolmogorov-Smirnov*
MI	*Mutual Information*
RFE	*Recursive Feature Elimination*
SMOTE	*Synthetic Minority Oversampling Technique*
MSE	*Mean Square Error*
AM	Aprendizagem de Máquina
AD	Árvore de Decisão
RL	Regressão Logística
RO	*Random Oversampling*
RU	*Random Undersampling*

Sumário

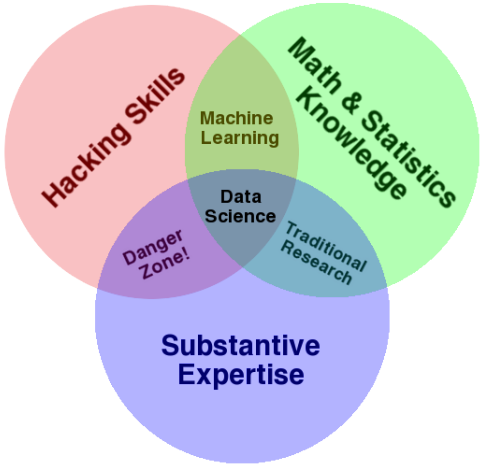
1 Introdução

Vivendo a era dos dados, onde milhões de informações são geradas por segundo através de smartphones, redes sociais, internet e por sensores, as empresas, startups e pesquisadores, buscam utilizar esses dados de forma estratégica. Com principais objetivos de aumentar os lucros, melhorar os processos existentes e gerar insights para novas soluções (HAN; PEI; KAMBER, 2011).

Os dados são considerados como ouro do século XXI, mas não basta tê-los, é necessário saber o que fazer com eles, como realmente transformá-los em algo valioso. Nesse contexto que entra a Ciência dos Dados, um termo antigo que passou a ser amplamente utilizado com o surgimento de Big Data e o desenvolvimento de Machine Learning.

Ciência dos Dados é a sistematização da análise de dados, que devido ao grande volume desses, é inviável a realização dessa tarefa manualmente. Então, são utilizadas técnicas estatísticas, matemáticas e computacionais para obter informação e conhecimento a partir dos dados através de um conjunto de princípios fundamentais (PROVOST; FAWCETT, 2013). Segundo (HAYASHI, 1998), Ciência dos Dados é o "conceito para unificar estatística, análise de dados, aprendizagem de máquina e seus métodos relacionados". Por ser uma área interdisciplinar, o perfil de um cientista de dados envolve conhecimento e habilidades de diferentes áreas. Drew Conway mostra através de um Diagrama de Venn (Figura 1) a interdisciplinaridade e interseção que representa essa área. Onde é destacada a importância da especialidade no domínio (*Substantive Expertise*), para definir os alvos corretos, levantar questionamentos e hipóteses sobre o problema específico.

Figura 1 – Diagrama de Venn de Drew Conway sobre Ciência dos Dados

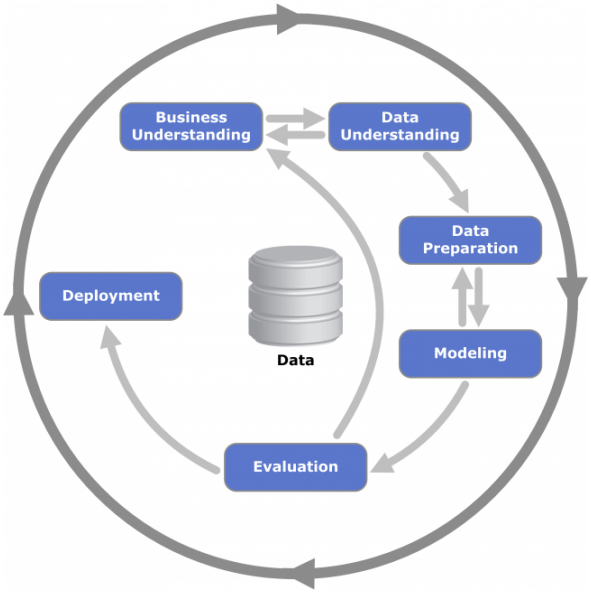


Os padrões identificados nos dados, interpretações ou avaliações são o resultado final

de um longo processo, que é dividido em cinco fases na Descoberta de Conhecimento em base de dados (KDD, do inglês *Knowledge Discovery in Database*) (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Esse processo sistemático define que o conhecimento é obtido após a realização da seleção dos dados, pré-processamento, transformação, mineração de dados e avaliação. O entendimento dessa jornada é importante para compreender que a partir da definição do problema existe um longo processo, que consome bastante tempo, até chegar em um resultado final.

Das metodologias utilizadas em projetos de Ciência dos Dados, CRISP-DM (*Cross Industry Standard Process for Data Mining*) (CHAPMAN et al., 2000) é a mais utilizada (PIATETSKY, 2014). Esse processo, ilustrado na Figura 2, consiste de 6 etapas (entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação e implantação) e pode ser considerado uma implementação do KDD (AZEVEDO; SANTOS, 2008). Por ser o mais utilizado e incluir fases como entendimento do negócio e implantação, é um processo mais completo comparado a outros, como SEMMA (*Sample, Explore, Modify, Model, Assess*) (SAS, 2014). Por isso, essa metodologia foi adotada para o trabalho e suas fases serão detalhadas e aplicadas.

Figura 2 – Processo CRISP-DM



A aplicação da metodologia será feita na análise de risco de crédito, problema que consiste em determinar se um determinado cliente deve ter a concessão do crédito, considerando o potencial retorno e o risco de inadimplência. Essa análise serve para definiros clientes bons e ruins, que devem ter o crédito aprovado e não aprovado, respectivamente. Com aproximadamente 520 mil registros de diferentes indivíduos, 191 atributos e uma variável alvo que determina a aprovação do crédito, será explorado as fases do processo CRISP-DM, aplicando um conjunto de técnincas nesta base de dados real de larga escala

para entender os problemas que os cientistas de dados se deparam no dia a dia. Outro objetivo é apresentar a relevância da preparação dos dados, mostrar como essa etapa pode melhorar os resultados, e explorar técnicas de modelagem para solucionar o problema de classificar a concessão do crédito com técnicas de aprendizagem de máquina.

2 Fundamentação Teórica

Este capítulo apresenta conceitos importantes para o entendimento deste trabalho. Primeiramente, será apresentada a metodologia CRISP-DM, que foi utilizada na construção da solução. Em seguida, são introduzidas algumas ténicas de pré-processamento, o conceito de aprendizagem de máquina, em especial o aprendizado supervisionado, e, por fim, a métrica de avaliação utilizada.

2.1 CRISP-DM

Cross Industry Standard Process for Data Mining (CRISP-DM) é uma metodologia para solucionar problemas em Ciência dos Dados, fornecendo um ciclo de vida definido pelos estágios de modo flexível. Foi desenvolvido pela Daimler Chrysler, SPSS e NCR em 2000 (CHAPMAN et al., 2000). A ideia da flexibilidade é mostrar que o processo não ocorre de maneira sequencial, mas sim de forma cíclica, onde o retorno a um estágio pode ocorrer, a fim de chegar em um resultado melhor e sempre focada no entendimento do negócio.

Os 6 estágios do CRISP-DM são:

1. **Compreensão do Negócio:** Fase inicial do processo, focada nos problemas, objetivos e especificações com uma perspectiva de negócio. A partir desse conhecimento é definido um problema de Ciência dos Dados para ser solucionado. É de suma importância, pois o sucesso da solução depende da importância e elaboração do problema. Apesar de ser no início, essa visão deve acompanhar o processo, e é um dos principais motivos dele ser cíclico, pois as fases não podem perder o foco no objetivo final.

2. **Compreensão dos Dados:** Os dados são a matéria-prima para a solução. Nessa fase é analisada a qualidade dos dados de acordo com o problema, explorar os dados, gerar os primeiros insights e formular as hipóteses.

3. **Preparação dos Dados:** Consiste em gerar um conjunto de dados final a partir dos dados não tratados. O que inclui etapas como a seleção dos dados (instâncias e atributos), integração, limpeza e transformação.

4. **Modelagem:** É nessa fase onde as técnicas de Mineração de Dados (fase correspondente no KDD) são aplicadas, com intuito de descobrir padrões nos dados. Diversas técnicas são utilizadas e a preparação dos dados tem forte influência no resultado obtido, por isso, muitas vezes é necessário voltar uma etapa.

5. **Avaliação:** O objetivo dessa fase é estimar os resultados do modelo construído anteriormente, e os principais focos são escolher o melhor modelo e validar se os resultados atendem as expectativas. A avaliação tenta ser o mais próximo possível da realidade, para isso é necessário cuidado na seleção dos dados que serão utilizados no teste e nas métricas utilizadas.

6. **Implantação:** Geralmente o modelo gerado não é o final do projeto. Mas sim, parte dele, quando são colocados em uso real. Essa etapa não é necessariamente realizada por um cientista de dados.

2.2 Pré-processamento

Uma das etapas em uma solução de ciência dos dados é o pré-processamento dos dados. Essa etapa, que precede a utilização dos modelos de aprendizagem de máquina, consome boa parte do tempo dos cientistas de dados. Dados com baixa qualidade vão gerar resultados finais de baixa qualidade (HAN; PEI; KAMBER, 2011).

