Leonardo José Schettini de Arruda

SISTEMA DE PREVISÃO DE PREÇO DE CRIPTOMOEDAS BASEADO NA POLARIDADE DO SENTIMENTO PÚBLICO

Trabalho de Graduação

Universidade Federal de Pernambuco
www.cin.ufpe.br/~graduacao

RECIFE
2018

Universidade Federal de Pernambuco
Centro de Informática
Graduação em Ciência da Computação

Leonardo José Schettini de Arruda

SISTEMA DE PREVISÃO DE PREÇO DE CRIPTOMOEDAS BASEADO NA POLARIDADE DO SENTIMENTO PÚBLICO

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.

Orientador: *Paulo Salgado Gomes de Mattos Neto*

RECIFE
2018

Leonardo José Schettini de Arruda
Sistema de Previsão de Preço de Criptomoedas Baseado na Polaridade do Sentimento Público/ Leonardo José Schettini de Arruda. – RECIFE, 2018-
61 p. : il. (algumas color.) ; 30 cm.

Orientador Paulo Salgado Gomes de Mattos Neto

Trabalho de Graduação – Universidade Federal de Pernambuco, 2018.

1. Classificação. 2. Séries temporais. 3. Processamento de linguagem natural. 4. Análise de sentimentos. 5. Criptomoedas. I. Orientador. II. Universidade Federal de Pernambuco. III. Centro de Informática. IV. Sistema de previsão de preço de criptomoedas baseado na polaridade do sentimento público

Dedico este trabalho a toda minha família, amigos e professores que me deram o suporte necessário para chegar até aqui.

Resumo

Até hoje, nem mesmo as principais moedas virtuais existentes conseguiram provar totalmente sua utilidade, entretanto, os criadores dos projetos e das moedas fornecem ao público sua utilidade em potencial, o que torna o mercado das criptomoedas altamente especulativo e menos inteligível. Este trabalho visa prever a alta volatilidade no preço de moedas virtuais levando em consideração que seu mercado é altamente especulativo, onde a popularidade da moeda é um direcionador direto do seu preço. Dessa forma, devido aos novos usuários, com o aumento da confiança geral da população sobre o projeto o preço também sobe. Similarmente, a medida que usuários se tornam descrentes da utilidade do projeto, o preço sofre uma queda. Este projeto aborda a previsão de preço como um problema de classificação binária, onde uma classe representa o aumento do preço enquanto a outra simboliza a estagnação e a descida no valor. Através dos experimentos, percebe-se que a polaridade do sentimento associado aos comentários sobre uma criptomoeda - gerada por entusiastas e usuários sobre a utilidade em potencial dos projetos - contém informações relevantes para classificação da flutuação do preço. Dentre os vários modelos de aprendizagem utilizados, o *Random Forest* apresentou o melhor resultado, atingindo 63.91% de acurácia.

Palavras-chave: Séries temporais, Classificação de séries temporais, Análise de sentimentos, Criptomoedas

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

Sumário

1 Introdução

Este trabalho é dividido em 5 capítulos. O primeiro capítulo contém o contexto, motivação e objetivo do projeto desenvolvido. Adicionalmente, são apresentados alguns trabalhos relacionados ao aqui apresentado. Enquanto no segundo capítulo são definidos conceitos importantes para entendimento completo do projeto descrito neste documento, o terceiro capítulo contém a metodologia utilizada para desenvolvimento do sistema proposto. No capítulo quatro são detalhados os experimentos realizados para avaliar o modelo detalhado neste trabalho. Além disso, ainda no capítulo 4, os resultados dos experimentos são analisados. Por fim, o capítulo cinco contém as conclusões obtidas e uma breve explicação sobre os trabalhos futuros.

1.1 Contexto

Grande parte das criptomoedas existentes se caracterizam principalmente pelo fato de garantirem confiabilidade criptográfica sem a necessidade de uma autoridade central - NAKAMOTO (2009) - tornando possível que transações sejam confirmadas sem a necessidade de um intermediário confiável, que é uma das atribuições dos bancos nos sistemas financeiros tradicionais. Isso influencia, majoritariamente, na ideologia por trás da criação de cada moeda, em seu impacto sócio-cultural e na sua precificação. Esta última que também é fortemente influenciada pela maturidade atual das criptomoedas. Nem mesmo as principais moedas virtuais existentes hoje conseguiram provar totalmente sua utilidade, entretanto, o que se tem é a utilidade em potencial dos projetos, o que torna o mercado das criptomoedas altamente especulativo, pois a opinião dos usuários - fator que direciona o preço das criptomoedas, KRISTOUFEK (2015) - pode variar durante o tempo.

O principal objetivo do *Bitcoin*, criptomoeda mais famosa e pioneira no mercado[1], é tornar-se uma moeda do dia-a-dia, facilitando pagamentos entre usuários, garantindo proteção contra fraudes com transações irreversíveis e criptograficamente seguras[2,3]. Porém, através da

[1] https://coinmarketcap.com/
[2] https://bitcoin.org/en/bitcoin-for-individuals
[3] https://bitcoin.org/en/bitcoin-for-businesses

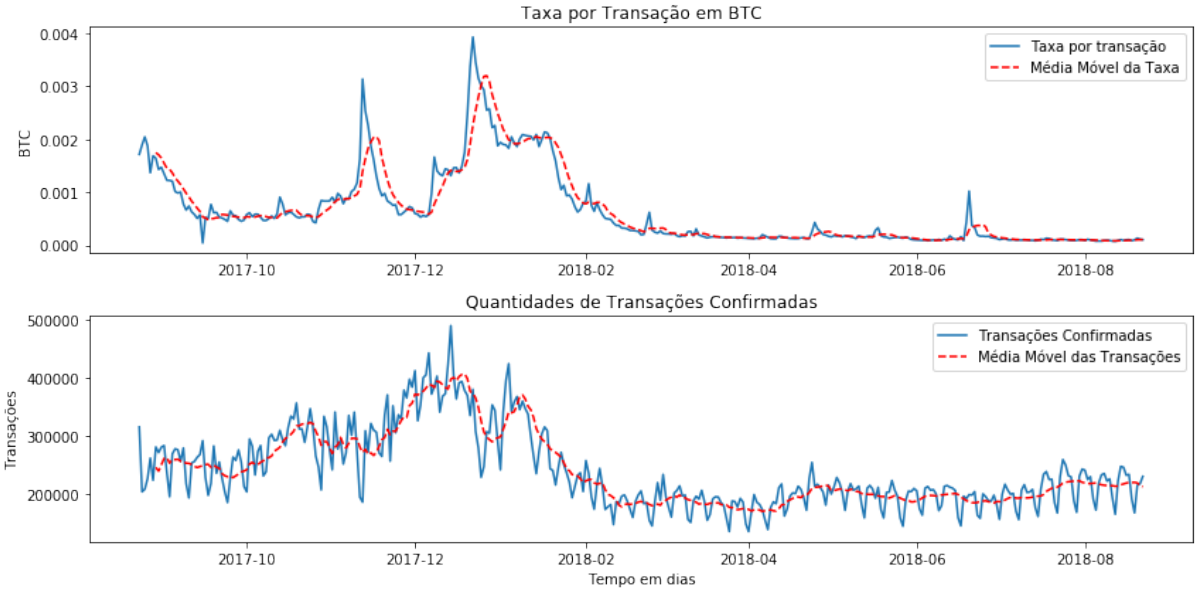

Figura 1.1: Comparação entre o volume de transações e a taxa por transação. Em ambos os gráficos, a linha tracejada representa uma suavização pela técnica de médias móveis com janela de 7 dias.

análise dos dados relacionados à rede *Bitcoin* - presentes no site *blockchain.com*[4] - é possível perceber que conforme o volume de transações na rede aumenta, os usuários que optam por usar tal meio de pagamento sofrem com taxas por transação cada vez mais altas.

Pelos dados utilizados nesta análise - ilustradas parcialmente na Figura 1.1 - pode-se perceber que a taxa por transação atingiu o máximo histórico em dezembro de 2017, quando chegou a custar em torno de 0.0039 *Bitcoin* (BTC), o que na época era equivalente a aproximadamente $59, tornando inviável a utilização da moeda para pequenos pagamentos.

1.2 Motivação

O mercado das criptomoedas é conhecido pela alta volatilidade no preço dos *tokens*[5]. O valor do BTC, por exemplo, chegou a aumentar 3 mil dólares do dia 6 ao dia 7 de dezembro, representando por volta de 20% do valor da moeda. Tamanha volatilidade, junto com a facilidade de comprar moedas, atrai investidores com diferentes níveis de experiência. Adicionalmente, existem diversos fóruns de discussão sobre as moedas virtuais, onde usuários expressam seus anseios, expectativas e avanços na tecnologia da moeda. Dado que o preço de moedas virtuais não é apoiado por metais preciosos ou instituições financeiras, um dos fatores atribuídos à volatilidade das criptomoedas é sua popularidade, KRISTOUFEK (2015).

[4] https://www.blockchain.com/en/charts
[5] Criptomoedas

1.3 Objetivos Gerais

Tendo em vista que a opinião dos usuários é um reflexo direto da notoriadade das moedas virtuais, o objetivo deste trabalho é incluir a opinião dos usuários na previsão do preço de criptomoedas. Para isso, serão utilizadas técnicas de análise de sentimento para extrair indicadores de popularidade de fóruns relacionados à moeda virtual em questão. Em seguida, estas informações serão utilizadas para prever flutuações no preço da criptomoeda estudada.

1.4 Trabalhos Relacionados

O mercado criado por criptomoedas é propício para que sejam feitos estudos sobre a previsão dos preços e número de transações. Estudos desse tipo já são amplamente realizados e formam uma área inteira na estatística, chamada de previsão de séries temporais[6]. De forma geral, os problemas nessa área podem ser divididos em dois grupos, previsão de séries temporais univariadas e multivariadas. O primeiro acontece quando valores passados de uma série temporal são utilizados para prever seus valores futuros. Já no segundo grupo, duas ou mais séries temporais sendo utilizadas para prever uma série em específico.

A hipótese de mercado eficiente - em inglês, *Efficient Market Hypothesis* (EMH) - diz que notícias se espalham rapidamente e são incorporadas ao preço dos títulos financeiros sem atraso. Dessa forma, nem a análise técnica (estudo do mercado em si) nem a análise fundamentalista (estudo de informações financeiras que ajudam investidores a determinar o real valor de ações) seriam capazes de entregar melhores retornos - com risco comparável - do que randomicamente investir em ações individuais, MALKIEL (2003).

Em QIAN; RASHEED (2007) é realizado uma análise dos extensivos testes feitos sobre o modelo de *Random Walk*, que devido a randomicidade do processo é comumente associada ao EMH. A partir desta análise argumenta-se que embora diversos resultados evidenciem que o preço de ações não seja verdadeiramente randômico, é amplamente aceito que o comportamento de tais preços se aproximam do processo de *Random Walk*.

Em BOLLEN; MAO; ZENG (2011) é pontuado que embora notícias exerçam influenta no preço de ações, o estado de humor e sentimento do público devem exercer um papel igualmente importante. Adicionalmente, descobre-se que apenas o humor "calma"se relaciona - através do teste de causalidade de Granger - com a série temporal estudada.

Entretanto, assim como mostrado em KIM et al. (2016), a polaridade de sentimento - categorizada em 5 grupos - contém mudanças que precedem as flutuações do preço do BTC. Resultado este que pode variar sua intensidade de acordo com o tipo de regime do mercado. Esta variação é demonstrada por PHILLIPS; GORSE (2018), que analisa a dependência entre um regime de bolha e a relação de criptomoedas com fatores de uso online potencialmente relevantes.

[6] *Time series forecasting*

2

Fundamentação

Este capítulo apresenta os conceitos básicos necessários para completo entendimento do trabalho aqui desenvolvido. Primeiramente serão introduzidas noções básicas de séries temporais. Em seguida, o capítulo explicitará conceitos sobre os testes estatísticos aplicado sobre as séries temporais utilizadas neste trabalho. Por fim, alguns conceitos de aprendizagem de máquina serão apresentados.

2.1 Análise de Séries Temporais

Uma série temporal é uma sequência de dados indexados em ordem temporal. Comumente, estas séries são formadas por observações sucessivas e igualmente espaçadas no tempo, BROCKWELL; DAVIS (2002). Adicionalmente, é possível dizer que séries temporais provém de um comportamento que pode ser descrito por um modelo matemático.

Com o principal objetivo de entender as características de seus fenômenos geradores, a análise de séries temporais resume-se em estudar o conjunto de dados e estimar um modelo matemático que tenha, possivelmente, gerado os dados analisados. Sendo assim, este campo da computação pode ser aplicado na previsão de dados futuros, classificação de um determinado comportamento e detecção de anomalias.

No contexto do trabalho descrito neste documento, a análise de séries temporais será aplicada no contexto de classificar o comportamento de aumento ou não do preço de uma determinada criptomoeda. Em seguida, serão introduzidos conceitos utilizados ao longo do trabalho.

2.1.1 Tendência e Sazonalidade

Enquanto tendência é caracterizado por mudanças sistemáticas que não apresentam periodicidade, a sazonalidade tem como principal característica a repetição em períodos fixos e conhecidos.

$$X_t = \beta t + \gamma s + Y_t \tag{2.1}$$

Na equação 2.1 verifica-se um modelo com um componente de tendência βt e um componente de sazonalidade γs. Respectivamente, estes componentes podem ser estimados pela técnica de mínimos quadrados e soma de ondas senoidais.

2.1.2 Ordem de Integração

A ordem de integração de uma série temporal faz referência a quantidade de vezes que o processo em questão precisa ser diferenciado para que se transforme num processo estacionário. Sendo assim, uma série com ordem de integração d, denotada como $I(d)$, precisa ser diferenciada d vezes para que um processo estacionário seja obtido.

Seja X um processo temporal, a primeira diferenciação de X, notada como ΔX é dado por:

$$\Delta X_t = X_t - X_{t-1}$$

Por sua vez, ΔX é tido como estacionário caso mantenha um equilíbrio estatístico, de forma que suas propriedades não se alterem pela mudança de tempo. Ou seja, estatísticas como a média e a variância, não se alteram ao longo do tempo.

2.1.3 Ruído Branco

Também chamado de processo Independente e Identicamente Distribuído (IID), é o modelo de séries temporais mais simples, sem tendência nem sazonalidade. O ruído branco é gerado por variáveis aleatórias IID com média zero e variância σ^2. Além disso, não existe dependência entre suas observações, significando que através do valor no ponto t não é possível prever o valor no ponto $t+h$.

2.1.4 Raiz Unitária

Em estatística, raiz unitária uma característica de alguns processos randômicos que pode ser detectada nas inferências estatísticas que envolvem séries temporais. É dito que um processo estocástico tem uma raiz unitária quando uma das raízes do modelo auto-regressivo - do processo em questão - é valorada com 1. Adicionalmente, processos estocásticos que contém raízes unitárias são, necessariamente, não-estacionários.

Dado um processo randômico X que pode ser descrito pelo seguinte modelo auto-regressivo de ordem p:

$$x_t = a_1 x_{t-1} + ... + a_p x_{t-p} + \varepsilon_t$$

Onde a_i se refere aos coeficientes do modelo auto-regressivo e $\mathscr{E}$ é um ruído branco de média zero e variância σ^2 constante. Neste caso, temos que X contém uma raiz unitária dado que, para pelo menos um i válido, $a_i = 1$.

2.1.5 Modelos Estatísticos

Na estatística, existem modelos capazes de modelar processos lineares. Assim como comentado em CHENG et al. (2015), no mundo real, a grande maioria das séries temporais utilizadas são compostas por processos não-lineares. Ainda assim, modelos estatísticos ainda são amplamente utilizados. Além de serem considerados como a performance base para qualquer modelo mais complexo, diversos testes estatísticos dependem da modelagem de processos lineares.

Existem basicamente dois modelos principais, o modelo Auto-Regressivo (AR) e o modelo Médias Móveis (MA). De forma resumida, o primeiro estima valores de um processo linear X utilizando suas ocorrências passadas, enquanto o segundo utiliza seu erro associado. Existe um modelo generalizado, chamado de Auto-Regressivo de Médias Móveis (ARMA). Estes modelos necessitam que a série temporal analisada tenha ordem de integração zero, ou seja, seja estacionária.

