

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Weuler Borges Santos Filho

**Prevendo o preço do Bitcoin com Redes Neurais
usando dados do Twitter e de Mercado**

Uberlândia, Brasil

2019

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Weuler Borges Santos Filho

**Prevendo o preço do Bitcoin com Redes Neurais usando
dados do Twitter e de Mercado**

Trabalho de conclusão de curso apresentado à
Faculdade de Computação da Universidade Fe-
deral de Uberlândia, Minas Gerais, como requi-
sito exigido parcial à obtenção do grau de Ba-
charel em Ciência da Computação.

Orientador: Prof. Dr. Ivan da Silva Sendin

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2019

Weuler Borges Santos Filho

Prevendo o preço do Bitcoin com Redes Neurais usando dados do Twitter e de Mercado

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 20 de dezembro de 2019:

Prof. Dr. Ivan da Silva Sendin
Orientador

Prof.^a Dra. Márcia Aparecida Fernandes
Professora

Prof. Dr. Luiz Cláudio Theodoro
Professor

Uberlândia, Brasil
2019

*Aos meus pais, que me mantiveram acreditando enquanto eu duvidava e a Ariane sem a qual
esse trabalho jamais teria acontecido.*

Resumo

Com o crescimento e a popularização do Twitter, as opiniões e sentimentos expressos em forma de *tweets*, representam uma grande quantidade de dados que são valiosas para as empresas e podem ser analisadas. Neste trabalho, analisamos *tweets* do período de 1 de Janeiro de 2019 até 31 de Março de 2019 com o objetivo de estabelecer relações entre o sentimento desses *tweets* e os movimentos do preço do Bitcoin. Para estudar essas relações, modelamos um problema de classificação, onde dadas as métricas para uma janela de tempo, classificamos se o preço vai subir ou descer na janela de tempo seguinte. Utilizamos em nossos experimentos técnicas de aprendizagem supervisionada treinando Rede Neural Perceptron Multicamadas para fazer a classificação. Para efeitos de comparação, treinamos também modelos utilizando *Random Forests* e *Support Vector Machines*. Além disso, enriquecemos os experimentos com dados do Mercado de Bitcoin e treinamos os mesmos modelos utilizando tais dados, obtendo vários resultados promissores. Utilizando um Perceptron de Múltiplas camadas, obtivemos uma precisão de 84% tendo como fonte de dados apenas dados do Twitter. Os modelos usando *Random Forests* e *Support Vector Machines* que serviram apenas para comparação, também mostraram resultados acima do esperado utilizando dados do Twitter, sendo que a *Random Forests* alcançou 76% de precisão e a *Support Vector Machine* 75% de precisão.

Palavras-chave: Bitcoin, Twitter, Redes Neurais, Análise de Sentimento, VADER.

Lista de ilustrações

Figura 1 – Gráfico histórico com o preço do Bitcoin. Retirado de (MARKETCAP, 2018)	10
Figura 2 – Exemplo de uma Rede Neural Perceptron Multicamadas com uma camada de entrada, duas camadas ocultas e uma camada de saída. Retirado de (NIELSON, 2018)	17
Figura 3 – Fluxograma da metodologia aplicada no presente trabalho	22
Figura 4 – Exemplo de análise de sentimento em Python	26

Lista de tabelas

Tabela 1 – Algumas palavras léxicas e seus respectivos score de sentimento. Retirado de (HUTTO; GILBERT, 2015)	16
Tabela 2 – Conjunto de dados entre Janeiro de 2019 e Março de 2019	24
Tabela 3 – Métricas obtidas para o experimento com dados do Twitter	32
Tabela 4 – Métricas obtidas para o experimento com dados do Mercado de Bitcoins . .	32
Tabela 5 – Métricas obtidas para o experimento com dados do Twitter e Mercado juntos	33
Tabela 6 – Métricas obtidas com as melhorias na PMC e dados do Twitter	33
Tabela 7 – Métricas obtidas com as melhorias na PMC e dados do Mercado de Bitcoins	33
Tabela 8 – Métricas obtidas com as melhorias na PMC e dados do Twitter e Mercado juntos	33
Tabela 9 – Métricas obtidas com a configuração original e dados do Twitter filtrados . .	34
Tabela 10 – Métricas obtidas com a configuração original e dados do Twitter filtrados junto dos dados de Mercado	34
Tabela 11 – Métricas obtidas com melhorias na PMC e dados do Twitter filtrados	34
Tabela 12 – Métricas obtidas com melhorias na PMC e dados do Twitter filtrados junto dos dados de Mercado	35
Tabela 13 – Métricas obtidas aplicando modelos treinados no Experimento 1 à 180 dias de dados do Mercado	35
Tabela 14 – Métricas obtidas aplicando modelos treinados com 180 dias de dados do Mercado aos dados de Mercado do Experimento 1	36

Lista de abreviaturas e siglas

ANEW	Affective Norms for English Words
API	Application Programming Interface
CSV	Comma-Separated-Values
GI	General Inquirer
LIWC	Linguistic Inquiry and Word Count
PMC	Perceptron Multicamadas
RF	Random Forest
RN	Rede Neural
SVM	Support Vector Machine
VADER	Valence Aware Dictionary and sEntiment Reasoner

Sumário

1	INTRODUÇÃO	9
1.1	Método	11
1.2	Resultados	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Bitcoin: o que é e como funciona	13
2.2	Twitter	14
2.3	Análise de Sentimentos	14
2.3.1	VADER	15
2.4	Redes Neurais Perceptron Multicamadas	17
2.5	Trabalhos Relacionados	19
3	DESENVOLVIMENTO	22
3.1	Coleta de dados do Twitter	22
3.2	Coleta de dados de mercado do Bitcoin	24
3.3	Ferramenta VADER em Python	24
3.4	Pré-processamento de dados	25
3.4.1	Vetores característica	27
3.4.2	Função objetivo	28
3.5	Treinamento das Redes Neurais Perceptron Multicamadas	29
3.5.1	Experimento original	30
3.5.2	Experimento original com melhorias na PMC	30
3.5.3	Experimento original com dados do Twitter filtrados	31
3.5.4	Experimento modificado com dados do Twitter filtrados	31
3.5.5	Validação cruzada de modelos com dados de mercado adicionais	31
4	RESULTADOS	32
4.1	Experimento Original	32
4.2	Experimento original com melhorias na PMC	33
4.3	Experimento original com dados do Twitter filtrados	34
4.4	Experimento modificado com dados do Twitter filtrados	34
4.5	Validação cruzada de modelos com dados de mercado adicionais	35
5	CONCLUSÃO	37
	REFERÊNCIAS	39

1 Introdução

O Bitcoin é um sistema *peer-to-peer* de dinheiro digital proposto em 2008 por Satoshi Nakamoto ([NAKAMOTO, 2008](#)). A moeda representou 66.6% de todo o mercado de criptomoedas ([COINMARKETCAP, 2018](#)) no dia 11 de Dezembro de 2019. A rede de Bitcoin usa um sistema de *proof-of-work* para gerar moedas e um sistema criptográfico para proteger transferências de dinheiro. Todas as transações feitas através da moeda são publicadas livremente na rede, a *blockchain* (“cadeia de blocos”), funcionando como um “livro-razão”, só que de forma pública, distribuída e imutável. Como a *blockchain* é pública, qualquer pessoa pode verificar as transações registradas nela a qualquer momento, mas não conseguirá fraudá-la. Um usuário mal-intencionado pode até adicionar ou retirar transações da *blockchain*, mas a fraude seria imediatamente percebida e rejeitada pelo resto da rede.

Apesar de ainda não haverem formas de usar Bitcoin de forma simples no dia-a-dia, a moeda passou de ferramenta de marketing ([ZANNI, 2015](#)) para uma moeda capaz de servir de refúgio para países em situação de crise e hiperinflação, como na Venezuela ([FRANCO, 2017](#)).

Além de servir muito bem como proteção contra a inflação, a moeda permite facilidade em transações internacionais, antes impossíveis sem um ou mais bancos intermediários. A medida que a quantidade de *tokens* cresce e a tecnologia fica mais conhecida, o valor unitário do Bitcoin sobe. Em 2013, quando o CoinMarketCap começou registrar seu valor, uma unidade da moeda era negociada a \$135,30. Três anos antes, no dia 22 de maio de 2010, 2 pizzas foram compradas por 10 mil unidades de Bitcoin. No dia (11 de dezembro de 2019) a moeda registra preço de \$7.169,67, bem abaixo do valor historicamente mais alto de aproximadamente \$19.000 dólares ([COINMARKETCAP, 2018](#)).

Apesar de historicamente o valor da moeda ter aumentado, dada sua natureza inovadora e seu real valor ainda por ser descoberto, a moeda é bastante volátil. Ainda segundo o CoinMarketCap, a moeda já viu seu preço subir mais de 100% e depois voltar ao patamar original num prazo de apenas 1 mês. Tal volatilidade e possibilidade de ganhos expressivos em pouco tempo atrai vários *traders*, que são pessoas que compram ativos unicamente com a intenção de especular e vende-los por um preço maior do que o comprado. Normalmente o prazo entre a compra e a venda é curto, variando de várias compras/vendas no mesmo dia (*day traders*) até um prazo de algumas semanas para os chamados *swing traders*. Essa prática será comumente referida ao longo do trabalho como *trading*.

Se tais operações podem ser realizadas por pessoas de forma lucrativa, deve haver também formas computacionais eficazes de prever posições lucrativas de operação. Nos últimos anos, as mais diversas técnicas de Inteligência Artificial foram empregadas para prever o mercado tradicional de ações. Redes Neurais (RNs), *Support Vector Machines* (SVMs), *Random*

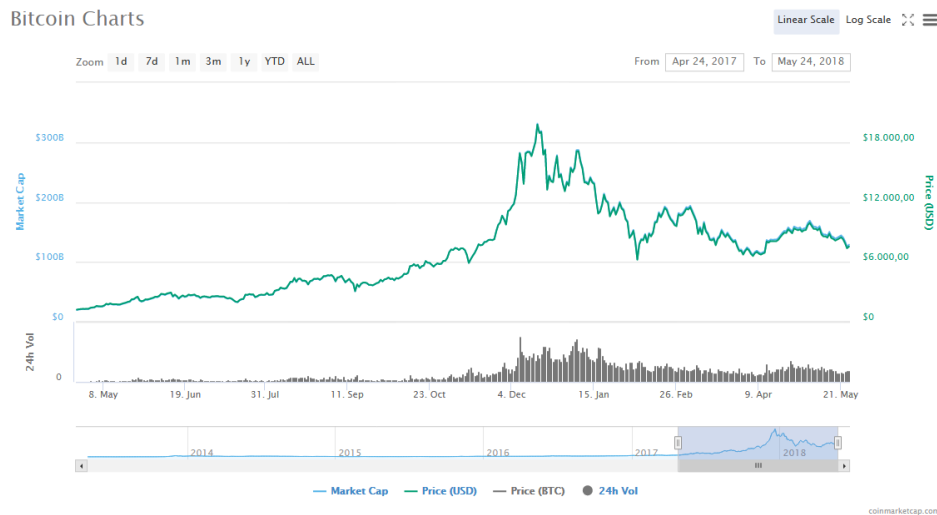


Figura 1 – Gráfico histórico com o preço do Bitcoin. Retirado de (MARKETCAP, 2018)

Forests (RFs) já foram usadas com sucesso para estes fins (KIMOTO et al., 1990) (KHAIDEM; SAHA; DEY, 2016) (HUANG; NAKAMORI; WANG, 2005). Outros trabalhos também foram bem sucedidos em utilizar análise de dados de redes sociais como o Twitter para prever o mercado de ações (SI et al., 2013).

