TRABALHO DE CONCLUSÃO DE CURSO

Clayton Raposo Veras e Lucas Mauro de Souza

Predição de preço de ações através de portais de notícias

Florianópolis
2018

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
SISTEMAS DE INFORMAÇÃO

Clayton Raposo Veras e Lucas Mauro de Souza

Predição do preço de ações através de portais de notícias

Trabalho Conclusão do Curso de Graduação em Sistemas de Informação do departamento de Informática e estatística da Universidade Federal de Santa Catarina como requisito para a obtenção do Título de Bacharel em Sistemas de informação
Orientador: Prof. Dr. Elder Rizzon Santos

Florianópolis
2018

Clayton Raposo Veras e Lucas Mauro de Souza

Predição do preço de ações através de portais de notícias

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador:

_____

Prof. Dr. Elder Rizzon Santos

Banca examinadora:

_____

Prof. Dr. Maicon Rafael Zatelli

_____

Prof. Dr. Mauro Roisenberg

Florianópolis - SC

2019/1

Resumo

A tecnologia avança a cada dia, e quanto mais ferramentas, aplicações e redes de informação são desenvolvidas, estes tipos de sistemas de informação vêm facilitando cada vez mais a divulgação de informação através da internet, torna-se cada vez mais comum o compartilhamento de informações dos mais variados tipos, desde registros de momentos pessoais, até informações e notícias sobre empresas e lançamento de produtos.

Baseado nessa premissa do compartilhamento de informações através da internet, este trabalho tem como principal objetivo identificar se existe alguma correlação entre notícias divulgadas na internet por portais de informação e a flutuação do valor de ações de empresas na bolsa de valores brasileira, a B3.

Utilizando técnicas e algoritmos de processamento de linguagem natural para tentar transformar o conteúdo de uma notícia para que este possa ser classificado em um formato compreensível pela máquina, aliados a algoritmos de predição que tentam identificar padrões e correlações entre diferentes dados, este trabalho tem como um de seus objetivos específicos criar um modelo de predição que correlacione informações relativas às transações de ações de uma empresa com as notícias divulgadas durante um dia, sendo estas notícias referentes às empresas específicas em estudo. Com estas informações se procura predizer se a tendência do valor de ação é de subida, queda ou estagnação.

Este trabalho utilizou grandes empresas de tecnologia como alvo do estudo, uma vez que por serem famosas e internacionais, é esperado um maior fluxo de notícias relacionadas a elas. Outro fator é o de estarem presentes na bolsa de valores brasileira com uma maior movimentação de ações. Serão estudadas ações e notícias de Apple, Microsoft e Tesla Motors.

Palavras-chave: PLN, Inteligência Artificial, Análise de Sentimento, Ações, Notícias, Bolsa de valores, Apple, Microsoft, Tesla Motors, Predição.

Abstract

Technology becomes more and more advanced each day. As more tools, applications and information networks are created, this kind of information systems provides and facilitates the spread of information throughout the internet. It becomes more common to share all types of information, from important personal moments to companies and products news.

Based on this premise of the popularization of information share over the internet, this paper has the main goal of identifying whether there is any correlation between news shared by communication portals and the fluctuation of the companies stock prices, on the Brazilian stock exchange, the B3.

Using natural language processing algorithms and techniques to try to transform the news content and classify this information into a format that computers can understand and process, allied to prediction algorithms that try to identify patterns and correlations between the stocks transaction data of a company, this paper has as a specific goal of creating a prediction model that correlates a company's stock transactions information with the company's news shared on internet communication portals of a same day. With this information, this study tries to predict whether the tendency of a stock price is to go up, down or maintain on the same level.

This paper studies big technology companies. Since they are international, world wide known companies, it is expected that online portals should publish a greater number of news about them than barely known companies. Another reason for choosing these companies is the fact that they are on the Brazilian stock exchange with a considerable stock move amount. The stocks and news of Apple, Microsoft and Tesla Motors are the ones to be studied.

Keywords: NLP, Artificial Intelligence, Sentiment Analysis, Stocks, News, Stock Exchange, Apple, Microsoft, Tesla Motors, Prediction.

Lista de figuras

Lista de tabelas

Lista de abreviaturas e siglas

IA …… Inteligência Artificial
PLN …….. Processamento de linguagem natural
RSS ……..…… (Originalmente, RDF Site Summary)
API ……..... Application Programming interface
JSON …….. JavaScript Object Notation
SMRF-TM ……. Stock Market Random Forest - Text Mining
TF-IDF …… Term frequency – Inverse document frequency
NLTK ……. Natural Language Toolkit
CRUD ……. Create, Read, Update, Delete
SGBD ……..... Sistema gerenciador de banco de dados
URL ……..... Uniform resource locator
MACD …….. Moving average convergence/divergence
RSI …… Relative strength index
CCI …….. Commodity Channel Index
POS ……..... Part-of-Speech
PETR4 ……. Sigla para Petroleo Brasileiro S.A. na BVMF
MM21 ……..... Médias móveis de exponencial 21
IFR ……..…… Índice de força relativa
NARX …….. Nonlinear autoregressive exogenous model
HTTP …….. Hypertext transfer protocol
HTML ……... Hypertext markup language
SVD ……... Singular value decomposition
ReLU …… Rectified Linear Unit
QP ……..... Quantidade de notícias positivas
QN ……..……... Quantidade de notícias negativas
QNE ……. Quantidade de notícias neutras
REST ……. Representation State Transfer

Sumário

1 Introdução

A computação de dados vem se tornando mais popular a cada dia, sendo de grande importância no processo de tornar a população mundial mais atenta a relações que até então eram praticamente impossíveis de serem analisadas devido à sua complexidade. O ramo estatístico da matemática já estuda estas relações complexas há bastante tempo, mas os avanços tecnológicos da informática vêm tornando os estudos mais assertivos e processáveis, de forma automática.

A Inteligência Artificial (IA) é um exemplo destes avanços. Por ser uma disciplina jovem, com sua estrutura, considerações e métodos não definidos tão claramente quanto aqueles de uma ciência mais madura, como a física, existe uma dificuldade em se chegar a uma definição exata para o termo IA, segundo Luger (2013). Apesar de ser considerada complexa, Luger (2013) apresenta uma das possíveis definições para essa área de estudo como um ramo da ciência da computação que se dedica a automação de comportamento inteligente.

Um dos problemas para se definir a inteligência artificial, segundo Luger (2013), vem do fato de que a própria inteligência não é muito bem definida nem compreendida. Embora a maioria das pessoas esteja certa de que reconhece o comportamento inteligente quando o vê, não é certo que alguém possa chegar perto de definir a inteligência de um modo que seria específico o suficiente para ajudar na avaliação de um programa de computador supostamente inteligente.

Este trabalho não visa definir o que é a inteligência, nem o que é inteligência no meio computacional. Por isso, é apresentado um dos possíveis conceitos do que seria IA. O estudo sobre a definição do que a inteligência representa no mundo computacional pode ser aprofundado em estudos mais amplos. Como apresenta Lustosa (2004), a inteligência artificial tenta entender o comportamento de entidades inteligentes, porém, ao contrário da filosofia e da psicologia, que estão mais preocupadas como o estudo da inteligência dentro de um contexto de relações humanas, a IA foca em como essas entidades podem ser criadas e utilizadas para determinados fins.

Um dos ramos da IA que vem sendo bastante utilizado para criar e entender a relação entre fenômenos, até então não estudados devido à sua complexidade, é o processamento de linguagem natural (PLN), que, segundo Chowdhury (2005), "é uma área de pesquisa e aplicação que explora como computadores podem ser usados para entender e manipular linguagem natural, seja em forma de texto ou fala, para realizar determinadas atividades."

Segundo Luger (2013), devido à enorme quantidade de conhecimento necessária para a compreensão de linguagem natural, a maioria dos trabalhos é realizada em áreas de problemas especializadas e bem conhecidas,visto que a compreensão de linguagem requer inferências sobre objetivo, conhecimento e suposições do locutor, bem como sobre o contexto da interação.

Todas linguagens têm suas características específicas, como símbolos e regras gramaticais, que demandam diferentes medidas para se realizar operações com elas.

Entretanto, para serem processadas por um computador, precisam ser tratadas com formalismo e determinismo. Em geral, a pesquisa em PLN tem avançado significativamente para o idioma inglês, mas para o português não há disponível a mesma quantia ou profundidade de material. Sendo assim, ao se analisar textos de cunho econômico, além de tratar o escopo de linguagem econômica, também é necessário considerar o idioma em que foi escrito.

Estes estudos e toda a tecnologia que vem sendo desenvolvida nas últimas décadas, nos apresenta novas possibilidades com relação a problemas já conhecidos, mas ainda sem uma definição concreta. Um destes problemas, que será abordado neste trabalho, diz respeito a identificar como a divulgação de notícias pelas mídias mais tradicionais, como por exemplo os jornais, podem influenciar empresas e o mercado de ações como um todo.

O mercado de ações e seus comportamentos são alvos frequentes de estudos em áreas como IA e estatística, visto que é um meio que não depende apenas de dados para explicar suas quedas e altas, a presença de fatores humanos para os quais não se dispõe de modelos matemáticos é um dos maiores problemas a serem tratados nesta área, até mais do que a quantidade massiva de dados a serem processados. Um exemplo disso é que, em geral, apesar dos esforços empreendidos pelas empresas visando aumentar seu valor, este pode sofrer impactos positivos ou negativos devido a decisões aparentemente banais, como escolher uma opção de investimento. A opção de investimento, por sua vez, está conectada a diversos outros fatores que influencia de forma positiva ou negativa sua rentabilidade.

Todas estas conexões entre os aspectos que estão relacionados a uma simples decisão, como escolher uma opção de investimento, torna as análises complexas demais para serem feitas sem a ajuda de um programa computacional, tornando difícil identificar, de forma rápida, a probabilidade desse tipo de decisão gerar o esperado aumento de valor almejado pela empresa. Como cita Procianoy & Antunes (2001). Mesmo existindo o consenso quase unânime quanto ao objetivo da empresa gerar valor aos proprietários, ainda permanece o interesse em avaliar e conhecer os efeitos das decisões de investimento sobre o valor das ações.

Como afirma Neto (2007), toda nova informação relevante trazida ao mercado tido como eficiente tem o poder de promover alterações nos valores dos ativos negociados, modificando seus livres preços de negociação e resultados de análises. Estas informações podem causar os mais diversos sentimentos às pessoas envolvidas com as empresas, sejam estas pessoas físicas ou jurídicas, que compraram cotas, a empresa que emite os títulos ou até mesmo os consumidores de produtos e ou serviços destas.

Segundo Fama (1991), o preço da ação no mercado de capitais eficiente é ajustado no exato momento em que informações relevantes (que afetam o fluxo de caixa futuro da empresa) tornam-se publicamente disponíveis. Além da disponibilidade das informações e dos sentimentos causados nas pessoas envolvidas com as empresas, diversos outros motivos podem causar flutuação no preço das ações, como alta procura ou venda dos títulos, baixo número de vendas de produtos ou contratações de serviços, etc.

Um exemplo prático de uma das possíveis formas de predizer as flutuações no mercado financeiro é a partir da análise de sentimentos de publicações em mídias sociais, através do uso de PLN. Segundo Pagolu (2016), “existe uma forte correlação entre subida e

queda em preços de ações de uma empresa em relação às opiniões ou emoções públicas em relação expressadas no Twitter sobre ela."

Embora existam diversas formas de avaliar a flutuação no mercado financeiro, neste trabalho propomos a relação entre notícias sobre um número determinado de empresas em portais de notícia com maior credibilidade no Brasil, jornais que em geral a população confia, com o intuito de, através de técnicas de inteligência artificial, identificar acontecimentos e demonstrar que estes têm relação direta com a flutuação nos valores das ações destas empresas. Desta forma, com um modelo preditivo, será possível ter um modelo que receba notícias automaticamente e determine a probabilidade de acréscimo ou decréscimo no valor da ação da empresa noticiada.

1.1 Objetivos

1.1.1 Objetivo Geral

Este trabalho de conclusão de curso tem como objetivo o desenvolvimento de um modelo preditivo, que irá correlacionar dados históricos dos valores de ações de empresas multinacionais de tecnologia com notícias relacionadas a elas. Este modelo será capaz de predizer a flutuação positiva ou negativa do valor de suas ações, de acordo com os acontecimentos diversos divulgados em portais de notícias.

1.1.2 Objetivos Específicos

Para o desenvolvimento do modelo, aquisição de notícias e correlação com os valores das ações, é necessário um estudo mais aprofundado sobre o estado da arte, ferramentas e técnicas disponíveis. Serão apresentados abaixo alguns dos objetivos específicos para a realização do modelo proposto:

1. Realizar pesquisa teórica sobre o estado da arte em extração e transformação de dados para embasamento teórico no assunto;
2. Realizar pesquisa teórica sobre previsão do mercado de ações para embasamento teórico relativo à volatilidade de ações;
3. Identificar algoritmos e técnicas de PLN disponíveis atualmente que são relevantes para o projeto;
4. Analisar algoritmos de aprendizado de máquina e data science no contexto de PLN;
5. Criar um modelo de predição para identificar flutuações nos valores de ações de empresas selecionadas a partir de notícias publicadas na web;
6. Implementar um protótipo de modelo preditivo capaz de associar notícias com flutuações de valores da bolsas;
7. Comparar a funcionalidade do modelo criado com notícias reais publicadas;
8. Comparar a funcionalidade do modelo criado à performance dos modelos dos trabalhos relacionados.

1.2 Escopo

O escopo do projeto se define no estudo de técnicas de processamento de linguagem natural, aplicadas ao setor da economia. Serão estudados históricos das empresas Apple, Microsoft e Tesla, indexadas no índice BOVESPA, visto que notícias sobre empresas de grande porte tendem a alcançar um número elevado de leitores, além de lançar produtos e serviços com maior frequência.

1.3 Premissas

O foco deste trabalho é de determinar a avaliação de relação entre notícias sobre empresas e o preço de suas ações na bolsa de valores. Para tanto, bibliotecas externas serão utilizadas para auxiliar neste processo, em especial na execução de tarefas que, embora dêem subsídio ao trabalho, não pertencem ao conjunto de objetivos deste, como obtenção de histórico de notícias e valores destas empresas. É considerado que os objetivos serão concluídos até o fim do cronograma estabelecido. Além disso, tem-se como verdade que a maior parte dos conhecimentos necessários para este trabalho será adquirida durante a execução dele.

1.4 Restrições

O código desenvolvido para esta solução estará disponível e poderá ser utilizado de forma livre e gratuita. A pesquisa se restringe a notícias no idioma português, publicados num número restrito de portais de notícias, sobre um conjunto seleto de empresas.

1.5 Método

Como ponto de partida, realizamos uma pesquisa bibliográfica, visto que pontos de variadas áreas de conhecimento são abordados neste trabalho. Foram feitas buscas sobre economia e inteligência artificial, com o intuito de conhecer o estado da arte destas áreas. Atenção especial foi dada ao funcionamento de bolsas de valores, técnicas de processamento de linguagem natural (PLN) e formas já conhecidas para se relacionar estes dois temas.

Além disso, foram buscadas ferramentas de software que auxiliem no processo de captura de notícias históricas em portais definidos sobre as empresas em pesquisa. Para se obter notícias históricas, foi definida a API de pesquisa do Google, já que possibilita ampla customização em seus parâmetros de busca, esta API será utilizada no processo de validação do algoritmo, uma vez que é necessária a busca de notícias em diferentes datas e o formato de resposta da API (JSON) facilita a manipulação e transformação das notícias, essa ferramenta foi utilizada na busca de notícias para validação do modelo, descrita na seção 5.4. Já para efetuar a captura diária de notícias, foi escolhido o Miniflux. Esta ferramenta captura e armazena notícias via RSS, trazendo as notícias e os metadados associados a elas.

Para a busca de notícias históricas e diárias, foram selecionadas algumas categorias que possibilitam uma grande variação de temas relacionadas às empresas. Desta forma,

selecionamos aqueles que contém todas as categorias definidas, possibilitando uma maior extração de dados.

Os portais selecionados têm grande fluxo de notícias diárias, além de apresentarem um serviço de RSS bem estruturado. Eles trazem classificação das notícias por categorias, sendo as seguintes selecionadas como importantes para este trabalho: política, economia, negócios e/ou empresas, empregos e/ou mercado de trabalho e tecnologia. Desta forma, a captura de notícias das empresas possui um leque variado de informações diárias disponíveis.

Para fazer a limpeza das notícias extraídas será utilizada a ferramenta NLTK. Nesta etapa serão removidos quaisquer termos que isolados de contexto não tragam significado. Todas as palavras serão padronizadas, desta forma, aquelas que são diferentes entre si, mas que carregam o mesmo significado básico, serão transformadas em um token único.

Para se obter os valores históricos e diários das ações indexadas na Bovespa, foi utilizada o serviço alphavantage.co. Com essa ferramenta, montou-se o histórico de valores das empresas, o qual associamos às datas das notícias processadas a fim de encontrar os padrões de flutuação conforme o conteúdo destas mesmas notícias.

O carregamento dos conjuntos de dados foi feita com as biblioteca Pandas e NumPy, que oferecem estruturas de dados como tabelas e vetores, além de operações de CRUD (do inglês, *Create, Read, Update, Delete*). Uma vez os dados carregados, serão aplicadas técnicas de PLN que possibilitem avaliar a correlação que as notícias têm com o preço de ações de empresas na bolsa de valores. Também serão avaliados quais tipos de notícias tendem a ser relevantes neste aspecto.

Foram criados scripts na linguagem Python para a utilização de todas as bibliotecas citadas, de forma que a coleta, transformação e processamento das informações fosse automatizadas. As notícias foram capturadas em uma máquina virtual com sistema operacional Ubuntu 16.10 que executa o servidor RSS chamado Miniflux.

Por fim, o modelo criado foi posto à prova com as notícias selecionadas em um intervalo de tempo aleatório e que estavam fora do conjunto de treinamento, com o intuito verificar a precisão em que predirá a flutuação dos valores conforme as notícias recebidas. Uma apresentação final também foi desenvolvida e apresentada, abordando os principais aspectos do trabalho e as conclusões alcançadas durante os experimentos.

2 Fundamentação Teórica

Neste capítulo são discutidos os conceitos fundamentais para o desenvolvimento deste trabalho, sem os quais não seria possível realizar o estudo. Eles servem de base para o desenvolvimento do sistema que será proposto no capítulo 4.

2.1 Processamento de Linguagem Natural

PLN é uma área de pesquisa e aplicação que explora como computadores podem entender e manipular texto ou fala em linguagem natural para realizar operações úteis, Chowdhury (2003). Está fundamentado em um número de disciplinas, como ciências da computação e informação, linguística, matemática, engenharia elétrica e eletrônica, inteligência artificial, robótica e psicologia. Suas áreas de aplicação incluem tradução de máquina, sumarização de textos em linguagem natural, reconhecimento de fala, entre outros.

2.1.1 Análise Textual

Comunicar-se por meio de linguagem natural, quer seja como texto ou como um ato de fala, depende enormemente das nossa habilidades na língua, como nosso conhecimento e expectativas dentro do domínio do discurso, Luger (2013). A compreensão de linguagem não é meramente a transmissão de palavras: ela também requer inferências sobre objetivo, conhecimento e suposições do locutor, bem como sobre o contexto da interação. A implementação de um programa para compreender linguagem natural requer que representemos conhecimento e expectativas do domínio e raciocinemos efetivamente sobre eles. Precisamos considerar questões como monotonicidade, revisão de crença, metáfora, planejamento, aprendizado e complexidades da prática da interação humana.

Segundo Luger (2013), não se pode simplesmente encadear os significados dicionarizados das palavras de um texto. Em vez disso, deve-se empregar um processo complexo de capturar padrões de palavras, analisar sentenças, construir uma representação do significado semântico e interpretar esse significado à luz do conhecimento do domínio do problema.

Há pelo menos três questões principais envolvidas na compreensão de uma linguagem, Luger (2013). Primeiro, presume-se uma grande quantidade de conhecimento humano. Os atos de linguagem descrevem relacionamentos em um mundo normalmente complexo. O conhecimento desses relacionamentos deve ser parte de qualquer sistema de compreensão de linguagem. Segundo, uma linguagem é baseada em padrões: fonemas são componentes de palavras e palavras constituem frases e sentenças. A ordenação de fonemas, palavras e sentenças não é aleatória. Não é possível haver comunicação sem uma grande restrição quanto ao uso desses componentes. Finalmente, os atos de linguagem são o produto de agentes, tanto de humanos quanto de um computador. Os agentes estão incorporados em um ambiente complexo com dimensões individual e

sociológica. Todos estes pontos levantados devem ser considerados ao se realizar análises textuais.

2.1.3 Análise Sintática

Diferentes tipos de análises requerem diferentes tipos de técnicas. Podemos considerar a análise sintática, que analisa a estrutura sintática de sentenças, verificando se elas são bem formadas e determina, também, uma estrutura linguística, Luger (2013). Este analisador define as principais relações linguísticas, como sujeito-verbo, verbo-objeto e substantivo-modificador, que formam o arcabouço para o analisador semântico, normalmente representado por uma árvore sintática.

Para a análise sintática, existem técnicas básicas, uma delas sendo a tokenização, que normalmente inicia o processo de análise textural. Nela são separados os termos que compõem o texto, como palavras e pontuações, para poder ser executado um processo individual para cada um deles (Cordeiro, 2017). De certa forma, a tokenização é um pré-processamento; uma identificação de unidades básicas a serem processadas e erros neste estágio induzem a mais erros nas fases seguintes da análise (TRIM, 2013).

Stemming é outra técnica básica que, como definido por Alvares (2005), é a "tarefa de identificar a subcadeia de uma palavra que sirva como uma representação única e não ambígua da mesma, e a de suas diversas variações". O resultado deste processo é chamado de stem e não é necessariamente o mesmo que o radical da palavra. (Cordeiro, 2017).

2.1.4 Análise Semântica

A interpretação semântica vem após a análise sintática e é a que produz uma representação do significado do texto. Esta análise se dá pelo conhecimento sobre o significado das palavras e a estrutura linguística, como papéis de substantivos ou a transitividade de verbos, Luger (2013). Assim, são identificados os agentes, objetos e instrumentos de uma sentença, por exemplo.

O analisador semântico também realiza verificações de consistência no que diz respeito às possibilidades de relações entre objetos, impedindo relações inválidas, Luger (2013). Um exemplo seria o verbo *pilotar* estar associado ao substantivo *estátua*.

2.1.5 Extração de Informação

O conceito de extração de informação é utilizado para extrair partes úteis de uma informação textual, segundo Chowdhury (2013), e normalmente faz uso das análises previamente mencionadas. Um grande número de técnicas é utilizado neste sentido e a extração de informação pode servir diversos propósitos, por exemplo: preparar um resumo de um texto, popular bancos de dados, preencher espaços vazios, ou, como exemplo de maior relevância para este trabalho, identificar palavras-chaves e informações dentro de frases. Isto significa que estas informações extraídas podem ser utilizadas em diferentes processos por outros sistemas, servindo como base para as mais variadas aplicações.

Chowdhury (2013) sugere que, embora diversas técnicas de extração de informação possam ser usadas com sucesso, revelar relações entre termos de um texto pode ser difícil. Não se pode utilizar um modelo único para se extrair informações de diferentes domínios. Isto se dá pelo fato de que diferentes estruturas, contextos e regras resultam em diferentes possibilidades de comunicação, que por sua vez requerem diferentes configurações de modelos.

2.2 Big Data

Um dos principais componentes necessários para se fazer a predição dos valores de ações das empresas, como é o objetivo deste trabalho, com certeza são dados. Serão utilizados dados das mais diversas fontes e formatos, dados de histórico de valores de ações, dados de notícias dos portais, dados sobre as empresas, etc. Todo esse conjunto de dados, será essencial para a criação de um modelo preditivo que nos permita transformar todos estes dados em informação e utilizá-la para fazer a predição dos valores de ações. Para isso, os conceitos de Big Data são alguns dos pilares deste trabalho, ao lado do processamento de linguagem natural e estatística.

Há alguns anos o mundo vem criando e armazenando uma grande quantidade de dados, gerando cada vez maiores e complexos conjuntos de dados espalhados mundo afora. Segundo Finlay (2014) estes grandes e complexos conjuntos de dados não são novidade para uma boa parte da indústria de tecnologia, visto que muitos destes conjuntos já existem há anos e por isso a noção de Big Data não é nova. Contudo, o conceito de Big Data, que se tornou uma expressão popular para classificar estes conjuntos de dados, extrapola esta classificação de conjuntos de dados grandes e passa a ser a classificação para conjuntos de dados enormes e complexos.

Mayer-Schönberger (2013) afirma não existir uma definição rigorosa do que pode-se considerar Big Data, mas que a ideia inicial foi a de que um conjunto de dados era classificado como Big Data, quando a tamanho do conjunto de dados sendo examinado superava o tamanho de memória dos computadores utilizados para o processamento de dados. O que gerou uma série de inovações tecnológicas mais tarde utilizadas para processar e explorar estes conjuntos de dados, como foi o caso do MapReduce e o Hadoop, desenvolvidos pelos engenheiros da Google. Já Wu et. al (2013) traz um conceito mais direto sobre o que seria o conceito de Big Data, descrevendo o conceito como um conjunto de dados de grande volume, complexos e em crescimento constante, com muitas fontes de dados autônomas.

Estes três conceitos apresentados resumem, em nosso contexto, o que é necessário para classificar o tipo de conjunto de dados necessário para se desenvolver o modelo preditivo que é foco deste trabalho. Serão necessárias diversas fontes, enormes quantidades de dados e também uma grande variedade de dados, para que possamos criar nossos conjuntos de dados necessários para treinar, testar e alimentar o modelo preditivo. Essa grande quantidade de dados levanta um grande ponto de atenção, que é o tempo de processamento, visto que a predição de valores de ações é algo que demanda um tempo de resposta relativamente curto. O modelo proposto não poderá ficar dias processando dados para realizar uma estimativa. Devido a este fato, será necessário utilizarmos alguns recursos de mineração de dados para agilizar o processamento dos dados.

2.3 Mineração de dados

Para analisar ou processar um enorme e complexo conjunto de dados é necessário um grande poder computacional, visto que se torna totalmente inviável que seres humanos façam esta análise, ou então um conjunto de ferramentas que minimizem o esforço de processamento, para que apenas grandes conjuntos de dados selecionados sejam utilizados e processados.

Finlay (2014) define Mineração de dados como um conjunto de técnicas automatizadas utilizadas para interrogar enormes bases de dados e fazer inferências sobre o que os dados significam. ARUMUGAM et. al (2010) traz uma definição mais simples sobre o termo mas que retrata a intenção por trás das técnicas. Para ele, o termo pode ser definido como fazer melhor uso dos dados. A ideia de se processar os grandes conjuntos de dados para não apenas reduzir o tamanho do conjunto processado, mas também para processar melhores dados, vem como uma grande ajuda ao se trabalhar com Big Data, principalmente quando falamos de diferentes fontes de dados.

Para Chen, Han e Yu (1996) Mineração de dados também pode ser referenciado como "*conhecimento descoberto em bases de dados"* e basicamente, significa um processo não trivial de extração de informação implícita, previamente desconhecida e potencialmente útil, dos dados guardados em bases de dados. Esta definição traz uma noção mais clara do que de fato significa mineração de dados, além de vir de encontro com o necessário para o desenvolvimento de um modelo preditivo baseado em parâmetros ainda não conhecidos, como é o caso das notícias relacionadas a empresas. Para utilizar notícias para prever possíveis variações de valor das ações, é necessário primeiro saber quais notícias de fato estão de fato relacionadas a uma empresa, para que depois disso seja possível passar por um algoritmo de PLN, que auxiliará na identificação de relevância da notícia. Algumas técnicas de Mineração de dados podem ser utilizadas para encontrar, como por exemplo a relação entre uma notícia e uma empresa específica. Esta seria uma das várias das etapas do processamento dos dados que serão utilizados na predição.