2.2.1 Dados Faltantes

Com dados do mundo real, um cenário comum é a falta de valores para alguns atributos. Existem diversas técnicas para contornar isso, são elas:

- Ignorar o dado faltante. Podendo ser ignorado a instância completa ou o atributo que contém o valor falante.
- Preenchimento manual. Caso seja possível ter informação do valor e a quantidade seja pequena, uma das opções é o preenchimento manual. Porém, na maioria dos casos é inviável, pois o número não costuma ser pequeno e consome muito tempo.
- Usar uma constante global para preencher os dados faltantes. Frequentemente, são utilizados o valor “Desconhecido” para atributos categóricos e os valores 0 ou -∞ para atributos numéricos.
- Usar valor estatístico do atributo para preencher os dados faltantes. Para atributos categóricos, os dados faltantes são preenchidos com a moda deste atribtuo. Para atributos numéricos é possível utilizar a média ou mediana.
- Preenchimento com o valor mais provável. Essa técnica consiste em criar um modelo de aprendizagem de máquina com os outros atributos para realizar a predição dos dados faltantes.

2.2.2 Outliers

Segundo (HAWKINS, 1980), outlier é uma observação que desvia muito das outras, despertando suspeita do mecanismo de geração utilizado. Essa observação aparenta ser inconsistente com o restante do conjunto de dados (JOHNSON; WICHERN, 1992). Outliers podem ser de dois tipos: univariado e multivariado.

2.2.2.1 Univariado

Considera a distribuição de apenas um atributo para detectar os outliers. Duas metodologias populares da literatura, são:

- *Z-Score*: Também conhecida como *Standard Score*, é uma maneira de representar um dado de acordo com a relação entre a média e desvio padrão de um grupo, Equação 2.1. Esse método utilizado para detecção de outliers, mapeia um dado (observação) em uma distribuição. Quando o *z-score* ultrapassa um limiar estabelecido, normalmente o valor absoluto 3, o dado é considerado um outlier. Esse valor representa a quantidade de vezes o desvio padrão que a observação está distante da média.

$$z_i = \frac{x_i - \overline{x}}{S} \tag{2.1}$$

  Onde $\overline{x}$ é a média e S é o desvio padrão.

- Método de Tukey (Box-Plot): Desenvolvido por John Tukey, esse método utiliza as informações dos quartis para detectar os outliers. O método é baseado na amplitude inter-quartil (interquartile range, IQR), que é o valor da distância entre o quartil superior e inferior. As observações que estão 1,5 × IQR abaixo do primeiro quartil ou acima do terceiro quartil, são consideradas outliers, Equação 2.4. O Box-Plot, Figura 3, é uma forma de visualizar a distribuição dos dados e identificar os outliers, de acordo com a forma descrita acima.

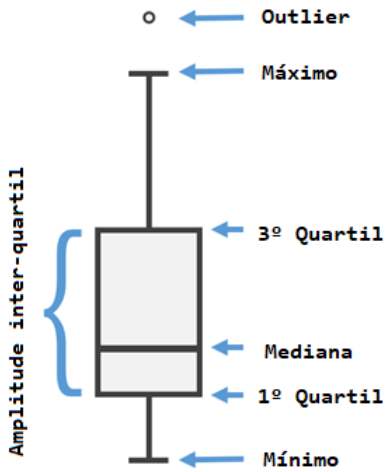
$$LimiteSuperior = \max(\min(dados), Q_1 - 1.5(Q_3 - Q_1)) \tag{2.2}$$

$$LimiteInferior = \min(\max(dados), Q_3 + 1.5(Q_3 - Q_1)) \tag{2.3}$$

$$f(x) = \begin{cases} 1, & \text{se } LimiteInferior > x > LimiteSuperior \\ 0, & \end{cases} \tag{2.4}$$

  A função $f(x)$, indica se uma observação x é um outlier ou não, com 1 e 0, respectivamente.

Figura 3 – Gráfico Box-Plot

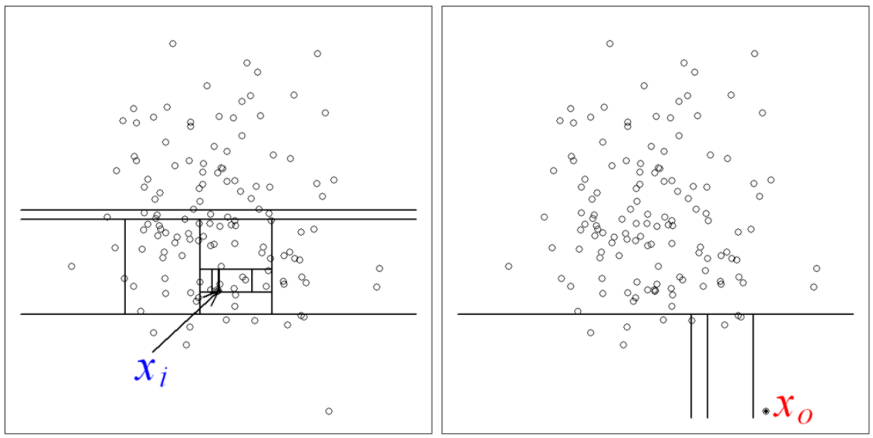


2.2.2.2 Multivariado

Considera um espaço n-dimensional, onde n é o número de atributos. Uma abordagem para detectar outliers de maneira multivariada é a seguinte:

- *Isolation Forest*: A partir da construção de um *ensemble* de Árvores Isoladas (*Isolation Trees*), os outliers são detectados quando apresentam um caminho curto entre a raiz da árvore e a folha (LIU; TING; ZHOU, 2008). Um caminho curto, representa uma separação simples, onde foi utilizada poucas partições para isolar determinada observação. As observações normais geralmente precisam de mais partições, como mostra a Figura 4, com dado normal em (a) e uma anomalia em (b).

Figura 4 – Isolamento dos pontos com Isolation Forest



(a) Isolating x_i (b) Isolating x_o

Em uma *Isolation Tree*, os atributos são selecionados aleatoriamente, assim como

o valor para a partição do atributo, respeitando o limite máximo e mínimo. Com isto, são gerados dois nós e o processo é repetido recursivamente. *Isolation Forest* constrói um conjunto destas árvores e considera o caminho médio para determinar se determinada observação é um outlier.

2.2.3 Transformação

Os dados são transformados no formato apropriado para mineração (HAN; PEI; KAMBER, 2011). Em alguns casos, a transformação é realizada na tentativa de generalização, mudança na representação dos dados, redução das possibilidades. Alguns exemplos de redução:

- Categorização: Transforma um atributo numérico em categórico. A partir de intervalos numéricos, que podem ser estabelecidos manualmente ou definido de maneira estatística (percentis, intervalos fixos calculados a partir do atributo).
- Normalização: Os atributos passam a ter a mesma escala, como de -1 a 1, ou 0 a 1. Essa modificação é muito importante para algoritmos que utilizam distância ou uma Rede Neural Artificial, por exemplo. Evitando que atributos originalmente em uma escala maior, tenha vantagem em relações a outros, de forma errônea. Uma normalização muito utilizada é a Normalização Min-Max, Equação 2.5, que realiza uma transformação linear nos dados, onde passam a ter a escala de 0 a 1, preservando as relações dos dados originais.

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}(min - max) + min \tag{2.5}$$

  Onde $x = (x_1, \ldots, x_n)$ e z_i é a i-ésima observação normalizada. Os valores min e max, correspondem ao mínimo e o máximo, respectivamente, do alcance estabelido para transformação.

- Binarização: Transformação dos atributos categóricos para atributos binários. A forma mais comum de codificar é através da codificação *One Hot*. Com esta, a quantidade de categorias presente em um atributo define o número de atributos que serão gerados, e cada instância só vai ter um desses novos atributos com o valor 1, os demais são 0 (POTDAR; PARDAWALA; PAI, 2017).

2.2.4 Seleção de Atributos

- *Filter*: Seleciona os atributos independente do modelo de AM que será utilizado. Essa metodologia consiste em aplicar métodos estatísticos, onde são calculados a correlação entre o atributo e o alvo. Para considerar um atributo relevante, existem diversas técnicas que também podem levar em conta outros fatores, como a variância

e a relação com outras características, por exemplo. Em geral, são feitas de forma univariada e consideram um atributo independente dos outros.

- *Wrapper*: Utiliza um modelo de AM para avaliar a performance dos atributos selecionados, e de maneira iterativa, modifica esse subconjunto até encontrar o melhor, através de heurísticas para guiar a busca do melhor subconjunto, evitando a busca exaustiva. Técnicas de algoritmos gulosos costumam ser utilizadas, onde pode ser feita a partir de Seleção Para Frente (forward selection) ou Eliminação Para Trás (backward elimination)

- *Embedded*: Essa abordagem utiliza modelos de AM que capturam a informação dos atributos que mais contribuiem para acurácia do modelo durante a fase de treinamento, ou seja, os que discriminam melhor os dados. A partir das informações da relevância de cada atributo, é realizado a seleção.

2.2.5 Desbalanceamento

O desbalanceamento ocorre quando existe uma diferença significante na quantidade de amostras de uma classe comparado as outras. Para problemas de classificação, isso pode gerar um viés para a classe majoritária, o que implica em um grau de acerto alto para essa classe e baixo para a classe minoritária. É necessário utilizar métricas que não camuflem essa realidade (HE; GARCIA, 2008). Como tentativa de melhorar a taxa de acerto da classe minoritária, várias técnicas foram propostas na literatura para reamostragem (*resampling*) dos dados, com intuito de igualar, ou aproximar, a distribuição das classes, podendo ser realizado o aumento da classe minoritária ou a diminuição da classe majoritária, essas técnicas são conhecidas como *oversampling* e *undersampling*, respectivamente. Também é possível combinar as duas abordagens.