No mercado de Bitcoin, foram feitos vários estudos (KRISTOUFEK, 2015), (KROLL; DAVEY; FELTEN, 2013), (SHAH; ZHANG, 2014), (KAMINSKI, 2013) foram feitos no sentido de tentar estabelecer correlação entre diversas variáveis e o preço do Bitcoin, nenhuma de forma extremamente conclusiva. Ainda assim, como não há nenhum tipo de regulação nesse sentido, existem (BLOOMBERG, 2017) diversos robôs tentando obter lucros através do mercado de Bitcoin.

(KAMINSKI, 2013) tentou estabelecer uma correlação através de regressão entre emoções em *tweets* e os indicadores de mercado do Bitcoin. Ao analisar *tweets* de um período de 104 dias entre 2013 e 2014, Kaminski (2013) concluiu que o Twitter funcionava muito bem como um espelho do mercado Bitcoin: quanto maior o volume de trades na rede Bitcoin, maior a quantidade de sinais de incerteza nos *tweets*, medida através de palavras como “hope”, “fear” e “worry”.

Apesar de parecer conclusivo, o estudo foi realizado num período completamente diferente do que vivemos agora. O valor de um *token* Bitcoin, no período em que foram analisados os *tweets*, flutuou entre \$600 e \$800 dólares, algo bem distante da alta histórica recente. Além disso, o Bitcoin ainda não era assunto de mídia e existiam na época muito menos criptomoedas alternativas ao Bitcoin do que as mais de 1600 existentes atualmente, segundo o (COINMARKETCAP, 2018).

Recentemente, (MAI et al., 2018) mostraram em um trabalho extremamente detalhado que dados presentes em fóruns e redes sociais são excelentes preditores do preço do Bitcoin,

apesar de que nem todas as mensagens e postagens têm a mesma relevância.

Esse ano, [Valencia, Gómez-Espinosa e Valdés-Aguirre \(2019\)](#) propuseram abordar o problema de uma forma diferente e que já obteve bastante sucesso no mercado de ações. Ao invés de lidar com ferramentas estatísticas como correlação e regressão, a proposta é tratar o problema como classificação, isto é, dadas as informações sobre um intervalo de tempo de tamanho t , queremos saber se o preço vai cair ou subir na janela seguinte. Em especial, o trabalho utilizava $t = 24$ horas, utilizando os dados de um dia para prever a direção do mercado no dia seguinte. Além disso, o trabalho propôs usar também dados históricos do mercado em adição aos dados de análise do Twitter.

[Valencia, Gómez-Espinosa e Valdés-Aguirre \(2019\)](#) obtiveram resultados bastante positivos em seu trabalho, alcançando 76% de precisão para uma Rede Neural Perceptron Multi-camadas (PMC) treinada com dados do mercado do Bitcoin em conjunto com sinais do Twitter. Apesar disso, ainda existe uma grande distância entre tornar esses resultados animadores algo que possa ser usado para criar uma estratégia de *trading* prática e efetiva. Neste trabalho, partiremos do ponto onde os autores parou, buscando diminuir essa distância respondendo as seguintes perguntas:

- (1) É possível reproduzir os bons resultados usando dados de outro período?
- (2) É possível melhorar os resultados do modelo utilizando algum filtro de qualidade nos *tweets*?
- (3) É possível treinar um “modelo universal” que, treinado em uma janela de tempo, obtém boa precisão prevendo o preço do Bitcoin em qualquer outra base de dados?

1.1 Método

Para verificar se os resultados de [Valencia, Gómez-Espinosa e Valdés-Aguirre \(2019\)](#) podem ser obtidos utilizando outra base de dados, precisamos reproduzir sua metodologia, com algumas restrições. Em seu trabalho, ele comparou o uso de Redes Neurais, SVMs e RF para Bitcoin, Ethereum, Ripple e Litecoin. Dos resultados do trabalho, foi possível observar que para o Bitcoin, as Redes Neurais foram superiores em todas as bases de dados. Poderíamos nos apoiar nesse resultado para justificar usar apenas Redes Neurais nesse trabalho, mas para completude, realizamos todos os experimentos também para RF e SVMs.

Realizamos ao todo 5 experimentos diferentes utilizando dados do Twitter e do Mercado de Bitcoin. Nos capítulos seguintes aprofundaremos nos passos executados para executar os experimentos e buscar respostas para as perguntas de pesquisa.

1.2 Resultados

Diante dos resultados obtidos pelos trabalhos anteriores, esperamos ser capazes de criar modelos com boa precisão utilizando dados de mercado e do Twitter. Além disso, esperamos que a aplicação de filtros de qualidade nos *tweets* torne os resultados estritamente melhores. Isso porquê, parece intuitivo pensar que é mais provável que os *tweets* mais *retweetados* e vistos, sejam os com maior potencial de impactar o mercado.

2 Fundamentação Teórica

Este capítulo apresenta um levantamento bibliográfico sobre Bitcoin, Análise de Sentimentos e Redes Neurais, três conceitos que fundamentam o trabalho.

2.1 Bitcoin: o que é e como funciona

O Bitcoin é uma coleção de conceitos e tecnologias que formam a base de dinheiro digital. As unidades de moeda chamadas bitcoins são usadas para armazenar e transmitir valor entre os participantes da rede bitcoin. Os usuários de Bitcoin se comunicam usando o protocolo bitcoin principalmente via Internet. A pilha de protocolos bitcoin, pode ser executada em uma ampla gama de dispositivos de computação, incluindo *laptops* e *smartphones*, tornando a tecnologia facilmente acessível (ANTONOPOULOS, 2014).

A moeda foi apresentada em 2008 por um programador ou por um grupo de programadores sob o pseudônimo Satoshi Nakamoto (NAKAMOTO, 2008). De código aberto, sua principal característica é ser descentralizada, isto é, não depender de uma autoridade central para a realização de transações financeiras. A tecnologia Bitcoin inclui recursos baseados em criptografia e assinaturas digitais para garantir a segurança da rede bitcoin (ANTONOPOULOS, 2014).

As transações são registradas em um banco de dados distribuídos, chamado de *block-chain*, que é um grande banco de dados que contém as transações que ocorreram no passado, bem como os atuais detentores dos fundos (ULRICH, 2017). Dessa maneira, não é necessária a confiança em um terceiro ou entidade central para que os dados de contabilidade estejam corretos e não sejam fraudados.

As moedas são geradas em blocos, através de pessoas que usam softwares de mineração digital. A quantidade de fundos disponibilizada é ajustada em uma crescente previsível e controlada – apenas 21 milhões de Bitcoins serão criadas até o ano 2140. Os cálculos feitos pelos “mineradores” ajudam a verificar as transações de toda a rede (CABRAL, 2013).

Todo usuário controla seus próprios fundos, por meio de uma chave privada criptográfica (FRANCO, 2014). Dessa forma, quando um usuário deseja gastar alguns fundos, ele deve usar essa chave privada para assinar uma mensagem informando para quem deseja enviar os fundos, bem como o valor a ser enviado. O usuário transmite essa mensagem assinada para a rede e todos os participantes da rede recebem uma cópia dela. Cada nó pode verificar independentemente a validade da mensagem e atualizar seu banco de dados interno adequadamente.

2.2 Twitter

Uma rede social muito popular atualmente é o Twitter¹, que possui mais de 330 milhões de usuários ativos por mês, segundo o portal de estatísticas Statista (2019). A rede social permite que os usuários publiquem pequenas mensagens para pessoas ou empresas de até 280 caracteres, conhecidos como “*tweets*”.

Além disso, os usuários podem incorporar os símbolos “@” e “#” em suas mensagens para garantir que membros específicos recebam as informações que estão divulgando. As pessoas usam o símbolo de hashtag² (#) antes de uma palavra-chave ou frase relevante nos tweets que publicam para classificá-los e facilitar a exibição deles na busca do Twitter, enquanto que o símbolo de arroba³ (@) é usado antes para se referir a uma conta específica no Twitter.

As mensagens de texto escrita pelos usuários do Twitter são mais fáceis de analisar devido ao limite de tamanho e porque os autores geralmente são diretos ao ponto (FRANCO, 2014). Assim, é mais fácil obter alta precisão na análise de sentimentos (LIU, 2012). Essas mensagens então, se tornou uma mina de ouro para dados opinativos, pois podem ser usados em experimentos de análise de opinião e análise de sentimentos (PAK; PAROUBEK, 2010). Portanto, essa rede social será importante para a extração de *tweets* que envolvem o assunto Bitcoin para a análise de influência do valor da moeda.

2.3 Análise de Sentimentos

Quando uma organização ou empresa precisa de opiniões públicas, a mesma conduz pesquisas de opinião. Essa aquisição de opiniões do público e do consumidor é importante para o marketing da empresa, pois é possível entender melhor o comportamento e o gosto dos seus consumidores.

Com o crescimento e a popularização das redes sociais, as opiniões em forma de comentários e avaliações representam uma grande quantidade de dados que são valiosas para as empresas. Por isso, profissionais que trabalham com ciência de dados desenvolveram ferramentas para gerar informações e *insights* a partir desses dados. Uma dessas ferramentas é a análise de sentimentos (CÉSAR, 2019).

De acordo com Liu (2012), o termo análise de sentimento apareceu pela primeira vez em (YI et al., 2003), mas as pesquisas sobre sentimentos apareceram mais cedo em (DAS; CHEN, 2001), (PANG; LEE; VAITHYANATHAN, 2002) e (TURNERY, 2002). A autora ainda descreve que a análise de sentimentos é o campo de estudo que analisa as opiniões, sentimentos, avaliações, atitudes e emoções das pessoas em relação a entidades como produtos, serviços, organizações, indivíduos, questões, eventos, tópicos e seus atributos (LIU, 2012).

¹ <https://twitter.com>

² <https://help.twitter.com/pt/using-twitter/how-to-use-hashtags>

³ <https://help.twitter.com/pt/using-twitter/mentions-and-replies>

Para a análise de sentimentos existem várias ferramentas para medir sentimentos, variando de abordagens lexicais a métodos de aprendizado de máquina. Apesar da grande popularidade de algumas, é uma dificuldade saber qual é o melhor para identificar a polaridade (positivas, negativas ou neutras) das mensagens e textos analisados. Para isso, existem os *benchmarks* (RIBEIRO et al., 2016) (ARAUJO et al., 2016) (ZIMBRA et al., 2018), que consiste em comparar as performances e desempenhos das ferramentas atuais.

Essas ferramentas de análise de sentimentos podem ser aplicáveis em várias áreas do conhecimento. Na área de jogos, por exemplo, a ferramenta *Opinion Finder* (WILSON et al., 2005) foi utilizada para analisar *tweets* da Copa do Mundo FIFA 2018 (FELICIANO; FARAH, 2018), enquanto que as ferramentas *Deeply Moving* (SOCHER et al., 2013), *Pattern* (SMEDT; DAELEMANS, 2012) e *SentiStrength* (THELWALL et al., 2010) foram utilizadas para analisar *tweets* sobre os Jogos Olímpicos de Inverno de Sochi em 2014 (KIRILENKO; STEPCHENKOVA, 2017). Na área da política, por exemplo, a ferramenta *VADER* (*Valence Aware Dictionary and sEntiment Reasoner*) (HUTTO; GILBERT, 2015) foi utilizada para analisar *tweets* em relação aos candidatos à presidência Donald Trump e Hillary Clinton nas eleições presidenciais dos Estados Unidos em 2016 (RAMTEKE et al., 2016), já a ferramenta *LIWC* (*Linguistic Inquiry and Word Count*) (TAUSCZIK; PENNEBAKER, 2010) foi utilizada no contexto das eleições federais alemãs para a investigação dos *tweets* (TUMASJAN et al., 2010).