Um dos pontos mais importantes ao se trabalhar com mineração de dados é saber com o que se vai trabalhar, em quais bases os dados se encontram, que tipos de dados serão processados, além de se saber quais técnicas de mineração serão utilizadas. Neste trabalho, serão utilizadas diversos tipos diferentes de bases de dados, que incluem desde Sistemas de Gerenciamento de Bancos de Dados (SGBDs) relacionais, até arquivos de texto e planilhas. Os dados de notícias, por outro lado, serão de um único tipo: textual. Por conta dessa variedade de bases de dados e o tipo único de dado, algumas das técnicas mais comuns de mineração de dados podem ser aplicadas para diminuir o número de dados processados pelo modelo, como por exemplo, identificação de padrões, classificação de dados, associação de dados, regressão e predição. Como todas as técnicas de mineração de dados citadas fazem parte de outras áreas de estudo distintas, as mesmas serão descritas com mais detalhes nas próximas seções do capítulo.

2.4 Inteligência artificial

Grande parte deste trabalho está contido na área de estudo de inteligência artificial, englobando desde PLN até as técnicas de mineração de dados citadas na seção anterior. Esta área de pesquisa não é nenhuma novidade no mundo computacional. Gevarter (1984), por exemplo, define IA como uma área de pesquisa da ciência da computação que se destina a criar programas de computador capazes de resolver problemas que se realizados por seres humanos necessitam de inteligência dos mesmos. SHWARTZ (1987) define IA de forma mais sucinta como a capacidade de uma máquina de imitar comportamento humano inteligente. Em ambas as definições é possível ver que a ideia no geral é criar programas considerados inteligentes, de forma que se possa automatizar tarefas que até então apenas humanos poderiam executar.

O conceito de inteligência artificial se encaixa em diversos momentos deste trabalho, desde o processamento de texto, como é descrito na seção destinada a PLN e como foi introduzido na seção de mineração de dados. Nesta segunda, foram apresentadas algumas técnicas que são também áreas de estudo dentro da inteligência artificial e que agora serão melhor detalhadas.

2.4.1 Reconhecimento de padrões

Fukunaga (1990) apresenta a meta da identificação de padrões como tornar claros os processos de tomada de decisão feitos por humanos e automatizar essas funções utilizando computadores. Luger (2013) também traz uma breve apresentação da responsabilidade dos métodos de reconhecimento de padrões caracterizando-os como métodos para identificação de estruturas ou os padrões nos dados. Utilizando grandes conjuntos de dados como fonte de informação, um sistema de reconhecimento de padrões é capaz de identificar tanto os padrões quanto anomalias nos dados.

2.4.2 Classificação de informação

Luger (2013) apresenta os métodos de classificação como responsáveis por decidir a qual categoria ou grupo pertence um valor de entrada. Lippmann (1987) traz uma definição mais prática de como funcionam os métodos de classificação. Segundo o autor, um método de classificação, também conhecido como classificador, determina qual classe, de um conjunto de **m** classes, é a mais representativa para um padrão de entrada desconhecida contendo **n** elementos. Os métodos de classificação são bastante úteis no contexto deste trabalho, possibilitando por exemplo dividir as notícias em grupos, definindo se uma notícia é uma influenciadora positiva ou negativa para um empresa.

2.5 Estatística

A estatística e seus métodos são bastante importantes quando se tenta prever qualquer tipo de situação baseando-se apenas em dados. Várias das áreas de estudo computacional que hoje são bastante conhecidas e trabalhadas têm suas origens nos

modelos matemáticos desenvolvidos pela estatística. Alguns ramos da inteligência artificial, como predição de valores, por exemplo, têm uma ligação forte e direta com esse ramo da matemática.

Devore (2006) apresenta a estatística como um grande meio de ajudar os seres humanos a fazer julgamentos mais inteligentes e tomar decisões, diante de incertezas e variações na informação recebida. Bussab e Morettin (2002) apresentam uma das áreas da estatística, a estatística inferencial, que tem como objetivo coletar, reduzir, analisar e modelar os dados para fazer uma dedução sobre a população (todos os dados de uma base de dados, por exemplo) à qual alguns dados analisados pertencem, ressaltando a previsão de dados como uma das grandes tarefas da última etapa da inferência estatística e como importante meio de se tomar decisões.

Neste trabalho, a estatística se faz presente em todo tempo, indo desde a análise dos dados históricos dos valores de ações das empresas, até o funcionamento do modelo preditivo em si. Tanto a regressão dos dados, quanto a predição de novos dados, fazem parte do conjunto de técnicas e conceitos que será bastante abordado neste trabalho e por isto ambos os temas tem seu próprio parágrafo de apresentação e descrição.

2.5.1 Regressão

Para que a estatística inferencial seja possível, primeiramente é necessário que sejam feitos tratamentos estatísticos sobre os dados retirados de uma base de dados, sejam os dados como um todo ou apenas uma porção destes, como uma amostra. A partir dos resultados obtidos das operações estatísticas é possível fazer uma nova análise dos dados, e a partir disso tomar decisões ou provar conceitos. A regressão é uma das ferramentas utilizadas neste tratamento. Barbetta (2012) nos apresenta o termo regressão como sendo resultado dos trabalhos de Francis Galton no século XIX, sendo que suas pesquisas buscavam explicar a razão de filhos de pais com características excepcionais herdarem características dos pais, mesmo que em menor intensidade. Devore (2006) conceitua o objetivo da regressão como a exploração da relação entre duas ou mais variáveis, para que se possa conhecer as propriedades desconhecidas de uma propriedade através das outras variáveis conhecidas.

A inteligência artificial e a estatística são duas áreas de estudo distintas, mas possuem algumas fortes ligações. Um bom exemplo dessa ligação entre os dois ramos de estudo, é a regressão. Luger (2013) mostra essa conexão entre os dois mundos explicando o funcionamento dos métodos Bayesianos, visto que estes suportam a interpretação de novas experiências com base nos conhecimentos adquiridos anteriormente. Ter à disposição tanto os métodos estatísticos quanto computacionais como, por exemplo, as redes bayesianas, para entender melhor uma variável através de seu histórico, seja em relação a tempo ou hereditariedade, abre um grande leque de possibilidades para que seja possível fazer inferências a respeito de uma determinada variável. Essa ideia é totalmente aplicável à predição de valores de ações. Identificar como uma ação foi afetada por uma notícia de um determinado tipo no passado pode ser bastante útil para entender a influência de uma notícia semelhante sobre uma ação no presente.

2.5.2 Predição

Através da regressão, torna-se possível entender como uma determinada variável é afetada por outras variáveis. Além disso, torna-se possível também prever como uma nova variável será afetada pelas mesmas variáveis. A predição de valores pode ser explicada através do conceito da probabilidade condicional que Devore (2006) apresenta como a probabilidade condicional de um evento acontecer visto que um outro evento já tenha acontecido. No contexto de predição de flutuação de ações baseados em notícias, podemos dizer que é a probabilidade de uma notícia afetar o valor das ações de uma empresa dado que a uma outra notícia já tenha afetado o valor das ações.

Da mesma forma como com os modelos estatísticos, podemos fazer a predição de variáveis utilizando a inteligência artificial. As redes neurais trazem exatamente a ideia da predição estatística, utilizando diversas variáveis em um processamento para predizer uma nova variável. Luger (2013) traz uma definição para redes neurais apresentando-a como um modelo em camadas onde novas informações são geradas ou informações existentes são adaptadas através das conexões entre as camadas, onde as camadas anteriores têm relação com as camadas posteriores, criando a mesma ideia de informações ancestrais apresentada na definição da predição estatística.

3 Trabalhos Relacionados

Nesta seção são apresentados de forma sucinta os desenvolvimentos de quatro artigos e uma dissertação. Estes trabalhos utilizam técnicas de aprendizagem de máquina para resolver problemas envolvendo processamento de linguagem natural, análise de mercado, de sentimentos e dados financeiros, que são conceitos relacionados a este trabalho desenvolvido.

3.1 Analysis of stock market using text mining and natural language processing

Fazer investimentos no mercado de ações é arriscado, pois por vezes é a maneira mais rápida tanto de se fazer quanto de se perder dinheiro. A análise deste mercado pode ser separado em duas categorias: técnica e fundamentalista. A primeira consiste em observar as tendências de preços para fazer uma predição, enquanto a segunda se volta a fatores econômicos. Embora ambas as abordagens sejam importantes, muitos *traders* acreditam que todas as informações necessárias podem ser encontradas dentro dos gráficos, pois eles seriam as próprias traduções do estado em que empresa se encontra. Todavia, para entender a performance de uma ação a longo prazo é necessário considerar outras técnicas, como fazer análise de notícias e rumores sobre as empresas, segundo Abdullah, et al. (2013).

O desafio que surge neste trabalho é de lidar com textos que podem ser escritos em diferentes formatos e oriundos de diferentes fontes. Para resolver este problema, foi proposto um *framework* que captura dados de diversas fontes, categoriza-os e extrai informações relacionadas para auxiliar na tomada de decisão.

Ao se avaliar as ações de uma empresa, deve-se considerar diversas questões envolvendo a segurança do investimento. Entretanto, o número de questões que surgem nos pensamentos de uma pessoa durante o processo de decisão é elevado, sendo ainda mais difícil considerando todos os fatores possíveis para todas as ações e as notícias relacionadas a elas. É impraticável para um humano, mas praticável para uma máquina.

O *framework* proposto neste trabalho coleta informações de fontes confiáveis, como os sites das próprias empresas e de *exchanges*, mas também aceita URLs de blogs e redes sociais, de acordo com a configuração do usuário. Na camada seguinte, é feita uma filtragem para garantir a relevância dos dados. Para isto, o usuário deve fornecer uma lista de ações e palavras-chave que lhe são de interesse.

É importante notar que a análise de gráficos não traduz o acontecimento de, por exemplo, uma lei governamental que force diretores de uma empresa a comprar uma grande quantia de ações, como dito por Abdullah, et al. (2013). Desta forma, qualquer notícia pode ser validada para se garantir que ela auxilie em tomadas de decisões.

O próximo passo é de categorizar as notícias. Uma vez aprovados pelos filtros, os dados das notícias são vistos como candidatos a informações valiosas e sofrem categorizações. Categorias são identificadas pelo compartilhamento de semântica e conjuntos de palavras dentre as notícias. Tal análise possibilita o armazenamento de recursos léxicos e semânticos na base de dados. Uma vez identificada a categoria, o sistema sabe como analisar os dados que pertencem a ela, apoiando-se na base de dados léxicos e semânticos construídos no processo de categorização.

Como os dados de diferentes fontes sobre uma ação compartilham atributos, eles podem pertencer à mesma categoria. Desta forma, informações podem ser obtidas a partir de quaisquer dados de uma categoria. Para se manter adaptável, caso o padrão das notícias mude com o tempo, o sistema atualiza suas bases léxica e semântica.

Caso não hajam critérios suficientes para categorizar determinados dados, o sistema lança uma exceção e aguarda a entrada do usuário, que é automaticamente considerada como válida e serve de aprendizado para o algoritmo. Além disso, o usuário pode configurar um fator de precisão usado para encontrar categorias.

Por fim, foi usada uma ferramenta open source para extrair informação, ou, em alguns casos, decisões dos textos analisados. Os resultados foram salvos em um arquivo para análise, que foi realizada com *Apache OpenNLP*: um conjunto de ferramentas para processamento de linguagem natural. A Figura 1 ilustra uma visão geral do sistema.

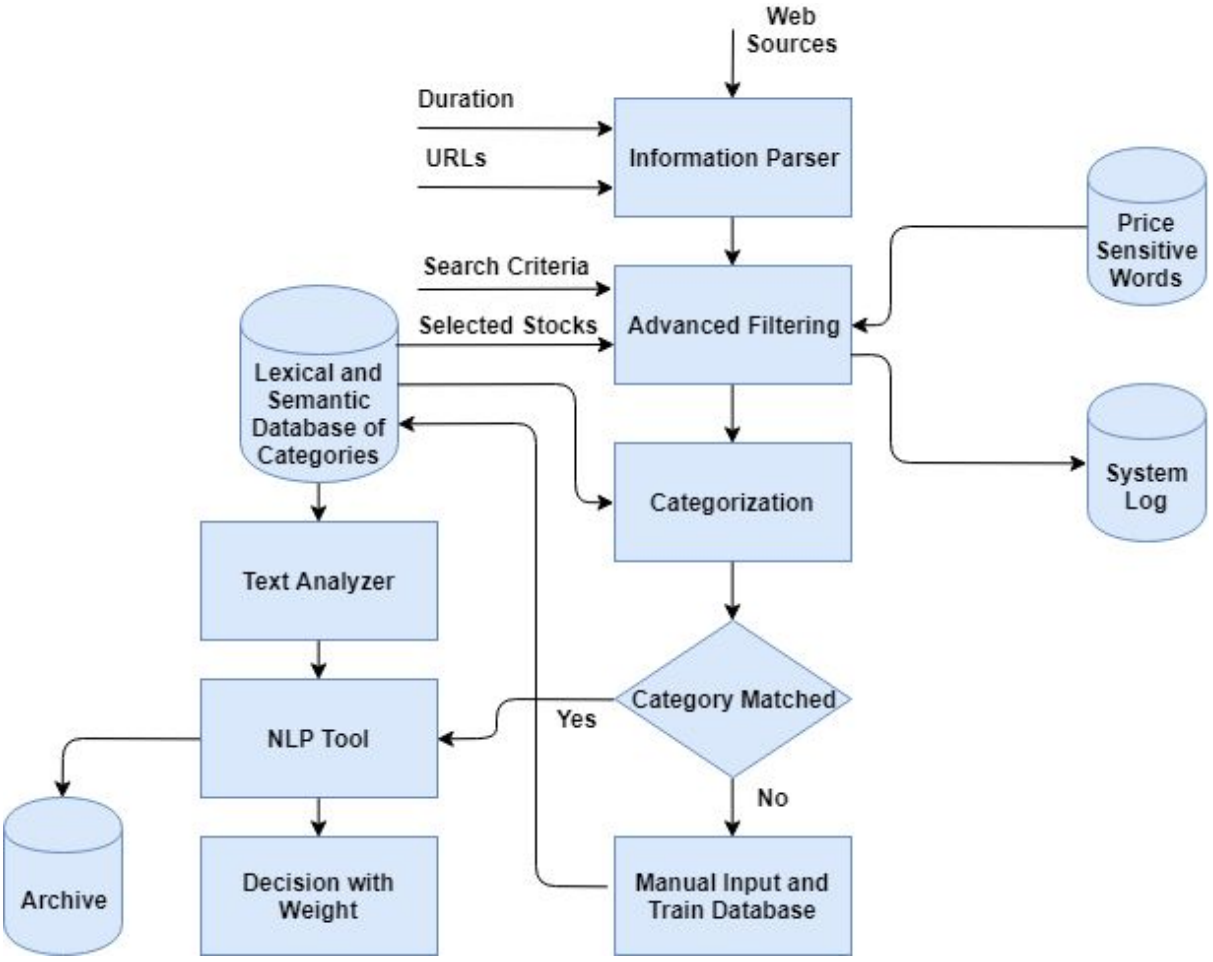


Figura 1 - Visão do processo de geral de captura, análise, categorização e extração de informação - Fonte Abdullah, et. al (2013).

Para se estabelecer fatores de decisão a partir dos dados analisados, foram comparadas as informações mais recentes do banco de dados com o intuito de identificar o impacto das notícias no mercado. Em caso de rumor, o sistema considera os fatores fundamentalistas para fornecer a saída da ação em questão. Isto significa que o peso de decisão por fatores fundamentalistas será alto e a análise técnica será menos significativa.

Como exemplo de experimento, temos quatro diferentes notícias, onde três representam um dos diretores de uma companhia expressando sua vontade de comprar um determinado volume de ações, enquanto uma destas indica que o diretor já realizou a compra. Embora tratem da mesma pessoa, as três primeiras notícias se encaixaram numa categoria e a última em outra, por terem valor semântico diferentes.

Seções dos textos são classificadas previamente à análise, com a intenção de se identificar diferentes elementos de significado. Primeiramente, é encontrado o tipo da mensagem, como “diretor de companhia X”, “expressa” e “sua intenção”. Em seguida, são

buscadas palavras que confirmem a decisão anterior, como “comprar” ou “vender”. Estas palavras também indicam quais dados efetivamente são importantes para se extrair da notícia, como o valor ou a quantia de ações, sendo a última categorização.

Este trabalho não apresenta resultados de performance. Entretanto, a estrutura de sua solução tem aparência congruente e ele apresenta um nível de detalhamento que se faz proveitoso para a base deste trabalho, sendo assim considerado um trabalho correlato.

3.2 Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques

Dados do mercado de ações são um exemplo de dados não estacionários. Em determinados momentos, é possível que hajam tendências, ciclos, um passeio aleatório ou a combinação dos três. É desejado que caso um determinado ano seja cíclico, o modelo esteja apto a seguir este padrão. Entretanto, valores de ações em um ano completo não estão isolados e há dias com passeio aleatório. Também é possível que os preços sejam afetados por fatores externos, como cenários políticos e o estado atual da economia do país.

A Hipótese de Mercado Eficiente, por Malkiel e Fama (1970), afirma que é possível prever os preços das ações, visto que eles são informacionalmente eficientes. Diversas técnicas foram desenvolvidas para predizer suas tendências. Dentre elas, Redes Neurais Artificiais, Máquina de Vetores de Suporte, Modelo Oculto de Markov, Algoritmos Genéticos, Florestas Aleatórias e outras.

Neste estudo foram utilizados dados de um histórico de dez anos de duas ações. 20% de um conjunto de dados de 10 anos foi subdividido em porções iguais com o intuito de realizar experimentos e obter o melhor conjunto possível de parâmetros que representem os dados.

Dez indicadores técnicos, listados abaixo, são dados como entrada aos modelos preditivos. Estes valores passam por uma camada de decisão que serve para converter valores contínuos para discretos, representando a tendência do valor da ação, como representada na Figura 2. Para representar subida e descida, são usados os valores +1 e -1, respectivamente. Desta maneira, os dados de entrada de cada um dos modelos preditivos mantém o mesmo padrão.

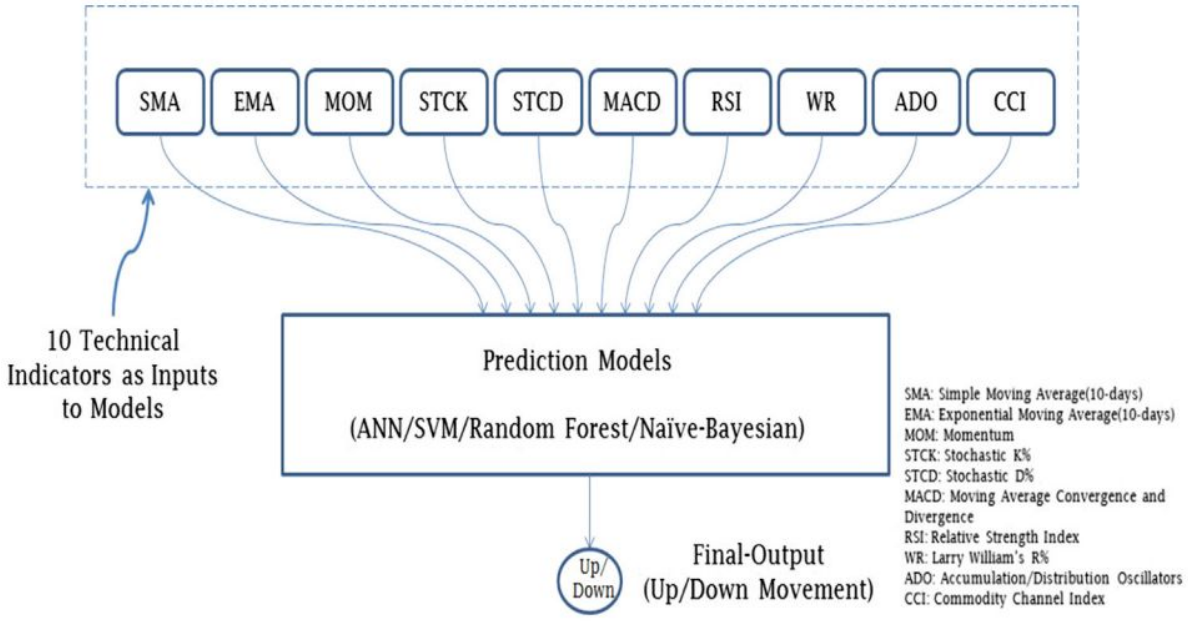


Figura 2 - Representação do modelo de predição, utilizando dez parâmetros para fazer a predição de movimento subida e descida de ação. Fonte: Patel, et. al (2014)

Os indicadores usados, juntamente com seus modos de operação, estão listados abaixo:

- *Médias Móveis*, uma simples técnica de análise que suaviza a curva dos dados. Duas médias foram usadas: simples e ponderada. Ambas considerando um intervalo de dez dias. Se o preço está abaixo da média, a tendência é de queda, se está acima, é de subida;
- *Stochastic K%*, *Stochastic D%* e *Lary Williams' R%* são osciladores estocásticos, que indicam claramente a tendência para qualquer ação. Quando crescentes, a maior probabilidade é de que os preços subam. Isso significa que se o valor estocástico num tempo *t* é maior que em *t-1*, a opinião de tendência é de subida e vice-versa;
- *MACD* segue a tendência da ação, então, se este valor sobe, o preço sobe.
- *RSI* geralmente é utilizado para identificar compras e vendas com preços extrapolantes. Varia de 0 a 100. Se exceder o valor de 70, significa que o preço de compra foi excessivo e que pode haver queda num curto prazo. O cenário inverso se aplica para um valor abaixo de 30;
- *CCI* mede a diferença entre as mudanças de preço de uma ação e a média de seus preços;

- *Oscilador A/D* também segue a tendência da ação;
- *Momentum* mede a razão de queda e subida dos preços de uma ação. Valores positivos indicam tendência de subida e vice-versa.

Cada um destes indicadores tem sua opinião própria sobre a tendência de uma ação. Isso significa que os modelos preditivos devem encontrar a correlação entre estas tendências de entrada e a tendência de saída.

Medidas de acurácia e precisão são usadas para avaliar a performance dos modelos propostos. O cálculo destes valores requer estimar a precisão e a revocação, que são avaliadas a partir de Verdadeiros Positivos, Falsos Positivos, Verdadeiros Negativos e Falsos Positivos.

O propósito de experimentar e comparar os modelos é de encontrar aquele que faz a melhor predição, considerando a melhor combinação de parâmetros encontrada. Os experimentos finais mostram que todos os modelos se comportaram bem com dados discretos como entrada mas a Máquina de Vetores de Suporte, Florestas Aleatórias e o Classificador Bayesiano Ingênuo obtiveram melhores resultados que a Rede Neural Artificial. Por fim, percebe-se que a acurácia dos modelos é próxima a 90%, exceto pela Rede Neural Artificial.

Performance of prediction models on discrete-valued comparison data set.

Stock/Index	Prediction Models			
	ANN		SVM	
	Accuracy	F-measure	Accuracy	F-measure
S&P BSE SENSEX	0.8669	0.8721	0.8869	0.8895
NIFTY 50	0.8724	0.8770	0.8909	0.8935
Reliance Industries	0.8709	0.8748	0.9072	0.9080
Infosys Ltd.	0.8572	0.8615	0.8880	0.8898
Average	0.8669	0.8714	0.8933	0.8952
	Random forest		Naive-Bayes	
	Accuracy	F-measure	Accuracy	F-measure
S&P BSE SENSEX	0.8959	0.8985	0.8984	0.9026
NIFTY 50	0.8952	0.8977	0.8952	0.8990
Reliance	0.9079	0.9087	0.9222	0.9234
Infosys	0.9001	0.9017	0.8919	0.8950
Average	0.8998	0.9017	0.9019	0.9050

Figura 3 - Performance de modelos de predição em um conjunto de dados discretos. Fonte: Patel, et. al (2014)

3.3 Sentiment analysis: capturing favorability using natural language processing

A análise de opiniões espontâneas em documentos é uma tarefa difícil e de extrema importância, pois estas podem influenciar a opinião pública e fazer circular rumores negativos online, o que pode causar grandes problemas para uma organização. Por conta disso, detecção de opiniões favoráveis e desfavoráveis automática, necessita de grande inteligência e profundo entendimento do contexto textual, com base no senso comum e conhecimento de domínio, bem como conhecimento linguístico, visto que a interpretação de opiniões pode ser discutível mesmo para humanos.

A principal tarefa da análise de sentimentos é a de identificar como os sentimentos são expressos em textos, além de quais expressões indicam positividade (opinião favorável) e negatividade (opinião desfavorável), sobre algum sujeito. Por isso, a análise de sentimentos envolve a identificação de expressões de sentimento, polaridade e a força de expressões e a relação destes com o sujeito.

Nasukawa e Yi (2003) buscam identificar fragmentos de texto que denotam algum sentimento sobre o sujeito dentro de documentos, classificando todo o documento como positivo ou negativo em relação ao seu tema. Nesta tarefa, a identificação de relações semânticas é chave para se ter precisão quanto à polaridade dos sentimentos.

Além de adjetivos, outras palavras podem ser usadas para expressar sentimentos, como substantivos, advérbios e verbos. Uma expressão de sentimento, utilizando adjetivos, denota o sentimento por trás de um substantivo, como por exemplo na sentença “Bom produto”. Da mesma forma, uma expressão de sentimento utilizando advérbio denota o sentimento para o seu verbo, como em “Tocar lindamente”, em que a polaridade do sentimento é herdada pelo verbo.

Por isso, expressões de sentimentos que utilizam adjetivos, advérbios e substantivos podem facilmente ser definidas como positivas ou negativas em termos de polaridade. Em um exemplo como “XXX supera YYY”, um sentimento positivo está diretamente direcionada ao sujeito e um sentimento negativo está diretamente direcionada ao objeto. Por outro lado, alguns verbos não denotam qualquer sentimento por si sós, mas simplesmente transferem sentimentos através de seus argumentos. Por exemplo, o verbo *ser* transmite o sentimento do seu complemento ao seu sujeito, como na frase “XXX é bom”, onde o sentimento positivo do complemento “bom” é transferido para o sujeito “XXX”.

Nasukawa e Yi (2003) propõem no framework utilizar processamento de linguagem natural (PLN) para analisar as relações semânticas entre sentimentos e o termo sujeito das

frases, um algoritmo Part-Of-Speech Tagger (POS Tagger) para fazer a desambiguação de algumas expressões polissêmicas, análise sintática para identificar relações entre as expressões de sentimento e o termo sujeito da frase. Também foi utilizado um framework Shallow Parser para identificar os limites das frases e suas dependências locais em conjunto com o POS tagger, ao invés de se utilizar um analisador completo, que tenta identificar toda a estrutura de dependências de todos os termos.

A assertividade do framework foi avaliada utilizando como base classificações manuais humanas. A fórmula da assertividade foi criada da seguinte maneira: saídas corretas do framework, baseadas no julgamento manual, divididas por todo os casos classificados como sentimentos positivos ou negativos pelo sistema.

Os erros do sistema, no geral, se deram por conta de sentenças de estrutura complexa, onde o contexto da entrada negava o sentimento do todo, ou seja, frases negativas com significado positivo. Por exemplo, na frase a seguir o framework identificou um sentimento positivo em uma frase negativa:

<entrada> (assunto="AALIYAH")
Se AALIYAH é tão boa, por que ela nunca ganhou um Grammy. Você gosta dela? Eles sabem disso? Você gosta deles?
<saída>
***+1** AALIYAH (AALIYAH)---be (was so)---good (good)*

O algoritmo não é capaz de entender o sarcasmo em frases e por conta disso acaba utilizando as palavras positivas da frase para classificá-la como **+1** (Classificação positiva). Consumindo dados abertos, neste caso reviews de câmeras, pode-se notar o tipo de frase em que o framework teve dificuldade na classificação do sentimento:

<entrada> (assunto="foto")
É difícil tirar um foto ruim utilizando essa câmera.
<saída>
***-1** ruim---foto (uma foto ruim)*

A frase foi classificada como negativa, mesmo sendo uma frase positiva em relação a câmera. Classificando o sentimento na frase como **-1** (Classificação negativa).

Nasukawa e Yi (2003) alcançaram cerca de 95% de precisão e 20% de recall utilizando apenas frases sem ambiguidade, sendo precisão a identificação de sentimentos positivos em frases e recall o número classificações positivas verdadeiras com o framework desenvolvido. Considerando também frases ambíguas, a precisão caiu para 75%. Na

arquitetura atual, é necessário o desenvolvimento manual de léxicos de sentimento, o que impossibilita sua utilização para outros tipos de contexto, além dos utilizados no experimento.