2.3 Aprendizagem de Máquina

Aprendizagem de Máquina é um ramo da Inteligência Artificial que permite que computadores aprendam comportamentos, detectem padrões e tomem decisões com uma interação mínima de humanos. Tom Mitchell define formalmente Aprendizagem de Máquina (MITCHELL, 1997) como:

> "Um programa aprende a partir da experiência E, em relação a uma classe de tarefas T, com medida de desempenho P, se seu desempenho em T, medido por P, melhora com E."
>
> (Tom Mitchell, 1997)

Aprendizagem de Máquina é dividida em três tipos:

- Aprendizado supervisionado: O modelo recebe os dados rotulados, com entradas e saídas desejadas, e tenta aprender uma regra mapear as entradas para as saídas. O objetivo é que o modelo consiga determinar corretamente a saída para exemplos ainda não vistos. São conhecidos como classificação e como regressão, os problemas para os dados discretos e os contínuos, respectivamente.
- Aprendizado não-supervisionado: O modelo analisa os dados fornecidos e tenta agrupar de alguma forma, descobrir os padrões. Para esse tipo de abordagem, não é fornecido um rótulo para o modelo.
- Aprendizado por reforço: Uma interação com ambiente dinâmico, onde o modelo realiza um objetivo e recebe feedback quanto a saída fornecida e busca aprender o comportamento ideal.

2.3.1 Classificadores

Os modelos de Aprendizagem de Máquina para problemas supervisionados de classificação, onde as saídas são discretas, são conhecidos como Classificadores.

2.3.1.1 Regressão Logística

Apesar do nome, a regressão logística é para problemas de classificação. Faz parte da família dos modelos lineares, onde temos a forma generalizada na Equação 2.6. A regressão logística difere dessa forma em dois aspectos principais: utiliza uma função logística e modela a probabilidade condicional da variável dependente em função das variáveis independentes, $P(Y = 1|X)$, que podemos expressar como $p(X)$. Usando a função logística temos o modelo na Equação 2.7. Esta forma ajusta melhor os dados comparado a regressão linear.

$$p(X) = \beta_0 + \beta_1 X \tag{2.6}$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}} \tag{2.7}$$

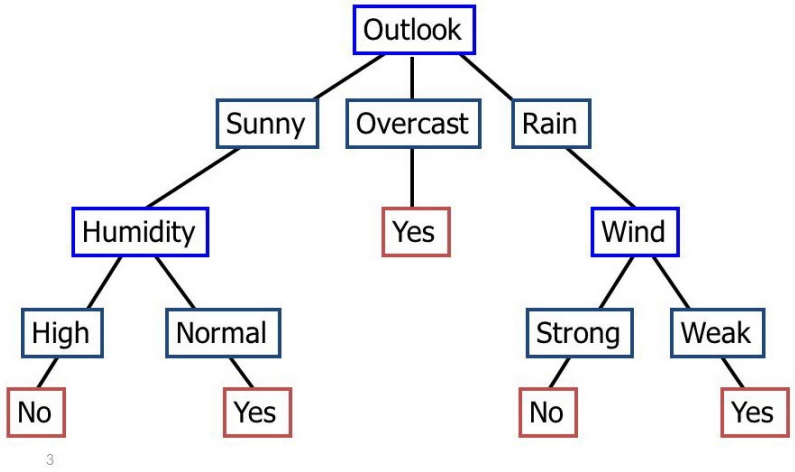
Onde β_1 são os coeficientes da entrada X.

2.3.1.2 Árvore de Decisão

Árvores de Decisão são uma das técnicas mais comuns em Aprendizagem de Máquina, por criar um modelo indutivo de fácil representação e entendimento. A indução é feita pela separação de regiões, onde um problema maior é dividido em vários problemas menores.

Cada nó interior verifica uma condição para o atributo, o caminho corresponde ao valor do atributo e as folhas são os alvos, como mostra o exemplo na Figura 5 . Os valores alvos são representados pelo caminho percorrido na árvore a partir da raiz. A árvore de decisão é construída da fase de treinamento, onde são selecionados os atributos e regras para compor os nós e definir as folhas (alvos).

Figura 5 – Árvore de Decisão



Essa construção é feita com a capacidade dos atributos discriminarem os alvos, ou seja, como é possível segmentar melhor os dados, de forma que diferencie os resultados esperados. Então, selecionando os que melhor realizam essa tarefa, de acordo com a métrica estabelecida, para serem os de mais alto nível na árvore.

Vale ressaltar que existe um cuidado com o crescimento da árvore, para evitar o overfitting, costuma-se aplicar o principio de Navalha de Occam. Uma prática comum para controlar o tamanho das árvores é a técnica de poda, onde é realizado um ajuste para diminuir o tamanho da árvore de acordo com o erro no conjunto de validação.

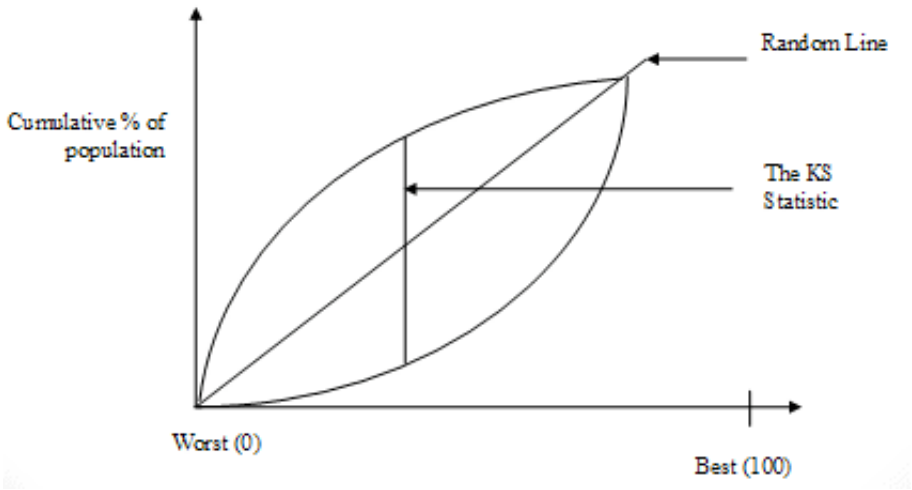
2.3.1.3 Gradient Boosting

Técnica de Aprendizagem de Máquina que constrói um modelo de comitê de classificadores fracos, tipicamente utilizando Árvores de Decisão. A técnica de Boosting consiste em ajustar os classificadores fracos de forma sequencial, dando pesos aos exemplos de acordo com o desempenho do classificador, onde os erros de classificação tem prioridade, com intuito que ele tenha maior chance de ser selecionado para próxima iteração. O resultado final, em uma técnica de boosting, é dado pelo voto majoritário ponderado.

2.4 Métrica de Avaliação

A principal métrica de avaliação neste é o Teste Kolmogorov-Smirnov, também conhecido como Teste KS. É um teste estatístico não paramétrico para comparar a igualdade entre as distribuições de probabilidades de dois grupos. O valor é dado pela máxima diferença absoluta entre as distribuições acumulada. Para avaliar um modelo de aprendizagem de máquina, considerando a classificação de um problema binário, o valor é calculado a partir dos acúmulos das probabilidades obtidas pelo modelo para as duas classes. O valor da estatística varia entre 0 e 1, quanto mais próximo de 1, representa um poder discriminatório maior do modelo.

Figura 6 – Teste Kolmogorov-Smirnov (KS)



A Figura 6 mostra a separação entre as distribuições das duas classes (curva superior e inferior).

3 Metodologia

Este capítulo, é destinado a explicar a metodologia adotada na construção de uma solução para análise de risco de crédito. O problema consiste em, a partir de um limiar, indicar se o cliente deve ter o crédito aprovado ou não, considerando o risco de inadimplência. A solução proposta, utiliza o processo CRISP-DM para resolver o problema a partir de dados históricos dos clientes, onde padrões escondidos são capturados com técnicas de aprendizagem de máquina e utilizados para prever o comportamento dos clientes novos.

Neste trabalho, serão exploradas técnicas de pré-processamento para melhorar a qualidade dos dados, assim como encontrar classificadores robustos para alcançar os melhores resultados.

Apesar da fase de avaliação ser após a modelagem, de acordo com o fluxo do CRISP-DM (seção 2.1), ela será realizada constantemente, pois serve para validar as ações que são tomadas ao longo do processo.

3.1 Compreensão dos Dados

A coleta dos dados não foi realizada durante esse trabalho, pois os dados foram fornecidos por terceiros. Porém, continua sendo necessário o entendimento dos dados utilizados, a natureza deles, escalas, etc. A base de dados contém 518.929 instâncias, 191 atributos e uma variável alvo. Os atributos são tanto contínuos como categóricos e a o alvo pode assumir duas classes, 0 ou 1, que indica não aprovação e aprovação, respectivamente.

Dos atributos, 161 são contínuos e 30 são nominais. Os dados são heterogênos, obtidos por diferentes fontes, contém informações financeiras, demográficas a respeito do endereço do cliente, e algumas informações públicas que caracterizam a pessoa.

Com uma análise prévia, já é possível identificar os dados faltantes, presença de outliers e necessidade de tratamento nos dados. A Análise Exploratória de Dados (AED) (TUKEY, 1977), será feita durante a preparação dos dados, e essas observações investigadas mais profundamente.

3.2 Preparação dos Dados

Nesta etapa, os dados são preparados para os algoritmos de aprendizagem de máquina. Apesar dos dados já estarem no formato tabular, onde cada instância representa um cliente e os atributos são informações sobre ele que o modelo utilizará para determinar se aprova o crédito ou não, é preciso tratar os dados faltantes, o que acontece frequentemente na

base de dados. Durante a análise inicial, já é perceptível a existência de outliers, quando os quartis apresentam uma grande diferença entre o primeiro quartil e o valor mínimo ou o terceiro quartil para o valor máximo. Também podem ser notados através da visualização da dispersão dos dados. Será realizado a AED para uma investigação mais profunda desse tópico. Além disso, também serão aplicadas transformações, seleção de atributos e técnicas de reamostragem para solucionar o problema do desbalanceamento, também detectado na seção 3.1.