2.2 Teste de Dickey-Fuller

O teste estatístico de Dickey-Fuller verifica a hipótese nula de que uma raiz unitária está presente num modelo AR. Dependendo da interpretação do teste, ele apresenta uma hipótese alternativa de que a modelo AR entrada é estacionária, ou estacionária em relação a uma tendência. De forma mais específica, o teste padrão de Dickey-Fuller verifica a existência de uma raiz unitária em séries temporais que possam ser descritas por um modelo AR de ordem 1:

$$x_t = \alpha + \beta t + \rho x_{t-1} + \varepsilon_t \tag{2.2}$$

Na equação 2.2, a variável α representa uma translação no eixo y - também chamado de *drift* - da série, β é um coeficiente sobre o componente de tendência t, ρ é um coeficiente do modelo AR e ε_t é o termo de erro. Esta equação pode ser reescrita com a primeira diferenciação da série X, que é feita a partir da subtração do termo x_{t-1} de ambos os lados da equação, resultando em:

$$x_t - x_{t-1} = \alpha + \beta t + (\rho - 1) x_{t-1} + \varepsilon_t \tag{2.3}$$

A equação 2.3, pode ser escrita utilizando o operador de primeira diferenciação, Δ. Adicionalmente, também é possível considerar que $\rho - 1 = \delta$, resultando em:

$$\Delta x_t = \alpha + \beta t + \delta x_{t-1} + \varepsilon_t \tag{2.4}$$

Como a hipótese nula do teste padrão de Dickey-Fuller é a presença de uma raiz unitária num modelo AR de ordem 1, o teste pode ser executado verificando se $\delta = 0$ na equação 2.4. É importante ressaltar que para a hipótese nula ser aceita, ρ, da equação 2.2, precisa ser igual a 1.

2.2.1 Teste *Augmented* Dickey-Fuller (ADF)

Com o objetivo de englobar séries mais complexas uma versão estendida do teste de raiz unitária foi criada e é referido como Teste Aumentado de Dickey-Fuller, ou ADF. Este teste verifica a hipótese nula de que a série temporal X é $I(n)$, onde $n \geq 1$ contra a hipótese alternativa de que ela é *I(0)*. Isso assume que a série pode ser descrita por um modelo ARMA(p, q).

Utilizando como base a equação 2.4, o teste pode ser expandido para modelos ARMA(p, q) da seguinte forma:

$$\Delta x_t = \alpha + \beta t + \delta x_{t-1} + \gamma_1 \Delta x_{t-1} + \cdots + \gamma_{p-1} \Delta x_{t-p+1} + \varepsilon_t \tag{2.5}$$

Com a hipótese nula de que $\gamma = 0$, significando a presença de uma raiz unitária e, consequentemente, provando que a série é não estacionária.

2.3 Teste de Kwiatkowski-Phillips-Schmidt-Shin (KPSS)

Assim como explicitado em KWIATKOWSKI et al. (1992) - artigo que originalmente propõe o teste de KPSS - testes de hipóteses são executados de forma que a hipótese nula é aceita ao menos que existam fortes evidências contra ela. Por esta razão, testes de raiz unitária falham em rejeitar a hipótese nula em diversas séries temporais econômicas. Este fator é fortemente relacionado à potência estatística[1] dos testes em questão.

Adicionalmente, estudos sugerem que - com o intuito de determinar que séries financeiras/econômicas são estacionárias ou integradas - sejam utilizados testes de raiz unitária em conjunto com testes onde a hipótese nula é a estacionariedade da série. Sendo assim, este trabalho realiza ambos os testes de ADF e de KPSS com o intuito de determinar a ordem de integração das séries utilizadas. Na presença de resultados conflitantes, é considerado que a série é integrada.

[1] https://en.wikipedia.org/wiki/Power_(statistics)

2.4 Teste de Causalidade de Granger

O teste de causalidade de Granger, GRANGER (1980), é um teste estatístico sobre um modelo Vetorial Auto Regresivo (VAR) bivariado que testa se uma determinada série temporal X contém informações relevantes para a previsão de uma outra série Y. A forma mais simples de entender a causalidade de Granger é pensar nela como **precedência**, ou seja, dizer que X "Granger-causa"Y significa quer mudanças em X precedem mudanças em Y.

Este teste assim como mostra o artigo GRANGER; HUANGB; YANG (2000), pode ser de grande utilidade para encontrar influências externas à própria série temporal analisada. Algo muito comum no mercado financeiro, onde eventos podem influenciar o comportamento da série temporal analisada. É importante notar que, para obter resultados confiáveis, os sinais que passam pelo teste de Granger precisam ser estacionários.

2.5 Aprendizagem de Máquina

De forma simples e direta, aprendizagem de máquina é um campo de estudo da computação responsável por permitir que máquinas executem tarefas as quais não foram explicitamente programadas para fazer. Em sua forma mais básica, aprendizagem de máquina é a utilização de algoritmos que interpretam e fazem inferências de dados para então realizar previsões sobre uma determinada questão. Estas definições permitem entendimento geral sobre o que é aprendizagem de máquina. Porém, falham em detalhar sobre como tais algoritmos aprendem e que tipo de previsões são capazes de fazer.

Em relação ao tipo de aprendizagem, os algoritmos podem ser divididos basicamente em dois grupos: **aprendizagem não-supervisionada**, onde as inferências provém apenas das características dos dados, e **aprendizagem supervisionada**, onde o algoritmo é alimentado tanto com as características dos dados como com as previsões alvo de cada entrada. No que diz respeito às previsões, também é possível dividi-las em dois grupos principais. O primeiro grupo, chamado de **regressão**, tem como saída desejada valores reais. O segundo, chamado de **classificação**, busca prever a categoria dos dados recebidos como entrada.

No contexto deste trabalho, na fase de treinamento (aprendizagem), será apenas utilizada a abordagem supervisionada. Além disso, a previsão da flutuação do preço de uma criptomoeda é abordado, neste trabalho, como um problema de classificação.

2.5.1 Aprendizagem Supervisionada

Neste caso, o algoritmo recebe - na fase de treinamento - dados contendo a "resposta correta"para cada instância. Utilizando este trabalho como exemplo, os algoritmos de aprendizagem são treinados com os dados relacionados ao sentimento dos usuários em um determinado dia em conjunto com a direção a qual o preço flutuou no dia seguinte (classe ou rótulo de saída).

Sendo assim, dado um conjunto de instâncias X e um conjunto de rótulos Y, onde para cada $x \in X$ existe um $y \in Y$ correspondente, algoritmos de aprendizagem supervisionada buscam inferir padrões que permitem gerar uma função de mapeamento F, tal que $F : X \Rightarrow Y$, capaz de prever um rótulo para um dado desconhecido, ou seja, um elemento não pertencente ao conjunto X.

2.5.2 Classificação

Problemas de classificação visam, essencialmente, identificar à qual categoria $y \in Y$ pertence uma instância de entrada x, onde Y é um conjunto finito de categorias. No contexto deste trabalho, o conjunto Y é composto por dois elementos: o primeiro representa um aumento no preço da criptomoeda em questão, o segundo representa um "não-aumento", ou seja, tanto uma descida como a não modificação do preço. Caracterizando o problema base deste documento como um problema de classificação binária.

2.5.3 *Averaged One Dependence Estimators* (AODE)

A tarefa de prever - a partir de um conjunto de treinamento classificado - a classe y de um elemento, representado pelo vetor $\vec{x}$, pode ser resumida em selecionar a classe y que tenha a maior probabilidade de acontecer, dado o elemento $\vec{x}$. Sendo assim, pela definição de probabilidade condicional, a classe escolhida é a que maximiza a seguinte equação:

$$P(y|\vec{x}) = \frac{P(y,\vec{x})}{P(\vec{x})} \tag{2.6}$$

A partir desta equação, é fácil perceber que - pela existência de um denominador comum - escolher a classe y que maximize $P(y|\vec{x})$ é o mesmo que escolher a classe que maximize $P(y,\vec{x})$ de tamanho n. Assim como discutido em WEBB; BOUGHTON; WANG (2005), com o objetivo de estimar a probabilidade conjunta da classe y e o elemento $\vec{x}$ com maior confidência a partir das frequências dos dados, a seguinte equação pode ser utilizada:

$$P(y,\vec{x}) = P(y)P(\vec{x}|y) \tag{2.7}$$

A partir da equação 2.7, o problema passa a ser que o elemento $\vec{x}$ não necessariamente ocorre no conjunto de treinamento, ou seja, não pode ser diretamente estimado a partir das amostras de treinamento. Para resolver este problema, o classificador *Naive Bayes* (NB) assume que as características dos dados são independentes dado a classe. Com esta assunção, este classificador estima $P(\vec{x}|y)$ através do produtório da probabilidade de ocorrência de toda característica x_i dada uma classe y:

$$P(\vec{x}|y) = \prod_{i=1}^{n} P(x_i|y) \tag{2.8}$$

O classificador AODE, proposto em WEBB; BOUGHTON; WANG (2005), surge para resolver o problema de independência do NB com pouco custo computacional adicional. Este classificador faz uma suposição mais fraca quanto a independência dos atributos. O AODE cria um classificador para cada atributo x_i, onde x_i é dependente da classe y. Cada um desses classificadores é chamado de *One Dependence Estimators* (ODE). A probabilidade de ocorrência da classe y em conjunto com o elemento $\vec{x}$, é então dada considerando essa dependência:

$$P(y,\vec{x}) = P(y,x_i)P(x|y,x_i) \tag{2.9}$$

Similarmente a estimativa de probabilidade utilizada pelo NB, o termo contendo a probabilidade condicional na equação 2.9 pode ser estimada da seguinte forma:

$$P(x|y,x_i) = \sum_{i:1\leq i\leq n \wedge F(x_i)\geq m} \hat{P}(x_i|y) \prod_{j=1}^{n} \hat{P}(x_j|y,x_i) \tag{2.10}$$

onde, $F(x_i)$ retorna a frequência em que o valor de x_i aparece na *i-ésima* característica de todos os elementos do conjunto de treinamento. Dessa forma, a frequência mínima m funciona como um filtro de características. O classificador AODE, por sua vez, retorna a classe y que maximize a equação 2.10.

Seguindo a definição de probabilidade condicional - o artigo que originalmente propõe este classificador (WEBB; BOUGHTON; WANG (2005)) - as probabilidades da equação 2.10 podem ser estimadas utilizando as seguintes equações:

$$\hat{P}(y) = \frac{F(y)+1}{K} \tag{2.11}$$

$$\hat{P}(y,x_i) = \frac{F(y,x_i)+1}{K+v_i} \tag{2.12}$$

$$\hat{P}(y,x_i,x_j) = \frac{F(y,x_i,x_j)+1}{K+v_i v_j} \tag{2.13}$$

onde $F(\cdot)$ é a frequência a qual a combinação de termos aparecem no conjunto de treinamento, K é o total de instâncias de treinamento e v_a é a quantidade de valores possíveis para o atributo a. Ao longo deste trabalho, esta forma de estimar as probabilidades será chamada de modo *paper*. Adicionalmente, outro modo - chamado de *count* - foi implementado. Neste último modo as probabilidades são estimadas a partir da frequência dos dados, ou seja, nas equações 2.11, 2.12 e 2.13 o termo $+1$ no numerador e o v_a no denominador são ignorados.

2.5.4 *Support Vector Machine* (SVM)

O SVM é um algoritmo de aprendizagem de máquina supervisionado que pode ser utilizado tanto para classificação como para regressão. No contexto deste trabalho este algoritmo foi apenas utilizado para classificação. SVM é um algoritmo baseado na ideia de encontrar o hiperplano que melhor divide um conjunto de dados pelas suas classes.

Este classificador representa as instâncias num plano *n-dimensional*, onde *n* é o total de características de cada elemento. Com isso, o modelo busca criar um hiperplano que divida a amostra de dados por classe de forma que este hiperplano tenha a maior distância possível para os elementos usados na fase de treinamento.

Por vezes, o conjunto de dados utilizados não são linearmente separáveis, o que traz problemas para o SVM, dado que quanto maior a sobreposição das instâncias de treinamento, maior é a perda na performance do modelo. Sendo assim, por vezes, aumentar a dimensionalidade dos dados é necessária para diminuir a sobreposição, aproximando o conjunto de um linearmente separável.

Além disso, este modelo contém dois hiper-parâmetros que ajustam o erro aceitável para criação do hiperplano separador e selecionam quais instâncias terão maior influência na criação deste hiperplano, respectivamente chamados de *C* e *gamma*. Quanto maior for o primeiro, menos erro é tolerado. Em contrapartida, quanto menor o segundo, as instâncias que exercem maior inflência na geração do hiperplano serão as mais distantes do mesmo.

2.5.5 Árvore de Decisão

De forma resumida, árvores de decisão são formadas por um conjunto de regras do tipo "se-então-senão", criadas a partir dos dados de treinamento. Sendo assim, este modelo é um algoritmo supervisionado que cria uma estrutura em formato de árvore onde cada nó representa representa uma condição que leva a duas ou mais arestas (opções) que eventualmente levam a uma folha, que simboliza a classificação ou decisão.

A principal vantagem deste modelo é a facilidade de interpretar os resultados obtidos. Dado que toda decisão tomada se baseia no conjunto de regras criados, descobrir como o algoritmo chegou em determinado resultado é trivial. Enquanto a maior desvantagem do modelo está na possibilidade de gerar *overfitting* dos dados, ou seja, aprender tão bem a amostra de treinamento que o modelo se torna incapaz de generalizar para o conjunto de teste. Adicionalmente, este modelo é sensível para mudanças no conjunto de treinamento e ruídos nos dados.

2.5.6 *Random Forest*

O *Random Forest* é um *ensemble* de árvores de decisão, ou seja uma combinação de diversas árvores. Durante seu treinamento, várias árvores de decisão são treinadas de forma que as características e instâncias usadas nesta fase são escolhidas aleatoriamente. Dessa forma, as unidades que formam o *Random Forest* são independentes.

Esta técnica permite, através da diversidade das árvores de decisão, que a resposta final seja construída a partir de pontos de vistas diferentes, onde erros aleatórios são mutualmente cancelados ao mesmo tempo que decisões corretas são reforçadas. Em comparação com uma única árvore de decisão, o *Random Forest* é mais robusto para mudanças no conjunto de treinamento. Adicionalmente, devido à diversidade garantida na fase de treinamento, este modelo é mais resistente ao *overfitting*.

2.5.7 *Multi Layer Perceptron* (MLP)

Este modelo é uma rede neural - conjunto de algoritmos conhecidos por sua capacidade de generalização - cuja unidade básica de processamento é o *perceptron*. Sua arquitetura se divide em uma camada de entrada, *n* camadas intermediárias e uma camada de saída.

A quantidade de camadas intermediárias, assim como o total de neurônios em cada uma destas camadas, é definido pelo usuário. Entretanto, a camada de entrada contém d neurônios, onde d é a dimensionalidade dos dados. Por fim, no contexto de classificação, a camada de saída é composta por c neurônios, onde c é a quantidade de rótulos possíveis.

Este modelo tem os neurônios da camada x totalmente conectados com o da camada $x+1$. Ele é treinado usando a técnica de *backpropagation* e os neurônios de saída podem receber diversas funções de ativação.

2.5.8 Deep Forest

Com o objetivo de explorar a possibilidade de criar modelos de aprendizagem profunda[2] a partir de módulos não-diferenciáveis, em ZHOU; FENG (2017), é proposto o *Deep Forest*, um ensemble de *Random Forest* com uma estrutura em camada e um pré-processamento de entrada que simula a extração de *features* de redes convolucionais.

[2] *Deep Learning*

No artigo citado, o sucesso de tais modelos de *Deep Learning* é atribuído a três características principais: processamento camada por camada, transformações *in-model* das características e suficiente complexidade do modelo. Zhou e Feng argumentam que modelos de aprendizagem profunda são muito mais complicados do que precisam. Por isso, o modelo proposto é capaz de determinar sua complexidade - quantidade de camadas - de uma maneira que depende dos dados. A arquitetura de camadas do modelo é descrita na Figura 2.1.

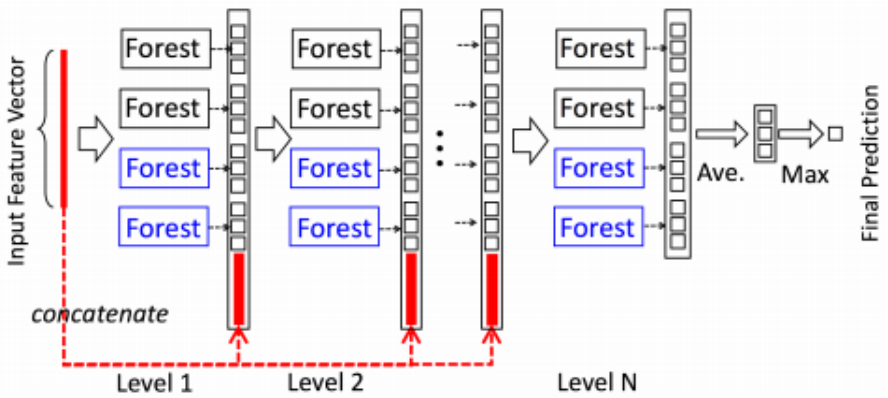

Figura 2.1: Arquitetura de cascata do *Deep Forest*. Imagem coletada de ZHOU; FENG (2017)

Para contemplar relações entre características, o *Deep Forest* utiliza janelas deslizantes para transformar a entrada em múltiplas instâncias, com o mesmo rótulo da entrada original. À princípio estão disponíveis dois métodos para tal transformação. Enquanto o primeiro é próprio para problemas de séries temporais univariados - onde apenas uma série temporal é utilizada - o segundo considera relações espaciais da entrada. Esta etapa é descrita na Figura 2.2.