3.4 Stock market random forest-text mining system mining critical indicators of stock market movements

A mineração de textos vem se destacando como área de pesquisa e vem sendo aplicada em uma variada gama de áreas de interesse. Em particular, podemos destacar a análise de crises de mercados de ações.

Devido à importância dos mercados de ações e sua influência direta na economia, o estudo buscou investigar os indicadores críticos que caracterizam os movimentos do mercado e que podem ser importantes para a tomada de decisão de negócios e financeira.

As técnicas de mineração de dados, apenas, não são capazes de prever com grande confiabilidade a flutuação de ações de uma empresa. Porém, em conjunto com mineração textual se mostra muito mais efetiva nesta tarefa. Para demonstrar esta melhora na predição, foi construído um modelo preditivo de flutuação utilizando mineração de dados e o método Florestas Aleatórias, que é um algoritmo de classificação supervisionada capaz de classificar grandes quantidades de dados com grande precisão. Este modelo pode ser dividido em dois estágios: (i) fazer o processamento de linguagem natural para análise de notícias e (ii) fazer a análise e classificação semântica dos indicadores chave.

Elagamy, Stanier e Sharp (2018) utilizaram notícias do jornal Financial Times no intervalo de 2008 a 2012 para treino e testes da aplicação. Para os testes, foram consideradas notícias sobre a crise do mercado de ações em Dubai de 2009. O período foi escolhido para abranger tanto pré quanto pós crise. Foram utilizados 544 artigos para treino, estes com cerca de 1031006 palavras.

A etapa de processamento de linguagem natural, citada anteriormente, inclui três tarefas, sendo a primeira a análise léxica (é feita a tokenização do texto, remoção de stop words e stemização), a segunda, de análise semântica é a etapa onde é feita a extração dos radicais das frases, gerando palavras únicas que podem ser aplicadas em diversas aplicações. Estas palavras são chamadas de unigramas e quando são utilizadas duas palavras são chamadas de bigramas. A terceira e última é a etapa de extração dos indicadores destes recursos.

O segundo estágio engloba a análise e a classificação dos recursos extraídos, em suas classes semânticas apropriadas, seguidas de uma validação cruzada. Os recursos são classificados primeiramente em: criticamente baixo, baixo, neutro, alto e criticamente alto,

para que depois sejam agrupados em três categorias citadas abaixo. Para descobrir conhecimentos ocultos e as relações entre os recursos extraídos, é aplicado o algoritmo Florestas Aleatórias, apresentado na Figura 4.

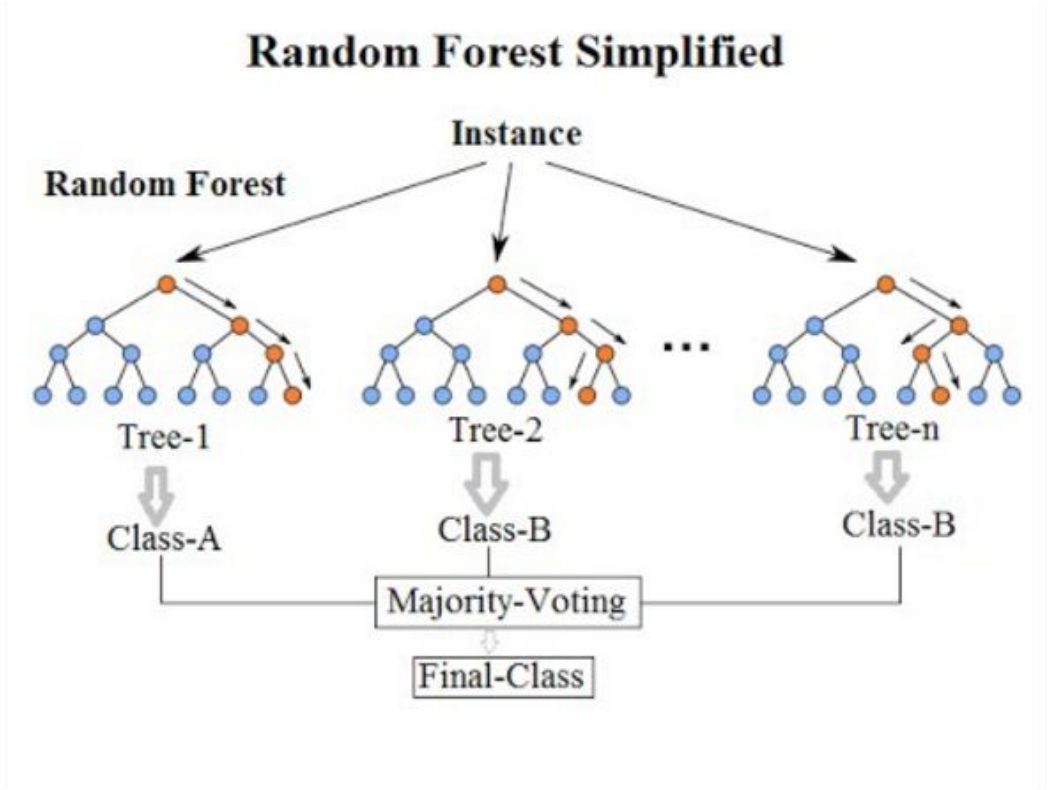


Figura 4 . Exemplo de como o algoritmo Random Forest encontra as relações entre classes. A abordagem SMRF-TM utiliza o mesmo princípio para descobrir relações entre atributos de unigramas. Fonte: Elagamy, Stanier e Sharp (2018)

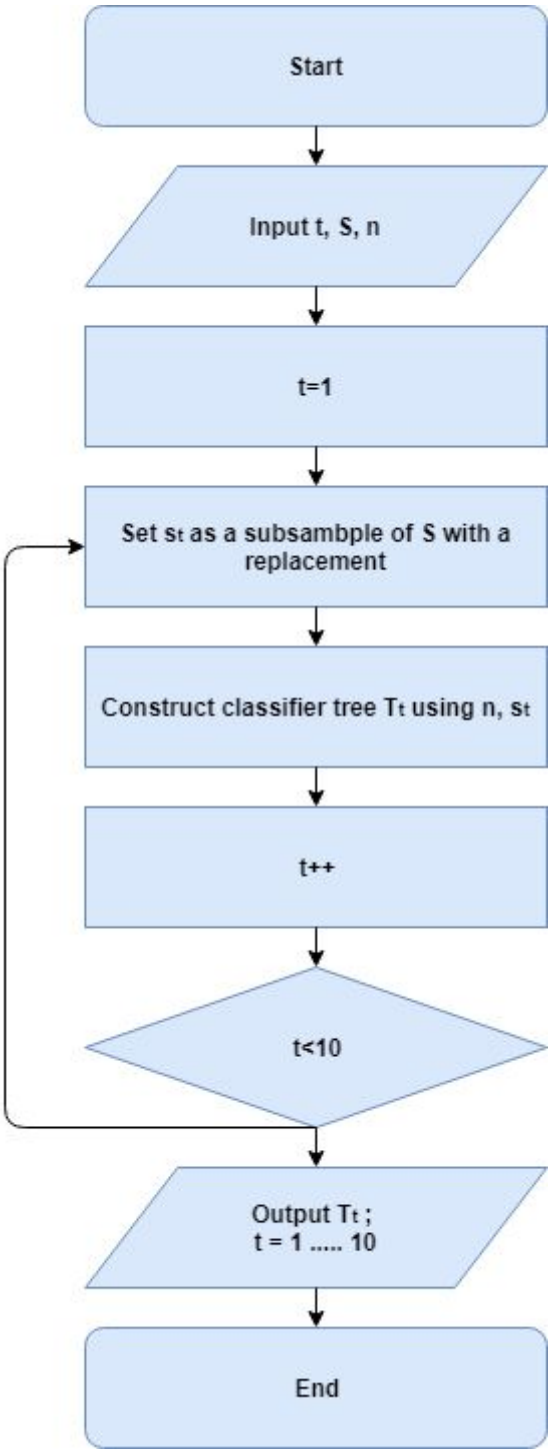


Figura 5. Esquema do algoritmo executado por cada árvore do SMRF-TM. Fonte: Elagamy, Stanier e Sharp (2018)

Os artigos, após classificados, foram divididos em três grupos, utilizando o algoritmo Maximização de Expectativas, que iterativamente mistura as estimativas de entrada do modelo a fim de encontrar um arranjo otimizado. O algoritmo classificou os artigos de

acordo com seu significado semântico, nas facetas econômica, social/geográfica ou política. A figura 6 apresenta alguns exemplos de unigramas classificados pelo algoritmo, neste caso exemplos da classe baixo.

Need	Debt	Conflict	Nakheel	Recess
Regime	Govern	Asset	Compliant	Destroy
Sale	Rate	Wage	Inflation	Risk
Elect	Fund	Report	Problem	Grow
Credit	Shortage	Downgrade	Emergency	Loan

Figura 6. Exemplos de unigramas classificados na classe “baixo”. Fonte: Elagamy, Stanier e Sharp (2018)

O mercado financeiro é um setor bastante importante e significante para um país e representa um papel crucial no crescimento do comércio e da indústria. Por isso, encontrar formas eficientes de analisar e visualizar os dados deste setor é considerada uma tarefa significante para a economia moderna, segundo Elagamy, Stanier e Sharp (2018).

O modelo desenvolvido estendeu as abordagens atuais de três para oito classes de classificação, utilizando o algoritmo Florestas Aleatórias. O estudo também demonstrou que o algoritmo Florestas Aleatórias pode ter um melhor desempenho do que outros algoritmos (Ex.: Floresta de Rotação, Bagging, J48, Rede Bayesiana, Tabela de Decisão e Toco de Decisão). Alcançou uma performance ainda melhor na classificação de artigos quando utilizou uma abordagem baseada em bigramas.

Classifier	RF	Rotation Forest	Bagging	J48	Bayes Net	Decision Table	Decision Stump
Cross Validation (folds)	40	40	50	30	50	20	5-10-20-30-40-50
Accuracy (%)	98.34	92.83	87.68	84.00	71.13	70.77	43.75
Precision (critical down class)	1.00	1.00	1.00	1.00	1.00	0.91	1.00
Recall (critical down class)	1.00	1.00	1.00	1.00	1.00	0.91	1.00
Precision (down class)	0.99	0.95	0.85	0.84	0.65	0.80	0.38
Recall (down class)	0.99	0.91	0.90	0.84	0.75	0.71	1.00
Precision (neutral class)	0.99	0.89	0.95	0.84	0.96	0.58	0.00
Recall (neutral class)	0.99	0.93	0.83	0.84	0.40	0.73	0.00
Precision (up class)	0.96	0.90	0.83	0.86	0.66	0.78	0.00
Recall (up class)	0.97	0.92	0.89	0.78	0.94	0.67	0.00
Precision (critical up class)	1.00	0.97	0.80	0.63	0.48	0.78	0.00
Recall (critical up class)	0.87	0.91	0.71	0.79	0.35	0.41	0.00

Figura 7 - Resumo dos resultados com melhor performance dos 7 classificadores utilizando unigramas. Fonte: Elagamy, Stanier e Sharp (2018)

Classifier	RF		Rotation Forest		Bagging	J48	Bayes Net	Decision Table	Decision Stump
Cross Validation (folds)	40	50	30	40	30	50	50	50	5-10-20-30-40-50
Accuracy (%)	98.89	98.89	86.21	86.21	81.98	81.25	75.36	73.52	43.75
Precision (critical down class)	1.00	1.00	1.00	1.00	0.98	0.96	1.00	1.00	0.96
Recall (critical down class)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.91	1.00
Precision (down class)	1.00	0.99	0.85	0.86	0.77	0.84	0.63	0.65	0.38
Recall (down class)	0.99	1.00	0.89	0.88	0.86	0.80	0.91	0.83	1.00
Precision (neutral class)	0.97	0.99	0.88	0.87	0.87	0.81	0.98	0.77	0.00
Recall (neutral class)	0.99	0.97	0.85	0.84	0.75	0.74	0.59	0.63	0.00
Precision (up class)	0.99	0.99	0.82	0.81	0.80	0.75	0.77	0.76	0.00
Recall (up class)	0.99	0.99	0.82	0.85	0.80	0.83	0.77	0.68	0.00
Precision (critical up class)	1.00	1.00	0.83	0.86	0.76	0.71	0.67	0.70	0.00
Recall (critical up class)	1.00	1.00	0.74	0.71	0.65	0.79	0.12	0.56	0.00

Figura 8 - Resumo dos resultados com melhor performance dos 7 classificadores com bigramas. Fonte: Elagamy, Stanier e Sharp (2018)

.

3.5 Trading system aplicado à BOVESPA utilizando redes neurais e computação evolutiva.

Araújo (2010) apresenta em seu trabalho uma ferramenta de monitoração do mercado, que, seguindo regras previamente definidas para reconhecimento de um cenário de operação, inicia ou finaliza automaticamente uma operação. Tais regras podem ser feitas a partir de dados de negociações, como volume, preço de fechamento, preço de abertura e também diversos indicadores de análise técnica. Dados de análise fundamentalista e ações de mercados correlatos também foram utilizados.

Com o intuito de se obter um bom resultado nas operações, é importante calcular o momento de saída da operação antes mesmo da entrada, pois desta forma é possível avaliar o risco com o possível ganho previstos. Sendo assim, podemos dizer que a previsão de movimentações futuras do mercado significa entender quais variáveis levam à previsão de ainda outras variáveis. Isso pode ser feito ao observar o passado do mercado e

identificar quais elementos são indicadores significativos para caracterizar o comportamento do próprio mercado.

Para alcançar o objetivo de desenvolver um sistema que faça tais predições, além de compras e vendas automatizadas, Araújo (2010) se baseou em uma metodologia de treinamento supervisionado para redes neurais, que são auxiliadas por algoritmos com características de computação evolutiva.

A principal motivação para o uso de redes neurais para a predição em ações é que (i) o mercado de ações é altamente complexo e difícil de modelar, então um modelo não-linear seria benéfico e (ii) um conjunto grande de séries de dados que interagem entre si é geralmente necessário para explicar o comportamento de uma ação, fato a que as redes neurais se adequam perfeitamente (Nyrgren, 2004).

Araújo (2010) modela uma rede neural preditiva baseando-se no trabalho apresentado por Kastra e Boyd (Kastra et al., 1996), que apresenta sua metodologia dividida nas seguintes 8 partes:

1. **Seleção de variáveis**: tarefa fundamental, mas também complexa devido à não linearidade e aos efeitos combinados de diversas possíveis variáveis a serem escolhidas. As escolhas podem ser ajudadas por teorias econômicas e emprego de técnicas de análise técnica e fundamentalista.
2. **Aquisição de dados**: nesta etapa se deve considerar a disponibilidade e a veracidade dos dados.
3. **Pré-processamento dos dados**: o intuito é de minimizar ruídos, aumentar importâncias de relacionamentos, detectar tendências e aplanar a distribuição variável para auxiliar a rede neural a aprender os padrões relevantes.
4. **Conjuntos de treinamento, validação e teste**: a rede neural deve aprender com o primeiro conjunto, que deve compor a maior parte dos dados. O segundo conjunto serve para avaliar a capacidade de generalização da rede após o treinamento, enquanto o conjunto de testes, que deve possuir dados recentes, serve para uma checagem final do aprendizado da rede neural e não deve ser usado para avaliar e ajustar as entradas da rede.
5. **Paradigmas de redes neurais**: define-se pelo conjunto de propriedades individuais dos neurônios e o modo em que suas entradas são combinadas, o que seria a neurodinâmica, e pela arquitetura, que consiste no número de neurônios em cada camada e número e tipo de interconexões entre eles.

6. **Critério de avaliação**: é comum utilizar a soma quadrática dos erros, mas Jingtao Yao ressalta que é necessário escolher o critério de acordo com a aplicação (Yao et al., 1999) e sugere o Gradiente como bom preditor de tendências.
7. **Treinamento da rede neural**: o objetivo é de fazer com que ela aprenda padrões através de ajustes dos seus pesos sinápticos, determinando um mínimo global da função de erro. Também deve-se considerar a taxa de aprendizagem, que é um valor que determina o tamanho das mudanças nos pesos, com o intuito de obter um número de erro menor.
8. **Implementação**: esta etapa deve ser considerada antes mesmo da coleta de dados, visto que pode restringir a escolha da rede neural e define as etapas de escolha dos dados, critérios de evolução e tempos de treinamento.

Os algoritmos genéticos utilizados em conjunto com as redes neurais têm como funcionamento a ideia de evolucionismo proposta por Charles Darwin. Funciona de forma que as *espécies* sejam criadas aleatoriamente e testadas sob recursos limitados. Os indivíduos melhor adaptados se reproduzem e propagam seu material genético às próximas gerações.

O objetivo é da união de ambos algoritmos é de escolher os parâmetros das redes neurais para conseguir o melhor resultado possível dentro de cada algoritmo, dentre os algoritmos de treino abordados, visando garantir o melhor resultado possível, com maior taxa de acerto e fugindo o máximo possível de convergências para mínimos locais e buscando os pontos mínimos globais. Os seguintes dados foram utilizados como entrada, fazendo-se combinações entre eles para treinar e validar os resultados:

- Cotações diárias de fechamento da PETR4;
- Cotações diárias de fechamento do IBOVESPA;
- Média móvel exponencial de 21 períodos (MM21) das cotações diárias de fechamento da PETR4;
- Índice de Força Relativa (IFR) com 14 períodos das cotações diárias de fechamento da PETR4;
- Volume diário de negócios da PETR4;
- Afastamento da MM21 em relação à cotação diária de fechamento da PETR4.

Os dados obtidos foram previamente analisados por Araújo (2010), garantindo a previsibilidade de fechamento através do expoente de Hurst, além da correlação de Pearson e o teste de hipótese de Student.

Antes de utilizados, os dados foram pré-processados com a intenção de eliminar diferenças de grandezas dos dados de entrada. Por exemplo, foram obtidos valores 0,1 e 121540800 para o mesmo atributo. Todos os dados sofreram normalizações como essa, facilitando e acelerando o treinamento da rede neural.

Em específico, foram utilizadas redes neurais do modelo NARX como base para tomadas de decisões pelo sistema de compra e venda. O motivo é de que a predição feita é "um passo no futuro" (one-step-ahead prediction), pois o intuito do sistema é de tomar ação dia a dia, com base na maior quantidade de dados disponíveis. Isto inclui os valores relativos ao dia atual, já sendo usados para a predição do dia futuro. As redes NARX se encaixam neste cenário, onde aprendem o comportamento dos dados de entrada e predizem se o dia seguinte terá movimento de alta ou baixa. Foi utilizada apenas uma camada escondida para a rede neural, pois não há vantagens aparentes no uso de múltiplas camadas escondidas (Qian; Rasheed, 2004). O algoritmo de aprendizado utilizado foi o de Retropropagação com Regularização Bayesiana.

O acoplamento entre as redes neurais e os algoritmos genéticos foi feito pela definição do treinamento dentro da função de aptidão. O critério a ser minimizado pelo algoritmo genético, medido pela função de aptidão, é o erro com relação entre treinamento e simulação. Esta função define o critério para classificação que representa o quão próximo um indivíduo está da solução desejada, maior o valor conforme maior a proximidade do alvo. Com isto, busca-se encontrar pesos e vieses ótimos para a rede.

As saídas do sistema de predição resultam na entrada do sistema de trocas, sendo elas:

- Preços do ativo: série de valores de cotação diária de fechamentos;
- Direção futura do preço: série de predições da direção do valor futuro;
- Capital inicial: valor financeiro a ser colocado na bolsa para compra e venda;
- Lote: quantidade de ações que compõem o lote de uma determinada ação.

Sempre que a previsão for de alta, o sistema comprará o maior número possível de lotes. Em um cenário de baixa, todas as ações serão liquidadas. Também foi executada a estratégia de *buy and hold*, que consiste em adquirir o maior número de ações com o capital

inicial e não realizar qualquer outra operação até o último dia. O objetivo é de comparar o desempenho do sistema automatizado com esta estratégia.

Inicialmente foi feito um estudo para avaliar se a rede neural iria gerar um conjunto puramente aleatório de previsões, indicando que ela não conseguiu encontrar qualquer característica dentre os dados que possibilitasse fazer uma previsão. Esta configuração dispunha de apenas duas entradas.

A camada escondida da rede neural foi testada com os números de 1 até 20 neurônios. Com 58,17%, os 20 neurônios obtiveram o melhor resultado e o modelo foi considerado válido, visto que acertos de 56% geralmente são considerados satisfatórios. Ao se aumentar a quantidade de atrasos para 10, obteve-se o resultado de 67,77%. Atrasos representam o número de informações de execuções anteriores que a rede irá guardar. Estes resultados foram obtidos a partir de dados que participaram do conjunto de treinamento. Com um conjunto de testes segregado, o modelo trouxe 54,55% de acerto com 10 neurônios na camada escondida, em contraste a 62,11% dos testes anteriores.

Em seguida, foi testado o sistema de múltiplas entradas com as seis entradas citadas anteriormente e foram considerados 10 atrasos de entrada e saída. O melhor resultado foi para 16 neurônios na camada escondida, resultando em 55,12% de acerto. Sendo esta taxa próxima a 56%, ela é considerada satisfatória.

Para testes do sistema de troca, foi escolhida a melhor provisão de entradas múltiplas, de 16 neurônios e 10 unidades de atraso. Considerou-se o tamanho de lote como 100 e o capital inicial como R$ 120.000,00. Também foi considerada a estratégia de *buy and hold*. O resultado da simulação pode ser visto na tabela 1.

Capital Inicial	R$ 120.000,00
Capital Final	R$ 409.838,36
Retorno	241,53%
Buy and hold	83,98%

Tabela 1 - Resultados da simulação do sistema compra e venda automática de ações - Fonte: Araújo (2010)

Podemos observar que o resultado obtido pelo sistema automático superou a estratégia *buy and hold*, se mostrando eficiente. Todavia, é importante ressaltar que, caso

houvesse acertado todas as operações, o capital final esperado seria de R$ 489.635.898,92, resultando em 407929,92% de retorno. Com estes resultados, pode-se concluir que os objetivos deste trabalho foram alcançados

3.6 Conclusões

Os cinco trabalhos relatados neste capítulo trouxeram importantes conhecimentos sobre maneiras de se tratar notícias e como as relacionar a valores na bolsa. Abdullah, et al. (2013) denota a importância de se ter categorias que identifiquem as notícias pela semântica e o conjunto de palavras que as compõem. Entretanto, percebe-se que é importante manter as definições da categoria dinâmicas, para que se ajustem automaticamente conforme mais notícias sejam processadas.

Patel et. al (2014) não trazem PLN, mas oferecem base de conhecimentos relativos à aprendizagem de máquina sobre flutuações dos valores da bolsa utilizando diferentes estatísticas como parâmetros entrada de um sistema. Também explicitam a importância da normalização dos dados a seleção dos parâmetros para uma maior eficiência na predição.

Nasukawa e Yi (2003) proporcionam base para PLN que será utilizada durante o desenvolvimento deste trabalho, mostrando pontos relevantes na análise de sentimentos de textos, como a exploração de adjetivos e advérbios atrelados a substantivos e verbos, além de alertar para problemas existentes. Elagamy, Stanier e Sharp (2018) agregam ao alicerce de PLN, demonstrando três etapas importantes para o processo.

Araújo (2010) utiliza redes neurais para predição dos valores da bolsa por não serem valores lineares, além de unir algoritmos genéticos para a melhor escolha de parâmetros. Isto nos ajudará a modelar nossos próprios algoritmos. Araújo (2010) também demonstra diferentes maneiras de se avaliar a performance do sistema proposto.

A pesquisa se estendeu a outros trabalhos, onde podemos perceber que redes neurais artificiais são bastante populares na literatura recente, segundo Krollner, Vanstone e Finnie (2010). Existe uma tendência de uso estabelecida para modelos de redes neurais e os aprimorar com algoritmos evolucionários ou até mesmo os combinando com tecnologias emergentes, formando um sistema híbrido. Em geral, os parâmetros de entrada mais utilizados pelos sistemas são de abertura, alta, baixa e fechamento diários. Também são muito utilizados dados matematicamente transformados como a média móvel simples.

A Tabela 2 sumariza a relação dos trabalhos correlatos com este presente trabalho, a um nível de macro detalhamento.

Autor	Aprendizado de Máquina	PLN	Aplicação em Bolsa de Valores
Abdullah, et al	X	X	X
Patel et. al	X		X
Nasukawa e Yi	X	X	
Elagamy, Stanier e Sharp	X	X	X
Araújo	X		X

Tabela 2 - Sumário dos trabalhos correlatos e assuntos abordados. Fonte: Autoria própria

No sentido inverso, este trabalho apresenta em adicional aos trabalhos correlatos o uso de tecnologia RSS para captura periódica de notícias, além de trabalhar somente com notícias no idioma português, visto que foram publicadas em portais brasileiros. Também foram realizadas sessões de testes manuais, comprovando o resultado obtido durante as fases de treinamento e teste. Por fim, diferentemente dos trabalhos correlatos, e com exceção de normalização, não foram realizadas transformações sobre os valores históricos da bolsa, como médias móveis ou técnicas similares.

4 Desenvolvimento

4.1 Método

Este trabalho visa prever flutuações relevantes nos valores de ações de empresas de tecnologia, através de modelos de predição que levam em consideração informações relativas às empresas, extraídas de notícias através de processamento de linguagem natural. Para isso, o modelo de predição foi dividido em partes de um processo, onde cada algoritmo é responsável por executar uma ação específica que gera como saída a entrada para a próxima etapa. Dessa forma, o processo completo considera as seguintes etapas: busca de notícias online, busca de valores históricos da bolsa de valores, treinamento do modelo de predição de sentimentos por processamento de linguagem natural, análise de sentimentos de notícias capturadas, estruturação do conjunto de dados, treinamento do modelo de predição e testes do modelo.

Embora contenha pequenas etapas que são detalhadas nos tópicos a seguir, podemos considerar que este processo se caracteriza em dois grupos que se distinguem fortemente: processamento de linguagem natural e processamento de dados históricos de valores de ações; sendo eles representados respectivamente pelo diagrama apresentado na Figura 9.

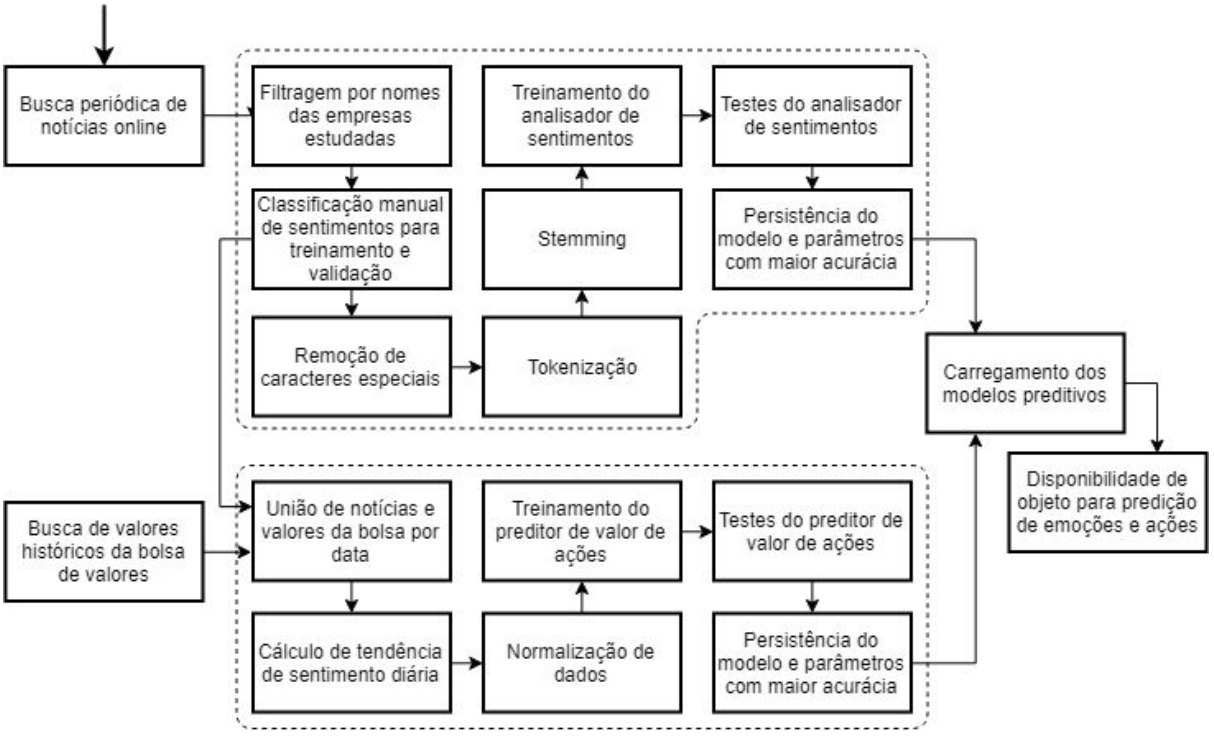


Figura 9 - Diagrama de fluxo de informação do modelo. Processamento de notícias e valores de ação.
Fonte: Autoria própria

4.1.1 Busca de notícias online

Uma das tecnologias mais comuns e utilizadas quando se fala sobre registro e organização de notícias online é o RSS, acrônimo de Rich Site Summary ou Really Simple Syndication, que em tradução para o Português pode ser definido como: Resumo rico do site ou Distribuição realmente simples, o que podemos resumir como uma forma simples de organizar e distribuir o conteúdo de um site.