Durante essa seção, passaremos pela modelagem e avaliação frequentemente, como forma de validar as técnicas que estão sendo aplicadas. A métrica utilizada será a estatística KS. Como o foco nessa etapa não é em encontrar o melhor modelo de aprendizagem de máquina, vamos utilizar o mesmo modelo para classificação, para que a comparação seja justa.

3.2.1 Dados Faltantes

É necessário representar o dado faltante de alguma maneira para depois executar algum modelo de AM. Outra opção é excluir a observação ou coluna que esta lacuna. Porém, é necessário bastante cuidado durante essa fase, excluir os dados podem reduzir significativamente o tamanho da base e o preenchimento errado pode enviesar os resultados.

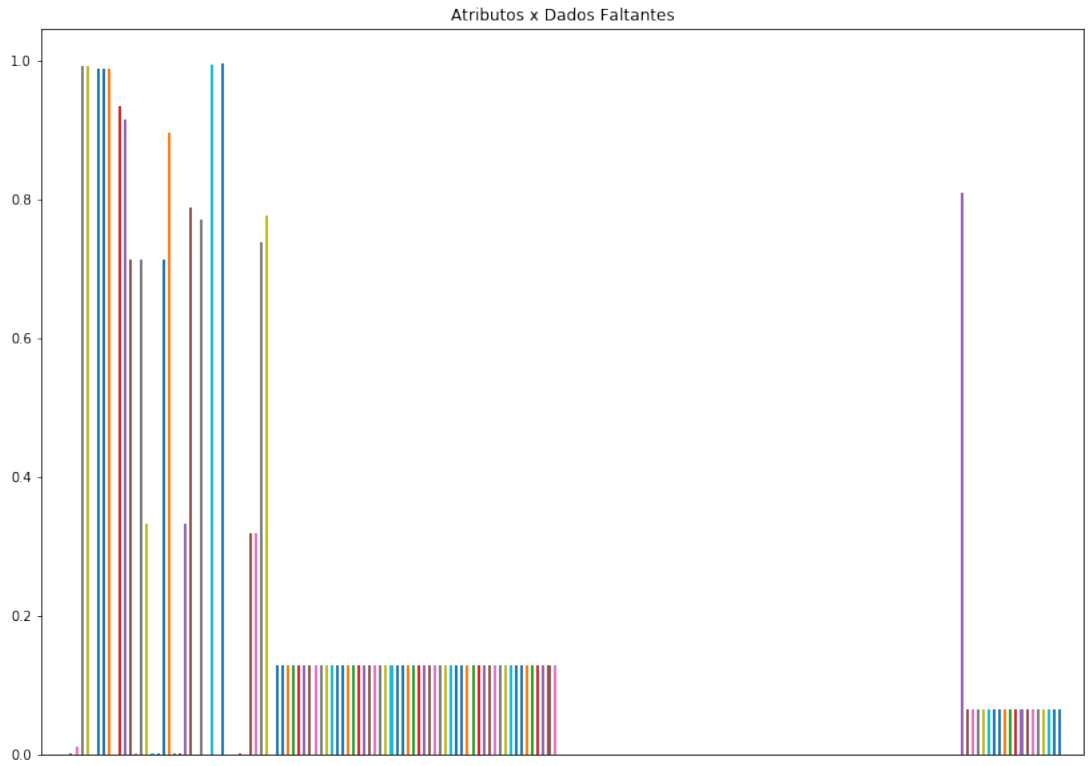
O gráfico da Figura 7, mostra que a quantidade de lacunas nos dados é bem alta, onde alguns atributos chegam a ter mais de 99% de observações sem sua característica preenchidas. Dos 191 atributos analisados, 91 contém alguma falta de informação. Com isso, algumas abordagens conhecidas na literatura, como remoção do atributo ou da instância, são descartadas para solucionar esse problema por completo. Até podem ser utilizadas, mas combinadas com outras técnicas. Apenas a exclusão, acarretará na perda quase completa da base de dados.

1. Ignorar o dado faltante. Instâncias com uma quantidade alta de atributos não preenchidos serão excluídos, dado um limiar da porcentagem de dados presentes.

2. Preenchimento com constante: É perceptível que a falta de informação em alguns atributos não significa um erro, e sim, que para determinada instância o atributo não faz muito sentido e foi permitido não preencher esses dados. Dois campos que servem de exemplo para esclarecer essa questão, são: “SIGLA_PARTIDIO_FILIADO” e “REMUNERACAO_SERVICO_MILITAR”. Esses campos não são obrigatórios para os clientes, e os valores faltantes podem ser substituído por 0 e a categoria “SEM_PARTIDO”, respectivamente.

   Porém, existem ocorrência de dados faltantes que não fazem sentido no mundo real. Como a falta da informação do sexo e a idade. A abordagem utilizada aqui, é fazer

Figura 7 – Porcentagem de dados faltantes por atributo



o preenchimento com a categoria "DESCONHECIDO" e o valor 0, para atributos categóricos e numéricos, respectivamente.

3. Preenchimento com mediana e média.

4. Predição do valor para o dado faltante. Para cada coluna com lacuna, foi construído um modelo de AM, um classificador ou um regressor, seja o atributo discreto ou contínuao, respectivamente.

   O treinamento foi realizado com a parte dos dados que estavam corretamente preenchidos. O processo foi dividio em duas etapas: (1) avaliação do modelo, onde a base de treinamento era dividia para testar a performance do estimador; e, (2) preenchimento com a predição dos valores dos dados faltantes, onde o modelo utilizava o conjunto de treinamento completo e realizava a predição dos dados faltantes. Com essa abordagem (2), não é possível validar se o preenchimento foi correto, pois não sabemos o valor esperado. O processo é dividido em duas etapas para obter a performance do modelo, permitindo testar diferentes configurações e encontrar os melhores parâmetros. Após validar se os resultados foram satisfatórios, é construído um modelo igual ao da fase anterior, mas que utiliza todos os dados rotulados (sem lacunas) para treinar. A Figura 8 e Figura 9, representam o fluxo de (1) e (2), respecitavamente. Apesar de ser mais sofisticada que as outras, tem um risco de inserir valores incorretos comparados a realidade.

Para classificação, foi utilizado *Random Forest* e para regressão, *Stochastic Gradient Descent*.

Figura 8 – Fluxo para avaliação do modelo construído para preenchimento dos dados

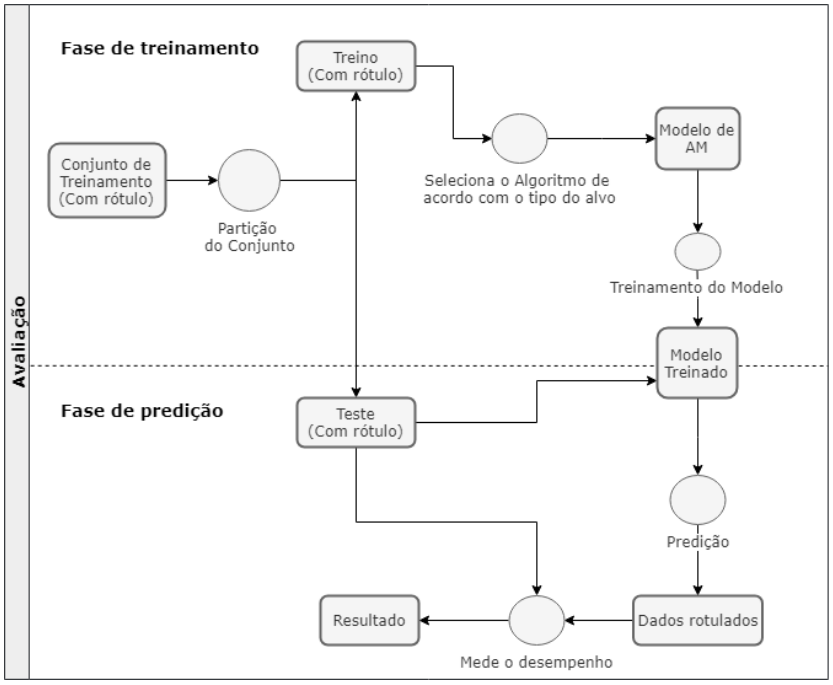
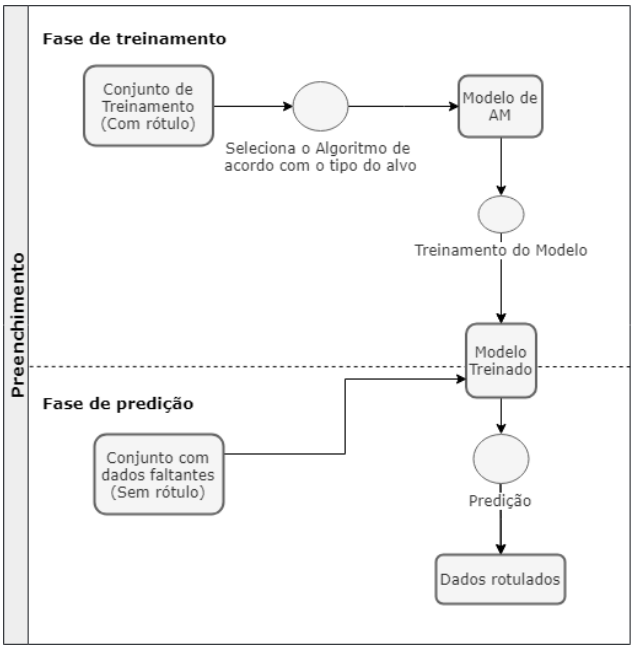


