Allan Gabriel Oliveira Lima
Caio Vinicius Soares Amaral
Thales Augusto Souto Rodriguez

FinStockESN-BR: Previsão de séries temporais de preços de ações utilizando análise de sentimento e aprendizado de máquina

São Paulo, SP

2023

Allan Gabriel Oliveira Lima
Caio Vinicius Soares Amaral
Thales Augusto Souto Rodriguez

FinStockESN-BR: Previsão de séries temporais de preços de ações utilizando análise de sentimento e aprendizado de máquina

Trabalho de conclusão de curso apresentado ao Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro.

Universidade de São Paulo – USP

Escola Politécnica

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

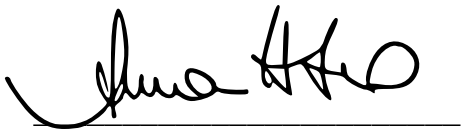
Orientador: Prof. Dra. Anna Helena Reali Costa

São Paulo, SP

2023

São Paulo, 12 de dezembro de 2023.

De acordo,



Anna Helena Reali Costa

Catalogação-na-publicação

Lima, Allan Gabriel Oliveira
FinStockESN-BR: Previsão de séries temporais de preços de ações utilizando análise de sentimento e aprendizado de máquina / A. G. O. Lima, C. V. S. Amaral, T. A. S. Rodriguez -- São Paulo, 2023.
95 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Previsão de Séries Temporais 2.Aprendizado de Máquina 3.Análise de Sentimento 4.Finanças Quantitativas I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t. III.Amaral, Caio Vinícius Soares IV.Rodriguez, Thales Augusto Souto

Agradecimentos

A Deus, fonte de força e perseverança ao longo do caminho. Aos meus pais, Adna e Marcelo, e à minha avó Abigail, verdadeiros pilares de amor e dedicação, que sempre estiveram ao meu lado, apoiando e acreditando em cada passo que dei. À professora Anna Reali, pela orientação precisa, conhecimento compartilhado e paciência ao guiar-nos nesta fase acadêmica. Aos meus colegas de projeto, Caio e Thales, cuja colaboração e empenho foram essenciais. E a todos os que, direta ou indiretamente, contribuíram para minha formação e para a conclusão deste trabalho.

Allan Gabriel Oliveira Lima

Gostaria de agradecer primeiramente a Deus, por ter me dado a capacidade de chegar até aqui. Depois gostaria de agradecer aos meus pais Ana e Claudio, por serem uma fonte incansável de apoio e encorajamento, sempre me incentivando a perseguir meus sonhos e estando presentes nos momentos bons e ruins. Além disso, sou grato aos meus colegas Allan e Thales, que me acompanharam ao longo do ano nesta jornada, e à professora Anna Reali, que nos guiou e aconselhou durante esse processo. Finalmente, agradeço a todos os professores, colegas e familiares que foram essenciais ao meu crescimento pessoal, acadêmico e professional nestes anos de graduação.

Caio Vinícius Soares Amaral

Minha profunda gratidão é dedicada aos meus pais, por seu amor incansável e suporte constante. Eles foram fundamentais em cada etapa da minha jornada, oferecendo não apenas encorajamento, mas também sendo exemplos vivos de dedicação e força. Um agradecimento sincero à professora Anna Reali, pela orientação, paciência e apoio que tornou este trabalho possível. Ao Allan e Caio, pela colaboração e dedicação, e a todos os professores, amigos e familiares que, de alguma forma, contribuíram para minha formação e para a realização deste trabalho.

Thales Augusto Souto Rodriguez

Resumo

Este trabalho apresenta o desenvolvimento de um modelo para a previsão de séries temporais de preços de ações no mercado financeiro brasileiro, utilizando Redes de Estado Eco com Integrador com Vazamento (Leaky-integrator Echo State Networks - LiESN). Com o objetivo de aprimorar a precisão das previsões, o modelo integra três conjuntos de dados distintos: análise fundamentalista, indicadores macroeconômicos e análises de sentimentos derivadas de notícias textuais, específicas para cada ação. O foco é avaliar o modelo de previsão desenvolvido e identificar quais tipos de dados de entrada são mais eficazes na previsão das séries temporais de preços. O projeto é dividido em três etapas principais. Inicialmente, o modelo de Processamento de Linguagem Natural FinBERT pt-br é incorporado ao estudo para a análise de sentimentos das notícias. Em seguida, o modelo de previsão é desenvolvido, agregando análises de sentimentos e comparando os resultados com diferentes tipos de dados de entrada. Finalmente, uma aplicação web é desenvolvida para monitorar notícias relacionadas a ações específicas e para exibir as previsões de preços de ações baseadas no modelo. Esta aplicação tem o objetivo de facilitar o acesso e a interação dos usuários com as previsões geradas.

Palavras-chave: Série Temporal, LiESN, Mercado Financeiro, Previsão de Preços, FinBERT pt-br, Análise de Sentimento, Finanças Quantitativas.

Abstract

This work presents the development of a model for forecasting time series of stock prices in the Brazilian financial market, employing Leaky-integrator Echo State Networks (LiESN). Aiming to enhance the accuracy of the forecasts, the model integrates three distinct data sets: fundamental analysis, macroeconomic indicators, and sentiment analyses derived from textual news, specific to each stock. The goal is to evaluate the developed prediction model and identify which types of input data are most effective in forecasting price time series. The project is structured into three main stages. Initially, the FinBERT pt-br Natural Language Processing model is incorporated into the study for sentiment analysis. Subsequently, the forecasting model is developed, integrating sentiment analyses and comparing the outcomes with different types of data inputs. Finally, a web application is developed to monitor news related to specific stocks and to display stock price forecasts based on the model. This application aims to facilitate user access and interaction with the generated forecasts.

Keywords: Time Series, LiESN, Financial Market, Price Forecasting, FinBERT pt-br, Sentiment Analysis, Quantitative Finance.

Lista de ilustrações

Figura 1 – Evolução da Posição das Pessoas Físicas na B3. CPF representa um investidor. Conta representa investidor na corretora. Figura extraída de (B3, 2023) . . . 12
Figura 2 – Diagrama de um MLP, com 4 entradas na camada de entrada, 4 camadas ocultas e uma saída na camada de saída. Retirado de (PéREZ-ENCISO; LAURA, 2019). . . . 24
Figura 3 – Representação gráfica dos componentes de um neurônio em um MLP. Retirado de (PéREZ-ENCISO; LAURA, 2019). . . . 25
Figura 4 – Arquitetura de uma ESN. Retirada de (SOH; DEMIRIS, 2014). . . . 27
Figura 5 – Exemplo de vetores para as palavras “King”, “Queen”, “Woman” e “Man”. Nota-se neste exemplo que King - Man + Woman = Queen. Retirada de (SUTOR et al., 2019). . . . 29
Figura 6 – Exemplo do funcionamento do CBOW e do Skip-Gram. O contexto é composto pelas palavras W-2, W-1, W+1 e W+2 ao redor da palavra W0. Retirado de (LING et al., 2015). . . . 31
Figura 7 – Os autores do Glove exemplificam o uso do algoritmo para analisar semelhanças entre palavras do inglês. “Gelo” ocorre muito mais frequentemente com “Sólido” do que “Gasoso”, enquanto o inverso é verdadeiro para “Vapor”. Retirado de (PENNINGTON; SOCHER; MANNING, 2014). . . . 31
Figura 8 – Arquitetura do Modelo de caixa preta do FinStockESN-BR. . . . 40
Figura 9 – Esquema do Gerador de Janelas no modelo FinStockESN-BR. . . . 41
Figura 10 – Representação do Time2Vec utilizado no modelo FinStockESN-BR. Adaptado de (BARROS et al., 2023) . . . 42
Figura 11 – Detalhamento do Modelo Codificador Previsor no FinStockESN-BR. . 43
Figura 12 – Arquitetura da Aplicação para Análise e Previsão de Preços de Ações, Integrando Coleta de Dados Históricos, Análise de Sentimento de Notícias e Recuperação de Dados Econômicos via Serviços. . . . 44
Figura 13 – Visão Geral Metodologia . . . 50
Figura 14 – Utilização do FinBERT PT-BR para cálculo de sentimento. . . . 53
Figura 15 – Exemplo utilizando Word2Vec para cálculo do grau de similaridade. . . 54
Figura 16 – Distribuição dos sentimentos das notícias: sentimento (de -1, indicando muito negativo a +1, muito positivo) versus a porcentagem de notícias. 64
Figura 17 – Distribuição do grau de relação das notícias, com o termo analisado “Petrobrás”. . . . 64
Figura 18 – Número de notícias por fonte utilizada. . . . 65

Figura 19 – Erros de validação encontrados para cada Word Embedding testado. . 71
Figura 20 – Evolução do erro mínimo na primeira sessão de ajuste fino. 72
Figura 21 – Arquitetura Implementada na AWS para a Aplicação Web. 76
Figura 22 – Tela Geral - A interface agrega todas as informações essenciais em um único local, permitindo ao usuário um acesso rápido às notícias, previsões e dados financeiros. 80
Figura 23 – Tela de Notícias - Exibe as notícias para cada ação e data. 80
Figura 24 – Tela de Previsões - Oferece uma representação gráfica das previsões de mercado 81
Figura 25 – Continuação Tela de Previsões - Oferece uma tabela de previsão 81
Figura 26 – Tela de Dados Financeiros - Apresenta dados financeiros detalhados, permitindo aos usuários acessar informações sobre a empresa selecionada. 82
Figura 27 – Evolução do Preço da WEG - R$ Mil 90
Figura 28 – Evolução da Receita da WEG - R$ Mil 91
Figura 29 – Evolução do Lucro da WEG - R$ Mil 91
Figura 30 – Evolução do Patrimônio Líquido da WEG - R$ Mil 92
Figura 31 – Evolução do Fluxo de Caixa Operacional da WEG - R$ Mil 92
Figura 32 – Evolução do IPCA - Mensal 93
Figura 33 – Evolução da SELIC - Mensal Acumulado 94
Figura 34 – Evolução do PIB - Mensal - R$ Milhão 94
Figura 35 – Evolução da Taxa de Desemprego - Mensal 95

Lista de tabelas

Tabela 1 – Descrição dos Dataframes Coletados 65
Tabela 2 – Descrição dos campos nos dataframes 66
Tabela 3 – As 20 Principais Empresas Listadas na B3 em 2023 - Retirado de (Equipe Toro Investimentos, 2023) 67
Tabela 4 – Estrutura do Dataframe de Cotações 68
Tabela 5 – Estrutura e Periodicidade dos Dataframes Macroeconômicos 68
Tabela 6 – Estrutura do DataFrame Final Consolidado 69
Tabela 7 – Tabela com os hiperparâmetros configurados manualmente, e os valores iniciais adotados com base na literatura e em testes empíricos 70
Tabela 8 – Janelas de valores para cada hiperparâmetro 72
Tabela 9 – Valores encontrados para cada hiperparâmetro 72
Tabela 10 – Intervalos dos hiperparâmetros na segunda sessão de ajuste fino 73
Tabela 11 – Valores encontrados para os hiperparâmetros na segunda sessão de ajuste fino 73
Tabela 12 – Comparação do Erro Médio Quadrático (RMSE) para diferentes configurações de entradas do modelo FinSTOCKESN-BR 74

Sumário

1 INTRODUÇÃO . . . 12
1.1 Motivação . . . 13
1.2 Objetivo . . . 14
1.3 Justificativa . . . 15
1.4 Organização do Trabalho . . . 15

2 ASPECTOS CONCEITUAIS . . . 17
2.1 Séries Temporais e Previsão de Preços de Ações . . . 17
2.2 Métricas de erro para previsões de séries temporais . . . 18
2.2.1 Root Mean Squared Error (RMSE) . . . 18
2.2.2 Mean Absolute Scaled Error (MASE) . . . 18
2.3 Mercado Acionário . . . 19
2.3.1 Análise Fundamentalista e Indicadores Macroeconômicos . . . 20
2.4 Instituições e Documentos . . . 22
2.4.1 Comissão de Valores Mobiliários (CVM) . . . 22
2.4.2 Banco Central do Brasil (BACEN) . . . 22
2.4.3 Documentos Contábeis . . . 22
2.4.4 Tipos de Informação Contábil . . . 23
2.5 Redes Neurais . . . 23
2.5.1 Multilayer Perceptron . . . 23
2.5.2 RNN (*Recurrent Neural Networks*) . . . 26
2.6 *Echo State Networks* (ESN) e *Leaky Integrator* ESN (LiESN) . . . 26
2.7 Processamento de Linguagem Natural (PLN) . . . 28
2.7.1 Características Únicas da Linguagem Humana em IA . . . 28
2.8 Representação vetorial de palavras . . . 29
2.8.1 Técnicas de Vetorização de Texto . . . 30
2.8.2 Classificação de Texto . . . 31
2.8.3 TF-IDF . . . 32
2.8.4 Busca simples . . . 33
2.9 Web scraping . . . 33
2.10 Análise de Sentimento . . . 34

3 REVISÃO DA LITERATURA . . . 35
3.1 Metodologia da Revisão de Literatura . . . 35
3.2 Trabalhos relevantes . . . 36

4 ESPECIFICAÇÃO: FINSTOCKESN-BR . . . 38
4.1 Requisitos . . . 38
4.1.1 Requisitos Funcionais . . . 39
4.1.2 Requisitos Não-funcionais . . . 39
4.2 Análise de Sentimentos BERT . . . 39
4.3 Arquitetura do FinStockESN-BR . . . 40
4.3.1 Gerador de Janelas de Dados . . . 41
4.3.2 Método de Codificação Temporal: Time2Vec . . . 42
4.3.3 Codificador e Previsor . . . 42
4.4 Aplicação Web . . . 43
4.4.1 Frontend . . . 44
4.4.2 Backend . . . 45
4.4.2.1 Coleta de Preços Históricos das Ações . . . 45
4.4.2.2 Coleta de Notícias Mais Recentes . . . 46
4.4.2.3 Coleta de Dados Microeconômicos e Macroeconômicos . . . 47
4.4.2.4 Serviço de Predição . . . 47