Para fazer o registro de notícias relativas às empresas escolhidas e os assuntos que permeiam este trabalho, foi utilizada uma ferramenta organizadoras de feed RSS, chamada Miniflux. Essa ferramenta faz registro e organização de notícias online, de acordo com sites fonte, categorias de assuntos e horário de publicação de uma notícia. Para utilização da ferramenta, foi criada uma máquina virtual com o sistema operacional Ubuntu 16.04, onde a ferramenta é executada. Ela foi configurada para buscar notícias de tempos em tempos e atualizar o seu banco de dados com as notícias, sempre que uma nova notícia fosse lançada. A ferramenta foi configurada para obter notícias sobre as 11 (onze) categorias distintas de 7 (sete) fontes (Portais de notícias online) distintas. As fontes e categorias selecionadas são apresentadas abaixo:

- **BBC Brasil**: Brasil, Economia e Internacional;
- **Folha de São Paulo**: Blog, Ciência, Mercado, Poder e Tecnologia;
- **G1**: Economia, Política e Concursos e Emprego;
- **O Globo**: Completo;
- **Olhar Digital**: Completo;
- **UOL**: Economia e Tecnologia;
- **Valor Econômico**: Completo.

A seleção das fontes foi feita na tentativa de utilizar apenas portais de notícias online com algum grau de credibilidade. Os principais aspectos utilizados na escolha dos portais foram a popularidade dos portais (Utilizando o conhecimento popular como parâmetro) e a frequência de notícias publicadas por dia. A seleção das categorias foi feita na tentativa de maximizar as possibilidades de captura de notícias relacionadas às empresas. Por se tratarem de empresas multinacionais, diversas categorias que não dizem respeito nem a tecnologia nem a emprego também foram adicionadas, como por exemplo a categoria Política, tendo em vista que em diversas ocasiões as empresas selecionadas (Microsoft, Apple e Tesla) já se envolveram em assuntos relacionados ao tema.

Com a seleção das fontes e das categorias e configuração destas no Miniflux, foi iniciada a captura de novas notícias periodicamente. As notícias foram capturadas do dia 22/08/2018 até o dia 05/04/2019. O objetivo desta captura é de criar um conjunto de dados heterogêneo, contendo notícias de diversos tipos e fontes distintas, que posteriormente seriam transformadas e filtradas. Isto significa que todas as notícias oriundas das fontes supracitadas dentro de suas categorias são armazenadas no banco em um formato padrão da ferramenta Miniflux, que possui, dentre diferentes atributos, “data” e “conteúdo”, como os mais relevantes para esta pesquisa.

De todo o conjunto de dados do banco, foram selecionadas somente aquelas que contivessem o nome de uma das três empresas estudadas na coluna “conteúdo”, ignorando

qualquer diferença entre caracteres maiúsculos e minúsculos. Em resumo, o servidor Miniflux recebe fontes e categorias, das quais irá buscar notícias, e armazena num banco. Este banco é consultado para dele se extrair notícias, filtradas de acordo com os nomes das notícias.

No total foram 519 notícias para Apple, 176 para Microsoft e 76 para Tesla, dentre um total de 23.602 notícias. Por fim, estes dados filtrados foram exportados para um arquivo em formato de valores separados por vírgula (comma separated values, CSV), que se assemelha a uma tabela, porém proporciona maior flexibilidade ao algoritmo de treinamento implementado.

4.1.2 Busca de valores históricos da bolsa de valores

Os valores históricos do índice BOVESPA para as três empresas estudadas foram obtidos do provedor de dados Alpha Vantage, que disponibiliza uma API acessível via protocolo HTTP. Os dados vêm no formato JavaScript Object Notation (JSON). Para cada dia, temos os atributos dos valores de Abertura, Fechamento, Alta e Baixa representados em reais, além do Volume representado por por unidade.

Para tanto, o usuário deve realizar uma requisição HTTP para www.alphavantage.co/query enviando os seguintes parâmetros:

function: qual dos diferentes formatos e agrupamentos de dados será utilizado na busca. Este trabalho utilizou o formato de série temporal diária, denominado pela API como *TIME_SERIES_DAILY*;

symbol: o símbolo da empresa; Microsoft é representada por *MSFT34.SA*, Apple por *AAPL34.SA* e Tesla por *TSLA34.SA*;

outputsize: explicita o tamanho do retorno da API. Neste trabalho utilizamos o formato completo, que é *full*;

apikey: uma chave única que determina qual usuário está realizando a busca, pode ser obtida ao se cadastrar no serviço através de uma conta de e-mail válida.

Estão disponíveis dados a partir do dia 29/02/2012 para Apple, 09/12/2010 para Microsoft e 19/01/2017 para Tesla. Todos os registros diários possuem o mesmo formato, desde o primeiro ao dia atual, para as três empresas. A Tabela 3 demonstra dois dias como exemplo para a empresa Microsoft, cujo símbolo é MSFT34.SA:

Data	Abertura	Fechamento	Alta	Baixa	Volume
2013-07-10	78,4100	78,4100	78,4100	78,4100	500
2013-07-11	80,4700	80,4900	80,4900	80,4700	6000

Tabela 3 - Representação de dados de ação da empresa Microsoft em dois dias, adquiridos da API - Fonte: Autoria própria

4.1.3 Análise de sentimentos de notícias diárias

O modelo proposto neste trabalho procura relacionar notícias sobre empresas multinacionais de tecnologia aos seus valores na bolsa. Esta possível relação se dá pelo sentimento das notícias, motivo pelo qual se faz necessário um modelo analisador de sentimentos. Para atingir este objetivo, são executadas diversas etapas, as quais são detalhadas abaixo e ilustradas pelo diagrama apresentado na Figura 9 Estes mesmos processos foram realizados para as notícias relativas a cada uma das três empresas em estudo, individualmente.

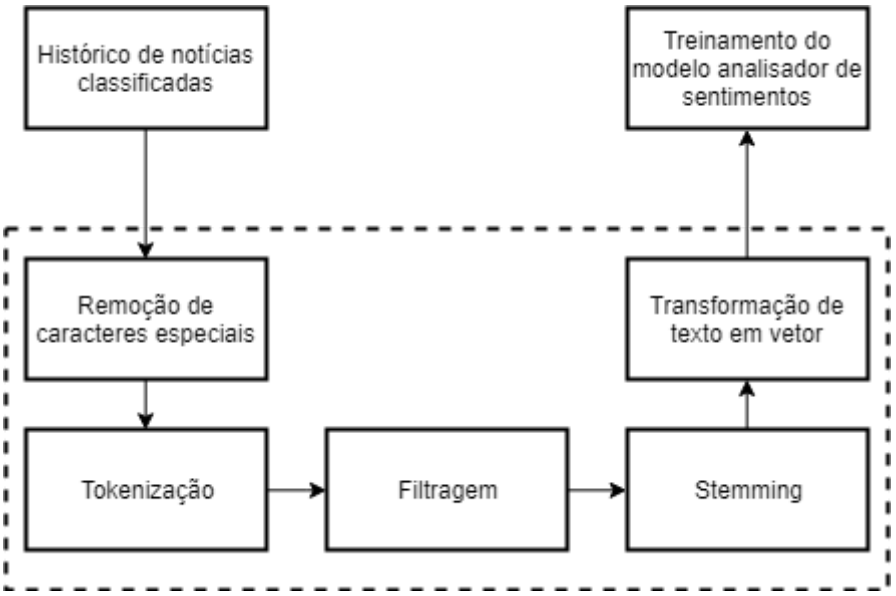


Figura 10 - Treinamento do analisador de sentimentos. Fonte: Autoria própria

Tratando este trabalho de avaliar a predição de ações para três empresas distintas, foi optado por criar três diferentes instâncias de um mesmo analisador, sendo um para cada empresa, recebendo dados exclusivos de cada uma destas empresas nas etapas de aprendizagem e teste.

Diferentes algoritmos foram utilizados no treinamento do modelo analisador de sentimentos, a fim de definir um algoritmo ótimo, ou seja, aquele que apresenta o melhor desempenho em relação a tempo de duração de treinamento e acurácia dos resultados. Estes algoritmos recebem como entrada vetores de palavras, originados das notícias obtidas através da API do servidor Miniflux. Já os sentimentos, que servem como classe para as notícias, foram classificados nestas notícias de forma manual pelos autores deste trabalho, conforme demonstra o diagrama da Figura 10.

Figura 11 - Processo de classificação de sentimento de notícias. Fonte: Autoria própria

Para a classificação foram utilizados valores inteiros, que são necessários para o processo de treinamento do classificador de sentimentos, sendo possíveis apenas três valores de representação de sentimento para uma notícia:

-1 : Que representa uma notícia negativa;

0 : Que representa uma notícia neutra; e

1 : Que representa uma notícia positiva.

Uma notícia é classificada como positiva quando menciona pelo menos um dos seguintes elementos: subida do valor da bolsa, lançamento de produtos, expansão da empresa, criação de parcerias ou aceitação do público; é considerada negativa quando trata de queda no valor da bolsa, produtos defeituosos, fechamento de lojas, encerramento de parcerias, processos jurídicos e demissões. Fora de todos estes assuntos, a notícia é classificada como neutra.

Tendo classificado o conjunto de dados, a próxima etapa é o pré-processamento, que faz, em primeiro lugar, a remoção de caracteres especiais, notícia por notícia, para que sejam utilizados nas próximas etapas apenas aqueles reconhecidos como parte da língua portuguesa. Todos os marcadores de linguagem HTML foram excluídos, juntamente seus atributos, assim como códigos especiais e quebras de linha.

Para as etapas seguintes foi necessário isolar cada palavra da notícia, construindo uma lista de palavras a partir da notícia inteira; este processo é chamado de tokenização. Desta lista, realizamos uma filtragem, ignorando quaisquer palavras menores que três caracteres e também palavras vazias, que são palavras que podem ser consideradas irrelevantes no processo de análise de um texto para extração de sentimento, pelo fato de não trazerem significado quando avaliadas isoladamente, sem a conexão com outros termos, ou seja: necessitam de um contexto, criado a partir de outras palavras a elas conectadas. Alguns exemplos são palavras que servem apenas como conectivos entre outras palavras, encontrando-se a relevância no verbo e no objeto direto da sentença, como: lhe, isso, ser, às, numa, etc.

Uma vez que os termos estejam preparados, realizamos o processo de stemming das palavras, que consiste em transformar uma palavra em qualquer variação para o formato de sua raiz, permitindo ao algoritmo identificar um termo mesmo que ele venha de contextos e com formas diferentes. Ilustramos os processos de tratamento de texto aplicados utilizando o exemplo apresentado abaixo.

Dada uma frase em seu formato original:

> **“fontes internas da apple informaram que empresas devem fazer investimentos na casa de u$ 365 milhões”**

Após os processos de remoção de palavras irrelevantes e stemming, obtemos a seguinte frase:

> **“font intern apple inform empres dev faz invest cas 365 milhõ”**

Por fim, transformamos nossos dados em vetores, uma vez que são compatíveis com todos os diferentes modelos abordados. Para tanto, utilizamos a estratégia de n-gramas, que é o processo de dividir uma frase em pequenas sentenças formadas por n termos sequenciais. O intuito foi de verificar diferentes formas de entrada trariam melhoria para a performance do analisador de sentimentos, considerando não somente a frequência de termos encontrados, mas também a frequência de sequências de termos. Abaixo ilustramos um exemplo de bigrama, que é um n-grama de dois termos, chamado de bigrama.

Dada uma frase em seu formato original:

> **“João disse olá”**

Aplicando a transformação por bigramas, obtemos as seguintes sentenças:

> **“João disse”** e
> **“disse olá”**

Cada dupla de valores do vetor corresponde a uma bigrama possível de nosso conjunto de dados inteiro, considerando todas as notícias já processadas. Ao calcular o vetor de uma nova notícia específica, é criada uma nova tupla com o número de ocorrências

de seus bigramas, devidamente mapeados conforme as posições específicas no vetor. Isto significa que o vetor de uma nova notícia se inicializa com todos os valores iguais a zero; conforme os bigramas são encontrados, seus valores correspondentes no vetor são acrescidos, gerando ao fim do processamento o número de ocorrências de cada bigrama. Num cenário hipotético, já com frases previamente analisadas, a frase “disse olá, ele disse olá” teria os seguintes valores mapeados no vetor conforme mostra a Tabela 4

Bigrama	Contagem
meu nome	0
disse olá	2
nome é	0
olá ele	1
ele disse	1
é pequeno	0

Tabela 4 - Representação da vetorização de bigramas. Fonte: Autoria própria

Com o intuito de comparar a performance da estratégia de bigramas, também utilizamos o formato de n-gramas onde n é igual a 1, ou seja, não se considera a sequência de termos, mas somente os próprios termos, e também onde n é igual a 3, conhecido como trigrama. A seguir na Tabela 5, é apresentada a comparação dentre estes três experimentos iniciais, com notícias referentes à empresa Apple, da qual há o maior número de notícias em nosso banco. Cada acurácia representa a média das acurácias de 10 iterações de treinamento e teste. Os dados completos, para todas as empresas, estão disponíveis na seção de Resultados (seção Seção 5.1.1):

Algoritmo	Token	Bigrama	Trigrama
AdaBoost	45,06%	48,39%	45,19%
Árvore de Decisão	45,32%	51,35%	49,80%
Gradient Boosting	51,21%	52,82%	47,43%
K Vizinhos Mais Próximos	42,37%	45,64%	45,89%
Regressão Logística	53,46%	53,07%	50,06%
Floresta Aleatória	52,94%	51,80%	49,16%

Tabela 5 - Acurácia do preditor de sentimentos utilizando token, bigrama e trigrama. Fonte: Autoria própria

Insatisfeitos com os resultados apresentados na Tabela 5, tentamos outra abordagem. Mantivemos o formato de n-gramas, porém alteramos o modelo de contagem, que anteriormente era simples, para o modelo "frequência do termo–inverso da frequência nos documentos" (term frequency–inverse document frequency, TF-IDF). Este modelo também faz a contagem das ocorrências das palavras, porém cada palavra tem seu valor calculado proporcionalmente ao número de ocorrências dela em todas as entradas. Isto significa que se todas as entradas possuem uma palavra, ela não é relevante por não ser específica. Em contrapartida, se apenas um por cento das entradas possui uma determinada palavra, isto significa que aquela é uma palavra chave e de alta relevância quando encontrada. A Figura 6 mostra as média das acurácias obtidas a partir de 10 iterações, também referentes às notícias da empresa Apple.

Algoritmo	Token	Bigrama	Trigrama
AdaBoost	46,92%	49,42%	47,11%
Árvore de Decisão	45,70%	47,30%	49,55%
Gradient Boosting	52,69%	50,25%	49,10%
K Vizinhos Mais Próximos	49,93%	51,15%	46,53%
Regressão Logística	54,10%	50,83%	50,38%
Floresta Aleatória	52,94%	47,88%	51,73%

Tabela 6 - Acurácia do preditor de sentimentos utilizando token, bigrama e trigrama com modelo de contagem de termos por TF-IDF . Fonte: Autoria própria

Com base nos resultados das tabelas Tabela 5 e a Tabela 6, podemos obter um panorama dos diferentes modelos e suas diferentes entradas. Os resultados completos, para 100 iterações, podem ser vistos na seção de Resultados (Seção 5.1).

Além disso, considerando que o grande número de dimensões dos conjuntos de dados aumenta a complexidade das técnicas de manipulação e degrada o desempenho dos algoritmos de mineração de dados, segundo Davis Une Miyashiro, Maria Camila Nardini Barioni (2009), sendo um processo de extração de atributos que busca alterar a representação de um conjunto, de maneira que a nova representação apresente uma dimensão menor que a original, porém mantendo as características inerentes das informações nela contida, foi testada a redução de dimensionalidade do vetor de atributos do analisador de sentimentos, treinado com também com TF-IDF.

Suas dimensões foram reduzidas para 100 e 200 valores, em dois testes separados. Uma número considerável, visto que o vetor da empresa Apple chegou a alcançar 5368 valores distintos. O método escolhido foi o de decomposição em valores singulares truncada (truncated singular value decomposition - SVD). Todavia, os resultados não apresentaram melhora significativa num primeiro experimento, como mostra a Tabela 7 com resultados para a empresa Apple. Os resultados completos podem ser vistos na seção de Resultados (seção 5.1.2.1).

Algoritmo	100 valores	200 valores
AdaBoost	49,53%	50,19%
Árvore de Decisão	48,69%	49,07%
Gradient Boosting	53,67%	52,37%
K Vizinhos Mais Próximos	52,53%	49,95%
Regressão Logística	52,78%	54,37%
Floresta Aleatória	52,49%	51,23%

Tabela 7 - Acurácia do preditor de sentimentos utilizando vetores de 100 e de 200 valores. Fonte: Autoria própria

Por fim, foi realizado um terceiro experimento consistindo de uma rede neural com cinco camadas. Uma rede rede recebendo 100, outra 200 e a terceira 500 atributos na primeira camada, números que significam as dimensões do vetor de termos textuais únicos, após uma redução em sua dimensionalidade que determina aqueles com maior relevância. Já as camadas seguintes recebem todas igualmente 50, 25, 10 e 1 atributos. A função de ativação foi a Unidade Linear Retificada (Rectified Linear Unit, ReLU) para todas as camadas, exceto a última, que utiliza função sigmóide. O treino foi feito com 100 épocas. Seus resultados completos também estão disponíveis em integridade na seção Resultados (seção 5.1.2.2), porém a Tabela 8 resume a saída da empresa Apple.

Entrada	Acurácia	Desvio Padrão
100 Valores	45,40%	3,38%
200 Valores	47,32%	3,01%
500 Valores	46,98%	3,13%

Tabela 8 - Acurácia do preditor de sentimentos utilizando redes neurais com 100, 200 e 500 atributos na primeira camada. Fonte: Autoria própria

Mesmo após variadas formas de testes, nenhum modelo atingiu uma acurácia desejável acima de 70%. Desta forma, foi definido que cada empresa utilizaria o modelo que lhe trouxesse a melhor performance, sendo Regressão Logística para Apple com 54,94% de acurácia, Gradient Boosting para Microsoft com 56,03% e Árvore de Decisão para Tesla com 56,95%, utilizando todos estes a vetorização em TF-IDF com n-grama onde n é igual a 1. Na seção Resultados (seção 5.1) esses aspectos são discutidos em detalhe.

4.1.4 Predição de valores de ações

Tendo estabelecido um modelo para análise de sentimentos para cada empresa, faz-se necessário organizar os dados coletados relativos às ações em um formato adequado para ser treinado pelo modelo preditor de valor de ações.

Como a modelagem de dados relativos às ações considera períodos diários e as notícias coletadas podem ter tempos distintos, para fazer a correlação entre o sentimento das notícias e valores de fechamento dos dados históricos e treinar o modelo, as notícias coletadas foram pré-processadas, com o intuito de identificar a tendência de sentimento das notícias de uma determinada empresa para um determinado dia. Neste caso, para o treinamento foram utilizados apenas os dias de fechamento da bolsa nos quais existiam notícias registradas no banco de dados histórico, e para cada dia em que há fechamento de valores de ações, foi calculada a tendência de sentimento do conjunto de notícias levantadas.

O cálculo deste valor foi feito a partir de um somatório do número de ocorrência de cada sentimento. Dessa forma, em um dia no qual foram registradas quatro notícias, das quais três são positivas e uma neutra, é considerado um dia positivo, por exemplo. Abaixo apresentamos as correlações utilizadas para identificar o sentimento predominante em um dia:

Sendo a quantidade de notícias positivas representado por **QP**, a quantidade de notícias negativas representado por **QN** e a quantidade de notícias neutras representado por **QNE**, segue a definição.

Dia positivo: **QP** > **QN** e **QP** > **QNE**

Dia neutro: **QNE** > **QP** e **QNE** > **QN**

Dia negativo: **QN** > **QP** e **QN** > **QNE**

Uma vez obtidas/inferidas/calculadas as tendências de sentimentos para cada dia em que há notícias capturadas e valores de fechamento de ação, é feito um mapeamento dos valores de ações relativos a cada um destes dias. A Tabela 9 demonstra este formato, com dados fictícios.

Data	Abertura	Fechamento	Alta	Baixa	Volume	Sentimento
10/05/2019	469,50	469,88	473,22	469,00	1400	1
11/05/2019	469,88	469,70	470,00	469,15	300	0

Tabela 9 - Representação do modelo com dados concatenados, valores de ação e média de sentimentos de notícias. Fonte: Autoria própria

Em seguida, os dados da tabela relativos às ações foram normalizados para que se encontrassem dentro de um intervalo de 0 a 1. O objetivo é diminuir ruídos do conjunto de dados, buscando um melhor treinamento, além de potencialmente acelerar o tempo de processamento. O método escolhido foi Min-Max, que determina para cada valor do conjunto sua proporção entre o maior e o menor valor do conjunto. Num exemplo de quatro valores como (20, 30, 40, 60), teríamos os valores normalizados de (0, 0,25, 0,5, 1), calculados conforme a fórmula representada na Figura 12.

$$x'_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

Figura 12 - Representação matemática do algoritmo Min-Max. Adaptado. Fonte: Abdelaziz (2018)

Embora buscássemos melhor performance com a normalização, muitos algoritmos tiveram seus resultados diminuídos, como podemos observar na Tabela 10 e por este motivo optamos por não fazer a normalização dos dados. Desta forma, a normalização deixou de ser utilizada.

Algoritmo	Acurácia s/ normalização	Acurácia c/ normalização
K Vizinhos Mais Próximos	98.19%	96.16%
Regressão Logística	93.64%	89.87%
Floresta Aleatória	95.96%	95.22%

Tabela 10 - Comparação dos algoritmos com dados normalizados e não normalizados. Fonte: Autoria própria

Neste ponto do processo, o atributo data, demonstrado na Tabela 9 foi desconsiderado, visto que ele só é necessário para efetuar a correlação dos valores relativos à ação e às notícias de um mesmo dia. Sendo assim, os dados estão adequados para treinamento, necessitando então do valor de classificação para cada tupla do conjunto de dados. Este valor foi calculado a partir dos valores de abertura e de fechamento da bolsa para cada dia: a proporção do fechamento da bolsa em relação à abertura do dia atual será considerada positiva se for maior ou igual a 3%, negativa se for menor ou igual a -3% e neutra quando se encontra entre 2.99% e -2.99%. Ressaltamos que estes valores foram

definidos com caráter experimental, sendo viável a verificação de diferentes intervalos. A Tabela 11 ilustra essa classificação com dados fictícios.

Abertura do Dia	Fechamento do Dia	Diferença	Classificação
1023	953	7,34%	1
953	971	-1,85%	0
971	840	13,49%	-1

Tabela 11 - Exemplo de classificação de tendência de valor de ação para um dia em positivo, negativo ou neutro. Fonte: Autoria própria

Neste ponto, com a tendência de sentimentos de cada dia, temos os dados prontos para treinamento por parte do preditor de valores de ação. Na Tabela 12 podemos verificar três exemplos hipotéticos de como se apresenta o conjunto de dados, sendo a coluna Tendência da Ação o valor usado para classificação das demais colunas que se caracterizam como atributos. Este mesmo formato foi utilizado para experimentar diferentes algoritmos para o modelo preditor de ações. Os resultados completos se encontram na seção Resultados (seção 5.2).

Abertura	Fechamento	Alta	Baixa	Volume	Tendência de Sentimento	Tendência da Ação
400	390	405	380	0,632	1	0
390	402,5	405	385	0,527	1	1
402,5	385	404	375	0,688	-1	-1

Tabela 12 - Exemplo de dados completos do modelo preditivo, valores relativos à ação, tendência de sentimento de notícia (média do dia) e tendência da ação. Fonte: Autoria própria

Todos os resultados de acurácia foram maiores que 89%, alguns atingindo inclusive 99%. Supomos que este valor fosse muito alto, devido ao fato de que trabalhos correlatos não alcançaram uma marca tão elevada mesmo utilizando técnicas mais avançadas, não só para predição como também para preparação dos dados. A partir disso, supomos que o motivo da alta acurácia seria devido à margem de diferença entre a abertura e o fechamento diário, estando muito grande para a real tendência dos dados e assim sendo grande parte deles classificada como uma só classe.

A fim de explorar esta hipótese, executamos testes com mais duas configurações de proporção. Uma define como tendência positiva se a proporção for maior ou igual a 1%, negativa se for menor ou igual a -1% e neutra quando se encontra entre 0,99% e -0,99%. Outra foi definida como positiva se maior ou igual a 0,3%, negativa se menor que -0,3% e neutra se estiver entre 0,29% e -0,29%. A tabela Tabela 13 traz uma prévia destes dados

para a empresa Microsoft. Para facilitação, vamos chamar a medida original de Medida 3%, a segunda de Medida 1% e a última definida como Medida 0,5%. Ressaltamos que os dados completos podem ser visualizados na seção de Resultados (seção 5.2).

Algoritmo	Medida 3%	Medida 1%	Medida 0,3%
Árvore de Decisão	99,71%	87,35%	75,12%
K Vizinhos Mais Próximos	98,23%	75,67%	57,70%
Regressão Linear	99,71%	76,79%	52,94%

Tabela 13 - Comparação de valores de acurácia para diferentes critérios de seleção de tendência do valor das ações. Fonte: Autoria própria

Como podemos observar, ao reduzir a medida da janela de categorização das diferenças, a acurácia foi reduzida, uma vez que está mais próxima dos dados reais e deixa de classificar a maior parte dos dados como uma só categoria.

5 Resultados

A seção anterior (Seção 4) tratou dos detalhes de pré processamento e treinamento dos modelos. Na seção atual detalharemos os resultados obtidos por cada modelo e seus parâmetros. Primeiramente serão trazidos os resultados completos do analisador de sentimentos, abordando todos os testes realizados em suas diferentes possibilidades: vetorizador de frequência simples, TF-IDF, com tokens, bigramas e trigramas, com redução de dimensionalidade e também redes neurais. A acurácia em todas as tabelas a seguir são referentes à média da acurácia de 100 rodadas de treinamento e teste para cada algoritmo, com o intuito de minimizar efeitos de aleatoriedade na partição dos dados de treino e teste.

Em seguida, são trazidos os resultados relativos ao modelo preditor de ações, com todos os algoritmos experimentados e seus parâmetros. Da mesma forma, a acurácia de suas tabelas são referentes à média de 100 iterações de treinamento, assim como os respectivo desvios padrões.

5.1 Analisador de sentimentos

Consideramos que nenhum algoritmo alcançou uma performance satisfatória, para nenhuma das três empresas estudadas, visto que sequer alcançaram 70% de acurácia. Entretanto, pudemos observar que as empresas para as quais existiam menos notícias no banco de dados obtiveram um maior desvio padrão. Os resultados da empresa Apple, com 519 notícias, mantiveram um desvio padrão máximo de 6,54%, enquanto a Microsoft 176 notícias obteve até 9,51% e Tesla com 76 notícias atingiu o máximo de 14,52%.

5.1.1 Vetorizador de frequência simples

As Tabela 14 a seguir representa a média dos 100 resultados do analisador de sentimentos para a empresa Apple utilizando a estratégia de vetorização de frequência simples dos elementos token, bigrama e trigrama. Embora os resultados de todas as estratégias tenham sido próximos, podemos ver que a melhor estratégia neste cenário é de utilizar o algoritmo de regressão logística com tokens, que alcançou 53,46% de acurácia nos testes.