Figura 9 – Fluxo da predição para preenchimento dos dados

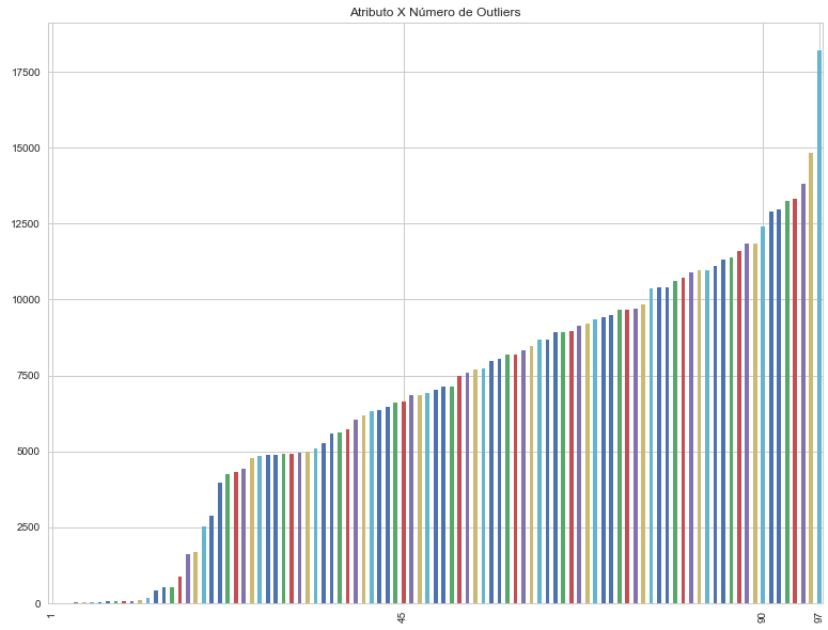


3.2.2 Detecção e tratamento de Outliers

3.2.2.1 Univariado

Analisando de maneira univariada, a base de dados apresentou uma quantidade muito alta de outliers, onde 97, dos 161 atributos contínuos, tem outliers. Esse resultado foi o mesmo para o método de Tukey e para o Z-Score de acordo com a Figura 10 e Figura 11, para cada abordagem respectivamente. A diferença foi na quantidade de outliers que cada atributo apresentou, no qual o método de Tukey apresentou uma totalidade maior de outliers.

Figura 10 – Outliers por atributo com Z-Score



O valor utilizado para caracterizar uma observação como outlier foi: $z > |3|$.

Uma das técnicas comumente utilizada na literatura para evitar que essas observações que diferem do comportamento padrão dos dados, é a remoção destes. Porém, se aplicada nesses dados, a base quase toda será perdida.

A maioria dos atributos não segue uma distribuição normal, na verdade, muitas vezes, os dados ficam concentrados no valor zero (ou outro valor baixo) e depois apresentam muitas ocorrências de valores intermediários ou muito altos. Um exemplo disso, ocorre em "EXPOSICAO_ENDERECO_FAVELA", Figura 12. As técnicas utilizadas para detectar outliers, assumem uma distribuição normal. Já o atributo "IDADE", apesar de conter outliers, tem um comportamento mais semelhante a distribuição mencionada, Figura 13, apresentando 321 outliers, enquanto "EXPOSICAO_ENDERECO_FAVELA" tem 54.714.

Figura 11 – Outliers por atributo com método de Tukey

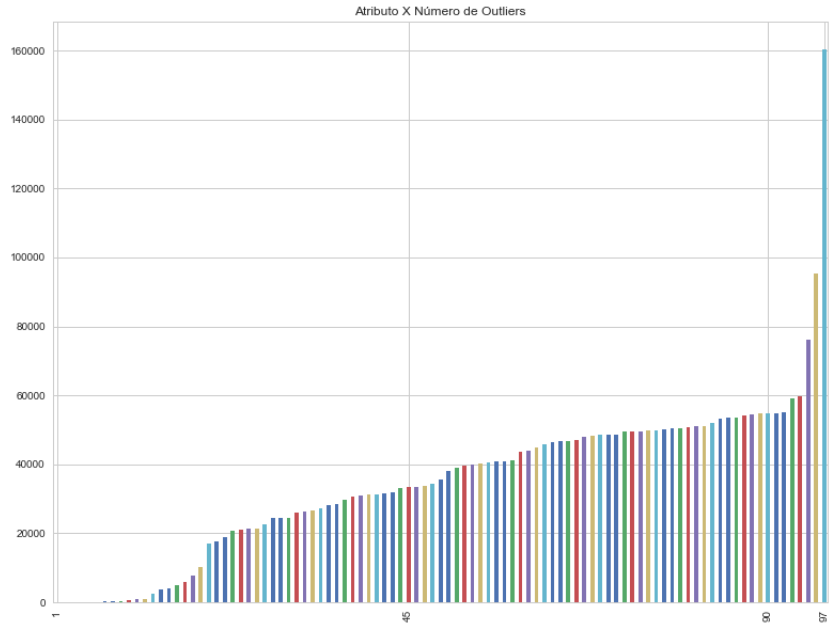
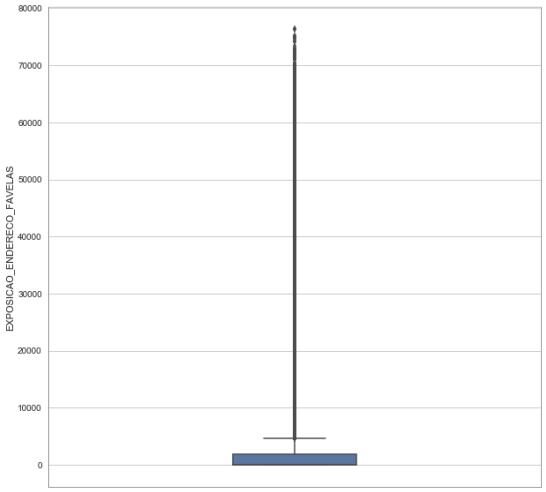


Figura 12 – Box-Plot do atributo EXPOSICAO_ENDERECO_FAVELA

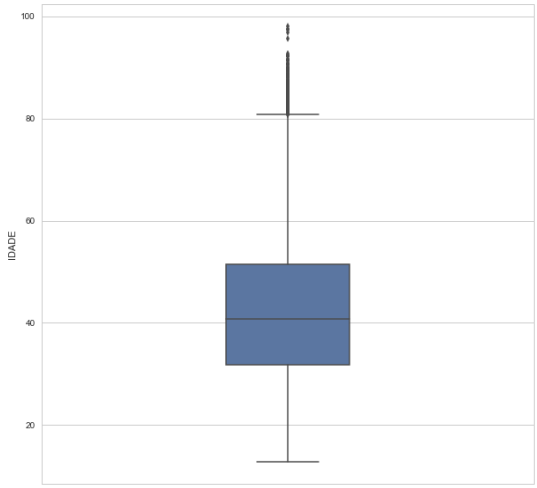


3.2.2.2 Multivariado

Será considerado as n dimensões para definir se um atributo é um outlier através do modelo não-supervisionado *Isolation Forest*. Essa abordagem não assume distribuição para os dados, e, a quantidade que será detectada é parametrizada através do conceito de "contaminação". De acordo com o próprio conceito de outliers (HAWKINS, 1980), poucas observações devem ser encontradas. A taxa utilizada deve ser pequena e esses exemplos detectados serão removidos do conjunto de treinamento.

Os resultados obtidos são apresentados no Capítulo 4, que explica os experimentos

Figura 13 – Box-Plot do atributo IDADE



realizados, variando a taxa de contaminação e o número de árvores para construção da floresta (*Isolation Forest*).

3.2.3 Transformação

Transformações são realizadas para os dados se adequarem ao formato dos modelos de AM e, muitas vezes, com intuito de aumentar a acurácia. Uma das técnicas que foi aplicada para todos os experimentos, foi a binarização dos dados categóricos com a codificação *One Hot*. A primeira motivação para isso, é que os modelos dos Experimentos (Capítulo 4), não suportam dados nominais, e, a transformação em dados discretos numéricos não seria suficiente, pois os algoritmos poderiam aprender erroneamente o compartamento quando considerarem a ordem dos valores.

A Normalização Min-Max também foi aplicada, especialmente, quando a escala dos atributos afeta o desempenho, o que não é o caso de algoritmos baseados em árvores.

3.2.4 Seleção de Atributos

Após realizar a binarização, a base passa a ter 295 atributos. Faz-se necessário uma análise da relevância destes. Alguns, podem não ter impacto para a variável alvo, influenciar de forma negativa ou apenas aumentar a complexidade do problema. Com a seleção dos atributos, temos a redução da dimensionalidade do problema, o que implica em treinamentos mais rápidos dos modelos de AM. Porém, reduzir não é o único foco. É preciso, pelo menos, manter a perfomance quando comparado aos dados completos. As técnicas aplicadas foram as seguintes:

3.2.4.1 Mutual Information

O conceito de Informação Mútua, do inglês *Mutual Information* (MI), é a quantificação da informação que uma variável aleatória tem acerca de outra. Este conceito está formente ligado a Entropia, e funciona para variáveis contínuas e discretas (MAREK et al., 2008). MI é definido como Equação 3.1.

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \tag{3.1}$$

Onde $P(x,y)$ é a distribuição de probabilidade conjunta de X e Y, e $P(x)$ e $P(y)$ são funções de probabilidade de distribuição marginal de X e Y, respectivamente. Para variáveis aleatórias contínuas, o somatório é substituído por um integral dupla Equação 3.2.

$$I(X;Y) = \int_Y \int_X P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \tag{3.2}$$

Para esse caso, $P(x,y)$ representa a função de densidade de probabilidade conjunta de X e Y, e $P(x)$ e $P(y)$ são funções de densidade de probabilidade marginal de X e Y, respectivamente.

Os atributos são ordenados de acordo com seu valor de MI, e apenas os k primeiros são selecionados, onde o valor k é a quantidade de atributos desejada para o modelo de AM. Essa é uma abordagem do tipo *Filter*.