5 MÉTODO DO TRABALHO . . . 49
5.1 Visão Geral Metodologia . . . 49
5.2 Desenvolvimento do Scraping de Notícias . . . 50
5.2.1 Seleção de Fontes de Notícias . . . 50
5.2.2 Concepção do Pipeline de Extração Automática . . . 51
5.2.3 Extração e Estruturação dos Dados . . . 51
5.2.4 Pré-processamento e Purificação dos Dados . . . 52
5.2.5 Conclusão da Coleta e Preparação dos Dados . . . 53
5.3 Cálculo de Sentimento . . . 53
5.4 Filtragem de Notícias por Empresa e Word Embeddings . . . 54
5.5 Aquisição dos Dados Fundamentalistas da CVM . . . 55
5.5.1 Metodologia de Coleta . . . 55
5.5.2 Construção dos Dataframes . . . 55
5.5.3 Cálculo dos Dados Trimestrais . . . 56
5.5.4 Preparação Final dos Dados . . . 56
5.6 Aquisição dos dados macroeconômicos através do BACEN . . . 56
5.7 Obtenção da Série Histórica de Preços . . . 56
5.7.1 Treinamento, Validação e Teste . . . 57
5.7.2 Parâmetros e otimização de hiperparâmetros . . . 57
5.8 Metodologia para a Construção da Aplicação Web . . . 58
5.8.1 Desenvolvimento do Frontend . . . 59
5.8.2 Desenvolvimento do Backend . . . 59

6 DESENVOLVIMENTO DO TRABALHO 60
6.1 Tecnologias Utilizadas 60
6.1.1 Python 60
6.1.2 Jupyter Notebook 60
6.1.3 Anaconda 61
6.1.4 TensorFlow 61
6.1.5 Gensim 61
6.1.6 NTLK 62
6.1.7 Pandas 62
6.1.8 Dask 63
6.2 Projeto e Implementação 63
6.2.1 Scraping de Notícias 63
6.2.2 Resultados cálculo do índice de sentimento 63
6.2.3 Aquisição dos Dados Fundamentalistas da CVM 65
6.2.4 Obtenção da Série Histórica de Preços 67
6.2.5 Aquisição de Dados Macroeconômicos 68
6.2.6 Junção Dados Fundamentalistas, Macroeconômicos e Série de Preços . . . 69
6.3 Definição de número de épocas e primeiros treinamentos 70
6.4 Escolha de WordEmbedding 70
6.5 Ajuste Fino (Finetuning) 71
6.5.1 Primeiro Ajuste Fino 71
6.5.2 Segundo Ajuste Fino 73
6.6 Resultados do modelo 73
6.7 Implementação da Aplicação Web 74
6.7.1 Backend 76
6.7.2 Implementação do Frontend em Next.js 79

7 CONSIDERAÇÕES FINAIS 83
7.1 Perspectivas de Continuidade 83

REFERÊNCIAS 85

APÊNDICES 89

APÊNDICE A – EXEMPLIFICAÇÃO DADOS COM GRÁFICOS . 90
A.1 Dados Fundamentalistas e Preço da WEG 90
A.2 Dados Macroeconômicos 93

1 Introdução

A bolsa brasileira, também conhecida como B3, é o principal mercado financeiro do país e é responsável por negociar ações, derivativos, títulos públicos e privados, entre outros ativos. Ela é considerada uma das bolsas de valores mais importantes da América Latina e tem atraído cada vez mais investidores estrangeiros, que buscam oportunidades de investimento em empresas brasileiras. Em um cenário globalizado de mercados financeiros pós pandemia, a evolução do mercado de ações brasileiro destaca-se como um caso notável. Em 2018, a Bolsa de Valores brasileira, registrava um número de cerca de 700 mil investidores Pessoa Física. No entanto, até dezembro de 2022, esse total havia saltado para 5 milhões, refletindo um crescimento explosivo de mais de 700% (B3, 2023).

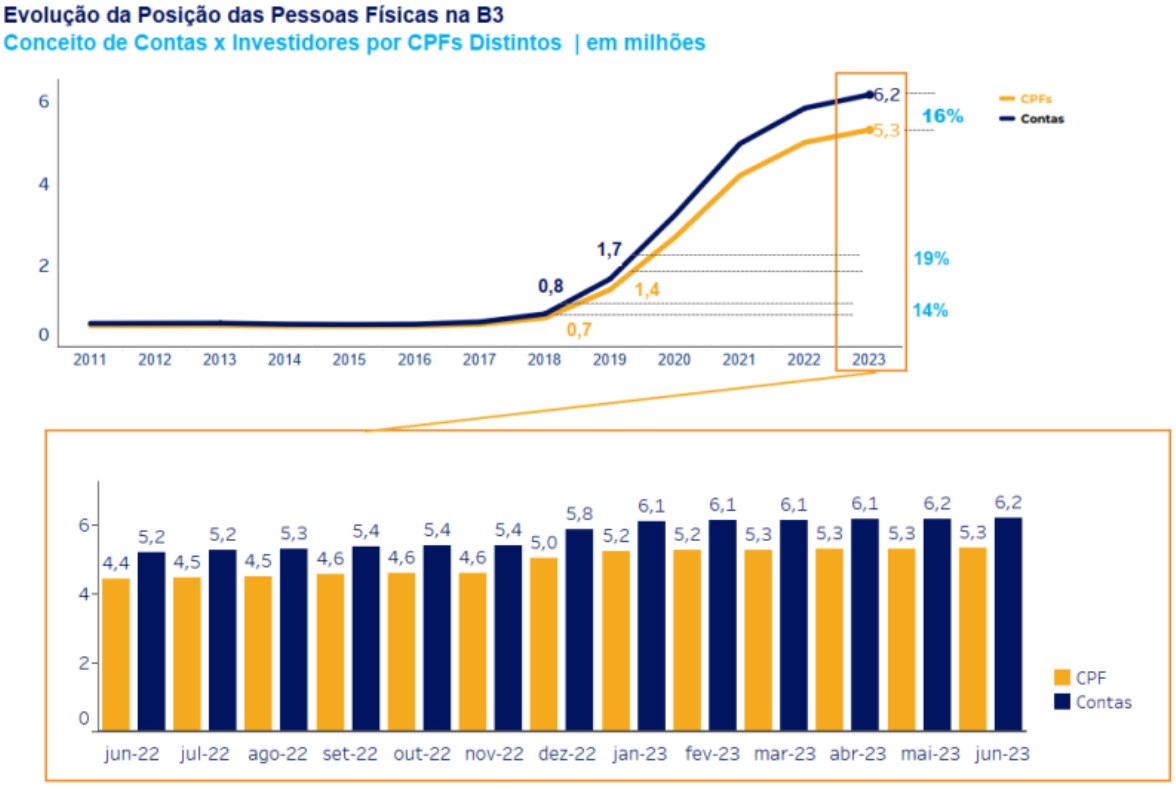


Figura 1 – Evolução da Posição das Pessoas Físicas na B3. CPF representa um investidor. Conta representa investidor na corretora. Figura extraída de (B3, 2023)

Este fenômeno pode ser atribuído a diversos fatores, desde a democratização do acesso à informação até as constantes mudanças na percepção e entendimento sobre o comportamento do mercado de ações. Neste contexto de crescimento acelerado e dinamismo, a demanda por ferramentas de análise voltadas especificamente para o mercado brasileiro torna-se relevante.

Embora existam diversas empresas estrangeiras que forneçam dados e análises abrangentes para múltiplos mercados ao redor do mundo (incluindo o mercado brasileiro), muitas delas carecem de uma visão mais localizada e aprofundada do ambiente econômico e sociocultural brasileiro.

1.1 Motivação

Para se ter uma compreensão da motivação por trás da modelagem no mercado financeiro, é essencial revisitar a história das teorias que moldaram essa área. Malkiel e Fama (1970) apresentaram ao mundo financeiro o conceito de "Mercado Eficiente". Segundo essa teoria, os preços das ações refletiam todas as informações disponíveis e, por isso, eram essencialmente imprevisíveis. Isso sugeria que nenhuma estratégia de investimento poderia consistentemente superar o mercado, considerando os riscos envolvidos.

Entretanto, nas décadas subsequentes, observou-se que o comportamento do mercado muitas vezes desafiava essa noção. Anomalias de mercado, como movimentos de preços que não pareciam estar alinhados com as informações disponíveis, começaram a desafiar a ideia dos mercados sempre eficientes. Essas observações levaram a um exame mais profundo do comportamento do investidor. Lo (2004) introduziu a Hipótese do Mercado Adaptativo (AMH). Contrariamente à ideia original de Fama, a AMH propõe que os investidores não são sempre racionais. Em vez disso, seu comportamento é influenciado por uma série de fatores psicológicos e emocionais, resultando em decisões de investimento que nem sempre seguem a lógica financeira pura.

A crescente compreensão de que o comportamento humano pode introduzir padrões irregulares e, por vezes, previsíveis no mercado, abriu portas para a utilização de técnicas avançadas de análise. A inteligência artificial, particularmente, emergiu como uma ferramenta poderosa para identificar tais anomalias. Ao contrário dos métodos tradicionais que muitas vezes não conseguem lidar com a complexidade e a natureza não linear dos dados de mercado, as técnicas de IA são particularmente aptas a identificar padrões ocultos em grandes conjuntos de dados, muitos dos quais podem ser influenciados por fatores psicológicos e emocionais que desafiam as premissas de racionalidade pura.

Um exemplo notável é a utilização de Echo State Networks (ESNs) otimizadas, que demonstraram sucesso na previsão de dados financeiros, ultrapassando métodos tradicionais de análise de séries temporais. Esses avanços são ilustrados por estudos como o de Liu et al. (2018), que destacam a eficácia das ESNs na modelagem de séries temporais financeiras. Além disso, um modelo híbrido baseado em ESN foi proposto por Trierweiler Ribeiro et al. (2021) para gerenciar incertezas em portfólios de investimento, ressaltando a importância de compreender as flutuações dos preços dos ativos para a gestão do risco financeiro e a tomada de decisões informadas.

Outro estudo focado na previsão de preços de ações de curto prazo com ESNs aplicou o expoente de Hurst para adaptar a inicialização do modelo, visando aprimorar a previsibilidade durante o treinamento (LIN; YANG; SONG, 2009). Isso demonstra a capacidade das ESNs de lidar com a natureza não-linear e dinâmica dos dados de séries temporais no mercado financeiro. Além disso. a incorporação de dados auxiliares, como indicadores macroeconômicos, indicadores fundamentalistas e análise de sentimentos derivadas de notícias textuais, pode oferecer uma compreensão mais holística das forças que movem os preços das ações. Esta abordagem multidimensional pode não apenas enriquecer a acurácia das previsões, mas também proporcionar *insights* mais profundos sobre as interações complexas dentro dos mercados financeiros.

1.2 Objetivo

O objetivo principal deste trabalho é desenhar e implementar uma metodologia para a previsão de preços de ações na bolsa de valores brasileira, empregando o *Leaky integrator Echo State Network* (LiESN), que já demonstrou eficácia em outros domínios. A escolha da LiESN é motivada por sua capacidade comprovada de lidar com a complexidade e não-linearidade inerentes às séries temporais financeiras.

Além disso, este trabalho se propõe a enriquecer a modelagem de previsão incorporando outras fontes de dados auxiliares. A análise de sentimentos derivada de notícias textuais, dados oriundos de análise fundamentalista e indicadores macroeconômicos serão integrados ao modelo de previsão. A integração dessas fontes de dados tem o potencial de capturar uma gama mais ampla de fatores que influenciam os preços das ações, possibilitando previsões mais informadas e, possivelmente, mais precisas.

Para facilitar a análise e como viés prático, será desenvolvida uma aplicação que permita a coleta automática de notícias e informações financeiras relacionadas a cada ação em particular. Esta aplicação incorporará essas informações na previsão de preços. Este componente prático visa não apenas automatizar o processo de coleta de dados, mas também explorar como as informações derivadas da análise de sentimentos podem ser utilizadas para aprimorar a precisão das previsões de preços.

De maneira geral, o objetivo é avaliar se a abordagem proposta pode oferecer uma vantagem competitiva para os investidores, proporcionando *insights* mais acurados sobre a direção futura dos preços das ações. A avaliação do desempenho do modelo proposto e a análise dos resultados obtidos serão importantes para entender o potencial e as limitações da abordagem, e para sugerir direções para pesquisas futuras na interseção entre inteligência artificial, análise de sentimentos, dados macro e microeconômicos e previsão do mercado financeiro.

1.3 Justificativa

Este trabalho fundamenta-se na premissa de que o mercado financeiro, e especificamente a bolsa de valores brasileira, constitui um ambiente de alta competitividade e constante mutabilidade, onde decisões informadas são cruciais para o sucesso de seus agentes. A volatilidade inerente ao mercado brasileiro, com suas peculiaridades, oferece um campo vasto para análise e investigação, tornando-o um cenário ideal para aplicação de ferramentas financeiras.

Além disso, o estudo baseia-se na ideia de examinar como a inclusão da análise de sentimentos, ao lado de dados fundamentalistas e indicadores macroeconômicos, pode enriquecer o modelo de previsão de preços de ativos. A hipótese é que as oscilações nos preços das ações podem ser parcialmente atribuídas a fatores comportamentais, onde as decisões de investimento são frequentemente influenciadas por sentimentos e percepções subjetivas dos agentes de mercado, divergindo das expectativas baseadas exclusivamente em análises financeiras tradicionais.

Portanto, a justificativa para este trabalho reside na necessidade de explorar metodologias avançadas que possam oferecer aos investidores ferramentas de previsão mais precisas e confiáveis. A integração da análise de sentimentos e dados macroeconômicos ao tradicional modelo de dados fundamentalistas promete não apenas uma contribuição acadêmica para o campo da economia e finanças, mas também implicações práticas para a estratégia de investimento e gestão de riscos.

Estudos recentes têm demonstrado que a integração de múltiplas fontes de dados pode enriquecer a modelagem e a previsão no domínio financeiro. Por exemplo, Cao (2022) empregou análise de sentimentos para prever movimentos de preços de ações, ilustrando como esta abordagem pode ser utilizada para capturar o sentimento do investidor e possivelmente prever movimentos de preços de ações.

Adicionalmente, o desenvolvimento de uma aplicação prática que facilite o acompanhamento diário das notícias e a previsão dos preços das ações traduz a intenção de transformar insights acadêmicos em soluções tangíveis, facilitando a tomada de decisões informadas em um ambiente de mercado dinâmico. Através de uma abordagem que une os campos da inteligência artificial, computação e análise financeira, este trabalho aspira não apenas a contribuir academicamente para tais campos, mas também a fornecer uma solução prática que pode servir de uso para a comunidade financeira e investidora.

1.4 Organização do Trabalho

O capítulo 2 contém explicações de vários conceitos teóricos que foram empregados ao longo deste trabalho, como séries temporais, métricas de erro, noções básicas sobre o

mercado de ações, etc. Esses conceitos estão reunidos de acordo com a área do conhecimento à qual pertencem, e eles estão explicados separadamente pois o entendimento prévio deles é essencial para compreender os detalhes do projeto.