Algoritmo	**Token**		**Bigrama**		**Trigrama**	
	Acurácia	**Desvio Padrão**	**Acurácia**	**Desvio Padrão**	**Acurácia**	**Desvio Padrão**
AdaBoost	45,06%	2,81%	49,42%	3,48%	45,19%	3,32%
Árvore de Decisão	45,32%	2,67%	47,30%	1,95%	49,80%	1,25%
Gradient	52,21%	2,21%	50,25%	3,55%	47,43%	3,71%

Boosting						
K Vizinhos Mais Próximos	42,37%	4,05%	51,15%	3,97%	45,89%	6,54%
Regressão Logística	53,46%	3,49%	50,83%	3,39%	50,06%	2,53%
Floresta Aleatória	52,95%	4,87%	47,88%	3,48%	49,16%	2,65%

Tabela 14 - Resultado do analisador de sentimentos para empresa Apple com vetorização de frequência simples. Fonte: Autoria própria

A Tabela 15 abaixo segue o mesmo padrão da tabela anterior, representando a média a média dos 100 resultados do analisador de sentimentos utilizando a estratégia de vetorização de contagem simples da frequência dos elementos token, bigrama e trigrama, porém para a empresa Microsoft. Neste caso, a melhor acurácia se dá pelo algoritmo de árvore de decisão, alcançando 53,39% de acurácia.

Algoritmo	**Token**		**Bigrama**		**Trigrama**	
	Acurácia	**Desvio Padrão**	**Acurácia**	**Desvio Padrão**	**Acurácia**	**Desvio Padrão**
AdaBoost	45,60%	7,63%	50,56%	4,61%	45,47%	6,16%
Árvore de Decisão	46,98%	7,28%	49,05%	2,92%	53,39%	3,04%
Gradient Boosting	51,70%	7,16%	51,32%	3,93%	50,56%	6,79%
K Vizinhos Mais Próximos	51,70%	4,23%	45,47%	7,47%	42,07%	9,51%
Regressão Logística	53,01%	6,61%	51,88%	5,61%	52,64%	7,71%
Floresta Aleatória	47,73%	6,80%	47,92%	6,54%	41,88%	9,23%

Tabela 15 - Resultado do analisador de sentimentos para empresa Microsoft com vetorização de frequência simples. Fonte: Autoria própria

Já a Tabela 16 abaixo trata da empresa Tesla, seguindo também o padrão das tabelas anteriores desta seção: média dos 100 resultados do analisador de sentimentos utilizando a estratégia de vetorização de contagem simples da frequência dos elementos token, bigrama e trigrama. Para este caso, destaca-se em acurácia o modelo de árvore de decisão com tokens como entrada, atingindo a marca de 54,78% de acurácia. Curiosamente o mesmo algoritmo, porém com entrada de bigramas, alcançou uma acurácia consideravelmente mais baixa, de 36,52%.

Algoritmo	Token		Bigrama		Trigrama	
	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão
AdaBoost	44,78%	12,75%	40,43%	9,13%	47,82%	8,91%
Árvore de Decisão	54,78%	9,76%	36,52%	7,82%	49,13%	8,26%
Gradient Boosting	50%	14,52%	46,08%	7,06%	51,73%	8,12%
K Vizinhos Mais Próximos	42,60%	11,99%	45,65%	8,07%	40%	8,65%
Regressão Logística	50,86%	6,44%	50%	7,07%	43,04%	11,07%
Floresta Aleatória	47,82%	4,78%	46,08%	8,95%	39,56%	3,91%

Tabela 16 - Resultado do analisador de sentimentos para empresa Tesla com vetorização de frequência simples. Fonte: Autoria própria

5.1.2 Vetorizador TF-IDF

A Tabela 17 a seguir representa a média dos 100 resultados do analisador de sentimentos para a empresa Apple, no mesmo formato em que a seção 5.1.1, porém utilizando a estratégia de vetorização de TF-IDF dos elementos token, bigrama e trigrama. Para este tipo de vetorização, percebemos que o modelo de regressão logística com tokens se saiu melhor nos testes, com 54,94% de acurácia.

Algoritmo	Token	Bigrama	Trigrama

	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão
AdaBoost	46,92%	3,68%	49,42%	3,21%	47,11%	3,41%
Árvore de Decisão	45,70%	4,56%	47,30%	3,16%	49,55%	2,04%
Gradient Boosting	44,30%	4,43%	50,25%	4,51%	49,10%	2,73%
K Vizinhos Mais Próximos	49,93%	3,96%	51,15%	2,93%	46,53%	3,61%
Regressão Logística	54,94%	3,34%	50,83%	4,90%	50,38%	2,26%
Floresta Aleatória	52,94%	4,13%	47,88%	4,87%	51,73%	4,96%

Tabela 17 - Resultado do analisador de sentimentos para empresa Apple com vetorização por TF-IDF.
Fonte: Autoria própria

A Tabela 18 a seguir permanece no padrão da tabelas anterior, trazendo a média a média dos 100 resultados do analisador de sentimentos utilizando a estratégia de vetorização TF-IDF dos elementos token, bigrama e trigrama, exceto que neste caso os dados são referentes à empresa Microsoft. Com os dados desta empresa, o modelo que atingiu a melhor performance foi gradient boosting, com 56,03% de acurácia utilizando tokens como entrada.

Algoritmo	Token		Bigrama		Trigrama	
	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão
AdaBoost	44,52%	6,91%	48,86%	5,43%	45,09%	9,18%
Árvore de Decisão	47,73%	6,53%	50,56%	5,18%	51,32%	4,20%
Gradient Boosting	56,03%	5,40%	48,86%	4,88%	55,28%	3,68%
K Vizinhos Mais Próximos	52,45%	6,18%	48,30%	7,92%	44,52%	9,21%
Regressão	53,96%	5,21%	54,52%	4,73%	49,81%	7,16%

Logística						
Floresta Aleatória	51,50%	8,05%	48,11%	4,81%	43,77%	6,94%

Tabela 18 - Resultado do analisador de sentimentos para empresa Microsoft com vetorização por TF-IDF. Fonte: Autoria própria

A Tabela 19 que se segue trata da empresa Tesla, apresentando, no mesmo padrão das tabelas anteriores desta seção, a média dos 100 resultados do analisador de sentimentos utilizando a estratégia de vetorização TF-IDF dos elementos token, bigrama e trigrama. A árvore de decisão se destaca como o melhor algoritmo para a empresa Tesla, que, novamente com entradas de token, atingiu 56,95% de acurácia.

Algoritmo	**Token**		**Bigrama**		**Trigrama**	
	Acurácia	**Desvio Padrão**	**Acurácia**	**Desvio Padrão**	**Acurácia**	**Desvio Padrão**
AdaBoost	37,82%	6,74%	43,04%	7,39%	43,04%	9,00%
Árvore de Decisão	56,95%	10,00%	42,60%	8,43%	48,26%	7,88%
Gradient Boosting	53,47%	5,84%	43,91%	9,21%	49,56%	11,53%
K Vizinhos Mais Próximos	50,86%	10,65%	46,52%	7,28%	38,69%	11,24%
Regressão Logística	46,08%	12,17%	48,26%	4,93%	40,86%	8,06%
Floresta Aleatória	50,86%	10,11%	41,73%	5,21%	48,69%	9,28%

Tabela 19 - Resultado do analisador de sentimentos para empresa Tesla com vetorização por TF-IDF. Fonte: Autoria própria

5.1.2.1 Redução de dimensionalidade

Esta seção traz os resultados relativos aos diferentes algoritmos testados com o vetorizador TF-IDF, com a redução da dimensionalidade dos atributos de entrada. A Tabela 20 é referente à dimensão de 100 atributos para Apple, Microsoft e Tesla, respectivamente,

enquanto a Tabela 21 trata de 200 atributos, respectivamente para as mesmas empresas. Como podemos observar, os dados não demonstraram variações significativas em relação às entradas com dimensões originais.

Algoritmo	Apple		Microsoft		Tesla	
	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão
AdaBoost	49,53%	3,88%	47,83%	5,58%	37,08%	11,16%
Árvore de Decisão	48,69%	4,29%	47,54%	7,07%	44,00%	10,49%
Gradient Boosting	53,67%	3,73%	50,96%	5,95%	45,73%	9,75%
K Vizinhos Mais Próximos	52,38%	3,70%	50,67%	6,44%	47,21%	10,04%
Regressão Logística	52,78%	3,59%	52,15%	6,40%	50,65%	11,64%
Floresta Aleatória	52,49%	3,59%	50,24%	6,98%	44,30%	10,06%

Tabela 20 - Resultado do analisador de sentimentos para todas as empresas, com vetorização por TF-IDF utilizando 100 atributos. Fonte: Autoria própria

Algoritmo	Apple		Microsoft		Tesla	
	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão
AdaBoost	50,19%	3,96%	47,05%	5,85%	38,69%	11,17%
Árvore de Decisão	49,07%	4,15%	47,26%	6,49%	43,86%	11,58%
Gradient Boosting	52,37%	3,35%	51,16%	6,89%	46,52%	9,61%
K Vizinhos Mais Próximos	49,95%	3,70%	51,64%	6,04%	47,13%	8,73%
Regressão	54,37%	3,97%	50,39%	6,79%	49,86%	10,53%

Logística						
Floresta Aleatória	51,23%	3,81%	49,79%	6,09%	42,95%	10,16%

Tabela 21 - Resultado do analisador de sentimentos para todas as empresas, com vetorização por TF-IDF utilizando 200 atributos. Fonte: Autoria própria

5.1.2.2 Redes Neurais

Esta seção traz os resultados completos referentes aos testes do analisador de sentimentos utilizado o modelo de rede neural, onde foram realizados testes também com 100 e 200 valores de entrada na primeira camada. Respectivamente para as empresas Apple, Microsoft e Tesla, podemos ver estes resultados na Tabela 22. Novamente não se obtém resultados significativos com este experimento. Não obstante, nota-se que a acurácia até mesmo se diminuiu em relação a outros modelos e entradas.

Entrada	**Apple**		**Microsoft**		**Tesla**	
	Acurácia	**Desvio Padrão**	**Acurácia**	**Desvio Padrão**	**Acurácia**	**Desvio Padrão**
100 Valores	45,40%	3,38%	50,45%	6,41%	47,04%	9,76%
200 Valores	47,32%	3,01%	50,24%	6,36%	48,65%	8,65%
500 Valores	46,98%	3,13%	51,13%	6,39%	50,60%	9,79%

Tabela 22 - Resultado do analisador de sentimentos para todas as empresas utilizando redes neurais. Fonte: Autoria própria

5.1.3 Discussão e comparações com trabalhos relacionados

Como pode ser observado pelos dados das tabelas apresentadas na seção 5.1, embora experimentos com diversos algoritmos e parâmetros tenham sido feitos, não se obteve uma acurácia alta como foi o caso dos trabalhos correlatos. Foram utilizados diferentes tipos de vetorização, como um de frequência simples e também TF-IDF, além redução de dimensionalidade, e até mesmo o uso de redes neurais.

Sendo assim, os modelos selecionados para cada empresa foram aqueles que apresentaram melhor acurácia, independentemente do tipo de algoritmo e parâmetros

utilizados. Regressão Logística para Apple, com 54,9% de acurácia, Gradient Boosting para Microsoft, com 56,03% de acurácia e Árvore de Decisão para Tesla, com 56,95% de acurácia. Todos estes recebem como entrada um vetor de parâmetros no formato TF-IDF que considera tokens únicos no texto.

Segundo Elagamy, Stanier e Sharp (2018), através de um modelo baseado em Florestas Aleatórias com entrada de tokens foi possível obter a acurácia de 98,34%, classificando corretamente 535 de 544 artigos. Já para bigramas, a acurácia alcançada foi de 98,89%, acertando a classe de 538 de 544 artigos.

Estes dados demonstram que o analisador de sentimentos deste trabalho não alcançou uma performance desejável, compreensível também pelo fato de que uma acurácia próxima de 50% não está muito longe das chances de acerto de um mero palpite.

5.2 Preditor de ações

Abaixo serão apresentados os resultados dos diferentes algoritmos usados para a predição dos valores da bolsa. As tendências de sentimentos diários que serve de entrada para este preditor foram calculadas utilizando o algoritmo que obteve a melhor acurácia para cada empresa. Foram feitos 100 ciclos de treinamento e testes, a fim de minimizar efeitos de aleatoriedade, sendo o valor de acurácia das tabelas a seguir a média de todas os ciclos. Na Tabela 23 podemos ver os resultados para as empresas Apple, Microsoft e Tesla.

Algoritmo	**Apple**		**Microsoft**		**Tesla**	
	Acurácia	**Desvio Padrão**	**Acurácia**	**Desvio Padrão**	**Acurácia**	**Desvio Padrão**
AdaBoost	91,87%	2,74%	99,77%	0,78%	97,48%	03,67%
Árvore de Decisão	97,20%	2,04%	99,28%	1,35%	97,42%	03,80%
Gradient Boosting	98,80%	1,40%	99,76%	0,78%	97,93%	03,64%
K Vizinhos Mais Próximos	90,08%	1,68%	98,09%	1,37%	96,03%	03,07%
Regressão Logística	89,89%	1,79%	98,12%	1,38%	96,29%	03,16%
Floresta Aleatória	96,00%	2,06%	99,57%	1,01%	97,30%	03,41%

Tabela 23 - Resultado do preditor de valor de ação para todas as empresas, considerando tendência de sentimento de notícias. Fonte: Autoria própria

Todos os modelos obtiveram uma acurácia alta, atingindo o mínimo de 89,95%, porém diversos alcançaram até mesmo 99%. Considerando que os analisadores de sentimentos utilizados obtiveram 54,94%, 56,03% e 56,95% de acurácia para Apple, Microsoft e Tesla respectivamente, números relativamente incertos, a eficácia do valor da tendência dos sentimentos neste modelo foi questionada. Com o propósito de fazer comparações, foi realizado um novo teste, porém ignorando totalmente os valores de tendência de sentimento das notícias. As colunas consideradas neste caso são: Abertura, Fechamento, Alta, Baixa e Volume. Os resultados obtidos estão expostos na Tabela 24.

Algoritmo	**Apple**		**Microsoft**		**Tesla**	
	Acurácia	**Desvio Padrão**	**Acurácia**	**Desvio Padrão**	**Acurácia**	**Desvio Padrão**
AdaBoost	93,02%	2,09%	99,67%	0,91%	96,50%	3,07%
Árvore de Decisão	97,46%	1,48%	99,61%	1,08%	97,29%	3,81%
Gradient Boosting	98,72%	1,16%	99,72%	0,96%	97,10%	3,15%
K Vizinhos Mais Próximos	89,93%	2,00%	97,99%	1,29%	96,12%	3,06%
Regressão Logística	89,78%	1,99%	98,02%	1,30%	97,17%	4,00%
Floresta Aleatória	97,24%	1,83%	99,28%	1,24%	97,71%	3,90%

Tabela 24 - Resultado do preditor de valor de ação para todas as empresas, desconsiderando tendência de sentimento de notícias. Fonte: Autoria própria

Como podemos observar, ao remover os valores de sentimentos diários, a acurácia e o desvio padrão se mantiveram muito semelhantes para todos os modelos. Este fator ajudou a levantar a hipótese de que o número para separar a proporção entre a abertura e o fechamento da ação num dia fosse muito grande, conforme detalhado na seção Predição de valores de ações (seção 4.1.4). As Tabelas 23 e 24 utilizaram a seguinte regra para classificar a tendência da ação em um dia:

Positiva se a proporção for maior ou igual a 3%;
Negativa se for menor ou igual a -3%;

Neutra quando se encontra entre 2,99% e -2,99%.

Sendo assim, a Tabela 25 a seguir representa da seguinte classificados da seguinte forma:

Positiva se a proporção for maior ou igual a 1%;
Negativa se for menor ou igual a -1%;
Neutra quando se encontra entre 0,99% e -0,99%.

Algoritmo	Apple		Microsoft		Tesla	
	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão
AdaBoost	73,96%	5,08%	81,20%	8,02%	95,55%	3,96%
Árvore de Decisão	87,45%	4,43%	87,35%	5,16%	95,97%	4,25%
Gradient Boosting	91,15%	3,13%	90,34%	4,30%	96,21%	4,00%
K Vizinhos Mais Próximos	5,05%	5,70%	75,67%	5,42%	95,33%	40,35%
Regressão Logística	47,81%	3,92%	76,79%	4,65%	95,44%	3,95%
Floresta Aleatória	85,72%	4,78%	87,77%	4,78%	95,96%	4,17%

Tabela 25 - Resultado do preditor de valor de ação com classificação de tendência utilizando diferença entre faixas de 1%. Fonte: Autoria própria

Foi realizado ainda mais um experimento, cujos resultados são encontrados na Tabela 26. As configurações utilizadas foram as seguintes:

Positiva se a proporção for maior ou igual a 0,3%;
Negativa se for menor ou igual a -0,3%;
Neutra quando se encontra entre 0,29% e -0,29%.

Algoritmo	Apple	Microsoft	Tesla

	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão	Acurácia	Desvio Padrão
AdaBoost	71,04%	5,01%	65,98%	7,03%	94,20%	4,29%
Árvore de Decisão	87,89%	4,45%	75,12%	7,06%	94,50%	4,35%
Gradient Boosting	93,13%	3,25%	80,00%	5,77%	94,15%	4,64%
K Vizinhos Mais Próximos	52,30%	4,56%	57,70%	76,75%	90,04%	86,77%
Regressão Logística	52,52%	3,41%	52,94%	6,12%	87,66%	5,68%
Floresta Aleatória	88,52%	4,25%	77,77%	5,88%	93,78%	4,85%

Tabela 26 - Resultado do preditor de valor de ação com classificação de tendência utilizando diferença entre faixas de 0,3%. Fonte: Autoria própria

Podemos observar que os resultados de acurácia são menores conforme menor a janela utilizada para separar as tendências diárias em positiva, neutra ou negativa. Isto abre espaço para maiores explorações em trabalhos futuros. Curiosamente os resultados para a empresa Tesla permaneceram elevados.

5.2.1 Comparações com trabalhos relacionados

Araújo (2010) afirma que o mercado de ações pode ser classificado pela teoria do caminho aleatório, que assume que os preços são completamente estocásticos e por natureza se comportam de maneira imprevisível e sem influência do passado. Isto significa que qualquer predição acima de 50% pode ser considerada vantajosa, visto que está acima da margem de um palpite.

Dentre os trabalhos relacionados, a pesquisa de Araújo (2010) se mostrou a mais sofisticada no assunto de ações, apresentando uma rede neural com algoritmo genético e parâmetros mais avançados que os utilizados neste trabalho, o que lhe garantiu uma acurácia de 67,77%, cerca de 30% a menos que os resultados obtidos na primeira tentativa desta pesquisa.

Esta diferença foi um dos fatores que nos levou a considerar que algo pudesse ser aprimorado, fazendo com que fizéssemos testes com diferentes medidas de classificação de um dia positivo, neutro ou negativo para a bolsa de valores, conforme detalhado na seção anterior (seção 5.2). Conforme se estreita a margem de diferença, maior se torna o erro nos testes. Todavia, acreditamos também que um número maior de dados seria benéfico, permitindo o melhor ajuste de pesos e distâncias dos algoritmos de classificação.

A solução proposta por Abdullah, et al. (2013) utilizou, dentre diversos algoritmos, o de árvore de decisão, também utilizado neste trabalho. Seus resultados de acurácia chegaram na máxima de 89.98%, em contraste a 97,46%, 99,61% e 97,29% para Apple, Microsoft e Tesla alcançados na primeira tentativa desta pesquisa. Este fato reforça a necessidade de ajustar os parâmetros de entrada do modelo. Contudo, ressaltamos a afirmação da teoria do caminho aleatório, mais uma vez levantando a possibilidade de que também estes resultados possam ser tendenciosos. Além disso, vale considerar que não somente diferentes empresas foram estudadas entre um trabalho e outro, mas também que seus valores foram atrelados a outro índice, em outra região geográfica.

5.3 Tempo de treinamento

Considerando que foram testados diversos algoritmos com diferentes tipos de entrada, os resultados referentes aos tempo de treinamento para cada combinação serão listados nesta seção para se manterem centralizados. Todos os testes foram realizados na mesma máquina, com a seguinte configuração:

- **Armazenamento**: Samsung SSD 860 EVO 250GB;
- **RAM**: 2 módulos de Crucial Ballistix Sport AT Gray 8GB, 2666MHz (2933MHz Overclock), DDR4;
- **Processador central**: AMD Ryzen 5 2600X, 6 núcleos, 3.6GHz (4.25GHz Max Turbo), processando em Dual Channel;
- **Processador gráfico**: MSI GeForce GTX 1060 IGAMER OC, processador com clock de 1759 MHz, memória com 6GB de tamanho, 8008MHz de clock e tipo GDDR5 de 192-bit.
- **Sistema operacional**: Windows 10 Education 64 bits, versão 1803, build 17134.675.

Embora os resultados de acurácia previamente apresentados sejam referentes a 100 iterações de treinos e testes, os resultados de tempo apresentados nesta seção dizem respeito a iteração única. Faz-se relevante advertir que somente os algoritmos de redes neurais foram processados pelo processador gráfico, todos os restantes foram calculados utilizando o processador central.

O conjunto de dados não é considerado grande pelos autores, visto que está na casa das dezenas de megabytes. Entretanto, entendemos os tempos de execução aceitáveis para todos os algoritmos e suas configurações, até mesmo para as redes neurais que por natureza são custosas. Maiores detalhes são descritos em cada subseção a seguir.

5.3.1 Remoção de símbolos especiais e stemming

Iniciamos com a Tabela 27, representando os tempos de execução para a remoção de símbolos especiais e stemming dos textos separadamente, empresa por empresa. Os tempos de execução mostram um crescimento conforme o tamanho do conjunto de dados, sendo o número de notícias da empresa Apple o maior com 519 notícias, seguido por Microsoft com 176 e por último Tesla com 76.

Tempo (segundos)	Remoção de símb. especiais	Stemming
Apple	0.04997730255126953	1.8583908081054688
Microsoft	0.015004158020019531	0.7311661243438721
Tesla	0.004000663757324219	0.2600405216217041

Tabela 27 - Resultado de tempo de execução da remoção de símbolos especiais e aplicação do método de stemming nos textos das notícias. Fonte: Autoria própria

5.3.2 Vetorizador de contagem simples

A Tabela 28 traz os dados relativos à vetorização das notícias para o vetorizador de contagem simples. Conforme as colunas da tabela, vemos que foram considerados três tipos de processamento: por tokens, por bigramas e por trigramas. Novamente se vê um número maior para a empresa Apple em relação às outras. Também se percebe o crescimento do tempo conforme o número do n-grama aumenta. Entretanto, todos os tempos foram considerados extremamente rápidos pelos autores.

Tempo (segundos)	Token	Bigrama	Trigrama
Apple	0.0430147647857666	0.08101868629455566	0.09628033638000488
Microsoft	0.0180041790008549	0.03200769424438477	0.03500819206237793
Tesla	0.0083739757537841 8	0.01100373268127441	0.01200056076049805

Tabela 28 - Resultado de tempo de execução da vetorização de contagem simples. Fonte: Autoria própria

5.3.3 Vetorizador TF-IDF

A seguir, os resultados da vetorização pela estratégia TF-IDF são demonstrados na Tabela 29. O padrão dos dados é o mesmo encontrado para a vetorização por contagem simples.

Tempo (segundos)	Token	Bigrama	Trigrama
Apple	0.04500985145568848	0.08701825141906738	0.09400463104248047

Microsoft	0.02002382278442383	0.03300738334655762	0.03499293327331543
Tesla	0.00800132751464844	0.01199698448181152	0.01400327682495117 2

Tabela 29 - Resultado de tempo de execução da vetorização TF-IDF. Fonte: Autoria própria

5.3.4 Redução de dimensionalidade

Os tempos de execução do algoritmo de redução de dimensionalidade também foram calculados, podendo ser conferidos na Tabela 30 abaixo. Mais uma vez, devido ao maior número de dados de entrada, os tempos para a empresa Apple são maiores que os tempos das outras empresas.

Tempo (segundos)	100 Valores	200 Valores	500 Valores
Apple	0.13703083992004395	0.28606438636779785	0.6380016803741455
Microsoft	0.08019089698791504	0.09134244918823242	0.1290283203125
Tesla	0.01900506019592285	0.02856230735778809	0.04259347915649414

Tabela 30 - Resultado do tempo de execução da dimensionalidade, utilizando 100, 200 e 500 valores. Fonte: Autoria própria

5.3.5 Analisador de sentimentos

A Tabela 30 traz o tempo utilizado pelo treinador de sentimentos, sendo que cada número representa a soma do tempo de treino de 70% dos dados com o tempo de predição de 30% dos dados. Nota-se que o tempo necessário pela rede neural é várias vezes maior que qualquer outro algoritmo, mesmo sendo calculado utilizando a placa de processamento gráfico, ao contrário dos restantes. Além disso, percebemos que o tempo necessário para o algoritmo Gradient Boosting é algumas vezes maior que os demais. Mesmo que possa ser considerado neste caso como curto, podemos inferir que com um volume muitas vezes maior de dados este tempo seria proporcionalmente maior.

Tempo (segundos)	Apple	Microsoft	Tesla
AdaBoost	0.511982202529907	0.083953857421875	0.039008378982544
Árvore de Decisão	0.066411256790161	0.006001710891724	0.001000642776489
Gradient Boosting	2.190494537353516	0.381987094879150	0.149033546447754

K Vizinhos Mais Próximos	0.028006553649902	0.003000736236572	0.001000165939331
Regressão Logística	0.019004583358765	0.005000829696655	0.002000570297241
Floresta Aleatória	0.028005838394165	0.010001182556152	0.008001565933227
Rede Neural	43.43654322624206	15.69668507575989	7.853874683380127

Tabela 31 - Resultado de tempo de execução do analisador de sentimentos para cada empresa, por cada algoritmo. Fonte: Autoria própria

5.3.6 Normalização de dados

A contagem de tempo para a normalização de dados foi realizada. Entretanto, o resultado retornado sempre foi igual a zero. Consideramos que o tempo foi excepcionalmente baixo, de modo a ser arredondado a zero em tempo de execução.

5.3.7 Preditor de ações

Assim como os resultados do analisador de sentimentos (seção 5.3.5), os resultados apresentados na Tabela 32 representam a soma do tempo de treinamento, utilizando 70% dos dados, com o tempo de predição, que usa 30%. Todos os resultados foram considerados velozes, com exceção do algoritmo Gradient Boosting, tendo uma visível diferença dentre os demais algoritmos, especialmente para o conjunto de dados da empresa Apple.

Tempo (segundos)	Apple	Microsoft	Tesla
AdaBoost	0.033006668090820	0.032006740570068	0.002000570297241
Árvore de Decisão	0.001008510589599	0.002001285552976	0.001000165939331
Gradient Boosting	0.105023622512817	0.015003681182861	0.014000654220581
K Vizinhos Mais Próximos	0.001000165939331	0.000424861907959	0.000985145568847
Regressão Logística	0.014003038406372	0.014002799987793	0.010002851486206
Floresta Aleatória	0.007003307342529	0.007009029388428	0.006001234054565

Tabela 32 - Resultado de tempo de execução do preditor de valor de ações cada empresa, por cada algoritmo. Fonte: Autoria própria

5.4 Validação do algoritmo

Apesar de todos os resultados adquiridos através dos testes realizados em cada um dos modelos utilizados na criação destes trabalho, foi decidido que um novo teste sobre os mesmos, funcionando em conjunto, poderiam agregar ainda mais informações úteis aos resultados obtidos.

Uma vez que estes resultados de treinamento foram adquiridos utilizam um único dataset de notícias e, pela limitação dos dias relacionados às notícias, consequentemente de valores de ação, cada um dividido em duas partes, para executar tanto o treinamento quanto os testes do modelo de predição proposto, foi idealizado um novo teste utilizando dados reais captados de forma manual, para confrontar com as saídas do modelo.

Para isso foram buscadas notícias de datas aleatórias do ano de 2019 (dois mil e dezenove) para cada uma das empresas. Foram levantadas notícias de dois dias para cada uma das empresa, cada uma das notícias foi classificada pelos autores deste de forma manual para ter um meio de comparação com a saída do modelo. Após classificadas as notícias, o texto principal (corpo da notícia) foi utilizado como entrada para o preditor de sentimentos do modelo, resultando em uma classe ([-1] negativa, [0] neutra e [1] positiva) atribuída a notícia. Para o desenvolvimento desta etapa de testes do trabalho, foi criada uma tabela com uma aba para cada empresa, nestas são apresentadas as novas notícias capturadas no formato original, os tratamentos de textos executados o resultado esperado da predição de sentimentos e o resultado predito, para cada uma das notícias. A Tabela 33 apresenta um exemplo de notícia presente nesta tabela.