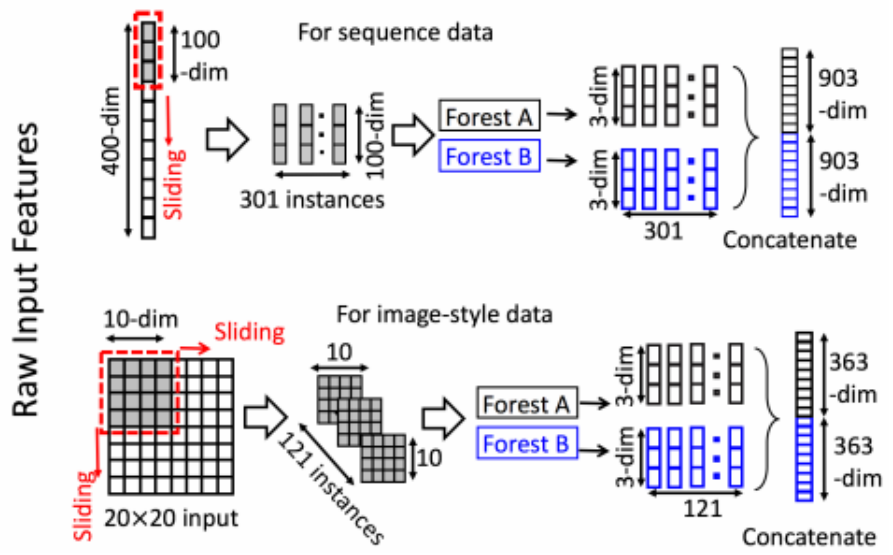

Figura 2.2: Extração de características do *Deep Forest*. Imagem coletada de ZHOU; FENG (2017)

3

Metodologia

O sistema proposto por este trabalho se resume em 4 etapas principais. A primeira, de pré-processamento dos dados, gera séries temporais representando o sentimento dos usuários da criptomoeda em questão bem como padroniza e transforma as séries. Em seguida, existe a fase de refinamento, onde o teste de causalidade de Granger - discutido na seção 2.4 - é utilizado para identificar as séries mais expressivas para prever as flutuações no preço da moeda virtual analisada. As séries relevantes são utilizadas como entrada para a etapa de aprendizagem. Nela, diversos modelos são treinados para que possam ser avaliados e comparados na quarta e última fase, de avaliação. Esta última, apesar de fazer parte da metodologia é discutida com mais detalhes no capítulo 4.

3.1 Tecnologias

O projeto aqui descrito foi desenvolvido utilizando a linguagem de programação Python[1], que oferece uma gama de bibliotecas contendo diversos algoritmos de aprendizagem de máquina. Dentre as bibliotecas utilizadas estão: *Pandas*, MCKINNEY (2010), utilizado para facilitar a manipulação do conjunto de dados; *statsmodels*[2] que possui implementações dos testes estatísticos utilizados neste trabalho e *Scipy*, JONES et al. (2001), utilizado por disponibilizar a biblioteca *Numpy*[3], que contém um poderoso *array* n-dimensional e uma implementação do *z-score* otimizada usando *numpy arrays*. Além disso, os gráficos deste trabalho foram gerados com auxílio da biblioteca *matplotlib*, HUNTER (2007).

O classificador AODE foi implementado manualmente, utilizando dos pacotes citados no parágrafo anterior. Para o *Deep Forest* foi utilizado o código disponibilizado por ZHOU; FENG (2017)[4]. O restante dos algoritmos de aprendizagem de máquina - utilizados ao longo deste projeto - foram providos pelo pacote *scikit-learn*, PEDREGOSA et al. (2011).

[1] https://www.python.org/
[2] https://www.statsmodels.org/stable/index.html
[3] http://www.numpy.org/
[4] https://github.com/kingfengji/gcForest

3.2 Base de Dados

Para desenvolvimento e execução do sistema aqui proposto, se faz necessário que o preço e a quantidade de transações de criptomoedas, assim como a opinião e sentimento de seus usuários, sejam coletados. No escopo do trabalho corrente, tais informações foram extraídas apenas para a moeda virtual com maior valor de mercado[5] nos dias de hoje, que é o *Bitcoin*. Entretanto, este sistema pode ser aplicado para qualquer criptomoeda existente, dado que exista uma fonte geradora constante de comentários (opiniões) sobre a mesma. Assim como proposto em PAK; PAROUBEK (2010), o Twitter pode ser utilizado como esta fonte de comentários.

Em relação ao preço do *Bitcoin*, foi utilizado o valor diário de fechamento, ou seja, o último preço registrado para os dias em questão. Estas informações foram extraídas do *Coindesk*[6]. Enquanto a série temporal da quantidade de transações diárias foi extraída do site *bitcoin.com*[7]. No que diz respeito às informações relacionadas às opiniões dos usuários, foram utilizados os dados coletados por KIM et al. (2016). Tais dados contém informações relacionadas às *threads* - formadas por um tópico principal e comentários - que foram postadas no fórum *Bitcoin talk*[8] datando desde 22 de Novembro de 2009 até 01 de Fevereiro de 2016. Amostras dos arquivos relativos ao *Bitcoin* estão ilustradas nas Tabelas 3.1 e 3.2. Nelas, é possível notar a existência de diversas entradas para a mesma data, onde cada uma acompanha sua pontuação dada pelo algoritmo VADER.

Data	Tópico	Conteúdo	Réplicas	Visto	VADER
2015-10-24	Bitcoin Passphrase [...]	What if you [...]	33	767	0,9836
2015-10-24	Is Bitcoin 'real [...]	There's a [...]	910	34914	0,2311
2015-10-25	China (Unofficially) [...]	I am surprised [...]	64	2300	0,6706

Tabela 3.1: Amostra das informações - utilizadas neste trabalho - relativas aos tópicos principais postadas no *Bitcoin Talk*.

Data	Comentário	VADER
2015-10-24	Good thing I haven't invested into something like [...]	-0,1618
2015-10-24	Made quite a few coins in 2013 and lost almost everything [...]	-0,296
2015-10-25	Well I don't have a whole bunch of experience yet [...]	-0,7096

Tabela 3.2: Amostra dos dados referentes às respostas dos tópicos postados no fórum *Bitcoin Talk*.

[5] https://coinmarketcap.com/
[6] https://www.coindesk.com/price
[7] https://charts.bitcoin.com/btc/chart/daily-transactions
[8] https://bitcointalk.org/

3.3 Pré-processamento

A etapa de pré-processamento pode ser dividida em 3 subetapas, correspondentes à categorização do sentimento dos usuários, transformação e padronização dos dados. Enquanto a primeira se resume em categorizar - entre classes pré-definidas - a polaridade do sentimento dos usuários acerca à moeda virtual escolhida. A última visa normalizar os dados dentro de um intervalo controlado.

A subetapa de transformação dos dados tem duas funções primordiais. Uma delas foca em transformar os dados categorizados na fase anterior em séries temporais. A outra modifica a série temporal contendo o preço em uma série binária, ou seja, com apenas dois valores possíveis, representando um aumento ou não no preço da moeda. Todas as etapas brevemente descritas acima, são detalhadas nas próximas seções na ordem em que foram executadas.

3.3.1 Categorização do Sentimento

Os documentos disponibilizados por KIM et al. (2016) dispõem, dentre outros dados, de informações sobre a polaridade do sentimento de cada um dos tópicos e comentários, assim como pode ser visto na Tabela 3.1. Para a análise de sentimento, o artigo citado utilizou o algoritmo VADER - proposto em HUTTO; GILBERT (2014) - que através de um sistema baseado em regras é capaz de tratar gírias e neologismo, encontrados com frequência em fóruns e redes sociais. Este algoritmo normaliza a polaridade do sentimento entre -1 e 1.

Neste trabalho, assim como em KIM et al. (2016), a pontuação da polaridade p foi classificada entre 5 categorias da seguinte forma:

$$
\begin{aligned}
-1 \leq p < -0.6 &= muito\ negativo \\
-0.6 \leq p < -0.2 &= negativo \\
-0.2 \leq p < 0.2 &= neutro \\
0.2 \leq p < 0.6 &= positivo \\
0.6 \leq p \leq 1 &= muito\ positivo
\end{aligned}
$$

A categorização do sentimento substitui a coluna **VADER** por outras 5 colunas - uma para cada categoria citada no parágrafo anterior - preenchidas com valores binários. Utilizando como exemplo a última entrada da Tabela 3.1, datando 25 de Outubro de 2015, a classificação do sentimento resulta na classe **muito positivo**, o que significa que, para este registro, apenas a coluna **muito positivo** será preenchida com **1**. Adicionalmente, pode-se descartar, das tabelas citadas, as colunas de **Tópico**, **Conteúdo** e **Comentário**, dado que todas as informações relevantes - para o trabalho aqui proposto - já foram extraídas e devidamente categorizadas.

3.3.2 Transformação dos Dados

Após categorizar o sentimento dos tópicos e das respostas relacionadas ao *Bitcoin* ainda se faz necessário combinar as múltiplas entradas para a mesma data em um único registro. Intuitivamente, esta combinação é feita pela soma das colunas restantes nas tabelas geradas pela etapa anterior. Como resultado, existem agora 5 séries temporais (sinais) para cada tipo de informação, tópicos e comentários. Similarmente, a partir das visualizações individuais de cada tópico e a contagem absoluta de tópicos e comentários por dia, três séries adicionais foram geradas. Uma delas representa o total de visualizações diárias do fórum *BitcoinTalk*, enquanto as outras equivalem ao total de tópicos e comentários postados diariamente.

Sendo assim, para cada tipo de informação (tópicos e comentários), existem 6 séries temporais, onde 5 delas refletem as polaridades de sentimento classificadas na etapa anterior, e uma representa o total de postagens diárias. Além destas informações existe uma outra, relativa ao total de visualizações diárias do fórum. Estas, somadas à série temporal da quantidade de transações diárias na rede *Bitcoin*, totalizam 14 sinais que podem ser utilizados na previsão da flutuação do preço.

Ao observar as séries temporais relativas às polaridades de sentimentos, exemplificadas pelos sentimentos muito negativos ilustrados na Figura 3.1, é possível perceber dois problemas primordiais. O primeiro diz respeito aos dados faltantes - mostrados como aparentes falhas nos gráficos - já o segundo, é relativo à escala dos dados. Com o objetivo de manter a continuidade temporal dos dados, as informações faltantes são preenchidas com o valor **zero**, o que representa a falta de postagens no dia em questão.

Sobre o segundo problema, é perceptível que a escala da quantidade de tópicos negativos é consideravelmente menor do que a escala da série de comentários também negativos. Com o intuito de normalizar todas as séries geradas dentro de um mesmo intervalo, mantendo as mesmas propriedades estatísticas da série original, os dados foram padronizados através do *z-score*. Considerando uma série temporal x, o *z-score* do ponto t é dado por:

$$zscore(x_t) = \frac{x_t - \mu_x}{\delta_x}$$

onde μ_x e δ_x são, respectivamente, a média e o desvio padrão da série temporal x.

3.3.3 Binarização das Séries de Saída

No contexto deste trabalho, a previsão de flutuação do preço do *Bitcoin* é abordado como um problema de classificação binária, ou seja, com apenas duas classes alvo[9]. Uma das classes em questão caracteriza o aumento no preço, enquanto a outra representa tanto a estaguinação como a descida do preço. A classe do dia t é referente à modificação do preço no dia $t+1$. Para

[9]Classes de saída ou rótulos

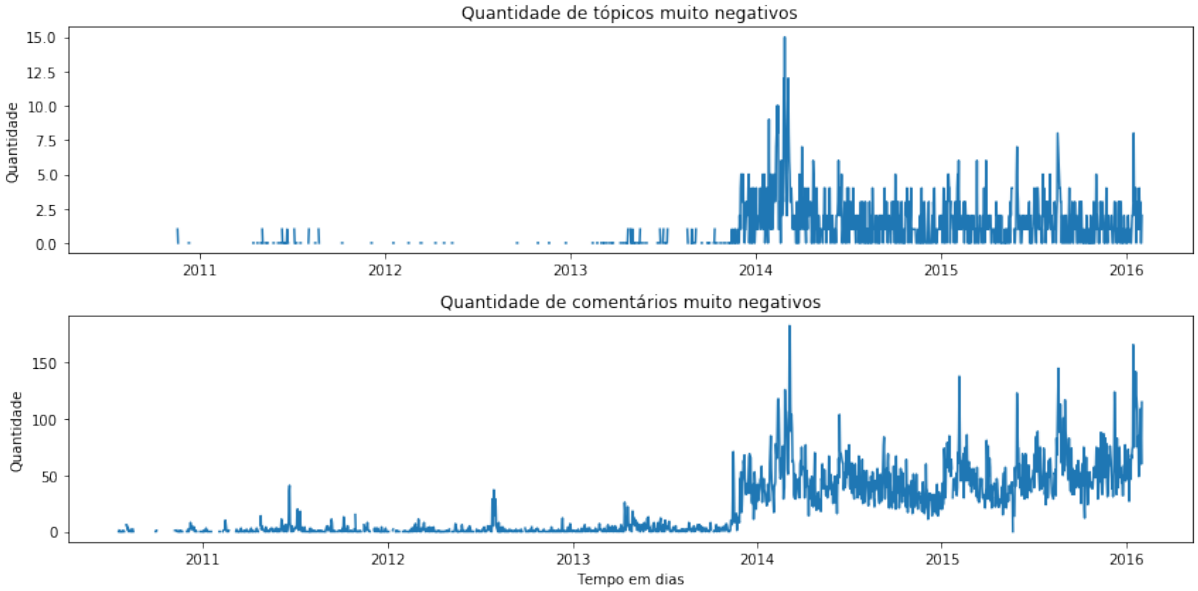

Figura 3.1: Séries temporais de tópicos e comentários categorizados como muito negativo.

binarizar a série do preço p, a seguinte fórmula é aplicada:

$$p_t = degrau(p_{t+1} - p_t)$$

onde a função *degrau* retorna -1 para valores iguais ou menores que zero e $+1$ para valores positivos. Vale a pena pontuar que $p_{t+1} - p_t$ é a primeira diferenciação da série p. Um exemplo dessa transformação pode ser encontrado na Tabela 3.3, na qual percebe-se que a classe de um determinado dia diz respeito a flutuação no dia seguinte. Isso é necessário para que informações de um determinado dia consigam prever a flutuação futura do preço.

Data	Preço	Primeira diferenciação	Classe
2016-01-25	390,66	-	+1
2016-01-26	391,43	0,77	+1
2016-01-27	394,63	3,20	-1
2016-01-28	379,38	-15,25	-1
2016-01-29	378,2	-1,18	-

Tabela 3.3: Exemplo do passo a passo da binarização da série do preço para alguns dias utilizados neste trabalho.

3.4 Refinamento dos Dados

A fase de refinamento visa filtrar apenas as séries com informações relevantes para a previsão do preço da criptomoeda escolhida. Além de reduzir o ruído, removendo séries

temporais não expressivas, a fase de refinamento reduz a dimensionalidade dos dados, o que diminui a complexidade do modelo de aprendizagem.

Para isso, foi utilizado o teste de Granger, descrito em detalhes na Seção 2.4. Com o objetivo de evitar resultados não confiáveis na fase de refinamento, as séries temporais aqui utilizadas foram reduzidas. Para tanto, foram descartadas todas as entradas com datas até Dezembro de 2013, onde estão presentes 99% dos dados faltantes, preenchidos manualmente na fase de pré-processamento.

Este teste estatístico assume que as séries - ou sinais - analisadas são estacionárias. Para tal fim, a primeira diferenciação foi calculada para todas as séries temporais utilizadas no trabalho aqui descrito. Com o intuito de confirmar a estacionariedade das séries, foram aplicados os testes estatísticos *AdFuller* e *KPSS*. Este processo é repetido até que ambos os testes concordem que o sinal em questão é estacionário. Foi-se constatado que todas as séries aqui utilizadas tem ordem de integração igual a 1 - concisamente denotadas como *I(1)* - significando que precisam ser diferenciadas apenas uma vez para se tornarem *I(0)*, ou seja, estacionárias.

Uma vez que todos os sinais são *I(0)*, o teste de Granger testa, no contexto deste trabalho, a hipótese nula de que as 14 séries de entrada **não contém** informações relevantes para a previsão do preço do *Bitcoin*. Adicionalmente, é válido perceber que, para este teste estatístico foi aplicado considerando um defasamento (*time lag*) variável entre 1 e 45. Enquanto os resultados completos podem ser encontrados no Apêndice A, aqui são apenas mostrados os resultados até a defasagem 13.