O capítulo 3 apresenta a revisão de literatura que foi feita em preparação para este projeto, detalhando alguns dos trabalhos relacionados que inspiraram e conduziram a pesquisa a ser realizada aqui. Esses trabalhos introduziram alguns elementos importantes usados neste projeto, como modelos de previsão de série temporal, análise de sentimento, técnicas de vetorização, etc.

No capítulo 4 está a especificação formal do projeto, elencando seus requisitos funcionais e não-funcionais bem como um diagrama da sua arquitetura. A especificação serve como um guia para quais são os objetivos que devem ser cumpridos no decorrer do trabalho e também como uma síntese das funcionalidades do sistema desenvolvido.

O capítulo 5 descreve a metodologia utilizada neste trabalho, como foi feita a pesquisa de referências, como foram obtidos os dados financeiros, e como o modelo ESN foi treinado. Ele entra em detalhes sobre as atividades que foram feitas ao longo do projeto, e como elas contribuíram para o resultado final.

No capítulo 6 são apresentadas as tecnologias auxiliares que foram usadas para pôr o projeto em prática, e também os resultados obtidos em cada etapa. Aqui estão descritos os dados que foram extraídos da internet e usados para o treinamento, os testes que foram realizados para validação, a performance final do modelo e desenvolvimentos da aplicação web.

Finalmente, no capítulo 7, está a conclusão do grupo sobre o trabalho que foi realizado ao longo do ano, seguida pelas referências bibliográficas utilizadas e o apêndice. Também são discutidas perspectivas de continuidade do projeto, como ele pode ser aprimorado no futuro, lições aprendidas, entre outras considerações finais.

2 Aspectos Conceituais

Este capítulo visa fundamentar os aspectos conceituais que dão suporte ao desenvolvimento do modelo de previsão de séries temporais, com foco específico na bolsa brasileira de valores.

2.1 Séries Temporais e Previsão de Preços de Ações

Séries temporais são uma sequência de pontos de dados, medidos ou registrados em um intervalo de tempo específico (BOX et al., 2016). No campo da engenharia financeira e econômica, séries temporais são essenciais para entender o comportamento dos mercados ao longo do tempo, sendo comumente aplicadas na análise de preços de ações, preços de commodities, taxas de câmbio, entre outros.

Existem diferentes métodos para prever séries temporais, que podem ser categorizados como métodos univariados ou multivariados. Os métodos univariados consideram apenas uma série temporal, sem levar em consideração outras variáveis, enquanto os métodos multivariados consideram múltiplas variáveis ou séries temporais simultaneamente ao fazer previsões (HYNDMAN; ATHANASOPOULOS, 2021).

Em termos de previsão de séries podemos elencar os seguintes tipos:

Previsão de um passo no futuro: Também referida como previsão de curto prazo, esta abordagem foca na predição do próximo valor imediato na série temporal, usando os dados disponíveis até o momento atual. Esse tipo de previsão é crucial quando as decisões precisam ser tomadas rapidamente baseadas na tendência mais recente dos dados (KUMAR, 2023).

Previsão do k-ésimo passo no futuro: Esta abordagem estende a previsão de um passo no futuro para prever o valor da série temporal em um ponto futuro específico, digamos, k passos à frente. Diferentemente da previsão de um passo, que utiliza as observações até o momento t para prever o valor no momento t+1, a previsão do k-ésimo passo usa as observações até o momento t para prever o valor no momento t+k, onde k é um inteiro que representa o número de passos à frente no tempo (JAIN, 2021).

Previsão multi-horizonte: Similar à previsão do k-ésimo passo, a previsão multi-horizonte busca prever valores em múltiplos pontos futuros simultaneamente. Esta abordagem é útil em cenários onde é benéfico entender como a série temporal poderá se comportar

ao longo de um intervalo de tempo, em vez de em um único ponto futuro (JAIN, 2021).

2.2 Métricas de erro para previsões de séries temporais

Para avaliar a precisão das previsões em séries temporais, frequentemente métricas como RMSE (Root Mean Square Error) e MASE (Mean Absolute Scaled Error) são utilizadas devido à sua capacidade de quantificar e comparar os erros de previsão. O RMSE mede a raiz do erro médio quadrático entre os valores previstos e reais, enquanto o MASE, ou Erro Médio Absoluto Escalonado, normaliza o erro pela variabilidade dos dados, tornando-o útil para comparações em diferentes escalas e séries temporais.

2.2.1 Root Mean Squared Error (RMSE)

Conceito e Cálculo: O Root Mean Squared Error (RMSE) é uma das métricas mais populares para avaliar o desempenho de modelos preditivos, particularmente em tarefas de regressão. O RMSE é calculado pela fórmula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - f_i)^2}$$

Interpretação: O quadrado dos erros serve para dois propósitos principais. Primeiro, garante que os erros sejam positivos, eliminando o problema de subestimação que poderia surgir devido ao cancelamento de erros positivos e negativos. Segundo, o quadrado dos erros penaliza desproporcionalmente erros maiores em comparação com erros menores.

O RMSE representa o desvio padrão dos resíduos (erros). Um RMSE mais baixo indica um modelo que pode ajustar ou prever os dados de forma mais precisa, enquanto um RMSE mais alto significa mais erro nas previsões do modelo. Uma desvantagem do RMSE é sua sensibilidade a outliers. Como os erros são quadrados antes de serem médios. o RMSE pode dar maior peso a erros maiores. O RMSE é amplamente utilizado em campos que vão desde a previsão financeira e epidemiologia até o processamento de linguagem natural e visão computacional.

2.2.2 Mean Absolute Scaled Error (MASE)

Conceito e Cálculo: O *Mean Absolute Scaled Error* (MASE) é uma métrica para avaliação de modelos de previsão, introduzida em (HYNDMAN; KOEHLER, 2006). A fórmula para calcular o MASE é dada por:

$$\text{MASE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|d_i - f_i|}{\frac{1}{T-1}\sum_{t=2}^{T}|d_t - d_{t-1}|}$$

Nesta equação, d_i representa o valor real observado, f_i é o valor previsto pelo modelo, n é o número total de observações, e T é o tamanho da série temporal. O denominador é uma média dos erros absolutos de uma previsão "Naive", que basicamente utiliza o valor da observação anterior como previsão para a próxima.

Interpretação e Previsor Naive: Um previsor Naive é o mais simples possível e, em séries temporais, geralmente usa o último valor observado como previsão para o próximo ponto de dados. Se o MASE é menor que 1, isso indica que o modelo de previsão está realizando melhor que este previsor simplista. Se o MASE for maior que 1, isso sugere que o modelo de previsão está performando pior que uma previsão Naive.

Invariância de Escala: Um dos principais benefícios do MASE é sua invariância de escala. Isso significa que o MASE pode ser usado para comparar modelos de previsão que operam em diferentes escalas, algo que muitas outras métricas não conseguem fazer de forma eficaz.

Manipulação de Zeros: O MASE é útil especialmente quando a série temporal contém valores zero ou quando a escala não tem um zero significativo. Diferentemente de métricas como o MAPE (Mean Absolute Percentage Error), que podem resultar em valores indefinidos ou distorcidos, o MASE evita esse problema.

Simetria e Interpretabilidade: O MASE é simétrico, o que significa que penaliza tanto previsões excessivas quanto insuficientes de maneira igual. Além disso, é uma métrica fácil de interpretar. Um valor de MASE acima de 1 indica que um modelo mais simples, como um previsor "Naive", teria tido um desempenho melhor que o modelo em consideração.

Normalidade Assintótica e Teste de Hipótese: As propriedades estatísticas do MASE mostraram-se robustas para o teste de hipóteses, especialmente em previsões de um único passo à frente. Em testes empíricos usando o teste de Diebold-Mariano para comparar duas séries de previsões, o MASE tem sido visto para se aproximar de uma distribuição normal.

2.3 Mercado Acionário

O mercado de ações é um componente crítico do sistema financeiro global, funcionando como um veículo que vincula empresas em busca de capital e investidores buscando retornos financeiros. Este segmento do mercado financeiro desempenha um papel importante na promoção do crescimento econômico e na mobilização de recursos dentro de um país (LEVINE; ZERVOS, 1998).

À medida que a economia cresce, a necessidade de financiamento para empresas em expansão aumenta. Estas empresas, por sua vez, buscam diversas fontes de capital. Podem obter financiamento através de empréstimos bancários (capital de terceiros), geração interna de lucros, reinvestimento desses lucros, ou, mais significativamente, através da emissão de ações no mercado acionário.

No centro deste sistema está o mercado de ações. Em termos simples, uma ação é uma fração do capital de uma empresa. Portanto, ao comprar ações, um investidor torna-se parcialmente proprietário dessa empresa. Cada ação no mercado é identificada por um código conhecido como "ticker". O ticker é uma combinação de letras que representa uma determinada empresa na bolsa de valores. Por exemplo, a Petrobrás é representada pelo ticker "PETR" na B3 (Bolsa Brasileira). O mercado à vista refere-se ao segmento do mercado onde as ações são compradas, vendidas e liquidadas (pagas) em até dois dias úteis após a transação (NETO, 2014).

Após a emissão inicial no mercado primário, as ações são então negociadas entre investidores no mercado secundário. Aqui, os investidores compram e vendem ações uns dos outros, proporcionando liquidez aos ativos. O investidor pode escolher entre negociar em lotes-padrão ou em lotes fracionários, com o último representando quantidades menores de ações. Geralmente, um lote é formado por uma centena de ações.

Dentro do universo das ações, existem diferentes tipos delas, sendo as mais comuns as ações ordinárias e as ações preferenciais. As ações ordinárias dão ao titular o direito de voto nas assembleias gerais da empresa, permitindo-lhes ter uma palavra a dizer na sua gestão. Em contraste, as ações preferenciais normalmente não oferecem direitos de voto, mas proporcionam certos benefícios financeiros, como dividendos preferenciais. No mercado brasileiro, na B3, estas categorias são identificáveis através dos tickers das ações: as ordinárias são marcadas com o número '3' ao final (ex: PETR3 para Petrobras), e as preferenciais, geralmente, com o número '4' (ex: PETR4 para Petrobras).

2.3.1 Análise Fundamentalista e Indicadores Macroeconômicos

A análise fundamentalista é uma estratégia-chave na avaliação de ativos no mercado financeiro. Ela se concentra na análise de fatores econômicos e financeiros internos de uma empresa ou ativo para determinar seu valor real. Em contraste, a análise técnica prioriza tendências de preços e padrões gráficos. Enquanto a análise técnica busca oportunidades de curto prazo com base em flutuações de preço, a análise fundamentalista está mais interessada na saúde financeira e operacional da empresa (GRAHAM, 2006).

Indicadores Fundamentalistas Puros: Os principais indicadores fundamentalistas puros incluem a receita, o lucro líquido, o patrimônio líquido, a dívida e o fluxo de caixa operacional de uma empresa. Por puro, quer-se dizer que não foram compostos entre

si ou com outros indicadores. Eles proporcionam uma visão direta da performance financeira e da saúde econômica da empresa, conforme discutido por Povoa (2012) ao elaborar sobre a importância da avaliação precisa de empresas no mercado financeiro.

- **Receita:** A receita é o montante total de dinheiro gerado pela empresa a partir de suas atividades operacionais principais, como vendas de produtos ou serviços.
- **Lucro Líquido:** O lucro líquido é obtido subtraindo todas as despesas operacionais, juros, impostos e despesas extraordinárias da receita total, representando o desempenho financeiro de uma empresa.
- **Patrimônio Líquido:** O patrimônio líquido é a diferença entre o total de ativos e o total de passivos de uma empresa, representando o valor residual dos ativos após a dedução dos passivos.
- **Dívida:** A dívida total de uma empresa inclui todos os passivos financeiros, como empréstimos e obrigações, que necessitam ser pagos no futuro.
- **Fluxo de Caixa Operacional:** O fluxo de caixa operacional é o dinheiro gerado pelas atividades operacionais da empresa, sendo crucial para entender a capacidade da empresa de gerar caixa suficiente para manter e expandir suas operações.

Indicadores Macroeconômicos: Os indicadores macroeconômicos são estatísticas que refletem a condição econômica de um país ou região, e têm um impacto direto nos mercados financeiros. A análise desses indicadores proporciona aos investidores uma compreensão clara do ambiente econômico, auxiliando na tomada de decisões de investimento. Alguns dos indicadores macroeconômicos mais influentes incluem a Taxa de Inflação, a Taxa de Desemprego, o Produto Interno Bruto (PIB), e a Taxa de Juros. Os indicadores macroeconômicos são essenciais para a análise fundamentalista, pois oferecem uma visão holística do ambiente econômico no qual as empresas operam, influenciando assim seu desempenho financeiro e operacional (DAMODARAN, 2012).

- **Taxa de Inflação:** A taxa de inflação é um indicador do nível de preços de bens e serviços e seu crescimento ao longo do tempo. No Brasil, o Instituto Brasileiro de Geografia e Estatística (IBGE) publica regularmente a taxa de inflação através do Índice Nacional de Preços ao Consumidor Amplo (IPCA) (IBGE, 2023).
- **Taxa de Desemprego:** Representa a porcentagem da força de trabalho que está desempregada e ativamente buscando emprego, sendo um indicador da saúde econômica do país ou região. O Banco Central do Brasil fornece dados sobre a taxa de desemprego como parte de suas estatísticas econômico-financeiras (BRASIL, 2023).

- **Produto Interno Bruto (PIB):** O PIB é a soma de todos os bens e serviços produzidos dentro de um país em um determinado período, indicando o tamanho e a saúde da economia. No Brasil, o Ministério da Economia apresenta a estimativa do crescimento real do PIB, assim como o Banco Central (ECONOMIA, 2023).
- **Taxa de Juros:** Definida pelo banco central, a taxa de juros afeta o custo do crédito para empresas e consumidores, impactando assim os gastos, investimentos e, consequentemente, os mercados financeiros. O Banco Central do Brasil regula e divulga a taxa Selic, que é a taxa de juros básica da economia brasileira (BRASIL, 2023).

2.4 Instituições e Documentos

Algumas instituições e documentos contábeis relevantes, principalmente para extração de dados necessários, são apresentados a seguir.

2.4.1 Comissão de Valores Mobiliários (CVM)

A Comissão de Valores Mobiliários (CVM) é uma entidade autárquica vinculada ao Ministério da Economia do Brasil. Sua função é regular e supervisionar o mercado de valores mobiliários, com o objetivo de assegurar seu funcionamento eficiente, protegendo investidores contra atos ilegais e assegurando a transparência do mercado. Ela também recebe e publica informações referentes as empresas listadas na bolsa brasileira.