Título	Notícia tratada	Data	Sentimento “manual”	Sentimento predito	Texto original (Json)
Invenção da Apple bloqueia iPhones, iPads e Macs roubados de suas lojas	"Ao longo do último ano, ocorrências de roubos em lojas da Apple estouraram ao redor dos Estados Unidos — o que não chega a ser uma surpresa, considerando a facilidade que malfeitores têm para furtar aparelhos em uma loja que conta com pouca (ou nenhuma) segurança e	2019-05-31	0	1	{'title': 'Invenção da Apple bloqueia iPhones, iPads e Macs roubados de suas lojas – MacMagazine.com.br', 'text': 'Ao longo do último ano, ocorrências de roubos em lojas da Apple estouraram ao redor dos Estados Unidos — o que não chega a ser uma surpresa, considerando a

	muitos dispositivos exibidos em meses para a degustação de clientes. A Maçã, entretanto, pode estar prestes a tomar uma atitude em relação a isso. Uma patente publicada recentemente pelo Escritório de Marcas e Patentes dos EUA (e descoberta pelo Patently Apple) descreve um novo sistema de segurança desenvolvido pela Apple para as suas lojas, envolvendo uma espécie de “cerca invisível” formada por ondas de rádiofrequência para detectar quando um aparelho do mostruário é furtado. O sistema é inteligente e reconhece quando o dispositivo roubado está se aproximando da “borda” da cerca, o que já dispara um alerta silencioso no sistema de segurança da loja; se o aparelho sai do perímetro, outro alerta é enviado; e, se o dispositivo				facilidade que malfeitores têm para furtar aparelhos em uma loja que conta com pouca (ou nenhuma) segurança e muitos dispositivos exibidos em meses para a degustação de clientes. A Maçã, entretanto, pode estar prestes a tomar uma atitude em relação a isso.\n\nUma patente publicada recentemente pelo Escritório de Marcas e Patentes dos EUA (e descoberta pelo Patently Apple) descreve um novo sistema de segurança desenvolvido pela Apple para as suas lojas, envolvendo uma espécie de “cerca invisível” formada por ondas de rádiofrequência para detectar quando um aparelho do mostruário é furtado.\n\nO sistema é inteligente e

	permanecer fora da cerca por mais que alguns minutos, ele é desativado e sua localização passa a ser transmitida para o sistema da loja. Essa desativação pode ocorrer de várias formas: em dispositivos touchscreen, por exemplo, a operação por toque pode ser desabilitada; já em aparelhos com botões ou teclados, as teclas podem deixar de funcionar. O aparelho passa a exibir em sua tela um aviso afirmando que ele foi levado da loja em questão, adicionando o número de telefone da Apple Store para que a equipe da loja ofereça instruções de como devolvê-lo. Caso o aparelho roubado seja recuperado, basta que a loja entre com as credenciais corporativas nele para que seu funcionamento volte ao normal. Obviamente, por se tratar de uma patente, não há como saber se a				reconhece quando o dispositivo roubado está se aproximando da “borda” da cerca, o que já dispara um alerta silencioso no sistema de segurança da loja; se o aparelho sai do perímetro, outro alerta é enviado; e, se o dispositivo permanecer fora da cerca por mais que alguns minutos, ele é desativado e sua localização passa a ser transmitida para o sistema da loja.\n\nEssa desativação pode ocorrer de várias formas: em dispositivos touchscreen, por exemplo, a operação por toque pode ser desabilitada; já em aparelhos com botões ou teclados, as teclas podem deixar de funcionar. O aparelho passa a exibir em sua tela um aviso afirmando que ele foi levado da loja em questão, adicionando o número de

	Apple tem intenções de transformar a tecnologia numa aplicação real em suas lojas ou qual o prazo para que o sistema passe a funcionar. Ainda assim, a ideia é boa — e certamente há de coibir a ação criminosos."				telefone da Apple Store para que a equipe da loja ofereça instruções de como devolvê-lo.\n\nCaso o aparelho roubado seja recuperado, basta que a loja entre com as credenciais corporativas nele para que seu funcionamento volte ao normal.\n\nObviamente, por se tratar de uma patente, não há como saber se a Apple tem intenções de transformar a tecnologia numa aplicação real em suas lojas ou qual o prazo para que o sistema passe a funcionar. Ainda assim, a ideia é boa — e certamente há de coibir a ação criminosos.', 'link': 'https://macmagazine.uol.com.br/2019/05/31/invencao-da-apple-bloqueia-iphones-ipads-e-macs-roubados-de-suas-lojas/', 'published': '2019-05-31T00:00:00'}

Tabela 33 - Exemplo de organização das novas notícias capturadas

Através da Tabela 33, para agrupamento das notícias, foi constatada uma diferença grande na classificação das notícias para cada uma das empresas, mas o resultado geral, média de sentimentos para cada um dos dias foi, na maioria dos casos, igual. Na tabela Tabela 34 são apresentados os resultados de classificação manual de notícias para cada uma das empresas para cada dia, efetuada pelos autores deste trabalho.

Empresa	**Data**	**Nº de Notícias negativas [-1]**	**Nº de Notícias neutras[0]**	**Nº de Notícias positivas [1]**	**Média de sentimento de notícias do dia**
Apple	2019-05-30	3	8	1	0
Apple	2019-05-31	1	8	4	0
Microsoft	2019-03-06	1	10	1	0
Microsoft	2019-03-08	2	6	1	0
Tesla	2019-03-01	0	3	3	1
Tesla	2019-03-22	5	1	1	-1

Tabela 34 - Resultado de classificação manual de notícias para testes de comparação do modelo. Fonte: Autoria própria

O sentimento dos textos das notícias acima foi feito também pelo modelo de predição, o resumo do resultado está apresentado na tabela Tabela 35, onde é possível notar a diferença na classificação de sentimentos, mas que não alterou muito o resultado final obtido (média de sentimentos de notícias do dia).

Empresa	**Data**	**Nº de Notícias negativas [-1]**	**Nº de Notícias neutras[0]**	**Nº de Notícias positivas [1]**	**Média de sentimento de notícias do dia**
Apple	2019-05-30	0	9	3	0
Apple	2019-05-31	0	7	6	0
Microsoft	2019-03-06	1	6	5	0
Microsoft	2019-03-08	1	5	3	0

Tesla	2019-03-01	0	6	0	0
Tesla	2019-03-22	0	7	0	0

Tabela 35 - Resultado de classificação do analisador de notícias para testes de comparação do modelo. Fonte: Autoria própria

Desconsiderando o resultado para a predição de sentimentos de notícias para a empresa Tesla, onde o resultado foi bem diferente do esperado e a média de sentimentos do dia foi bastante alterada, as outras duas empresas não sofreram grande impacto com os erros do analisador de sentimentos.

Com a constatação de que o modelo de predição de sentimentos de notícias de fato consegue classificar boa parte das notícias sem influenciar muito o resultado do dia,iniciou-se a segunda etapa do testes do modelo preditivo proposto, uma vez que o resultado de média de sentimentos das ações estava de acordo com o que se esperava. Para validar o funcionamento do restante do modelo e de toda a integração, foram coletados os dados relativos às ações de cada uma das empresas nas datas das notícias utilizadas na análise de sentimentos, descrita nas tabelas acima. Para cada data foram calculadas as diferenças e classificações de flutuação manualmente, de acordo com os parâmetro de classificação estipulados no trabalho (diferença entre o valor de abertura e fechamento das ações). A Tabela 36 apresenta os dados coletados e os valores adotados como esperados para saída do modelo. Alguns dados foram omitidos na tabela abaixo visando uma melhor compreensão do método de comparação utilizado, apenas os dados coletados que foram utilizados no cálculo de flutuação estão sendo apresentados, além dos valores criados através desse cálculo. A classificação de flutuação teve como valor limite entre as faixas 0,003 (0,03%) de diferença entre abertura e fechamento, resultando nas classificações apresentadas na tabela abaixo:

Empresa	**Data**	**Abertura**	**Fechamento**	**% flutuação**	**$ flutuação**	**Classificação de flutuação**
Apple	2019-05-30	708,00	711,00	0,0042	3,00	1
Apple	2019-05-31	690,10	690,90	0,0011	0,80	0
Microsoft	2019-03-06	426,90	427,34	0,0010	0,44	0
Microsoft	2019-03-08	427,60	425,24	-0,0055	-2,36	-1
Tesla	2019-03-01	287,37	287,37	0	0,00	0
Tesla	2019-03-22	281,46	281,46	0	0,00	0

Tabela 36 - Resultado de classificação manual de flutuação do valor de ação para testes de comparação do modelo. Fonte: Autoria própria

A partir dessa análise manual o intuito do algoritmo, bem como um dos objetivos deste trabalho, era de poder utilizar o modelo preditivo, já treinado, para prever a tendência do valor de fechamento de uma ação.Utilizando como entradas do modelo os valores de abertura, maior e menor valor, volume de transações e a média dos sentimentos das notícias, cada uma das empresas para cada um dos dias do teste, passaram pelo preditor de tendência de ação. Os resultados podem ser vistos na Tabela 37, e como pode ser visto, os resultados não estão muito diferentes do calculado de forma manual.

Empresa	**Data**	**Abertura**	**Predição de tendência de flutuação**
Apple	2019-05-30	708,00	1
Apple	2019-05-31	690,10	0
Microsoft	2019-03-06	426,90	0
Microsoft	2019-03-08	427,60	0
Tesla	2019-03-01	287,37	0
Tesla	2019-03-22	281,46	0

Tabela 37 - Resultado de predição da classificação de flutuação do valor de ação para testes de comparação do modelo. Fonte: Autoria própria

O algoritmo não conseguiu prever que no dia 08 de março de 2019 a tendência do preço das ações da microsoft seria negativa, mas as tendências preditas para os demais dias, de todas as empresas, foram corretas. Neste teste, como já era sabido que a diferença entre abertura e fechamento das ações era menor que o valor de 3% utilizado nos treinamentos, o modelo foi treinado novamente para identificar variações no intervalo de 0,03%. Dessa forma os valores de tendência tiveram uma variação um pouco maior, como pode ser visto na Tabela 36. Por exemplo no 30 de maio de 2019 a tendência das ações da Apple foi positiva, considerando uma variação no intervalo de -0,03% e 0,03%.

Os testes manuais foram executados apenas para dois dias de notícias de cada empresa. Uma vez que a classificação manual, a busca e a separação das notícias é um processo bastante demorado e no caso deste tipo de teste totalmente dependente de uma pessoa para execução de pelo menos metade das ações necessárias. Por conta disso, este teste foi realizado apenas para colaborar com os resultados dos testes dos algoritmos citados durante os parágrafos da sessão 5. Uma vez que se acredita que existe alguma correlação entre valores de ações e as notícias divulgadas sobre as empresas em um dia.

6 Conclusões e Trabalhos Futuros

Podemos afirmar que o objetivo geral deste trabalho foi atingido, sendo desenvolvido com sucesso um modelo preditivo, capaz de relacionar dados históricos de valores de ações de empresas multinacionais a notícias a elas relacionadas, publicadas em portais de notícias online.

O pré processamento de notícias para treinamento do modelo analisador de sentimentos utilizou diversas técnicas e diferentes algoritmos. Contudo, os resultados não demonstraram grandes diferenças em acurácia e se mantiveram sempre abaixo de 60%, o que é considerável um nível baixo de performance pelos autores deste trabalho. Uma hipótese levantada como motivo da baixa acurácia é o número de notícias disponíveis, de forma que o modelo não possui exemplos suficientes para ajustar seus pesos e performar com eficiência. Além disso, fica a possibilidade de se considerar os títulos das notícias juntamente ao conteúdo das mesmas.

O modelo final prevê flutuações de preços com elevada acurácia; entretanto, tornamos a mencionar aqui o módulo de análise de sentimentos. Embora funcional, não obteve os resultados de acurácia desejados. Isto nos faz questionar a relevância dos sentimentos como entrada do modelo preditivo, visto que até mesmo testes que não incluíram sentimentos como atributos trouxeram resultados muito semelhantes aos testes de cenário contrário.

Não obstante, foram realizados testes manuais em um período específico com o intuito de validar, em menor escala, o funcionamento do algoritmo como um todo. Os testes manuais tiveram como objetivo principal mostrar a existência de alguma correlação entre os valores das ações das empresas e as notícias relacionadas a estas que são lançadas pela mídia durante um dia.

Uma vez que os testes realizados alcançaram resultados muito semelhantes aos que seriam calculados por uma pessoa, acredita-se que o algoritmo conseguiu de alguma forma executar o que foi proposto, correlacionar notícias de portais online com a tendência do valor de fechamento das ações de uma empresa.

Apesar de rudimentares, os testes se mostraram bem sucedidos em alguns aspectos importantes. Foram validados o funcionamento do analisador de sentimentos, para as notícias do teste. Utilizando os resultados de predição de sentimentos de forma individual o algoritmo criado apresenta algumas falhas, mas utilizando o contexto deste trabalho (a média de sentimentos de notícias de um dia) o resultado foi satisfatório e totalmente dentro do esperado. Da mesma forma o preditor de valores, utilizando como base de comparação as tendências calculadas manualmente pelos autores deste trabalho, o preditor errou a tendência de 1 dos 6 dias de testes.

Quanto aos objetivos específicos desta pesquisa, afirmamos que o objetivo 1, relativo à pesquisa teórica sobre o estado da arte em extração e transformação de dados, foi realizada e diferentes técnicas até mesmo foram utilizadas. O mesmo pode ser dito sobre o objetivo 2, conforme conhecimentos adquiridos a partir dos os trabalhos relacionados destacados na seção 3.

Os objetivos de número 3 e 4 também foram concluídos, inclusive ambos se mesclando com o objetivo de número 1, visto que diferentes técnicas de pré processamento de dados e variados algoritmos foram aplicados no processamento de linguagem natural.

Estando objetivo específico de número 5 fortemente atrelado ao objetivo geral desta obra, que foi atingido, afirmamos que ambos foram concluídos com sucesso. O modelo preditivo demonstrou elevada acurácia para os casos de teste a que foi submetido.

Os objetivo 6 e 7 que tinham como objetivo colocar o modelo preditivo a prova, utilizando notícias fora do conjunto de dados utilizado no treinamento do mesmo. Não foi possível fazer os testes de comparação do modelo com notícias em tempo real, como era a intenção ao início do trabalho, mas foi possível criar um protótipo de modelo preditivo capaz de associar notícias com a flutuação do valor de fechamento de ação de uma empresa. Ambos os objetivos foram alcançados com certo sucesso, como descrito na seção 5.4.

Foram feitas comparações com modelos preditivos aplicados pelos trabalhos correlatos a esta pesquisa, mais especificamente nas seções 5.1.3 e 5.2.1. Sendo assim, foi atingido também o objetivo de número 8.

A partir desta pesquisa, percebemos que é pode ser beneficial uma análise mais aprofundada nos resultados dos testes, explorando matrizes de confusão e possíveis hipóteses que delas possam surgir. Para o classificador de ações em geral, também fica a proposta de utilizar diferentes parâmetros para a predição de ações, que possam ser obtidas através de diferentes fontes, além de diferentes configurações para determinar se um dia da bolsa é classificado como positivo, neutro ou negativo. É possível inclusive se utilizar métricas geradas a partir dos dados originais, como médias móveis, por exemplo.

Utilizar um maior conjunto de testes também pode se mostrar benéfico para a acurácia geral dos modelos. Isto pode ser obtido com a captura de notícias por um tempo maior que o realizado nesta pesquisa, ou, até mesmo, utilizando dados oriundos de terceiros. Desta forma se amplia não somente o número de notícias com as quais parametrizar o analisador de sentimentos, mas também o número de dias de mercado considerados pelo analisador de sentimentos, visto que ele é treinado com base somente em dias para os quais há notícias no conjunto.

A criação de uma interface para o usuário, talvez criando uma API com modelo de arquitetura REST, parametrizável, possa facilitar os testes e validações do modelo. O formato atual do preditor, apesar de funcional, ainda é bastante rudimentar no que diz respeito a interação de algum usuário não habituado com o padrão de funcionamento do modelo. Esta falta de uma interface simples dificulta bastante a realização dos testes e validações do mesmo, sendo no momento necessária a alteração direta no código fonte para testar qualquer modificação

Outro fator interessante é de verificar o impacto das notícias sobre a liquidez das ações, não somente o preço, visto que é possível que haja grande volume de transações, mesmo sem alterar o preço. Também acreditamos que é importante fazer com que seja possível configurar o uso da aplicação para diferentes empresas a partir de uma configuração, sem que haja mudanças manuais no código.

Referências

ASSAF NETO, Alexandre. Finanças corporativas e valor. 3° ed. São Paulo: Atlas, 2001.

LUGER, George F.. Inteligência Artificial. 6. ed. São Paulo: Pearson Education, 2013. 632 p.

PAGOLU, Venkata Sasank et al. Sentiment analysis of Twitter data for predicting stock market movements. 2016 International Conference On Signal Processing, Communication, Power And Embedded System (scopes), [s.l.], v. 0, n. 0, p.1345-1350, out. 2016. IEEE. http://dx.doi.org/10.1109/scopes.2016.7955659.

LUSTOSA, Volney Gadelha. O Estado da Arte em Inteligência Artificial. Colabor@: Revista Digital da CVA, Brasília, v. 2, n. 8, p.1-11, set. 2004. Semestral. Disponível em: <http://pead.ucpel.tche.br/revistas/index.php/colabora/article/viewFile/60/53>. Acesso em: 25 maio 2018.

CHOWDHURY, Gobinda G.. Natural language processing. Annual Review Of Information Science And Technology, [s.l.], v. 37, n. 1, p.51-89, 31 jan. 2005. Wiley. http://dx.doi.org/10.1002/aris.1440370103.

FAMA, Eugene F.. Efficient Capital Markets: II. The Journal Of Finance, [s.l.], v. 46, n. 5, p.1575-1617, dez. 1991. JSTOR. http://dx.doi.org/10.2307/2328565.

ANTUNES, M. A.; PROCIANOY, J. L. Os efeitos das decisões de investimentos das empresas sobre os preços de suas ações no mercado de capitais. Revista de Administração, v. 38, n. 1, p. 5-14, 2003.

SILVA, César Augusto Tibúrcio; PEREIRA, Vinícius Alves dos Santos. FATOS RELEVANTES E SUA INFLUÊNCIA NO PREÇO DAS AÇÕE S NO BRASIL. In: CONGRESSO USP DE INICIAÇÃO CIENTÍFICA EM CONTABILIDADE, 5., 2008, São Paulo. Anais... . São Paulo: Fea, 2008. p. 1 - 15. Disponível em: <http://www.congressousp.fipecafi.org/anais/artigos82008/575.pdf>. Acesso em: 26 maio 2018.

ALVARES, Reinaldo V. Investigação do Processo de Stemming na Língua Portuguesa. Niterói, São Paulo. 2005.

TRIM, Craig. The Art of Tokenization. 2013. Disponível em: <https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization>. Acesso em 10 de outubro de 2018.

FINLAY, Steven. Predictive analytics, data mining, and big data: myths, misconceptions and methods. New York: Palgrave Macmillan, c2014. xii, 248 p. ISBN 9781137379276.

MAYER-SCHÖNBERGER, Viktor; CUKIER, Kenneth. Big data: a revolution that will transform how we live, work, and think. Boston: Houghton Mifflin Company, c2013. 252 p. ISBN 9780544227750.

WU, Xindong; ZHU, Xingquan; WU, Gong-qing. Data mining with big data. Ieee Transactions On Knowledge And Data Engineering, [s.l.], v. 26, n. 1, p.97-107, jan. 2014. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tkde.2013.109. Disponível em: <https://ieeexplore.ieee.org/document/6547630>. Acesso em: 23 nov. 2018.

ARUMUGAM, Paliah et al. Financial Stock Market Forecast using Data Mining Techniques. International Multiconference Of Engineers And Computer Scientists, Hong Kong, v. 1, n. 1, p.1-6, 17 mar. 2010. Anual. Disponível em: <https://www.researchgate.net/publication/44260645_Financial_Stock_Market_Forecast_using_Data_Mining_Techniques>. Acesso em: 23 nov. 2018.

CHEN, Ming-syan; HAN, Jiawei; YU, P.s.. Data mining: an overview from a database perspective. Ieee Transactions On Knowledge And Data Engineering, [s.l.], v. 8, n. 6, p.866-883, 1996. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/69.553155. Disponível em: <https://ieeexplore.ieee.org/abstract/document/553155/authors#authors>. Acesso em: 23 nov. 2018.

BARIONI, Maria Camila Nardini; Miyashiro, Davis Une. Estudo de Técnicas de Redução de Dimensionalidade. Iniciação Científica, Universidade Federal do ABC, 2009. Disponível em: <http://ic.ufabc.edu.br/II_SIC_UFABC/resumos/paper_5_151.pdf>. Acesso em: 3 jun. 2019.

GEVARTER, William B. Artificial intelligence, expert systems, computer vision, and natural language processing. Park Ridge, N.J.: Noyes, c1984. 226p. ISBN 0815509944 (enc.).

SHWARTZ, Steven P. Applied natural language processing. Princeton: Petrocelli Books, c1987. xvii, 293p. ISBN 0894332600 : (broch.)

ABDELAZIZ, et. al. A Machine Learning Model for Improving Healthcare services on Cloud Computing Environment. https://doi.org/10.1016/j.measurement.2018.01.022. Disponível em:

<https://www.researchgate.net/profile/Mohamed_Elhoseny4/publication/322892893_A_Machine_Learning_Model_for_Improving_Healthcare_services_on_Cloud_Computing_Environment/links/5a762c89a6fdccbb3c07abf4/A-Machine-Learning-Model-for-Improving-Healthcare-services-on-Cloud-Computing-Environment.pdf>. Acesso em: 5 jun. 2019.

FUKUNAGA, Keinosuke. Introduction to statistical pattern recognition. 2nd. ed. San Diego: Morgan Kaufmann, 1990. 591p. (Computer science and scientific computing) ISBN 0122698517

LIPPMANN, R.. An introduction to computing with neural nets. Ieee Assp Magazine, [s.l.], v. 4, n. 2, p.4-22, 1987. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/massp.1987.1165576. Disponível em: <https://ieeexplore.ieee.org/abstract/document/1165576>. Acesso em: 24 nov. 2018.

BUSSAB, Wilton de Oliveira; MORETTIN, Pedro Alberto. Estatística básica. 5. ed. São Paulo: Saraiva, 2002. 526p. ISBN 9788502034976.

DEVORE, Jay L. Probabilidade e estatística: para engenharia e ciências. São Paulo: Thomson, 2006. xiii, 692 p. ISBN 9788522104598.

BARBETTA, Pedro Alberto. Estatística aplicada às ciências sociais. 8. ed. rev. Florianópolis: Ed. da UFSC, 2012. 315 p. ISBN 9788532806048.

