Pontifícia Universidade Católica
do Rio de Janeiro

André Davys Carvalho Melo de Oliveira

Agrupamento de ações por Embeddings Textuais na previsão de preços

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática do Departamento de Informática da PUC-Rio.

Orientador: Prof. Sergio Colcher

Rio de Janeiro
Março de 2020

Pontifícia Universidade Católica
do Rio de Janeiro

André Davys Carvalho Melo de Oliveira

Agrupamento de ações por Embeddings Textuais na previsão de preços

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática do Departamento de Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo.

Prof. Sergio Colcher
Orientador
Departamento de Informática – PUC-Rio

Prof. Ruy Luiz Milidiú
Departamento de Informática – PUC-Rio

Prof. Marco Serpa Molinaro
Departamento de Informática – PUC-Rio

Rio de Janeiro, 10 de Março 2020

André Davys Carvalho Melo de Oliveira

Bacharel em Ciência da Computação (2016) na Universidade Federal do Ceará - Campus Quixadá (UFC)

Ficha Catalográfica

André Davys Carvalho Melo de Oliveira

Agrupamento de ações por Embeddings Textuais na previsão de preços / André Davys Carvalho Melo de Oliveira; orientador: Sergio Colcher. – Rio de janeiro: PUC-Rio, Departamento de Informática, 2020.

v., 72 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui bibliografia

1. Predição de Séries Temporais. 2. Mercado de Ações. 3. Aprendizado de Máquina. I. Sergio Colcher. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Agradecimentos

Aos meus pais, Marcelo Melo e Helena Carvalho, que sempre se esforçaram para me dar a melhor educação e me motivaram a seguir bons caminhos e que muita das vezes sacrificaram seus sonhos em favor dos meus. E não há dúvidas que o que sou hoje é reflexo de seus ensinamentos.

As minhas avós, Rosa Carvalho e Maria Oliveira, que fizeram de tudo para que pudesse sempre ter o do bom e do melhor e sempre me motivaram a nunca desistir de meus sonhos.

Ao Abraão, Álan, Alysson, André, Dalai, João Vitor, Lauro, Luísa, Micaele, Raul, Pedro, Rodrigo, Rômulo, Sérgio e Vinícius, meus irmãos nessa cidade, que sempre me apoiaram a nunca desistir e mesmo apesar das dificuldades e contratempos me deram forças para seguir nesta longa jornada.

Ao meus colegas de laboratório NIT BTG-Pactual, especialmente a equipe do Digibot, equipe na qual pude aprender bastante sobre o mercado financeiro, aprendizado de máquina, engenharia de software e xadrez. Em especial, ao meu amigo Pedro Ferreira, que sempre esteve contribuindo com ideias e sugestões para o desenvolvimento deste trabalho.

Agradeço aos meus orientadores Sérgio Colcher e Rui Milidiú, pela orientação e conselhos, contribuindo para minha vida acadêmica e profissional.

Agradeço a todos os meus amigos e colegas de PUC-Rio, por fazerem de minha jornada aqui inesquecível e que trará boas recordações.

Aos professores que tive durante esta jornada acadêmica que contribuíram bastante para minha formação.

Aos professores participantes da banca examinadora pelo tempo, pelas valiosas colaborações, comentários, críticas e sugestões.

A tia Ângela, por todos os dias fazer o melhor café do departamento de Informática.

A Deus por me proteger, abençoar e me guiar pelo caminho certo, me permitindo estar aqui agradecendo, encerrando um ciclo importante em minha vida.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Resumo

André Davys Carvalho Melo de Oliveira; Sergio Colcher. **Agrupamento de ações por Embeddings Textuais na previsão de preços**. Rio de Janeiro, 2020. 72p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Realizar previsões de preços no mercado de ações é uma tarefa difícil devido ao fato de o mercado financeiro ser um ambiente altamente dinâmico, complexo e caótico. Para algumas teorias financeiras, usar as informações disponíveis para tentar prever o preço de uma ação a curto prazo é um esforço em vão já que ele sofre a influência de diversos fatores externos e, em decorrência, sua variação assemelha-se à de um passeio aleatório. Estudos recentes, como (37) e (51), abordam o problema com modelos de predição específicos para o comportamento do preço de uma ação isolada. Neste trabalho, apresenta-se uma proposta para prever variações de preço tendo como base conjuntos de ações consideradas similares. O objetivo é criar um modelo capaz de prever se o preço de diferentes ações tendem a subir ou não a curto prazo, considerando informações de ações pertencentes a conjuntos similares com base em duas fontes de informações: os dados históricos das ações e as notícias do Google Trends. No estudo proposto, primeiramente é aplicado um método para identificar conjuntos de ações similares para então criar um modelo de predição baseado em redes neurais LSTM (long short-term memory) para esses conjuntos. Mais especificamente, foram conduzidos dois experimentos: (1) aplicação do algoritmo K-Means para a identificação dos conjuntos de ações similares, seguida da utilização de uma rede neural LSTM para realizar as previsões, e (2) aplicação do algoritmo DBSCAN para a criação dos conjuntos seguida da mesma rede LSTM para prever as variações de preço. O estudo foi realizado em um conjunto com 51 ações do mercado acionário brasileiro, e os experimentos sugeriram que utilizar um método para criar conjuntos de ações similares melhora os resultados em aproximadamente 7% de acurácia e *f1-score*, e 8% de *recall* e *precision* quando comparados a modelos para ações isoladas.

Palavras-chave

Predição de Séries Temporais; Mercado de Ações; Aprendizado de Máquina.

Abstract

André Davys Carvalho Melo de Oliveira; Sergio Colcher (Advisor). **Stock Clustering Based on Textual Embeddings Applied to Price Prediction**. Rio de Janeiro, 2020. 72p. Dissertação de mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Predicting stock market prices is a hard task. The main reason for that is due to the fact its environment is highly dynamic, intrinsically complex and chaotic. The traditional economic theories tell us that trying to predict short-term stock price movements is a wasted effort because the market is influenced by several external events and its behavior approximates a random walk. Recent studies, such as (37) and (51), address this problem and create specific prediction models for the price behavior of an isolated stock. This work presents a proposal to predict price movements based on stock sets considered similar. Our goal is building a model to identify whether the price tends to bullishness or bearishness in the (near) future, considering stock information from similar sets based on two sources of information: historical stock data and Google Trends news. Firstly, the proposed study applies a method to identify similar stock sets and then creates a predictive model based on LSTM (long short-term memory) for these sets. More specifically, two experiments were conducted: (1) using the K-Means algorithm to identify similar stock sets and then using a LSTM neural network to predict stock price movements for these stock sets; (2) using the DBSCAN algorithm to identify similar stock sets and then using the same LSTM neural network to forecast stock price movements. The study was conducted over 51 stocks of the brazilian stock market. The results suggested that using an algorithm to identify similar stock clusters yields an improvement of approximately 7% in accuracy and f1-score and 8% in recall and precision when compared to specific models for isolated stocks.

Keywords

Forecasting Time Series; Stock Market; Machine Learning.

Sumário

Lista de figuras

Lista de tabelas

Aquele que trabalha duro pode superar um gênio, mas de nada adianta trabalho duro se você não confia em si mesmo

Rock Lee, *Naruto.*

1
Introdução

O mercado de ações é um lugar onde grandes quantidades de dinheiro são investidas e negociadas todos os dias em todo o mundo, o que tem motivado investidores, analistas financeiros, empresas e organizações a estarem sempre procurando vantagens competitivas que propiciem ganhos. Nesse cenário, prever o comportamento do mercado tem se tornado uma tarefa importante. Por outro lado, as teorias financeiras tradicionais defendem ser impossível predizer o preço de ações no mercado financeiro. Em 1965, Eugene Fama introduziu a *Hipótese do Mercado Eficiente* (HME) (14), que afirma não ser possível desenvolver um sistema de previsão baseado nas informações disponíveis, uma vez que tais informações já estão refletidas no preço atual da ação. No mesmo sentido, o modelo *random walk*, apresentado em (34), compara a variação do preço das ações a curto prazo com "passeios aleatórios na *Wall Street*".

Recentemente, com avanços nas áreas de inteligência artificial, aprendizado de máquina e estatística, pesquisadores têm mostrado cada vez mais métodos capazes de tratar esse problema. Técnicas como *Artificial Neural Networks* (ANN) (2), (7), (36), (40); Algoritmos Genéticos (9), (41); Métodos de Regressão Linear (5), (42), *Support Vector Machines* (SVM) (22), (29); e *Autoregressive Integrated Moving average* (ARIMA) (21), (45) têm se mostrado capazes de prever, com alguma qualidade, o comportamento de ações no mercado.

A predição do comportamento do preço de ações recai em uma importante tarefa de aprendizado de máquina: a predição de séries temporais. Basicamente, nessa tarefa existe o interesse em obter conhecimento sobre o comportamento de uma variável no futuro baseado em observações dessa variável ao longo do tempo. No problema de predição do mercado de ações (*stock market prediction*) a variável observada é o preço da ação e estamos interessados em saber esse preço no futuro baseado no seu histórico em um período de tempo; ou seja, o problema é predizer o valor do preço de uma ação x_{t+1} baseados no seu histórico de preços em um período de tempo, $x_1, x_2, ..., x_t$.

É possível separar as estratégias de predição do mercado de ações em dois grupos: (i) os métodos que usam apenas informações da série temporal,

que são as técnicas baseadas em dados históricos das ações como o preço (17),(49),(51), e (ii) os métodos baseados nesses mesmos dados históricos acrescidos de outras informações textuais associadas ao mercado de ações; geralmente, essas informações adicionais são notícias financeiras ou dados de redes sociais relacionados com o mercado de ações (20),(37), (43), (47). Além disso, existem trabalhos que buscam prever o valor de índices do mercado, que por sua vez são compostos por um conjunto de ações, tais como: S&P 500 (3), Dow Jones (20), iBovespa (10). Existem ainda trabalhos que buscam prever o comportamento do preço de ações específicas (11), (37).

Essa pesquisa parte da hipótese de que os preços de ações de empresas que são influenciados pelos mesmos fatores e variáveis têm a tendência de subir ou cair junto. Na recente tragédia ocorrida na barragem de Brumadinho, por exemplo, estima-se que a Vale sofreu um prejuízo de aproximadamente 70 bilhões. Consequentemente, o mercado financeiro acredita que a tendência é de que os preços de ações que são influenciadas pelos mesmos fatores também caiam, o que motiva encontrar o conjunto de ações que se relacionam entre si ou que estejam sujeitas aos mesmos fatores condicionantes. Nesse sentido, o principal objetivo deste trabalho é:

– Investigar se o uso de informações de um conjunto de ações similares contribui para o modelo de predição.

Os trabalhos encontrados na literatura buscam criar modelos de predição para tratar ações isoladas, assim criam um modelo por ação. Quando os trabalhos criam modelos para tratar conjuntos de ações, eles utilizam a série temporal do índice da bolsa de valores de um mercado (por exemplo, iBovespa) para prever o comportamento deste índice no futuro. Entretanto, não é criado um modelo de predição mais genérico capaz de realizar previsões para um conjunto de ações com diferentes séries históricas de preços. Neste sentido, a principal contribuição desta pesquisa é realizar um estudo para a criação de um modelo capaz de realizar previsões para um conjunto de ações que possuem diferentes históricos de preços, no qual as ações pertencentes a este conjunto são consideradas similares por algum critério. Portanto, para atingir o principal objetivo dessa pesquisa, são definidos os seguintes objetivos específicos:

1. Apresentar uma abordagem para a criação de vetores que possam representar uma ação.

2. Identificar conjuntos de ações similares com base nos vetores que representam as ações.

3. Usar informações de um conjunto de ações similares para criar um modelo de predição para o conjunto.

Neste trabalho, é proposto um método para a criação de vetores para representar ações do mercado financeiro em um espaço vetorial, o que torna possível utilizar uma medida de distância/similaridade entre estes vetores para comparar ações. Estes vetores são chamados de *stock embeddings* e são criados a partir de uma base de documentos textuais que descrevem uma ação utilizando o algoritmo Doc2vec (26).

Após ter ações do mercado representadas em um espaço vetorial através de seus respectivos *stock embeddings*, algoritmos de clusterização são aplicados para selecionar de forma automática conjuntos de ações que são consideradas similares por algum critério. No qual estes conjuntos de ações similares proveem a base para o modelo de predição.

O modelo de predição é treinado em um conjunto de ações similares para identificar as movimentações positivas de preços de cada ação deste conjunto. Para efeito de avaliação, define-se que uma movimentação no preço de uma ação é positiva, se o preço de abertura atual é 1 centavo maior em relação ao preço de abertura no minuto anterior. Neste modelo são avaliadas as métricas de *precision*, *recall*, acurácia e *f1-score* para a identificação de movimentações de preços, em que a classe positiva da classificação representa quando o preço possui uma variação positiva e a classe negativa representa quando o preço não possui essa mesma variação.

O estudo realizado neste trabalho foi testado em um conjunto de 51 ações pertencentes ao índice Ibovespa, um dos principais índices do mercado acionário brasileiro. Os experimentos realizados sugerem que usar informações de ações similares para prever uma ação específica melhoram os resultados em aproximadamente 8% de *precision* e *recall* médios e 7% acurácia e *f1-score* médios quando comparados aos modelos de ações isoladas.

Este trabalho está organizado da seguinte maneira: No Capítulo 2 são apresentados alguns trabalhos que utilizam alguma abordagem de aprendizado para o problema de predição do mercado acionário. No Capítulo 3 são apresentados os conjuntos de dados usados neste trabalho e como tais conjuntos são usados para criar importantes *features* para o modelo de predição. No Capítulo 4 são apresentados os algoritmos usados em cada etapa do estudo realizado nesta pesquisa. No Capítulo 5 é detalhada a metodologia aplicada para a criação deste modelo e, no Capítulo 6, são descritos os experimentos utilizados para a avaliação do modelo proposto e seus resultados.

2
Trabalhos relacionados

A maioria dos estudos que desenvolveram classificadores do mercado de ações usando técnicas computacionais de inteligência, tais como inteligência artificial, algoritmos genéticos, *fuzzy system* e aprendizado de máquina (9), (43), (44) usaram informações quantitativas, cujas características são apenas dados históricos, como preço, índices e indicadores de análise técnica (29), (45). Outros trabalhos, em menor quantidade, além de utilizar dados históricos também usam dados textuais, geralmente notícias financeiras ou o humor proveniente de mídias sociais relacionadas ao mercado de ações (20), (37).

Na Seção 2.1 mostra-se que utilizar indicadores de análise técnica para prever o mercado de ações podem ser bastante úteis e geram uma melhora na qualidade dos resultados. Na Seção 2.2 são apresentados trabalhos que usam dados textuais para auxiliar na etapa de predição. Finalmente, na Seção 2.3 são mostrados trabalhos que usaram alguma metodologia de aprendizado para realizar previsões sobre ações isoladas do mercado acionário brasileiro.

2.1
Análise técnica

Kimoto e Asakawa (24) criaram um sistema para predizer quais são os melhores momentos de compra e venda de ações do TOPIX(*Tokyo Exchange Prices Indexes*) usando redes neurais. Para auxiliar na predição de pontos de compra e venda das ações, os autores usaram indicadores técnicos de impulso (*Momentum technical indicators*) o que gerou predições mais precisas sobre o mercado, e consequentemente, nas simulações de compra e venda de ações os indicadores geraram um maior lucro. Este trabalho segue a ideia de Kimoto e Asakawa (24) para utilizar os indicadores técnicos de impulso como *features* adicionais para o modelo de predição.

Cheng, Chen e Wei (9) em 2010, propuseram um método baseado em algoritmos genéticos e na teoria de conjuntos aproximados (*rough sets*) para prever o índice da bolsa de valores do mercado de Taiwan (TAIEX). Os autores usam a teoria de conjuntos aproximados para selecionar indicadores técnicos essenciais na previsão do preço futuro de uma ação. Feito isso, usam algoritmos genéticos para refinar as regras extraídas dos indicadores e obter

uma previsão mais precisa sobre o preço futuro. Nos resultados é mostrado que usar os indicadores para criar recursos adicionais na etapa de previsão, trazem melhorias nos quesitos de acurácia e retorno absoluto do preço das ações.

Wei (46) usa indicadores técnicos para "alimentar" um sistema ANFIS (*adaptive network-based fuzzy inference system*) e realizar predições do índice da bolsa de valores do mercado de Taiwan (TAIEX). Wei usa algoritmos genéticos para otimizar a escolha dos hiperparâmetros do ANFIS. Para avaliar o método o autor usa a raiz dos erros quadrados médios (RSME) e consegue uma melhora média de 2.6 pontos por ano, alcançando o estado da arte para o TAIEX.

O presente trabalho combina os indicadores técnicos usados por (9) e (46) para criar um conjunto de indicadores. Esses indicadores são usados como recursos adicionais para o modelo de predição e atuam como informações de especialistas de domínio para o modelo de predição. Todos os indicadores usados no modelo estão descritos detalhadamente na Seção 3.1.1.

2.2 Fontes externas

Em 2015, Attigeri *et. al* (5) criaram um modelo de predição usando recursos analíticos de *big data*, análises de mídia social e aprendizado de máquina para prever a tendência no mercado de ações. Em seus resultados os autores mostram que o mercado de ações é afetado por notícias políticas e econômicas e que as mídias sociais influenciam no preço das ações.

Em 2016, Li *et. al* (30) estudou a relevância da mudança de humor dos usuários do Twitter quando eles compartilham *tweets* sobre as tendências dos preços das ações da empresa para prever o comportamento do mercado de ações. Seu estudo mostra que as palavras relacionadas às emoções têm um certo grau de correlação com a tendência geral do mercado de ações. Especialmente, os *tweets* tristes têm um impacto significativamente maior no mercado de ações do que em outras classes (feliz, raiva, medo e surpresa).

Sun *et. al* (43) em 2017, coletaram dados de diferentes mídias sociais (microblogs, salas de bate-papo, fóruns da web) e investigaram alguns modelos de aprendizado de máquina para fazer uma análise de sentimentos dos *posts* nas mídias sociais. Os autores encontraram uma forte correlação entre o sentimento em postagens de salas de bate-papo e o comportamento do mercado de ações, o que indica que ele pode ser usado como uma *feature* para melhorar a previsão do comportamento do mercado de ações. Esta abordagem alcança uma precisão de $71,3\%$ usando um comitê de modelos que incluem máquinas de vetores de suporte (SVM), regressão linear (LR), Naive-Bayes (NB) e rede neural

recorrente de memória de curto prazo (LSTM).

Em 2018, Hu *et. al* (20) propuseram um algoritmo *sine cosine* (ISCA) para otimizar os pesos e *bias* da *back propagation neural networks* (BPNN). Em outras palavras, os autores criaram uma nova arquitetura de rede neural, combinando ISCA e BPNN, para prever as movimentações do preço dos índices do mercado acionário americano (*Dow Jones Industrial Average* (DJIA) e S&P 500). O estudo mostra que o uso de dados provenientes do Google trends melhoram a previsão do mercado de ações, alcançando uma precisão de $88,98\%$ para o índice DJIA e uma precisão de $86,81\%$ para o índice S&P 500. É possível concluir que recursos extraídos da plataforma do Google trends são promissores para realizar predições no mercado de ações.

Assim como nos trabalho citados anteriormente, nesta pesquisa também é utilizado dados de fontes externas para auxiliar o modelo de predição. Motivados por (21), este trabalho utiliza dados de manchetes de notícias coletadas da plataforma do Google Trends para criar *features* textuais e assim gerar mais recursos para auxiliar o modelo de predição. A ideia é que estas notícias forneçam conhecimento sobre os principais eventos e assuntos que influenciem o mercado financeiro.

2.3 Mercado acionário brasileiro

De Faria *et. al* (10), realizaram um estudo comparando redes neurais e métodos de suavização exponencial para a predição do preço do principal índice do mercado brasileiro (Ibovespa). Nos experimentos realizados, os autores mostram que redes neurais artificiais performam melhor, em termos de RMSE (raiz dos erros quadrados médios), do que métodos de suavização exponencial.