2.4.2 Banco Central do Brasil (BACEN)

O Banco Central do Brasil (BACEN) é a principal autoridade monetária do país, responsável pela regulação da quantidade de dinheiro em circulação, taxas de juros e a inflação. O BACEN também supervisiona o Sistema Financeiro Nacional, garantindo sua estabilidade e solidez. Muitas informações macroeconômicas são disponibilizadas por essa instituição.

2.4.3 Documentos Contábeis

- **Informações Trimestrais (ITR):** As Informações Trimestrais (ITR) são relatórios financeiros que empresas de capital aberto devem apresentar trimestralmente à CVM. Estes documentos detalham a performance financeira da empresa no período.
- **Demonstrações Financeiras Padronizadas (DFP):** As Demonstrações Financeiras Padronizadas (DFP) consistem em um conjunto de demonstrações financeiras

anuais. Incluem o balanço patrimonial, a demonstração do resultado e o fluxo de caixa, fornecendo uma análise consolidada do desempenho financeiro anual da empresa.

- **Formulário Cadastral (FCA):** O Formulário Cadastral (FCA) é um documento que contém informações cadastrais da empresa, incluindo identificação, atividade principal e dados dos administradores.

2.4.4 Tipos de Informação Contábil

- **Balanço Patrimonial Ativo (BPA) e Passivo (BPP):** oferecem uma visão da situação financeira da empresa em um determinado momento, mostrando ativos, passivos e patrimônio líquido.
- **A Demonstração do Resultado do Exercício (DRE):** apresenta de forma detalhada os resultados da empresa em um período específico, mostrando a formação do resultado líquido a partir da receita, custos e despesas.
- **Fluxo de Caixa (DFC):** detalham as movimentações financeiras da empresa, incluindo entradas e saídas de caixa, sendo fundamentais para entender a capacidade de geração de caixa da empresa.

2.5 Redes Neurais

Redes neurais são algoritmos de aprendizado de máquina inspirados na estrutura e função dos neurônios no cérebro humano. Eles são especialmente proficientes em tarefas que envolvem grandes volumes de dados e padrões complexos, sendo amplamente utilizados em aplicações de visão computacional, processamento de linguagem natural e previsão de séries temporais.

As redes neurais são compostas por camadas de neurônios interconectados que transmitem informações entre si. Através de processos de treinamento, essas redes "aprendem" a extrair características e padrões dos dados, otimizando suas conexões para produzir saídas desejadas.

2.5.1 Multilayer Perceptron

Um Multilayer Perceptron (MLP) é um tipo de rede neural, composta por múltiplas camadas de neurônios, conforme ilustra a figura 2. Um MLP é composto por três tipos principais de camadas:

Camada de Entrada: Cada neurônio nessa camada representa uma característica (feature) dos dados de entrada. Não realiza nenhum cálculo; apenas transmite as informações adiante.

Camadas Ocultas: Essas camadas estão localizadas entre a camada de entrada e a camada de saída. Os neurônios nas camadas ocultas realizam cálculos intermediários e introduzem a capacidade da rede de aprender representações complexas dos dados.

Camada de Saída: A camada de saída produz as previsões ou classificações finais da rede, dependendo do tipo de tarefa. Cada neurônio na camada de saída representa uma classe ou valor de saída desejado.

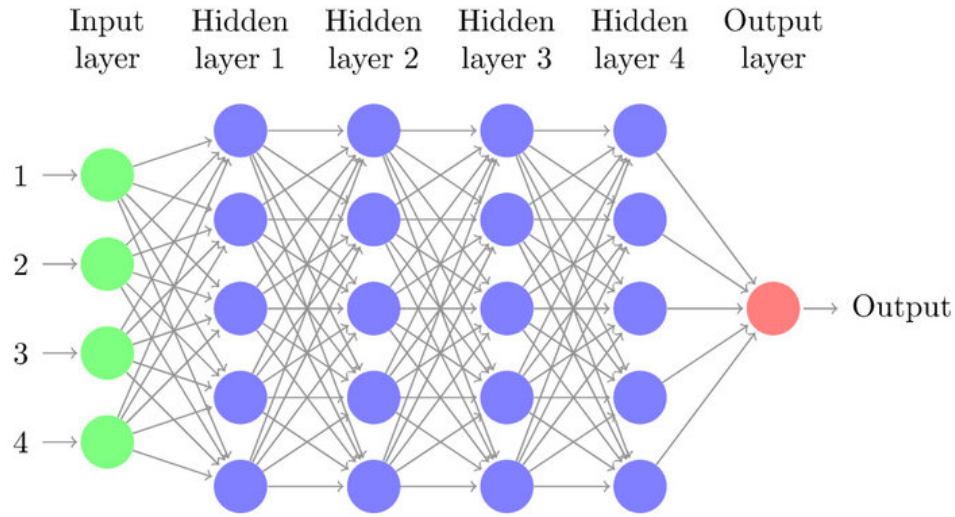


Figura 2 – Diagrama de um MLP, com 4 entradas na camada de entrada, 4 camadas ocultas e uma saída na camada de saída. Retirado de (PéREZ-ENCISO; LAURA, 2019).

O neurônio, também chamado de perceptron, é a unidade básica dentro de um MLP. Cada perceptron realiza duas operações principais, conforme ilustra a figura 3:

Combinação Linear: Calcula uma soma ponderada das entradas (incluindo um termo de viés) usando pesos associados a cada entrada. A fórmula da combinação linear é a seguinte: $z = \sum_{i=1}^{n}(x_i \cdot w_i) + b$, onde z é a soma ponderada das entradas, x_i são as entradas, w_i são os pesos associados às entradas e b é o termo de viés (bias).

Função de Ativação: Após a combinação linear, o valor z é passado por uma função de ativação não linear para produzir a saída do perceptron. Diferentes funções de ativação, como a função sigmoide, a função ReLU (Rectified Linear Unit) ou a função tangente hiperbólica (tanh), podem ser usadas. A escolha da função de ativação depende da tarefa e das características do problema. A saída do neurônio é portanto $y = f(z)$.

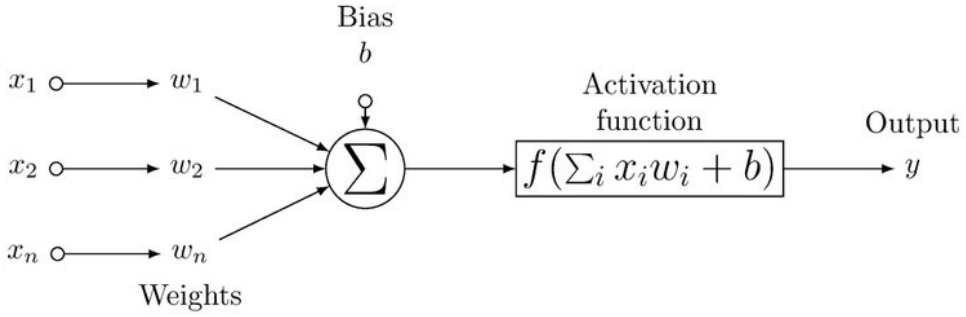


Figura 3 – Representação gráfica dos componentes de um neurônio em um MLP. Retirado de (PéREZ-ENCISO; LAURA, 2019).

O funcionamento de um MLP envolve a propagação direta (*forward propagation*) dos dados e a retropropagação (*backpropagation*) dos erros durante o treinamento.

Propagação Direta: Os dados de entrada são alimentados na rede a partir da camada de entrada. Os valores são propagados pelas camadas ocultas até alcançarem a camada de saída. Cada perceptron realiza suas operações de combinação linear e ativação.

Retropropagação: Durante o treinamento, o erro entre as saídas reais e as saídas desejadas é calculado. Esse erro é então propagado de volta pela rede usando o algoritmo de retropropagação, e os pesos e vieses são ajustados para minimizar o erro. Isso envolve o uso de algoritmos de otimização, como o gradiente descendente, para encontrar os pesos e vieses ideais.

As equações específicas variam com base na função de ativação escolhida, mas a estrutura geral das operações de combinação linear e ativação é comum a todos os perceptrons dentro do MLP. Existem arquiteturas consideravelmente mais complexas, como os Transformers e Echo State Networks (ESN), que usam estruturas mais elaboradas do que apenas as camadas de perceptrons, e são analisadas com mais detalhes nas próximas seções.

Uma vez que a rede está treinada por meio do método de propagação direta e retropropagação, ela pode receber novas entradas e devolver a saída, que pode ser a resposta de uma classificação ou regressão, em um problema de processamento de linguagem natural, visão computacional e muito mais. Quando há múltiplas camadas ocultas, a rede é capaz de aprender representações complexas de dados, conseguindo distinguir entre conjuntos de dados que não são linearmente separáveis e tornando-se uma parte fundamental do campo de redes neurais profundas (*deep learning*).

2.5.2 RNN (*Recurrent Neural Networks*)

Redes Neurais Recorrentes (RNNs) são uma classe de redes neurais que possuem conexões recorrentes, tornando-as adequadas para lidar com dados sequenciais. As RNNs podem capturar dependências temporais de longo prazo e são geralmente utilizadas em tarefas que requerem a compreensão de sequências temporais (GOODFELLOW YOSHUA BENGIO, 2016). Um desafio comum em RNNs é o problema do desaparecimento do gradiente, onde a rede tem dificuldade em aprender dependências de longo alcance. Várias arquiteturas, como *Long Short-Term Memory* (LSTM) e *Gated Recurrent Units* (GRU), foram propostas para mitigar este problema.

2.6 *Echo State Networks* (ESN) e *Leaky Integrator* ESN (LiESN)

Echo State Networks (ESNs) são uma subclassificação de Redes Neurais Recorrentes (RNNs), que se destacam pela capacidade de lidar com tarefas relacionadas a séries temporais. As ESNs possuem uma camada de reservatório com conexões dinâmicas que não são treinadas, simplificando o processo de treinamento e ajudando a capturar dinâmicas temporais complexas (JAEGER, 2001). O algoritmo de *Leaky Integrator Echo State Networks* (LiESN) é uma variante do modelo de rede neural recorrente ESN. O LiESN é projetado para resolver problemas de classificação e previsão em séries temporais.

O modelo ESN é composto por uma camada de entrada, uma camada oculta (também conhecida como reservatório) e uma camada de saída, conforme ilustra a figura 4. A camada de entrada recebe os dados de entrada, que são alimentados na camada oculta. A camada oculta é composta por um grande número de neurônios que são interconectados, formando um grafo de aleatoriedade estruturada. Os pesos das conexões entre os neurônios são gerados aleatoriamente e permanecem fixos durante o treinamento. A camada de saída é responsável por produzir as saídas previstas com base nos padrões aprendidos na camada oculta.

O LiESN inclui uma integração “leaky” (vazante) na camada de reservatório, permitindo maior capacidade de modelagem para sistemas temporais lentos ou séries temporais com diferentes escalas de tempo (LUKOŠEVIČIUS, 2012). Esse mecanismo permite que a informação armazenada nos neurônios da camada oculta se degrade com o tempo, permitindo que o modelo se adapte a mudanças nas entradas ao longo do tempo. O mecanismo de vazamento é controlado por um parâmetro chamado “taxa de vazamento” (*leak rate*), que determina a velocidade com que a informação se degrada.

O modelo LiESN é treinado usando um algoritmo de aprendizagem supervisionado, como a regressão linear ou a rede neural de camada única. Durante o treinamento, o modelo ajusta os pesos das conexões entre a camada oculta e a camada de saída, usando os dados de entrada e saída correspondentes.

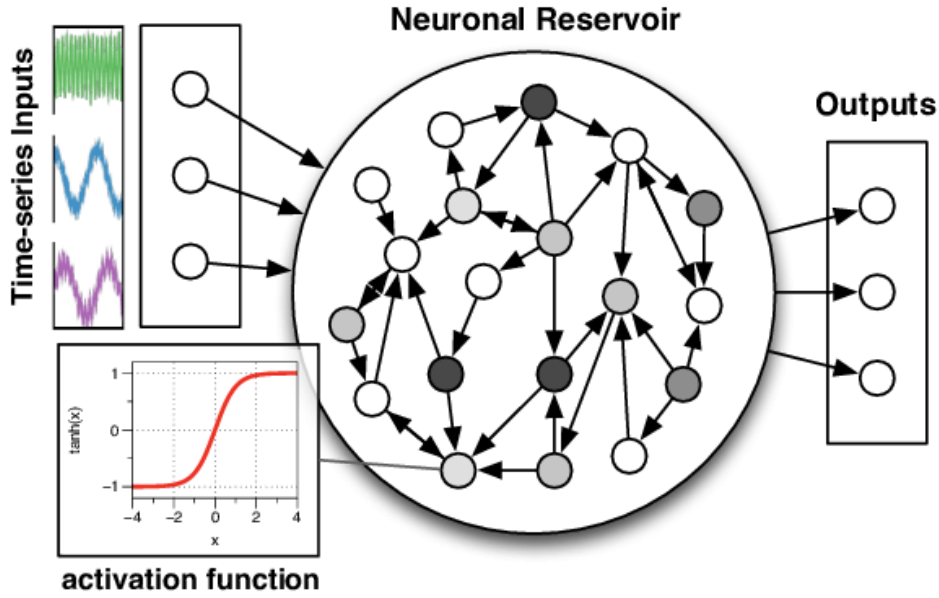


Figura 4 – Arquitetura de uma ESN. Retirada de (SOH; DEMIRIS, 2014).

A estrutura de uma LiESN segue a mesma estrutura básica de uma ESN, mas com uma modificação na camada de reservatório. As partes principais são:

Camada de Entrada: Recebe os dados de entrada e os transmite para a camada de reservatório.

Camada de Reservatório: Composta por neurônios com conexões recorrentes e uma integração "leaky", é responsável por manter um "eco" das entradas passadas, permitindo capturar dinâmicas temporais em várias escalas de tempo.

Camada de Saída: Utiliza o estado da camada de reservatório para gerar a saída da rede.

Sua dinâmica pode ser descrita pelas seguintes equações:

1. **Atualização da Camada de Reservatório:**

$$x(t) = (1 - \alpha)x(t-1) + \alpha f(W_{\text{in}}u(t) + Wx(t-1)) \tag{2.1}$$

onde α é o parâmetro de vazamento (leakage), $x(t)$ é o estado da camada de reservatório no tempo t, $u(t)$ é o vetor de entrada no tempo t, W_{in} e W são as matrizes de pesos da camada de entrada e da camada de reservatório, respectivamente, f é a função de ativação (geralmente a função tangente hiperbólica).

2. **Cálculo da Saída:**

$$y(t) = W_{\text{out}}x(t) \tag{2.2}$$

onde $y(t)$ é o vetor de saída no tempo t, W_{out} é a matriz de pesos da camada de saída.