APÊNDICE A - Aplicação de predição de preços de ações através de portais de notícia

```
# training.news_handler.py
from copy import copy


class NewsHandler:

    def __init__(self, company_data):
        super().__init__()
        self.news_history = company_data

    def get_sentiments_per_day(self):
        sentiments_of_all_days = {}
        for index, row in self.news_history.iterrows():
            publishment_date = str(row['published_at'])[:10]
            sentiment = int(row['sentiment'])

            if publishment_date not in sentiments_of_all_days:
                day_sentiments = [sentiment]
                sentiments_of_all_days[publishment_date] = day_sentiments
            else:
                day_sentiments = sentiments_of_all_days[publishment_date]
                day_sentiments.append(sentiment)
                sentiments_of_all_days[publishment_date] = day_sentiments
        return sentiments_of_all_days

    def analyse_day_sentiment(self, day_sentiments):
        grouped_day_sentiments = copy(day_sentiments)
        for key, value in grouped_day_sentiments.items():
            day_sentiments = grouped_day_sentiments[key]
            positive_count = 0
            negative_count = 0
            neutral_count = 0
            for sentiment in day_sentiments:
                if sentiment == 0:
                    neutral_count += 1
                elif sentiment == 1:
                    positive_count += 1
                else:
                    negative_count += 1
            if negative_count > positive_count and negative_count > neutral_count:
```

```
            grouped_day_sentiments[key] = -1
        elif positive_count > negative_count and positive_count > neutral_count:
            grouped_day_sentiments[key] = 1
        elif neutral_count > negative_count and neutral_count > positive_count:
            grouped_day_sentiments[key] = 0
        else:
            grouped_day_sentiments[key] = 0
    return grouped_day_sentiments

def get_polarized_days(self):
    day_sentiments = self.get_sentiments_per_day()
    predicted_sentiments = self.analyse_day_sentiment(day_sentiments)

    return predicted_sentiments
```

```python
# training.sentiment_analyzer_generator.py
from sklearn.feature_extraction.text import CountVectorizer

from util import TextCleaner


def run(model, company_data):
    vectorizer = CountVectorizer(ngram_range=(1, 3))
    news_content = []
    news_classes = []

    for index, row in company_data.iterrows():
        content = row['content']
        content = TextCleaner.remove_special_symbols(content)
        content = TextCleaner.lemmatize_text(content)

        news_content.append(content)
        news_classes.append(str(row['sentiment']))

    company_news = news_content
    x = vectorizer.fit_transform(company_news)
    y = news_classes

    model.fit(x, y)

    return model, vectorize



# training.stock_predictor_generator.py
import json
import warnings
from copy import deepcopy

import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

from training.news_handler import NewsHandler
from util import Stock

warnings.filterwarnings('ignore', category=Warning)


def pre_process(company_data, company_code):
    stock_historical_values = Stock.get_history(company_code)
```

```
    filtered_dates, dates_sentiment = set_sentiments_per_day(company_data,
stock_historical_values)

    history = json.dumps(filtered_dates)

    df = pd.read_json(history, encoding='utf8')

    df = df.T
    df.columns = ['open', 'high', 'low', 'close', 'volume']

    df = df.assign(sentiment=pd.Series(dates_sentiment).values)

    num_of_rows = len(df.index)
    labels = np.empty(shape=(num_of_rows, 1))

    threshold = 0.03
    for i in range(0, num_of_rows):
        percentage = 1 - (df['close'][i] / df['open'][i])
        if percentage >= threshold:
            labels[i] = 1
        elif percentage <= -threshold:
            labels[i] = -1
        else:
            labels[i] = 0

    return df[['low', 'high', 'open', 'volume', 'sentiment']].values, labels.ravel()


def set_sentiments_per_day(company_data, stock_days):
    stock_days_copy = deepcopy(stock_days)
    handler = NewsHandler(company_data)
    polarized_days = handler.get_polarized_days()
    days_to_remove = []
    sentiments = []
    for day in stock_days_copy:
        try:
            sentiments.append(polarized_days[day])
        except:
            days_to_remove.append(day)
    for day in days_to_remove:
        del stock_days_copy[day]
    return stock_days_copy, sentiments


def run(model, company_code, company_data):
    x, y = pre_process(company_data, company_code)
```

```
scaler = MinMaxScaler()
x = scaler.fit_transform(x)
model.fit(x, y)

return model, scaler
```

```
# util.Normaliser.py
from datetime import datetime

from sklearn.externals import joblib
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
dateformat = '%Y-%m-%d %H:%M:%S'


def fit(o):
    scaler.fit(o)
    return scaler.fit(o)


def transform(o):
    return scaler.transform(o)


def fit_transform(o):
    fit(o)
    return transform(o)


def persist(company):
    path = f'persistence/stock/{company}/normaliser.joblob'
    try:
        print(f'Persisting normaliser into {path}')
        joblib.dump(scaler, path)
    except Exception as e:
        print(f'Unable to persist into {path}')
        print(str(e))


def load(company):
    print(f'{company} sentiment analysis model load start: ',
datetime.now().strftime(dateformat))
    global scaler
    path = f'persistence/stock/{company}/normaliser.joblob'
    scaler = joblib.load(path)
    print(f'{company} sentiment analysis model load finish: ',
datetime.now().strftime(dateformat))
```

```
# util.Stock.py
import json
from time import sleep
from urllib import request

API_KEY = '_CHANGE_HERE_'
APPLE = 'APPLE'
MICROSOFT = 'MICROSOFT'
TESLA = 'TESLA'
APPLE_CODE = 'AAPL34.SA'
MICROSOFT_CODE = 'MSFT34.SA'
TESLA_CODE = 'TSLA34.SA'

history = {APPLE_CODE: None, MICROSOFT_CODE: None, TESLA_CODE: None}


def get_history(company_code):
    if history[company_code] is not None:
        return history[company_code]
    else:
        url_template =
'https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol={}&outp
utsize=full&apikey={}'
        url = url_template.format(company_code, API_KEY)

        content = request.urlopen(url).read()
        content = json.loads(content)

        key = 'Time Series (Daily)'
        if key not in content:
            while key not in content:
                print('Could not fetch historical time series. Waiting 5 seconds.')
                sleep(5)
                print('Waiting is over.')
                content = request.urlopen(url).read()
                content = json.loads(content)

        data = content['Time Series (Daily)']
        history[company_code] = data

        return data
```

```
# util.TextCleaner.py
import re

import nltk
from nltk.stem import SnowballStemmer

language = 'portuguese'
stemmer = SnowballStemmer(language)
portuguese_stopwords = nltk.corpus.stopwords.words(language)

htmlTagsAndSymbolsRegex = re.compile(r'(<[^>]+>)|(&#[0-9]+;)')
lineBreaksAndTabsRegex = re.compile(r'(\\n|\n|\\t|\t)+')


def remove_special_symbols(text):
    text = htmlTagsAndSymbolsRegex.sub('', text)
    text = lineBreaksAndTabsRegex.sub(' ', text)
    return text.strip().lower()


def lemmatize_text(text):
    tokens = nltk.tokenize.word_tokenize(text, language='portuguese')
    tokens = [token for token in tokens if len(token) > 2]
    tokens = [token for token in tokens if token not in portuguese_stopwords]
    tokens = [stemmer.stem(token) for token in tokens]
    return ' '.join(tokens)
```

```
# results.py
sentiment_analyzer_only_results = {
    'APPLE': {
        'Sequential': [],
        'LogisticRegression': [],
        'DecisionTreeClassifier': [],
        'RandomForestClassifier': [],
        'KNeighborsClassifier': [],
        'AdaBoostClassifier': [],
        'GradientBoostingClassifier': []
    },
    'MICROSOFT': {
        'Sequential': [],
        'LogisticRegression': [],
        'DecisionTreeClassifier': [],
        'RandomForestClassifier': [],
        'KNeighborsClassifier': [],
        'AdaBoostClassifier': [],
        'GradientBoostingClassifier': []
    },
    'TESLA': {
        'Sequential': [],
        'LogisticRegression': [],
        'DecisionTreeClassifier': [],
        'RandomForestClassifier': [],
        'KNeighborsClassifier': [],
        'AdaBoostClassifier': [],
        'GradientBoostingClassifier': []
    }
}
stock_predictor_results = {
    'APPLE': {
        'LogisticRegression': {
            'LogisticRegression': [],
            'DecisionTreeClassifier': [],
            'RandomForestClassifier': [],
            'KNeighborsClassifier': [],
            'AdaBoostClassifier': [],
            'GradientBoostingClassifier': []
        },
        'DecisionTreeClassifier': {
            'LogisticRegression': [],
            'DecisionTreeClassifier': [],
            'RandomForestClassifier': [],
            'KNeighborsClassifier': [],
            'AdaBoostClassifier': [],
```

```
            'GradientBoostingClassifier': []
        },
        'RandomForestClassifier': {
            'LogisticRegression': [],
            'DecisionTreeClassifier': [],
            'RandomForestClassifier': [],
            'KNeighborsClassifier': [],
            'AdaBoostClassifier': [],
            'GradientBoostingClassifier': []
        },
        'KNeighborsClassifier': {
            'LogisticRegression': [],
            'DecisionTreeClassifier': [],
            'RandomForestClassifier': [],
            'KNeighborsClassifier': [],
            'AdaBoostClassifier': [],
            'GradientBoostingClassifier': []
        },
        'AdaBoostClassifier': {
            'LogisticRegression': [],
            'DecisionTreeClassifier': [],
            'RandomForestClassifier': [],
            'KNeighborsClassifier': [],
            'AdaBoostClassifier': [],
            'GradientBoostingClassifier': []
        },
        'GradientBoostingClassifier': {
            'LogisticRegression': [],
            'DecisionTreeClassifier': [],
            'RandomForestClassifier': [],
            'KNeighborsClassifier': [],
            'AdaBoostClassifier': [],
            'GradientBoostingClassifier': []
        }
    },
    'MICROSOFT': {
        'LogisticRegression': {
            'LogisticRegression': [],
            'DecisionTreeClassifier': [],
            'RandomForestClassifier': [],
            'KNeighborsClassifier': [],
            'AdaBoostClassifier': [],
            'GradientBoostingClassifier': []
        },
        'DecisionTreeClassifier': {
            'LogisticRegression': [],
```

```
        'DecisionTreeClassifier': [],
        'RandomForestClassifier': [],
        'KNeighborsClassifier': [],
        'AdaBoostClassifier': [],
        'GradientBoostingClassifier': []
    },
    'RandomForestClassifier': {
        'LogisticRegression': [],
        'DecisionTreeClassifier': [],
        'RandomForestClassifier': [],
        'KNeighborsClassifier': [],
        'AdaBoostClassifier': [],
        'GradientBoostingClassifier': []
    },
    'KNeighborsClassifier': {
        'LogisticRegression': [],
        'DecisionTreeClassifier': [],
        'RandomForestClassifier': [],
        'KNeighborsClassifier': [],
        'AdaBoostClassifier': [],
        'GradientBoostingClassifier': []
    },
    'AdaBoostClassifier': {
        'LogisticRegression': [],
        'DecisionTreeClassifier': [],
        'RandomForestClassifier': [],
        'KNeighborsClassifier': [],
        'AdaBoostClassifier': [],
        'GradientBoostingClassifier': []
    },
    'GradientBoostingClassifier': {
        'LogisticRegression': [],
        'DecisionTreeClassifier': [],
        'RandomForestClassifier': [],
        'KNeighborsClassifier': [],
        'AdaBoostClassifier': [],
        'GradientBoostingClassifier': []
    }
},
'TESLA': {
    'LogisticRegression': {
        'LogisticRegression': [],
        'DecisionTreeClassifier': [],
        'RandomForestClassifier': [],
        'KNeighborsClassifier': [],
        'AdaBoostClassifier': [],
```

```
            'GradientBoostingClassifier': []
        },
        'DecisionTreeClassifier': {
            'LogisticRegression': [],
            'DecisionTreeClassifier': [],
            'RandomForestClassifier': [],
            'KNeighborsClassifier': [],
            'AdaBoostClassifier': [],
            'GradientBoostingClassifier': []
        },
        'RandomForestClassifier': {
            'LogisticRegression': [],
            'DecisionTreeClassifier': [],
            'RandomForestClassifier': [],
            'KNeighborsClassifier': [],
            'AdaBoostClassifier': [],
            'GradientBoostingClassifier': []
        },
        'KNeighborsClassifier': {
            'LogisticRegression': [],
            'DecisionTreeClassifier': [],
            'RandomForestClassifier': [],
            'KNeighborsClassifier': [],
            'AdaBoostClassifier': [],
            'GradientBoostingClassifier': []
        },
        'AdaBoostClassifier': {
            'LogisticRegression': [],
            'DecisionTreeClassifier': [],
            'RandomForestClassifier': [],
            'KNeighborsClassifier': [],
            'AdaBoostClassifier': [],
            'GradientBoostingClassifier': []
        },
        'GradientBoostingClassifier': {
            'LogisticRegression': [],
            'DecisionTreeClassifier': [],
            'RandomForestClassifier': [],
            'KNeighborsClassifier': [],
            'AdaBoostClassifier': [],
            'GradientBoostingClassifier': []
        }
    }
}
```

```
# train.py
import json
from copy import deepcopy
from time import time

import numpy as np
import pandas as pd
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.utils import shuffle

import results
from training import sentiment_analyzer_generator
from training import stock_predictor_generator
from training.news_handler import NewsHandler
from util import Stock
from util import TextCleaner

print("BEGINNING THE SCRIPT")

sentiment_analyzer_models = [
    LogisticRegression(multi_class='auto', solver='lbfgs'),
    DecisionTreeClassifier(),
    RandomForestClassifier(),
    KNeighborsClassifier(),
    AdaBoostClassifier(),
    GradientBoostingClassifier()
]

stock_predictor_models = [
    LogisticRegression(multi_class='auto', solver='lbfgs'),
    DecisionTreeClassifier(),
    RandomForestClassifier(),
    KNeighborsClassifier(),
    AdaBoostClassifier(),
    GradientBoostingClassifier()
]

print("LOADING DATA FILES")
apple = pd.read_json(f'data/{Stock.APPLE}/news.json', encoding='utf8')
microsoft = pd.read_json(f'data/{Stock.MICROSOFT}/news.json', encoding='utf8')
tesla = pd.read_json(f'data/{Stock.TESLA}/news.json', encoding='utf8')
```

```
init_time = time()

companies = [Stock.APPLE, Stock.MICROSOFT, Stock.TESLA]
companies_data = {Stock.APPLE: apple, Stock.MICROSOFT: microsoft, Stock.TESLA:
tesla}
company_codes = {Stock.APPLE: 'AAPL34.SA', Stock.MICROSOFT: 'MSFT34.SA',
Stock.TESLA: 'TSLA34.SA'}

print("BEGINNING THE TRAINING")
for i in range(0, 1):
    for company in companies:

        for sentiment_analyzer_model in sentiment_analyzer_models:
            sentiment_analyzer_model_name = type(sentiment_analyzer_model).__name__

            company_data = shuffle(deepcopy(companies_data[company]))

            num_of_rows = company_data.shape[0]
            split = int(num_of_rows * 0.7)

            training_data = company_data.iloc[:split, :]
            testing_data = company_data.iloc[split - num_of_rows:, :]

            trained_sentiment_analyzer, sentiment_vectorizer =
sentiment_analyzer_generator.run(
                sentiment_analyzer_model,
                training_data)

            testing_news = []
            news_classes = []
            for index, row in testing_data.iterrows():
                content = row['content']
                content = TextCleaner.remove_special_symbols(content)
                content = TextCleaner.lemmatize_text(content)
                testing_news.append(content)
                news_classes.append(str(row['sentiment']))

            vectorized_testing_news = sentiment_vectorizer.transform(testing_news)
            score = trained_sentiment_analyzer.score(vectorized_testing_news, news_classes)

results.sentiment_analyzer_only_results[company][sentiment_analyzer_model_name].appen
d(score)

            results_with_current_sentiment_analyzer = {}
```

```
        for stock_predictor_model in stock_predictor_models:
            stock_predictor_model_name = type(stock_predictor_model).__name__
            company_code = company_codes[company]
            try:
                trained_stock_predictor, scaler =
stock_predictor_generator.run(stock_predictor_model, company_code,
                                                              training_data)
            except:
                continue

            testing_news = []
            for index, row in testing_data.iterrows():
                content = row['content']
                content = TextCleaner.remove_special_symbols(content)
                content = TextCleaner.lemmatize_text(content)
                testing_news.append(content)

            vectorized_testing_news = sentiment_vectorizer.transform(testing_news)
            news_sentiments = trained_sentiment_analyzer.predict(vectorized_testing_news)

            testing_data["sentiment"] = news_sentiments

            stock_historical_values = Stock.get_history(company_code)
            stock_history_for_polarized_days = deepcopy(stock_historical_values)
            handler = NewsHandler(testing_data)
            polarized_days = handler.get_polarized_days()
            days_to_remove = []
            sentiments = []
            for day in stock_history_for_polarized_days:
                try:
                    sentiments.append(polarized_days[day])
                except:
                    days_to_remove.append(day)
            for day in days_to_remove:
                del stock_history_for_polarized_days[day]

            history = json.dumps(stock_history_for_polarized_days)
            df = pd.read_json(history, encoding='utf8')
            df = df.T
            df.columns = ['open', 'high', 'low', 'close', 'volume']
            df = df.assign(sentiment=pd.Series(sentiments).values)

            num_of_rows = len(df.index)
            labels = np.empty(shape=(num_of_rows, 1))

            threshold = 0.03
```

```
            for j in range(0, num_of_rows):
                percentage = 1 - (df['close'][j] / df['open'][j])
                if percentage >= threshold:
                    labels[j] = 1
                elif percentage <= -threshold:
                    labels[j] = -1
                else:
                    labels[j] = 0

            x = df[['low', 'high', 'open', 'volume', 'sentiment']].values
            x = scaler.transform(x)
            y = labels.ravel()

            score = stock_predictor_model.score(x, y)
            results.stock_predictor_results[company][sentiment_analyzer_model_name][
                stock_predictor_model_name].append(score)

            print(
                f"Score of {score} for the {company} company with the following
models:")
            print(f"  - For sentiment analysis: {sentiment_analyzer_model_name}")
            print(f"  - For stock prediction: {stock_predictor_model_name}")
            print()

finish_time = time()
print("FINISHING THE SCRIPT")

print()
print(f'Init time: {init_time}')
print(f'Finish time: {finish_time}')
print(f'Total execution time: {finish_time - init_time}')
print(f'Sentiment analyzer results: {results.sentiment_analyzer_only_results}')
print(f"Stock predictor results: {results.stock_predictor_results}")

file = open('results.txt', 'w')
file.write(f'Init time: {init_time}\n')
file.write(f'Finish time: {finish_time}\n')
file.write(f'Total execution time: {finish_time - init_time}\n')
file.write(f'Sentiment analyzer only: {results.sentiment_analyzer_only_results}\n')
file.write(f'Stock predictor: {results.stock_predictor_results}\n')

file.close()
```

APÊNDICE B - Artigo

Predição de preço de ações através de portais de notícias

Clayton Raposo Veras[1], Lucas Mauro de Souza[1]

[1]Departamento de informática e Estatística - Universidade Federal de Santa Catarina (UFSC) Caixa Postal 476 - 88040-900 - Florianópolis - SC - Brazil

{veras.clayton, lucasmaurodesouza}@gmail.com

Abstract. Using natural language processing techniques to transform the content of news into a machine understandable format, so that it could be classified into positive, negative or neutral, combined with predictive algorithms, this paper has as main goal to create a predictive model that could correlate information relative to big technology companies, extracted from the news, and their stock price fluctuation. With this information, it is sought to do a prediction of the fluctuation of the stock price of the companies into a day.

Resumo. Utilizando técnicas e algoritmos de processamento de linguagem natural para tentar transformar o conteúdo de notícia em um formato compreensível pela máquina, de forma que possa ser classificado em positivas, negativas ou neutras, aliados a algoritmos de predição que tentam identificar padrões e correlações entre diferentes dados, este trabalho tem como principal criar um modelo preditivo que correlacione informações relativas à empresas de tecnologia retiradas de notícias com a flutuação de seu valor de ação. Com estas informações se procura predizer se a tendência do valor de ação é de subida, queda ou estagnação.

1. Introdução

A computação de dados vem se tornando mais popular a cada dia, sendo de grande importância no processo de tornar a população mundial mais atenta a relações que até então eram praticamente impossíveis de serem analisadas devido à sua complexidade. O ramo estatístico da matemática já estuda estas relações complexas há bastante tempo, mas os avanços tecnológicos da informática vêm tornando os estudos mais assertivos e processáveis, de forma automática.

A Inteligência Artificial (IA) é um exemplo destes avanços. Por ser uma disciplina jovem, com sua estrutura, considerações e métodos não definidos tão claramente quanto aqueles de uma ciência mais madura, como a física, existe uma dificuldade em se chegar a

uma definição exata para o termo IA, segundo Luger (2013). Apesar de ser considerada complexa, Luger (2013) apresenta uma das possíveis definições para essa área de estudo como um ramo da ciência da computação que se dedica a automação de comportamento inteligente.

Este trabalho não visa definir o que é a inteligência, nem o que é inteligência no meio computacional. Por isso, é apresentado um dos possíveis conceitos do que seria IA. O estudo sobre a definição do que a inteligência representa no mundo computacional pode ser aprofundado em estudos mais amplos. Como apresenta Lustosa (2004), a inteligência artificial tenta entender o comportamento de entidades inteligentes, porém, ao contrário da filosofia e da psicologia, que estão mais preocupadas como o estudo da inteligência dentro de um contexto de relações humanas, a IA foca em como essas entidades podem ser criadas e utilizadas para determinados fins.

Todas linguagens têm suas características específicas, como símbolos e regras gramaticais, que demandam diferentes medidas para se realizar operações com elas. Entretanto, para serem processadas por um computador, precisam ser tratadas com formalismo e determinismo. Em geral, a pesquisa em PLN tem avançado significativamente para o idioma inglês, mas para o português não há disponível a mesma quantia ou profundidade de material. Sendo assim, ao se analisar textos de cunho econômico, além de tratar o escopo de linguagem econômica, também é necessário considerar o idioma em que foi escrito.

O mercado de ações e seus comportamentos são alvos frequentes de estudos em áreas como IA e estatística, visto que é um meio que não depende apenas de dados para explicar suas quedas e altas, a presença de fatores humanos para os quais não se dispõe de modelos matemáticos é um dos maiores problemas a serem tratados nesta área, até mais do que a quantidade massiva de dados a serem processados. Um exemplo disso é que, em geral, apesar dos esforços empreendidos pelas empresas visando aumentar seu valor, este pode sofrer impactos positivos ou negativos devido a decisões aparentemente banais, como escolher uma opção de investimento. A opção de investimento, por sua vez, está conectada a diversos outros fatores que influencia de forma positiva ou negativa sua rentabilidade.

Embora existam diversas formas de avaliar a flutuação no mercado financeiro, neste trabalho propomos a relação entre notícias sobre um número determinado de empresas em portais de notícia com maior credibilidade no Brasil, jornais que em geral a população confia, com o intuito de, através de técnicas de inteligência artificial, identificar acontecimentos e demonstrar que estes têm relação direta com a flutuação nos valores das ações destas empresas. Desta forma, com um modelo preditivo, será possível ter um modelo que receba notícias automaticamente e determine a probabilidade de acréscimo ou decréscimo no valor da ação da empresa noticiada.

2. Escopo

O escopo do projeto se define no estudo de técnicas de processamento de linguagem natural, aplicadas ao setor da economia. Serão estudados históricos das empresas Apple, Microsoft e

Tesla, indexadas no índice BOVESPA, visto que notícias sobre empresas de grande porte tendem a alcançar um número elevado de leitores, além de lançar produtos e serviços com maior frequência.

O foco deste trabalho é de determinar a avaliação de relação entre notícias sobre empresas e o preço de suas ações na bolsa de valores. Para tanto, bibliotecas externas serão utilizadas para auxiliar neste processo, em especial na execução de tarefas que, embora dêem subsídio ao trabalho, não pertencem ao conjunto de objetivos deste, como obtenção de histórico de notícias e valores destas empresas. É considerado que os objetivos serão concluídos até o fim do cronograma estabelecido. Além disso, tem-se como verdade que a maior parte dos conhecimentos necessários para este trabalho será adquirida durante a execução dele.

3. Fundamentação teórica

3.1. Processamento de linguagem natural

Há pelo menos três questões principais envolvidas na compreensão de uma linguagem, Luger (2013). Primeiro, presume-se uma grande quantidade de conhecimento humano. Os atos de linguagem descrevem relacionamentos em um mundo normalmente complexo. O conhecimento desses relacionamentos deve ser parte de qualquer sistema de compreensão de linguagem. Segundo, uma linguagem é baseada em padrões: fonemas são componentes de palavras e palavras constituem frases e sentenças. A ordenação de fonemas, palavras e sentenças não é aleatória. Não é possível haver comunicação sem uma grande restrição quanto ao uso desses componentes. Finalmente, os atos de linguagem são o produto de agentes, tanto de humanos quanto de um computador. Os agentes estão incorporados em um ambiente complexo com dimensões individual e sociológica. Todos estes pontos levantados devem ser considerados ao se realizar análises textuais.

3.2. Análise sintática

Para a análise sintática, existem técnicas básicas, uma delas sendo a tokenização, que normalmente inicia o processo de análise textural. Nela são separados os termos que compõem o texto, como palavras e pontuações, para poder ser executado um processo individual para cada um deles (Cordeiro, 2017). De certa forma, a tokenização é um pré-processamento; uma identificação de unidades básicas a serem processadas e erros neste estágio induzem a mais erros nas fases seguintes da análise (TRIM, 2013).

Stemming é outra técnica básica que, como definido por Alvares (2005), é a "tarefa de identificar a subcadeia de uma palavra que sirva como uma representação única e não ambígua da mesma, e a de suas diversas variações". O resultado deste processo é chamado de stem e não é necessariamente o mesmo que o radical da palavra. (Cordeiro, 2017).

3.3. Análise semântica

A interpretação semântica vem após a análise sintática e é a que produz uma representação do significado do texto. Esta análise se dá pelo conhecimento sobre o significado das palavras e a estrutura linguística, como papéis de substantivos ou a transitividade de verbos, Luger (2013). Assim, são identificados os agentes, objetos e instrumentos de uma sentença, por exemplo.

O analisador semântico também realiza verificações de consistência no que diz respeito às possibilidades de relações entre objetos, impedindo relações inválidas, Luger (2013). Um exemplo seria o verbo *pilotar* estar associado ao substantivo *estátua*.

3.4. Extração de informação

O conceito de extração de informação é utilizado para extrair partes úteis de uma informação textual, segundo Chowdhury (2013), e normalmente faz uso das análises previamente mencionadas. Um grande número de técnicas é utilizado neste sentido e a extração de informação pode servir diversos propósitos, por exemplo: preparar um resumo de um texto, popular bancos de dados, preencher espaços vazios, ou, como exemplo de maior relevância para este trabalho, identificar palavras-chaves e informações dentro de frases. Isto significa que estas informações extraídas podem ser utilizadas em diferentes processos por outros sistemas, servindo como base para as mais variadas aplicações.

3.5. Big Data

Mayer-Schönberger (2013) afirma não existir uma definição rigorosa do que pode-se considerar Big Data, mas que a ideia inicial foi a de que um conjunto de dados era classificado como Big Data, quando a tamanho do conjunto de dados sendo examinado superava o tamanho de memória dos computadores utilizados para o processamento de dados. O que gerou uma série de inovações tecnológicas mais tarde utilizadas para processar e explorar estes conjuntos de dados, como foi o caso do MapReduce e o Hadoop, desenvolvidos pelos engenheiros da Google. Já Wu et. al (2013) traz um conceito mais direto sobre o que seria o conceito de Big Data, descrevendo o conceito como um conjunto de dados de grande volume, complexos e em crescimento constante, com muitas fontes de dados autônomas.

3.6. Mineração de dados

Finlay (2014) define Mineração de dados como um conjunto de técnicas automatizadas utilizadas para interrogar enormes bases de dados e fazer inferências sobre o que os dados significam. ARUMUGAM et. al (2010) traz uma definição mais simples sobre o termo mas que retrata a intenção por trás das técnicas. Para ele, o termo pode ser definido como fazer melhor uso dos dados. A ideia de se processar os grandes conjuntos de dados para não apenas reduzir o tamanho do conjunto processado, mas também para processar melhores dados, vem como uma grande ajuda ao se trabalhar com Big Data, principalmente quando falamos de diferentes fontes de dados.

3.7. Inteligência artificial

Grande parte deste trabalho está contido na área de estudo de inteligência artificial, englobando desde PLN até as técnicas de mineração de dados citadas na seção anterior. Esta área de pesquisa não é nenhuma novidade no mundo computacional. Gevarter (1984), por exemplo, define IA como uma área de pesquisa da ciência da computação que se destina a criar programas de computador capazes de resolver problemas que se realizados por seres humanos necessitam de inteligência dos mesmos. SHWARTZ (1987) define IA de forma mais sucinta como a capacidade de uma máquina de imitar comportamento humano inteligente. Em ambas as definições é possível ver que a ideia no geral é criar programas considerados inteligentes, de forma que se possa automatizar tarefas que até então apenas humanos poderiam executar.

O conceito de inteligência artificial se encaixa em diversos momentos deste trabalho, desde o processamento de texto, como é descrito na seção destinada a PLN e como foi introduzido na seção de mineração de dados. Nesta segunda, foram apresentadas algumas técnicas que são também áreas de estudo dentro da inteligência artificial e que agora serão melhor detalhadas.

3.8. Reconhecimento de padrões

Fukunaga (1990) apresenta a meta da identificação de padrões como tornar claros os processos de tomada de decisão feitos por humanos e automatizar essas funções utilizando computadores. Luger (2013) também traz uma breve apresentação da responsabilidade dos métodos de reconhecimento de padrões caracterizando-os como métodos para identificação de estruturas ou os padrões nos dados. Utilizando grandes conjuntos de dados como fonte de informação, um sistema de reconhecimento de padrões é capaz de identificar tanto os padrões quanto anomalias nos dados.

3.9. Classificação de informação

Luger (2013) apresenta os métodos de classificação como responsáveis por decidir a qual categoria ou grupo pertence um valor de entrada. Lippmann (1987) traz uma definição mais prática de como funcionam os métodos de classificação. Segundo o autor, um método de classificação, também conhecido como classificador, determina qual classe, de um conjunto de **m** classes, é a mais representativa para um padrão de entrada desconhecida contendo **n** elementos. Os métodos de classificação são bastante úteis no contexto deste trabalho, possibilitando por exemplo dividir as notícias em grupos, definindo se uma notícia é uma influenciadora positiva ou negativa para um empresa.

3.10. Estatística

Devore (2006) apresenta a estatística como um grande meio de ajudar os seres humanos a fazer julgamentos mais inteligentes e tomar decisões, diante de incertezas e variações na

informação recebida. Bussab e Morettin (2002) apresentam uma das áreas da estatística, a estatística inferencial, que tem como objetivo coletar, reduzir, analisar e modelar os dados para fazer uma dedução sobre a população (todos os dados de uma base de dados, por exemplo) à qual alguns dados analisados pertencem, ressaltando a previsão de dados como uma das grandes tarefas da última etapa da inferência estatística e como importante meio de se tomar decisões.

A inteligência artificial e a estatística são duas áreas de estudo distintas, mas possuem algumas fortes ligações. Um bom exemplo dessa ligação entre os dois ramos de estudo, é a regressão. Luger (2013) mostra essa conexão entre os dois mundos explicando o funcionamento dos métodos Bayesianos, visto que estes suportam a interpretação de novas experiências com base nos conhecimentos adquiridos anteriormente. Ter à disposição tanto os métodos estatísticos quanto computacionais como, por exemplo, as redes bayesianas, para entender melhor uma variável através de seu histórico, seja em relação a tempo ou hereditariedade, abre um grande leque de possibilidades para que seja possível fazer inferências a respeito de uma determinada variável. Essa ideia é totalmente aplicável à predição de valores de ações. Identificar como uma ação foi afetada por uma notícia de um determinado tipo no passado pode ser bastante útil para entender a influência de uma notícia semelhante sobre uma ação no presente.

Da mesma forma como com os modelos estatísticos, podemos fazer a predição de variáveis utilizando a inteligência artificial. As redes neurais trazem exatamente a ideia da predição estatística, utilizando diversas variáveis em um processamento para predizer uma nova variável. Luger (2013) traz uma definição para redes neurais apresentando-a como um modelo em camadas onde novas informações são geradas ou informações existentes são adaptadas através das conexões entre as camadas, onde as camadas anteriores têm relação com as camadas posteriores, criando a mesma ideia de informações ancestrais apresentada na definição da predição estatística.

4. Desenvolvimento

4.1. Método

Este trabalho visa prever flutuações relevantes nos valores de ações de empresas de tecnologia, através de modelos de predição que levam em consideração informações relativas às empresas, extraídas de notícias através de processamento de linguagem natural. Para isso, o modelo de predição foi dividido em partes de um processo, onde cada algoritmo é responsável por executar uma ação específica que gera como saída a entrada para a próxima etapa. Dessa forma, o processo completo considera as seguintes etapas: busca de notícias online, busca de valores históricos da bolsa de valores, treinamento do modelo de predição de sentimentos por processamento de linguagem natural, análise de sentimentos de notícias capturadas, estruturação do conjunto de dados, treinamento do modelo de predição e testes do modelo, como apresentado na Figura 1.