Dentre as áreas de conhecimento citadas acima, ainda tem a de mercado financeiro, assunto principal deste trabalho. Para análise dos *tweets*, a ferramenta *VADER* (HUTTO; GILBERT, 2015) será utilizado.

2.3.1 VADER

A ferramenta *VADER* foi desenvolvida por (HUTTO; GILBERT, 2015) para análise de sentimentos no contexto de microblogs (Twitter e Facebook, por exemplo). O funcionamento da ferramenta se baseia em um dicionário em Inglês que mapeia características léxicas que são associados às suas medidas de intensidade de emoção. Algumas das características representadas no dicionário são, por exemplo, palavras que expressam raiva, felicidade e tristeza.

O dicionário léxico é um conjunto de palavras usadas para comunicação textual que foi construído a partir de dicionários bem formados, como o *LIWC* (TAUSCZIK; PENNEBAKER, 2010), *ANEW* (*Affective Norms for English Words*) (BRADLEY et al., 1999) e *GI* (*General Inquirer*) (STONE; DUNPHY; SMITH, 1966). Ainda foram acrescentados emoticons⁴, siglas⁵ e gírias⁶ que também expressam sentimentos.

O *score* de sentimento tem classificação em uma escala de “-4 (polaridade extremamente negativo)” a “4 (polaridade extremamente positivo)”, com permissão para “0 (polaridade

⁴ https://en.wikipedia.org/wiki/List_of_emoticons

⁵ http://en.wikipedia.org/wiki/List_of_acronyms

⁶ <https://www.internetslang.com/>

neutro)”. Como pode ser visto na Tabela 1, cada palavra tem o seu *score* de sentimento. Essa tabela possui somente 5 palavras e 2 emoticons, porém a lista original utilizada na ferramenta dispõe de mais de 7500 palavras.

Tabela 1 – Algumas palavras léxicas e seus respectivos score de sentimento. Retirado de (HUTTO; GILBERT, 2015)

Palavra	Score de Sentimento
okay	0.9
bad	-2.5
great	3.1
horrible	-2.5
sucks e sux	-1.5
:(	-1.9
:)	2.0

Em uma frase, as palavras não são as únicas a afetar a emoção. As pontuações podem intensificar a frase sem modificar a semântica, como por exemplo, a frase “*The food here is good!!!*” é mais intenso que “*The food here is good.*”. Palavras em *caps lock* podem enfatizar um sentimento relevante na presença de palavras sem *caps lock*, como por exemplo, a frase “*The food here is GREAT!*” é mais intenso que “*The food here is great.*”. Enquanto que a conjunção *mas* (“*but*”, em inglês) sinaliza uma mudança de polaridade com o sentimento do texto após a conjunção, como por exemplo, a frase “*The food here is great, but the service is horrible.*”.

Com as adversidades em relação às palavras e frases descritos acima, para a confecção final do dicionário léxico, os desenvolvedores de (HUTTO; GILBERT, 2015) recrutaram avaliadores humanos da AMT⁷ (*Amazon Mechanical Turk*) afim de receber dados significativos e de boa qualidade. Esse procedimento se baseia no conceito de sabedoria da multidão: a opinião coletiva é muitas vezes mais confiável do que a opinião individual (SUROWIECKI, 2005).

Os dicionários léxicos utilizados para fazer análises de sentimentos, em sua maioria, estão em língua inglesa, por ser considerado um idioma maduro e cheio de recursos (CHEKIMA; ALFRED, 2018) e este trabalho se beneficiará disso. Por causa disso, alguns pesquisadores brasileiros estão criando conjuntos de dados em língua portuguesa (CAVALCANTE; BARBOSA, 2017) (MORAES; MANSSOUR; SILVEIRA, 2015), ou criando métodos que estimam sentimentos em redes sociais para a língua portuguesa (SOUZA; PEREIRA; DALIP, 2017) (AGUIAR et al., 2018) (KANSANON; BRANDAO; DE PAULA PINTO, 2018).

⁷ <https://www.mturk.com/>

2.4 Redes Neurais Perceptron Multicamadas

Redes Neurais são uma técnica muito conhecida de *Machine Learning* (Aprendizado de Máquina, em português). Uma das técnicas muito conhecida de Redes Neurais é o Perceptron Multicamadas (PMC).

Uma Rede Neural PMC consiste em uma camada de entrada (*input layer*) com neurônios de entrada, uma ou mais camadas ocultas (*hidden layers*) e uma camada de saída (*output layer*) com neurônios de saída (NIELSON, 2018). A camada oculta possui esse nome porque não é possível prever a saída desejada nas camadas intermediárias.

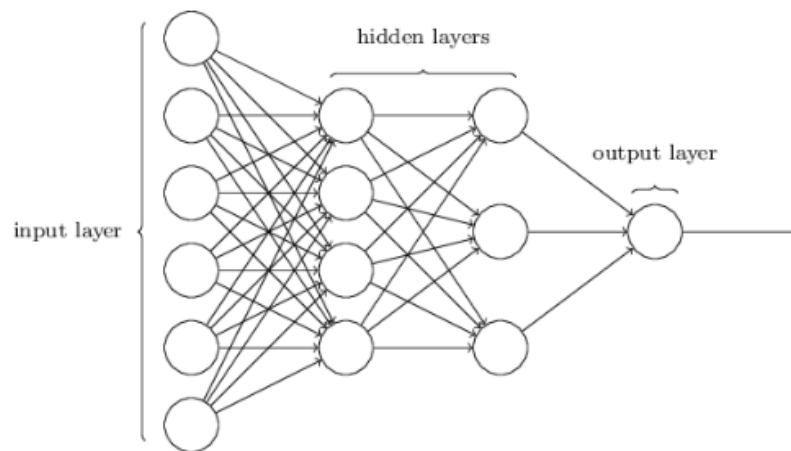


Figura 2 – Exemplo de uma Rede Neural Perceptron Multicamadas com uma camada de entrada, duas camadas ocultas e uma camada de saída. Retirado de (NIELSON, 2018)

O PMC utiliza o algoritmo de retropropagação (*Backpropagation*), cálculo do erro ocorrido na camada de saída da rede neural, para treinamento de aprendizagem supervisionada, em que cada amostra de dados utilizada apresenta um rótulo informando a que classificação ela se encaixa. A ideia geral é fazer com que a rede aprenda os padrões referentes a cada tipo de classe, pois, quando uma amostra desconhecida for fornecida à rede, ela seja capaz de estabelecer a qual classe tal amostra pertence.

O algoritmo de backpropagation consiste em duas fases, de acordo com:

1. O passo para frente (*forward pass*), onde as entradas são passadas através da rede e as previsões de saída são obtidas.
2. O passo para trás (*backward pass*), onde recalcula o valor dos pesos da camada de saída e assim procede para as camadas anteriores, de trás para a frente, realizando a retropropagação do erro obtido pela rede.

De acordo com DA SILVA, SPATTI e FLAUZINO (2010), o treinamento da rede é composto de 4 passos:

1. Inicialização

- Atribuição de valores aleatórios para os pesos e limites
- Escolha de valores iniciais que influencia o comportamento da rede
- Na ausência de conhecimento prévio os pesos e limites devem ter valores iniciais aleatórios e pequenos uniformemente distribuídos

2. Ativação

- Calcular os valores dos neurônios da camada oculta
- Calcular os valores dos neurônios da camada de saída

3. Treinamento dos Pesos

- Calcular os erros dos neurônios das camadas de saída e oculta
- Calcular a correção dos pesos
- Atualizar os pesos dos neurônios das camadas de saída e oculta

4. Iteração

- Repetir o processo a partir do passo 2 até que satisfaça o critério de erro

Durante a fase de treinamento, a saída do perceptron multicamada para uma determinada entrada pode não ser igual à saída desejada. Um sinal de erro é definido como a diferença entre a saída desejada e a real. O treinamento usa a magnitude desse sinal de erro para determinar em que grau os pesos na rede devem ser ajustados para que o erro geral do perceptron seja reduzido. Então, o objetivo do treinamento do PMC é otimizar os pesos para que a rede neural possa aprender a mapear corretamente as entradas para as saídas resultando em um menor erro (GARDNER; DORLING, 1998).

O uso de algoritmos e técnicas de *Machine Learning* para análise de sentimentos são muito utilizados, pois esses algoritmos são mais adaptáveis às alterações de entradas. Esses algoritmos geralmente são usados para classificação binária e previsões de sentimentos, como positivos ou negativos (AHMAD et al., 2017). Alguns trabalhos interessantes na literatura aplicam a técnica de análise de sentimento em textos de mídias sociais (Twitter, Tumblr, Facebook) comparando diversas técnicas de *Machine Learning* e podem ser vistos em (MAMGAIN et al., 2016), (AKHTAR et al., 2017), (KUMAR; JAISWAL, 2017), (RAMADHANI; GOO, 2017).

Para o presente trabalho, as Redes Neurais Perceptron Multicamadas serão utilizadas para a classificação dos *tweets*.

2.5 Trabalhos Relacionados

Esta seção tem como objetivo descrever trabalhos relacionados ao tema principal desse trabalho: predição do preço do Bitcoin. Os trabalhos propostos por [Valencia, Gómez-Espinosa e Valdés-Aguirre \(2019\)](#), [Fung et al. \(2019\)](#), [Jain et al. \(2018\)](#) e [Abraham, Higdon e Nelson \(2018\)](#) serão apresentados a seguir. Ao final, será discutido brevemente a relação destes trabalhos com a proposta deste trabalho.

[Abraham, Higdon e Nelson \(2018\)](#) apresentam um método de prever mudanças em preços do Bitcoin e Ethereum utilizando dados do Twitter e do Google Trends⁸.

Para a análise de sentimentos, os autores utilizaram VADER (*Valence Aware Dictionary and sEntiment Reasoner*), um método muito utilizado para o contexto de mídias sociais. Nesse caso, o método determina se os tweets geralmente são positivos ou negativos em suas opiniões sobre criptomoedas.

Para a análise dos dados, os autores fazem um detalhamento sobre quais dados são adequados como entradas do modelo de regressão linear múltipla. Isso inclui determinar quantos dos *tweets* possuem de fato algum sentimento e estabelecer uma relação entre o sentimento dos *tweets* sobre as criptomoedas e as mudanças nos preços das criptomoedas. Pois, se não contiver uma relação entre estas métricas e a mudança de preços, então não serão consideradas como entradas do modelo.

Os autores descobriram que a análise de sentimentos é menos eficaz para alterações de preços de criptomoedas em um ambiente em que os preços estão caindo, pois as pessoas que twittam sobre criptomoedas, mesmo quando seus preços caem, têm interesse nelas além da oportunidade de investimento, tornando os tweets tendenciosos a positivos. E que ao analisar os tweets, descobriram que o volume de *tweets* junto com o Google Trends estão altamente correlacionados com os preços das criptomoedas quando os preços aumentam e quando caem.