No contexto de aprendizado de máquina, as LiESNs são valiosas para tarefas que exigem a captura de dependências temporais complexas sem a necessidade de um treinamento intensivo de parâmetros, especialmente em séries temporais com diferentes escalas de tempo. A adição do termo de vazamento permite uma melhor modelagem de dinâmicas lentas, o que pode ser útil em várias aplicações práticas, como a previsão de séries temporais em mercados financeiros.

Em suma, LiESN é uma estrutura robusta e eficaz que tem sido aplicado com sucesso em uma variedade de problemas de classificação e previsão em séries temporais, incluindo previsão de tráfego, previsão de demanda de energia, previsão de preços de ações, detecção de falhas em equipamentos, entre outros.

2.7 Processamento de Linguagem Natural (PLN)

O processamento de linguagem natural (PLN) é uma área especializada no desenvolvimento de técnicas e algoritmos que lidam com dados de linguagem natural não estruturados, seja como entrada ou saída. A linguagem humana é extremamente ambígua (considere, por exemplo, a diferença entre as frases “Comi pizza com amigos” e “Comi pizza com azeitonas”) (GOLDBERG, 2017).

2.7.1 Características Únicas da Linguagem Humana em IA

A linguagem humana tem características distintas que a diferenciam de outros domínios de dados. Primeiramente, ela é discreta e simbólica. A linguagem é construída a partir de elementos básicos, caracteres, que por sua vez formam palavras. Essas palavras denotam vários objetos, conceitos, ações e ideias. Em contraste com o processamento de imagens, onde a tonalidade e intensidade de uma cor podem ser facilmente modificadas de forma contínua ao alterar valores RGB em um espectro, as palavras não podem ser alteradas de forma tão direta sem mudar seu significado completamente.

Em segundo lugar, a linguagem humana é composicional. Caracteres formam palavras, palavras formam frases e frases formam parágrafos. O significado de uma frase frequentemente transcende os significados das palavras individuais que a compõem. Portanto, o contexto textual deve ser considerado em vez de tratar cada palavra isoladamente.

Em terceiro lugar, esses atributos contribuem para a dispersão da linguagem. As combinações possíveis de palavras em qualquer idioma dado são quase infinitas. Isso apresenta um desafio para o aprendizado supervisionado; mesmo com um grande conjunto

de dados, é provável que um algoritmo encontre frases que nunca viu antes, diferindo consideravelmente daquelas em que foi treinado.

2.8 Representação vetorial de palavras

A representação vetorial de palavras, também conhecida como "embedding" de palavras, refere-se à prática de converter palavras ou frases em vetores de números. Essas representações vetoriais, geralmente, são criadas de tal forma que palavras semanticamente similares estão próximas no espaço vetorial, conforme ilustra a figura 5. Isso torna a representação vetorial de palavras e frases particularmente úteis em aplicações de PLN ao capturar numericamente as semelhanças semânticas entre as palavras.

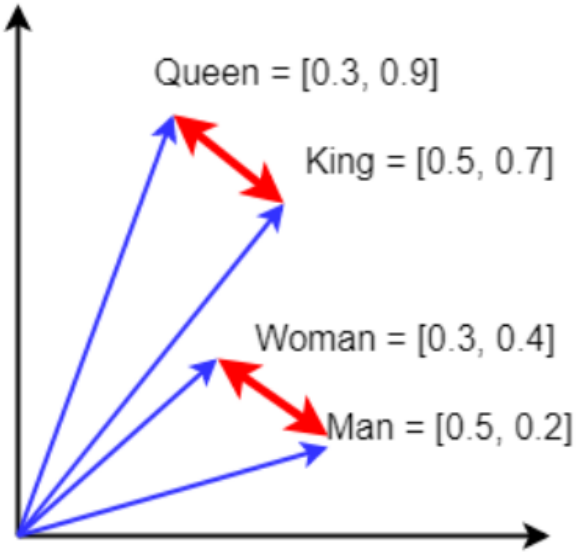


Figura 5 – Exemplo de vetores para as palavras "King", "Queen", "Woman" e "Man". Nota-se neste exemplo que King - Man + Woman = Queen. Retirada de (SUTOR et al., 2019).

Existem duas categorias distintas de algoritmos de vetorização (BARONI; DINU; KRUSZEWSKI, 2014):

Modelos de contagem (*Distributional Semantic Models*): Utilizam vetores como meio de identificar o contexto, através do uso de ferramentas que visam identificar semelhanças geométricas entre os vetores. Em outras palavras, estes modelos partem do intuito que palavras similares ocorrem em contextos similares, e ao construir uma matriz de ocorrência das palavras com dimensão grande o suficiente, espera-se que seja possível identificar essa semelhança entre palavras.

Modelos de Previsão: Trazem uma forma invertida de solucionar o problema de representar as palavras. Ao invés de construir um vetor para depois encontrar a semelhança

entre palavras, treina-se um modelo preditivo utilizando inteligência artificial, que busca prever quais palavras aparecem em um dado contexto, e atribui vetores a cada palavra de forma a maximizar a probabilidade de um contexto. O intuito é que no final do treinamento, palavras com significados semelhantes possuam vetores semelhantes também.

A similaridade por cosseno é uma medida comum usada para avaliar a similaridade entre dois vetores numéricos em um espaço multidimensional. É amplamente aplicada em tarefas de mineração de texto, recuperação de informações e PLN, especialmente quando se trata de comparar documentos ou palavras em um contexto semântico.

Ela é baseada no conceito de ângulo entre dois vetores em um espaço vetorial: quanto menor o ângulo entre dois vetores, maior é a similaridade entre eles. A fórmula para calcular a similaridade por cosseno entre dois vetores A e B vem da definição do produto interno:

$$\cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||}, \tag{2.3}$$

onde A é o vetor que representa uma palavra ou frase e B um outro vetor que representa uma palavra ou frase.

2.8.1 Técnicas de Vetorização de Texto

Word2Vec: é uma técnica específica de vetorização que utiliza redes neurais para produzir representações vetoriais de palavras (SEBASTIAN; ISA, 2020). Foi desenvolvida por Mikolov et al. e tornou-se uma das abordagens mais populares para a geração de *embeddings* após ter um modelo da Google treinado em 100 bilhões de palavras em inglês liberado ao público. O algoritmo propõe duas formas principais para gerar *embeddings*: o *Skip-Gram* e o *Continuous Bag of Words* (CBOW) (LING et al., 2015). O primeiro prevê as palavras do contexto a partir de uma palavra alvo, enquanto o segundo prevê uma palavra alvo com base nas palavras de contexto. Em ambos os casos, a ordem das palavras do contexto não importa. A figura 6 ilustra o funcionamento do CBOW.

Wang2Vec: Wang2Vec é um modelo de *embedding* de palavras que, assim como Word2Vec, foi criado para gerar representações vetoriais de palavras ricas em contexto semântico. Desenvolvido por Ling et al. (2015), este modelo é baseado no Word2Vec com uma particularidade: ele leva em consideração a ordem das palavras presentes no contexto. Ele também apresenta duas formas de gerar o *embedding*: *Structured Skip-gram* e *Continuous Bag-Of-Words*.

GloVe: que significa “Global Vectors for Word Representation”, é um modelo que gera representações vetoriais ao levar em conta as estatísticas globais de co-ocorrência

de palavras no corpus. Isto significa que, para um conjunto de textos grande, o custo computacional de treinamento do GloVe é relativamente alto (PENNINGTON; SOCHER; MANNING, 2014), uma vez que é necessário carregar um dicionário com todas as palavras do corpus e sua frequência. A figura 7 ilustra esta técnica.

FastText: (BOJANOWSKI et al., 2016) propõe um algorítmo que diferencia-se por considerar não a palavra como unidade básica, mas sim n-gramas de caracteres que constroem palavras, permitindo a geração de *embeddings* para palavras fora do vocabulário e lidando melhor com problemas de palavras mal-escritas ou em diferentes formas morfológicas.

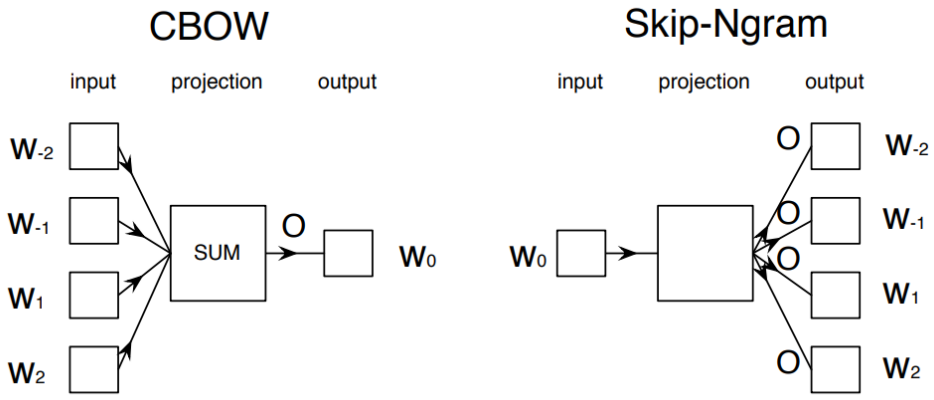


Figura 6 – Exemplo do funcionamento do CBOW e do Skip-Gram. O contexto é composto pelas palavras W-2, W-1, W+1 e W+2 ao redor da palavra W0. Retirado de (LING et al., 2015).

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Figura 7 – Os autores do Glove exemplificam o uso do algoritmo para analisar semelhanças entre palavras do inglês. “Gelo” ocorre muito mais frequentemente com “Sólido” do que “Gasoso”, enquanto o inverso é verdadeiro para “Vapor”. Retirado de (PENNINGTON; SOCHER; MANNING, 2014).

2.8.2 Classificação de Texto

A classificação de texto é o processo de categorizar texto em grupos organizados (GUARINO, 1995). No contexto financeiro, pode referir-se à categorização de notícias com base em seu conteúdo e potencial impacto. Por exemplo, uma notícia pode ser classificada

como tendo relação com a Petrobrás, ou classificada como uma notícia política. É de interesse particular as técnicas de categorização de textos no domínio de *word embeddings* (ALYOUSEF, 2021).

2.8.3 TF-IDF

O algoritmo TF-IDF (*Term Frequency - Inverse Document Frequency*) é uma técnica de PLN amplamente utilizada para avaliar a importância relativa de palavras ou termos em um documento dentro de um conjunto de documentos. Esse método é particularmente útil na indexação e recuperação de informações, e é uma ferramenta básica em sistemas modernos de busca e classificação de documentos (JIANG et al., 2021). Ele é composto das seguintes etapas:

TF (*Term Frequency*): mede a frequência de um termo específico em um documento:

$$TF(t,d) = \frac{\text{Número de vezes que o termo } t \text{ aparece no documento } d}{\text{Número total de termos no documento } d}. \tag{2.4}$$

Isso significa que ele indica a proporção de vezes que um termo específico aparece em relação ao tamanho do documento. Quanto mais vezes um termo aparece em um documento, maior será o valor de TF para esse termo nesse documento.

IDF (*Inverse Document Frequency*): mede a importância relativa de um termo em relação a todo o conjunto de documentos:

$$IDF(t) = \ln\left(\frac{\text{Número total de documentos}}{\text{Número de documentos contendo o termo } t}\right). \tag{2.5}$$

O IDF penaliza termos que aparecem em muitos documentos e dá mais importância a termos raros que são específicos de alguns documentos. Termos que aparecem em muitos documentos têm um IDF mais baixo, enquanto termos raros têm um IDF mais alto.

TF-IDF combinado: o TF-IDF combina as informações de frequência de termo (TF) e frequência inversa de documento (IDF) para calcular uma pontuação que reflete a importância de um termo em um documento específico em relação ao conjunto de documentos:

$$\text{TF-IDF}(t,d) = TF(t,d) \times IDF(t). \tag{2.6}$$

Isso significa que um alto valor de TF-IDF indica que um termo é importante em um documento específico, mas não é muito comum em outros documentos do conjunto. Um baixo valor de TF-IDF indica que o termo é comum em muitos documentos ou não é muito importante para o documento específico.

No caso do presente trabalho, os documentos em questão são os textos das notícias. Para cada um deles, será gerado um vetor com todas as palavras contidas na notícia e sua respectiva pontuação TF-IDF.

Para descobrir as notícias mais relevantes para uma empresa, fazemos uma *query*: escrevemos um texto simples contendo somente o nome da empresa, como "Petrobras", "Vale" ou "Itaú" e computamos o TF-IDF dele. Agora, para comparar esse vetor gerado com os das notícias, usamos a propriedade da similaridade por cosseno.

2.8.4 Busca simples

A busca simples refere-se ao processo de procurar termos ou palavras específicas em um texto. É uma das técnicas mais fundamentais e amplamente utilizadas em sistemas de recuperação de informações (ESEN, 2022).

Enquanto métodos mais avançados, como *embeddings* ou TF-IDF, capturam nuances e contextos, a busca simples oferece uma maneira rápida e direta de identificar a presença ou ausência de termos específicos.

No contexto deste trabalho, a busca simples pode ser usada para rastrear a menção de termos específicos, como nomes de empresas, produtos ou nome de setores, fornecendo um método rápido para filtrar e categorizar notícias.

2.9 Web scraping

Web scraping é o processo de coletar automaticamente informações ou dados de sites da web. Isso é feito por meio de programas de computador (ou bots) que acessam páginas da web, analisam seu conteúdo e extraem dados específicos de interesse. Os dados coletados podem incluir texto, imagens, links, tabelas, informações de produtos, preços, notícias, comentários de usuários e muito mais.

Muitas vezes, a principal finalidade do web scraping é coletar dados para fins de análise, pesquisa ou monitoramento. Isso pode incluir a coleta de dados de preços de produtos para comparar preços de diferentes lojas, coleta de dados de mídias sociais para análise de sentimentos, ou coleta de informações de mercado para tomar decisões de negócios informadas. Neste caso, o web scraping está sendo utilizado para coletar de sites jornalísticos conteúdo referente a empresas listadas na bolsa para análise e indexação.

O web scraping é bastante facilitado se o site expõe uma API pública que permite fazer pedidos HTTP já padronizados para acessar as informações do site, como preços, estatísticas, e outros dados. Infelizmente poucos sites possuem tal recurso, o que significa que para extrair informações deles é necessário usar um programa que acessa as páginas dos sites, interage com os elementos presentes nelas e baixa o arquivo HTML da página

que pode então ser analisado para encontrar dados como título, data, autor, corpo da notícia, etc.