3.2.4.2 Recursive Feature Elimination

A Eliminação de Atributos Recursiva, em inglês *Recursive Feature Elimination* (RFE), proposto por GUYON et al., é uma abordagem do tipo *Wrapper*. A seleção de um subconjunto de atributos é feita através de otimização gulosa. Inicialmente, é construído um modelo com todos os atributos e feito um ranking com a contribuição dos atributos para a saída, então, são removidos os que menos contribuiram (ZENG et al., 2009). O processo é repetido até atingir o número desejado de atributos ou de acordo com o desempenho.

3.2.4.3 Seleção de atributo baseado em Árvores

A seleção de atributos baseado em árvore, é uma abordadgem do tipo *Embedded*, onde é utilizada uma árvore de decisão ou *ensemble* de árvores de decisão para selecionar os atributos mais relevantes. Durante a fase de treinamento, onde é construída a árvore, é feito o cálculo da importância de cada atritubo para decidir qual será seu nó, de acordo com alguma medida de pureza. A subseção 2.3.1.2 explica em mais detalhes como é feita a construção. Com os valores calculados e uma heurística para encontrar um limiar a partir destes valores, são selecionados apenas os atributos que são maiores ou iguais a este limiar.

3.2.4.4 Seleção de atributo baseado em Modelos Lineares Regularizados

Semelhante as árvores de decisão, modelos lineares também utilizam um mecanismo para medir o impacto do atributo no alvo. Porém, a forma de calcular é diferente. Como foi visto na subseção 2.3.1.1, os modelos lineares, nesse caso a Regressão Logística, calculam os pesos dos coeficientes para os atributos na fase de treinamento. Um coeficentes alto representam uma grande importância daquele atributo para a variável dependente (alvo).

A regularização é uma penalização nos coeficientes que encoraja a soma dos valores absolutos a serem pequenas (NG, 2004). O objetivo é que valores muito altos podem induzir a um *overfitting*, e o que o modelo deve buscar é uma generalização. Modelos penalizados com L1 tem soluções esparças (TIBSHIRANI, 1996), onde atributos fracos são forçados a ter coeficiente zero, facilitando na seleção de atributos. Vale ressaltar que a seleção não precisa ser apenas dos valores acima de zero. O limiar pode ser parametrizado com uma heurística ou um valor, assim como é feito na seleção utilizando árvore de decisão. Também faz parte da família *Embedded*.

3.2.5 Desbalanceamento

O conjunto de dados para análise de crédito está desbalanceado, $65,5\%$ dos exemplos são da classe 1, que representa a aprovação de crédito. A separação dos conjuntos de treinamento e teste foram feitos de maneira estratificada, então a proporção é mantida, a Tabela 1 mostra a distribuição das classes.

Nesta seção, será apresentada técnicas para balanceamento dos dados.

Tabela 1 – Desbalanceamento entre as classes

Classe	**Conjunto de Treinamento**	**Conjunto de Teste**	**Proporção**
0	134098	44700	(34,5%)
1	255098	85033	(65,5%)

3.2.5.1 Random Oversampling e Random Undersampling

Ambas técnicas consistem em selecionar alguns exemplos aleatóriamente e utilizá-los para atingir o nível de balanceamento desejado. Com *Oversampling*, os dados são selecionados da classe minoritária e replicadas. Para *Undersampling*, a seleção é feita na classe majoritária, e os exemplos são removidos (HE; GARCIA, 2008).

3.2.5.2 SMOTE

O SMOTE (*Synthetic Minority Oversampling Technique*), é uma técnica de *Oversampling* que cria exemplos artificiais. Primeiro, escolhe aleatoriamente um exemplo x_i da classe

minoritária, depois seleciona os k vizinhos mais próximos de acordo com alguma métrica de distância. Dos k vizinhos, seleciona um aleatoriamente, z. O novo exemplo, x, é criado a partir da diferença entre os vetores z e x_i multiplicado por um número aleatório entre $[0, 1]$, δ, e depois somado ao ao ponto x_i, Equação 3.3.

$$x = x_i + (z - x_i) \times \delta \tag{3.3}$$

3.3 Modelagem

Com os dados preparados, essa etapa tem como objetivo encontrar a melhor técnica de classficação para análise de risco de crédito para os clientes a partir de suas características. Neste trabalho, o melhor modelo é definido pelo maior valor da estatística KS (seção 2.4) no conjunto de teste. Essa métrica de avaliação foi selecionada por representar bem a acertividade do modelo para ambas as classes, considerando as probabilidades dos estimadores, e por ser conhecida no contexto de análise de risco de crédito.

Uma das abordagens para modelar o problema, é identificar o perfil do cliente, através de métodos não-supervisionados de *clustering*, já que não existe uma característica nos dados explícita para isso, ou seja, não existe um atributo *TIPO_CLIENTE* ou *PERFIL_CLIENTE*. O objetivo é juntar clientes que apresentam características parecidas para construir classificadores especializados para cada grupo encontrado. Essa metodologia, consiste em, para cada *cluster* (grupo), que representa um tipo de cliente, construir um classificador, com o alvo original do problema (aprovação de crédito), utilizando as instâncias que pertencem ao *cluster* para treiná-lo. Após o treinamento, cada *cluster* tem um modelo associado já treinado e pronto para realizar a predição. Durante a fase de teste, será identificado qual *cluster* a instância pertence e utilizar o classificador associado a ele para predição. Neste estudo, todos os *clusters* tem o modelo de AM do mesmo tipo, mas são treinados com dados diferentes.

4 Experimentos e Análises

A metodologia proposta, que segue o processo CRISP-DM para investigar diferentes técninas no pré-processamento e escolha do modelo de aprendizagem, gera uma grande quantidade de combinações para serem testadas. Neste capítulo, vamos apresentar os experimentos realizados em cada fase, para encontrar as melhores abordagens, de acordo com a métrica KS, e depois avaliar os resultados obtidos.

4.1 Base de dados

Como foi mencionado anteriormente, a base de dadaos, que contém 518.929 exemplos, foi dividida em dois conjuntos, um para treinamento e outro para teste, com 75% e 25% dos dados, respectivamente. A base de dados originalmente possui 191 atributos, mas ao longo esse número pode variar, devido a binarização e seleção de atributos, por exemplo. O processo de binarização vai ocorrer sempre, então, podemos considerar que os conjuntos iniciais para os modelos de AM tem 295 atributos.

As abordagens aplicadas durante a etapa de preparação dos dados, foram realizadas especificamente no conjunto de treinamento, e, quando necessário, apenas aplicadas no conjunto de testes. Por exemplo, na seleção de atributos, independente do tipo, a escolha dos atributos é feita no conjunto de treinamento, depois, na fase de predição, é apenas filtrado do conjunto de teste os atributos desejados.

4.2 Tecnologias

Os experimentos realizados para construção da solução foram feitos na linguagem de programação Python, uma das mais utilizadas atualmente para Ciência dos Dados e Aprendizagem de Máquina, com várias bibliotecas implementadas por terceiros para isso.

Algumas dessas são utilizadas neste projeto, como Scikit-Learn[1], ou sk-learn, que é um *framework open-source* com implementações de algoritmos para Aprendizagem de Máquina e tratamento dos dados. Para manipulação dos dados, Pandas[2] e Numpy[3]. A primeira, oferece estruturas para armazenamento dos dados, lidos diretamente de um arquivo de texto, a principal característica é a implementação de funções para manipulação dos dados de maneira rápida e flexível. Esta biblioteca utiliza Numpy para criar suas estruturas de dados, esta última, é um pacote da linguagem Python para array multi-

[1] <https://scikit-learn.org/stable/>
[2] <https://pandas.pydata.org/>
[3] <https://numpy.org/>

dimensionais e matrizes, que suporta funções matemáticas de alta complexidade, algebra linear, estatística e outras funcionalidades. As técnicas de *sampling* foram feitas utilizando o pacote Imbalanced-Learn[4]. E para visualização de dados, Matplotlib[5], Seaborn[6] e Scikit-plot[7].

A maioria dos experimentos com os modelos do sklearn foram realizados com os parâmetros *default*, de acordo com a versão 19.1. Quando algum parâmetro for alterado, será especificado nesta seção.

4.3 Experimento 1

O primeiro experimento, foi realizado para testar o impacto de diferentes técnicas no preenchimento dos dados faltantes. Os testes foram feitos com a Árvore de Decisão.

Tabela 2 – Resultado do preenchimento dos dados faltantes

Atributos numéricos	**Atributos categóricos**	**KS**
0	"DESCONHECIDO"	14,4
Mediana	"DESCONHECIDO"	14,3
Média	"DESCONHECIDO"	14,2
Predição	"DESCONHECIDO"	**15,6**
0	Predição	14,4
Mediana 0	Predição	14,2

Preencher os dados faltantes é uma tarefa delicada e muito importante, pois essa etapa precisa ser realizada, diferente de algumas que são opcionais durante a preparação da base. Consideramos apenas o preenchimento, pois a exclusão das instâncias ou atributos causaria a perda de quase todos os dados.

As técnicas mais simples, como preenchimento com zero ou mediana para atributos numéricos e uma nova categoria, "desconhecido", adicionado para os atributos categóricos, demonstrou um bom desempenho. Essa abordagem é mais conservadora e tem uma tendência menor a inserir dados incorretos.

A predição dos dados faltantes foi feita para 15 atributos numéricos e 3 categóricos. Essa abordagem obteve bons resultados para árvore de decisão, porém esse resultado nem sempre foi replicado em outros modelos de AM. Mas conseguiu inserir os dados, próximo ao que seria na realidade, de acordo com a fase de avaliação, apresentada em Figura 8.