Considerando um nível de confiança de 95%, é possível rejeitar a hipótese nula para qualquer *p-value* inferior a 0.05. Os *p-values* resultantes do teste de causalidade de Granger podem ser encontrado nas Tabelas 3.4, 3.5 e 3.6 onde - para melhor visualização - os resultados que permitem declinar a hipótese nula estão marcados em negrito. Os resultados completos estão nas Tableas A.1, A.2 e A.3

3.4.1 Análise dos Resultados

Analisando os resultados é perceptível que tanto os tópicos como os comentários positivos contém informações úteis para a previsão do preço independente da defasagem testada, pois para todo *time lag* o *p-value* resultante está dentro do nível de significância de 5%. Além destas, também são considerados como relevantes informações que rejeitam a hipótese nula do teste de Granger em pelo menos 75% das defasagens testadas. Dessa forma, para a próxima fase são consideradas as séries temporais diárias relativas aos tópicos e comentários positivos e muito positivos, comentários negativos e total de tópicos e comentários.

Quando a atenção é voltada para os resultados do teste de causalidade de Granger entre o preço e as informações restantes, percebe-se que para boa parte das defasagens testadas, tais informações **não** apresentam melhoria na previsão do preço. É importante notar que estas análises foram feitas a partir das tabelas presentes no Apêndice A. Sendo assim, as Tabelas 3.4,

3.5 e 3.6 podem levar a conclusões diferentes.

Time lag	Muito negativos		Negativos		Visualizações
	Respostas	Tópicos	Respostas	Tópicos	
01	$\mathbf{2.204 \times 10^{-2}}$	5.632×10^{-1}	1.461×10^{-1}	3.238×10^{-1}	6.140×10^{-2}
02	9.619×10^{-2}	4.791×10^{-1}	$\mathbf{2.606 \times 10^{-2}}$	1.792×10^{-1}	$\mathbf{4.356 \times 10^{-2}}$
03	$\mathbf{3.288 \times 10^{-3}}$	1.761×10^{-1}	$\mathbf{6.660 \times 10^{-4}}$	3.053×10^{-1}	$\mathbf{3.330 \times 10^{-3}}$
04	$\mathbf{4.875 \times 10^{-3}}$	1.725×10^{-1}	$\mathbf{1.225 \times 10^{-3}}$	5.279×10^{-2}	$\mathbf{7.034 \times 10^{-4}}$
05	$\mathbf{1.159 \times 10^{-2}}$	2.578×10^{-1}	$\mathbf{3.275 \times 10^{-3}}$	6.085×10^{-2}	$\mathbf{1.823 \times 10^{-3}}$
06	$\mathbf{9.932 \times 10^{-3}}$	1.607×10^{-1}	$\mathbf{1.240 \times 10^{-2}}$	1.809×10^{-1}	$\mathbf{1.419 \times 10^{-2}}$
07	$\mathbf{2.009 \times 10^{-2}}$	2.005×10^{-1}	$\mathbf{2.510 \times 10^{-2}}$	3.050×10^{-1}	$\mathbf{2.614 \times 10^{-2}}$
08	$\mathbf{2.177 \times 10^{-2}}$	2.395×10^{-1}	$\mathbf{2.314 \times 10^{-2}}$	3.994×10^{-1}	$\mathbf{1.728 \times 10^{-2}}$
09	$\mathbf{3.539 \times 10^{-2}}$	1.140×10^{-1}	$\mathbf{1.994 \times 10^{-2}}$	4.254×10^{-1}	$\mathbf{1.836 \times 10^{-2}}$
10	$\mathbf{1.507 \times 10^{-2}}$	6.278×10^{-2}	$\mathbf{2.623 \times 10^{-2}}$	3.538×10^{-1}	$\mathbf{2.217 \times 10^{-2}}$
11	$\mathbf{1.402 \times 10^{-2}}$	9.129×10^{-2}	$\mathbf{2.468 \times 10^{-2}}$	4.154×10^{-1}	7.140×10^{-2}
12	$\mathbf{2.332 \times 10^{-2}}$	1.410×10^{-1}	$\mathbf{1.595 \times 10^{-2}}$	4.685×10^{-1}	8.988×10^{-2}
13	$\mathbf{2.111 \times 10^{-2}}$	1.825×10^{-1}	$\mathbf{1.217 \times 10^{-2}}$	5.357×10^{-1}	7.340×10^{-2}

Tabela 3.4: *P-values* resultantes do teste de Granger aplicado entre o total de visualizações do fórum, comentários e tópicos negativos e muito negativos com o preço do *Bitcoin*.

Time lag	Neutros		Total		Transações
	Respostas	Tópicos	Respostas	Tópicos	
01	6.112×10^{-2}	5.727×10^{-1}	$\mathbf{3.729 \times 10^{-2}}$	8.422×10^{-1}	8.177×10^{-2}
02	2.616×10^{-1}	$\mathbf{3.163 \times 10^{-3}}$	9.704×10^{-2}	$\mathbf{1.506 \times 10^{-2}}$	3.335×10^{-1}
03	$\mathbf{1.042 \times 10^{-2}}$	$\mathbf{4.304 \times 10^{-3}}$	$\mathbf{1.496 \times 10^{-4}}$	$\mathbf{2.666 \times 10^{-4}}$	5.627×10^{-1}
04	$\mathbf{6.156 \times 10^{-3}}$	$\mathbf{1.185 \times 10^{-3}}$	$\mathbf{1.218 \times 10^{-4}}$	$\mathbf{2.677 \times 10^{-6}}$	7.033×10^{-1}
05	$\mathbf{7.948 \times 10^{-3}}$	$\mathbf{7.180 \times 10^{-3}}$	$\mathbf{3.018 \times 10^{-4}}$	$\mathbf{5.006 \times 10^{-5}}$	7.476×10^{-1}
06	$\mathbf{4.211 \times 10^{-2}}$	$\mathbf{9.786 \times 10^{-3}}$	$\mathbf{3.971 \times 10^{-3}}$	$\mathbf{7.898 \times 10^{-4}}$	9.110×10^{-1}
07	9.657×10^{-2}	$\mathbf{6.057 \times 10^{-3}}$	$\mathbf{8.313 \times 10^{-3}}$	$\mathbf{6.625 \times 10^{-4}}$	9.326×10^{-1}
08	9.692×10^{-2}	$\mathbf{9.179 \times 10^{-3}}$	$\mathbf{7.211 \times 10^{-3}}$	$\mathbf{1.054 \times 10^{-3}}$	9.582×10^{-1}
09	1.098×10^{-1}	$\mathbf{2.336 \times 10^{-2}}$	$\mathbf{1.086 \times 10^{-2}}$	$\mathbf{2.420 \times 10^{-3}}$	9.724×10^{-1}
10	1.309×10^{-1}	$\mathbf{3.215 \times 10^{-2}}$	$\mathbf{9.583 \times 10^{-3}}$	$\mathbf{1.917 \times 10^{-3}}$	9.914×10^{-1}
11	8.845×10^{-2}	5.805×10^{-2}	$\mathbf{4.891 \times 10^{-3}}$	$\mathbf{2.623 \times 10^{-3}}$	9.883×10^{-1}
12	1.206×10^{-1}	5.747×10^{-2}	$\mathbf{6.653 \times 10^{-3}}$	$\mathbf{2.063 \times 10^{-3}}$	9.954×10^{-1}
13	1.691×10^{-1}	6.027×10^{-2}	$\mathbf{6.462 \times 10^{-3}}$	$\mathbf{4.166 \times 10^{-3}}$	9.984×10^{-1}

Tabela 3.5: *P-values* resultantes do teste de causalidade de Granger aplicado entre os comentários e tópicos neutros, total de postagens diárias e total de transações diárias com o preço do *Bitcoin*.

Time lag	Muito positivos		Positivos	
	Respostas	Tópicos	Respostas	Tópicos
01	3.817×10^{-1}	8.018×10^{-1}	$\mathbf{8.915 \times 10^{-3}}$	$\mathbf{4.852 \times 10^{-2}}$
02	4.416×10^{-1}	6.042×10^{-1}	$\mathbf{3.122 \times 10^{-2}}$	$\mathbf{4.322 \times 10^{-2}}$
03	$\mathbf{2.201 \times 10^{-3}}$	$\mathbf{1.825 \times 10^{-3}}$	$\mathbf{9.802 \times 10^{-6}}$	$\mathbf{2.131 \times 10^{-4}}$
04	$\mathbf{1.226 \times 10^{-3}}$	$\mathbf{2.106 \times 10^{-6}}$	$\mathbf{3.430 \times 10^{-6}}$	$\mathbf{9.414 \times 10^{-5}}$
05	$\mathbf{2.686 \times 10^{-3}}$	$\mathbf{5.947 \times 10^{-6}}$	$\mathbf{1.606 \times 10^{-5}}$	$\mathbf{3.686 \times 10^{-4}}$
06	$\mathbf{3.383 \times 10^{-2}}$	$\mathbf{9.171 \times 10^{-5}}$	$\mathbf{3.809 \times 10^{-4}}$	$\mathbf{9.993 \times 10^{-4}}$
07	$\mathbf{4.270 \times 10^{-2}}$	$\mathbf{4.402 \times 10^{-5}}$	$\mathbf{1.591 \times 10^{-3}}$	$\mathbf{6.009 \times 10^{-3}}$
08	$\mathbf{3.302 \times 10^{-2}}$	$\mathbf{1.856 \times 10^{-4}}$	$\mathbf{1.943 \times 10^{-3}}$	$\mathbf{2.335 \times 10^{-3}}$
09	5.271×10^{-2}	$\mathbf{3.184 \times 10^{-4}}$	$\mathbf{3.403 \times 10^{-3}}$	$\mathbf{4.586 \times 10^{-3}}$
10	$\mathbf{4.220 \times 10^{-2}}$	$\mathbf{5.928 \times 10^{-4}}$	$\mathbf{3.254 \times 10^{-3}}$	$\mathbf{8.641 \times 10^{-3}}$
11	$\mathbf{1.529 \times 10^{-2}}$	$\mathbf{8.031 \times 10^{-4}}$	$\mathbf{2.338 \times 10^{-3}}$	$\mathbf{1.405 \times 10^{-2}}$
12	$\mathbf{1.933 \times 10^{-2}}$	$\mathbf{5.407 \times 10^{-4}}$	$\mathbf{4.477 \times 10^{-3}}$	$\mathbf{2.374 \times 10^{-2}}$
13	$\mathbf{1.055 \times 10^{-2}}$	$\mathbf{2.347 \times 10^{-4}}$	$\mathbf{6.542 \times 10^{-3}}$	$\mathbf{2.128 \times 10^{-2}}$

Tabela 3.6: *P-values* resultantes do teste de causalidade de Granger aplicado entre os comentários e tópicos positivos e muito positivos com o preço do *Bitcoin*.

3.5 Treinamento

Por este trabalho abordar um problema envolvendo séries temporais, manter a ordem temporal entre os conjuntos de treinamento e de teste é fundamental para simular os dados reais. Sendo assim, o conjunto de treinamento é formado pelos primeiros 75% dos dados, enquanto o restante representa o conjunto de teste.

4

Experimentos

O foco deste capítulo é detalhar a configuração dos experimentos realizados e analisar os resultados obtidos. Durante as análises de resultados feitas neste capítulo, será mostrada apenas uma amostra dos resultados obtidos no experimento em questão. Os resultados completos podem ser encontrados no Apêndice B.

A medida em que os experimentos foram executados, possíveis melhorias foram encontradas. Enquanto algumas formaram novas hipóteses a serem testadas, outras são apenas descritas na Seção 5.1, de trabalhos futuros. Nas subseções a seguir, são detalhados os resultados iniciais e hipóteses que visam melhorar o desempenho do sistema proposto neste trabalho.

4.1 Avaliação dos experimentos

Para avaliar a performance dos modelos utilizados nos experimentos deste capítulo, se faz necessário a utilização das métricas de acurácia, precisão e cobertura, onde as duas últimas são calculadas de diferentes formas. Adicionalmente, para alguns casos, a matriz de confusão resultante será analisada.

A acurácia é calculada pelo total de instâncias classificadas corretamente dividido pelo total de elementos no conjunto de teste. Desta forma, tal métrica é a mais útil quando se deseja ter uma visão geral do modelo avaliado. As métricas de precisão e cobertura ajudam a melhor entender o funcionamento do modelo em questão em relação ao problema base deste trabalho.

Tanto a precisão como a cobertura - *precision* e *recall*, respectivamente - foram calculadas de três formas diferentes. A primeira dela, chamada de *macro*, oferece uma visão global da performance e é calculada a partir da média aritmética das outras duas, descritas nas equações 4.1, 4.2, 4.3 e 4.4, tal que:

$$Precision_{positive} = \frac{TP}{TP+FP} \tag{4.1}$$

$$Precision_{negative} = \frac{TN}{TN+FN} \tag{4.2}$$

$$Recall_{positive} = \frac{TP}{TP+FN} \tag{4.3}$$

$$Recall_{negative} = \frac{TN}{TN+FP} \tag{4.4}$$

onde TP, TN são - respectivamente - os valores verdadeiramente positivos e verdadeiramente negativos e os termos FP e FN representam, nessa ordem, os falsos positivos e os falsos negativos. Nas tabelas de resultados que serão apresentadas, as colunas correspondentes à precisão e cobertura referem-se às métricas *macro*.

4.2 Hipótese inicial

Por se tratar de um problema de classificação, um dos algoritmos de aprendizagem mais simples possíveis é o *Naive Bayes*. Inspirado por KIM et al. (2016) - para gerar os primeiros resultados - foi utilizado o AODE, um modelo mais robusto do que o *Naive Bayes* principalmente pela suposição de independência discutida na Seção 2.5.3.

Dado que o processo de desenvolvimento realizado por ambos os trabalhos são parecidos, espera-se que os resultados também sejam. A principal diferença, entretanto, é o tamanho do conjunto de dados. Após o tratamento dos dados faltantes, detalhado na Subseção 3.3.2, o conjunto total de dados cresce consideravelmente, visto que os dados disponibilizados - por KIM et al. (2016) - datam desde Novembro de 2009.

Para o presente experimento, dois classificadores AODE, detalhado na seção 2.5.3, foram utilizados. Enquanto um se encontra no modo *paper*, o outro estimou suas probabilidades no modo *count*. Ambos os classificadores utilizaram $m = 1$ para o filtro de características, presente na equação 2.10.

Outro fator importante é que, por se tratar de um algoritmo determinístico, o experimento foi executado apenas uma vez para cada tamanho de janela deslizante considerado. Para este experimento, o tamanho da janela utilizado variou entre 1 e 45.

4.2.1 Análise dos Resultados

A acurácia máxima obtida foi de 54,34% para o classificador modo *paper* utilizando uma janela de 4 dias. O único outro resultado relevante - ou seja, minimamente diferente de 50% - foi para janela de tamanho 3, onde o classificador estimando as probabilidades no modo *paper* obteve 53.35%.

Como comentado, para os outros resultados deste experimento - presentes parcialmente na Tabela 4.1 e completos na Tabela B.1 - a performance é similar a escolher aleatóriamente qualquer uma das duas classes possíveis. A razão mais provável para tais resultados é a quantidade de ruído adicionado ao tratar os dados faltantes na Subseção 3.3.2. Esta hipótese é testada na subseção seguinte.

	AODE					
Janela	modo *paper*			modo *count*		
	Precisão	Cobertura	Acurácia	Precisão	Cobertura	Acurácia
01	0,4963	0,4963	0,4960	0,4997	0,4997	0,4980
02	0,4851	0,4852	0,4861	0,4751	0,4770	0,4743
03	0,5329	0,5316	0,5335	0,5081	0,5072	0,5039
04	**0,5429**	**0,5425**	**0,5434**	0,4932	0,4946	0,4901
05	0,4905	0,4905	0,4910	0,5149	0,5113	0,5069
06	0,5035	0,5035	0,5029	0,4862	0,4895	0,4851
07	0,4838	0,4839	0,4831	0,4861	0,4896	0,4851
08	0,4712	0,4712	0,4712	0,4859	0,4897	0,4851
09	0,4747	0,4750	0,4742	0,4919	0,4948	0,4900
10	0,5001	0,5001	0,5000	0,4757	0,4850	0,4801

Tabela 4.1: Tabela com resultados do experimento correspondente à primeira Hipótese do projeto. Contém os resultado dos classificadores AODE utilizando janelas deslizantes que variam entre 1 e 10.

4.3 Hipótese de Melhoria - Reduzindo o Conjunto de Dados

Com o intuito de desconsiderar períodos onde poucos dados foram coletados, os primeiro 1250 pontos de dados - contendo 99% dos dados faltantes - foram desconsiderados. Dessa forma o novo conjunto de dados, utilizado em todos os experimentos em seguida, datam a partir de Dezembro de 2013. Este experimento contém a mesma configuração do anterior, fazendo com que a única diferença seja o corte de dados explicado no parágrafo anterior.

4.3.1 Análise dos Resultados

Na Tabela 4.2, é possível encontrar parte dos resultados obtidos pela execução do experimento corrente. Os resultados completos se fazem presentes na Tabela B.2. A partir deste experimento, fica evidente que quando dados faltam em demasia a performance do classificador é prejudicada. O classificador que obteve a melhor performance neste experimento foi o AODE no modo *count* com os dados divididos em janelas de 2 dias. Apesar de existirem configurações que entregam maior precisão macro, este modelo apresenta melhor *trade-off*[1] entre precisão e recall macro. Este fator pode ser confirmado através da análise de outras configurações.