De Oliveira *et. al* (12) propuseram um modelo de redes neurais para prever as mudanças do preço de ações do mercado brasileiro. Nos experimentos apresentados os autores avaliaram o modelo com a ação PETR4, negociada na BM&FBOVESPA, da companhia Pétrobras. Os autores usam indicadores de análise técnica para criar recursos adicionais na etapa de predição e avaliam o modelo proposto usando a métrica POCID (*Prediction of Change in Direction*). Eles testaram diferentes configurações de tamanho da janela de predição (passos à frente a serem preditos) e alcançaram um POCID de 93.62% no conjunto de teste e 87.5% no conjunto de validação.

De Melo (11) desenvolveu um modelo baseado em redes neurais LSTM para predizer se o preço de abertura de uma ação específica tende a subir, descer ou permanecer estável no próximo minuto. Para treinar o modelo de aprendizado, o autor utiliza os *top trends* com seus artigos relacionados, cole-

tados a partir da plataforma do *Google Trends*, e os dados históricos de preços da ação. Para avaliar o modelo proposto, o autor utiliza a métrica de acurácia para as ações PETR4 (PETROBRAS), ABEV4 (AMBEV) e ITSA4 (Itausa) alcançando uma acurácia de 69.24%, 67.42% e 69.66%, respectivamente.

Nelson *et. al* (37) conduziram um estudo aplicando redes neurais LSTM para prever se o preço de uma ação específica tende a subir ou não nos próximos 15 minutos. No treinamento do algoritmo de predição foram usados os dados históricos de preços e indicadores de análise técnica dos últimos 10 meses. Os autores realizaram experimentos nos ativos BOVA11, BBDC4, CIEL3, ITUB4, PETR4 verificando as métricas de acurácia, *precision*, *recall* e *F1-score*. Os autores comparam as redes LSTMs, *Multi-Layer perceptron*, *random forest* e um método aleatório baseado na distribuição de probabilidade das classes. De acordo com os resultados é possível concluir que as redes LSTM, em geral, são melhores do que as outras abordagens.

O modelo de predição construído neste trabalho é inspirado nas arquiteturas apresentada por (11) e (37). Basicamente, o modelo de predição consiste em uma rede neural LSTM com recursos de indicadores de análise técnica, assim como foi usado em (37), e manchetes de notícias do Google Trends, como usado em (11). É importante destacar que os trabalhos citados criam um modelo de predição para tratar uma ação isolada, diferentemente dos modelos apresentados nesta pesquisa, que por sua vez, cria um modelo de predição para tratar um conjunto de ações. Cada etapa para a construção desse modelo mais genérico capaz de realizar previsões de preços para conjuntos de ações é descrita em detalhes na Seção 5.3.

3
Descrição dos conjuntos de dados

Neste trabalho são utilizados dois conjuntos de dados principais, um que possui informações dos preços das ações no mercado, chamados de *dados temporais*, e outro que possui informações textuais de notícias do mercado e a descrição de cada ação do mercado, chamadas de *dados textuais*.

O conjunto de dados temporais descreve a série temporal de preços para cada ação, contendo informações do mercado financeiro como: preço de abertura, preço de fechamento, volume de negócios, quantidade negociada etc. Esse conjunto é apresentado detalhadamente na Seção 3.1.

Já o conjunto de dados textuais possui manchetes de notícias sobre o mercado financeiro de diferentes *websites* de jornais, como *Money Times*[1], Uol[2], Estadão[3], *Space Money*[4], O Globo[5] etc. Além disso, existe um subconjunto de textos que descrevem cada ação contida no conjunto de dados temporais. O conjunto de dados textuais, incluindo notícias e descrição das ações, é descrito em mais detalhes na Seção 3.2.

3.1
Conjunto de dados temporais

O conjunto de dados temporais consiste no histórico de preço das ações coletadas no *ftp-site* da B3 – Brasil Bolsa Balcão S.A. A B3, antiga BMFBOVESPA – é a bolsa de valores mais importante do Brasil.

A B3 é uma das principais empresas de infraestrutura do mercado financeiro, incluindo os índices Ibovespa, IBrX-50, IBrX, Itag, entre outros. A empresa reúne uma tradição de inovação em produtos e tecnologia e é uma das maiores em valor de mercado, com uma posição global proeminente no setor do mercado de ações (52).

Diariamente, a B3 fornece dados de mercado de seus ativos financeiros via ftp [6]. A partir desses dados, De Melo (11) construiu um conjunto cole-

[1] `https://moneytimes.com.br`
[2] `https://noticias.uol.com.br`
[3] `https://economia.estadao.com.br`
[4] `https://spacemoney.com.br`
[5] `https://oglobo.globo.com/`
[6] `ftp://ftp.bmf.com.br/MarketData`

Tabela 3.1: Amostra da série histórica de preços para a ação VIVT4 no dia 15/08/2016.

hora	**symbol**	**open**	**close**	**max**	**min**	**volume**	**negócios**	**qtt**
10:03	VIVT4	45,55	45,55	45,55	45,55	32116	2	700
10:04	VIVT4	45,56	45,80	45,81	45,52	119727	15	2600
10:05	VIVT4	45,78	45,77	45,78	45,77	18443	19	400
10:06	VIVT4	45,84	45,74	46,02	45,50	410711	60	8900
10:07	VIVT4	45,59	45,69	45,69	45,59	55184	68	1200
10:09	VIVT4	45,74	45,75	45,75	45,74	55286	72	1200
10:10	VIVT4	45,71	45,73	45,73	45,66	27608	78	600
10:11	VIVT4	45,72	45,74	45,74	45,71	23030	82	500
10:12	VIVT4	45,71	45,74	45,74	45,61	427813	108	9300
10:13	VIVT4	45,75	45,94	46,04	45,74	993426	187	21500

tando informações *intraday* do preço histórico para diferentes ações em uma granularidade de 1 minuto no período de 15/08/2016 a 30/11/2016.

Na Tabela 3.1 é apresentado um exemplo de série temporal com o preço histórico da ação VIVT4. As informações coletadas nas colunas são: o preço de abertura (*open*), o preço de fechamento (*close*), o preço máximo (*max*), o preço mínimo (*min*), o volume negociado (*volume*), a quantidade de negociações (*negócios*) e a quantidade negociada (*qtt*).

Juntando as séries temporais de cada ação e contabilizando suas movimentações sobre o preço de abertura, tem-se 466.018 (34,8%) variações de alta e 875.355 variações (65,2%) de baixa, evidenciando um conjunto de dados não balanceado.

Na predição do mercado financeiro é comum analistas financeiros buscarem por recursos ou fontes que tragam alguma informação adicional para auxiliar nessa atividade. Consequentemente, ao longo dos anos, pesquisadores e economistas criaram uma porção de indicadores de análise técnica que capturam diferentes informações dada uma série histórica de preços de uma ação. Visto isso, cientistas da computação e desenvolvedores de software criaram ferramentas e bibliotecas para gerar automaticamente os indicadores de análise técnica de um histórico de preços de uma ação. Com a intenção de incrementar as informações para enviar ao modelo de predição foram criados diferentes indicadores de análise técnica. Esses indicadores que são usados como *features* são descritos logo adiante.

3.1.1 Indicadores de Análise Técnica

Os indicadores de análise técnica usados na fase de predição estão descritos na Tabela 3.2. Cada um desses indicadores são obtidos da biblioteca

TA-Lib. O *TA-Lib* é amplamente utilizado pelos desenvolvedores de software de negociação que precisam realizar análises técnicas dos dados do mercado financeiro. É usado o *wrapper* do Python para *TA-Lib* [7] para criar seis indicadores técnicos, em que cada um desses indicadores são gerados automaticamente dada a série histórica de uma ação.

Indicador técnico	Descrição
RSI - *Relative Strength Index*	Um indicador que mede a velocidade e a mudança das variações de preços
MA - *Moving average*	É uma média móvel de diferentes subconjuntos do conjunto de dados completo.
ROC - *Rate of change*	Um indicador que mede a variação percentual no preço de um período para o próximo.
CMO - *Chande Momentum Oscillator*	Captura os ganhos e as perdas recentes para o variação de preços durante um período.
PPO - *Percentage Price Oscillator*	Um indicador que mostra a diferença percentual entre duas médias móveis. O sinal do PPO indica pontos promissores para compra e venda.
MACD - *Moving Average Convergence Divergence*	É a diferença entre duas médias móveis exponenciais. O MACD sinaliza mudanças de tendência e indica o início da nova direção da tendência.

Tabela 3.2: Descrição dos indicadores técnicos aplicados na fase de predição.

O indicador RSI (*Relative Strength Index*) criado por J. Welles Wilder (48) está descrito na fórmula 3-1.

$$RSI = 100 - (\frac{100}{1 + \frac{P}{N}}) \tag{3-1}$$

em que P é o valor médio de mudanças positivas dos preços e N é o valor médio de mudanças negativas dos preços. Este indicador busca por divergências entre os preços máximos e mínimos das ações, em que estas divergências podem indicar ou não uma reversão iminente, ou seja, se o preço de uma ação está subindo, devido a reversão iminente, o preço desta ação irá descer e vice-versa.

O indicador MA (*Moving Average*) é simplesmente uma média de um determinado período de tempo, em que este período vai aumentando com o passar do tempo. Suponha que, temos uma sequência de n dias, os valores $x_1, x_2, ...x_n$ são os preços da ação nos n dias e que o tamanho da janela de tempo é $\frac{n}{2}$. Neste caso, é calculada $\frac{n}{2}$ médias, em que a primeira média é

[7] http://mrjbq7.github.io/ta-lib

uma média aritmética da sequência $(x_1, x_2, ..., x_{\frac{n}{2}})$, a segunda média é a média aritmética da sequência $x_2, x_3, ..., x_{\frac{n}{2}+1}$, e assim sucessivamente.

Para uma sequência $Y = \{y_1, y_2, ..., y_n\}$ e uma sequência $X = \{x_1, x_2, ..., x_m\}$, em que X é uma subsequência contígua de Y, o indicador MA é generalizado na fórmula 3-2.

$$MA = \frac{\sum_{i=1}^{k} x_i}{k} \tag{3-2}$$

em que k é o tamanho da sequência X, tal que $X = \{y_j, y_{j+1}, ..., y_{j+k}\}$ e j é o índice do primeiro preço da sequência Y.

O indicador ROC(*rate of change*) mede a variação percentual no preço entre o preço atual e o preço de k dias atrás da ação. O indicador ROC é descrito na equação 3-3.

$$ROC = \frac{x_i - x_{i-k}}{x_{i-k}} \cdot 100 \tag{3-3}$$

em que x_i é o preço atual e x_{i-k} é o preço de k dias atrás.

O indicador CMO (*Chande Momentun Oscilator*) desenvolvido por Tushar S. Chande (8) calcula a diferença entre a soma dos ganhos recentes e a soma das perdas recentes e, em seguida, divide o resultado pela soma de todos os movimentos de preços em um período. O indicador CMO é descrito na equação 3-4.

$$CMO = \frac{ups - downs}{ups + downs} \cdot 100 \tag{3-4}$$

em que *ups* é a soma das movimentações de alta e *downs* é a soma das movimentações de baixas no preço da ação. Com este indicador, é possível capturar as informações das oscilações entre os ganhos e perdas recentes no preço.

O indicador PPO (*percentage price oscilator*) é a diferença percentual entre duas médias móveis (*moving average*). PPO está descrito na equação 3-5.

$$PPO = \frac{sMA - fMA}{fMA} \cdot 100 \tag{3-5}$$

em que sMA e fMA são duas médias móveis, tal que sMA é calculada em um período maior do que a de fMA, ou seja, sMA é uma média móvel lenta e fMA uma média móvel rápida. Quando PPO é positivo temos promissores pontos para compra, pois o preço tende a subir no futuro. Em contrapartida, quando o PPO é negativo temos promissores pontos para venda, pois o preço tende a cair no futuro.

O indicador MACD (*moving average convergence divergence*) criado por Gerald Appel (4) é a diferença entre duas médias móveis exponenciais, uma rápida e uma lenta. Uma média exponencial de um período é definida em 3-6.

$$MME_i = (x_i + MME_{i-1}) \cdot K + MME_{i-1} \tag{3-6}$$

em que x_i é o preço atual e $K \in (0, 1)$ é uma constante exponencial.

O indicador MACD é calculado através da equação 3-7.

$$MACD = MME(k) - MME(j) \tag{3-7}$$

em que MME é a média móvel exponencial (definido em 3-6), k e j são os tamanhos do período de tempo, tal que $k < j$. Os valores para k e j usados foram 12 e 26, respectivamente.

3.2 Conjunto de dados textuais

Um de nossos conjuntos de dados textuais é composto pelas manchetes das principais notícias extraídas da plataforma do Google Trends[8] a cada minuto. Dessa forma, esse conjunto de dados consiste nas "notícias do momento" e estas são enviados para o modelo de aprendizado com a função de trazer informações sobre os principais assuntos e tópicos que estão sendo comentados no mundo. A hipótese é que estas informações provenientes do Google Trends forneçam conhecimento dos principais acontecimentos do mundo para o modelo de aprendizado.

O Google Trends é uma página da web pública que analisa a popularidade das principais pesquisas com base em buscas no buscador do Google. Na página inicial do Google Trends é possível explorar os principais assuntos que estão em alta em tempo real acessando os *trends stories*. Esses *trends stories* dependem da tecnologia do grafo de conhecimento do buscador do Google, do Google Notícias e do YouTube, para detectar quando os tópicos e assuntos são tendências nessas três plataformas.

Esse grafo de conhecimento do Google é um sistema, lançado pelo Google em maio de 2012, que visa entender fatos sobre eventos e lugares do mundo real e como essas entidades estão conectadas entre si. Assim, assumimos que, usando as informações do grafo de conhecimento do Google, o modelo de predição adquira conhecimento sobre os fatos e acontecimentos que influenciem o mercado de ações. Esses dados textuais geram os *news embeddings* descritos detalhadamente no Capítulo 5 A seguir, é mostrado exemplos do nosso conjunto de dados de notícias:

[8] `https://trends.google.com.br/trends/?geo=BR`

```
1. Vale vira para queda e Ibovespa passa a
   cair; dólar sobe e volta aos R$ 3,17.

2. Petróleo sobe em meio a expectativas de
   que Opep se mova para limitar produção.

3. Dólar avança após Temer indicar
   preocupação com câmbio; Bolsa sobe.
```

O outro conjunto de dados textuais é usado para descrever uma ação do mercado financeiro e foi coletado a partir do site da BMFBOVESPA[9], em que nesse site é possível obter informações como: o perfil da empresa, a sua atividade principal, a sua classificação setorial, seu *website* e códigos (coluna *symbol* da Tabela 3.1) de negociação no mercado de ações.

Para cada empresa, tentamos de forma automática enriquecer os dados acessando seu respectivo *website* e coletando os textos que a descrevem. Nessa fase, por meio de expressões regulares, no *website* de cada empresa são buscados os textos que fazem referência a alguma das seguintes palavras-chaves:

```
 empresa, perfil, quem somos, visão geral,
apresentação, institucional, companhia,
sobre, historia, institutional, company,
about, presentation, profile, who we are
```

Dessa forma é possível obter mais informações que descrevem cada uma das empresas. Feito isso, todas essas informações são concatenadas para criar um único documento que representa cada empresa. A seguir é mostrado um exemplo de documento que descreve a ação PETR4 obtido pelo método descrito anteriormente:

[9] `http://bvmf.bmfbovespa.com.br/cias-listadas/empresas-listadas/BuscaEmpresaListada.aspx?idioma=pt-br`

```
Petróleo, Gás e Biocombustíveis; Exploração, Refino e Distribuição; 33.000.167/0001-01; Nossa marca e nossa identidade são compostas por diversos elementos, que comunicam nosso jeito de ser. Nós estamos presentes em 19 países dos continentes listados abaixo, administrando a exploração de óleo e gás destas áreas. Através de joint ventures e demais parcerias, nossas unidades incorporam o mais avançado em tecnologia, mantendo-se referência mundial no setor energético.; Conheça outras empresas que fazem parte do Sistema Petrobras, como a Petrobras Distribuidora e a Transpetro.; Petróleo. Gás E Energia, PETROLEO BRASILEIRO S.A. PETROBRAS.
```

Para atingir o objetivo deste trabalho, estes documentos são usados para criar representações vetoriais para cada ação do mercado financeiro, assim tornando possível mensurar a similaridade ou distância entre duas ações em um espaço vetorial. Os métodos PV-DM e PV-DBOW, descritos na Seção 3.3.2, são usados para a geração destes vetores, convenientemente, denominamos estes vetores de *stock embeddings*. Na Seção 3.3.1 são mostrados diferentes métodos para criar representações para dados textuais. Esses métodos foram utilizados para a criaçãõ dos *news embeddings* e *stock embeddings* usados neste trabalho.

3.3 Representando Dados Textuais

Em aplicações de Aprendizado de Máquina que usam dados textuais para obter alguma informação sobre o domínio é necessário usar algum método para representar estes dados. Para isso, durante os anos foram desenvolvidos diferentes métodos para representar e extrair informações de textos, tais como: saco de palavras (*bag-of-words*), TF-IDF (*frequency-inverse document frequency*) e vetores densos de palavras (*word embeddings*). Visto que este trabalho utiliza dados textuais provenientes de notícias extraídas do Google Trends e documentos de textos que descrevem uma ação coletados do B3 (Brasil Bolsa Balcão S.A), diferentes formas para a realização desta tarefa são estudadas nesta pesquisa.

O TF-IDF é uma forma simples de extrair informações de textos para o uso em modelos de *machine learning*. Este algoritmo é uma medida estatística

usada para determinar a importância de uma palavra em um documento pertencente a uma coleção de documentos (corpus).

Para cada palavra de um documento, este método calcula um valor baseado na frequência da palavra naquele documento (Term Frequency) e no inverso da porcentagem de documentos em que aquela palavra aparece (Inverse Document Frequency), de modo que a palavra com maior valor de TF-IDF em um documento tende a estar mais relacionada a ele (31).

3.3.1 Representação Espacial de Palavras

A representação de textos em vetores densos de palavras surgiu devido a falta de escalabilidade dos métodos BoW (*Bag-of-Words*) e TF-IDF. Pois para uma grande quantidade de dados utilizar tais métodos são inviáveis computacionalmente. Desde então, diversos algoritmos ((15), (35), (32), (39)) foram criados para gerar vetores de palavras em diversas línguas, incluindo o português do Brasil.

Em 2013, Mikolov *et. al* (35) propôs dois modelos de redes neurais para representar vetorialmente palavras: o CBoW (*continuous bag of words*) e o Skip-gram, descritos na Figura 3.1. Ambos modelos mostram como redes neurais usam o contexto de um texto para aprender a representar cada uma de suas palavras.

Para estes dois modelos assuma que a palavra atual de uma sentença é w_i e o contexto é dado pela sequência de palavras $w_{i-k_1}, \ldots, w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}, \ldots w_{i+k_2}$, em que k_1 e k_2 são contantes e parâmetros de cada modelo.

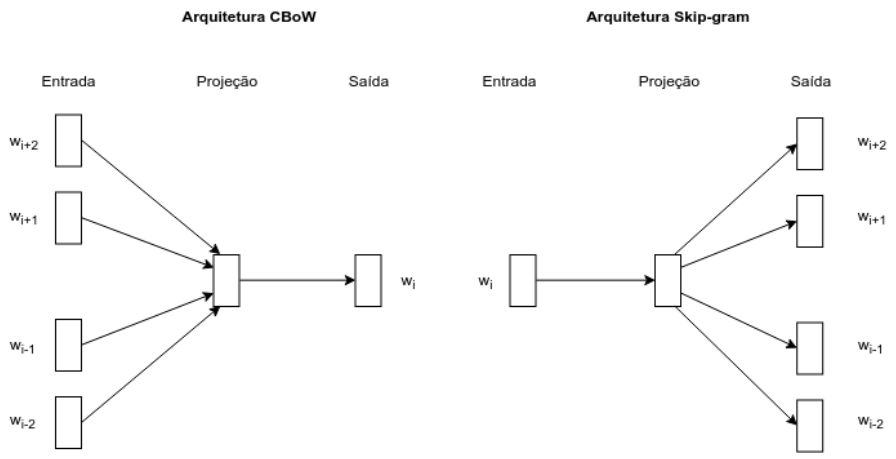

Figura 3.1: Arquiteturas dos modelos CBoW e Skip-gram. Adaptado de (35).