[Jain et al. \(2018\)](#) pretendem prever o preço futuro de duas criptomoedas, Bitcoin e Litecoin, usando um modelo de regressão multi-linear. Sua metodologia extrai e analisa *tweets* que são filtradas com o nome das criptomoedas, assim como dados de preços simultâneos extraídos do Coindesk⁹ e explora recursos significativos mapeados com os preços simultâneos das criptomoedas para criar curvas de previsão que podem prever os preços das criptomoedas em um futuro próximo.

Para a análise de sentimento, os autores utilizaram o TextBlob¹⁰ que é uma biblioteca Python. Para a análise dos dados, utilizaram o modelo de regressão linear múltiplo.

Os autores propuseram um *framework* que funciona em duas fases: fase de treinamento

⁸ Ferramenta do Google que mostra os mais populares termos buscados em um passado recente. Disponível em: <https://trends.google.com.br/trends>

⁹ <https://www.coindesk.com/price>

¹⁰ <https://textblob.readthedocs.io/en/dev/>

e fase de detecção. Basicamente, a frase de treinamento consiste em coletar os dados do Twitter, com relação às criptomoedas, convertê-los para o formato CSV e então, analisados pela polaridade. A quantidade de *tweets* positivos, negativos e neutros são contados e armazenados. Esses números contados são mapeados com a média do preços que ocorrem na duração correspondente de duas horas. O modelo então só é validado se o resultado é aceitável, pronto para ser usado para prever o futuro preço das criptomoedas. Caso contrário, um novo modelo é formado e o processo de treinamento e teste é repetido até que se obtenha um modelo aceitável. Na fase de detecção, *tweets* em tempo real são usados como entrada para o modelo e o modelo faz previsão do preço médio pela duração de duas horas.

Os resultados mostraram que o preço do Bitcoin não é muito afetado pelos sentimentos dos *tweets* em comparação ao preço do Litecoin. Os autores acreditam que a flutuação no preço do Bitcoin depende de outros fatores como custo de mineração e fator econômico.

Fung et al. (2019) descreve em seu artigo estratégias de *trading* utilizando notícias e *tweets* relacionados a Bitcoin. Os autores usaram análise de sentimento para cada notícia e *tweets* utilizando VADER e alguns modelos de ajuste para prever o mercado Bitcoin.

Para a estratégia de *trading*, os autores escolheram três modelos a partir da observação e testes dos dados empíricos. O primeiro modelo (Modelo 1) foi regressão logística com base na análise de sentimentos das notícias, o segundo modelo (Modelo 2) foi regressão logística com base na análise de sentimentos dos *tweets* e o terceiro modelo (Modelo 3) foi uma regressão linear de série temporal sobre preço e volume de tweets.

A partir dos dados empíricos, os autores observaram que existe um viés positivo com relação ao sentimento de notícias e *tweets*. Principalmente, devido à prevalência de criptomoedas nos últimos anos, a maioria das notícias relacionadas exerce uma visão otimista.

Ambas as estratégias de *trading* do Modelo 1 e do Modelo 3 forneceram um bom resultado em retorno e precisão no movimento de precificação. Pelo resultado, o sentimento das notícias é um bom preditor do retorno do Modelo 1. A estratégia do Modelo 3 fornece o melhor resultado, mas os autores usam o preço previsto em vez do preço real no dia anterior para a tomada de decisão.

Embora o volume de tweets possa rastrear o preço do estudo empírico, os autores apontam que um bom resultado também pode advir da sorte e das condições do mercado. Além disso, parece que todas as estratégias de *trading* são a favor do mercado em alta. Por isso, os autores acham necessário estender o período de teste para verificar se o bom desempenho depende das condições e da sorte do mercado.

Valencia, Gómez-Espinosa e Valdés-Aguirre (2019) estudam o comportamento dos mercados por meio da aplicação de técnicas de análise de sentimentos e aprendizado de máquina para a tarefa de previsão do mercado de ações. Para isso, propõe usar ferramentas de aprendizado de máquina e dados disponíveis de mídia social para prever o preço das criptomoedas

Bitcoin, Ethereum, Ripple e Litecoin.

Os autores ainda comparam a utilização de RN, SVM e RF ao utilizar elementos do Twitter e dados do mercado como características (*features*) de entradas.

Para a análise de sentimentos, os autores utilizaram o VADER. Para o aprendizado de máquina, O perceptron multicamadas (PMCs) foi utilizado, que é um tipo de RN. Para avaliar cada modelo foram utilizados as métricas de acurácia, precisão, *recall* e *f₁-score*.

Os resultados mostraram que é possível prever o mercado de criptomoedas usando aprendizado de máquina e análise de sentimentos, onde os dados do Twitter por si só poderiam ser usados para prever certas criptomoedas e que a RN supera os outros modelos.

Todos os trabalhos comentados acima abordam uma maneira diferente de prever o preço do Bitcoin e outras moedas. Entretanto, somente o último utiliza redes neurais junto com a análise de sentimentos para a análise dos dados obtidos do Twitter. Neste trabalho, utilizaremos redes neurais como nosso principal preditor, mas também compararemos seus resultados com previsões feitas por RF e SVMs.

3 Desenvolvimento

Nessa seção serão descritos a coleta dos dados do Twitter e de Mercado, o funcionamento da ferramenta VADER em Python e a preparação dos dados para serem utilizados no treinamento dos modelos de PMC, RF e SVMs, afim de calcular suas métricas de Precisão (*Precision*), Acurácia (*Accuracy*), *Recall* e F_1 -score. Por fim, serão reproduzidos vários experimentos diferentes para cumprimento dos objetivos do trabalho.

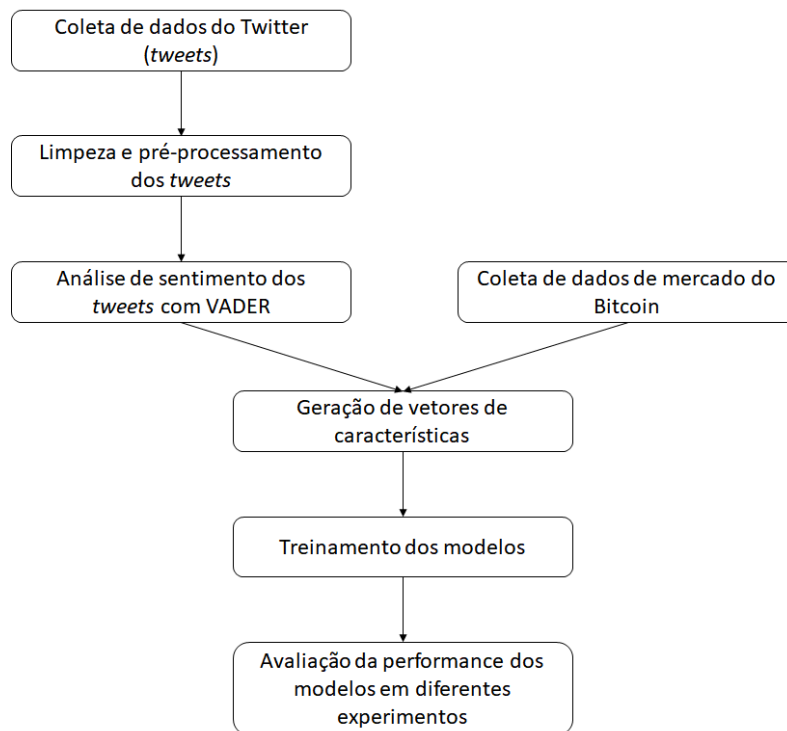


Figura 3 – Fluxograma da metodologia aplicada no presente trabalho

Nas seções a seguir serão descritos em detalhes a execução, desafios e aprendizados de cada etapa. O código do projeto está disponível no GitHub¹.

3.1 Coleta de dados do Twitter

É uma prática comum de grandes plataformas oferecerem APIs de desenvolvimento. O objetivo é facilitar a criação de um ecossistema de serviços e projetos relacionados à plataforma, fortalecendo ainda mais sua presença na Internet. Um exemplo clássico do uso dessas APIs é a possibilidade de se autenticar em um site usando sua conta do Facebook, Google ou Twitter.

¹ <https://github.com/filhoweuler/bitcoin-price-analysis>

Outro uso comum dessas APIs é a criação de ferramentas que ofereçam algum serviço que a plataforma original não provê de forma satisfatória. Em 2008, por exemplo, [Dodsworth \(2008\)](#) percebeu que existiam grandes problemas de usabilidade na *timeline*² do Twitter e criou o TweetDeck para cobrir essa lacuna usando a API do Twitter.

Para a realização dos experimentos deste trabalho, o período de coleta dos *tweets* foi de 1 de Janeiro até o dia 31 de Março de 2019. O motivo da escolha desse período foi que ele apresentou relativa estabilidade, tanto no volume de *tweets* quanto no preço do Bitcoin. Com isso, inicialmente, a coleta dos dados do Twitter consistia em usar API³ oferecida por eles para buscar e filtrar os dados sobre *tweets* de Bitcoins através de algumas chamadas simples.

Infelizmente, isso se mostrou rapidamente inviável, pois, apesar de a API do Twitter oferecer limites gratuitos razoáveis para dados em tempo real⁴, para ter o direito de explorar *tweets* com mais de 30 dias de idade é necessário adquirir uma assinatura “Premium”⁵. Ainda que estivéssemos dispostos a investir na API para acelerar a pesquisa, a versão mais cara da API que custa \$1899 dólares por mês permitia apenas 2500 requisições com 500 *tweets* cada por mês, totalizando 1250000 *tweets* num mês.

Pela análise do volume de *tweets* disponível em [BitInfoCharts](#) o volume de *tweets* no período alvo era de pelo menos 20.000 *tweets* por dia, totalizando pelo menos 1.800.000 *tweets* em 90 dias. Dados os limites da API do Twitter, seriam necessários dois meses e \$3.798 dólares para coletar os dados necessários, sendo isso motivo suficiente para descartar essa opção.

Uma próxima tentativa foi considerar a compra desses dados de empresas que já eram clientes da API do Twitter, mas todas as opções encontradas se mostraram pouco amigáveis tanto na facilidade de discutir as necessidades quanto para fazer um orçamento. A opção que escolhemos para a pesquisa surgiu da leitura do artigo ([FUNG et al., 2019](#)) que usou uma ferramenta de Web Scrapping desenvolvida em Python⁶ chamada TwitterScrapper⁷.

Apesar do Twitter dizer ter uma política “*scraper-friendly*”⁸, enfrentamos uma série de bloqueios por parte da plataforma. Um dos motivos para os bloqueios é que o TwitterScrapper não respeitava o *robots.txt* do Twitter que estabelecia um intervalo de pelo menos 1 segundo entre requisições consecutivas. Ao impor esse limite no nosso programa, o tempo para receber um bloqueio aumentou significativamente.

Para cada *tweet* coletamos a quantidade de *likes*, respostas, *retweets*, *timestamp* e o seu texto. Esses dados foram armazenados em um banco de dados temporal chamado InfluxDB⁹.

² Fluxo de *tweets* das contas que o usuário segue no Twitter. Disponível em: <https://help.twitter.com/pt/using-twitter/twitter-timeline>

³ <https://developer.twitter.com/>

⁴ <https://developer.twitter.com/en/docs/tweets/filter-realtime/overview>

⁵ <https://developer.twitter.com/en/pricing/search-fullarchive>

⁶ <https://www.python.org/>

⁷ <https://github.com/taspinar/twitterscraper>

⁸ <https://twitter.com/robots.txt>

⁹ <https://www.influxdata.com/>

Ao todo, coletamos 3.736.807 *tweets* que continham as palavras “bitcoin” ou “btc” (incluindo variações de capitalização) para o período entre 1 de Janeiro de 2019 e 31 de Março de 2019, com uma média de 41.520 *tweets* por dia.