O modelo AODE no modo *count*, utilizando uma janela de 28 dias - ilustrada no Apêndice B - apresenta precisão macro mais alta, porém, sua cobertura macro é de apenas 47%. Isso significa que esta configuração classifica a grande maioria das instâncias de teste como sendo de uma classe, enquanto a outra classe é raramente escolhida. Entretanto, quando escolhida, é o rótulo correto. Afirmações que podem ser confirmadas pelas métricas $Precision_{positive}$, $Precision_{negative}$, $Recall_{positive}$ e $Recall_{negative}$, respectivamente valoradas com: 0.4945, 1, 1 e

[1] Expressão em inglês que significa escolher algo em detrimento de outra.

	AODE					
Janela	modo *paper*			modo *count*		
	Precisão	Cobertura	Acurácia	Precisão	Cobertura	Acurácia
01	0,5482	0,5450	0,5463	0,5323	0,5316	0,5309
02	0,4903	0,4912	0,4896	**0,5979**	**0,5891**	**0,5876**
03	0,4984	0,4991	0,4974	0,5981	0,5763	0,5751
04	0,5131	0,5046	0,5025	0,5537	0,5401	0,5388
05	0,4695	0,4945	0,4922	0,5506	0,5351	0,5336
06	0,5638	0,5101	0,5077	0,5439	0,5299	0,5284
07	0,4765	0,4940	0,4895	0,5576	0,5393	0,5364
08	0,4602	0,4944	0,4895	0,5608	0,5346	0,5312
09	0,5623	0,5396	0,5364	0,5723	0,5349	0,5312
10	0,5754	0,5449	0,5416	0,5723	0,5349	0,5312

Tabela 4.2: Tabela com resultados do experimento correspondente à primeira Hipótese de Melhoria. Contém o resultado dos classificadores AODE utilizando janelas deslizantes que variam entre 1 e 10.

0.0312.

Por se tratar de um problema de classificação binária, caso a saída do modelo seja invertida - ou seja, o que é positivo vira negativo e vice-versa - as métricas de acurácia, precisão e cobertura macro se transformam em seu complemento para 1. Dessa forma, a configuração que apresenta maior precisão são os modelos no modo *paper* com janelas entre 24 e 26 dias - informações presentes na Tabela B.2. Entretanto, tal inversão seria prejudicial à métrica de acurácia.

4.4 Hipótese de Melhoria - Outros modelos

Neste experimento são utilizados os classificadores SVM, *Random Forest* e MLP. Estes classificadores foram escolhidos devido a suas diferenças, o que permite - a partir dos resultados - direcionar hipóteses futuras de melhoria para priorizar determinadas características.

O SVM é um classificador discriminativo que pode se beneficiar de um *kernel* quadrático, pois este permite a criação de um hiperplano separador curvo. Já o *Random Forest* é um algoritmo de *ensemble*, o que permite a combinação eficiente de classificadores fracos. Por fim, o MLP que faz parte de um grupo de algoritmos conhecido por sua capacidade de generalização, chamado de redes neurais.

4.4.1 Configuração do Experimento

O *Random Forest*[2] foi testado utilizando duas quantidades de estimadores: 200 e 500. A altura máxima de cada árvore ficasse definida em 30, mantendo o tempo de treinamento baixo.

[2]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Por sua vez, o MLP foi executado com a configuração padrão[3] do *scikit-learn*: Apenas uma camada escondida com 100 neurônios e com taxa de aprendizagem constante em $0,01$.

Para o SVM, a configuração padrão[4] do *scikit-learn* foi utilizada como base. A partir dela, algumas combinações variando apenas os parâmetros *C* e *gamma* - descritos na seção 2.5.4 - foram testadas. Experimentalmente, os valores $C = 10^5$ e $gamma = 10^{-5}$ resultaram no melhor desempenho. Cada configuração neste experimento rodou um total de 15 vezes. Os resultados mostram as métricas da execução que obteve maior acurácia.

4.4.2 Análise dos Resultados

Para o presente experimento os resultados foram divididos em dois grupos. O primeiro grupo contém os resultados parciais e na íntegra do SVM e da MLP, respectivamente nas Tabelas 4.3 e B.3. O segundo grupo é formado pelos resultados das duas configurações testadas para o *Random Forest*, seus resultados parciais e completos estão - nesta ordem - nas Tabelas 4.4 e B.4.

	SVM			MLP		
Janela	Precisão	Cobertura	Acurácia	Precisão	Cobertura	Acurácia
19	**0,5852**	**0,5800**	**0,5767**	0,5687	0,5677	0,5661
20	0,5627	0,5588	0,5555	**0,6135**	**0,6136**	**0,6137**
21	0,5446	0,5425	0,5396	0,5810	0,5789	0,5767
22	0,5710	0,5686	0,5661	0,6029	0,6030	0,6031
23	0,5670	0,5588	0,5531	0,5849	0,5825	0,5797
24	0,5451	0,5382	0,5319	0,6069	0,6036	0,6063
33	0,5610	0,5603	0,5591	0,6124	0,6049	0,6021
34	0,5541	0,5541	0,5537	0,5587	0,5587	0,5591
35	0,5348	0,5348	0,5351	0,5731	0,5731	0,5729
36	0,4977	0,4977	0,4972	**0,6114**	**0,6112**	**0,6108**

Tabela 4.3: Tabela com resultados do experimento correspondente à segunda Hipótese de Melhoria. Contém o resultado dos classificadores SVM e MLP utilizando janelas deslizantes que variam de 19 a 24 e 33 a 36.

Para o SVM, sua acurácia atinge a máxima de 57.67% quando o tamanho da janela é de 19 dias. A partir da matriz de confusão para esta configuração - ilustrada na figura 4.1 - percebe-se que o **não** aumento no preço é detectado com maior precisão. 61,9% para esta classe contra 55% para um aumento no preço, dados que podem ser calculados usando as Equações 4.1 e 4.2.

Analisando os resultados da MLP, percebe-se que o melhor tamanho de janela é de 20 dias. Entretanto, janelas próximas - presentes na Tabela 4.3 - atingem performance parecida. Através da análise da matriz de confusão (Figura 4.2) da melhor janela, é perceptível que este

[3]https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
[4]https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

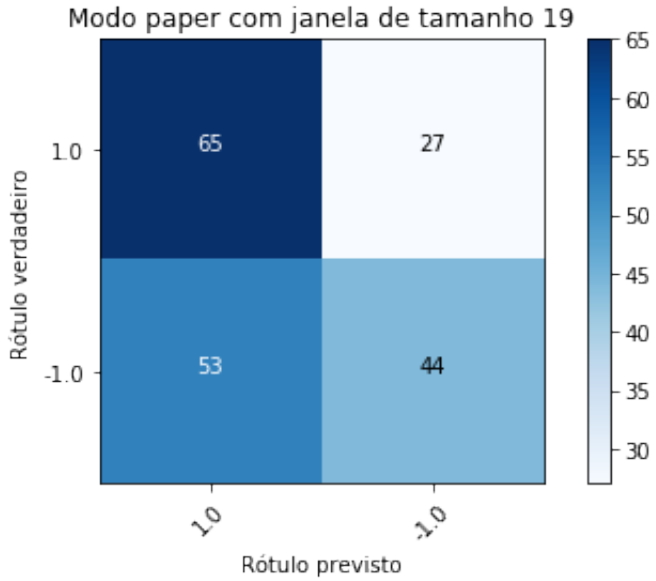

Figura 4.1: Matriz de confusão do SVM utilizando uma janela deslizante de tamanho 19.

modelo consegue - diferentemente do anterior - prever ambas as classes de forma consistente, atingindo uma precisão de 60,2% para um aumento no preço e de 62,5% para a outra classe.

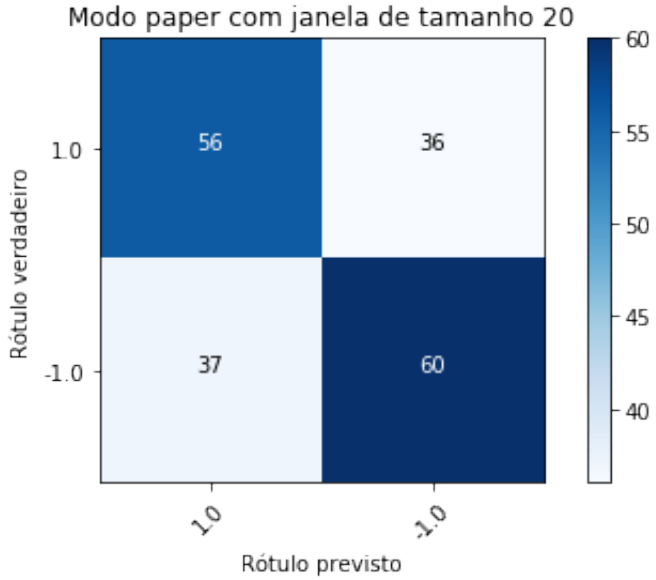

Figura 4.2: Matriz de confusão da MLP utilizando uma janela deslizante de tamanho 20.

O modelo - dentre todos testados neste experimento - que apresentou a melhor acurácia foi o *Random Forest* com 200 árvores de decisão utilizando uma janela de 24 dias. A razão pela qual o aumento na quantidade de árvores neste modelo diminui a acurácia final pode ser atribuída ao aumento da correlação entre as árvores. Estas que devem ser altamente independentes para que o *ensemble* funcione corretamente.

Janela	Random Forest					
	200 Árvores			500 Árvores		
	Precisão	Cobertura	Acurácia	Precisão	Cobertura	Acurácia
06	0,5803	0,5803	0,5803	0,5754	0,5749	0,5751
07	0,5431	0,5423	0,5416	0,5781	0,5778	0,5781
08	0,5702	0,5665	0,5677	0,5310	0,5306	0,5312
09	0,5827	0,5767	0,5781	**0,5833**	**0,5829**	**0,5833**
10	0,5535	0,5509	0,5520	0,5670	0,5609	0,5625
23	0,5600	0,5531	0,5585	0,5793	0,5771	0,5797
24	**0,6254**	**0,6187**	**0,6223**	0,5742	0,5713	0,5744
25	0,5686	0,5661	0,5691	0,5704	0,5702	0,5691
26	0,5773	0,5764	0,5744	0,5736	0,5723	0,5744
27	0,5880	0,5880	0,5882	0,5661	0,5658	0,5668

Tabela 4.4: Tabela com resultados do experimento correspondente à segunda Hipótese de Melhoria. Contém o resultado dos classificadores *Random Forest* - com 200 e 500 árvores de decisão - utilizando janelas deslizantes que variam 06 a 10 e 23 a 27.

A partir dos melhores resultados dentre as 15 execuções de cada modelo, é possível afirmar que o aumento da complexidade do modelo aumenta a acurácia em relação ao problema. Entretanto, ao analisar a média das métricas obtidas nas 15 execuções de cada modelo, é possível perceber que existe um problema intrínseco nos modelos aqui testados: a variância, trazendo a acurácia média para a casa dos 50%. Este problema apenas não se faz presente no SVM, pois este é um modelo determinístico.

4.5 Hipótese de Melhoria - Complexidade dinâmica

Uma das principais características do modelo *Deep Forest* é sua capacidade de determinar a complexidade necessária para o problema em questão de forma dinâmica e dependente dos dados. Dessa forma, espera-se que o modelo tenha complexidade suficiente para superar a performance dos experimentos anteriores, porém, não tanto a ponto de prejudicar a sua própria. A partir dos resultados obtidos no experimento anterior, também é esperado que os resultados desta hipótese de melhoria demonstrem uma redução na variância entre as execuções.

4.5.1 Configuração do experimento

Em ZHOU; FENG (2017), é dito que o *Deep Forest* em sua configuração padrão pode ser utilizado para várias tarefas diferentes. É com esta configuração que todos os resultados do artigo citado são calculados.

Em detalhes, na configuração padrão, cada camada contém quatro *Random Forests* normais e outras quatro *Random Forests* verdadeiramente aleatórias. Para cada floresta aleatória são utilizadas 500 árvores de decisão. Entretanto, para este trabalho, será utilizado um total de 200 árvores de decisão, pois esta quantidade foi a que obteve melhor performance no experimento anterior.

4.5.2 Análise dos Resultados

Os principais resultados deste experimento estão indicados na Tabela 4.5. A melhor acurácia obtida pelo *Deep Forest* foi de 57,36% para a janela de tamanho 16. Aparentemente, as múltiplas *Random Forest* presentes neste modelo tiveram performances contrastantes, a ponto de nivelar a acurácia final abaixo das obtidas no experimento anterior.

Entretanto, o modelo em questão pode ser considerado mais consistente do que o *Random Forest* com 200 árvores de decisão, estudado no experimento anterior. Analisando a acurácia média do melhor tamanho de janela para ambos os modelos, é possível perceber que o *Deep Forest* - que atingiu 55,96% - têm menor variância dentre as 15 execuções. O *Random Forest*, por sua vez, atingiu a acurácia média de 52,01%.

Janela	Deep Forest		
	Precisão	Cobertura	Acurácia
15	0,5685	0,5524	0,5473
16	**0,5765**	**0,5751**	**0,5736**
17	0,5288	0,5249	0,5210
18	0,5033	0,5032	0,5052
19	0,4717	0,4731	0,4761
34	0,4710	0,4746	0,4784
35	0,5449	0,5371	0,5405
36	0,5576	0,5550	0,5567
37	0,4823	0,4839	0,4864
38	0,4936	0,4944	0,4972

Tabela 4.5: Tabela com resultados do experimento correspondente à hipótese de complexidade dinâmica. Contém o resultado dos classificadores *Random Forest* - com 200 árvores de decisão - utilizando janelas deslizantes que variam 06 a 10 e 23 a 27.

4.6 Modificando o filtro

Por este trabalho abordar a previsão da flutuação do preço do *Bitcoin* como um problema de classificação, existe a possibilidade de que as informações apontadas pelo teste de causalidade de Granger, na Seção 3.4, como mais relevantes não tenham a mesma importância para o problema de classificação.

Sendo assim, é válido realizar uma busca exaustiva entre todas as combinações possíveis das séries temporais de entrada. Pelo tempo de execução necessário para esta técnica, apenas as combinações contendo entre 1 e 4 informações foram testadas. Para isso, foi utilizado o melhor classificador encontrado pelos experimentos anteriores: o *Random Forest* com 200 árvores de decisão. Cada combinação foi executada para tamanhos de janela entre 1 e 10 dias. Para cada janela, o modelo escolhido foi executado 10 vezes.

O melhor filtro encontrado, dentre os testados, foi utilizando as informações referentes ao total de respostas e visualizações combinadas com o total de comentários positivos e tópicos muito negativos. Durante a busca exaustiva, a melhor acurácia obtida pelo *Random Forest* foi de 63.91% utilizando janela com 1 dia de tamanho.

5

Conclusão

Compartilhar opiniões dos diferentes aspectos do dia-a-dia tem se tornado cada vez mais comum. Para isso, pessoas utilizam plataformas como o *Twitter* e/ou fóruns especializados. As opiniões das pessoas podem ser coletadas e tratadas como uma fonte para extração de sentimentos, PAK; PAROUBEK (2010). Para este trabalho, foi utilizada uma base de dados formada pelas postagens feitas no fórum *bitcointalk*.

Bases de dados formadas por documentos de texto são utilizadas em diversas aplicações de aprendizagem de máquina, como por exemplo na construção de modelos de ranqueamento, geração de resposta, extração de entidades nomeadas, entre outras. A partir desta base de dados é possível extrair informações que facilitem a resolução de um determinado problema. Para a construção de modelos de ranqueamento, por exemplo, informações como a frequência mínima e máxima dos termos podem ser utilizados como características do modelo em questão[1]. Quando estas bases são utilizadas para previsão de séries temporais, normalmente, o sentimento/humor associado aos textos são extraídos, BOLLEN; MAO; ZENG (2011).

Dado que o sentimento dos usuários de criptomoedas é tido como direcionador do preço destas moedas - KRISTOUFEK (2015) - neste trabalho, a base de documentos é transformada numa base de opiniões, formada pela polaridade do sentimento associado às postagens do *bitcointalk*. Em seguida, o teste de causalidade de Granger foi executado com o objetivo de utilizar como entrada para o modelo de aprendizagem apenas informações relevantes para previsão das flutuações.

Através dos resultados obtidos neste trabalho, é seguro afirmar que a polaridade do sentimento dos usuários de uma criptomoeda pode ser utilizada para classificar as flutuações no preço. É necessário lembrar que as informações que serão relevantes para cada moeda virtual pode variar, sendo assim, é importante que toda a metodologia deste projeto seja replicada.

Para aplicar a metodologia apresentada neste trabalho, é necessário considerar que a busca exaustiva - sobre quais informações devem ser utilizadas na entrada dos modelos de aprendizagem - se mostrou mais efetiva do que o teste de causalidade de Granger para encontrar as séries mais expressivas. Entretanto, a velocidade de execução da busca exaustiva é consideravelmente maior

[1] https://www.microsoft.com/en-us/research/project/mslr/

do que o teste estatístico. Isso, combinado com os resultados obtidos, mostrou que o teste de Granger é o mais efetivo para a fase de filtragem.