O modelo CBoW busca predizer a palavra atual dado o contexto de uma sentença, ou seja, a entrada da rede CBoW é a sequência de pala-

vras $w_{i-k_1}, \ldots, w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}, \ldots, w_{i+k_2}$ e sua saída deveria ser a palavra w_i. Em contrapartida, o modelo Skip-gram busca predizer um contexto dada a palavra atual de uma sentença. Dessa forma, a entrada da rede Skip-gram é a palavra w_i e sua saída deveria ser o contexto $w_{i-k_1}, \ldots, w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}, \ldots, w_{i+k_2}$. Note que para ambos os modelos o tamanho do contexto é parametrizável e depende da configuração de cada implementação. Uma diferença entre os resultados obtidos a partir dos dois modelos destacada por Mikolov (35), é que o Skip-gram funciona bem com uma pequena quantidade de dados de treinamento e possui boas representações para palavras ou frases raras. Já o CBoW possui representações melhores para palavras mais frequentes e é bem mais rápido para treinar do que o Skip-gram. Atualmente, é possível encontrar vetores pré-treinados que implementam estes dois métodos, estes vetores de palavras pré-treinados são conhecidos na literatura como *word2vec* (35).

Em 2015, Ling (32) fez pequenas mudanças nos modelos *word2vec* (Skip-gram e CBoW) para gerar representações de vetores de palavras mais refinadas que incorporam também a sintaxe do contexto. Estes modelos mais sofisticados são conhecidos como *wang2vec*. Os autores mostram que usando *wang2vec* ao invés de *word2vec*, eles obtêm um resultado melhor para as tarefas de *pos-tagging* e *dependency parsing* na área de processamento de linguagem natural.

3.3.2 *Doc2vec*

Outro método para a criação de vetores para representar palavras é o *Doc2vec* ou *paragraph vector*. Diferentemente das abordagens da Seção 3.3.1, esse é um modelo de aprendizado não supervisionado que busca por representações vetoriais densas para parágrafos, sentenças ou documentos, independentemente de seu tamanho (26).

Na Seção 3.3.1, foi visto que os vetores de palavras contribuem para a predição da próxima palavra em um dado documento. Do mesmo modo, os vetores do *Doc2vec* contribuem para essa tarefa trazendo informações ausentes do contexto atual e esses podem atuar como uma memória para o assunto do documento. Em (26) é apresentado dois métodos para a criação destes vetores, o *Distributed Memory Model of Paragraph Vectors* (PV-DM) e *Distributed Bag of Words of Paragraph Vectores* (PV-DBOW).

Na Figura 3.2, é apresentada a arquitetura do modelo PV-DM, em que na camada de entrada $w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}$ são os vetores de cada palavra da sentença e D_k é o vetor *doc2vec*. Essa arquitetura é similar a do modelo CBOW apresentada na Figura 3.1, a diferença é que há uma "palavra" especial

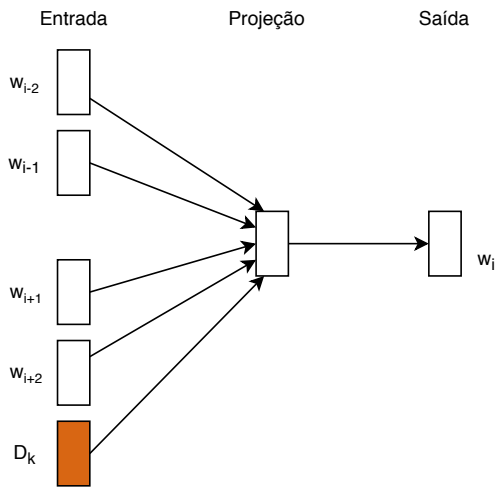

Figura 3.2: Arquitetura do modelo PV-DM. Adaptado de (26).

D_k que traz informação do contexto/tópico do documento (26). Em cada passo do treinamento, o modelo atualiza os pesos dos vetores do documento e de cada palavra. Basicamente, esse algoritmo é uma adaptação do *word2vec* para gerar vetores que representam documentos. Esses vetores gerados podem ser usados para tarefas como encontrar semelhanças entre frases, parágrafos ou documentos. Já o modelo PV-DBOW é semelhante ao modelo Skip-gram apresentado na Figura 3.1, enquanto o Skip-gram buscar representar as palavras em vetores, PV-DBOW pretende representar vetorialmente as informações de um documento.

4
Modelos de Clusterização e Previsão

Neste Capítulo são apresentados os métodos de previsão usados neste trabalho. Na Seção 4.1 são apresentados métodos para realizar as clusterizações a partir dos *stock embeddings* que representam uma ação. Na Seção 4.2 é apresentado o algorítmo de predição utilizado para identificar as movimentações de preços de ações no mercado.

4.1
Clusterização

Agrupar objetos e produtos para os seres humanos é uma tarefa fácil, já que conseguimos identificar padrões levando em consideração vários atributos como cor, forma, tamanho, peso de forma rápida e simples. Para as máquinas o conceito é semelhante, porém, bem mais desafiador. O processo para agrupamento de dados baseia-se no conceito de similaridade, no qual a ideia principal é encontrar itens semelhantes de acordo com seus atributos.

Visto essa necessidade, a clusterização é uma técnica de aprendizado não supervisionado que busca agrupar dados de forma automática baseado em alguma métrica de similaridade ou distância, em que os objetos que pertencem ao mesmo grupo (*cluster*) são considerados similares.

É possível encontrar na literatura diferentes algoritmos para realizar a clusterização de objetos. Com o intenção de encontrar ativos financeiros similares, neste trabalho foram testados dois famosos algoritmos, o K-Means (33) e o DBSCAN (13).

4.1.1
K-Means

O K-Means (33) é um método de aprendizado não supervisionado que busca encontrar objetos similares e agrupá-los em k *clusters*, em que k é um hiperparâmetro do modelo. Dado um conjunto $O = \{o_1, o_2, \cdots, o_n\}$ de n objetos, o método busca dividir os objetos em k *clusters* minimizando a métrica WCSS (*within-cluster sum of squares*). Essa métrica calcula a soma das distâncias ao quadrado de cada objeto dentro de um *cluster* para seu centróide ou média. Estas médias são pontos artificiais que serão usados como

referências para calcular a distância entre os objetos. A função objetivo do K-Means é definida na Equação 4-1.

$$\underset{s}{arg\,min} \sum_{i=1}^{n} \sum_{x \in S_i} \|x - \mu_i\|^2 \tag{4-1}$$

em que μ_i é a média dos objetos que estão no *cluster* S_i.

O algoritmo é dividido em três etapas: escolher centróides, atribuir objetos aos *clusters* e atualizar os centróides. O passo a passo do algoritmo para obter os k *clusters* é descrito a seguir:

1. O algoritmo inicialmente escolhe k centróides arbitrariamente e os usa para atribuir objetos a cada *cluster*.

2. Seja o_i o objeto que será atribuído a algum *cluster* e $S = \{s_1, s_2, \cdots, s_k\}$ os centróides de cada *cluster* e d uma função que calcula a distância entre dois objetos, fazemos $\underset{s_j}{arg\,min}(d(o_i, s_j) \mid \forall s_j \in S)$ para adicionar o objeto o_i ao *cluster* que ele está mais próximo.

3. Uma vez que obtemos novos objetos para cada *cluster* pelo passo anterior, cada centróide é recalculado com a média dos valores de seu respectivo *cluster*.

4. Os passos 2 e 3 são repetidos até que os *clusters* não sejam atualizados ou até atingir o número máximo de iterações do algoritmo.

4.1.2 DBSCAN

O algoritmo DBSCAN (*Density Based Spatial Clustering of Application with Noise)* é um método de clusterização baseado em densidade, proposto por Ester *et. al* (13), capaz de identificar *clusters* com diferentes tamanhos e menos sensíveis à *outliers*.

A ideia do método é que, se um ponto está presente em um *cluster*, então a vizinhança deste ponto possui no mínimo k pontos no mesmo *cluster*, em que k é um parâmetro do método. A vizinhança de um ponto p é definida pela Equação 4-2.

$$N(p) = q \in P \mid 1 - s(p, q) < \epsilon \tag{4-2}$$

em que P é o conjunto de pontos, s é função de similaridade entre dois pontos e ϵ é um limiar de distância.

Se a vizinhança de um ponto p possui mais do que k pontos, então p é denominado como um **ponto central**. Se a vizinhança de um ponto p é menor

que k mas contém algum ponto central, então p é denominado como um **ponto de borda**.

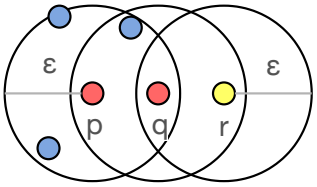

Figura 4.1: Exemplo da vizinhança dos pontos.

Para o número mínimo de pontos $k = 3$, na Figura 4.1 os pontos vermelhos representam os pontos centrais e os pontos azuis e o ponto amarelo representam os pontos de borda. Além disso, dizemos que os pontos azuis e o ponto q são **alcançáveis por densidade diretamente** a partir de p, uma vez que p alcança no mínimo k pontos

Cada ponto p é **conectado por densidade** a um ponto q, se existe um ponto r tal que p e q são alcançáveis por densidade a partir de r. Na Figura 4.1, o ponto r é conectado por densidade ao ponto p, pois r é alcançado por densidade a partir de q, que por sua vez, é alcançado por densidade a partir de p.

Dada essas definições é possível definir que um *cluster* C obtido pelo método DBSCAN é um conjunto de pontos conectados por densidade, tal que C satisfaz as seguintes restrições:

1. $\forall(p, q) \mid$ se $p \in C$ é alcançável por densidade a partir de p, então $q \in C$
2. $\forall(p, q) \in C \mid p$ é conectado por densidade a q.

Dessa forma, o método busca por um *cluster* verificando a vizinhança de um dado ponto. O algoritmo inicia com um ponto p escolhido arbitrariamente. Se p é um ponto central, então um novo *cluster* é criado contendo o ponto p como um centróide e todos os pontos alcançáveis por densidade a partir de p. Caso p seja um ponto de borda, então o algoritmo busca pelo próximo ponto. Esse procedimento é repetido iterativamente até que nenhum ponto possa ser adicionado em algum *cluster* ou atingir o número máximo de iterações.

4.2 Redes Neurais

Inspirados no sistema neural do cérebro humano, pesquisadores desenvolveram um modelo de neurônio matemático capaz de aprender e generalizar padrões a partir da experiência. As redes neurais artificiais são compostas por esses neurônios/nós interligados, em que cada nó recebe um sinal de entrada,

a informação de outros nós e estímulos externos; processa essa entrada através de uma função de ativação e produz um sinal de saída para outros nós e saídas externas (50). Ao longo dos anos, pesquisadores usaram esta ideia de neurônio para criar diferentes modelos de aprendizado.

Em 1982, Hopperfield (19) introduz o conceito de "memória" para as redes neurais artificiais propondo as RNNs (redes neurais recorrentes). As redes recorrentes são comumente usadas em problemas que trabalham com sequências e séries temporais. Em 1989, LeCun *et. al* (28) propõem as CNNs (redes neurais convolucionais), essas redes vem sendo aplicada com sucesso em visão computacional e em processamento e análise de imagens. Em 1997, Hochreiter e Schmidhuber (18) adicionam células LSTM (*Long Short-Term Memory*) às RNNs para criar um novo modelo de aprendizado, as redes neurais LSTM, essa rede vêm sendo usada como uma solução para o *vanishing gradient problem*, problema comum em redes neurais recorrentes.

4.2.1 Redes Neurais Recorrentes

As redes neurais artificiais tradicionais são algoritmos de aprendizado supervisionado que não levam em consideração a sequência dos dados para fazer predições. Entretanto, em muitas aplicações ter informação da sequência é importante. Por exemplo, para prever a próxima palavra de um texto é interessante conhecer as palavras que o precedem em uma sentença, ou para predizer o preço futuro de uma ação no mercado é conveniente saber a informação do preço da ação no presente (27). Para resolver este tipo de problema, as RNNs são usadas. Uma RNN é um tipo de rede neural artificial projetada para reconhecer padrões em sequências de dados, como texto, áudio ou dados de séries numéricas. Essas arquiteturas possuem uma espécie de memória, permitindo assim capturar informações sobre o que foi computado anteriormente. Na teoria, com o uso das RNNs é possível capturar informações de sequências arbitrariamente longas, mas na prática é limitada apenas a informação de poucos passos anteriores. Mais formalmente, uma RNN é uma função que mapeia os elementos x_i de uma sequência $X = x_0, x_1, \ldots, x_t$ em elementos o_i de uma sequência $O = o_0, o_1, \ldots, o_t$ tal que o_i depende de todos os elementos da sequência $X' = x_0, \ldots, x_{t'}, \quad \forall t' \leq t$.

Basicamente, a rede possui unidades de entrada $U = x_0, x_1, \ldots, x_{t-1}, x_t$, unidades de saída $V = o_0, o_1, \ldots, o_{t-1}, o_t$ e unidades escondidas $W = s_0, s_1, \ldots, s_{t+1}, s_t$, em que as unidades escondidas são o estado atual da rede e atuam como a memória da rede em cada passo. Na Figura 4.2, para o tempo t, temos a entrada x_t, o estado oculto s_t e a saída o_t. O estado atual s_t é dado por

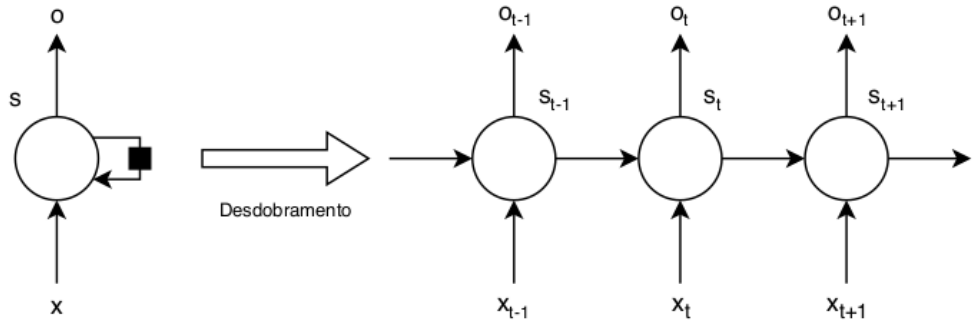

Figura 4.2: Desdobramento do *loop* de uma RNN ao longo do tempo. Adaptado de (27).

$s_t = f(x_t, s_{t-1})$, em que $f : \mathbb{R}^{n\times m} \times \mathbb{R}^{r\times s} \to \mathbb{R}^{n\times m}$ é uma função de ativação (como tanh ou ReLU) que mapeia o estado do passo anterior s_{t-1} e a entrada do passo atual x_t para um novo estado s_t. Dessa forma, o estado atual da rede implicitamente captura informações de todos os estados anteriores.

Nas RNNs os gradientes do erro são calculados em relação aos parâmetros U, V e W, e algoritmos de gradiente estocástico são aplicados para estimar bons parâmetros da rede (23). Para realizar o cálculo dos gradientes em cada passo é usado o algoritmo BPTT (*backpropagation throughout in time*), que por sua vez é uma variação do algoritmo *backpropagation* (6).

As RNNs sofrem do *vanishing gradient problem*, esse problema faz com que o gradiente da função, em cada passo do algoritmo BPTT, exploda (vá para infinito) ou "mingue"(vá para 0). Assim, tornando impossível o aprendizado da rede. Como uma forma de minimizar esse problema, Hochreiter e Schmidhuber (18) propoem as redes LSTM.

4.2.2 *Long Short-Term Memory* (LSTM)

As redes *Long Short-Term Memory* (LSTM), propostas por Hochreiter e Schmidhuber (18), são uma variação das redes neurais recorrentes projetadas para guardar informações de longo prazo evitando o *vanishing gradient problem*. As LSTM são inspiradas nas unidades da memória RAM de um computador, em que em cada unidade da camada oculta é possível ler, gravar e excluir informações de computações anteriores.

Nessa arquitetura, as informações são armazenadas em estruturas chamadas de células. As células, apresentadas na Figura 4.3, são responsáveis por armazenar e manipular informações na "memória" da rede. Para manipular as informações nas células, são usados três portões: o *forget gate*, *input gate* e

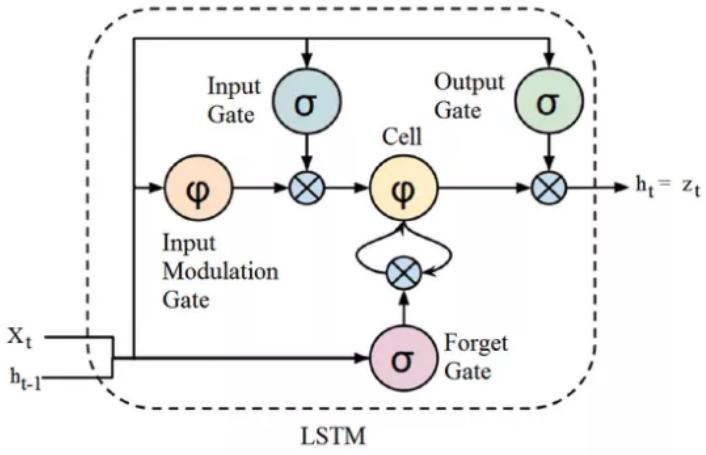

Figura 4.3: Estrutura de uma célula LSTM.

output gate. O processamento de uma célula LSTM é dado pelo conjunto de Equações a seguir:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{4-3}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4-4}$$

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{4-5}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{4-6}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{4-7}$$

$$h_t = o_t \cdot tanh(C_t) \tag{4-8}$$

em que h_{t-1} é a saída da célula anterior, x_t é a entrada atual da célula, W é a matriz de pesos e b é o vetor de *bias* (38).

No processamento de uma célula LSTM, o primeiro passo é decidir qual informação será "esquecida" (excluída da memória). Para isso é usada uma camada sigmóide manipulada pelo *forget gate*, essa camada recebe x_t e h_{t-1}, realiza a operação descrita na Equação 4-3 e retorna a saída f_t. Se para um determinado estado de célula $f_t = 0$, então a informação é esquecida, se $f_t = 1$, então a informação é armazenada para uso futuro.

No próximo passo o *input gate* é usado para decidir quais novas informações serão armazenadas (adicionadas na memória). Para isso, recebemos x_t e h_{t-1} e foi realizado a operação descrita na Equação 4-4, com o intuito de selecionar quais valores da memória serão atualizados. Feito isso, x_t e h_{t-1} são enviados para uma camada *tanh*, em que nesta camada a operação descrita na Equação 4-5 é realizada para obter novos valores, $\tilde{C}_t$.

Para atualizar a memória é aplicada $f_t \cdot C_{t-1}$ para "esquecer" as informações da memória e $i_t \cdot \tilde{C}_t$ para obter novos valores que serão adicionados na

memória, esse processo é descrito pela Equação 4-6.

A função de extrair informações úteis de uma célula LSTM e apresentá-las como uma saída é feita pelo *output gate*. Primeiro, outra camada sigmóide é usada para decidir quais informações serão "lembradas" para gerar um vetor de saída o_t (operação descrita na Equação 4-7). Feito isso, a operação da Equação 4-8 é realizada para gerar uma saída h_t, esta por sua vez, será a entrada da próxima célula.