Tabela 2 – Conjunto de dados entre Janeiro de 2019 e Março de 2019

Total de <i>Tweets</i>	Média de <i>Tweets</i> por dia
3.736.807	41.520

3.2 Coleta de dados de mercado do Bitcoin

Se por um lado tivemos dificuldade para obter dados públicos do Twitter através de sua API, a dificuldade no caso dos dados de mercado do Bitcoin foi escolher uma API dentre as disponíveis. A transparência de dados é um dos princípios fundamentais do Bitcoin e isso viabilizou a criação de inúmeras ótimas APIs, que disponibilizam os mais diversos dados crus e estatísticas agregadas.

Normalmente, essas APIs oferecem níveis gratuitos generosos e que são mais que suficientes para acadêmicos e curiosos. Mas, para mais acesso a dados, maior quantidade de requisições e autorização para explorar os dados comercialmente é cobrado uma chave *premium*.

Dentre as inúmeras opções, escolhemos a CryptoCompare¹⁰ usada por (VALENCIA; GÓMEZ-ESPINOSA; VALDÉS-AGUIRRE, 2019). Uma das vantagens desse serviço em específico é a grande quantidade de *Exchanges de Criptomoedas*, moedas e dados sobre cada uma delas que ele agrega.

Coletamos e armazenamos no InfluxDB os dados de hora em hora do dia 1 de Janeiro de 2019 até o dia 27 de Novembro de 2019. Os dados até 31 de Março serão usados para treinar os modelos que usarão dados do Twitter e do mercado. Já os dados posteriores a essa data serão usados para testar um modelo baseado apenas no mercado em um período de tempo maior. Usar a janela de tempo de 1 hora é conveniente no momento de transformar os dados em vetores de características para a rede neural, como explicaremos mais a frente.

3.3 Ferramenta VADER em Python

A ferramenta VADER (HUTTO; GILBERT, 2015) para a análise de sentimento dos *tweets* possui uma implementação em código-aberto¹¹ para a linguagem Python versão 3.8. Para instalar e usar só precisa seguir os passos do tutorial disponível no site Pypi¹².

¹⁰ <https://www.cryptocompare.com/>

¹¹ <https://github.com/cjhutto/vaderSentiment>

¹² <https://pypi.org/project/vaderSentiment/>

O *score* de sentimento calculado em Python retorna uma pontuação no intervalo de -1 a 1, do mais negativo ao mais positivo. O *score* de uma frase é calculada somando-se as pontuações de cada palavra listada no dicionário VADER na frase. Palavras individuais têm um *score* entre -4 a 4, mas o *score* de sentimento retornada de uma sentença está entre -1 e 1. No entanto, os autores de VADER aplicaram uma normalização ao total para mapeá-lo para um valor entre -1 e 1.

A função de normalização utilizada por [Hutto e Gilbert \(2015\)](#) é:

$$\frac{x}{\sqrt{x^2 + \alpha}} \quad (3.1)$$

onde, x é a soma do *score* de sentimentos das palavras pertencentes a sentença e α é um parâmetro de normalização que foi definido como 15¹³, valor escolhido pelos autores para que a normalização fosse próximo de -1 e 1. Assim, sempre que x aumenta, o seu *score* fica mais próximo de -1 ou 1.

O algoritmo em Python tem como saída 4 classes de sentimentos: positivo (pos), negativo (neg), neutro (neu) e *score* (compound). Os valores limiares que ([HUTTO; GILBERT, 2015](#)) utilizaram foram:

Sentimento positivo: $score \geq 0.05$;

Sentimento neutro: $(score > -0.05)$ e $(score < 0.05)$;

Sentimento negativo: $score \leq -0.05$.

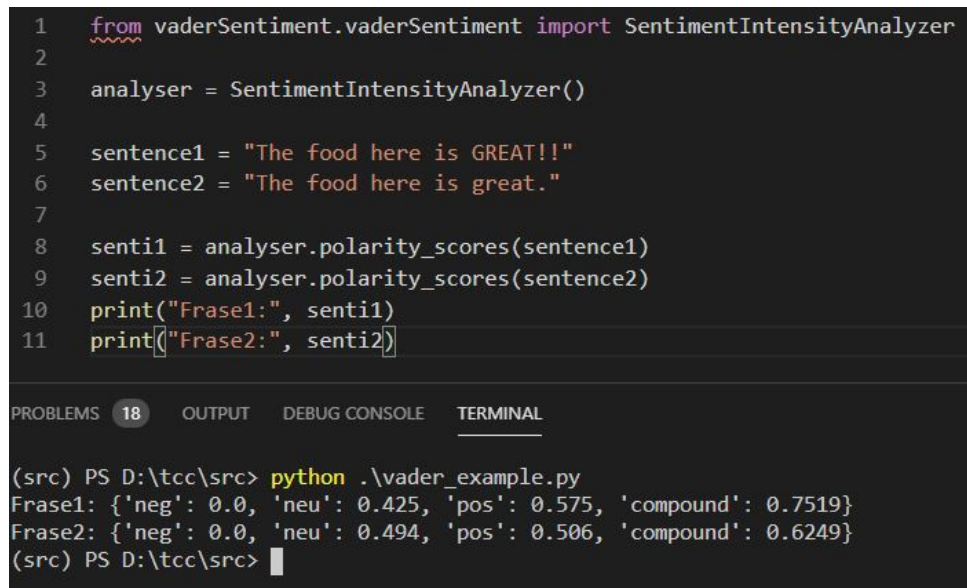
A Figura 4 apresenta o *score* de sentimento das frases “The food here is GREAT!!” (Frase 1) e “The food here is great.” (Frase 2) em Python. Como comentado em 2.3.1, a Frase 1 é mais intensa que a Frase 2 já que o *score* da Frase 1 foi maior que o da Frase 2. O *score* (compound) será a métrica utilizada para determinar o sentimento de uma frase, no caso deste trabalho, os *tweets* sobre Bitcoin.

Nos aprofundaremos em como usamos os dados gerados pelo VADER na seção seguinte.

3.4 Pré-processamento de dados

Nesta etapa, vamos agregar e processar os dados obtidos do Twitter e do mercado para gerar o vetor de características que será usado como entrada nos modelos de redes neurais.

¹³ <https://github.com/nltk/nltk/issues/1490>



```
1 from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
2
3 analyser = SentimentIntensityAnalyzer()
4
5 sentence1 = "The food here is GREAT!!"
6 sentence2 = "The food here is great."
7
8 senti1 = analyser.polarity_scores(sentence1)
9 senti2 = analyser.polarity_scores(sentence2)
10 print("Frase1:", senti1)
11 print("Frase2:", senti2)
```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL

```
(src) PS D:\tcc\src> python .\vader_example.py
Frase1: {'neg': 0.0, 'neu': 0.425, 'pos': 0.575, 'compound': 0.7519}
Frase2: {'neg': 0.0, 'neu': 0.494, 'pos': 0.506, 'compound': 0.6249}
(src) PS D:\tcc\src>
```

Figura 4 – Exemplo de análise de sentimento em Python

Os *tweets* obtidos através do *scrapping* contém uma série de elementos indesejados, como por exemplo, *links*, menções de usuários, espaços extras e *URLs* estão entre esses elementos que podem prejudicar a boa obtenção de sinais do texto. Por isso, antes de fazer qualquer análise, passamos os textos obtidos por uma função de limpeza que retorna somente o “texto útil” de um *tweet*.

Existem diversas técnicas de pré-processamento de dados (VIJAYARANI et al., 2015) (RANI et al., 2015), mas para o escopo deste trabalho, apenas remover os elementos que não agregam na análise basta. Para isso, utilizamos a implementação padrão de RegEx¹⁴ contida no módulo *re*, de expressões regulares, do Python.

Nessa etapa, vamos gerar dois conjuntos de dados distintos baseados nos *tweets*. No primeiro deles, teremos todos os *tweets* obtidos no período de estudo. No segundo conjunto, reuniremos os *tweets* mais relevantes do primeiro conjunto. Esse segundo conjunto de dados do Twitter será usado mais a frente para testar a hipótese de que um filtro de qualidade pode melhorar os resultados do modelo.

Existem várias possibilidades para decidir quais são os *tweets* mais relevantes. Temos métricas de *retweets*, *likes* e número de respostas. Podemos usar alguma combinação dessas métricas para definir um limiar arbitrário a partir do qual consideramos um *tweet* relevante. Outra possibilidade, é buscar também é criar uma lista de perfis famosos e só considerar *tweets* desses perfis.

A primeira abordagem tem um problema em potencial de que, se escolhermos um limiar muito baixo, estaremos aceitando *tweets* arbitrariamente pouco influentes. Por outro lado, se escolhermos um limiar muito alto, corremos o risco de ficar sem dados o suficiente para analisar

¹⁴ <https://docs.python.org/3/library/re.html>

em certos dias.

Usar apenas os *tweets* de perfis mais famosos também requer criar uma definição arbitrária de qual seriam esses perfis relevantes e tem o mesmo problema da primeira abordagem, pois em determinados dias esses perfis podem simplesmente não publicar nada relevante, aumentando artificialmente a influência dos perfis que acabarem publicando algo naquele dia. Além disso, o alcance no Twitter é dinâmico. Mesmo perfis muito famosos, eventualmente tem *tweets* que poucas pessoas vêem. Ao mesmo tempo, perfis completamente desconhecidos eventualmente podem ter um *tweet* viralizando e influenciando o humor do mercado.

Dito isso, criamos uma abordagem alternativa: para cada dia, a base de dados alternativa vai ter os 10% mais relevantes. Para determinar a relevância de um *tweet*, utilizaremos o número de *retweets*, já que ela está diretamente relacionada com o alcance efetivo de um *tweet*.

3.4.1 Vetores característica

Assim que os dados foram coletados e processados, podemos definir um vetor de característica V para um determinado tempo t , que será usado como entrada da nossa rede neural. O vetor de características que será utilizado foi definido por (VALENCIA; GÓMEZ-ESPINOSA; VALDÉS-AGUIRRE, 2019), como:

$$V(t) = \begin{bmatrix} neu, \\ norm, \\ neg, \\ pos, \\ pol, \\ close, \\ high, \\ low, \\ open, \\ volumeto \end{bmatrix} \quad (3.2)$$

onde,

neu é a média de sentimentos neutros definido por $\frac{\sum_{i=1}^n t_{neu}}{n}$

$norm$ é a soma dos *scores* de cada palavra definida por $\frac{\sum_{i=1}^n t_{norm}}{n}$

neg é a média de sentimentos negativos definido por $\frac{\sum_{i=1}^n t_{neg}}{n}$

pos é a média dos sentimentos positivos definido por $\frac{\sum_{i=1}^n t_{pos}}{n}$

pol é a média geométrica entre *pos* e *neg* definido por $\sqrt{V_{pos}V_{neg}}$

close é o preço fechado no período de tempo

high é o preço mais alto no período de tempo

low é o preço mais baixo no período de tempo

open é o preço de abertura no período de tempo

volumeto é o volume de *trading* no período de tempo

Com t_{neu} , t_{neg} , t_{pos} , t_{norm} o VADER calculou o *score* para cada *tweet*, e n são todos os *tweets* que compunham o período de tempo de uma determinada criptomoeda.