5.1 Trabalhos futuros

Além das hipóteses de melhoria exploradas no Capítulo 4 existem modificações que podem ser aplicadas em partes específicas do sistema proposto com o objetivo de explorar novos - e possivelmente melhores - resultados.

A primeira possibilidade está na etapa de pré-processamento, mais especificamente nas transformações aplicadas nos dados. Alguns dos classificadores utilizados, como por exemplo o AODE e o *Random Forest*, associam elementos de acordo com suas características. Sendo assim, discretizar as séries temporais de entrada pode ajudar na performance dos modelos em questão. De forma específica, a discretização destas informações aumenta a contagem das características, permitindo estimitivas de probabilidades mais precisas para o AODE. Para o *Random Forest*, o impacto acontecerá na entropia dos atributos.

Outra abordagem para tratar os dados de entrada é utilizar as palavras do texto como as características do modelo. Em problemas de classificação textual e recuperação de informação, os valores destes atributos podem ser preenchidos de diversas maneiras, como por exemplo com a frequência dos termos ou com um valor binário indicando a presença ou não da palavra. Entretanto, assim como em COLIANNI; ROSALES; SIGNOROTTI (2015), estas características podem ser preenchidas com um valor representando a polaridade do sentimento desta palavra. Para que isso seja possível, é necessário que outro analisador de sentimento seja utilizado, como por exemplo o *SentiWordNet*[2].

Mudando a perspectiva, os classificadores que apresentaram os melhores resultados nos experimentos executados podem ser combinados utilizando técnicas de *ensemble*. Para isso, podem ser utilizadas técnicas como *voting* e *stacking*, assim como em QIAN; RASHEED (2007).

Adicionalmente, outras informações podem substituir - ou agregar - os dados de entrada. Emoções, quando comparados com a polaridade do sentimento, dão um entendimento mais profundo da opinião das pessoas. Esse tipo de modificação pode ser benéfica para a performance do sistema proposto. Além disso, o próprio *Bitcoin* contém diversas outras informações que podem ser utilizadas para classificação da flutuação, como: quantidade média de transações por bloco, lucro dos mineradores, custo por transação, entre outras.

[2] sentiwordnet.isti.cnr.it

Referências

BOLLEN, J.; MAO, H.; ZENG, X. Twitter mood predicts the stock market. **Journal of Computational Science**, [S.l.], v.2, n.1, p.1 – 8, 2011.

BROCKWELL, P. J.; DAVIS, R. A. **Introduction to Time Series and Forecasting**. 2nd.ed. [S.l.]: Springer, 2002.

CHENG, C. et al. Time series forecasting for nonlinear and non-stationary processes: a review and comparative study. **IIE Transactions**, [S.l.], v.47, n.10, p.1053–1071, 2015.

COLIANNI, S.; ROSALES, S. M.; SIGNOROTTI, M. Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis. In: **Anais...** [S.l.: s.n.], 2015.

GRANGER, C. Testing for causality: a personal viewpoint. **Journal of Economic Dynamics and Control**, [S.l.], v.2, p.329 – 352, 1980.

GRANGER, C. W.; HUANGB, B.-N.; YANG, C.-W. A bivariate causality between stock prices and exchange rates: evidence from recent asianflu. **The Quarterly Review of Economics and Finance**, [S.l.], v.40, n.3, p.337 – 354, 2000.

HUNTER, J. D. Matplotlib: a 2d graphics environment. **Computing in Science Engineering**, [S.l.], v.9, n.3, p.90–95, May 2007.

HUTTO, C. J.; GILBERT, E. VADER: a parsimonious rule-based model for sentiment analysis of social media text. In: ICWSM. **Anais...** [S.l.: s.n.], 2014.

JONES, E. et al. **SciPy**: open source scientific tools for Python. [Online; Acessado 6 de Dezembro de 2018].

KIM, Y. B. et al. Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies. **PLOS ONE**, [S.l.], v.11, n.8, p.1–17, 08 2016.

KRISTOUFEK, L. What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis. **PLOS ONE**, [S.l.], v.10, n.4, p.1–15, 04 2015.

KWIATKOWSKI, D. et al. Testing the null hypothesis of stationarity against the alternative of a unit root: how sure are we that economic time series have a unit root? **Journal of Econometrics**, [S.l.], v.54, n.1, p.159 – 178, 1992.

MALKIEL, B. G. The Efficient Market Hypothesis and Its Critics. **Journal of Economic Perspectives**, [S.l.], v.17, n.1, p.59–82, March 2003.

MCKINNEY, W. Data Structures for Statistical Computing in Python. In: PYTHON IN SCIENCE CONFERENCE, 9. **Proceedings...** [S.l.: s.n.], 2010. p.51 – 56.

NAKAMOTO, S. **Bitcoin**: a peer-to-peer electronic cash system. 2009.

PAK, A.; PAROUBEK, P. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In: SEVENTH CONFERENCE ON INTERNATIONAL LANGUAGE RESOURCES AND EVALUATION (LREC'10). **Proceedings...** European Languages Resources Association (ELRA), 2010.

PEDREGOSA, F. et al. Scikit-learn: machine learning in Python. **Journal of Machine Learning Research**, [S.l.], v.12, p.2825–2830, 2011.

PHILLIPS, R. C.; GORSE, D. Cryptocurrency price drivers: wavelet coherence analysis revisited. **PLOS ONE**, [S.l.], v.13, n.4, p.1–21, 04 2018.

QIAN, B.; RASHEED, K. Stock market prediction with multiple classifiers. **Applied Intelligence**, [S.l.], v.26, n.1, p.25–33, Feb 2007.

WEBB, G. I.; BOUGHTON, J. R.; WANG, Z. Not So Naive Bayes: aggregating one-dependence estimators. **Machine Learning**, [S.l.], v.58, n.1, p.5–24, Jan 2005.

ZHOU, Z.-H.; FENG, J. Deep Forest. In: EUROPEAN LANGUAGES RESOURCES ASSOCIATION (ELRA). **Anais...** [S.l.: s.n.], 2017.

Apêndice

A

Metodologia

Aqui serão apresentados, na íntegra, todos os resultados obtidos nos testes de causalidade de Granger executados no Capítulo 3.

Time lag	Muito positivos		Positivos	
	Respostas	Tópicos	Respostas	Tópicos
01	3.817×10^{-1}	8.018×10^{-1}	$\mathbf{8.915 \times 10^{-3}}$	$\mathbf{4.852 \times 10^{-2}}$
02	4.416×10^{-1}	6.042×10^{-1}	$\mathbf{3.122 \times 10^{-2}}$	$\mathbf{4.322 \times 10^{-2}}$
03	$\mathbf{2.201 \times 10^{-3}}$	$\mathbf{1.825 \times 10^{-3}}$	$\mathbf{9.802 \times 10^{-6}}$	$\mathbf{2.131 \times 10^{-4}}$
04	$\mathbf{1.226 \times 10^{-3}}$	$\mathbf{2.106 \times 10^{-6}}$	$\mathbf{3.430 \times 10^{-6}}$	$\mathbf{9.414 \times 10^{-5}}$
05	$\mathbf{2.686 \times 10^{-3}}$	$\mathbf{5.947 \times 10^{-6}}$	$\mathbf{1.606 \times 10^{-5}}$	$\mathbf{3.686 \times 10^{-4}}$
06	$\mathbf{3.383 \times 10^{-2}}$	$\mathbf{9.171 \times 10^{-5}}$	$\mathbf{3.809 \times 10^{-4}}$	$\mathbf{9.993 \times 10^{-4}}$
07	$\mathbf{4.270 \times 10^{-2}}$	$\mathbf{4.402 \times 10^{-5}}$	$\mathbf{1.591 \times 10^{-3}}$	$\mathbf{6.009 \times 10^{-3}}$
08	$\mathbf{3.302 \times 10^{-2}}$	$\mathbf{1.856 \times 10^{-4}}$	$\mathbf{1.943 \times 10^{-3}}$	$\mathbf{2.335 \times 10^{-3}}$
09	5.271×10^{-2}	$\mathbf{3.184 \times 10^{-4}}$	$\mathbf{3.403 \times 10^{-3}}$	$\mathbf{4.586 \times 10^{-3}}$
10	$\mathbf{4.220 \times 10^{-2}}$	$\mathbf{5.928 \times 10^{-4}}$	$\mathbf{3.254 \times 10^{-3}}$	$\mathbf{8.641 \times 10^{-3}}$
11	$\mathbf{1.529 \times 10^{-2}}$	$\mathbf{8.031 \times 10^{-4}}$	$\mathbf{2.338 \times 10^{-3}}$	$\mathbf{1.405 \times 10^{-2}}$
12	$\mathbf{1.933 \times 10^{-2}}$	$\mathbf{5.407 \times 10^{-4}}$	$\mathbf{4.477 \times 10^{-3}}$	$\mathbf{2.374 \times 10^{-2}}$
13	$\mathbf{1.055 \times 10^{-2}}$	$\mathbf{2.347 \times 10^{-4}}$	$\mathbf{6.542 \times 10^{-3}}$	$\mathbf{2.128 \times 10^{-2}}$
14	$\mathbf{9.320 \times 10^{-3}}$	$\mathbf{1.925 \times 10^{-4}}$	$\mathbf{4.418 \times 10^{-3}}$	$\mathbf{2.341 \times 10^{-2}}$
15	$\mathbf{1.383 \times 10^{-2}}$	$\mathbf{1.941 \times 10^{-4}}$	$\mathbf{7.203 \times 10^{-3}}$	$\mathbf{9.987 \times 10^{-3}}$
16	$\mathbf{2.627 \times 10^{-2}}$	$\mathbf{5.833 \times 10^{-4}}$	$\mathbf{5.367 \times 10^{-3}}$	$\mathbf{2.739 \times 10^{-2}}$
17	$\mathbf{2.951 \times 10^{-2}}$	$\mathbf{6.508 \times 10^{-4}}$	$\mathbf{4.511 \times 10^{-3}}$	$\mathbf{1.912 \times 10^{-2}}$
18	1.335×10^{-1}	$\mathbf{2.312 \times 10^{-3}}$	$\mathbf{1.873 \times 10^{-2}}$	$\mathbf{4.552 \times 10^{-2}}$
19	$\mathbf{3.825 \times 10^{-2}}$	$\mathbf{2.505 \times 10^{-3}}$	$\mathbf{6.789 \times 10^{-3}}$	$\mathbf{1.263 \times 10^{-2}}$
20	6.471×10^{-2}	$\mathbf{4.853 \times 10^{-4}}$	$\mathbf{8.525 \times 10^{-3}}$	$\mathbf{8.881 \times 10^{-3}}$
21	$\mathbf{1.215 \times 10^{-2}}$	$\mathbf{1.176 \times 10^{-3}}$	$\mathbf{3.492 \times 10^{-3}}$	$\mathbf{1.011 \times 10^{-2}}$
22	$\mathbf{1.330 \times 10^{-2}}$	$\mathbf{2.338 \times 10^{-3}}$	$\mathbf{4.670 \times 10^{-3}}$	$\mathbf{7.170 \times 10^{-3}}$
23	$\mathbf{1.032 \times 10^{-2}}$	$\mathbf{5.865 \times 10^{-3}}$	$\mathbf{2.726 \times 10^{-3}}$	$\mathbf{5.218 \times 10^{-3}}$
24	$\mathbf{4.087 \times 10^{-3}}$	$\mathbf{5.017 \times 10^{-3}}$	$\mathbf{2.412 \times 10^{-3}}$	$\mathbf{7.679 \times 10^{-3}}$
25	$\mathbf{3.997 \times 10^{-3}}$	$\mathbf{5.645 \times 10^{-3}}$	$\mathbf{2.011 \times 10^{-3}}$	$\mathbf{1.429 \times 10^{-2}}$
26	$\mathbf{2.729 \times 10^{-3}}$	$\mathbf{2.043 \times 10^{-3}}$	$\mathbf{6.260 \times 10^{-4}}$	$\mathbf{2.278 \times 10^{-2}}$
27	$\mathbf{2.192 \times 10^{-3}}$	$\mathbf{4.671 \times 10^{-3}}$	$\mathbf{5.067 \times 10^{-4}}$	$\mathbf{2.833 \times 10^{-2}}$
28	$\mathbf{3.314 \times 10^{-3}}$	$\mathbf{9.389 \times 10^{-3}}$	$\mathbf{1.017 \times 10^{-3}}$	$\mathbf{1.849 \times 10^{-2}}$
29	$\mathbf{3.765 \times 10^{-3}}$	$\mathbf{1.536 \times 10^{-2}}$	$\mathbf{1.215 \times 10^{-3}}$	$\mathbf{2.084 \times 10^{-2}}$
30	$\mathbf{4.713 \times 10^{-3}}$	$\mathbf{3.499 \times 10^{-2}}$	$\mathbf{3.507 \times 10^{-3}}$	$\mathbf{1.840 \times 10^{-2}}$
31	$\mathbf{8.304 \times 10^{-3}}$	$\mathbf{2.920 \times 10^{-2}}$	$\mathbf{5.000 \times 10^{-3}}$	$\mathbf{1.067 \times 10^{-2}}$
32	$\mathbf{6.487 \times 10^{-3}}$	$\mathbf{2.494 \times 10^{-2}}$	$\mathbf{5.528 \times 10^{-3}}$	$\mathbf{8.491 \times 10^{-3}}$
33	$\mathbf{6.057 \times 10^{-3}}$	$\mathbf{2.265 \times 10^{-2}}$	$\mathbf{1.154 \times 10^{-3}}$	$\mathbf{8.667 \times 10^{-3}}$
34	$\mathbf{6.678 \times 10^{-3}}$	$\mathbf{2.063 \times 10^{-2}}$	$\mathbf{2.832 \times 10^{-4}}$	$\mathbf{6.066 \times 10^{-3}}$
35	$\mathbf{4.514 \times 10^{-3}}$	$\mathbf{1.961 \times 10^{-3}}$	$\mathbf{3.837 \times 10^{-5}}$	$\mathbf{1.710 \times 10^{-3}}$
36	$\mathbf{4.287 \times 10^{-3}}$	$\mathbf{1.821 \times 10^{-3}}$	$\mathbf{4.219 \times 10^{-5}}$	$\mathbf{1.094 \times 10^{-3}}$
37	$\mathbf{4.934 \times 10^{-3}}$	$\mathbf{8.910 \times 10^{-4}}$	$\mathbf{3.028 \times 10^{-5}}$	$\mathbf{2.006 \times 10^{-3}}$
38	$\mathbf{5.685 \times 10^{-3}}$	$\mathbf{2.065 \times 10^{-4}}$	$\mathbf{1.898 \times 10^{-5}}$	$\mathbf{3.539 \times 10^{-3}}$
39	$\mathbf{5.981 \times 10^{-3}}$	$\mathbf{2.504 \times 10^{-4}}$	$\mathbf{2.628 \times 10^{-5}}$	$\mathbf{3.067 \times 10^{-3}}$
40	$\mathbf{7.253 \times 10^{-3}}$	$\mathbf{2.407 \times 10^{-4}}$	$\mathbf{2.240 \times 10^{-5}}$	$\mathbf{1.747 \times 10^{-3}}$
41	$\mathbf{9.170 \times 10^{-3}}$	$\mathbf{3.842 \times 10^{-4}}$	$\mathbf{2.470 \times 10^{-5}}$	$\mathbf{2.390 \times 10^{-3}}$
42	$\mathbf{1.763 \times 10^{-2}}$	$\mathbf{1.700 \times 10^{-3}}$	$\mathbf{1.216 \times 10^{-4}}$	$\mathbf{2.343 \times 10^{-3}}$
43	$\mathbf{9.302 \times 10^{-3}}$	$\mathbf{1.931 \times 10^{-3}}$	$\mathbf{1.129 \times 10^{-5}}$	$\mathbf{2.256 \times 10^{-3}}$
44	$\mathbf{1.209 \times 10^{-2}}$	$\mathbf{1.741 \times 10^{-3}}$	$\mathbf{1.963 \times 10^{-5}}$	$\mathbf{1.613 \times 10^{-3}}$
45	$\mathbf{1.331 \times 10^{-2}}$	$\mathbf{2.783 \times 10^{-3}}$	$\mathbf{2.225 \times 10^{-5}}$	$\mathbf{2.119 \times 10^{-3}}$

Tabela A.1: *P-value* resultante do teste de causalidade de Granger aplicado entre os comentários e tópicos positivos e muito positivos com o preço do *Bitcoin*.