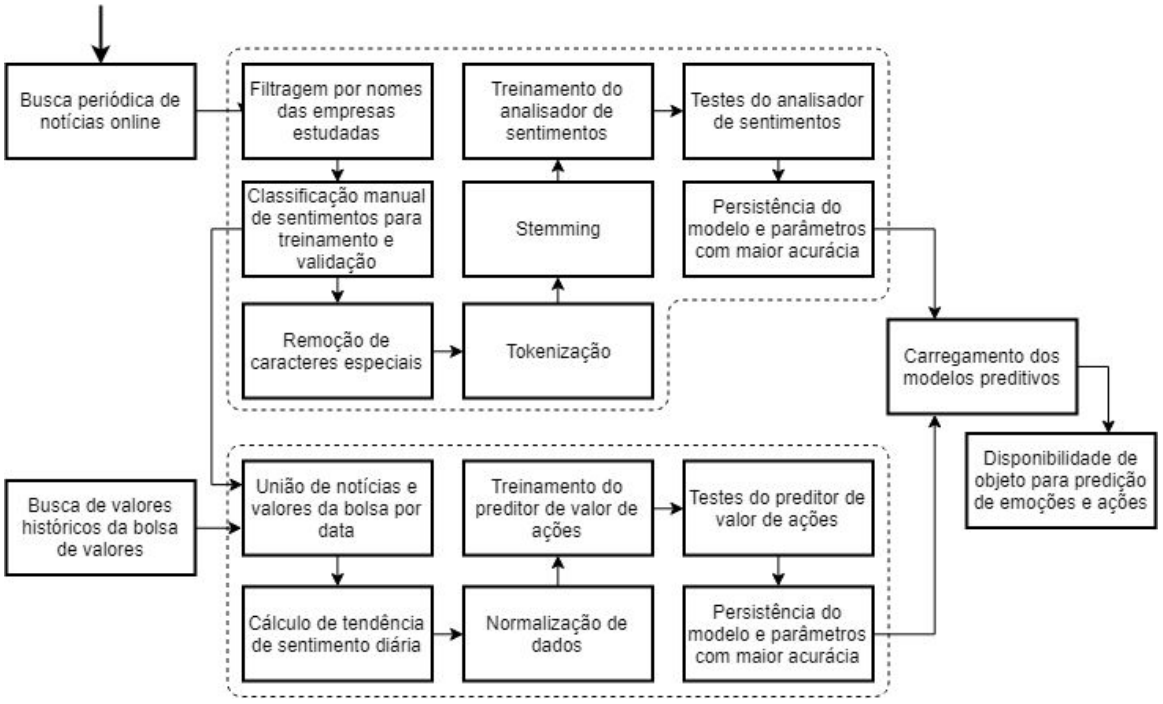


Figura 1 - Diagrama de fluxo de informação do modelo. Processamento de notícias e valores de ação.

4.2. Busca online de notícias

Para fazer o registro de notícias relativas às empresas escolhidas e os assuntos que permeiam este trabalho, foi utilizada uma ferramenta organizadoras de feed RSS, chamada Miniflux. Essa ferramenta faz registro e organização de notícias online, de acordo com sites fonte, categorias de assuntos e horário de publicação de uma notícia. Para utilização da ferramenta, foi criada uma máquina virtual com o sistema operacional Ubuntu 16.04, onde a ferramenta é executada. Ela foi configurada para buscar notícias de tempos em tempos e atualizar o seu banco de dados com as notícias, sempre que uma nova notícia fosse lançada. A ferramenta foi configurada para obter notícias sobre as 11 (onze) categorias distintas de 7 (sete) fontes (Portais de notícias online) distintas. As fontes e categorias selecionadas são apresentadas abaixo:

- **BBC Brasil**: Brasil, Economia e Internacional;
- **Folha de São Paulo**: Blog, Ciência, Mercado, Poder e Tecnologia;
- **G1**: Economia, Política e Concursos e Emprego;
- **O Globo**: Completo;
- **Olhar Digital**: Completo;
- **UOL**: Economia e Tecnologia;
- **Valor Econômico**: Completo.

Com a seleção das fontes e das categorias e configuração destas no Miniflux, foi iniciada a captura de novas notícias periodicamente. As notícias foram capturadas do dia

22/08/2018 até o dia 05/04/2019. O objetivo desta captura é de criar um conjunto de dados heterogêneo, contendo notícias de diversos tipos e fontes distintas, que posteriormente seriam transformadas e filtradas. Isto significa que todas as notícias oriundas das fontes supracitadas dentro de suas categorias são armazenadas no banco em um formato padrão da ferramenta Miniflux, que possui, dentre diferentes atributos, "data" e "conteúdo", como os mais relevantes para esta pesquisa.

No total foram 519 notícias para Apple, 176 para Microsoft e 76 para Tesla, dentre um total de 23.602 notícias. Por fim, estes dados filtrados foram exportados para um arquivo em formato de valores separados por vírgula (comma separated values, CSV), que se assemelha a uma tabela, porém proporciona maior flexibilidade ao algoritmo de treinamento implementado.

4.3. Busca online de notícias

Os valores históricos do índice BOVESPA para as três empresas estudadas foram obtidos do provedor de dados Alpha Vantage, que disponibiliza uma API acessível via protocolo HTTP. Os dados vêm no formato JavaScript Object Notation (JSON). Para cada dia, temos os atributos dos valores de Abertura, Fechamento, Alta e Baixa representados em reais, além do Volume representado por por unidade.

Para tanto, o usuário deve realizar uma requisição HTTP para www.alphavantage.co/query enviando os seguintes parâmetros:

- **function**: qual dos diferentes formatos e agrupamentos de dados será utilizado na busca. Este trabalho utilizou o formato de série temporal diária, denominado pela API como *TIME_SERIES_DAILY*;
- **symbol**: o símbolo da empresa; Microsoft é representada por *MSFT34.SA*, Apple por *AAPL34.SA* e Tesla por *TSLA34.SA*;
- **outputsize**: explicita o tamanho do retorno da API. Neste trabalho utilizamos o formato completo, que é *full*;
- **apikey**: uma chave única que determina qual usuário está realizando a busca, pode ser obtida ao se cadastrar no serviço através de uma conta de e-mail válida.

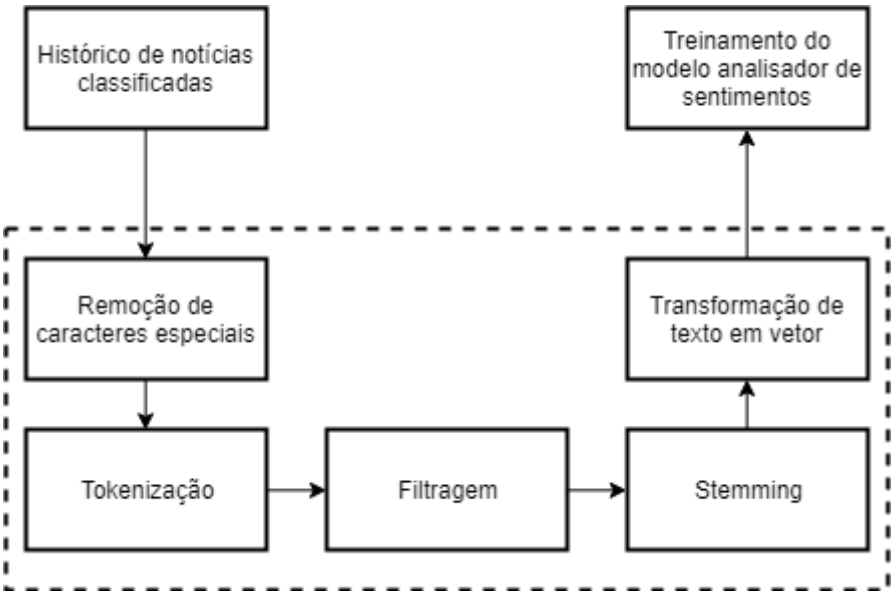
Estão disponíveis dados a partir do dia 29/02/2012 para Apple, 09/12/2010 para Microsoft e 19/01/2017 para Tesla. Todos os registros diários possuem o mesmo formato, desde o primeiro ao dia atual, para as três empresas. A Tabela 1 demonstra dois dias como exemplo para a empresa Microsoft, cujo símbolo é MSFT34.SA:

Data	Abertura	Fechamento	Alta	Baixa	Volume
2013-07-10	78,4100	78,4100	78,4100	78,4100	500
2013-07-11	80,4700	80,4900	80,4900	80,4700	6000

Tabela 1 - Representação de dados de ação da empresa Microsoft em dois dias, adquiridos da API

4.4. Análise de sentimentos de notícias diárias

O modelo proposto neste trabalho procura relacionar notícias sobre empresas multinacionais de tecnologia aos seus valores na bolsa. Esta possível relação se dá pelo sentimento das notícias, motivo pelo qual se faz necessário um modelo analisador de sentimentos. Para atingir este objetivo, são executadas diversas etapas, as quais são detalhadas abaixo e ilustradas pelo diagrama apresentado na Figura 2 Estes mesmos processos foram realizados para as notícias relativas a cada uma das três empresas em estudo, individualmente.



Figuara 2 - Treinamento do analisador de sentimentos.

Diferentes algoritmos foram utilizados no treinamento do modelo analisador de sentimentos, a fim de definir um algoritmo ótimo, ou seja, aquele que apresenta o melhor desempenho em relação a tempo de duração de treinamento e acurácia dos resultados. Estes algoritmos recebem como entrada vetores de palavras, originados das notícias obtidas através da API do servidor Miniflux. Já os sentimentos, que servem como classe para as notícias, foram classificados nestas notícias de forma manual pelos autores deste trabalho, conforme demonstra o diagrama da Figura 3.

Figura 3 - Processo de classificação de sentimento de notícias.

Para a classificação foram utilizados valores inteiros, que são necessários para o processo de treinamento do classificador de sentimentos, sendo possíveis apenas três valores de representação de sentimento para uma notícia:

-1 : Que representa uma notícia negativa;

0 : Que representa uma notícia neutra; e

1 : Que representa uma notícia positiva.

Para as etapas seguintes foi necessário isolar cada palavra da notícia, construindo uma lista de palavras a partir da notícia inteira; este processo é chamado de tokenização. Desta lista, realizamos uma filtragem, ignorando quaisquer palavras menores que três caracteres e também palavras vazias, que são palavras que podem ser consideradas irrelevantes no processo de análise de um texto para extração de sentimento, pelo fato de não trazerem significado quando avaliadas isoladamente, sem a conexão com outros termos, ou seja: necessitam de um contexto, criado a partir de outras palavras a elas conectadas. Alguns exemplos são palavras que servem apenas como conectivos entre outras palavras, encontrando-se a relevância no verbo e no objeto direto da sentença, como: lhe, isso, ser, às, numa, etc.

Uma vez que os termos estejam preparados, realizamos o processo de stemming das palavras, que consiste em transformar uma palavra em qualquer variação para o formato de sua raiz, permitindo ao algoritmo identificar um termo mesmo que ele venha de contextos e com formas diferentes. Ilustramos os processos de tratamento de texto aplicados utilizando o exemplo apresentado abaixo.

Dada uma frase em seu formato original:

> **"fontes internas da apple informaram que empresas devem fazer investimentos na casa de u$ 365 milhões"**

Após os processos de remoção de palavras irrelevantes e stemming, obtemos a seguinte frase:

> **"font intern apple inform empres dev faz invest cas 365 milhõ"**

Por fim, transformamos nossos dados em vetores, uma vez que são compatíveis com todos os diferentes modelos abordados. Para tanto, utilizamos a estratégia de n-gramas, que é o processo de dividir uma frase em pequenas sentenças formadas por n termos sequenciais. O intuito foi de verificar diferentes formas de entrada trariam melhoria para a performance do analisador de sentimentos, considerando não somente a frequência de termos encontrados, mas também a frequência de sequências de termos. Abaixo ilustramos um exemplo de bigrama, que é um n-grama de dois termos, chamado de bigrama.

Dada uma frase em seu formato original:

> **"João disse olá"**

Aplicando a transformação por bigramas, obtemos as seguintes sentenças:

> **"João disse"** e
> **"disse olá"**

Cada dupla de valores do vetor corresponde a uma bigrama possível de nosso conjunto de dados inteiro, considerando todas as notícias já processadas. Ao calcular o vetor de uma nova notícia específica, é criada uma nova tupla com o número de ocorrências de seus bigramas, devidamente mapeados conforme as posições específicas no vetor. Isto significa que o vetor de uma nova notícia se inicializa com todos os valores iguais a zero; conforme os bigramas são encontrados, seus valores correspondentes no vetor são acrescidos, gerando ao fim do processamento o número de ocorrências de cada bigrama.

Mantivemos o formato de n-gramas, porém alteramos o modelo de contagem, que anteriormente era simples, para o modelo "frequência do termo–inverso da frequência nos documentos" (term frequency–inverse document frequency, TF-IDF). Este modelo também

faz a contagem das ocorrências das palavras, porém cada palavra tem seu valor calculado proporcionalmente ao número de ocorrências dela em todas as entradas. Isto significa que se todas as entradas possuem uma palavra, ela não é relevante por não ser específica. Em contrapartida, se apenas um por cento das entradas possui uma determinada palavra, isto significa que aquela é uma palavra chave e de alta relevância quando encontrada.

Mesmo após variadas formas de testes com diferentes configurações de vetorização e modelos de predição, nenhum modelo atingiu uma acurácia desejável acima de 70%. Desta forma, foi definido que cada empresa utilizaria o modelo que lhe trouxesse a melhor performance, sendo Regressão Logística para Apple com 54,94% de acurácia, Gradient Boosting para Microsoft com 56,03% e Árvore de Decisão para Tesla com 56,95%, utilizando todos estes a vetorização em TF-IDF com n-grama onde n é igual a 1. Na seção Resultados (seção 5.1) esses aspectos são discutidos em detalhe.

4.5. Predição de valores de ações

Como a modelagem de dados relativos às ações considera períodos diários e as notícias coletadas podem ter tempos distintos, para fazer a correlação entre o sentimento das notícias e valores de fechamento dos dados históricos e treinar o modelo, as notícias coletadas foram pré-processadas, com o intuito de identificar a tendência de sentimento das notícias de uma determinada empresa para um determinado dia. Neste caso, para o treinamento foram utilizados apenas os dias de fechamento da bolsa nos quais existiam notícias registradas no banco de dados histórico, e para cada dia em que há fechamento de valores de ações, foi calculada a tendência de sentimento do conjunto de notícias levantadas.

O cálculo deste valor foi feito a partir de um somatório do número de ocorrência de cada sentimento. Dessa forma, em um dia no qual foram registradas quatro notícias, das quais três são positivas e uma neutra, é considerado um dia positivo, por exemplo. Abaixo apresentamos as correlações utilizadas para identificar o sentimento predominante em um dia:

Sendo a quantidade de notícias positivas representado por **QP**, a quantidade de notícias negativas representado por **QN** e a quantidade de notícias neutras representado por **QNE**, segue a definição.

Dia positivo: **QP** > **QN** e **QP** > **QNE**

Dia neutro: **QNE** > **QP** e **QNE** > **QN**

Dia negativo: **QN** > **QP** e **QN** > **QNE**

Uma vez obtidas/inferidas/calculadas as tendências de sentimentos para cada dia em que há notícias capturadas e valores de fechamento de ação, é feito um mapeamento dos valores de ações relativos a cada um destes dias. A Tabela 2 demonstra este formato, com dados fictícios.

Data	Abertura	Fechamento	Alta	Baixa	Volume	Sentimento
10/05/2019	469,50	469,88	473,22	469,00	1400	1
11/05/2019	469,88	469,70	470,00	469,15	300	0

Tabela 2 - Representação do modelo com dados concatenados, valores de ação e média de sentimentos de notícias.

Neste ponto, com a tendência de sentimentos de cada dia, temos os dados prontos para treinamento por parte do preditor de valores de ação. Na Tabela 3 podemos verificar três exemplos hipotéticos de como se apresenta o conjunto de dados, sendo a coluna Tendência da Ação o valor usado para classificação das demais colunas que se caracterizam como atributos. Este mesmo formato foi utilizado para experimentar diferentes algoritmos para o modelo preditor de ações.

Abertura	Fechamento	Alta	Baixa	Volume	Tendência de Sentimento	Tendência da Ação
400	390	405	380	0,632	1	0
390	402,5	405	385	0,527	1	1
402,5	385	404	375	0,688	-1	-1

Tabela 3 - Exemplo de dados completos do modelo preditivo, valores relativos à ação, tendência de sentimento de notícia (média do dia) e tendência da ação.

Todos os resultados de acurácia foram maiores que 89%, alguns atingindo inclusive 99%. Supomos que este valor fosse muito alto, devido ao fato de que trabalhos correlatos não alcançaram uma marca tão elevada mesmo utilizando técnicas mais avançadas, não só para predição como também para preparação dos dados. A partir disso, supomos que o motivo da alta acurácia seria devido à margem de diferença entre a abertura e o fechamento diário, estando muito grande para a real tendência dos dados e assim sendo grande parte deles classificada como uma só classe.

A fim de explorar esta hipótese, executamos testes com mais duas configurações de proporção. Uma define como tendência positiva se a proporção for maior ou igual a 1%, negativa se for menor ou igual a -1% e neutra quando se encontra entre 0,99% e -0,99%. Outra foi definida como positiva se maior ou igual a 0,3%, negativa se menor que -0,3% e neutra se estiver entre 0,29% e -0,29%. A tabela Tabela 4 traz uma prévia destes dados para a empresa Microsoft. Para facilitação, vamos chamar a medida original de Medida 3%, a segunda de Medida 1% e a última definida como Medida 0,5%.

Algoritmo	Medida 3%	Medida 1%	Medida 0,3%
Árvore de Decisão	99,71%	87,35%	75,12%
K Vizinhos Mais Próximos	98,23%	75,67%	57,70%
Regressão Linear	99,71%	76,79%	52,94%

Tabela 4 - Comparação de valores de acurácia para diferentes critérios de seleção de tendência do valor das ações.

Como podemos observar, ao reduzir a medida da janela de categorização das diferenças, a acurácia foi reduzida, uma vez que está mais próxima dos dados reais e deixa de classificar a maior parte dos dados como uma só categoria.

5. Conclusão

5.1. Acurácia do classificador de sentimentos

Os modelos selecionados para cada empresa foram aqueles que apresentaram melhor acurácia, independentemente do tipo de algoritmo e parâmetros utilizados. Regressão Logística para Apple, com 54,9% de acurácia, Gradient Boosting para Microsoft, com 56,03% de acurácia e Árvore de Decisão para Tesla, com 56,95% de acurácia. Todos estes recebem como entrada um vetor de parâmetros no formato TF-IDF que considera tokens únicos no texto.

5.1. Acurácia do preditor de flutuação de ações

Todos os modelos obtiveram uma acurácia alta, atingindo o mínimo de 89,95%, porém diversos alcançaram até mesmo 99%. Considerando que os analisadores de sentimentos utilizados obtiveram 54,94%, 56,03% e 56,95% de acurácia para Apple, Microsoft e Tesla respectivamente, números relativamente incertos, a eficácia do valor da tendência dos sentimentos neste modelo foi questionada. Com o propósito de fazer comparações, foi realizado um novo teste, porém ignorando totalmente os valores de tendência de sentimento das notícias. As colunas consideradas neste caso são: Abertura, Fechamento, Alta, Baixa e Volume.

Ao remover os valores de sentimentos diários, a acurácia e o desvio padrão se mantiveram muito semelhantes para todos os modelos. Este fator ajudou a levantar a hipótese de que o número para separar a proporção entre a abertura e o fechamento da ação num dia fosse muito grande.

Foi observado que os resultados de acurácia são menores conforme menor a janela utilizada para separar as tendências diárias em positiva, neutra ou negativa. Isto abre espaço para maiores explorações em trabalhos futuros. Curiosamente os resultados para a empresa Tesla permaneceram elevados.

5.3. Teste prático de funcionamento do modelo completo

Apesar de todos os resultados adquiridos através dos testes realizados em cada um dos modelos utilizados na criação destes trabalho, foi decidido que um novo teste sobre os mesmos, funcionando em conjunto, poderiam agregar ainda mais informações úteis aos resultados obtidos.

Para isso foram buscadas notícias de datas aleatórias do ano de 2019 (dois mil e dezenove) para cada uma das empresas. Foram levantadas notícias de dois dias para cada uma das empresa, cada uma das notícias foi classificada pelos autores deste de forma manual para ter um meio de comparação com a saída do modelo. Após classificadas as notícias, o texto principal (corpo da notícia) foi utilizado como entrada para o preditor de sentimentos do modelo, resultando em uma classe ([-1] negativa, [0] neutra e [1] positiva) atribuída a notícia. Para o desenvolvimento desta etapa de testes do trabalho, foi criada uma tabela com uma aba para cada empresa, nestas são apresentadas as novas notícias capturadas no formato original, os tratamentos de textos executados o resultado esperado da predição de sentimentos e o resultado predito, para cada uma das notícias.

Ao aplicar as notícias ao algoritmo, o mesmo não conseguiu prever, por exemplo, que no dia 08 de março de 2019 a tendência do preço das ações da microsoft seria negativa, mas para os outros 5 dias de notícias (5 de 6 dias levantados) as tendências preditas, de todas as empresas, foram corretas. Neste teste, como já era sabido que a diferença entre abertura e fechamento das ações era menor que o valor de 3% utilizado nos treinamentos, o modelo foi treinado novamente para identificar variações no intervalo de 0,03%. Dessa forma os valores de tendência tiveram uma variação um pouco maior, como pode ser visto na Tabela 36. Por exemplo no 30 de maio de 2019 a tendência das ações da Apple foi positiva, considerando uma variação no intervalo de -0,03% e 0,03%.

Os testes manuais foram executados apenas para dois dias de notícias de cada empresa. Uma vez que a classificação manual, a busca e a separação das notícias é um processo bastante demorado e no caso deste tipo de teste totalmente dependente de uma pessoa para execução de pelo menos metade das ações necessárias. Por conta disso, este teste foi realizado apenas para colaborar com os resultados dos testes dos algoritmos efetuados. Uma vez que se acredita que existe alguma correlação entre valores de ações e as notícias divulgadas sobre as empresas em um dia.

5.4. Resultados do experimento

Podemos afirmar que o objetivo geral deste trabalho foi atingido, sendo desenvolvido com sucesso um modelo preditivo, capaz de relacionar dados históricos de valores de ações de empresas multinacionais a notícias a elas relacionadas, publicadas em portais de notícias online.

O pré processamento de notícias para treinamento do modelo analisador de sentimentos utilizou diversas técnicas e diferentes algoritmos. Contudo, os resultados não demonstraram grandes diferenças em acurácia e se mantiveram sempre abaixo de 60%, o que

é considerável um nível baixo de performance pelos autores deste trabalho. Uma hipótese levantada como motivo da baixa acurácia é o número de notícias disponíveis, de forma que o modelo não possui exemplos suficientes para ajustar seus pesos e performar com eficiência. Além disso, fica a possibilidade de se considerar os títulos das notícias juntamente ao conteúdo das mesmas.

O modelo final prevê flutuações de preços com elevada acurácia; entretanto, tornamos a mencionar aqui o módulo de análise de sentimentos. Embora funcional, não obteve os resultados de acurácia desejados. Isto nos faz questionar a relevância dos sentimentos como entrada do modelo preditivo, visto que até mesmo testes que não incluíram sentimentos como atributos trouxeram resultados muito semelhantes aos testes de cenário contrário.

Uma vez que os testes realizados alcançaram resultados muito semelhantes aos que seriam calculados por uma pessoa, acredita-se que o algoritmo conseguiu de alguma forma executar o que foi proposto, correlacionar notícias de portais online com a tendência do valor de fechamento das ações de uma empresa.

Referências

ASSAF NETO, Alexandre (2001), Finanças corporativas e valor, 3° ed. São Paulo: Atlas.

LUGER, George F. (2013), Inteligência Artificial, 6th ed. São Paulo: Pearson Education.

PAGOLU, Venkata Sasank et al. (2016), Sentiment analysis of Twitter data for predicting stock market movements. International Conference On Signal Processing, Communication, Power And Embedded System (scopes), IEEE. http://dx.doi.org/10.1109/scopes.2016.7955659.

LUSTOSA, Volney Gadelha (2004). O Estado da Arte em Inteligência Artificial. Revista Digital da CVA, Brasília. http://pead.ucpel.tche.br/revistas/index.php/colabora/article/viewFile/60/53.

CHOWDHURY, Gobinda G. (2005), Natural language processing. Annual Review Of Information Science And Technology. Wiley. http://dx.doi.org/10.1002/aris.1440370103.

FAMA, Eugene F. (1991), Efficient Capital Markets: II. The Journal Of Finance. http://dx.doi.org/10.2307/2328565.

ANTUNES, M. A. and PROCIANOY, J. L (2003). Os efeitos das decisões de investimentos das empresas sobre os preços de suas ações no mercado de capitais. Revista de Administração.

SILVA, César Augusto Tibúrcio and PEREIRA, Vinícius Alves dos Santos (2008). FATOS RELEVANTES E SUA INFLUÊNCIA NO PREÇO DAS AÇÕE S NO BRASIL, CONGRESSO USP DE INICIAÇÃO CIENTÍFICA EM

CONTABILIDADE, São Paulo. http://www.congressousp.fipecafi.org/anais/artigos82008/575.pdf.

ALVARES, Reinaldo V. (2005), Investigação do Processo de Stemming na Língua Portuguesa. São Paulo.

TRIM, Craig (2013), The Art of Tokenization. https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization.

FINLAY, Steven (2014), Predictive analytics, data mining, and big data: myths, misconceptions and methods. New York

MAYER-SCHÖNBERGER, Viktor and CUKIER, Kenneth (2013), Big data: a revolution that will transform how we live, work, and think. Boston.

WU, Xindong; ZHU, Xingquan and WU, Gong-qing (2014), Data mining with big data. Ieee Transactions On Knowledge And Data Engineering, Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tkde.2013.109. https://ieeexplore.ieee.org/document/6547630.

ARUMUGAM, Paliah et al. (2010), Financial Stock Market Forecast using Data Mining Techniques. International Multiconference Of Engineers And Computer Scientists, Hong Kong. https://www.researchgate.net/publication/44260645_Financial_Stock_Market_ForEcast_using_Data_Mining_Techniques.

CHEN, Ming-syan; HAN, Jiawei and YU, P.s. (1996), Data mining: an overview from a database perspective. IEEE Transactions On Knowledge And Data Engineering, http://dx.doi.org/10.1109/69.553155. https://ieeexplore.ieee.org/abstract/document/553155/authors#authors

BARIONI, Maria Camila Nardini and Miyashiro, Davis Une (2009), Estudo de Técnicas de Redução de Dimensionalidade. Universidade Federal do ABC. Disponível em: http://ic.ufabc.edu.br/II_SIC_UFABC/resumos/paper_5_151.pdf

GEVARTER, William B. (1984), Artificial intelligence, expert systems, computer vision, and natural language processing. Park Ridge.

SHWARTZ, Steven P. (1987), Applied natural language processing. Petrocelli Books, Princeton.

ABDELAZIZ, et. al. (2018), A Machine Learning Model for Improving Healthcare services on Cloud Computing Environment. https://doi.org/10.1016/j.measurement.2018.01.022. https://www.researchgate.net/profile/Mohamed_Elhoseny4/publication/322892893_A Machine_Learning_Model_for_Improving_Healthcare_services_on_Cloud_Computing_Environment/links/5a762c89a6fdccbb3c07abf4/A-Machine-Learning-Model-for-Improving-Healthcare-services-on-Cloud-Computing-Environment.pdf

FUKUNAGA, Keinosuke. Introduction to statistical pattern recognition. 2nd. ed. San Diego: Morgan Kaufmann, 1990. 591p. (Computer science and scientific computing) ISBN 0122698517

LIPPMANN, R. (1987), An introduction to computing with neural nets. IEEE Assp Magazine, http://dx.doi.org/10.1109/massp.1987.1165576. https://ieeexplore.ieee.org/abstract/document/1165576

BUSSAB, Wilton de Oliveira and MORETTIN, Pedro Alberto (2002), Estatística básica. São Paulo.

DEVORE, Jay L. (2006), Probabilidade e estatística: para engenharia e ciências. São Paulo.

BARBETTA, Pedro Alberto (2012), Estatística aplicada às ciências sociais. 8. ed. Florianópolis