5
Metodologia para criação do modelo

Este trabalho tem por principal objetivo realizar um estudo para a criação de um modelo capaz de realizar previsões de movimentações de preço para um conjunto de ações similares no mercado financeiro. Para efeito de avaliação deste trabalho, define-se que uma movimentação no preço de uma ação é positiva, se o seu preço teve uma variação de α em relação ao preço no minuto anterior, em que $\alpha = 0.01$ representa 1 centavo. Na Figura 5.1 é apresentada a metodologia adotada neste trabalho para cumprir o objetivo.

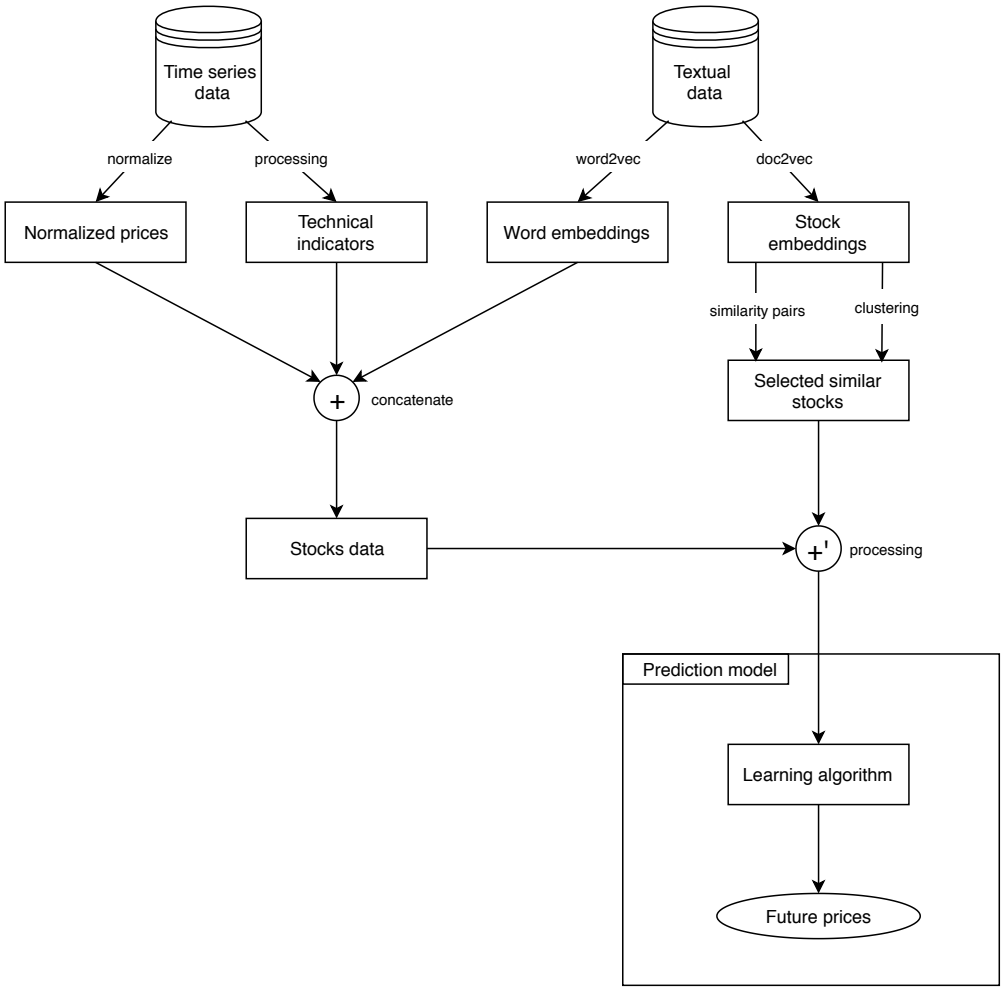

Figura 5.1: Metodologia aplicada para desenvolver o preditor para um conjunto de ações.

A partir da Figura 5.1 é possível visualizar que inicialmente os dados são usados para criar *features* para o modelo de predição. No qual os históricos são utilizados para criar indicadores técnicos e os dados textuais são usados para gerar vetores de *news embeddings* e *stock embeddings*.

Os *news embeddings* recebem as manchetes de notícias coletadas e são representações vetoriais dos *top trends* coletados do Google Trends em cada período de tempo, assim trazem informações das "notícias ou assuntos do momento" para o modelo de aprendizado. Os *news embeddings* são concatenados com os dados temporais para criar os conjuntos de dados das ações que são o insumo utilizado para realizar o treinamento do modelo de predição.

Os *stock embeddings*, por sua vez, recebem os documentos que descrevem uma ação e são vetores que representam as ações em um espaço vetorial em comum, no qual estes vetores são utilizados por métodos de clusterização para selecionar conjuntos de ações consideradas similares.

Após selecionar os conjuntos de ações similares a partir dos *stock embeddings*, a metodologia proposta cria um modelo de aprendizado para cada conjunto, ou seja, o modelo é treinado a partir das dos *news embeddings* e das séries temporais das ações pertencentes a este conjunto. Semanticamente este modelo possui informações das "notícias do momento" provenientes dos *news embedding* e de ações que são influenciadas pelos mesmos fatores do mercado obtidas a partir dos *stock embeddings*. Dessa forma, este modelo criado para o conjunto similar pode observar padrões entre as ações para realizar suas previsões. Portanto, dado um conjunto de ações similares é criado um único modelo de aprendizado capaz de realizar predições para diferentes ações.

Na próxima Seção é mostrado o pré-processamento que é realizado nos dados antes de serem enviados para o modelo de predição. Na Seção 5.2 é definido o processo para identificar conjuntos de ações similares a partir de seus *stock embeddings* utilizando os métodos de clusterização descritos na Seção **??**. Na Seção 5.3 é detalhada a arquitetura do modelo de aprendizado proposto pela metodologia descrita neste trabalho.

5.1 Pré-processamento dos dados

Neste trabalho, para cada ação, são criados indicadores de análise técnica com base em seus dados históricos, conforme descrito na Seção 3.1.1. Dessa forma, nosso conjunto de *features* históricas é composta pela união das colunas que possuem informações quantitativas da Tabela 3.1 com o conjunto de indicadores de análise técnica. Com o intuito de normalizar os dados antes de enviá-los para o modelo de predição, para cada *feature* de nossos dados é

aplicada a transformação da Equação 5-1.

$$transform(x) = \frac{x}{max(X) - min(X)} - \frac{min(X)}{max(X) - min(X)}, \quad \forall x \in X \quad (5\text{-}1)$$

em que X representa uma *feature* de nossos dados temporais.

Dessa forma, os dados temporais presentes no conjunto de treino são normalizados no intervalo $[0, 1]$, em que 0 e 1 são os valores mínimos e máximos de cada coluna, respectivamente. Esses dados possuem as séries temporais que são preditas pelo modelo de predição proposto neste trabalho.

5.2 Identificando conjuntos de ações similares

Este trabalho utiliza dois métodos de clusterização para selecionar os conjuntos de ações similares a partir dos *stock embeddings* que representam as ações. Assuma que o conjunto $S = \{S_1, S_2, \cdots, S_k\}$ representa k *clusters*, uma ação a é considerada similar a uma ação b se, e somente se, existe S_i tal que $a \in S_i$ e $b \in S_i$, para todo $S_i \in S$. Em outras palavras, a ação a é considerada similar a ação b se após o processo clusterização a e b pertencem ao mesmo *cluster*.

Com a intenção de gerar uma visualização dos *stock embeddings* em um espaço de 2 dimensões, é reduzida a dimensionalidade de cada vetor utilizando o algoritmo MDS (*Multi-dimensional Scale*), comumente aplicada para essa tarefa. Na Figura 5.2 é apresentada uma visualização dos *stock embeddings* representadas em um espaço de 2 dimensões.

Como dito anteriormente, é importante identificar os conjuntos de ações similares, pois a hipótese é de que ações similares estejam sujeitas às mesmas variáveis e são influenciadas pelos mesmos fatores. Portanto, para realizar essa tarefa são utilizados os métodos de clusterização K-Means e DBSCAN, descritos na Seção 4.1. Na Seção 6.2 são apresentados os resultados para cada um dos experimentos realizados para os métodos de K-Means e DBSCAN.

5.3 Arquitetura do modelo de predição

A arquitetura do modelo de predição é composta por quatro camadas, baseada nas redes neurais LSTM, conforme descrito na Figura 5.3. A camada de entrada recebe uma tupla (x_i, y_i), em que x_i é um vetor que contém as informações quantitativas de cada ação de um *cluster* e y_i é o vetor médio dos *embeddings* das notícias mais acessadas, e envia para a primeira camada escondida (camada LSTM). A camada LSTM, no instante t, recebe a tupla

Figura 5.2: Ações presentes em nosso conjunto de dados representadas em um espaço de 2 dimensões.

(x_i, y_i) e a rede tenta capturar informações da sequência para gerar um vetor de pesos w_i, em que este vetor w_i é enviado como entrada para a camada seguinte.

Na camada seguinte, temos uma camada densa com uma função de ativação linear definida pela Equação 5-2.

$$f(x) = W \cdot x + B \tag{5-2}$$

Essa camada densa recebe a saída da camada LSTM para gerar um vetor com os preços no instante $t + 1$ de cada ação presente em um *cluster*. Feito isso, esse novo vetor, que possui as predições de preço futuro de cada ação é enviado para a saída da rede. Para avaliar a variação de preço, para cada ação é verificado se o preço predito é maior ou não em relação ao seu preço anterior.

Na fase de treinamento, o modelo recebe a tupla (x_i, y_i) e busca otimizar os parâmetros da rede minimizando o erro quadrado médio definido na Equação 5-3.

$$MSE = \frac{1}{n} \cdot \sum_{i=0}^{n} (\hat{x}_i - x_i)^2 \tag{5-3}$$

em que x_i é o preço real no instante i, $\hat{x}_i$ é o preço predito no instante i e n é a quantidade de exemplos a serem preditos.

O treinamento do modelo é realizado em lotes (*batchs)*, ou seja, em cada passo do treinamento uma subsequência do conjunto de dados de treino é enviada para o modelo. Consequentemente, os parâmetros da rede são

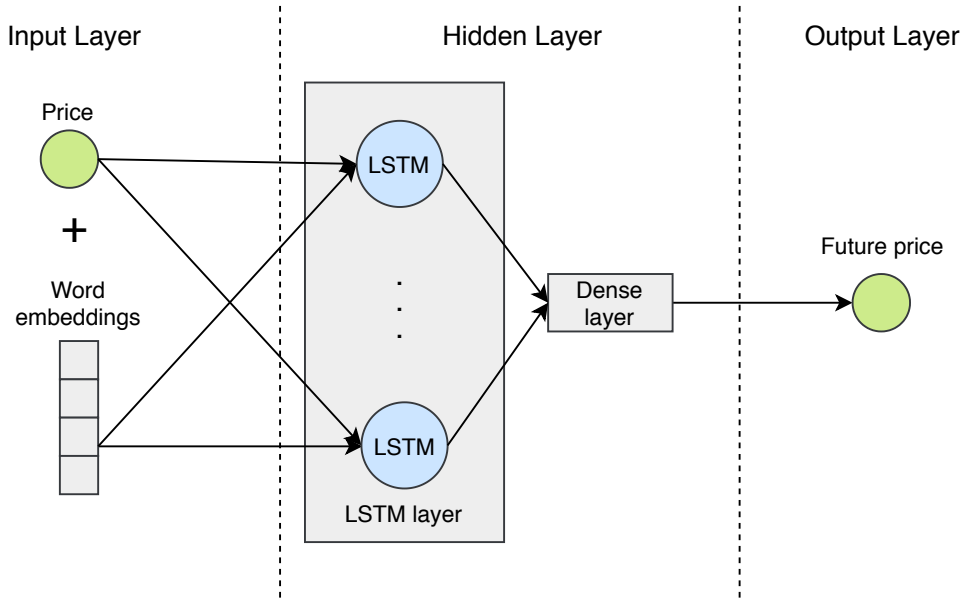

Figura 5.3: Arquitetura do modelo de predição LSTM.

atualizados com base no MSE em cada *batch*, esse processo é repetido algumas vezes até o fim de uma época de treinamento, que acontece no momento em que os dados de treino acabam. No final de cada época, o modelo treinado é avaliado utilizando o erro absoluto médio, definido pela Equação 5-4. É importante lembrar que a quantidade de épocas de treinamento e o tamanho do lote são hiperparâmetros do modelo.

$$MAE = \frac{1}{n} \cdot \sum_{i=0}^{n} (\hat{x}_i - x_i) \tag{5-4}$$

em que x_i é o preço real no instante i, $\hat{x}_i$ é o preço predito no instante i e n é a quantidade de exemplos a serem preditos.

Existem diferentes algoritmos para lidar com o problema do gradiente descendente no treinamento de redes neurais artificiais. Neste trabalho foi adotado o *Adaptive Moment Estimation* (Adam) *Optimizer* (25), um algoritmo de otimização baseada no gradiente descendente estocástico. Este é comumente usado para treinar modelos de aprendizado profundo pois o método consegue lidar bem com esse problema.

O treinamento do modelo termina quando o modelo atinge a quantidade máxima de épocas. Feito isso, temos um modelo treinado para um *cluster*. Portanto, a metodologia proposta nesse trabalho utiliza a arquitetura descrita acima e cria um modelo de predição para cada *cluster* gerado no processo de clusterização.

Vale ressaltar que todo o procedimento descrito acima foi desenvolvido em Python 3 utilizando a biblioteca do *tensorflow* v1.15 (1). No próximo Capítulo, são apresentados os resultados dos modelos para cada *cluster*, em

que esses são obtidos pelos diferentes processos de clusterizações.

6
Resultados

Para avaliar o desempenho do modelo, usamos métricas comumente usadas em problemas de classificação. As métricas usadas são acurácia (6-1), *precision* (6-2), *recall* (6-3) e *f1-score* (6-4).

Essas métricas são calculadas com base nas previsões feitas corretamente para a classe positiva (tp) e negativa (tn), e nas predições feitas incorretamente para a classe positiva (fp) e negativa (fn). Neste trabalho, uma previsão é correta se a variação de preço para alta ou não, é um tp ou um tn, respectivamente. Uma previsão é considerada incorreta se a variação de preço para alta ou não, é um fp ou um fn, respectivamente.

$$acuracia = \frac{tp + tn}{tp + tn + fp + fn} \tag{6-1}$$

$$precision = \frac{tp}{tp + fp} \tag{6-2}$$

$$recall = \frac{tp}{tp + fn} \tag{6-3}$$

$$f1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{6-4}$$

Neste trabalho, foram realizados diferentes experimentos para validar as *features* usadas na etapa predição.

6.1
Geração dos *embeddings*

Neste trabalho foi utilizado o modelo CBOW e Skip-gram para criar diferentes *news embeddings* para representar as notícias do Google Trends. Esses *embeddings* criados foram comparados com os *word embeddings* pré-treinados de word2vec (35) e wang2vec (32) para decidir qual representação traz mais informação sobre os acontecimentos que influenciam o mercado. Foi realizado um simples experimento utilizando um modelo LSTM de uma camada treinado para uma ação em que este modelo recebe o histórico de preços da ação PETR4 e os *news embeddings* de cada um dos modelos. Neste experimento, avaliamos a métrica de *f1-score* para decidir qual modelo de *news embeddings* seria utilizado nos experimentos futuros.

Seja uma notícia que é composta por n palavras, para representá-la, os vetores de cada palavra presente nessa notícia são somados e divididos por n, ou seja, o vetor de uma notícia é formado pela média dos vetores de suas palavras, no qual o vetor de uma palavra é obtido através dos modelos de *news embeddings*.

De acordo com os experimentos realizados foi possível perceber que utilizar os *news embeddings* criados pelo método CBOW geram os melhores resultados em termos de *f1-score* para prever as variações de preços de uma ação no futuro. Além disso, utilizar os vetores pré-treinados que alcançam o estado da arte em diferentes atividades de Processamento de Linguagem Natural não trazem bons resultados quando aplicados no contexto do mercado de ações. Portanto, para avaliar o modelo de predição deste trabalho, as *features* de dados temporais são usadas juntamente com os vetores de notícias obtidos pelo método CBOW, para prever se as variações de preços futuro das ações são maiores ou não do que o preço atual.

Neste trabalho, o método de (26) é usado para criar vetores que representam cada ação. Para a criação dos vetores de documentos de uma ação são realizados experimentos usando os dois métodos apresentados na Seção 3.3.2. Foi feita uma avaliação empírica dos vetores gerados em cada um dos métodos e foi observado que ao usar o modelo PV-DM obtivemos melhores representações. Estes *stock embeddings* são usados para identificar os conjuntos de ações similares do mercado financeiro.

6.2 Experimentos das clusterizações

Nessa seção é apresentado como foi conduzido os experimentos para definir os hiperparâmetros de cada método de clusterização. Assumindo que um *cluster* contém m ações, o modelo de predição proposto neste trabalho recebe as informações históricas das m ações, seus indicadores técnicos e as notícias mais acessadas do Google Trends para retornar um vetor com os preços futuros das m ações. Essa execução é ilustrada na Figura 6.1, em que a "caixa preta" é o modelo gerado pela arquitetura descrita na Seção 5.3.

Para selecionar os conjuntos de ações similares é usada a distância euclidiana para definir a distância entre duas ações e são aplicados os algoritmos K-Means e DBSCAN no processo de clusterização. Como dito anteriormente, o método K-Means assume que a quantidade de *clusters* é conhecida. Todavia, neste trabalho essa informação não está disponível. Visto isso, o método do cotovelo é utilizado para estimar um bom valor para a quantidade de *clusters*. A ideia deste método é executar o K-Means variando a quantidade de *clusters*

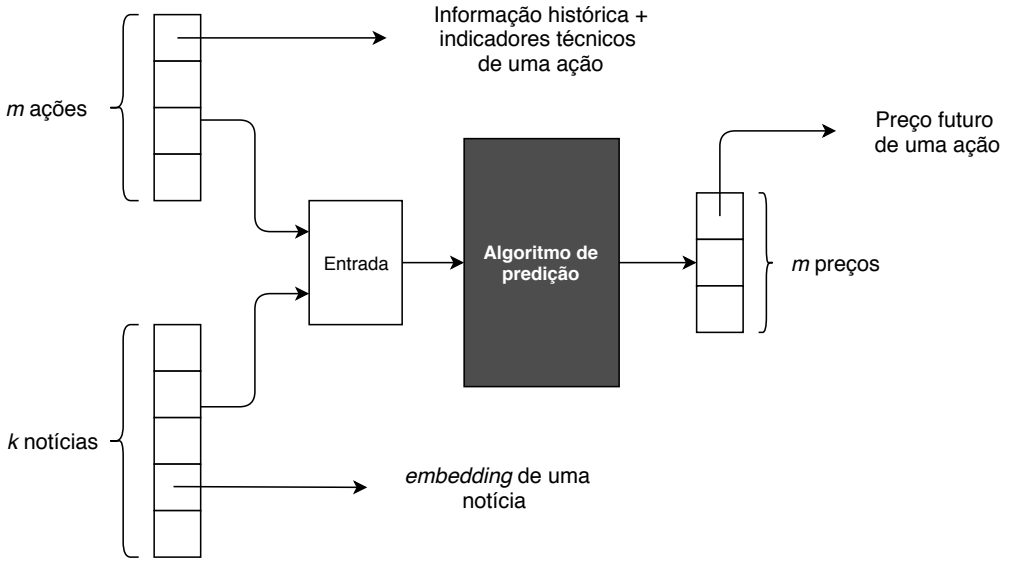

Figura 6.1: Pipeline de execução do modelo de predição.

e calcular o WCSS em cada iteração. Pelo gráfico do método do cotovelo apresentado na Figura 6.2, é perceptível que com aproximadamente 20 *clusters* não há um ganho significativo em termos de WCSS. Portanto, foi selecionado 10, 15 e 20 *clusters* para avaliar o método proposto neste trabalho.