2.10 Análise de Sentimento

A análise de sentimentos é um ramo da ciência da computação e da inteligência artificial que busca determinar o tom emocional por trás de uma série de palavras. Este campo tem ganhado relevância, principalmente no contexto de análise de opiniões em redes sociais, reviews de produtos e, como neste trabalho, no âmbito das notícias relacionadas ao mercado financeiro (YADAV et al., 2020; CAMBRIA; BROWN, 2016). O impacto das notícias no mercado de ações é inegável. Uma notícia negativa pode derrubar o preço de uma ação, enquanto notícias positivas podem impulsionar seu valor. A análise automática e em tempo real de sentimentos de notícias pode, portanto, oferecer insights valiosos para previsões mais precisas (ARRATIA et al., 2021).

Segundo (MEHTA; PANDYA, 2020), as técnicas para análise de sentimento podem ser divididas em duas grandes áreas: as baseadas em léxico e as baseadas em machine learning. As técnicas baseadas em léxico geralmente consistem em usar um grande dicionário que contém as palavras que podem aparecer em um texto e sua respectiva pontuação. Por exemplo, as palavras são previamente atribuídas um score de -1 a +1 onde -1 é totalmente negativo e +1 totalmente positivo, e o sentimento do texto completo é a soma das pontuações das palavras nele. Também deve-se levar em conta conectivos que invertem o significado da palavra seguinte, como "não", "anti", etc.

Neste projeto, entretanto, a técnica a ser usada é aquela baseada em machine learning, que consiste em usar modelos de treinamento supervisionado ou não para prever o sentimento do texto, como Naive Bayes, árvores de decisão e redes neurais. O modelo usado aqui é o FinBERT PT-BR, desenvolvido por (SANTOS, 2022), que é baseado na família de modelos de linguagem natural BERT desenvolvida pelo Google, muito conhecidos por representarem o estado da arte neste campo de pesquisa. O FinBERT especializa-se em calcular o sentimento de notícias do mercado financeiro brasileiro, que é o alvo deste projeto.

3 Revisão da Literatura

Este capítulo de revisão da literatura apresenta uma análise dos estudos e pesquisas relevantes ao tema de previsão de preços de ações utilizando técnicas de aprendizado de máquina.

Aqui estão descritos brevemente alguns projetos que inspiraram a realização deste trabalho e que também estão relacionados ao tema geral de previsão de séries temporais usando Machine Learning e suas aplicações, e a metodologia por trás dos estudos realizados.

3.1 Metodologia da Revisão de Literatura

A pesquisa bibliográfica foi realizada com o objetivo de fundamentar teoricamente o grupo sobre a aplicação de técnicas modernas de inteligência artificial nos tópicos relevantes para o projeto, como análise de sentimento e previsão de séries temporais.

Foi utilizado o *Google Scholar* como ferramenta de busca para a seleção de artigos e jornais científicos. As palavras chaves utilizadas incluíram termos como *'Echo State Networks'*, 'Analise de Sentimento', 'Previsão de Séries Temporais em Mercados Financeiros', 'Previsão de Séries Temporais' na lingua inglesa. Os trabalhos encontrados por este método são em inglês.

As reuniões de acompanhamento com a orientadora foram essenciais para o direcionamento da pesquisa bibliográfica, onde foram sugeridas referências de estudos anteriores realizados sob sua orientação, bem como trabalhos de destaque na área. Este processo de orientação acadêmica garantiu que o referencial teórico estivesse alinhado com as práticas e descobertas mais recentes em inteligência artificial e finanças. Os trabalhos selecionados para análise detalhada foram escolhidos com base em sua relevância para os métodos utilizados e o contexto de aplicação deste estudo.

A revisão de literatura incluiu uma análise de artigos que exploraram as redes neurais recorrentes, especificamente Echo State Networks (ESNs), no contexto da previsão de séries temporais. Estudos foram revistos para identificar abordagens e inovações no uso de ESNs e suas variantes em diferentes domínios de aplicação. Para complementar a pesquisa e esclarecer aspectos técnicos, recorreu-se a blogs especializados em tecnologia e inteligência artificial, como GeeksForGeeks, Medium e TowardsDataScience, que foram consultados para entender melhor a implementação prática das técnicas estudadas.

3.2 Trabalhos relevantes

A pesquisa em PLN é um campo vasto e complexo, onde técnicas de machine learning e modelos de atenção são amplamente aplicados. No trabalho realizado por Santos (2022), foi desenvolvido um modelo de análise de sentimentos em textos em português relacionados ao mercado financeiro. O autor utilizou a arquitetura de redes neurais BERT com o objetivo de auxiliar o processo de tomada de decisões no mercado financeiro através da inteligência artificial. Santos treinou o modelo em duas etapas: modelagem de linguagem e modelagem de sentimentos. O modelo de linguagem foi treinado com mais de 1,4 milhão de textos de notícias financeiras em português, e a partir desse treinamento, foi possível construir um classificador de sentimentos com poucos textos rotulados, obtendo resultados satisfatórios.

O modelo proposto por Santos obteve resultados superiores aos modelos atuais no estado da arte e pode ser utilizado para construção de índices de sentimento, estratégias de investimento e análise de dados macroeconômicos. Para validar dados e modelos, o autor seguiu uma metodologia de anotação manual e garantiu a qualidade dos modelos de aprendizado de máquina. Em relação ao modelo de previsão de séries temporais de preços de ações na bolsa brasileira, o trabalho de Santos pode contribuir significativamente, visto que a análise de sentimentos e o índice de sentimentos propostos em seu estudo podem ser incorporados como variáveis adicionais no modelo de previsão, melhorando a precisão das previsões.

Na dissertação de mestrado de Paiva (PAIVA, 2023), o autor desenvolveu um modelo chamado *Sentiment-Aware Reinforcement Learning Intelligent Trading System (ITS-SentARL)* que se baseia nas técnicas de aprendizado por reforço e análise de sentimentos para fazer recomendações de investimento voltadas ao mercado americano, estudando empresas como Apple, Amazon, Boeing, entre outras. Ele utiliza o histórico do preço da ação como entrada do seu modelo junto com a análise de sentimentos de notícias financeiras em inglês, extraídas do Wall Street Journal por exemplo. Diferentemente deste trabalho, que busca prever os valores futuros da série temporal do preço de uma ação, o modelo de Lima retorna apenas se o preço da ação vai subir, cair ou se manter igual.

Para fazer o aprendizado de máquina ele usa um algoritmo Advantage Actor-Critic (A2C), e compara três modelos para ver qual tem a melhor eficiência: o seu modelo A2C com análise de sentimentos, A2C sem análise de sentimentos e uma estratégia tradicional Buy and Hold (BH), onde não é feita mais nenhuma operação depois depois da compra do ativo. Ele chegou à conclusão de que o seu modelo é o mais rentável, fornecendo constantemente taxas de retorno do investimento mais altas que as outras alternativas.

Já o artigo (BARROS et al., 2023) desenvolve uma abordagem inovadora para tratar séries multivariadas caracterizadas por irregularidades nos dados, principalmente em

casos como taxas de amostragem variáveis ou dados ausentes. O interesse neste trabalho está ligado à implementação de uma arquitetura codificador-decodificador utilizado, no caso a inovação chave é o uso RNNs para processar cada série temporal de entrada em uma representação de tamanho fixo, que pode posteriormente servir de entrada para outra rede neural. Uma das características das entradas no trabalho é a presença de séries que possuem tamanhos diferentes. Alguns dias podem apresentar centenas de notícias, enquanto outros apenas algumas.

No contexto de previsão de séries temporais, a dimensão do tempo é um componente crítico. Os autores do Time2Vec (KAZEMI et al., 2019) apresentam uma abordagem diferente para realizar a codificação do tempo que servirá de entrada para o modelo. Ao invés de utilizar métodos tradicionais de codificação temporal, em que o projetista do sistema manualmente escolhe as frequências temporais e como deve ser feito a extração de informação da dimensão temporal, o Time2Vec propõe a utilização de parâmetros treináveis com aprendizado de máquina para realizar a codificação do tempo de forma automática. Essa codificação é útil para a identificação de frequências temporais onde emergem padrões importantes para o sistema. Resta ao projetista escolher apenas o tamanho do vetor que representará o tempo.

Taehwan Kim e Brian R. King (KIM; KING, 2020) propuseram uma arquitetura de Deep Echo State Network para lidar com a previsão de séries temporais. O trabalho reconhece que as redes neurais artificiais têm sido utilizadas para modelagem e previsão de séries temporais em diversos domínios, mas muitas vezes encontram limitações ao lidar com dados não-lineares e caóticos.

O artigo (BAI et al., 2023) propõe um método avançado de previsão com uma estrutura de rede neural profunda e algoritmo de treinamento sem retropropagação, mostrando-se também promissor para lidar com séries temporais não estacionárias.

Esses trabalhos inspiraram a proposta atual, descrita nos próximos capítulos.

4 Especificação: FinStockESN-Br

O sistema, denominado FinStockESN-Br, é uma aplicação de previsão de preços de ações baseada em Leaky-integrator Echo State Networks (LiESN) para o mercado financeiro brasileiro. Seu objetivo é investigar quais tipos de dados de entrada podem melhor ajudar na previsão das séries temporais de preços das ações listadas no mercado brasileiro. Além disso, oferece uma interface para o acompanhamento das ações no mercado. A arquitetura do sistema combina os seguintes tipos de dados separados por ação:

- Preço das ações
- Dados de análise fundamentalista como mencionado em 2.3.1
- Indicadores macroeconômicos como também mencionado em 2.3.1
- Análise de sentimento provenientes de notícias textuais relacionadas ao mercado financeiro

O FinStockESN-Br é é estruturado em três módulos principais e uma aplicação de acompanhamento:

Módulo de coleta de dados: responsável pela coleta de todos os tipos de dados relevantes.

Módulo de coleta de notícias e análise de sentimentos: responsável pela coleta de notícias e utilização do modelo FinBERT pt-br para análise de sentimento, já desenvolvido em um trabalho anterior por Santos (2022).

Módulo de previsão: responsável pelo desenvolvimento do modelo LiESN, agregando o índice de sentimentos por ação e comparando os resultados com diferentes tipos de entrada.

Aplicação de acompanhamento: por sua vez, permite o monitoramento das notícias diárias de uma determinada ação e a visualização da previsão de seu preço, com base no modelo desenvolvido.

4.1 Requisitos

Aqui serão apresentados os requisitos funcionais e não-funcionais do sistema.

4.1.1 Requisitos Funcionais

Os seguintes requisitos funcionais deverão ser atendidos:

Coleta de dados Será necessário a coleta e o processamento de dados de diferentes fontes, como análise fundamentalista, indicadores macroeconômicos e notícias textuais relacionadas ao mercado financeiro e as empresas.

Análise de sentimento O sistema deve incorporar o índice de sentimento FinBERT pt-br para analisar o sentimento das notícias textuais relacionadas às ações específicas.

Previsão de séries temporais O sistema deve implementar um modelo de previsão com base em Leaky integrator Echo State Networks (LiESN) para prever os preços das ações

Interface de usuário O sistema deve fornecer uma interface de usuário que permita a visualização das notícias diárias relacionadas a uma ação específica e a previsão do preço da ação.

4.1.2 Requisitos Não-funcionais

São inclusos também os seguintes requisitos não funcionais:

Múltiplos tipos de dados O estudo deve processar e analisar os diferentes tipos de dados e é fundamental que seja criada uma estrutura consistente com a entrada do modelo de previsão ESN.

Experiência do usuário intuitiva O foco do trabalho não é desenvolver uma interface web sofisticada, entretanto, a interface de usuário deve ser intuitiva e fácil de usar, permitindo que os usuários acessem e compreendam as informações apresentadas com facilidade.

4.2 Análise de Sentimentos BERT

A primeira parte do trabalho consistirá da criação de um módulo capaz de processar notícias do mercado financeiro e avaliar se elas são positivas, negativas ou neutras com respeito a uma determinada empresa. Sabe-se bem que o mercado financeiro em geral costuma ser bastante responsivo e volátil no curto prazo quando reage a notícias sobre uma das empresas que é negociada na bolsa. Rumores de escândalos, resultados ruins ou novas aquisições são suficientes para fazer o preço de uma ação cair ou subir consideravelmente, oferecendo oportunidades a investidores.

Assim, a proposta dessa parte do trabalho é automatizar a coleta de notícias relevantes do mercado financeiro, avaliar se são boas ou ruins, e a qual empresa dizem respeito. A principal tecnologia de machine learning (ML) a ser usada nessa etapa é o processamento de linguagem natural (NLP), um campo da ciência da computação cujo foco é desenvolver programas capazes de analisar a linguagem humana e responder da mesma forma que uma pessoa faria. Nesse caso o foco do projeto é em análise de sentimentos, ou seja, descobrir se o texto está falando bem, mal, ou é neutro com respeito a um determinado assunto.

Entretanto, tão importante como definir se a notícia é boa ou ruim é detectar a qual ação registrada na bolsa ela está se referindo. Isso também pode ser feito por meio de processamento de linguagem natural, ao buscar referências a empresas da bolsa no texto da notícia. Com o ticker da ação (o código que a representa na bolsa, como PETR4, para Petrobras Ação Preferencial) e o dado sobre o sentimento da notícia, é possível fazer uma previsão sobre como o preço da ação se comportará. Esses dados sobre o sentimento da notícia servirão como entrada em outro algoritmo que fará a precificação.

4.3 Arquitetura do FinStockESN-BR

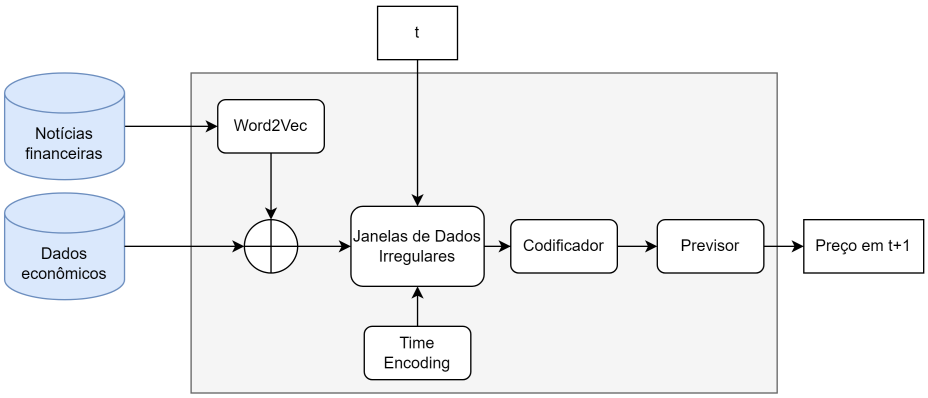


Figura 8 – Arquitetura do Modelo de caixa preta do FinStockESN-BR.