Para os dados categóricos, temos o problema de desbalanceamento também, e o modelo utilizado não consegue aprender bem o comportamento das classes minoritárias,

[4] <https://imbalanced-learn.org/en/stable/index.html>
[5] <https://matplotlib.org/>
[6] <https://seaborn.pydata.org/>
[7] <https://scikit-plot.readthedocs.io/en/stable/>

conforme Tabela 3. Apesar de ter uma alta acurácia, $95,71\%$, acontecem muitos erros para as classes que tem menos representação e uma sensibilidade baixa. Esse erro introduz um viés para a classe majoritária, mas vale ressaltar, que ainda assim, é um viés menor do que realizado o preenchimento dos dados faltantes com a estatística moda (valor mais frequente). E nem sempre é possível prever o valor de algum atributo com alta performance. Por exemplo, o atributo "SEXO", está balanceado, mas o modelo tem uma taxa de acerto de $67,74\%$. Por isso, considerar um novo valor "desconhecido" apresenta ganhos significativos, por ser um valor neutro.

Tabela 3 – Resultado da classificação do atributo "RENDA_VIZINHANCA"

Classe	**Precision**	**Recall**	**Nº Instâncias**
0	$0,89$	$0,36$	300
1	$0,87$	$0,49$	4653
2	$0,88$	$0,51$	2845
3	$0,88$	$0,38$	60
4	$0,96$	1,00	89381

A árvore de decisão teve melhor desempenho com o preenchimento dos valores numéricos com aprendizagem de máquina. A taxa de erro que foi medida na fase de avaliação, com MSE (*Mean Square Error*), variou entre 10^{-4} e 10^{-2}. Com essa abordagem, foi inserida mais diversidade durante o preenchimento.

4.4 Experimento 2

O seguinte experimento, foi realizado para testar o impacto da remoção de outliers no conjunto de treinamento, de maneira multivariada, através do modelo *Isolation Forest*. O objetivo é aumentar a perfomance, obtendo uma generalização maior após remover essas observações. Após a remoção, a avaliação foi feita com Árvore de Decisão.

Foram variados dois parâmetros principais do *Isolation Forest*, a taxa de contaminação e o número de árvores utilizadas para construir o *ensemble*, os resultados se encontram na Tabela 4.

A remoção de outliers do conjunto de treinamento, detectados de maneira multivariada, aumentaram a performance da classificação, comparado a utilização do conjunto de dados original. A melhor configuração encontrada para a *Isolation Forest*, foi a contaminação 0,5% e 400 árvores para construção da floresta. Com esses experimentos, podemos perceber que na maioria dos casos, é melhor utilizar um número relativamente alto de árvores devido a grande quantidade de atributos presentes nesta base, facilitando para o algoritmo conseguir encontrar melhor o que realmente são outliers.

Tabela 4 – Resultado da remoção de outliers com Isolation Forest

Contaminação	**Nº de Árvores***	**Acurácia**	**ROC**	**KS**
0%	**	61,22%	57,21	14,4
5%	100	60,88%	56,99	14,0
5%	400	60,82%	57,07	14,1
1%	100	61,21%	57,24	14,5
1%	400	61,15%	57,20	14,4
0,5%	100	61,20%	57,23	14,5
0,5%	400	61,25%	57,35	**14,7**
0,1%	100	61,19%	57,21	14,4
0,1%	400	61,26%	57,28	14,6

*Representa o número de *Isolation Trees* utilizados para construir a *Isolation Forest*.
**Referente a avaliação sem remoção de outliers.

Valores muito altos para a taxa de contaminação, eliminam da base muitos exemplos, com isso, temos a perda de informação para o classificador, que deixa de aprender o comportamento de algumas regiões. Além disso, não necessariamente esses dados são outliers, de acordo com a definição em (HAWKINS, 1980), eles deve representam poucas observações, um comportamento incomum.

Essa análise foi feita com resultados obtidos com testes em Árvores de Decisão. Que no geral, são modelos robustos a outliers, o que mostra o verdadeiro ganho na utilização da técnica, mas que não pode ser considerada como verdade para todo modelo de AM. E, de acordo com a Tabela 4, nem sempre a remoção vai trazer resultados melhores.

4.5 Experimento 3

Como foi discutido ao longo do trabalho, existe uma grande vantagem em diminuir a dimensionalidade do problema. Atributos desnecessários podem estar presentes na base de dados, sem influência ou até mesmo atrapalhando. Nesta seção, são utilizadas diferentes técnicas e parâmetros para seleção de um subconjunto de atributos que sejam capazes de manter ou melhorar a performance do modelo Árvore de Decisão. Os resultados são apresentado na Tabela 5.

Foi possível alcançar o mesmo resultado com um número de atributos menor, de acordo com a Tabela 5. Com RFE, houve uma redução de aproximadamente 22% dos atributos e os resultados ainda foram ligeiramente melhores comparado a base de dados completa. Também foram encontrados resultados bem próximos, mas com uma grande redução de atributos.

Tabela 5 – Resultado da Seleção de Atributos

Método	**Tipo**	**Limiar***	**Nº de Atributos**	**KS**
Mutual Information	*Filter*	175	175	13,9
Mutual Information	*Filter*	200	200	14,3
Mutual Information	*Filter*	225	230	14,3
Regressão Logística	*Embedded*	0,0001	268	14,4
Random Forest	*Embedded*	0,1 × Mediana	202	14,1
Extra Trees	*Embedded*	0,25 × Mediana	199	14,2
RFE	*Wrapper*	200	200	14,1
RFE	*Wrapper*	230	230	14,5
RFE	*Wrapper*	245	245	14,5

*Limiar utilizado para seleção dos atributos. Para o tipo *Embedded*, foi utilizado um valor fixo ou heurística. Para as demais, representa o número de atributos que devem ser selecionados

4.6 Experimento 4

O desbalanceamento dos dados faz com que boa parte dos modelos de aprendizagem de máquina tenha um desempenho satisfatório para a classe majoritária, mas ruim para outra, a classe minoritária. Como foi visto na Tabela 1, o conjunto de dados utilizado neste trabalho está desbalanceado. Para balancear o conjunto de treinamento, com objetivo de aumentar a performance dos modelos, principalmente para a classe minoritária, aplicamos as seguintes técnicas de *sampling*: SMOTE, *Random Oversampling* (RO), *Random Undersampling* (RU). E os modelos para avaliar foram Árvore de Decisão (AD) e Regressão Logística (RL).

	Normal			**RO**			**RU**			**SMOTE**		
	KS	**ROC**	**Recall**	**KS**	**ROC**	**Recall**	**KS**	**ROC**	**Recall**	**KS**	**ROC**	**Recall**
AD	14,4	57,2	44,4	14,4	57,2	44,0	14,6	57,3	57,0	14,4	57,2	45,0
RL	27,3	68,4	24,0	27,2	68,4	66,2	27,1	68,3	66,2	27,0	68,2	65,4

O *Recall* calculado é da classe minoritária, a classe de não aprovação (0)

As abordagens de *Sampling* não aumentaram tanto o valor do KS, em muitos casos, essa taxa teve uma ligeira queda. Apesar da medida KS considerar ambas as classes, diferente do foco da acurácia que o importante são os acertos totais, ela faz o cálculo a partir dos acúmulos das distribuições de probabilidades das observações serem de determinada classe. O valor encontrado reflete a separação máxima em determinado limiar, mas não exatamente a taxa de acerto para cada classe de acordo com o limiar utilizado pelo modelo, que por padrão é 0,5. Existem diversas métricas de avaliação, cada uma com objetivo diferente. O que buscamos aqui, é uma acertividade satisfatória para classe minoritária. Para isso, consideramos duas outras métricas, ROC e *Recall*.

Levando em conta essas duas métricas, e não apenas o KS, com a reamostragem dos dados, conseguimos classificar de maneira correta mais exemplos da classe minoritária em alguns casos com o limiar padrão. Porém, quando esse aumento ocorreu, houve uma queda na acertividade para da otura classe. Como a métrica KS, a principal neste projeto, não teve ganhos severos, e por indicar que existe um limiar que separa bem as classes, a técnica de *sampling* não pode ser considerada a melhor abordagem.

4.7 Experimento 5

Depois de investigar melhorias da classificação através da preparação dos dados, construindo um conjunto de treinamento melhor, é feita uma investigação nas técnicas para modelar o problema. Ao longo das etapas, classificadores mais simples foram utilizados, para obter respostas mais rápidas e buscar uma generalização da técnica. O *Gradient Boosting* foi o modelo selecionado para classificação, devido a sua robustez e alta performance, de acordo com a literatura e por resultados em competições atualmente. Também pelos resultados apresentados nos experimentos.

A outra abordagem implementada, foi a detecção do perfil do cliente, através de clustering com *Mini Batch K-Means*, para construir classificadores especializados. A metodologia foi detalhada na seção 3.3. Neste experimento, foi fixado o mesmo classificador, *Gradient Boosting*, para todos os *clusters*, mas cada um utilizou diferentes porções de dados para o treinamento e teste, restringindo aos dados do seu *cluster* apenas. Os testes foram realizados com diferentes números de clusters. Os resultados estão presentes na Tabela 6.