É importante salientar que o período de tempo t que usamos para a realização dos experimentos no trabalho foi de 24 horas, isto é, as *features* relacionadas a *tweets* foram computadas agregando todos os *tweets* num período de 24 horas, enquanto as *features* de mercado utilizaram dados agregados dessa mesma janela de tempo.

Além disso, usaremos diferentes sub-seções do vetor de características em cada experimento. As métricas *neu*, *norm*, *neg*, *pos* e *pol* serão usadas nos experimentos com dados apenas do Twitter e nos experimentos que usam os dois dados agregados. Já as métricas *close*, *high*, *low*, *open* e *volumeto* serão usados nos experimentos com dados do Mercado de Bitcoins e nos experimentos com os dados agregados.

3.4.2 Função objetivo

Além de saber as características de cada período de tempo t , para treinar as PMCs precisamos também encontrar qual foi o resultado daquele conjunto de dados. Chamamos esse resultado de função objetivo. Para cada período de tempo t , estamos interessados em estabelecer uma relação entre o vetor característica daquele período e o aumento ou decréscimo do preço no período seguinte $t + 1$.

A função objetivo foi definido de acordo com [Valencia, Gómez-Espinosa e Valdés-Aguirre \(2019\)](#):

$$Z(t) = \begin{cases} 1, & \text{se } fechamento(t+1) - fechamento(t) > 0 : \\ -1, & \text{se } fechamento(t+1) - fechamento(t) < 0 : \end{cases} \quad (3.3)$$

onde, $fechamento(t)$ é uma função que retorna o preço de fechamento no período t . Isso quer dizer que se o preço aumentar de t para $t + 1$, a função $Z(t) = 1$. Caso contrário, -1 .

3.5 Treinamento das Redes Neurais Perceptron Multicamadas

Todos os modelos deste trabalho foram implementados com a ajuda da biblioteca de aprendizado de máquina para Python chamada Scikit-learn¹⁵. Uma das implementações de PMC disponíveis na biblioteca é a `MLPClassifier`, que foi a implementação usada durante todos os experimentos do trabalho. Essa classe possui diversos parâmetros de customização, cada um com um valor padrão já testado e amplamente usado no mercado. Nas descrições de PMC apresentadas nas subseções a seguir destacaremos apenas os parâmetros alterados.

Além disso, é importante notar que apesar de os dados originais estarem agrupados por hora, para o treinamento dos modelos eles foram agrupados diariamente, de forma que então usamos os dados de um dia para prever a direção do preço no dia seguinte.

Em todos os experimentos seguimos também a estratégia de treinamento do autor original: treinar 50 modelos usando os mesmos dados e selecionar dentre eles os com melhor performance. Existem várias formas de avaliar a performance de uma rede neural. Neste trabalho, utilizaremos acurácia, precisão, *recall* e *f₁-score* que são definidas como segue:

$$precision = \frac{TP}{(TP + FP)} \quad (3.4)$$

$$recall = \frac{TP}{(TP + FN)} \quad (3.5)$$

$$f_1\text{-score} = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \quad (3.6)$$

$$accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (3.7)$$

onde, no nosso contexto

- *Verdadeiro Positivo (TP)*: um verdadeiro positivo é uma classificação correta de aumento de preço, ou seja, o modelo indica subida de preço e o preço realmente sobe;
- *Verdadeiro Negativo (TN)*: quando o preço cai e o modelo indicou corretamente a queda;
- *Falso Positivo (FP)*: quando o modelo indica que o preço subirá, mas na realidade ele cai;
- *Falso Negativo (FN)*: quando o modelo indica que o preço cairá, mas na realidade ele sobe.

Na prática, a acurácia mede a quantidade de acertos em geral, tanto positivos (subida de preço) quanto negativos (queda de preço). A precisão mede quantas das classificações de subida

¹⁵ <https://scikit-learn.org/stable/>

de preço foram corretas, dentre todas as classificações de subida de preço feitas. Isto é, ela mede a porcentagem de classificação de subidas de preço corretas. O *recall*, mede quantas previsões de subida foram feitas dentre as oportunidades reais de subida disponíveis. É uma medida que nos mostra quantas oportunidades o modelo está aproveitando. O f_1 -score é a média harmônica entre precisão e o *recall*.

O impacto real de cada uma dessas métricas no lucro de cada um desses modelos se expostos ao mercado real, depende largamente da estratégia de mercado usada. Supondo uma estratégia extremamente simples onde o operador compra quando o modelo indica que o preço vai subir e vende quando o modelo indica queda, precisão é a métrica mais importante. Isso porquê o objetivo é acertar o máximo possível as previsões de lucro feitas. Por exemplo, parece intuitivo ver que pode ser melhor perder algumas oportunidades de lucro tendo um baixo *recall* mas acertando frequentemente (alta precisão) do que aproveitar mais oportunidades perdendo em precisão. Portanto, em todos os experimentos usaremos a precisão para escolher o vencedor dentre os modelos treinados.

Para cumprir os objetivos do trabalho, realizamos vários experimentos, cada um treinando diversos modelos de PMC. Cada experimento utilizando diferentes bases de dados e diferentes configurações de treinamento. Os diferentes experimentos estão descritos nas subseções que seguem. Os resultados para cada um dos experimentos serão apresentados no capítulo seguinte.

3.5.1 Experimento original

Neste experimento, reproduzimos o experimento realizado por (VALENCIA; GÓMEZ-ESPINOSA; VALDÉS-AGUIRRE, 2019) para um PMC usando dados do Bitcoin apenas. O objetivo aqui é obter uma linha de base para nossos próprios experimentos e verificar se os dados seguiriam o mesmo padrão dos dados apresentados no artigo citado, para uma base de dados de um período de diferente. Em específico, usamos os dados obtidos do dia 01 de Janeiro de 2019 até 31 de Março de 2019.

Como no experimento original, utilizamos o solver “adam”, parâmetro α de 0.0002, apenas uma camada oculta com 30 neurônios e a tangente hiperbólica como função de ativação. No caso do PMC treinado usando dados do Twitter e Bitcoin em conjunto, utilizamos 55 neurônios na camada oculta, também como no experimento original.

3.5.2 Experimento original com melhorias na PMC

Analisando o experimento anterior, identificamos a possibilidade de melhorar os resultados fazendo algumas pequenas mudanças na PMC. Por exemplo, na documentação do MLP-Classifer do Scikit-learn ¹⁶ é informado que o solver ‘adam’ é indicado para bases de dados

¹⁶ https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

com alguns milhares de pontos. No caso de experimentos com poucos pontos de dados como o nosso, o mais indicado é o 'lbfgs'.

Além disso, nesse experimento utilizamos apenas 10 neurônios na camada oculta. No caso da nossa rede, esse valor representa duas vezes o número de neurônios na camada de entrada é tido empiricamente como um bom ponto de partida para começar testar a quantidade de neurônios na camada oculta.

Para alguns testes no experimento anterior o Scikit-learn estava exibindo um aviso informando que a quantidade máxima de iterações nos dados de treinamento foi atingida sem que o modelo pudesse convergir. Por isso, aumentamos o número máximo de iterações para 500. Para prevenir *over-fitting* com esse grande número de iterações, também adicionamos um parâmetro permitindo o *early stopping*.

3.5.3 Experimento original com dados do Twitter filtrados

Nesse experimento colocaremos a prova a tese de que os resultados para redes treinadas usando dados do Twitter podem ser melhoradas utilizando apenas dados de *tweets* mais relevantes para o período de tempo. Para isso, utilizamos a configuração da seção 3.5.1 mas substituindo a base de dados completo por uma base de dados que inclui apenas os *tweets* entre os 10% mais compartilhados.

3.5.4 Experimento modificado com dados do Twitter filtrados

Para comparação, reproduzimos uma versão com as melhorias na PMC da seção 3.5.1 e dados filtrados do Twitter.

3.5.5 Validação cruzada de modelos com dados de mercado adicionais

Esse último experimento pretende fazer a validação cruzada de modelos treinados em diferentes bases de dados. Devido à maior dificuldade de obter dados do Twitter, esse experimento foi realizado apenas com 180 dias adicionais de dados do mercado. Num primeiro momento, utilizamos os melhores modelos treinados com os dados de mercado nos experimentos anteriores para tentar prever o mercado na base de 180 dias. Depois, treinamos um modelo nessa base de 180 dias e utilizamos como teste a base de dados original de 90 dias.

4 Resultados

Neste capítulo serão mostrados os resultados da análise dos dados após o treinamento da Rede Neural PMC e também será feita uma comparação com os resultados obtidos pelos autores [Valencia, Gómez-Espinosa e Valdés-Aguirre \(2019\)](#).

É importante ressaltar que nas tabelas abaixo destacamos apenas os modelos com a melhor performance dentre outros 50 treinados para cada experimento. Para escolher o melhor modelo dentre esses 50, utilizamos a precisão como critério, pelos motivos apresentados em [3.5](#).

4.1 Experimento Original

É interessante observar como nesse experimento obtivemos resultados significativamente melhores que os obtidos por ([VALENCIA; GÓMEZ-ESPINOSA; VALDÉS-AGUIRRE, 2019](#)) para os modelos treinados com dados do Twitter. Em especial, a Random Forest e a SVM tiveram um ganho significativo de performance, abrindo a possibilidade de lucrar com tais modelos dentro de uma estratégia de trading real.

Para os modelos treinados com dados do mercado, observamos a PMC ganhando em performance da Random Forest e SVM. Nos modelos usando dados de Twitter e Mercado simultaneamente, essa tendência também se repetiu. Além disso, apesar da PMC treinada com dados de Mercado e Twitter ter apresentado uma pior precisão, é possível que essa PMC tenha um melhor desempenho na prática. Tal afirmação pode ser testada construindo uma estratégia de trading e empregando previsões dos dois modelos.

Tabela 3 – Métricas obtidas para o experimento com dados do Twitter

	<i>Precision</i>	<i>Accuracy</i>	<i>Recall</i>	F_1
PMC	0,61	0,59	0,81	0,70
Random Forest	0,76	0,66	0,625	0,68
SVM	0,75	0,70	0,75	0,75

Tabela 4 – Métricas obtidas para o experimento com dados do Mercado de Bitcoins

	<i>Precision</i>	<i>Accuracy</i>	<i>Recall</i>	F_1
PMC	0,72	0,70	0,81	0,76
Random Forest	0,66	0,62	0,75	0,70
SVM	0,62	0,55	0,62	0,62

Tabela 5 – Métricas obtidas para o experimento com dados do Twitter e Mercado juntos

	<i>Precision</i>	<i>Accuracy</i>	<i>Recall</i>	<i>F₁</i>
PMC	0,70	0,70	0,87	0,77
Random Forest	0,61	0,55	0,68	0,64
SVM	0,57	0,55	0,93	0,71

4.2 Experimento original com melhorias na PMC

Foram recomendadas pela própria documentação do *sklearn*¹, por exemplo, usar o solucionador de otimização de peso ‘lbfgs’ ao invés do ‘adam’ para o treinamento. Segundo a documentação o ‘lbfgs’ obtém um desempenho superior para treinamentos com poucos pontos de dados, como no nosso caso. Outras mudanças vieram de observações empíricas na melhora do desempenho. Dentre elas, temos a diminuição no número de neurônios na camada oculta de 30 para 10 e de 55 para 20, utilização de 500 iterações e aplicação do “early stopping”.