O modelo FinStockESN-BR pode ser conceitualizado como um sistema operacional em regime de 'caixa preta', caracterizado por propriedades distintas:

1. O sistema é projetado para processar um conjunto abrangente de dados de entrada correspondentes a um instante temporal específico, denotado por t, e, com base nesses dados, elabora uma projeção para o instante subsequente, $t + 1$ dia.

2. Dado o instante t, são geradas janelas de dados contendo os dados em um período dos últimos 7 dias para cada série temporal.

3. Uma codificação temporal dos dados é realizada através do algoritmo Time2Vec.
4. Os dados passam por um codificador, que codifica as múltiplas séries temporais de tamanhos irregulares em uma saída de tamanho fixo e interpretável pelo modelo previsor.
5. O componente previsor prevê o valor da ação no período $t + 1$.

Cada um destes componentes será melhor explicado nas seções seguintes.

4.3.1 Gerador de Janelas de Dados

Este módulo processa pontos de dados de diversas origens (como sentimentos, IPCA, volume de negociações, preço de ações, entre outros). Cada ponto de dado é definido por uma dupla: um valor numérico representando a informação em si e um marcador temporal indicando o momento de origem do dado. No caso de dados de sentimentos, estes apresentam uma granularidade temporal até o segundo, vinculada ao momento de publicação da notícia que gerou o sentimento. Para os demais tipos de dados (como volume, preço da ação, etc.), a precisão temporal é diária, marcando a data de publicação da informação.

O papel principal do Gerador de Janelas é selecionar, de acordo com um determinado momento no tempo (representado pelo valor t), todos os pontos de dados disponíveis no intervalo entre t-7 dias e t na base de dados. Após essa seleção, os dados são encaminhados para o codificador, passando antes por um processo de codificação temporal.

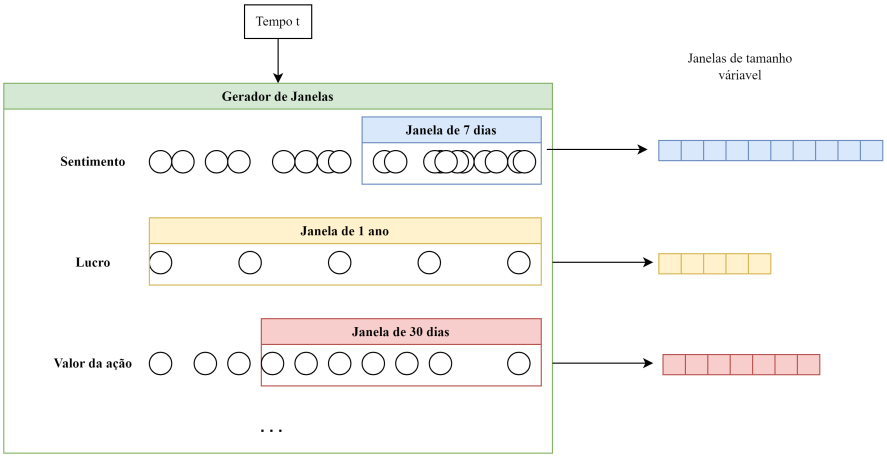


Figura 9 – Esquema do Gerador de Janelas no modelo FinStockESN-BR.

4.3.2 Método de Codificação Temporal: Time2Vec

Neste sistema, a codificação temporal é realizada através do uso do Time2Vec, uma técnica avançada para decomposição em frequência dos tempos associados a cada ponto de dado. A ideia central do Time2Vec é possibilitar ao modelo a identificação e captura de padrões temporais recorrentes, tais como ciclos diários, semanais, anuais, entre outros. O Time2Vec atua de forma automatizada para realizar essa tarefa complexa.

Um aspecto crucial no uso do Time2Vec é a definição do hiperparâmetro "L", que representa a dimensão escolhida para a execução da codificação temporal. Este parâmetro é ajustável e influencia diretamente a performance do modelo.

Durante o processo de codificação, o codificador de tempo recebe uma janela de dados temporais e aplica o Time2Vec. Como resultado, cada ponto dentro desta janela é transformado, sendo representado por um valor numérico juntamente com um vetor de dimensão "L". Este processo enriquece os dados com uma camada adicional de informação temporal, antes de enviar os resultados para o codificador e previsor.

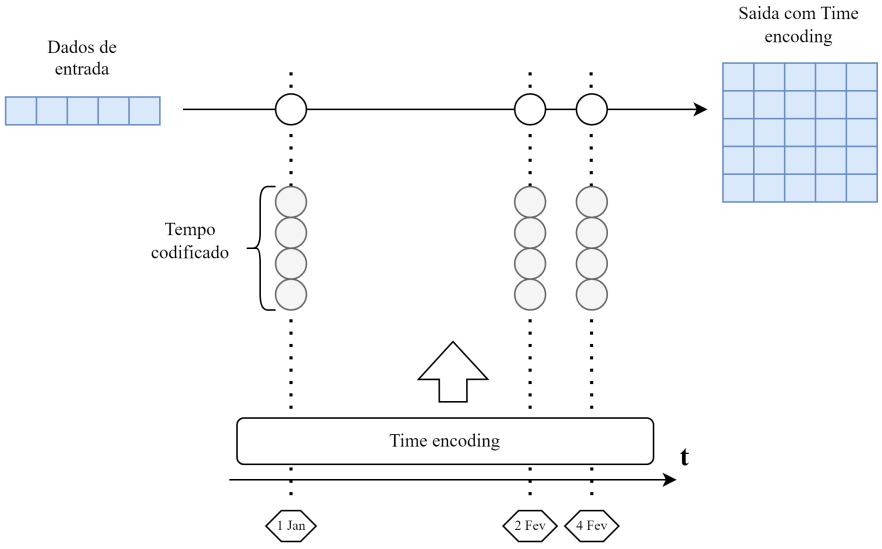


Figura 10 – Representação do Time2Vec utilizado no modelo FinStockESN-BR. Adaptado de (BARROS et al., 2023)

4.3.3 Codificador e Previsor

Este trabalho enfoca a geração de janelas temporais de tamanho variável: um dia podemos ter uma janela com 5000 pontos de sentimento, e no dia seguinte, essa quantidade pode cair para apenas 1000 notícias.

A estrutura do codificador empregado é inspirada no modelo proposto pelo BRACIS (2023). Cada série temporal é processada individualmente por uma instância distinta de

Redes Neurais Recorrentes (RNN), convertendo-as em representações de tamanho fixo, como ilustrado na Figura 10. Utilizamos GRUs, uma arquitetura RNN comprovadamente eficaz em tarefas de codificação de séries temporais. É essa escolha de design que permite o tamanho das janelas temporais serem váriaveis durante o treinamento e a inferência do modelo, já que as RNNs podem processar sequências de comprimentos arbitrários, enquanto que a largura e saída de cada instância de RNN é pré-definida com base no número de características de sua série temporal de entrada.

Os resultados produzidos por cada instância de RNN são então concatenados e encaminhados ao previsor.

O previsor consiste em uma rede LiESN, que recebe o vetor de entrada e o transmite para o decodificador. Este, por sua vez, é uma rede neural densa composta por três camadas densas. O output do decodificador é um valor numérico único, representando a previsão do valor da ação para o tempo t + 1 dia.

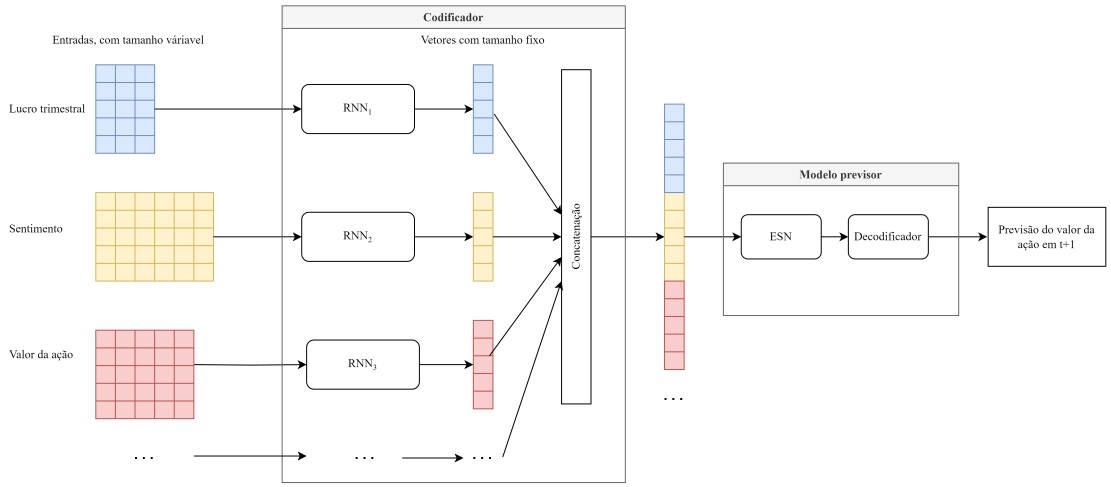


Figura 11 – Detalhamento do Modelo Codificador Previsor no FinStockESN-BR.

4.4 Aplicação Web

A aplicação é projetada para a visualização de notícias, dados e previsões do preço das ações, com o intuito de fornecer uma interface de fácil utilização. Para isso, ela utiliza o modelo LiESN previamente desenvolvido neste trabalho, dados históricos de preços, sentimento de notícias e indicadores macroeconômicos e microeconômicos.

A figura 12 ilustra a arquitetura e o fluxo de dados para a aplicação. A interface do usuário permite que sejam fornecidos um ticker de ação e uma data, através de campos específicos. Essa entrada aciona requisições via API que dispara uma série de serviços no backend. Os serviços de coleta de preços e notícias devem ser acionados automaticamente e periodicamente.

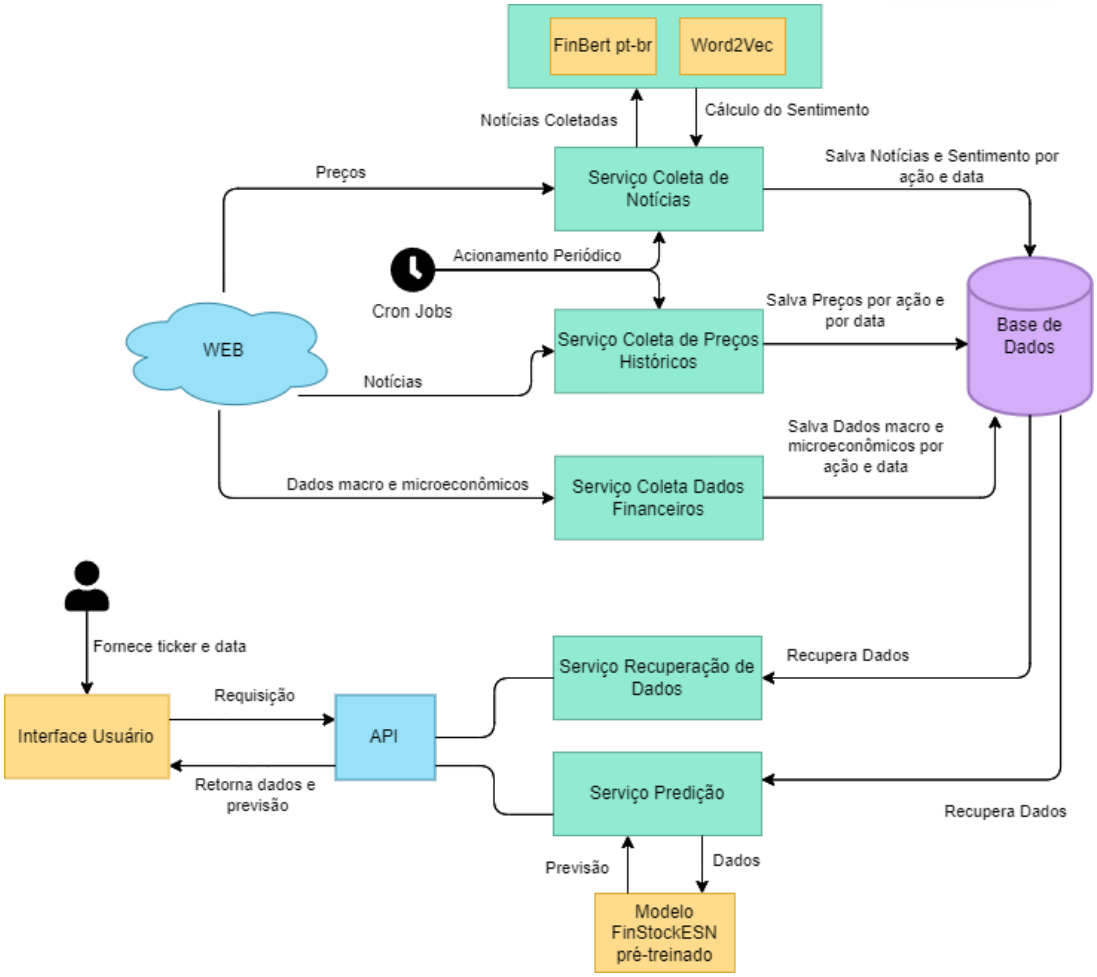


Figura 12 – Arquitetura da Aplicação para Análise e Previsão de Preços de Ações, Integrando Coleta de Dados Históricos, Análise de Sentimento de Notícias e Recuperação de Dados Econômicos via Serviços.

Como é usual em muitas aplicações atuais, ela é composta por duas partes principais: o frontend e o backend.

4.4.1 Frontend

O frontend da aplicação será responsável pela interação com o usuário e apresentação dos dados. Seus elementos principais incluem:

- **Campo seletor de ação:** Permite ao usuário selecionar uma ação específica para análise.
- **Campo para inserção de data:** Habilita a escolha de uma data específica para a previsão.

- **Gráfico da previsão:** Exibe a previsão do preço das ações, iniciando do primeiro dia seguinte ao que houver preço disponível na base de dados até sete dias seguintes a data selecionada no campo anterior.
- **Espaço para apresentação das notícias:** Mostra as notícias relacionadas à ação e à data selecionada.
- **Espaço para dados macro e microeconômicos:** Apresenta dados econômicos referentes à ação selecionada.

4.4.2 Backend

O backend é dividido em serviços, cada um responsável por funcionalidades específicas:

1. **Coleta de Preços Históricos das Ações**: Este serviço é responsável por coletar dados históricos de preços de ações.
2. **Coleta de Notícias Mais Recentes**: Realiza a coleta contínua de notícias, atualizando a base de dados periodicamente.
3. **Coleta de Dados Microeconômicos e Macroeconômicos**: Agrega informações econômicas relevantes de diversas fontes.
4. **Serviço de Predição**: Utiliza os dados coletados e o modelo previamente treinado para retornar a previsão do preço das ações.