São realizados experimentos com o modelo de predição para as clusterizações obtidas pelo K-Means com 10, 15 e 20 clusters. Esses resultados são descritos no Capítulo 6 e no Apêndice A é apresentada a clusterização usando o método do K-Means para essas quantidades de *clusters*.

No processo de clusterização usando o método DBSCAN, não há o problema de ter conhecimento do número de *clusters* apriori, porém, o método é sensível aos parâmetros ϵ e k , que representam o raio de alcance de cada ponto e o número mínimo de pontos, respectivamente. Em nosso conjunto de dados possuímos as séries de preço de apenas 51 ações, essa pouca quantidade de ações influencia em nossa clusterização, uma vez que o DBSCAN consegue encontrar bons *clusters* para grandes quantidades de pontos em um espaço. Deste modo, foi realizado uma busca exaustiva para encontrar um valor de ϵ aceitável mantendo sempre o valor de $k = 2$. É importante ressaltar que o parâmetro $k = 2$ foi escolhido pois também queremos identificar pares similares para avaliar o desempenho do modelo de predição. Após selecionar os conjuntos de ações similares (*clusters*) é criado um modelo para cada *cluster*. Dado um *cluster*, este modelo é capaz de realizar as previsões de preço futuro para cada ação presente nesse *cluster*. O modelo de predição é descrito detalhadamente na Seção 5.3.

São realizados experimentos com o classificador variando o valor de

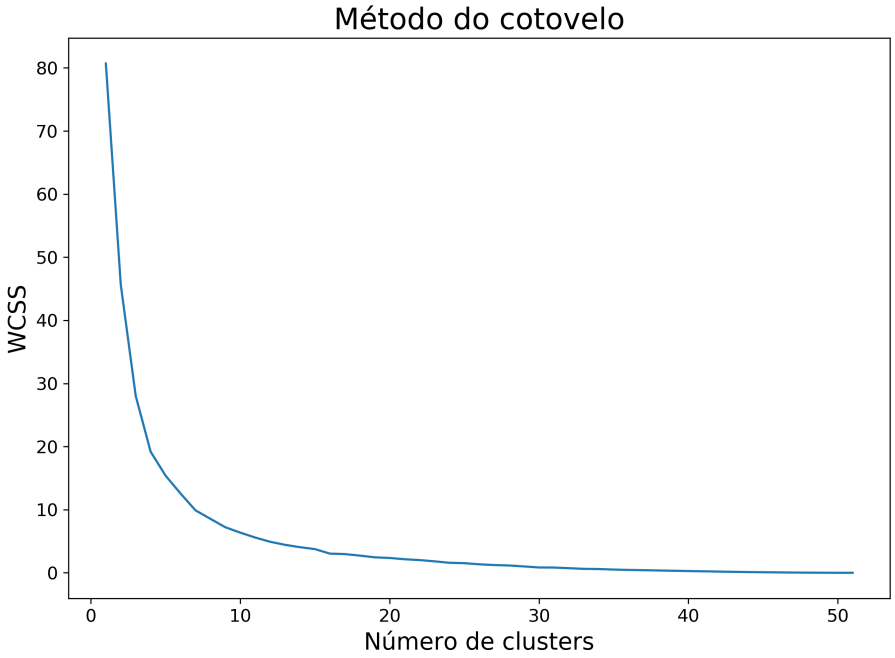

Figura 6.2: Método do cotovelo para estimar a quantidade de *clusters* do K-Means.

ϵ, para avaliar o modelo de predição em diferentes clusterizações. Esses experimentos do modelo de predição são executados a partir das clusterizações obtidas pelo DBSCAN para $\epsilon = 0.2$, $\epsilon = 0.3$ e $\epsilon = 0.35$, esses resultados são apresentados no Capítulo 6. No Apêndice B é mostrada a clusterização do método DBSCAN para estes hiperparâmetros.

6.3 Experimentos usando *news embeddings*

Neste trabalho, como mencionado anteriormente, utiliza os NE (*N embeddings*) para representar as principais notícias da plataforma do Google Trends. Para avaliar a influência dessas *features* textuais é usado o modelo de predição descrito na Figura 5.3 para prever as variações de preço em um conjunto de validação para a ação $PETR3$. Na Tabela 6.1 são mostrados os resultados das métricas de *precision*, *recall* e *f1-score*, para os modelos LSTM sem NE e com NE.

Tabela 6.1: Resultados do modelo com *features* textuais.

Método	**precision**	**recall**	**f1-score**
LSTM sem NE	0.66	0.66	0.66
LSTM com NE	0.73	0.71	0.73

De acordo com os resultados, o modelo que utiliza *word embeddings* melhora os resultados em 0.05 pontos de *recall* e 0.06 pontos de *precision* e *f1-score*. Portanto, é possível concluir que os *word embeddings* são promissores para a realização dessa tarefa e podem influenciar positivamente a performance do modelo de aprendizado.

6.4 Experimentos do preditor para conjuntos de ações

Para avaliar abordagem apresentada nesta pesquisa, são criados dois conjuntos de modelos de predição. No primeiro conjunto de modelos são criados modelos treinados para tratar ações isoladas: o Baseline e o LSTM. O modelo Baseline realiza previsões de movimentações de preço com base na distribuição das classes no conjunto de treino. Já o LSTM que utiliza redes LSTM com *features* de indicadores de análise técnica e os *news embeddings*.

No segundo conjunto são criados modelos treinados para um conjunto de ações similares: o KM-LSTM e o DB-LSTM. Os dois modelos utilizam uma etapa para identificar ações similares e utilizam uma rede LSTM com *features* de indicadores de análise técnica e os *news embeddings* para realizar as previsões de preço. A diferença entre os dois métodos é o método de clusterização usado para identificar os conjuntos de ações similares, o KM-LSTM usa o K-Means enquanto o DB-LSTM usa o DBSCAN.

6.4.1 Avaliação utilizando K-Means

Nas Tabelas C.1, C.2 e C.3, encontradas no Apêndice C, são apresentados os resultados do método KM-LSTM e LSTM para clusterizações obtidas pelo K-Means com 10, 15 e 20 *clusters*, respectivamente.

De acordo com a Tabela C.1 é possível perceber que para algumas ações, usar a informação extra do *cluster* melhora os resultados, como é o caso das ações MRFG3, ECOR3 e USIM5. Porém, em outras ações a informação extra adiciona ruído e faz com que o classificador perca performance, como é o caso das ações UGPA3, RADL3 e BRFS3. Além disso, percebe-se que para todas as ações do *cluster* 7, o método KM-LSTM é superior ao LSTM, isso nos sugere que as açoes desse *cluster* são um pouco correlacionadas, e consequentemente, são influenciadas pelos mesmos fatores ou variáveis. Seguindo o mesmo raciocínio percebe-se que para as ações do *cluster* 9, a informação extra não agrega valor para o modelo de predição, pois não gera bons resultados. Isso nos leva a acreditar que esse *cluster* contém ações que

não são similares, e consequentemente, que as ações desse *cluster* não sejam correlacionadas.

De acordo com a Tabela C.2 observa-se que em 42 ações a abordagem proposta alcança melhores resultados e vemos que para a maioria dos *clusters*, todas as suas ações melhoram os resultados quando comparadas a abordagem de tratar ações isoladas. Isso nos sugere que criar modelos para um conjunto de ações consideradas similares geram resultados promissores.

De acordo com a tabela C.3 percebe-se que a abordagem proposta é melhor em 45 ações e em 70% dos *clusters* suas ações possuem resultados melhores quando comparadas aos modelos de ações isoladas. É importante destacar que apesar de existir casos em que a abordagem proposta é inferior em aproximadamente 10%, como por exemplo a ação QUAL3, a abordagem proposta se mostrou superior aos modelos de ações isoladas. Dessa forma, vemos que os experimentos realizados com o KM-LSTM trazem indícios de que criar modelos para realizar previsões em conjuntos de ações consideradas similares nos trazem resultados promissores. Além disso, nota-se que quanto melhor o método para identificar ações similares melhor a performance do modelo de predição.

6.4.2 Avaliação utilizando DBSCAN

Nas Tabelas D.1, D.2 e D.3, encontradas no Apêndice D, são apresentados os resultados do método DB-LSTM e LSTM para clusterizações obtidas pelo DBSCAN com raio de alcance $\epsilon = 0.25, \epsilon = 0.3$ e $\epsilon = 0.35$, respectivamente.

De acordo com a Tabela D.1, o DB-LSTM é superior em todas as ações para todas as métricas avaliadas. Porém, neste experimento o DBSCAN consegue formar 8 *clusters* de 2 ações cada e 43 ações não foram "clusterizadas" pelo método, ou seja, o método identifica como ações similares apenas pares de ações que estão bem próximas no espaço.

Apesar do método encontrar poucos conjuntos de ações similares, pode-se concluir que as ações similares influenciam positivamente na fase predição, uma vez que o modelo de aprendizado é melhor quando usa as informações do *cluster* para uma ação específica.

Foi realizado um segundo experimento, em que neste o raio de alcance do método de clusterização é maior, o que resulta em uma maior quantidade de *clusters* formados. Nesse experimento, o DBSCAN forma 13 *clusters*, sendo 3 *clusters* composto por 3 ações, 1 *cluster* formado por 4 ações e os 8 *clusters* restantes contendo 2 ações cada.

Nesse experimento é encontrada uma maior variedade nos *clusters* gera-

dos e 29 ações foram "clusterizadas". De acordo com a Tabela D.2, percebe-se que o método DB-LSTM é superior em todas ações para todas as métricas avaliadas. Logo, esse experimento fortalece a hipótese de que usar ações similares para gerar *features*, trazem informações úteis para a predição de uma ação.

Com o propósito do processo de clusterização encontrar uma maior quantidade de conjuntos de ações similares foi conduzido um terceiro experimento em que o raio de alcance do DBSCAN é 0.35. Os resultados deste experimentos são apresentados na Tabela D.3.

De acordo com os resultados do terceiro experimento é possível observar que o método DB-LSTM é melhor em todas as ações, com exceção da ação CSNA3. Ao analisar o experimento 2 e 3 percebe-se que a inclusão da ação UGPA3 no *cluster* 3 resulta em uma piora nos resultados para todas as ações presentes neste *cluster*. Consequentemente, pode-se concluir que a ação UGPA3 não traz informações úteis para o modelo desse *cluster* e que a ação UGPA3 não é afetada pelos mesmos fatores e variáveis que as ações desse *cluster*. Além disso, observa-se que para os outros *clusters* o modelo DB-LSTM é superior em todas as métricas. Portanto, os resultados do experimento 3 fortalecem ainda mais a hipótese de que quando é possível identificar conjuntos de ações similares para treinar o modelo de predição obtém-se uma melhora nos resultados.

6.4.3 Comparação dos resultados

Para verificar a capacidade de aprendizado dos modelos de predições foi definido um modelo aleatório como *baseline*. Em que este modelo realiza previsões de alta ou baixa baseados na distribuição destas classes no conjunto de treino.

Nas Figuras 6.3, 6.4, 6.5 e 6.6 são apresentados os resultados para o conjunto de teste de cada modelo de predição para as métricas de *precision*, *recall*, *f1-score* e acurácia respectivamente. Nos resultados apresentados a seguir, definimos de ***baseline*** o modelo aleatório, **LSTM** o modelo criado para ações isoladas, **KM-LSTM** o modelo que utiliza o K-Means para realizar a clusterização das ações e **DB-LSTM** é o modelo que utiliza o DBSCAN como algoritmo de clusterização.

De acordo com os gráficos pode-se perceber que para todas as métricas, os modelos que utilizam uma etapa de *clusterização* (KMEANS e DBSCAN) para identificar ações similares possuem os valores de média e mediana maiores (colunas avg e med, respectivamente) e o desvio padrão menor (coluna std) em relação aos modelos de ações isoladas (LSTM). A Tabela 6.2 apresenta essas

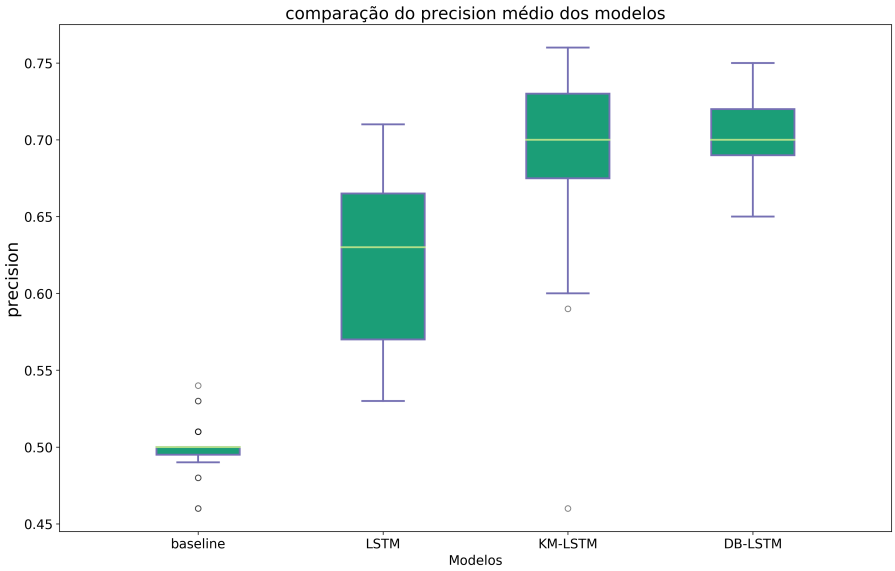

Figura 6.3: *Precision* dos modelos de predição para o conjunto de teste.

medidas para cada uma das métricas.

Tabela 6.2: Resultados do método DB-LSTM com raio de alcance $\epsilon = 0.35$, do KM-LSTM com 20 clusters e do LSTM.

método	precision			recall			f1-score			acurácia		
	avg	med	std	avg	med	std	avg	med	std	avg	med	std
Baseline	0.50	0.50	**0.014**	0.47	0.50	0.088	0.39	0.40	0.083	0.498	0.499	**0.015**
LSTM	0.62	0.63	0.055	0.61	0.61	0.052	0.61	0.61	0.052	0.618	0.628	0.055
KM-LTM	0.69	0.70	0.053	0.67	0.68	0.051	0.67	0.68	0.050	0.688	0.694	0.053
DB-LSTM	**0.70**	0.70	0.024	**0.69**	**0.69**	**0.029**	**0.68**	0.68	**0.029**	**0.699**	**0.701**	0.024

Para verificar se há uma diferença estaticamente significativa entre os modelos apresentados neste trabalho foi conduzido o teste de hipótese Kruskal-Wallis. Definimos o limite $\alpha = 0.05$ para realizar o teste de hipótese. A Tabela 6.3 apresenta os resultados do teste comparando os modelos LSTM, KM-LSTM e DB-LSTM.

De acordo com o teste de hipótese é possível perceber que há uma diferença estatisticamente significativa entre os modelos para ações consideradas similares (DB-LSTM e KM-LSTM) em relação ao modelo de ações isoladas (LSTM). A partir dos resultados mostrados nas Tabelas 6.2 e 6.3, pode-se concluir que os modelos criados para um conjunto de ações consideradas similares trazem uma melhora estatisticamente significativa em relação aos modelos criados para prever ações isoladas. Além disso, observa-se que ao comparar os modelos KM-LSTM e DB-LSTM, o DB-LSTM empiricamente alcança melhores resultados para as métricas avaliadas do que o KM-LSTM, mas não pode-se dizer que essa diferença é estatisticamente significativa, pois o *p-value* entre eles é maior do que o valor de α definido apriori.

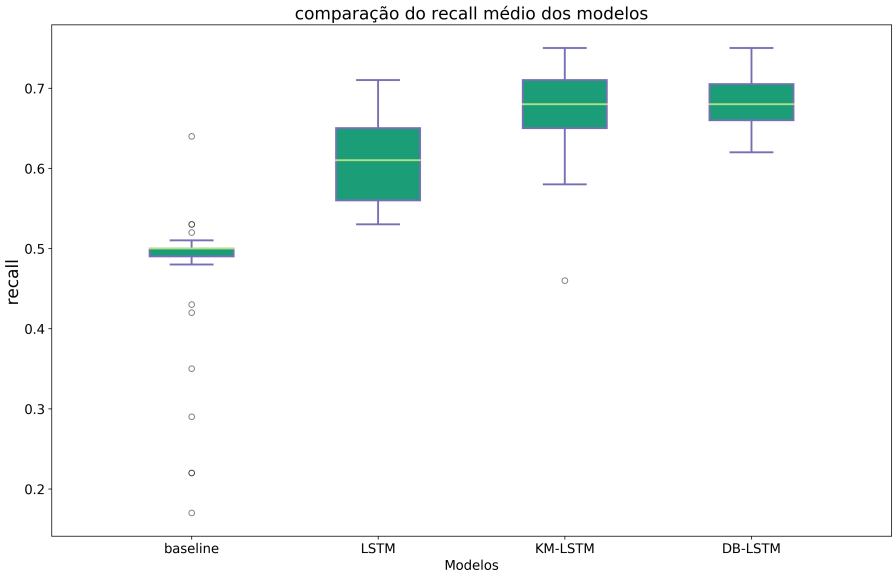

Figura 6.4: *Recall* dos modelos de predição para o conjunto de teste.

Tabela 6.3: Teste estatístico de hipótese Kruskal-Wallis para comparar os modelos de predição.

Par de modelos	Kruskal-Wallis H	*p-value*
LSTM, KM-LSTM	20.334035	$6.503344945 \cdot 10^{-6}$
LSTM, DB-LSTM	27.486151	$1.582234487 \cdot 10^{-7}$
KM-LSTM, DB-LSTM	0.265986	0.6060369279

Na Tabela 6.4 é apresentada uma comparação dos métodos KM-LSTM e DB-LSTM com os modelos criados em outros trabalhos encontrados na literatura. É importante ressaltar que os conjuntos de dados utilizado entre os trabalhos são diferentes e que os outros trabalhos desenvolveram modelos especialmente para uma ação.

No trabalho de (43) o modelo foi avaliado em 100 ações pertencentes à Bolsa de Valores de Xangai e a Bolsa de Valores de Shenzhen. Em (37) os autores avaliam o modelo de predição nas ações BOVA11, BBDC4, ITUB4, CIEL3 e PETR4 do mercado brasileiro. Em (11) o modelo desenvolvido é avaliado nas ações PETR4, ABEV3 e ITSA4 do mercado brasileiro. Já nos métodos DB-LSTM e KM-LSTM são validados em um conjunto de 51 ações do mercado brasileiro.

De acordo com os resultados encontrados, é possível perceber que os modelos criados para um conjunto de ações consideradas similares por um procedimento de clusterização, alcançam resultados tão bons quanto aos modelos desenvolvidos especialmente para uma ação.

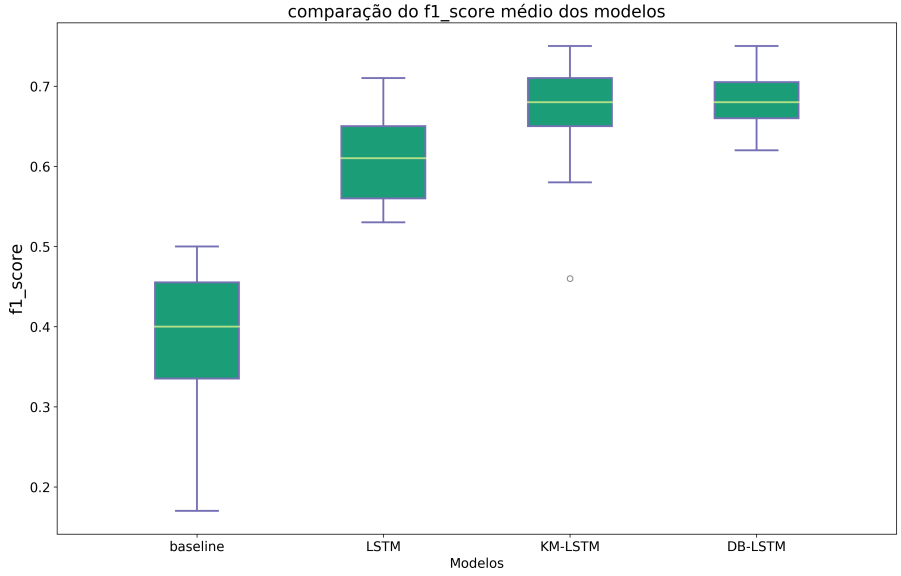

Figura 6.5: *F1-score* dos modelos de predição para o conjunto de teste.