É interessante observar a melhora significativa do modelo usando apenas dados do Twitter, que passa a ter uma precisão de 84%, superando a PMC treinada apenas com dados do Mercado.

Tabela 6 – Métricas obtidas com as melhorias na PMC e dados do Twitter

	<i>Precision</i>	<i>Accuracy</i>	<i>Recall</i>	<i>F₁</i>
PMC	0,84	0,74	0,68	0,75
Random Forest	0,76	0,66	0,625	0,68
SVM	0,75	0,70	0,75	0,75

Tabela 7 – Métricas obtidas com as melhorias na PMC e dados do Mercado de Bitcoins

	<i>Precision</i>	<i>Accuracy</i>	<i>Recall</i>	<i>F₁</i>
PMC	0,70	0,66	0,75	0,72
Random Forest	0,66	0,62	0,75	0,70
SVM	0,62	0,55	0,62	0,62

Tabela 8 – Métricas obtidas com as melhorias na PMC e dados do Twitter e Mercado juntos

	<i>Precision</i>	<i>Accuracy</i>	<i>Recall</i>	<i>F₁</i>
PMC	0,71	0,62	0,62	0,66
Random Forest	0,61	0,55	0,68	0,64
SVM	0,57	0,55	0,93	0,71

¹ https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

4.3 Experimento original com dados do Twitter filtrados

Nesse experimento, reproduzimos a configuração original, utilizando apenas o top 10% *tweets* mais relevantes de cada dia, ordenados por número de *retweets*. Diferente do esperado, as métricas de todos os modelos treinados com esses dados filtrados se mostraram piores. Na tentativa de obter melhores resultados, testamos diferentes limiares como 5, 20, 30 ou 50% *tweets* mais relevantes, mas não foi observado nenhum padrão de melhora ou piora dos resultados, além de nenhum dos modelos obter resultados perto dos registrados na Tabela 6.

Tabela 9 – Métricas obtidas com a configuração original e dados do Twitter filtrados

	<i>Precision</i>	<i>Accuracy</i>	<i>Recall</i>	F_1
PMC	0,60	0,55	0,75	0,66
Random Forest	0,47	0,37	0,56	0,51
SVM	0,57	0,51	0,68	0,62

Tabela 10 – Métricas obtidas com a configuração original e dados do Twitter filtrados junto dos dados de Mercado

	<i>Precision</i>	<i>Accuracy</i>	<i>Recall</i>	F_1
PMC	0,71	0,70	0,89	0,78
Random Forest	0,55	0,48	0,62	0,58
SVM	0,63	0,62	0,87	0,73

4.4 Experimento modificado com dados do Twitter filtrados

Dessa vez, reproduzimos os experimento anterior, implementando as melhorias na PMC sugeridas em 4.2.

Tabela 11 – Métricas obtidas com melhorias na PMC e dados do Twitter filtrados

	<i>Precision</i>	<i>Accuracy</i>	<i>Recall</i>	F_1
PMC	0,75	0,70	0,75	0,75
Random Forest	0,47	0,37	0,56	0,51
SVM	0,57	0,51	0,68	0,62

Tabela 12 – Métricas obtidas com melhorias na PMC e dados do Twitter filtrados junto dos dados de Mercado

	<i>Precision</i>	<i>Accuracy</i>	<i>Recall</i>	<i>F₁</i>
PMC	0,70	0,66	0,75	0,72
Random Forest	0,55	0,48	0,62	0,58
SVM	0,63	0,62	0,87	0,73

4.5 Validação cruzada de modelos com dados de mercado adicionais

Com esse experimento, buscamos responder nossa última pergunta de pesquisa: é possível aplicar um modelo treinado em um determinado período de tempo em outro período?

Para isso, coletamos 180 dias adicionais de dados de mercado, do dia 01 de Abril de 2019 até 28 de Setembro de 2019. Primeiro, testamos os modelos já treinados para dados de mercado no experimento 4.2 utilizando dados desse período de tempo expandido. Apesar do modelo ter apresentado boa performance quando testado contra os dados no período de tempo original, teve um desempenho próximo de aleatório quando testado nos dados novos. Os resultados desse teste estão na Tabela 13.

No segundo teste, treinamos e testamos nos dados expandidos (180 dias) modelos utilizando o método da seção 4.2. Após o treinamento, os modelos foram testados contra os 90 dias de dados originais. O desempenho dos modelos nesse teste foi ligeiramente melhor que no teste anterior e está descrito na Tabela 14.

Esse experimento indica uma dificuldade de generalizar modelos treinadas e com boa performance em um período de tempo para outros períodos. Apesar dessa dificuldade, pode ser possível colocar modelos com essas limitações em produção utilizando uma janela deslizante de dados e uma política de constante re-treinamento do modelo baseado nessa janela.

Tabela 13 – Métricas obtidas aplicando modelos treinados no Experimento 1 à 180 dias de dados do Mercado

	<i>Precision</i>	<i>Accuracy</i>	<i>Recall</i>	<i>F₁</i>
PMC	0,50	0,45	0,29	0,36
Random Forest	0,56	0,51	0,50	0,53
SVM	0,56	0,56	0,96	0,71

Tabela 14 – Métricas obtidas aplicando modelos treinados com 180 dias de dados do Mercado aos dados de Mercado do Experimento 1

	<i>Precision</i>	<i>Accuracy</i>	<i>Recall</i>	<i>F₁</i>
PMC	0,55	0,52	0,63	0,59
Random Forest	0,57	0,54	0,65	0,60
SVM	0,54	0,51	0,61	0,57

5 Conclusão

Neste trabalho, mostramos com sucesso que é possível criar modelos de preditores da direção do preço do Bitcoin utilizando Redes Neurais, Random Forests e SVMs tendo como fonte de dados tanto o Twitter quanto métricas de mercado. Utilizando um Perceptron de Múltiplas camadas, obtivemos uma precisão de 84% tendo como fonte de dados apenas dados do Twitter, que superou a precisão de 72% da PMC treinada utilizando dados do Mercado. Os modelos usando Random Forests e SVMs que serviriam apenas para comparação, também mostraram resultados acima do esperado utilizando dados do Twitter, sendo que a Random Forest alcançou 76% de precisão e a SVM 75% de precisão.

Quando comparado com o trabalho de (VALENCIA; GÓMEZ-ESPINOSA; VALDÉS-AGUIRRE, 2019), nosso trabalho observou várias diferenças e pode oferecer diversas contribuições para o assunto. A primeira grande diferença foi que nenhum modelo treinado no trabalho anterior foi capaz de obter precisão ou acurácia maior que 50% utilizando apenas dados do Twitter. Como mostramos na Tabela 3, nosso trabalho obteve resultados significativamente melhores que um classificador aleatório utilizando rigorosamente o mesmo método do trabalho original. A razão para diferença observada com os dados do Twitter é objeto de estudo para um outro trabalho, mas podemos especular que a qualidade dos tweets aumentou pois a quantidade de pessoas falando sobre o assunto diminuiu. Para efeitos de comparação, no período de 16 de Fevereiro de 2018 até 21 de Abril de 2018, foram registrados 13.096.598 tweets relacionados a Bitcoin. No período de 90 dias em que realizamos o presente trabalho registramos apenas 3.736.807 tweets.

Fomos também capazes de obter resultados positivos em uma janela de tempo com um volume muito menor de *tweets* e um mercado muito mais estável. Para efeitos de comparação, o preço do Bitcoin na janela estudada por (VALENCIA; GÓMEZ-ESPINOSA; VALDÉS-AGUIRRE, 2019) chegou variar mais de R\$ 20.000 reais e o preço tinha uma tendência clara de queda. Na janela de tempo que estudamos, a diferença entre o preço máximo e o preço mínimo foi de apenas R\$ 3.279 reais, sem existir uma tendência clara de mercado. Esses resultados são um grande passo na direção de poder usar esses modelos no mundo real, já que mostra que não precisamos de uma tendência clara no mercado para que os modelos funcionem.

Outro resultado interessante foi que com pequenas mudanças de configuração da PMC observamos a precisão do modelo treinado com dados do Twitter ir de 61% (Tabela 3) para 84% (Tabela 6) no modelo treinado com configurações ajustadas. Esse resultado mostra que é possível utilizar um modelo como este como suporte de tomadas de decisão de compra ou venda no mundo real.

O nosso experimento com um filtro de qualidade nos dados do Twitter não obteve os re-

sultados esperados e piorou as métricas de todos os modelos treinados com esse dado. Investigar a fundo a razão de tal piora é tema para um bom trabalho futuro.

Com os nossos testes cruzados de modelos, aprendemos que o período no qual o modelo foi treinado importa muito para a obtenção de bons resultados. Disso e dos resultados anteriores, concluímos que pode não ser fácil obter um modelo único que funcione bem em qualquer situação, mas que pode ser possível colocar esse modelo em produção utilizando dados de uma janela de tempo recente e deslizando.

Para trabalhos futuros, seria interessante testar se é possível melhorar os bons resultados obtidos com uma RNN (*Recurrent Neural Network*), como a LSTM (*Long Short-Term Memory*) que já foi usada em trabalhos bem sucedidos no mercado financeiro. É importante também notar que avaliar modelos de previsão de mercado em termos apenas de suas métricas (acurácia, precisão, etc) distancia um pouco essa avaliação do mundo real. Isso porque existe ainda uma grande distância entre a obtenção de modelos com boas métricas e o sucesso em empregar esses modelos para obter lucros no mercado. Um trabalho que treine e teste modelos em função de estratégias de *trading* reais seria um ótimo próximo passo.

Referências

- ABRAHAM, J.; HIGDON, D.; NELSON, J. Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review*, v. 1, n. 3, 2018. Citado na página 19.
- AGUIAR, E. J. de et al. Análise de sentimento em redes sociais para a língua portuguesa utilizando algoritmos de classificação. In: SBC. *Anais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. [S.l.], 2018. Citado na página 16.
- AHMAD, M. et al. Machine Learning Techniques for Sentiment Analysis: A Review. *Int. J. Multidiscip. Sci. Eng*, v. 8, n. 3, p. 27–32, 2017. Citado na página 18.
- AKHTAR, M. S. et al. A Multilayer perceptron based ensemble technique for fine-grained financial sentiment analysis. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, p. 540–546, 2017. Citado na página 18.
- ANTONPOULOS, A. M. *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2014. ISBN 1449374042, 9781449374044. Citado na página 13.
- ARAÚJO, M. L. D. et al. iFeel 2.0: A multilingual benchmarking system for sentence-level sentiment analysis. In: *Tenth International AAAI Conference on Web and Social Media*. [S.l.: s.n.], 2016. Citado na página 15.
- BLOOMBERG. *Traders automatizados aproveitam bitcoin para obter lucro rápido - Notícias - UOL Economia*. 2017. Disponível em: <<https://economia.uol.com.br/noticias/bloomberg/2017/01/17/traders-automatizados-aproveitam-bitcoin-para-obter-lucro-rapido.htm>>. Citado na página 10.
- BRADLEY, M. M. et al. *Affective Norms for English Words (ANEW): Instruction manual and affective ratings*. 1999. Citado na página 15.
- CABRAL, R. *Tudo sobre o Bitcoin: a história, os usos e a política por trás da moeda forte digital*. 2013. Disponível em: <<https://gizmodo.uol.com.br/tudo-sobre-o-bitcoin/>>. Citado na página 13.
- CAVALCANTE, P. E. C.; BARBOSA, Y. d. A. M. Um dataset para análise de sentimentos na língua portuguesa. 2017. Citado na página 16.
- CÉSAR, P. *Análise de Sentimentos com Machine Learning*. 2019. Disponível em: <<https://www.datageeks.com.br/analise-de-sentimentos/>>. Citado na página 14.
- CHEKIMA, K.; ALFRED, R. Non-English Sentiment Dictionary Construction. *Advanced Science Letters*, v. 24, p. 1416–1420, 2018. Citado na página 16.
- COINMARKETCAP. *Cryptocurrency Market Capitalizations*. 2018. Disponível em: <<https://coinmarketcap.com/>>. Citado 2 vezes nas páginas 9 e 10.
- DA SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. *REDES NEURAIS ARTIFICIAIS PARA ENGENHARIA E: CIENCIAS APLICADAS - CURSO PRATICO*. [S.l.]: ARTLIBER, 2010. ISBN 9788588098534. Citado na página 17.