Tabela 6 – Resultado da classificação após a fase de *clustering*

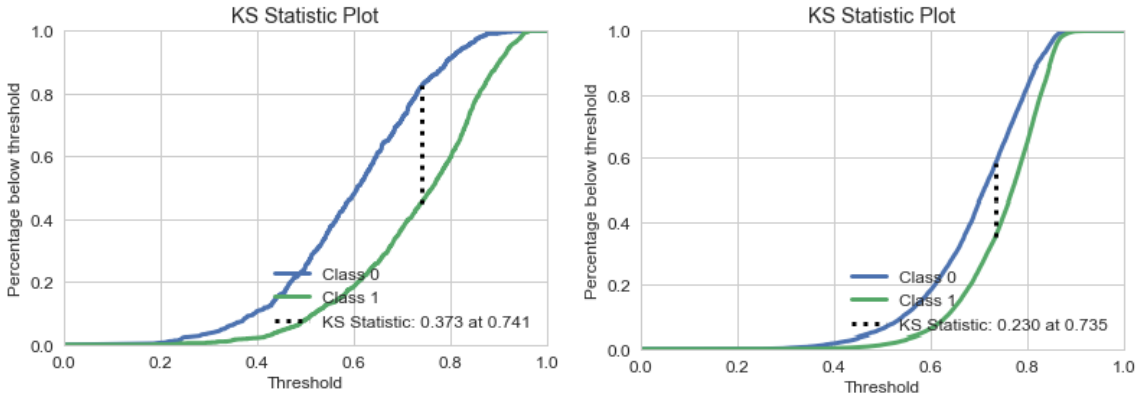
k	**Acurácia**	**ROC**	**KS**
1	69,39%	71,40	31,3
5	**69,53%**	71,62	**31,4**
8	69,52%	**71,65**	**31,4**
12	69,36%	71,50	31,2
20	69,35%	71,37	31,0
30	69,17%	71,03	30,4

O valor 1 implica em apenas um grupo e um classificador, ou seja, sem *clustering*.

Com a criação de um modelo para cada tipo de cliente houve uma melhora nos resultados quando comparado a maneira tradicional, com apenas um classificador. Porém, é preciso encontrar o k ideal, onde k é o número de *clusters*, ou, quantidade de tipos de clientes. Houve uma variação nos resultados obtidos para cada cluster. Para alguns tipos

de clientes, o desempenho obtido foi muito bom, enquanto para outros, os resultados foram ruins, como segue na Figura 14 em *(a)* e *(b)*, respectivamente.

Figura 14 – Resultado obtido pelo classificador *Gradient Boosting* em diferentes *Clusters*



(a) Resultado bom para o *Cluster X (KS 0.373)*

(b) Resultado ruim para o *Cluster Y (KS 0.230)*

Os valores da estatístsica KS na imagem estão entre 0 e 1

O classificador *Gradient Boosting* pode não ter sido o ideal para determinadas regiões. De forma geral, até mesmo sem *clustering*, ele apresenta bons resultados, principalmente quando comparado aos classificadores utilizados anteriormente, Árvore de Decisão e Regressão Logística.

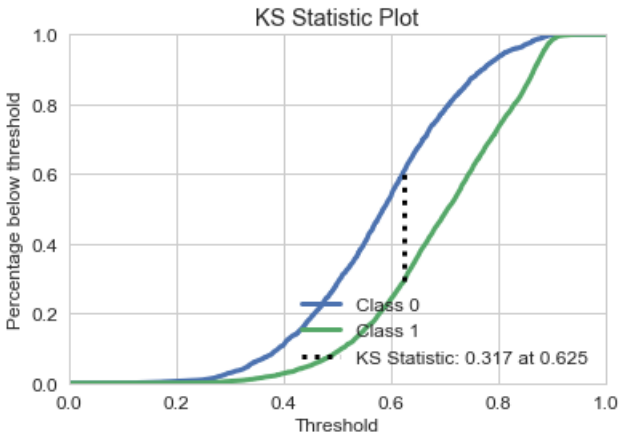
4.8 Experimento 6

Por último, as melhores técnicas foram combinadas para preparação dos dados e modelagem.

- Preenchimento dos dados faltantes: Predição para atributos numéricos e adição de nova constante para atributos categóricos
- Remoção de Outliers: *Isolation Forest* com 400 (árvores) e 0.5% de contaminação.
- Seleção de atributos com Regressão Logística através da abordagem *Embedded*,
- Seleção de classificador utilizando *Clustering* com $k = 12$. Modelo utilizado nos *clusters* = *Gradient Boosting*

De acordo com os resultados da Figura 15, a combinação de algumas técnicas durante a preparação dos dados e a escolha da modelagem correta, trouxeram o melhor resultado até o presente momento. O KS de 31,7 (na escala de 0 a 100) confima a vantagem de pré-processar bem os dados, dado que a mesma abordagem de modelagem que foi utilizada em experimentos anteriores sem realizar o tratamento prévio obtiveram resultados inferiores.

Figura 15 – Gráfico do KS obtido com a combinação das abordagens



KS 0.317, calculado entre 0 e 1.

5 Conclusão

O objetivo deste trabalho foi construir uma solução para análise de risco de crédito através do processo CRISP-DM, ressaltando a importância de cada etapa em projetos de Ciência dos Dados. A etapa de preparação dos dados costuma ser a parte que consome mais tempo. Este trabalho mostra a importância dessa atividade e como ela pode melhorar os resultados finais.

Foi implementado um arcabouço para preenchimento dos dados faltantes através de aprendizagem de máquina, onde os outros atributos são utilizados para identificar padrões do atributo com lacunas. O pré-processamento foi investigado em mais detalhes, aprofundando em outras formas de preencher os dados faltantes, remoção de outliers, seleção de atributos e sampling para balanceamento. Além da preparação, foi apresentada diferentes maneiras para modelagem.

Foi obtido um resultado final de 31,7 com a métrica KS para o conjunto de teste. Considerado um bom resultado para esse problema, segundo os especialistas que forneceram os dados utilizados. A validação é feita durante a compreensão do negócio, primeira fase do CRISP-DM. Confirmando a importância do tratamento dos dados e de seguir um processo para construir uma solução robusta.

5.1 Trabalhos Futuros

Como cada etapa do pré-processamento e modelagem geram uma quantidade grande de experimentos, devido a quantidade de técnicas e parâmetros, neste estudo foi impossível realizar todas combinações. E, talvez, nem seja necessária executar todas. Um trabalho futuro, é utilizar algoritmos de otimização para combinar as técnicas e buscar a melhor solução, sem fazer uma busca exaustiva ou ser através pela intuição.

A remoção de outliers apresentou um ganho durante a avaliação. Outras técnicas para detecção de maneira multivariada podem ser investigadas. E também, maneiras para correção ao invés de excluir os dados, por exemplo, suavização, discretização ou outras transformações.

A seleção de modelos para áreas específicas constituiu parte da solução. O desempenho foi melhor comparado a abordagem tradicional, onde todos os dados são utilizado para treinamento de um único modelo para predição. Algumas áreas tiveram desempenho ruim, uma possível solução para isso, é a introdução de diversidade nos modelos, através de Seleção Dinâmica de Classificadores, na qual existe a possibilidade de outro modelo conseguir mapear melhor aquela região.

Referências

AZEVEDO, A. I. R. L.; SANTOS, M. F. Kdd, semma and crisp-dm: a parallel overview. *IADS-DM*, 2008. Citado na página 10.

CHAPMAN, P. et al. Crisp-dm 1.0 step-by-step data mining guide. 2000. Citado 2 vezes nas páginas 10 e 12.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37, 1996. Citado na página 10.

GUYON, I. et al. Gene selection for cancer classification using support vector machines. *Machine learning*, Springer, v. 46, n. 1-3, p. 389–422, 2002. Citado na página 28.

HAN, J.; PEI, J.; KAMBER, M. *Data mining: concepts and techniques*. [S.l.]: Elsevier, 2011. Citado 3 vezes nas páginas 9, 13 e 16.

HAWKINS, D. M. *Identification of outliers*. [S.l.]: Springer, 1980. v. 11. Citado 3 vezes nas páginas 14, 26 e 34.

HAYASHI, C. What is data science? fundamental concepts and a heuristic example. In: *Data Science, Classification, and Related Methods*. [S.l.]: Springer, 1998. p. 40–51. Citado na página 9.

HE, H.; GARCIA, E. A. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, Ieee, n. 9, p. 1263–1284, 2008. Citado 2 vezes nas páginas 17 e 29.

JOHNSON, R. A.; WICHERN, D. Applied multivariate statistical analysis. prentice hall, englewood cliffs, nj. *Applied multivariate statistical analysis. Prentice-Hall, Englewood Cliffs, NJ.*, 1992. Citado na página 14.

LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. In: IEEE. *2008 Eighth IEEE International Conference on Data Mining*. [S.l.], 2008. p. 413–422. Citado na página 15.

MAREK, T. et al. On the estimation of mutual information. In: JCMF PRAGUE. *Proceedings of ROBUST*. [S.l.], 2008. Citado na página 28.

MITCHELL, T. *Machine Learning*. McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Disponível em: <https://books.google.com.br/books?id=EoYBngEACAAJ>. Citado na página 17.

NG, A. Y. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In: ACM. *Proceedings of the twenty-first international conference on Machine learning*. [S.l.], 2004. p. 78. Citado na página 29.

PIATETSKY, G. Crisp-dm, still the top methodology for analytics, data mining, or data science projects. *KDD News*, 2014. Citado na página 10.

POTDAR, K.; PARDAWALA, T. S.; PAI, C. D. A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, v. 175, n. 4, p. 7–9, 2017. Citado na página 16.

PROVOST, F.; FAWCETT, T. *Data Science for Business: What you need to know about data mining and data-analytic thinking*. [S.l.]: "O'Reilly Media, Inc.", 2013. Citado na página 9.

SAS, I. *SAS Enterprise Miner – SEMMA*. 2014. Disponível em: <https://www.sas.com/en_us/software/enterprise-miner.html>. Citado na página 10.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, JSTOR, p. 267–288, 1996. Citado na página 29.

TUKEY, J. W. *Exploratory data analysis*. [S.l.]: Reading, Mass., 1977. v. 2. Citado na página 21.

ZENG, X. et al. Feature selection using recursive feature elimination for handwritten digit recognition. In: IEEE. *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. [S.l.], 2009. p. 1205–1208. Citado na página 28.