Tabela 6.4: Comparação dos métodos desenvolvidos nesse trabalho com outros trabalhos.

Trabalho	Mercado	Acurácia média	n ações	n modelos
Sun (43)	China	0.713	100	100
Nelson (37)	Brasil	0.543	5	5
De Melo (11)	Brasil	0.690	3	3
DB-LSTM	Brasil	0.702	51	25
KM-LSTM	Brasil	0.689	51	20

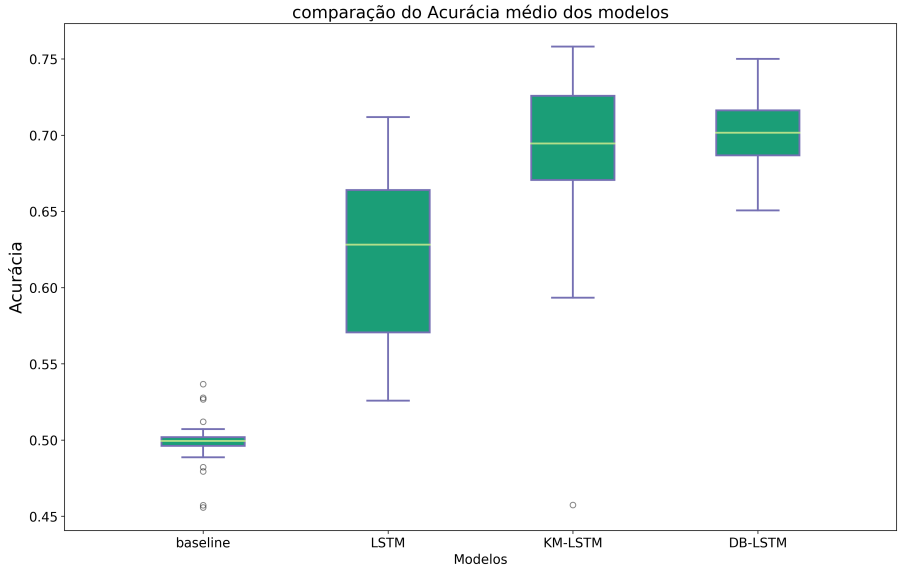

Figura 6.6: Acurácia dos modelos de predição para o conjunto de teste.

7
Considerações finais

Neste trabalho foi realizado um estudo para a criação de um modelo de aprendizado capaz de predizer as variações de preços para um conjunto de ações similares. Para identificar ações similares foi apresentado um método para criar representações vetoriais de uma ação baseado em dados textuais que a descrevem.

Uma vez que se definiu os vetores representativos das ações do mercado financeiro, algoritmos de clusterização foram aplicados para identificar e agrupar as ações em conjuntos similares (*clusters* de ações), que forneceram a base para o modelo de aprendizado.

O modelo de aprendizado é uma arquitetura baseada em redes LSTM, que foi aplicado na predição das variações de preços de um conjunto de 51 ações do mercado acionário brasileiro que compõem o índice Ibovespa. Foram aplicados os métodos KM-LSTM e DB-LSTM para identificar os clusters sobre os quais foram criados modelos de aprendizado. As predições foram avaliadas a partir das métricas de *precision*, *recall*, *f1-score* e acurácia.

De acordo com os resultados obtidos nesta pesquisa é possível perceber que os dois métodos que usam a informação de ações consideradas similares, DB-LSTM e KM-LSTM, são superiores ao modelo treinado com ações isoladas, e que o método DB-LSTM, em média, é 8% melhor em termos de *precision* e *recall* e 7% melhor em acurácia e *f1-score* quando comparado ao modelo treinado com ações isoladas. Além disso, o método DB-LSTM mostrou-se mais consistente do que o KM-LSTM, pois seus resultados foram superiores em 97% das ações quando comparados ao modelo de ações isoladas. Isso sugere que o método DB-LSTM é capaz de realizar previsões mais precisas do que o método KM-LSTM. Um dos motivos é o fato de o K-Means ingenuamente considerar que cada ação pertencerá a algum *cluster*, o que resulta em *clusters* que podem conter ações que não são tão similares. Já no método DB-LSTM, o DBSCAN tem a garantia de que, se as ações estão presentes no mesmo *cluster*, então elas tendem a ser similares, pois a distância entre elas é no máximo ϵ, valor do raio de alcance definido apriori pelo método DBSCAN.

Com estes resultados, há indícios de que é promissor identificar ações similares para prever o preço de uma ação, uma vez que as ações similares

acrescentam mais informações para o modelo de aprendizado. Ademais, comparados os resultados obtidos pelo estudo aqui realizado com trabalhos que preveem ações específicas de outros mercados financeiros, mesmo utilizando um modelo para cada *cluster*, foi possível obter resultados comparáveis a modelos criados especialmente para uma ação.

Pode-se, portanto, apontar como principais contribuições deste trabalho a proposição de:

1. uma nova abordagem para criar um modelo de predição para um conjunto de ações consideradas similares no mercado financeiro.

2. uma abordagem para a criação de *stock embeddings* para representar as ações do mercado acionário.

Alguns possíveis trabalhos e caminhos que podem ser seguidos no futuro são:

1. estudar diferentes métodos para criar os documentos que descrevem uma ação, como por exemplo utilizar documentos corporativos e informações de mercado das empresas. Isso possibilita a criação de *stock embeddings* mais refinados.

2. utilizar diferentes metodologias para realizar o processo de clusterização com o propósito de encontrar conjuntos de ações similares, uma vez que identificar a similaridade entre ativos financeiros é fundamental para o modelo de predição proposto.

3. aplicar o modelo proposto para estimar a expectativa de retorno de diferentes ativos financeiros, tais como: fundos de renda fixa, fundos multimercado, fundos cambial, entre outros. Dessa forma, saímos um pouco do domínio do mercado de ações e aplicamos o método para mercado financeiro de modo geral, isso possibilita obter uma expectativa de retorno melhor para diferentes ativos financeiros e, consequentemente, melhorar os resultados de metodologias utilizadas para diferentes problemas do mercado financeiro, como o problema de seleção de portfólio.

Referências bibliográficas

[1] ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G. S.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; GOODFELLOW, I.; HARP, A.; IRVING, G.; ISARD, M.; JIA, Y.; JOZEFOWICZ, R.; KAISER, L.; KUDLUR, M.; LEVENBERG, J.; MANÉ, D.; MONGA, R.; MOORE, S.; MURRAY, D.; OLAH, C.; SCHUSTER, M.; SHLENS, J.; STEINER, B.; SUTSKEVER, I.; TALWAR, K.; TUCKER, P.; VANHOUCKE, V.; VASUDEVAN, V.; VIÉGAS, F.; VINYALS, O.; WARDEN, P.; WATTENBERG, M.; WICKE, M.; YU, Y. ; ZHENG, X.. **TensorFlow: Large-scale machine learning on heterogeneous systems**, 2015. Software available from tensorflow.org.

[2] ABHISHEK, K.; KHAIRWA, A.; PRATAP, T. ; PRAKASH, S.. **A stock market prediction model using artificial neural network**. In: COMPUTING COMMUNICATION & NETWORKING TECHNOLOGIES (ICCCNT), 2012 THIRD INTERNATIONAL CONFERENCE ON, p. 1–5. IEEE, 2012.

[3] ALTHELAYA, K. A.; EL-ALFY, E.-S. M. ; MOHAMMED, S.. **Evaluation of bidirectional lstm for short-and long-term stock market prediction**. In: 2018 9TH INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION SYSTEMS (ICICS), p. 151–156. IEEE, 2018.

[4] APPEL, G.; HITSCHLER, W. F.. **Stock market trading systems**. Irwin Professional Pub, 1980.

[5] ATTIGERI, G. V.; MM, M. P.; PAI, R. M. ; NAYAK, A.. **Stock market prediction: A big data approach**. In: TENCON 2015-2015 IEEE REGION 10 CONFERENCE, p. 1–5. IEEE, 2015.

[6] BENGIO, Y.; SIMARD, P.; FRASCONI, P. ; OTHERS. **Learning long-term dependencies with gradient descent is difficult**. IEEE transactions on neural networks, 5(2):157–166, 1994.

[7] BOONPENG, S.; JEATRAKUL, P.. **Enhance the performance of neural networks for stock market prediction: An analytical study**.

In: DIGITAL INFORMATION MANAGEMENT (ICDIM), 2014 NINTH INTERNATIONAL CONFERENCE ON, p. 1–6. IEEE, 2014.

[8] CHANDE, T. S.; KROLL, S.. **The new technical trader: boost your profit by plugging into the latest indicators**. John Wiley & Sons Inc, 1994.

[9] CHENG, C.-H.; CHEN, T.-L. ; WEI, L.-Y.. **A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting**. Information Sciences, 180(9):1610–1629, 2010.

[10] DE FARIA, E.; ALBUQUERQUE, M. P.; GONZALEZ, J.; CAVALCANTE, J. ; ALBUQUERQUE, M. P.. **Predicting the brazilian stock market through neural networks and adaptive exponential smoothing methods**. Expert Systems with Applications, 36(10):12506–12509, 2009.

[11] DE MELO, J. P. F.. **Predicting trends in the stock market**. 2018.

[12] DE OLIVEIRA, F. A.; NOBRE, C. N. ; ZARATE, L. E.. **Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index–case study of petr4, petrobras, brazil**. Expert Systems with Applications, 40(18):7596–7606, 2013.

[13] ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. ; OTHERS. **A density-based algorithm for discovering clusters in large spatial databases with noise.** In: KDD, volumen 96, p. 226–231, 1996.

[14] FAMA, E. F.. **The behavior of stock-market prices**. The journal of Business, 38(1):34–105, 1965.

[15] GRAVE, E.; BOJANOWSKI, P.; GUPTA, P.; JOULIN, A. ; MIKOLOV, T.. **Learning word vectors for 157 languages**. In: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC 2018), 2018.

[17] HAFEZI, R.; SHAHRABI, J. ; HADAVANDI, E.. **A bat-neural network multi-agent system (bnnmas) for stock price prediction: Case study of dax stock price**. Applied Soft Computing, 29:196–210, 2015.

[18] HOCHREITER, S.; SCHMIDHUBER, J.. **Long short-term memory**. Neural computation, 9(8):1735–1780, 1997.

[19] HOPFIELD, J. J.. **Neural networks and physical systems with emergent collective computational abilities**. Proceedings of the national academy of sciences, 79(8):2554–2558, 1982.

[20] HU, H.; TANG, L.; ZHANG, S. ; WANG, H.. **Predicting the direction of stock markets using optimized neural networks with google trends**. Neurocomputing, 285:188–195, 2018.

[21] HUANG, C.-J.; YANG, D.-X. ; CHUANG, Y.-T.. **Application of wrapper approach and composite classifier to the stock trend prediction**. Expert Systems with Applications, 34(4):2870–2878, 2008.

[22] IACOMIN, R.. **Stock market prediction**. In: SYSTEM THEORY, CONTROL AND COMPUTING (ICSTCC), 2015 19TH INTERNATIONAL CONFERENCE ON, p. 200–205. IEEE, 2015.

[23] GOODFELLOW, I.; BENGIO, Y. ; COURVILLE, A.. **Deep Learning**. MIT Press, 2016. `http://www.deeplearningbook.org`.

[24] KIMOTO, T.; ASAKAWA, K.; YODA, M. ; TAKEOKA, M.. **Stock market prediction system with modular neural networks**. In: 1990 IJCNN INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, p. 1–6. IEEE, 1990.

[25] KINGMA, D. P.; BA, J.. **Adam: A method for stochastic optimization**. arXiv preprint arXiv:1412.6980, 2014.

[26] LE, Q.; MIKOLOV, T.. **Distributed representations of sentences and documents**. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 1188–1196, 2014.

[27] LECUN, Y.; BENGIO, Y. ; HINTON, G.. **Deep learning**. nature, 521(7553):436, 2015.

[28] LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W. ; JACKEL, L. D.. **Backpropagation applied to handwritten zip code recognition**. Neural computation, 1(4):541–551, 1989.

[29] LEE, M.-C.. **Using support vector machine with a hybrid feature selection method to the stock trend prediction**. Expert Systems with Applications, 36(8):10896–10904, 2009.

[30] LI, Q.; ZHOU, B. ; LIU, Q.. **Can twitter posts predict stock behavior?: A study of stock market with twitter social emotion**. In: CLOUD COMPUTING AND BIG DATA ANALYSIS (ICCCBDA), 2016 IEEE INTERNATIONAL CONFERENCE ON, p. 359–364. IEEE, 2016.

[31] LILLEBERG, J.; ZHU, Y. ; ZHANG, Y.. **Support vector machines and word2vec for text classification with semantic features**. In: COGNITIVE INFORMATICS & COGNITIVE COMPUTING (ICCI* CC), 2015 IEEE 14TH INTERNATIONAL CONFERENCE ON, p. 136–140. IEEE, 2015.

[32] LING, W.; DYER, C.; BLACK, A. W. ; TRANCOSO, I.. **Two/too simple adaptations of word2vec for syntax problems**. In: PROCEEDINGS OF THE 2015 CONFERENCE OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: HUMAN LANGUAGE TECHNOLOGIES, p. 1299–1304, 2015.

[33] MACQUEEN, J.; OTHERS. **Some methods for classification and analysis of multivariate observations**. In: PROCEEDINGS OF THE FIFTH BERKELEY SYMPOSIUM ON MATHEMATICAL STATISTICS AND PROBABILITY, volumen 1, p. 281–297. Oakland, CA, USA, 4967.

[34] MALKIEL, B. G.. **A random walk down Wall Street: the time-tested strategy for successful investing**. WW Norton & Company, 2007.

[35] MIKOLOV, T.; CHEN, K.; CORRADO, G. ; DEAN, J.. **Efficient estimation of word representations in vector space**. arXiv preprint arXiv:1301.3781, 2013.

[36] MINGYUE, Q.; CHENG, L. ; YU, S.. **Application of the artifical neural network in predicting the direction of stock market index**. In: 2016 10TH INTERNATIONAL CONFERENCE ON COMPLEX, INTELLIGENT, AND SOFTWARE INTENSIVE SYSTEMS (CISIS), p. 219–223. IEEE, 2016.

[37] NELSON, D. M.; PEREIRA, A. C. ; DE OLIVEIRA, R. A.. **Stock market's price movement prediction with lstm neural networks**. In: 2017 INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN), p. 1419–1426. IEEE, 2017.

[38] OLAH, C.. **Understanding lstm networks**. 2015.

[39] PENNINGTON, J.; SOCHER, R. ; MANNING, C.. **Glove: Global vectors for word representation**. In: PROCEEDINGS OF THE 2014 CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING (EMNLP), p. 1532–1543, 2014.

[40] RASEL, R. I.; SULTANA, N. ; HASAN, N.. **Financial instability analysis using ann and feature selection technique: application to stock**

market price prediction. In: INNOVATIONS IN SCIENCE, ENGINEERING AND TECHNOLOGY (ICISET), INTERNATIONAL CONFERENCE ON, p. 1–4. IEEE, 2016.

[41] SABLE, S.; PORWAL, A. ; SINGH, U.. **Stock price prediction using genetic algorithms and evolution strategies**. In: ELECTRONICS, COMMUNICATION AND AEROSPACE TECHNOLOGY (ICECA), 2017 INTERNATIONAL CONFERENCE OF, volumen 2, p. 549–553. IEEE, 2017.

[42] SIEW, H. L.; NORDIN, M. J.. **Regression techniques for the prediction of stock price trend**. Statistics in Science, Business, and Engineering (ICSSBE) Langkawi Universiti Kuala Lumpur, p. 1–5, 2012.

[43] SUN, T.; WANG, J.; ZHANG, P.; CAO, Y.; LIU, B. ; WANG, D.. **Predicting stock price returns using microblog sentiment for chinese stock market**. In: BIG DATA COMPUTING AND COMMUNICATIONS (BIGCOM), 2017 3RD INTERNATIONAL CONFERENCE ON, p. 87–96. IEEE, 2017.

[44] TAN, T. Z.; QUEK, C. ; NG, G. S.. **Brain-inspired genetic complementary learning for stock market prediction**. In: EVOLUTIONARY COMPUTATION, 2005. THE 2005 IEEE CONGRESS ON, volumen 3, p. 2653–2660. IEEE, 2005.

[45] WANG, J.-H.; LEU, J.-Y.. **Stock market trend prediction using arima-based neural networks**. In: IEEE INT. CONF. NEURAL NETWORKS, volumen 4, p. 2160–2165, 1996.

[46] WEI, L.-Y.. **A hybrid model based on anfis and adaptive expectation genetic algorithm to forecast taiex**. Economic Modelling, 33:893–899, 2013.

[47] WENG, B.; LU, L.; WANG, X.; MEGAHED, F. M. ; MARTINEZ, W.. **Predicting short-term stock prices using ensemble methods and online data sources**. Expert Systems with Applications, 2018.

[48] WILDER, J. W.. **New concepts in technical trading systems**. Trend Research, 1978.

[49] YANG, B.; GONG, Z.-J. ; YANG, W.. **Stock market index prediction using deep neural network ensemble**. In: CONTROL CONFERENCE (CCC), 2017 36TH CHINESE, p. 3882–3887. IEEE, 2017.

[50] ZHANG, G.; PATUWO, B. E. ; HU, M. Y.. **Forecasting with artificial neural networks:: The state of the art**. International journal of forecasting, 14(1):35–62, 1998.

[51] ZHOU, F.; ZHOU, H.-M.; YANG, Z. ; YANG, L.. **Emd2fnn: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction**. Expert Systems with Applications, 115:136–151, 2019.

[52] **B3 - brasil bolsa balcao**. `http://www.b3.com.br/pt_br/`. Accessado em: 17/10/2019.