- DAS, S.; CHEN, M. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In: BANGKOK, THAILAND. *Proceedings of the Asia Pacific finance association annual conference (APFA)*. [S.l.], 2001. v. 35, p. 43. Citado na página 14.
- DODSWORTH, I. *TweetDeck*. 2008. Disponível em: <<https://tweetdeck.twitter.com/>>. Citado na página 23.
- FELICIANO, P.; FARAH, P. R. Avaliação de Algoritmos de Análise de Sentimentos em Tweets no Domínio da Copa do Mundo FIFA 2018. n. October, 2018. Citado na página 15.
- FRANCO, M. *Venezuelanos investem em bitcoin para encarar desemprego, hiperinflação e falta de notas | Mundo | G1*. 2017. Disponível em: <<https://g1.globo.com/mundo/noticia/venezuelanos-investem-em-bitcoin-para-encara-desemprego-hiperinflacao-e-falta-de-notas.ghtml>>. Citado na página 9.
- FRANCO, P. *Understanding Bitcoin: Cryptography, Engineering and Economics*. [S.l.]: Wiley, 2014. ISBN 9781119019169. Citado 2 vezes nas páginas 13 e 14.
- FUNG, P. M. Y. et al. Bitcoin Trading Strategies Using News and Tweets with Sentiment Analysis. 2019. Citado 3 vezes nas páginas 19, 20 e 23.
- GARDNER, M. W.; DORLING, S. R. Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences. *Atmospheric Environment*, v. 32, n. 14-15, p. 2627–2636, 1998. ISSN 13522310. Citado na página 18.
- HUANG, W.; NAKAMORI, Y.; WANG, S. Forecasting stock market movement direction with support vector machine. *Computers & OR*, v. 32, p. 2513–2522, 2005. Citado na página 10.
- HUTTO, C. J.; GILBERT, E. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. In: *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*. [S.l.: s.n.], 2015. Citado 5 vezes nas páginas 6, 15, 16, 24 e 25.
- JAIN, A. et al. Forecasting Price of Cryptocurrencies Using Tweets Sentiment Analysis. *2018 11th International Conference on Contemporary Computing, IC3 2018*, p. 2–4, 2018. Citado na página 19.
- KAMINSKI, J. C. Nowcasting the Bitcoin Market with Twitter Signals. n. September 2014, p. 1–16, 2013. Citado na página 10.
- KANSAON, D. P.; BRANDAO, M. A.; DE PAULA PINTO, S. A. Análise de Sentimentos em Tweets em Português Brasileiro. In: SBC. [S.l.], 2018. Citado na página 16.
- KHAIDEM, L.; SAHA, S.; DEY, S. R. Predicting the direction of stock market prices using random forest. v. 00, n. 00, p. 1–20, 2016. Disponível em: <<http://arxiv.org/abs/1605.00003>>. Citado na página 10.
- KIMOTO, T. et al. Stock market prediction system with modular neural networks. In: *1990 IJCNN International Joint Conference on Neural Networks*. [S.l.: s.n.], 1990. p. 1–6 vol.1. ISSN null. Citado na página 10.
- KIRILENKO, A. P.; STEPCHENKOVA, S. O. Sochi 2014 Olympics on Twitter: Perspectives of hosts and guests. *Tourism Management*, Elsevier Ltd, v. 63, p. 54–65, 2017. ISSN 02615177. Citado na página 15.

- KRISTOUFEK, L. What are the main drivers of the bitcoin price? Evidence from wavelet coherence analysis. *PLoS ONE*, v. 10, n. 4, p. 1–15, 2015. ISSN 19326203. Citado na página 10.
- KROLL, J. a.; DAVEY, I. C.; FELTEN, E. W. The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries. *The Twelfth Workshop on the Economics of Information Security (WEIS 2013)*, n. Weis, p. 1–21, 2013. ISSN 15437221. Citado na página 10.
- KUMAR, A.; JAISWAL, A. Empirical study of twitter and tumblr for sentiment analysis using soft computing techniques. *Lecture Notes in Engineering and Computer Science*, v. 1, p. 472–476, 2017. ISSN 20780958. Citado na página 18.
- LIU, B. *Sentiment Analysis and Opinion Mining*. [S.l.]: Morgan & Claypool Publishers, 2012. ISBN 1608458849, 9781608458844. Citado na página 14.
- MAI, F. et al. How Does Social Media Impact Bitcoin Value? A Test of the Silent Majority Hypothesis. *Journal of Management Information Systems*, Routledge, v. 35, n. 1, p. 19–52, 2018. Disponível em: <<https://doi.org/10.1080/07421222.2018.1440774>>. Citado na página 10.
- MAMGAIN, N. et al. Sentiment analysis of top colleges in India using Twitter data. *2016 International Conference on Computational Techniques in Information and Communication Technologies, ICCTICT 2016 - Proceedings*, p. 525–530, 2016. Citado na página 18.
- MARKETCAP. *Bitcoin price, charts, market cap, and other metrics*. 2018. Disponível em: <<https://coinmarketcap.com/currencies/bitcoin/>>. Citado 2 vezes nas páginas 5 e 10.
- MORAES, S. M. W.; MANSSOUR, I. H.; SILVEIRA, M. S. 7x1 PT : um Corpus extraído do Twitter para Análise de Sentimentos em Língua Portuguesa. p. 21–25, 2015. Citado na página 16.
- NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *www.bitcoin.org*, p. 9, 2008. ISSN 09254560. Disponível em: <<https://bitcoin.org/bitcoin.pdf>>. Citado 2 vezes nas páginas 9 e 13.
- NIELSON, M. Neural Networks and Deep Learning. *Artificial Intelligence*, p. 389–411, 2018. Citado 2 vezes nas páginas 5 e 17.
- PAK, A.; PAROUBEK, P. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In: *Proceedings of LREC*. [S.l.: s.n.], 2010. v. 10. Citado na página 14.
- PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up? Sentiment Classification using Machine Learning Techniques. 2002. Citado na página 14.
- RAMADHANI, A. M.; GOO, H. S. Twitter sentiment analysis using deep learning methods. *Proceedings - 2017 7th International Annual Engineering Seminar, InAES 2017*, 2017. Citado na página 18.
- RAMTEKE, J. et al. Election result prediction using Twitter sentiment analysis. In: *IEEE. 2016 international conference on inventive computation technologies (ICICT)*. [S.l.], 2016. v. 1, p. 1–5. Citado na página 15.

- RANI, S. et al. Evaluation of stemming techniques for text classification. *International Journal of Computer Science and Mobile Computing*, v. 4, n. 3, p. 165–171, 2015. Citado na página 26.
- RIBEIRO, F. N. et al. SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, Ribeiro et al., v. 5, n. 1, p. 1–29, 2016. ISSN 21931127. Citado na página 15.
- SHAH, D.; ZHANG, K. Bayesian regression and Bitcoin. *2014 52nd Annual Allerton Conference on Communication, Control, and Computing, Allerton 2014*, v. 2, n. 1, p. 409–414, 2014. ISSN 1387-6694. Citado na página 10.
- SI, J. et al. Exploiting Topic based Twitter Sentiment for Stock Prediction. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, 2013. p. 24–29. Disponível em: <<https://www.aclweb.org/anthology/P13-2005>>. Citado na página 10.
- SMEDT, T. D.; DAELEMANS, W. Pattern for python. *Journal of Machine Learning Research*, v. 13, n. Jun, p. 2063–2067, 2012. Citado na página 15.
- SOCHER, R. et al. Recursive deep models for semantic compositionality over a sentiment treebank. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, p. 1631–1642, 2013. Citado na página 15.
- SOUZA, K. F. de; PEREIRA, M. H. R.; DALIP, D. H. Unilex: Método léxico para análise de sentimentos textuais sobre conteúdo de tweets em português brasileiro. *Abakós*, v. 5, n. 2, p. 79–96, 2017. Citado na página 16.
- STATISTA. *Twitter: number of active users 2010-2019* | Statista. 2019. Disponível em: <<https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/>>. Citado na página 14.
- STONE, P. J.; DUNPHY, D. C.; SMITH, M. S. The general inquirer: A computer approach to content analysis. MIT press, 1966. Citado na página 15.
- SUROWIECKI, J. *The Wisdom of Crowds*. [S.l.]: Anchor, 2005. ISBN 0385721706. Citado na página 16.
- TAUSCZIK, Y. R.; PENNEBAKER, J. W. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, v. 29, n. 1, p. 24–54, 2010. ISSN 0261927X. Citado na página 15.
- THELWALL, M. et al. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, Wiley Online Library, v. 61, n. 12, p. 2544–2558, 2010. Citado na página 15.
- TUMASJAN, A. et al. Predicting elections with twitter: What 140 characters reveal about political sentiment. In: *Fourth international AAAI conference on weblogs and social media*. [S.l.: s.n.], 2010. Citado na página 15.
- TURNEY, P. D. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. p. 8, 2002. Citado na página 14.

ULRICH, F. *Bitcoin: A moeda na era digital*. [S.l.]: LVM Editora, 2017. ISBN 9788581190761. Citado na página 13.

VALENCIA, F.; GÓMEZ-ESPINOSA, A.; VALDÉS-AGUIRRE, B. Price movement prediction of cryptocurrencies using sentiment analysis and machine learning. *Entropy*, v. 21, n. 6, 2019. ISSN 10994300. Citado 9 vezes nas páginas 11, 19, 20, 24, 27, 28, 30, 32 e 37.

VIJAYARANI, S. et al. Preprocessing Techniques for Text Mining -An Overview. v. 5, n. 1, p. 7–16, 2015. Citado na página 26.

WILSON, T. et al. Opinionfinder: A system for subjectivity analysis. *HLT/EMNLP 2005 - Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, p. 34–35, 2005. Citado na página 15.

YI, J. et al. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. *Proceedings - IEEE International Conference on Data Mining, ICDM*, p. 427–434, 2003. ISSN 15504786. Citado na página 14.

ZANNI, M. *No comércio, bitcoin ainda serve apenas para marketing - 18/01/2015 - Mercado - Folha de S.Paulo*. 2015. Disponível em: <<https://www1.folha.uol.com.br/mercado/2015/01/1576004-no-comercio-bitcoin-ainda-serve-apenas-para-marketing.shtml>>. Citado na página 9.

ZIMBRA, D. et al. The state-of-the-art in twitter sentiment analysis: A review and benchmark evaluation. *ACM Transactions on Management Information Systems*, v. 9, n. 2, 2018. ISSN 21586578. Citado na página 15.