Time lag	Muito negativos		Negativos		*Visualizações*
	Respostas	Tópicos	Respostas	Tópicos	
01	$\mathbf{2.204 \times 10^{-2}}$	5.632×10^{-1}	1.461×10^{-1}	3.238×10^{-1}	6.140×10^{-2}
02	9.619×10^{-2}	4.791×10^{-1}	$\mathbf{2.606 \times 10^{-2}}$	1.792×10^{-1}	$\mathbf{4.356 \times 10^{-2}}$
03	$\mathbf{3.288 \times 10^{-3}}$	1.761×10^{-1}	$\mathbf{6.660 \times 10^{-4}}$	3.053×10^{-1}	$\mathbf{3.330 \times 10^{-3}}$
04	$\mathbf{4.875 \times 10^{-3}}$	1.725×10^{-1}	$\mathbf{1.225 \times 10^{-3}}$	5.279×10^{-2}	$\mathbf{7.034 \times 10^{-4}}$
05	$\mathbf{1.159 \times 10^{-2}}$	2.578×10^{-1}	$\mathbf{3.275 \times 10^{-3}}$	6.085×10^{-2}	$\mathbf{1.823 \times 10^{-3}}$
06	$\mathbf{9.932 \times 10^{-3}}$	1.607×10^{-1}	$\mathbf{1.240 \times 10^{-2}}$	1.809×10^{-1}	$\mathbf{1.419 \times 10^{-2}}$
07	$\mathbf{2.009 \times 10^{-2}}$	2.005×10^{-1}	$\mathbf{2.510 \times 10^{-2}}$	3.050×10^{-1}	$\mathbf{2.614 \times 10^{-2}}$
08	$\mathbf{2.177 \times 10^{-2}}$	2.395×10^{-1}	$\mathbf{2.314 \times 10^{-2}}$	3.994×10^{-1}	$\mathbf{1.728 \times 10^{-2}}$
09	$\mathbf{3.539 \times 10^{-2}}$	1.140×10^{-1}	$\mathbf{1.994 \times 10^{-2}}$	4.254×10^{-1}	$\mathbf{1.836 \times 10^{-2}}$
10	$\mathbf{1.507 \times 10^{-2}}$	6.278×10^{-2}	$\mathbf{2.623 \times 10^{-2}}$	3.538×10^{-1}	$\mathbf{2.217 \times 10^{-2}}$
11	$\mathbf{1.402 \times 10^{-2}}$	9.129×10^{-2}	$\mathbf{2.468 \times 10^{-2}}$	4.154×10^{-1}	7.140×10^{-2}
12	$\mathbf{2.332 \times 10^{-2}}$	1.410×10^{-1}	$\mathbf{1.595 \times 10^{-2}}$	4.685×10^{-1}	8.988×10^{-2}
13	$\mathbf{2.111 \times 10^{-2}}$	1.825×10^{-1}	$\mathbf{1.217 \times 10^{-2}}$	5.357×10^{-1}	7.340×10^{-2}
14	$\mathbf{1.777 \times 10^{-2}}$	2.729×10^{-1}	$\mathbf{8.167 \times 10^{-3}}$	6.070×10^{-1}	5.434×10^{-2}
15	$\mathbf{2.184 \times 10^{-2}}$	2.591×10^{-1}	$\mathbf{1.298 \times 10^{-2}}$	5.519×10^{-1}	7.081×10^{-2}
16	$\mathbf{4.243 \times 10^{-2}}$	2.996×10^{-1}	$\mathbf{2.611 \times 10^{-2}}$	5.484×10^{-1}	1.435×10^{-1}
17	$\mathbf{3.115 \times 10^{-2}}$	4.409×10^{-1}	$\mathbf{3.433 \times 10^{-2}}$	6.502×10^{-1}	1.641×10^{-1}
18	5.803×10^{-2}	2.536×10^{-1}	9.586×10^{-2}	6.285×10^{-1}	2.400×10^{-1}
19	6.843×10^{-2}	1.121×10^{-1}	6.319×10^{-2}	4.981×10^{-1}	1.518×10^{-1}
20	1.015×10^{-1}	1.148×10^{-1}	$\mathbf{4.981 \times 10^{-2}}$	5.207×10^{-1}	1.896×10^{-1}
21	1.207×10^{-1}	1.648×10^{-1}	6.093×10^{-2}	4.338×10^{-1}	1.906×10^{-1}
22	1.758×10^{-1}	1.719×10^{-1}	5.769×10^{-2}	4.880×10^{-1}	2.225×10^{-1}
23	1.976×10^{-1}	2.056×10^{-1}	5.393×10^{-2}	4.649×10^{-1}	3.147×10^{-1}
24	2.117×10^{-1}	3.161×10^{-1}	6.033×10^{-2}	5.474×10^{-1}	3.163×10^{-1}
25	2.246×10^{-1}	4.459×10^{-1}	5.427×10^{-2}	6.090×10^{-1}	3.126×10^{-1}
26	1.287×10^{-1}	4.045×10^{-1}	$\mathbf{3.299 \times 10^{-2}}$	4.899×10^{-1}	3.448×10^{-1}
27	7.990×10^{-2}	2.441×10^{-1}	$\mathbf{2.210 \times 10^{-2}}$	5.866×10^{-1}	3.265×10^{-1}
28	5.772×10^{-2}	1.400×10^{-1}	$\mathbf{1.316 \times 10^{-2}}$	6.856×10^{-1}	2.826×10^{-1}
29	6.236×10^{-2}	1.547×10^{-1}	$\mathbf{1.675 \times 10^{-2}}$	5.978×10^{-1}	3.329×10^{-1}
30	5.428×10^{-2}	$\mathbf{4.552 \times 10^{-2}}$	$\mathbf{3.154 \times 10^{-2}}$	3.661×10^{-1}	3.962×10^{-1}
31	$\mathbf{4.863 \times 10^{-2}}$	9.078×10^{-2}	$\mathbf{3.865 \times 10^{-2}}$	3.232×10^{-1}	4.784×10^{-1}
32	6.255×10^{-2}	1.192×10^{-1}	$\mathbf{4.289 \times 10^{-2}}$	4.929×10^{-1}	4.666×10^{-1}
33	7.179×10^{-2}	2.157×10^{-1}	$\mathbf{4.617 \times 10^{-2}}$	3.812×10^{-1}	5.931×10^{-1}
34	7.716×10^{-2}	1.462×10^{-1}	$\mathbf{3.277 \times 10^{-2}}$	4.333×10^{-1}	5.993×10^{-1}
35	$\mathbf{2.665 \times 10^{-2}}$	1.545×10^{-1}	$\mathbf{6.552 \times 10^{-3}}$	3.358×10^{-1}	6.510×10^{-1}
36	$\mathbf{3.178 \times 10^{-2}}$	1.550×10^{-1}	$\mathbf{6.512 \times 10^{-3}}$	2.909×10^{-1}	7.051×10^{-1}
37	$\mathbf{2.873 \times 10^{-2}}$	$\mathbf{3.314 \times 10^{-2}}$	$\mathbf{4.407 \times 10^{-3}}$	4.122×10^{-1}	7.680×10^{-1}
38	$\mathbf{2.637 \times 10^{-2}}$	$\mathbf{3.072 \times 10^{-2}}$	$\mathbf{3.622 \times 10^{-3}}$	5.424×10^{-1}	8.168×10^{-1}
39	$\mathbf{2.645 \times 10^{-2}}$	$\mathbf{3.798 \times 10^{-2}}$	$\mathbf{3.843 \times 10^{-3}}$	5.788×10^{-1}	8.138×10^{-1}
40	$\mathbf{1.060 \times 10^{-2}}$	$\mathbf{2.718 \times 10^{-2}}$	$\mathbf{1.321 \times 10^{-3}}$	6.335×10^{-1}	8.018×10^{-1}
41	$\mathbf{1.214 \times 10^{-2}}$	$\mathbf{3.587 \times 10^{-2}}$	$\mathbf{1.430 \times 10^{-3}}$	6.788×10^{-1}	8.493×10^{-1}
42	$\mathbf{2.570 \times 10^{-2}}$	$\mathbf{2.749 \times 10^{-2}}$	$\mathbf{4.124 \times 10^{-3}}$	8.253×10^{-1}	9.352×10^{-1}
43	$\mathbf{1.991 \times 10^{-2}}$	$\mathbf{2.254 \times 10^{-2}}$	$\mathbf{4.802 \times 10^{-3}}$	6.679×10^{-1}	9.388×10^{-1}
44	$\mathbf{1.856 \times 10^{-2}}$	$\mathbf{2.275 \times 10^{-2}}$	$\mathbf{6.261 \times 10^{-3}}$	7.001×10^{-1}	9.578×10^{-1}
45	$\mathbf{7.134 \times 10^{-3}}$	$\mathbf{8.468 \times 10^{-3}}$	$\mathbf{4.337 \times 10^{-3}}$	6.406×10^{-1}	9.617×10^{-1}

Tabela A.2: *P-value* resultante do teste de causalidade de Granger aplicado entre o total de visualizações, comentários e tópicos negativos e muito negativos com o preço do *Bitcoin*.

Time lag	Neutros		Total		*Transações*
	Respostas	Tópicos	Respostas	Tópicos	
01	6.112×10^{-2}	5.727×10^{-1}	$\mathbf{3.729 \times 10^{-2}}$	8.422×10^{-1}	8.177×10^{-2}
02	2.616×10^{-1}	$\mathbf{3.163 \times 10^{-3}}$	9.704×10^{-2}	$\mathbf{1.506 \times 10^{-2}}$	3.335×10^{-1}
03	$\mathbf{1.042 \times 10^{-2}}$	$\mathbf{4.304 \times 10^{-3}}$	$\mathbf{1.496 \times 10^{-4}}$	$\mathbf{2.666 \times 10^{-4}}$	5.627×10^{-1}
04	$\mathbf{6.156 \times 10^{-3}}$	$\mathbf{1.185 \times 10^{-3}}$	$\mathbf{1.218 \times 10^{-4}}$	$\mathbf{2.677 \times 10^{-6}}$	7.033×10^{-1}
05	$\mathbf{7.948 \times 10^{-3}}$	$\mathbf{7.180 \times 10^{-3}}$	$\mathbf{3.018 \times 10^{-4}}$	$\mathbf{5.006 \times 10^{-5}}$	7.476×10^{-1}
06	$\mathbf{4.211 \times 10^{-2}}$	$\mathbf{9.786 \times 10^{-3}}$	$\mathbf{3.971 \times 10^{-3}}$	$\mathbf{7.898 \times 10^{-4}}$	9.110×10^{-1}
07	9.657×10^{-2}	$\mathbf{6.057 \times 10^{-3}}$	$\mathbf{8.313 \times 10^{-3}}$	$\mathbf{6.625 \times 10^{-4}}$	9.326×10^{-1}
08	9.692×10^{-2}	$\mathbf{9.179 \times 10^{-3}}$	$\mathbf{7.211 \times 10^{-3}}$	$\mathbf{1.054 \times 10^{-3}}$	9.582×10^{-1}
09	1.098×10^{-1}	$\mathbf{2.336 \times 10^{-2}}$	$\mathbf{1.086 \times 10^{-2}}$	$\mathbf{2.420 \times 10^{-3}}$	9.724×10^{-1}
10	1.309×10^{-1}	$\mathbf{3.215 \times 10^{-2}}$	$\mathbf{9.583 \times 10^{-3}}$	$\mathbf{1.917 \times 10^{-3}}$	9.914×10^{-1}
11	8.845×10^{-2}	5.805×10^{-2}	$\mathbf{4.891 \times 10^{-3}}$	$\mathbf{2.623 \times 10^{-3}}$	9.883×10^{-1}
12	1.206×10^{-1}	5.747×10^{-2}	$\mathbf{6.653 \times 10^{-3}}$	$\mathbf{2.063 \times 10^{-3}}$	9.954×10^{-1}
13	1.691×10^{-1}	6.027×10^{-2}	$\mathbf{6.462 \times 10^{-3}}$	$\mathbf{4.166 \times 10^{-3}}$	9.984×10^{-1}
14	1.615×10^{-1}	$\mathbf{1.669 \times 10^{-2}}$	$\mathbf{5.273 \times 10^{-3}}$	$\mathbf{2.599 \times 10^{-3}}$	9.984×10^{-1}
15	2.103×10^{-1}	$\mathbf{2.142 \times 10^{-2}}$	$\mathbf{8.864 \times 10^{-3}}$	$\mathbf{7.365 \times 10^{-4}}$	9.985×10^{-1}
16	2.820×10^{-1}	$\mathbf{2.248 \times 10^{-2}}$	$\mathbf{1.794 \times 10^{-2}}$	$\mathbf{1.762 \times 10^{-3}}$	9.990×10^{-1}
17	3.364×10^{-1}	$\mathbf{2.137 \times 10^{-2}}$	$\mathbf{1.963 \times 10^{-2}}$	$\mathbf{3.279 \times 10^{-3}}$	9.968×10^{-1}
18	4.997×10^{-1}	1.095×10^{-1}	7.262×10^{-2}	$\mathbf{1.352 \times 10^{-2}}$	9.986×10^{-1}
19	1.406×10^{-1}	9.277×10^{-2}	$\mathbf{2.245 \times 10^{-2}}$	$\mathbf{2.810 \times 10^{-2}}$	9.962×10^{-1}
20	2.035×10^{-1}	1.714×10^{-1}	$\mathbf{3.739 \times 10^{-2}}$	$\mathbf{3.904 \times 10^{-2}}$	9.926×10^{-1}
21	2.387×10^{-1}	1.893×10^{-1}	$\mathbf{2.209 \times 10^{-2}}$	$\mathbf{3.974 \times 10^{-2}}$	9.838×10^{-1}
22	2.341×10^{-1}	2.121×10^{-1}	$\mathbf{2.902 \times 10^{-2}}$	$\mathbf{4.663 \times 10^{-2}}$	9.896×10^{-1}
23	1.228×10^{-1}	2.242×10^{-1}	$\mathbf{1.490 \times 10^{-2}}$	$\mathbf{2.995 \times 10^{-2}}$	9.905×10^{-1}
24	1.235×10^{-1}	2.700×10^{-1}	$\mathbf{1.283 \times 10^{-2}}$	$\mathbf{4.437 \times 10^{-2}}$	9.930×10^{-1}
25	8.488×10^{-2}	2.954×10^{-1}	$\mathbf{8.142 \times 10^{-3}}$	5.647×10^{-2}	9.973×10^{-1}
26	$\mathbf{2.800 \times 10^{-2}}$	3.138×10^{-1}	$\mathbf{3.227 \times 10^{-3}}$	8.013×10^{-2}	9.965×10^{-1}
27	$\mathbf{1.555 \times 10^{-2}}$	1.811×10^{-1}	$\mathbf{1.827 \times 10^{-3}}$	8.202×10^{-2}	9.981×10^{-1}
28	$\mathbf{1.798 \times 10^{-2}}$	1.541×10^{-1}	$\mathbf{1.739 \times 10^{-3}}$	1.402×10^{-1}	9.989×10^{-1}
29	$\mathbf{2.544 \times 10^{-2}}$	1.641×10^{-1}	$\mathbf{1.968 \times 10^{-3}}$	1.211×10^{-1}	9.992×10^{-1}
30	5.686×10^{-2}	1.719×10^{-1}	$\mathbf{4.563 \times 10^{-3}}$	1.167×10^{-1}	9.992×10^{-1}
31	5.678×10^{-2}	1.355×10^{-1}	$\mathbf{6.613 \times 10^{-3}}$	6.358×10^{-2}	9.978×10^{-1}
32	5.449×10^{-2}	2.182×10^{-1}	$\mathbf{6.626 \times 10^{-3}}$	6.224×10^{-2}	9.983×10^{-1}
33	6.453×10^{-2}	2.393×10^{-1}	$\mathbf{5.266 \times 10^{-3}}$	5.540×10^{-2}	9.988×10^{-1}
34	$\mathbf{3.777 \times 10^{-2}}$	1.233×10^{-1}	$\mathbf{2.927 \times 10^{-3}}$	$\mathbf{3.914 \times 10^{-3}}$	9.990×10^{-1}
35	$\mathbf{1.377 \times 10^{-2}}$	9.028×10^{-2}	$\mathbf{4.981 \times 10^{-4}}$	$\mathbf{2.889 \times 10^{-4}}$	9.992×10^{-1}
36	$\mathbf{1.575 \times 10^{-2}}$	6.016×10^{-2}	$\mathbf{6.796 \times 10^{-4}}$	$\mathbf{3.944 \times 10^{-4}}$	9.989×10^{-1}
37	$\mathbf{1.069 \times 10^{-2}}$	$\mathbf{3.384 \times 10^{-2}}$	$\mathbf{5.577 \times 10^{-4}}$	$\mathbf{1.510 \times 10^{-4}}$	9.984×10^{-1}
38	$\mathbf{6.845 \times 10^{-3}}$	1.053×10^{-1}	$\mathbf{5.235 \times 10^{-4}}$	$\mathbf{8.143 \times 10^{-5}}$	9.988×10^{-1}
39	$\mathbf{3.761 \times 10^{-3}}$	$\mathbf{2.821 \times 10^{-2}}$	$\mathbf{4.123 \times 10^{-4}}$	$\mathbf{7.421 \times 10^{-5}}$	9.989×10^{-1}
40	$\mathbf{7.112 \times 10^{-4}}$	$\mathbf{8.240 \times 10^{-3}}$	$\mathbf{1.590 \times 10^{-4}}$	$\mathbf{7.953 \times 10^{-6}}$	9.991×10^{-1}
41	$\mathbf{7.054 \times 10^{-4}}$	$\mathbf{1.396 \times 10^{-2}}$	$\mathbf{1.790 \times 10^{-4}}$	$\mathbf{1.080 \times 10^{-5}}$	9.986×10^{-1}
42	$\mathbf{2.817 \times 10^{-3}}$	$\mathbf{1.702 \times 10^{-2}}$	$\mathbf{5.974 \times 10^{-4}}$	$\mathbf{1.026 \times 10^{-5}}$	9.992×10^{-1}
43	$\mathbf{1.297 \times 10^{-3}}$	$\mathbf{2.301 \times 10^{-2}}$	$\mathbf{1.532 \times 10^{-4}}$	$\mathbf{8.874 \times 10^{-6}}$	9.992×10^{-1}
44	$\mathbf{1.905 \times 10^{-3}}$	$\mathbf{2.391 \times 10^{-2}}$	$\mathbf{2.159 \times 10^{-4}}$	$\mathbf{1.648 \times 10^{-5}}$	9.990×10^{-1}
45	$\mathbf{1.563 \times 10^{-3}}$	$\mathbf{2.621 \times 10^{-2}}$	$\mathbf{1.997 \times 10^{-4}}$	$\mathbf{5.368 \times 10^{-5}}$	9.990×10^{-1}

Tabela A.3: *P-value* resultante do teste de causalidade de Granger aplicado entre os comentários e tópicos neutros, total de postagens diárias e total de transações diárias com o preço do *Bitcoin*.