A
Clusterização do método K-Means

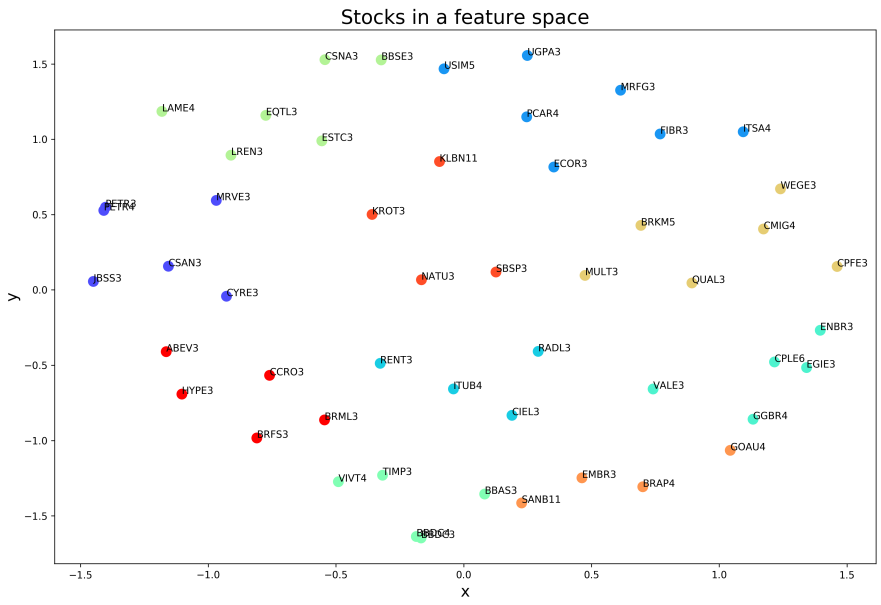

Figura A.1: K-Means aplicado para 10 *clusters*

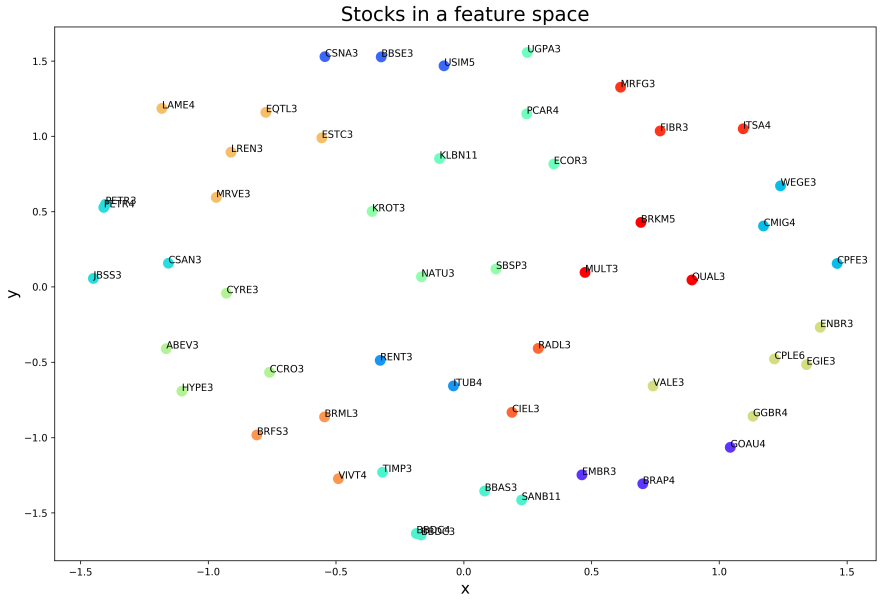

Figura A.2: K-Means aplicado para 15 *clusters*

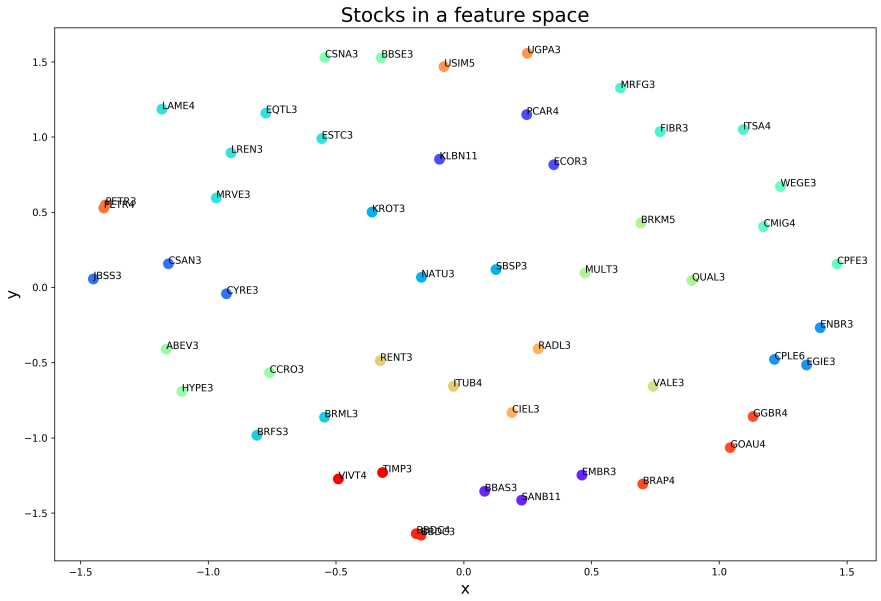

Figura A.3: K-Means aplicado para 20 *clusters*

B
Clusterização do método DBSCAN

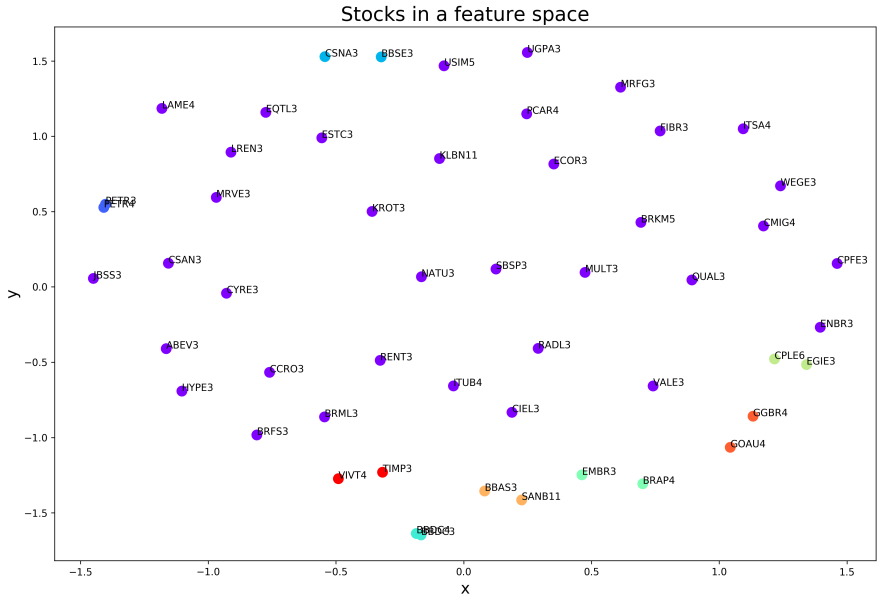

Figura B.1: DBSCAN aplicado para $\epsilon = 0.25$ geramos 9 *clusters*

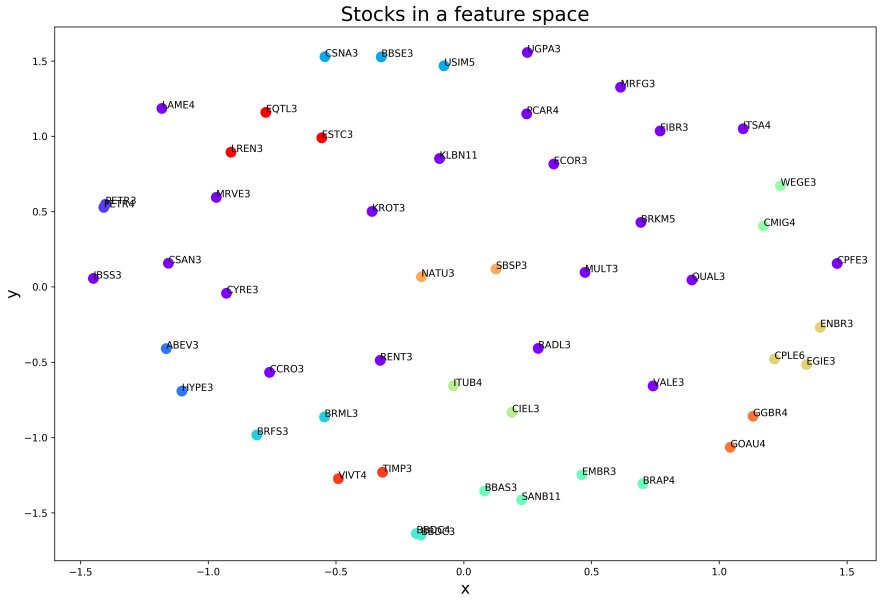

Figura B.2: DBSCAN aplicado para $\epsilon = 0.3$ geramos 13 *clusters*

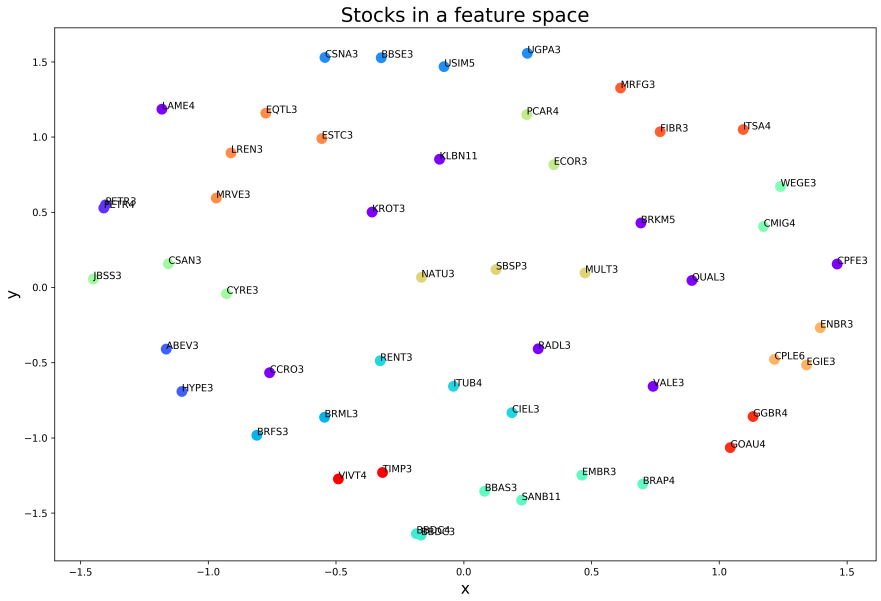

Figura B.3: DBSCAN aplicado para $\epsilon = 0.25$ geramos 16 *clusters*

C
Resultados do modelo KM-LSTM

Tabela C.1: Resultados do método KM-LSTM com 10 *clusters* e do LSTM

		recall		**precision**		**f1-score**		**acurácia**	
nº cluster	ação	KM-LSTM	LSTM	KM-LSTM	LSTM	KM-LSTM	LSTM	KM-LSTM	LSTM
0	JBSS3	0,6	**0,61**	0,6	**0,61**	0,6	**0,61**	0,63	**0,64**
0	PETR3	0,64	0,64	0,64	0,64	0,64	0,64	**0,65**	0,64
0	PETR4	**0,66**	0,61	**0,66**	0,61	**0,66**	0,61	**0,67**	0,61
0	CYRE3	0,56	**0,61**	0,56	**0,61**	0,56	**0,61**	0,59	**0,65**
0	MRVE3	0,54	**0,65**	0,54	**0,65**	0,54	**0,65**	0,55	**0,68**
0	CSAN3	0,57	**0,62**	0,57	**0,62**	0,57	**0,62**	0,57	**0,63**
1	FIBR3	0,55	**0,61**	0,55	**0,61**	0,55	**0,61**	0,55	**0,61**
1	MRFG3	0,63	**0,67**	0,63	**0,67**	0,63	**0,67**	0,67	**0,73**
1	ECOR3	0,61	**0,65**	0,61	**0,65**	0,61	**0,65**	0,64	**0,69**
1	USIM5	0,62	**0,65**	0,62	**0,65**	0,62	**0,65**	0,67	0,71
1	ITSA4	0,53	**0,63**	0,53	**0,63**	0,53	**0,63**	0,53	**0,67**
1	UGPA3	**0,71**	0,66	**0,71**	0,66	**0,71**	0,66	**0,71**	0,66
1	PCAR4	0,68	**0,7**	0,68	**0,7**	0,68	**0,7**	0,68	**0,7**
2	ITUB4	0,56	**0,65**	0,56	**0,65**	0,56	**0,65**	0,56	**0,66**
2	RADL3	**0,71**	0,64	**0,71**	0,64	**0,71**	0,64	**0,71**	0,64
2	RENT3	0,69	**0,72**	0,69	**0,72**	0,69	**0,72**	0,7	**0,73**
2	CIEL3	0,62	**0,66**	0,62	**0,66**	0,62	**0,66**	0,62	**0,66**
3	VALE3	0,54	**0,63**	0,54	**0,63**	0,54	**0,63**	0,54	**0,64**
3	ENBR3	0,57	**0,66**	0,57	**0,66**	0,57	**0,66**	0,58	**0,69**
3	CPLE6	0,66	**0,68**	0,66	**0,68**	0,66	**0,68**	0,67	**0,69**
3	GGBR4	0,55	**0,65**	0,55	**0,65**	0,55	**0,65**	0,55	**0,67**
3	EGIE3	**0,65**	0,61	**0,65**	0,61	**0,65**	0,61	**0,66**	0,61
4	BBDC4	0,6	**0,69**	0,6	**0,69**	0,6	**0,69**	0,6	**0,69**
4	TIMP3	0,57	**0,68**	0,57	**0,68**	0,57	**0,68**	0,59	**0,74**
4	VIVT4	**0,69**	0,67	**0,69**	0,67	**0,69**	0,67	**0,7**	0,68
4	BBDC3	0,6	**0,74**	0,6	**0,74**	0,6	**0,74**	0,6	**0,75**

4	BBAS3	0,53	**0,69**	0,53	**0,69**	0,53	**0,69**	0,53	**0,7**
5	LAME4	0,62	**0,67**	0,62	**0,67**	0,62	**0,67**	0,61	**0,69**
5	ESTC3	0,63	**0,65**	0,63	**0,65**	0,63	**0,65**	0,64	**0,67**
5	LREN3	0,57	**0,59**	0,57	**0,59**	0,57	**0,59**	0,57	**0,6**
5	CSNA3	**0,65**	0,64	**0,65**	0,64	**0,65**	0,64	0,66	0,66
5	BBSE3	0,57	**0,63**	0,57	**0,63**	0,57	**0,63**	0,58	**0,63**
5	EQTL3	0,56	**0,7**	0,56	**0,7**	0,56	**0,7**	0,56	**0,71**
6	MULT3	0,71	0,71	0,71	0,71	0,71	0,71	0,71	**0,72**
6	CPFE3	**0,61**	0,58	**0,61**	0,58	**0,61**	0,58	**0,63**	0,6
6	CMIG4	0,65	**0,66**	0,65	**0,66**	0,65	**0,66**	0,69	**0,7**
6	QUAL3	**0,72**	0,65	**0,72**	0,65	**0,72**	0,65	**0,74**	0,66
6	BRKM5	0,64	**0,69**	0,64	**0,69**	0,64	**0,69**	0,65	**0,69**
6	WEGE3	0,61	**0,65**	0,61	**0,65**	0,61	**0,65**	0,63	**0,67**
7	EMBR3	0,61	**0,67**	0,61	**0,67**	0,61	**0,67**	0,63	**0,7**
7	GOAU4	0,53	**0,6**	0,53	**0,6**	0,53	**0,6**	0,54	**0,63**
7	SANB11	0,53	**0,68**	0,53	**0,68**	0,53	**0,68**	0,53	**0,7**
7	BRAP4	0,56	**0,68**	0,56	**0,68**	0,56	**0,68**	0,56	**0,69**
8	KLBN11	0,57	**0,69**	0,57	**0,69**	0,57	**0,69**	0,58	**0,71**
8	SBSP3	0,64	**0,68**	0,64	**0,68**	0,64	**0,68**	0,65	**0,69**
8	NATU3	**0,65**	0,64	**0,65**	0,64	**0,65**	0,64	**0,66**	0,65
8	KROT3	0,52	**0,65**	0,52	**0,65**	0,52	**0,65**	0,53	**0,67**
9	BRFS3	**0,67**	0,58	**0,67**	0,58	**0,67**	0,58	**0,68**	0,58
9	HYPE3	**0,62**	0,6	**0,62**	0,6	**0,62**	0,6	**0,63**	0,61
9	BRML3	**0,56**	0,54	**0,56**	0,54	**0,56**	0,54	**0,57**	0,56
9	ABEV3	**0,61**	0,6	**0,61**	0,6	**0,61**	0,6	**0,63**	0,62
9	CCRO3	0,61	**0,62**	0,61	**0,62**	0,61	**0,62**	**0,63**	0,62

Tabela C.2: Resultados do método KM-LSTM com 15 *clusters* e do LSTM

		recall		**precision**		**f1-score**		**acurácia**	
nº cluster	ação	KM-LSTM	LSTM	KM-LSTM	LSTM	KM-LSTM	LSTM	KM-LSTM	LSTM
0	EMBR3	0.63	**0.7**	0.61	**0.67**	0.6	**0.66**	0.63	**0.7**
0	GOAU4	0.54	**0.62**	0.53	**0.6**	0.5	**0.57**	0.54	**0.62**
0	BRAP4	0.56	**0.68**	0.56	**0.66**	0.54	**0.65**	0.56	**0.68**
1	USIM5	0.67	**0.68**	0.62	**0.63**	0.6	**0.6**	0.67	**0.68**
1	CSNA3	0.66	**0.69**	0.65	**0.67**	0.63	**0.66**	0.66	**0.69**
1	BBSE3	0.58	**0.75**	0.57	**0.73**	0.57	**0.73**	0.58	**0.75**

2	ITUB4	0.56	**0.66**	0.56	**0.65**	0.55	**0.65**	0.56	**0.66**
2	RENT3	**0.7**	0.67	**0.69**	0.67	**0.69**	0.66	**0.7**	0.67
3	CMIG4	0.69	**0.74**	0.65	**0.68**	0.62	**0.66**	0.69	**0.74**
3	CPFE3	0.63	**0.66**	0.61	**0.63**	0.59	**0.62**	0.63	0.66
3	WEGE3	0.63	**0.76**	0.61	**0.73**	0.6	**0.72**	0.63	**0.76**
4	PETR4	0.67	**0.72**	0.66	**0.7**	0.65	**0.69**	0.67	**0.72**
4	JBSS3	0.63	**0.66**	0.6	**0.63**	0.58	**0.6**	0.63	**0.66**
4	PETR3	0.65	**0.72**	0.64	**0.7**	0.64	**0.7**	0.65	**0.72**
4	CSAN3	0.57	**0.68**	0.57	**0.67**	0.57	**0.67**	0.57	**0.68**
5	BBDC3	0.6	**0.71**	0.6	**0.69**	0.59	**0.69**	0.6	**0.71**
5	BBAS3	0.53	**0.65**	0.53	**0.64**	0.53	**0.64**	0.53	**0.65**
5	SANB11	0.53	**0.67**	0.53	**0.66**	0.52	**0.65**	0.53	**0.67**
5	TIMP3	0.59	**0.67**	0.57	**0.62**	0.54	**0.6**	0.59	**0.67**
5	BBDC4	0.6	**0.69**	0.6	**0.69**	0.59	**0.68**	0.6	**0.69**
6	ECOR3	0.64	**0.71**	0.61	**0.66**	0.59	**0.64**	0.64	**0.71**
6	PCAR4	0.68	**0.75**	0.68	**0.74**	0.67	**0.74**	0.68	**0.75**
6	UGPA3	0.71	**0.74**	0.71	**0.73**	0.7	**0.73**	0.71	**0.74**
6	KLBN11	0.58	**0.73**	0.57	**0.69**	0.55	**0.68**	0.58	**0.73**
7	SBSP3	0.65	**0.73**	0.64	**0.72**	0.63	**0.71**	0.65	**0.73**
7	NATU3	0.66	**0.69**	0.65	**0.69**	0.65	**0.68**	0.66	**0.69**
7	KROT3	0.53	**0.72**	0.52	**0.71**	0.51	**0.7**	0.53	**0.72**
8	CCRO3	0.63	**0.65**	0.61	**0.64**	0.6	**0.62**	0.63	**0.65**
8	ABEV3	0.63	**0.64**	0.61	**0.64**	0.59	**0.62**	0.63	**0.64**
8	CYRE3	0.59	0.59	0.56	**0.59**	0.54	**0.55**	0.59	0.59
8	HYPE3	**0.63**	0.6	**0.62**	0.58	**0.61**	0.57	**0.63**	0.6
9	GGBR4	0.55	**0.67**	0.55	**0.65**	0.53	**0.64**	0.55	**0.67**
9	EGIE3	**0.66**	0.61	**0.65**	0.61	**0.64**	0.6	**0.66**	0.61
9	CPLE6	0.67	**0.69**	0.66	**0.68**	0.66	**0.68**	0.67	**0.69**
9	ENBR3	0.58	**0.69**	0.57	**0.66**	0.55	**0.65**	0.58	**0.69**
9	VALE3	0.54	**0.64**	0.54	**0.63**	0.53	**0.63**	0.54	**0.64**
10	MRVE3	0.55	**0.67**	0.54	**0.64**	0.53	**0.63**	0.55	**0.67**
10	ESTC3	**0.64**	0.59	**0.63**	0.58	**0.62**	0.57	**0.64**	0.59
10	LAME4	0.61	**0.62**	**0.62**	0.61	0.6	0.6	0.61	**0.62**
10	EQTL3	0.56	**0.73**	0.56	**0.72**	0.56	**0.72**	0.56	**0.73**
10	LREN3	0.57	**0.63**	0.57	**0.63**	0.56	**0.62**	0.57	**0.63**
11	BRML3	0.57	**0.58**	0.56	**0.57**	0.53	**0.54**	0.57	**0.58**
11	BRFS3	**0.68**	0.65	**0.67**	0.64	**0.67**	0.64	**0.68**	0.65
11	VIVT4	**0.7**	0.64	**0.69**	0.64	**0.68**	0.63	**0.7**	0.64
12	CIEL3	0.62	**0.73**	0.62	**0.72**	0.61	**0.71**	0.62	**0.73**