B

Experimentos

Aqui serão apresentados, na íntegra, todos os resultados obtidos nos experimentos executados no Capítulo 4.

	AODE					
Janela	modo *paper*			modo *count*		
	Precisão	Cobertura	Acurácia	Precisão	Cobertura	Acurácia
01	0,4963	0,4963	0,4960	0,4997	0,4997	0,4980
02	0,4851	0,4852	0,4861	0,4751	0,4770	0,4743
03	0,5329	0,5316	0,5335	0,5081	0,5072	0,5039
04	**0,5429**	**0,5425**	**0,5434**	0,4932	0,4946	0,4901
05	0,4905	0,4905	0,4910	0,5149	0,5113	0,5069
06	0,5035	0,5035	0,5029	0,4862	0,4895	0,4851
07	0,4838	0,4839	0,4831	0,4861	0,4896	0,4851
08	0,4712	0,4712	0,4712	0,4859	0,4897	0,4851
09	0,4747	0,4750	0,4742	0,4919	0,4948	0,4900
10	0,5001	0,5001	0,5	0,4757	0,4850	0,4801
11	0,4990	0,4990	0,5	0,5056	0,5032	0,4980
12	0,4902	0,4902	0,4900	0,4984	0,4990	0,4940
13	0,2465	0,5	0,4930	0,4919	0,4959	0,4910
14	0,2465	0,5	0,4930	0,4916	0,4961	0,4910
15	0,2465	0,5	0,4930	0,4807	0,4925	0,4870
16	0,2465	0,5	0,4930	0,4962	0,4988	0,4930
17	0,2470	0,5	0,4940	0,4724	0,4932	0,4880
18	0,2470	0,5	0,4940	0,4494	0,4872	0,4820
19	0,2470	0,5	0,4940	0,4442	0,4894	0,4840
20	0,2470	0,5	0,4940	0,4749	0,4954	0,4900
21	0,2475	0,5	0,4950	0,4713	0,4956	0,4910
22	0,2475	0,5	0,4950	0,4536	0,4915	0,4870
23	0,2475	0,5	0,4950	0,5384	0,5056	0,5009
24	0,2475	0,5	0,4950	0,5098	0,5015	0,4970
25	0,248	0,5	0,496	0,4463	0,4937	0,49
26	0,248	0,5	0,496	0,4114	0,4877	0,484
27	0,248	0,5	0,496	0,4282	0,4938	0,49
28	0,248	0,5	0,496	0,4747	0,4978	0,494
29	0,2474	0,5	0,4949	0,4120	0,4917	0,4869
30	0,2474	0,5	0,4949	0,4741	0,4977	0,4929
31	0,2474	0,5	0,4949	0,4547	0,4957	0,4909
32	0,2474	0,5	0,4949	0,4131	0,4958	0,4909
33	0,2479	0,5	0,4959	0,3719	0,4959	0,4919
34	0,2479	0,5	0,4959	0,3464	0,4938	0,4899
35	0,2479	0,5	0,4959	0,3464	0,4938	0,4899
36	0,2479	0,5	0,4959	0,2469	0,4959	0,4919
37	0,2474	0,5	0,4949	0,3714	0,4958	0,4909
38	0,2474	0,5	0,4949	0,4974	0,4999	0,4949
39	0,2474	0,5	0,4949	0,5813	0,5019	0,4969
40	0,2474	0,5	0,4949	0,5813	0,5019	0,4969
41	0,2469	0,5	0,4939	0,4607	0,4978	0,4919
42	0,2469	0,5	0,4939	0,3882	0,4937	0,4879
43	0,2469	0,5	0,4939	0,2449	0,4918	0,4858
44	0,2469	0,5	0,4939	0,2449	0,4918	0,4858
45	0,2474	0,5	0,4949	0,2474	0,5	0,4949

Tabela B.1: Tabela com resultados do experimento correspondente aos primeiros resultados deste trabalho. Contém o resultado dos classificadores AODE utilizando janelas deslizantes que variam entre 1 e 45.

Janela	AODE					
	modo *paper*			modo *count*		
	Precisão	Cobertura	Acurácia	Precisão	Cobertura	Acurácia
01	0,5482	0,5450	0,5463	0,5323	0,5316	0,5309
02	0,4903	0,4912	0,4896	0,5979	0,5891	0,5876
03	0,4984	0,4991	0,4974	0,5981	0,5763	0,5751
04	0,5131	0,5046	0,5025	0,5537	0,5401	0,5388
05	0,4695	0,4945	0,4922	0,5506	0,5351	0,5336
06	0,5638	0,5101	0,5077	0,5439	0,5299	0,5284
07	0,4765	0,4940	0,4895	0,5576	0,5393	0,5364
08	0,4602	0,4944	0,4895	0,5608	0,5346	0,5312
09	0,5623	0,5396	0,5364	0,5723	0,5349	0,5312
10	0,5754	0,5449	0,5416	0,5723	0,5349	0,5312
11	0,5807	0,5494	0,5445	0,5979	0,5350	0,5287
12	0,5913	0,5547	0,5497	0,5908	0,5399	0,5340
13	0,2539	0,5000	0,5078	0,6271	0,5455	0,5392
14	0,2539	0,5000	0,5078	0,6183	0,5404	0,5340
15	0,2552	0,5000	0,5105	0,5558	0,5237	0,5157
16	0,2552	0,5000	0,5105	0,6169	0,5401	0,5315
17	0,2552	0,5000	0,5105	0,6722	0,5407	0,5315
18	0,2552	0,5000	0,5105	0,6150	0,5251	0,5157
19	0,2433	0,5000	0,4867	0,5961	0,5246	0,5132
20	0,2566	0,5000	0,5132	0,5458	0,5091	0,4973
21	0,2566	0,5000	0,5132	0,6136	0,5249	0,5132
22	0,2566	0,5000	0,5132	0,6236	0,5200	0,5079
23	0,2579	0,5000	0,5159	0,5431	0,5044	0,4893
24	0,2420	0,5000	0,4840	0,5431	0,5044	0,4893
25	0,2420	0,5000	0,4840	0,5431	0,5044	0,4893
26	0,2420	0,5000	0,4840	0,5431	0,5044	0,4893
27	0,2566	0,5000	0,5133	0,6209	0,5101	0,4973
28	0,2433	0,5000	0,4866	0,7472	0,5156	0,5026
29	0,2433	0,5000	0,4866	0,5778	0,5049	0,4919
30	0,2566	0,5000	0,5133	0,4931	0,4994	0,4866
31	0,2553	0,5000	0,5107	0,4945	0,4997	0,4892
32	0,2446	0,5000	0,4892	0,7459	0,5052	0,4946
33	0,2553	0,5000	0,5107	0,7459	0,5052	0,4946
34	0,2446	0,5000	0,4892	0,7459	0,5052	0,4946
35	0,2459	0,5000	0,4918	0,2459	0,5000	0,4918
36	0,2459	0,5000	0,4918	0,2459	0,5000	0,4918
37	0,2540	0,5000	0,5081	0,2459	0,5000	0,4918
38	0,2540	0,5000	0,5081	0,2459	0,5000	0,4918
39	0,2472	0,5000	0,4945	0,2472	0,5000	0,4945
40	0,2527	0,5000	0,5054	0,2472	0,5000	0,4945
41	0,2527	0,5000	0,5054	0,2472	0,5000	0,4945
42	0,2527	0,5000	0,5054	0,2472	0,5000	0,4945
43	0,2540	0,5000	0,5081	0,2459	0,5000	0,4918
44	0,2540	0,5000	0,5081	0,2459	0,5000	0,4918
45	0,2540	0,5000	0,5081	0,2459	0,5000	0,4918

Tabela B.2: Tabela com resultados do experimento correspondente à primeira Hipótese de Melhoria. Contém o resultado dos classificadores AODE utilizando janelas deslizantes que variam entre 1 e 45.

Janela	SVM			MLP		
	Precisão	Cobertura	Acurácia	Precisão	Cobertura	Acurácia
01	0,2525	0,5	0,5051	0,6497	0,5578	0,5618
02	0,5397	0,5055	0,5103	0,45	0,4902	0,4948
03	0,4489	0,4550	0,4559	0,5620	0,5590	0,5595
04	0,5128	0,5126	0,5129	0,5374	0,5344	0,5336
05	0,4914	0,4917	0,4922	0,5390	0,5389	0,5388
06	0,4543	0,4602	0,4611	0,5142	0,5135	0,5129
07	0,5174	0,5167	0,5156	0,5847	0,5794	0,5781
08	0,5215	0,5213	0,5208	0,5575	0,5566	0,5572
09	0,5415	0,5413	0,5416	0,5584	0,5578	0,5572
10	0,5374	0,5370	0,5364	0,5417	0,5409	0,5416
11	0,5300	0,5263	0,5235	0,5315	0,5266	0,5235
12	0,5465	0,5417	0,5392	0,5345	0,5344	0,5340
13	0,5494	0,5464	0,5445	0,5499	0,5499	0,5497
14	0,5767	0,5725	0,5706	0,5310	0,5301	0,5287
15	0,5125	0,5121	0,5105	0,5526	0,5498	0,5473
16	0,5383	0,5379	0,5368	0,5298	0,5285	0,5263
17	0,5646	0,5605	0,5578	0,5366	0,5366	0,5368
18	0,5255	0,5238	0,5210	0,5419	0,5420	0,5421
19	0,5852	0,5800	0,5767	0,5687	0,5677	0,5661
20	0,5627	0,5588	0,5555	0,6135	0,6136	0,6137
21	0,5446	0,5425	0,5396	0,5810	0,5789	0,5767
22	0,5710	0,5686	0,5661	0,6029	0,6030	0,6031
23	0,5670	0,5588	0,5531	0,5849	0,5825	0,5797
24	0,5451	0,5382	0,5319	0,6069	0,6036	0,6063
25	0,5503	0,5468	0,5425	0,5852	0,5853	0,5851
26	0,5642	0,5581	0,5531	0,5897	0,5891	0,5904
27	0,5302	0,5278	0,5240	0,5779	0,5779	0,5775
28	0,5349	0,5327	0,5294	0,5998	0,5996	0,5989
29	0,5516	0,5487	0,5454	0,5938	0,5938	0,5935
30	0,5583	0,5576	0,5561	0,5824	0,5799	0,5775
31	0,5569	0,5555	0,5537	0,5715	0,5672	0,5645
32	0,5332	0,5330	0,5322	0,6032	0,5990	0,5967
33	0,5610	0,5603	0,5591	0,6124	0,6049	0,6021
34	0,5541	0,5541	0,5537	0,5587	0,5587	0,5591
35	0,5348	0,5348	0,5351	0,5731	0,5731	0,5729
36	0,4977	0,4977	0,4972	0,6114	0,6112	0,6108
37	0,4924	0,4924	0,4918	0,5735	0,5717	0,5729
38	0,5088	0,5087	0,5081	0,5741	0,5736	0,5729
39	0,4891	0,4891	0,4891	0,5705	0,5703	0,5706
40	0,4836	0,4836	0,4836	0,5328	0,5327	0,5326
41	0,5269	0,5268	0,5271	0,5433	0,5431	0,5434
42	0,5108	0,5108	0,5108	0,5324	0,5318	0,5326
43	0,5164	0,5155	0,5136	0,5651	0,5639	0,5628
44	0,5209	0,5204	0,5191	0,5558	0,5535	0,5519
45	0,5154	0,5150	0,5136	0,5267	0,5259	0,5245

Tabela B.3: Tabela com resultados do experimento correspondente à segunda Hipótese de Melhoria. Contém o resultado dos classificadores SVM e MLP utilizando janelas deslizantes que variam entre 1 e 45.

Janela	Random Forest					
	200 Árvores			500 Árvores		
	Precisão	Cobertura	Acurácia	Precisão	Cobertura	Acurácia
01	0,5456	0,5393	0,5412	0,5274	0,5239	0,5257
02	0,5515	0,5510	0,5515	0,5415	0,5403	0,5412
03	0,5567	0,5482	0,5492	0,5613	0,5534	0,5544
04	0,5723	0,5694	0,5699	0,5658	0,5643	0,5647
05	0,5760	0,5747	0,5751	0,5889	0,5794	0,5803
06	0,5803	0,5803	0,5803	0,5754	0,5749	0,5751
07	0,5431	0,5423	0,5416	0,5781	0,5778	0,5781
08	0,5702	0,5665	0,5677	0,5310	0,5306	0,5312
09	0,5827	0,5767	0,5781	0,5833	0,5829	0,5833
10	0,5535	0,5509	0,5520	0,5670	0,5609	0,5625
11	0,5817	0,5734	0,5759	0,5810	0,5810	0,5811
12	0,5521	0,5509	0,5497	0,5539	0,5514	0,5497
13	0,5707	0,5697	0,5706	0,5601	0,5592	0,5602
14	0,5441	0,5438	0,5445	0,5336	0,5335	0,5340
15	0,5682	0,5682	0,5684	0,5530	0,5529	0,5526
16	0,5536	0,5534	0,5526	0,5526	0,5505	0,5526
17	0,5594	0,5550	0,5578	0,5734	0,5724	0,5736
18	0,5680	0,5673	0,5684	0,5414	0,5406	0,5421
19	0,5926	0,5830	0,5873	0,5602	0,5587	0,5608
20	0,5667	0,5630	0,5661	0,5606	0,5606	0,5608
21	0,5218	0,5192	0,5238	0,5494	0,5481	0,5502
22	0,5708	0,5698	0,5714	0,5493	0,5484	0,5502
23	0,5600	0,5531	0,5585	0,5793	0,5771	0,5797
24	0,6254	0,6187	0,6223	0,5742	0,5713	0,5744
25	0,5686	0,5661	0,5691	0,5704	0,5702	0,5691
26	0,5773	0,5764	0,5744	0,5736	0,5723	0,5744
27	0,5880	0,5880	0,5882	0,5661	0,5658	0,5668
28	0,5835	0,5834	0,5828	0,5499	0,5487	0,5508
29	0,5563	0,5528	0,5561	0,5553	0,5545	0,5561
30	0,5779	0,5779	0,5775	0,5452	0,5452	0,5454
31	0,5762	0,5731	0,5752	0,5424	0,5410	0,5430
32	0,5758	0,5733	0,5752	0,5541	0,5541	0,5537
33	0,5532	0,5525	0,5537	0,5641	0,5632	0,5645
34	0,5588	0,5575	0,5591	0,5538	0,5515	0,5537
35	0,5733	0,5732	0,5729	0,5513	0,5500	0,5513
36	0,5684	0,5660	0,5675	0,5684	0,5660	0,5675
37	0,5645	0,5600	0,5621	0,5843	0,5758	0,5783
38	0,5681	0,5662	0,5675	0,5833	0,5759	0,5783
39	0,5596	0,5596	0,5597	0,5664	0,5642	0,5652
40	0,5484	0,5414	0,5434	0,5606	0,5588	0,5597
41	0,5966	0,5911	0,5923	0,5680	0,5576	0,5597
42	0,5445	0,5422	0,5434	0,5490	0,5490	0,5489
43	0,5460	0,5456	0,5464	0,5460	0,5458	0,5464
44	0,5516	0,5510	0,5519	0,5636	0,5612	0,5628
45	0,5571	0,5566	0,5573	0,5406	0,5397	0,5409

Tabela B.4: Tabela com resultados do experimento correspondente à segunda Hipótese de Melhoria. Contém o resultado do *Random Forest* utilizando janelas deslizantes que variam entre 1 e 45.