12	RADL3	**0.71**	0.7	**0.71**	0.7	**0.7**	0.69	**0.71**	0.7
13	ITSA4	0.53	**0.7**	0.53	**0.65**	0.49	**0.62**	0.53	**0.7**
13	MRFG3	0.67	**0.69**	0.63	**0.64**	0.59	**0.61**	0.67	**0.69**
13	FIBR3	0.55	**0.64**	0.55	**0.63**	0.54	**0.62**	0.55	**0.64**
14	QUAL3	**0.74**	0.64	**0.72**	0.62	**0.71**	0.62	**0.74**	0.64
14	MULT3	0.71	**0.72**	0.71	**0.72**	0.71	**0.72**	0.71	**0.72**
14	BRKM5	0.65	**0.72**	0.64	**0.71**	0.64	**0.7**	0.65	**0.72**

Tabela C.3: Resultados do método KM-LSTM com 20 *clusters* e do LSTM

		recall		**precision**		**f1-score**		**acurácia**	
nº cluster	ação	KM-LSTM	LSTM	KM-LSTM	LSTM	KM-LSTM	LSTM	KM-LSTM	LSTM
0	EMBR3	0.63	**0.72**	0.61	**0.69**	0.6	**0.67**	0.63	**0.72**
0	BBAS3	0.53	**0.66**	0.53	**0.65**	0.53	**0.65**	0.53	**0.66**
0	SANB11	0.53	**0.65**	0.53	**0.64**	0.52	**0.63**	0.53	**0.65**
1	KLBN11	0.58	**0.72**	0.57	**0.69**	0.55	**0.68**	0.58	**0.72**
1	PCAR4	0.68	**0.75**	0.67	**0.75**	0.67	**0.75**	0.68	**0.75**
1	ECOR3	0.64	**0.69**	0.61	**0.65**	0.59	**0.63**	0.64	**0.69**
2	CYRE3	0.59	**0.69**	0.57	**0.65**	0.54	**0.62**	0.59	**0.69**
2	JBSS3	0.63	**0.7**	0.6	**0.65**	0.58	**0.63**	0.63	**0.7**
2	CSAN3	0.57	**0.68**	0.57	**0.67**	0.57	**0.67**	0.57	**0.68**
3	ENBR3	0.58	**0.69**	0.57	**0.66**	0.55	**0.64**	0.58	**0.69**
3	CPLE6	0.67	**0.71**	0.66	**0.7**	0.66	**0.7**	0.67	**0.71**
3	EGIE3	0.66	**0.69**	0.65	**0.68**	0.64	**0.68**	0.66	**0.69**
4	SBSP3	0.65	**0.73**	0.64	**0.71**	0.63	**0.71**	0.65	**0.73**
4	KROT3	0.53	**0.72**	0.52	**0.7**	0.51	**0.7**	0.53	**0.72**
4	NATU3	0.66	**0.69**	0.65	**0.69**	0.65	**0.68**	0.66	**0.69**
5	BRML3	**0.57**	0.46	**0.56**	0.47	**0.53**	0.43	**0.57**	0.46
5	BRFS3	0.68	**0.69**	0.67	**0.69**	0.67	**0.68**	0.68	**0.69**
6	LAME4	0.61	**0.62**	0.61	0.61	0.6	0.6	0.61	**0.62**
6	MRVE3	0.55	**0.67**	0.55	**0.64**	0.53	**0.63**	0.55	**0.67**
6	ESTC3	**0.64**	0.59	**0.63**	0.59	**0.62**	0.57	**0.64**	0.59
6	LREN3	0.57	**0.63**	0.57	**0.63**	0.56	**0.62**	0.57	**0.63**
6	EQTL3	0.56	**0.73**	0.56	**0.72**	0.56	**0.72**	0.56	**0.73**
7	FIBR3	0.55	**0.64**	0.54	**0.63**	0.54	**0.62**	0.55	**0.64**
7	MRFG3	0.67	**0.69**	0.62	**0.63**	0.59	**0.61**	0.67	**0.69**
7	ITSA4	0.53	**0.7**	0.52	**0.65**	0.49	**0.62**	0.53	**0.7**

8	CMIG4	0.69	**0.74**	0.64	**0.68**	0.62	**0.66**	0.69	**0.74**
8	WEGE3	0.63	**0.76**	0.61	**0.72**	0.6	**0.72**	0.63	**0.76**
8	CPFE3	0.63	**0.66**	0.6	**0.63**	0.59	**0.62**	0.63	**0.66**
9	BBSE3	0.58	**0.73**	0.57	**0.72**	0.57	**0.72**	0.58	**0.73**
9	CSNA3	0.66	**0.73**	0.64	**0.7**	0.63	**0.7**	0.66	**0.73**
10	HYPE3	0.63	**0.65**	0.62	**0.64**	0.61	**0.62**	0.63	**0.65**
10	ABEV3	**0.63**	0.6	**0.61**	0.59	**0.59**	0.58	**0.63**	0.6
10	CCRO3	0.63	**0.66**	0.61	**0.64**	0.6	**0.63**	0.63	**0.66**
11	MULT3	0.71	**0.72**	0.71	**0.72**	0.71	**0.72**	0.71	**0.72**
11	QUAL3	**0.74**	0.64	**0.72**	0.63	**0.71**	0.62	**0.74**	0.64
11	BRKM5	0.65	**0.72**	0.64	**0.71**	0.64	**0.7**	0.65	**0.72**
12	VALE3	0.54	**0.69**	0.54	**0.68**	0.53	**0.68**	0.54	**0.69**
13	ITUB4	0.56	**0.66**	0.56	**0.65**	0.55	**0.65**	0.56	**0.66**
13	RENT3	**0.7**	0.67	**0.7**	0.67	**0.69**	0.66	**0.7**	0.67
14	RADL3	**0.71**	0.7	**0.71**	0.69	**0.7**	0.69	**0.71**	0.7
14	CIEL3	0.62	**0.73**	0.62	**0.71**	0.61	**0.71**	0.62	**0.73**
15	UGPA3	0.71	0.71	0.7	**0.71**	0.7	**0.71**	0.71	**0.71**
15	USIM5	0.67	**0.72**	0.62	**0.66**	0.6	**0.63**	0.67	**0.72**
16	PETR4	0.67	**0.74**	0.65	**0.72**	0.65	**0.71**	0.67	**0.74**
16	PETR3	0.65	**0.75**	0.64	**0.73**	0.64	**0.73**	0.65	**0.75**
17	GGBR4	0.55	**0.68**	0.54	**0.65**	0.53	**0.65**	0.55	**0.68**
17	GOAU4	0.54	**0.73**	0.53	**0.67**	0.5	**0.65**	0.54	**0.73**
17	BRAP4	0.56	**0.7**	0.55	**0.67**	0.54	**0.67**	0.56	**0.7**
18	BBDC4	0.6	**0.7**	0.59	**0.69**	0.59	**0.69**	0.6	**0.7**
18	BBDC3	0.6	**0.75**	0.59	**0.74**	0.59	**0.74**	0.6	**0.75**
19	VIVT4	**0.7**	0.69	**0.69**	0.68	0.68	0.68	**0.7**	0.69
19	TIMP3	0.59	**0.73**	0.57	**0.67**	0.54	**0.65**	0.59	**0.73**

D
Resultados do modelo DB-LSTM

Tabela D.1: Resultados do método DB-LSTM com raio de alcance $\epsilon = 0.25$ e do LSTM

		recall		**precision**		**f1-score**		**acurácia**	
nº cluster	ação	DB-LSTM	LSTM	DB-LSTM	LSTM	DB-LSTM	LSTM	DB-LSTM	LSTM
1	PETR3	**0.71**	0.64	**0.71**	0.64	**0.71**	0.64	**0.7085**	0.6441
1	PETR4	**0.67**	0.66	**0.67**	0.66	**0.67**	0.66	**0.6656**	0.6611
2	CSNA3	**0.69**	0.65	**0.69**	0.65	**0.69**	0.65	**0.691**	0.6459
2	BBSE3	**0.66**	0.57	**0.66**	0.57	**0.66**	0.57	**0.663**	0.5719
3	BBDC3	**0.7**	0.6	**0.7**	0.6	**0.7**	0.6	**0.704**	0.596
3	BBDC4	**0.66**	0.6	**0.66**	0.6	**0.66**	0.6	**0.6616**	0.596
4	BRAP4	**0.69**	0.56	**0.69**	0.56	**0.69**	0.56	**0.689**	0.556
4	EMBR3	**0.69**	0.61	**0.69**	0.61	**0.69**	0.61	**0.6874**	0.615
5	CPLE6	**0.69**	0.66	**0.69**	0.66	**0.69**	0.66	**0.694**	0.66
5	EGIE3	**0.69**	0.65	**0.69**	0.65	**0.69**	0.65	**0.6885**	0.65
6	BBAS3	**0.58**	0.53	**0.58**	0.53	**0.58**	0.53	**0.5777**	0.5314
6	SANB11	**0.64**	0.53	**0.64**	0.53	**0.64**	0.53	**0.64**	0.5314
7	GOUA4	**0.66**	0.54	**0.66**	0.54	**0.66**	0.54	**0.664**	0.5335
7	GGBR4	**0.68**	0.55	**0.68**	0.55	**0.68**	0.55	**0.6761**	0.5503
8	VIVT4	**0.71**	0.69	**0.71**	0.69	**0.71**	0.69	**0.709**	0.687
8	TIMP3	**0.68**	0.57	**0.68**	0.57	**0.68**	0.57	**0.6796**	0.5721

Tabela D.2: Resultados do método DB-LSTM com raio de alcance $\epsilon = 0.3$ e do LSTM

		recall		**precision**		**f1-score**		**acurácia**	
nº cluster	ação	DB-LSTM	LSTM	DB-LSTM	LSTM	DB-LSTM	LSTM	DB-LSTM	LSTM
1	PETR3	**0.71**	0.64	**0.71**	0.64	**0.71**	0.64	**0.7085**	0.6441
1	PETR4	**0.67**	0.66	**0.67**	0.66	**0.67**	0.66	**0.6656**	0.6611

2	ABEV3	**0.63**	0.61	**0.63**	0.61	**0.63**	0.61	**0.6304**	0.6067
2	HYPE3	**0.68**	0.62	**0.68**	0.62	**0.68**	0.62	**0.6808**	0.6193
3	CSNA3	**0.69**	0.65	**0.69**	0.65	**0.69**	0.65	**0.6939**	0.646
3	BBSE3	**0.7**	0.57	**0.7**	0.57	**0.7**	0.57	**0.6974**	0.5719
3	USIM5	**0.65**	0.62	**0.65**	0.62	**0.65**	0.62	**0.6476**	0.6234
4	BRML3	**0.62**	0.56	**0.62**	0.56	**0.62**	0.56	**0.6179**	0.5583
4	BRFS3	**0.69**	0.67	**0.69**	0.67	**0.69**	0.67	**0.6924**	0.6702
5	BBDC3	**0.7**	0.6	**0.7**	0.6	**0.7**	0.6	**0.704**	0.5954
5	BBDC4	**0.66**	0.6	**0.66**	0.6	**0.66**	0.6	**0.6616**	0.5956
6	BRAP4	**0.67**	0.56	**0.67**	0.56	**0.67**	0.56	**0.6691**	0.556
6	BBAS3	**0.68**	0.53	**0.68**	0.53	**0.68**	0.53	**0.6775**	0.5314
6	EMBR3	**0.66**	0.61	**0.66**	0.61	**0.66**	0.61	**0.6636**	0.6159
6	SANB11	**0.68**	0.53	**0.68**	0.53	**0.68**	0.53	**0.6769**	0.5314
7	CMIG4	**0.66**	0.65	**0.66**	0.65	**0.66**	0.65	**0.6604**	0.6474
7	WEGE3	**0.7**	0.61	**0.7**	0.61	**0.7**	0.61	**0.7015**	0.6111
8	ITUB4	**0.65**	0.56	**0.65**	0.56	**0.65**	0.56	**0.6484**	0.562
8	CIEL3	**0.7**	0.62	**0.7**	0.62	**0.7**	0.62	**0.6991**	0.62
9	ENBR3	**0.67**	0.57	**0.67**	0.57	**0.67**	0.57	**0.6722**	0.5715
9	CPLE6	**0.71**	0.66	**0.71**	0.66	**0.71**	0.66	**0.709**	0.66
9	EGIE3	**0.69**	0.65	**0.69**	0.65	**0.69**	0.65	**0.6933**	0.6496
10	SBSP3	**0.68**	0.64	**0.68**	0.64	**0.68**	0.64	**0.679**	0.6384
10	NATU3	**0.69**	0.65	**0.69**	0.65	**0.69**	0.65	**0.6923**	0.6535
11	GOAU4	**0.66**	0.53	**0.66**	0.53	**0.66**	0.53	**0.664**	0.5335
11	GGBR4	**0.68**	0.55	**0.68**	0.55	**0.68**	0.55	**0.6762**	0.5504
12	VIVT4	**0.71**	0.69	**0.71**	0.69	**0.71**	0.69	**0.7094**	0.6869
12	TIMP3	**0.68**	0.57	**0.68**	0.57	**0.68**	0.57	**0.6796**	0.5722
13	ESTC3	**0.68**	0.63	**0.68**	0.63	**0.68**	0.63	**0.6844**	0.6261
13	EQTL3	**0.68**	0.56	**0.68**	0.56	**0.68**	0.56	**0.6815**	0.5605
13	LREN3	**0.63**	0.57	**0.63**	0.57	**0.63**	0.57	**0.6316**	0.5691

Tabela D.3: Resultados do método DB-LSTM com raio de alcance $\epsilon = 0.35$ e do LSTM

		recall		**precision**		**f1-score**		**acurácia**	
nº cluster	ação	DB-LSTM	LSTM	DB-LSTM	LSTM	DB-LSTM	LSTM	DB-LSTM	LSTM
1	PETR3	**0.71**	0.64	**0.71**	0.64	**0.71**	0.64	**0.7085**	0.6442
1	PETR4	**0.67**	0.66	**0.67**	0.66	**0.67**	0.66	**0.6656**	0.6611
2	ABEV3	**0.63**	0.61	**0.63**	0.61	**0.63**	0.61	**0.6305**	0.6068

2	HYPE3	**0.68**	0.62	**0.68**	0.62	**0.68**	0.62	**0.6809**	0.6193
3	CSNA3	0.63	**0.65**	0.63	**0.65**	0.63	**0.65**	0.6334	**0.6460**
3	BBSE3	**0.68**	0.57	**0.68**	0.57	**0.68**	0.57	**0.6804**	0.5719
3	USIM5	**0.64**	0.62	**0.64**	0.62	**0.64**	0.62	**0.6450**	0.6234
3	UGPA3	0.71	0.71	0.71	0.71	0.71	0.71	**0.7148**	0.7067
4	BRML3	**0.62**	0.56	**0.62**	0.56	**0.62**	0.56	**0.6179**	0.5583
4	BRFS3	**0.69**	0.67	**0.69**	0.67	**0.69**	0.67	**0.6925**	0.6702
5	RENT3	**0.71**	0.69	**0.71**	0.69	**0.71**	0.69	**0.7093**	0.6949
5	ITUB4	**0.69**	0.56	**0.69**	0.56	**0.69**	0.56	**0.6892**	0.5619
5	CIEL3	**0.67**	0.62	**0.67**	0.62	**0.67**	0.62	**0.6691**	0.6198
6	BBDC3	**0.7**	0.6	**0.7**	0.6	**0.7**	0.6	**0.7040**	0.5953
6	BBDC4	**0.66**	0.6	**0.66**	0.6	**0.66**	0.6	**0.6616**	0.5957
7	BRAP4	**0.67**	0.56	**0.67**	0.56	**0.67**	0.56	**0.6691**	0.5561
7	BBAS3	**0.68**	0.53	**0.68**	0.53	**0.68**	0.53	**0.6776**	0.5314
7	EMBR3	**0.66**	0.61	**0.66**	0.61	**0.66**	0.61	**0.6636**	0.6149
7	SANB11	**0.68**	0.53	**0.68**	0.53	**0.68**	0.53	**0.6769**	0.5314
8	CMIG4	**0.66**	0.65	**0.66**	0.65	**0.66**	0.65	**0.6604**	0.6475
8	WEGE3	**0.7**	0.61	**0.7**	0.61	**0.7**	0.61	**0.7015**	0.6112
9	CYRE3	**0.65**	0.56	**0.65**	0.56	**0.65**	0.56	**0.6547**	0.5644
9	JBSS3	**0.65**	0.6	**0.65**	0.6	**0.65**	0.6	**0.6545**	0.6035
9	CSAN3	**0.71**	0.57	**0.71**	0.57	**0.71**	0.57	**0.7077**	0.5716
10	PCAR4	**0.75**	0.68	**0.75**	0.68	**0.75**	0.68	**0.7457**	0.6750
10	ECOR3	**0.66**	0.61	**0.66**	0.61	**0.66**	0.61	**0.6561**	0.6132
11	MULT3	**0.73**	0.71	**0.73**	0.71	**0.73**	0.71	**0.7275**	0.7064
11	SBSP3	**0.73**	0.64	**0.73**	0.64	**0.73**	0.64	**0.7296**	0.6384
11	NATU3	**0.72**	0.65	**0.72**	0.65	**0.72**	0.65	**0.7166**	0.6535
12	ENBR3	**0.67**	0.57	**0.67**	0.57	0.67	0.57	**0.6723**	0.5715
12	CPLE6	**0.71**	0.66	**0.71**	0.66	**0.71**	0.66	**0.7091**	0.6599
12	EGIE3	**0.69**	0.65	**0.69**	0.65	**0.69**	0.65	**0.6933**	0.6496
13	MRVE3	**0.67**	0.54	**0.67**	0.54	**0.67**	0.54	**0.6747**	0.5434
13	ESTC3	**0.68**	0.63	**0.68**	0.63	**0.68**	0.63	**0.6818**	0.6261
13	EQTL3	**0.71**	0.56	**0.71**	0.56	**0.71**	0.56	**0.7125**	0.5605
13	LREN3	**0.66**	0.57	**0.66**	0.57	**0.66**	0.57	**0.6594**	0.5691
14	ITSA4	**0.63**	0.53	**0.63**	0.53	**0.63**	0.53	**0.6326**	0.5253
14	MRFG3	**0.67**	0.63	**0.67**	0.63	**0.67**	0.63	**0.6705**	0.6253
14	FIBR3	**0.67**	0.55	**0.67**	0.55	**0.67**	0.55	**0.6684**	0.5453
15	GOAU4	**0.66**	0.53	**0.66**	0.53	**0.66**	0.53	**0.6640**	0.5335
15	GGBR4	**0.68**	0.55	**0.68**	0.55	**0.68**	0.55	**0.6762**	0.5504
16	VIVT4	**0.71**	0.69	**0.71**	0.69	**0.71**	0.69	**0.7094**	0.6869

16	TIMP3	**0.68**	0.57	**0.68**	0.57	**0.68**	0.57	**0.6796**	0.5722