4.4.2.1 Coleta de Preços Históricos das Ações

O propósito desse serviço é adquirir e armazenar informações históricas de preços de ações para um período específico.

Funcionalidade Geral: O serviço inicia recebendo parâmetros que definem o período para coleta dos dados. Estes parâmetros incluem o número de dias anteriores a data atual em que está sendo feita a requisição ou um intervalo de datas específico. Há uma validação para assegurar que os parâmetros estejam de acordo com uma limitação do intervalo de datas para um máximo de 30 dias.

Processo de Coleta de Dados: O serviço procede com as seguintes etapas para a coleta de dados:

1. **Determinação do Período de Coleta**: Com base nos parâmetros recebidos, o período para download dos dados é definido, podendo ser um número específico de dias ou um intervalo entre duas datas.

2. **Acesso a Lista de Tickers**: Uma lista de tickers (códigos de ações) é acessada a partir de um repositório de dados. Cada ticker representa uma ação individual a ter dados coletados.

3. **Coleta de Dados Históricos de Ações**: Para cada ticker, o serviço coleta dados históricos de mercado, incluindo o preço de abertura, preço de fechamento e volume de ações negociadas.

4. **Armazenamento dos Dados**: Após a coleta, cada conjunto de dados é armazenado em um banco de dados. Cada registro contém informações com o código da ação (Ticker), data e os itens anteriores.

4.4.2.2 Coleta de Notícias Mais Recentes

Este serviço, integrado ao backend da aplicação web, é dedicado à coleta e armazenamento de notícias recentes relacionadas a ações específicas.

Funcionalidade Geral: O serviço é responsável por duas operações principais: coletar notícias relacionadas a ações especificadas e, posteriormente, disponibilizar a recuperação dessas notícias para visualização.

Processo de Coleta de Notícias: O serviço inicia com as seguintes etapas para a coleta de notícias:

1. **Acesso a Lista de Consultas**: O serviço obtém uma lista de consultas de um repositório de dados, previamente montada, onde cada consulta está relacionada a um ticker específico.

2. **Execução de Consults**: Utiliza o serviço de notícias para buscar as últimas notícias com base nas consultas definidas.

3. **Filtragem de Notícias por Tempo**: Se um filtro de tempo é fornecido, apenas notícias publicadas dentro desse intervalo são consideradas. Esse parâmetro é necessário porque o serviço deverá ser invocado periodicamente, de hora em hora, para buscar as notícias referentes à última hora.

4. **Análise de Sentimento com FinBERT pt-br**: As notícias coletadas são processadas pelo FinBERT pt-br, para cálculo do sentimento das notícias.

5. **Armazenamento de Notícias e Sentimentos**: Após a análise, as notícias junto com os sentimentos calculados são armazenados em um banco de dados. Isso inclui informações como título, descrição, URL, data de publicação, fonte e o sentimento associado.

Integração com API para Recuperação de Notícias: Além da coleta, o serviço oferece uma função integrada através de uma API para recuperar notícias armazenadas. Esta funcionalidade permite acessar as notícias com base no ticker da ação e a data de publicação, para serem exibidas pelo frontend.

4.4.2.3 Coleta de Dados Microeconômicos e Macroeconômicos

Este serviço é dedicado à coleta de dados econômicos relevantes tanto em nível microeconômico quanto macroeconômico.

Funcionalidade Geral: O objetivo deste serviço é agregar informações econômicas que podem influenciar os preços das ações. Os dados microeconômicos são específicos para cada empresa, enquanto os dados macroeconômicos abrangem indicadores econômicos mais amplos.

Processo de Coleta de Dados: O serviço segue as seguintes etapas para coletar dados econômicos:

1. **Acesso a Lista de Empresas**: O serviço inicia com o acesso a uma lista de empresas em um repositório de dados, cujos dados econômicos são necessários para análise.
2. **Coleta de Dados Microeconômicos**: Esta etapa envolve a coleta de informações econômicas específicas de cada empresa, dentre os especificados na 2.3.1.
3. **Coleta de Dados Macroeconômicos**: Aqui, o serviço busca dados mais amplos, como taxas de juros, inflação, taxas de desemprego, PIB e outros indicadores que impactam o mercado de ações em geral, também especificados na 2.3.1.
4. **Armazenamento dos Dados**: Após a coleta, os dados são armazenados em um banco de dados para serem utilizado.

Integração com API para Recuperação de Notícias: Da mesma forma que o serviço de coleta de notícias, o serviço oferece uma função integrada através de uma API para recuperar os dados armazenados a serem exibidas pelo frontend.

4.4.2.4 Serviço de Predição

Este serviço, integrado ao backend da aplicação web, é dedicado à previsão de preços de ações em si utilizando o modelo desenvolvido e os dados coletados.

Funcionalidade Geral: O serviço realiza previsões de preços de ações para um determinado período futuro, com base em parâmetros fornecidos pelo usuário e dados históricos.

Processo de Previsão: O serviço segue as seguintes etapas para realizar as previsões:

1. **Validação e Conversão de Dados**: Valida e converte a data inicial para o formato adequado. Em seguida, conecta-se ao banco de dados para buscar dados históricos do ticker especificado.

2. **Processamento dos Dados Históricos**: Os dados são processados, filtrando-se até a data inicial fornecida. Em seguida, são aplicadas transformações para adequar os dados ao modelo.

3. **Carregamento de Modelos e Scalers**: O serviço carrega o modelo de previsão e scalers (para normalização de dados com o mesmo utilizado no treinamento do modelo) de um repositório. O modelo é treinado previamente, como detalhado ao longo deste trabalho. Utilizando uma ferramenta de exportação, ele é armazenado em um repositório para utilização pela aplicação web.

4. **Normalização e Previsão dos Dados**: Os dados são normalizados e, em seguida, é realizada a previsão dos preços das ações.

5. **Adição de Previsões Futuras**: Além de prever os preços com base nos dados históricos, o serviço estende as previsões para dias adicionais, conforme especificado no parâmetro de previsão. Neste trabalho, esse parâmetro será de 7 dias.

6. **Formatação e Retorno dos Resultados**: Os resultados das previsões são formatados e retornados, incluindo a data da previsão, o ticker e o preço previsto. Se disponíveis, os preços reais para comparação também são incluídos.

Recebimento e Retorno de Parâmetros via API GET: O serviço recebe parâmetros por meio de uma requisição API GET. Os usuários fornecem o ticker da ação, a data inicial e o número de dias para a previsão. Com base nesses parâmetros, o serviço executa o processo de previsão e retorna os dados correspondentes.

Concluindo, o FinStockESN-Br constitui uma plataforma de previsão de preços de ações, integrando técnicas de LiESN, análise de sentimentos baseada em NLP e dados econômicos diversificados. Este capítulo forneceu uma visão da especificação do projeto, delineando a arquitetura e os módulos interativos do sistema. No próximo capítulo, será abordada os meios para sua implementação.

5 Método do trabalho

Este capítulo descreve a metodologia empregada no desenvolvimento do modelo de previsão de séries temporais de preços de ações na bolsa brasileira utilizando Leaky integrator Echo State Networks (LiESN), complementado por dados de análise fundamentalista, indicadores macroeconômicos e análise de sentimentos oriundos de notícias textuais do mercado financeiro. A intenção é estabelecer uma abordagem para a construção, validação e aplicação do modelo proposto.

Inicialmente, serão discutidas as etapas da aquisição de dados, englobando o desenvolvimento do scraping de notícias, a utilização do FinBERT pt-br para análise de sentimentos, e a coleta de dados fundamentalistas e macroeconômicos. Além disso, será abordado o tratamento dos dados adquiridos, garantindo sua qualidade e relevância para o treinamento do modelo. E por fim, a seção 5.8 *Construção de uma aplicação web para visualização dos dados* discute o método para a implementação de uma interface web, que tem como objetivo disponibilizar as previsões e os dados relevantes de forma acessível e interativa para os usuários.

5.1 Visão Geral Metodologia

Na figura 13 é possível observar a visão do projeto e dos componentes que serão desenvolvidos. O primeiro passo é definir quais empresas serão usadas como objeto de estudo, e a partir daí começar a coleta dos dados a respeito delas, como a série temporal do preço da ação, dados fundamentalistas tirados de seus relatórios financeiros e notícias sobre a empresa nos principais sites de mercado financeiro. Em seguida deverão ser obtidos indicadores gerais da economia brasileira. Uma vez que esses dados forem tratados e o FinBERT pt-br adaptado para reconhecer o sentimento da notícia vinculado ao ticker da ação, devem ser definidas as métricas que serão usadas para avaliar o modelo que fará a previsão da precificação. Depois será feita a previsão do preço da ação efetivamente e a análise dos resultados. Por fim, todos esses passos serão reunidos em uma aplicação Web que, a partir do ticker da ação, fará um web scraping para achar notícias relevantes e o histórico de preços, e então irá prever o preço nos dias seguintes usando o modelo LiESN previamente treinado.

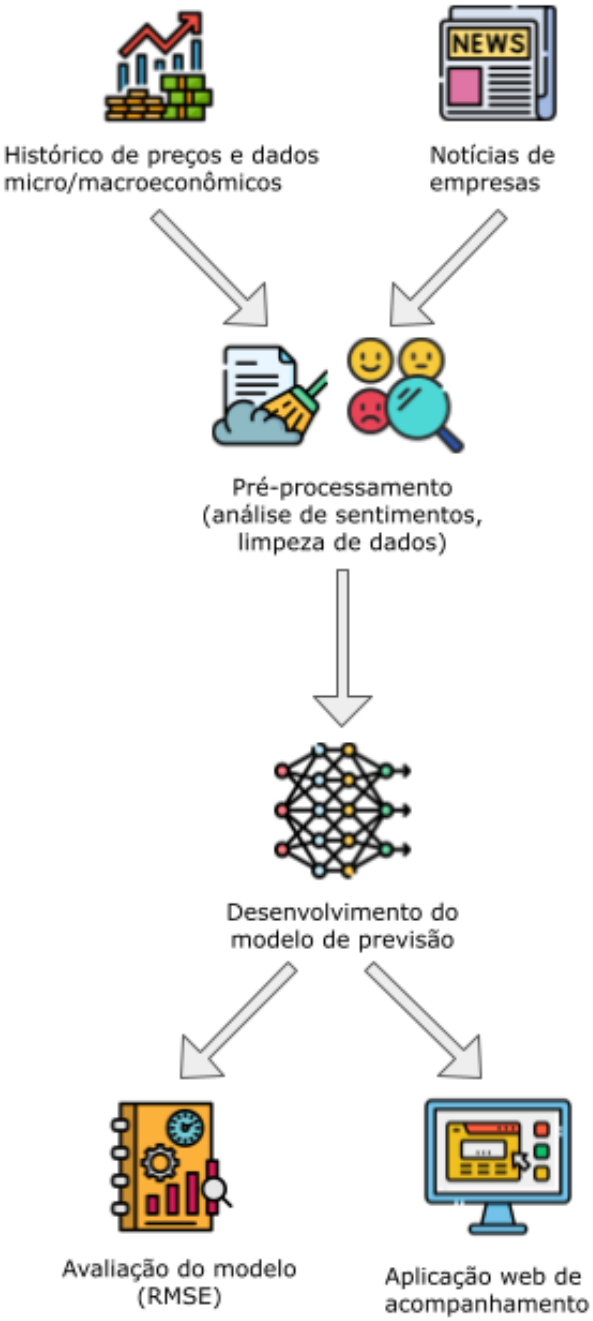


Figura 13 – Visão Geral Metodologia

5.2 Desenvolvimento do Scraping de Notícias

5.2.1 Seleção de Fontes de Notícias

A seleção de fontes é o primeiro passo no desenvolvimento do processo de scraping. Neste projeto, o foco recai sobre um espectro de sites que inclui grandes portais de notícias como Valor Econômico e BBC Brasil, e sites especializados como BrazilJournal e Suno. A relevância desses sites reside não apenas no volume de informações que produzem mas

também na especificidade e influência que têm sobre o setor financeiro.

5.2.2 Concepção do Pipeline de Extração Automática

A fim de facilitar o processo de coleta dessas informações, foi desenvolvido um pipeline para a coleta automática das notícias desses sites de forma eficiente. Optou-se por utilizar o Scrapy como ferramenta principal neste processo, que fornece uma estrutura de web-crawling open-source e de alto nível para Python.

O Scrapy permite definir agentes individuais para acessar, ler os dados de uma página, e escolher como tratar as informações do site. Estes agentes são denominados "Crawlers". Crawlers podem ser definidos para atuarem em um domínio específico (tal como todos os endereços eletrônicos prefixados por "www.google.com.br/"), e podem ter comportamentos únicos dentro de seu domínio. Cada instância de uma página gera uma instância única de um Crawler, não havendo um limite teórico para o número de Crawlers agindo em paralelo, salvo limitações práticas e de hardware.

5.2.3 Extração e Estruturação dos Dados

Todos os Crawlers definidos neste projeto compartilham o mesmo começo ao buscar o arquivo robots.txt do site-alvo. A maioria das páginas web possuem um robots.txt disponível, que segue um padrão usado por websites para comunicar a web crawlers e outros agentes da web quais páginas ou seções do site não devem ser processadas ou escaneadas. Este arquivo é importante para indexadores como o Google ou o Bing poderem escanear o site e saberem quais páginas listar (como notícias) e quais páginas não devem ser listadas (como menus de administradores, páginas de configuração, etc). Essencialmente, o robots serve como um guia sobre o que pode ser acessado publicamente e o que é restrito, e como esse conteúdo deve ser acessado pelos crawlers.

Respeitar o arquivo robots.txt não é apenas uma prática ética, mas também evita possíveis banimentos ou restrições impostas pelo site. Assim, nossos crawler foram projetados para seguir as diretrizes, observando regras como taxa de download e seções restritas. Após a análise do robots.txt, o crawler identifica o sitemap do site. Um sitemap é um arquivo XML que enumera URLs de um site junto com metadados adicionais sobre cada URL (como a última atualização e categoria da página).

Com a lista de URLs escaneada, é iniciado o processo de instânciamento de crawlers para cada página, que vão extrair das páginas de notícia o título, o texto da notícia, a data de publicação e o link da página. No fim do processo, é gerado um arquivo .csv com as informações extraídas de todos os sites visitados.

5.2.4 Pré-processamento e Purificação dos Dados

Com a conclusão da etapa de coleta, nos deparamos com um extenso conjunto de cerca de 1,2 milhões de notícias em sua forma bruta. Porém, antes de avançar para a análise desses dados, é imperativo submetê-los a um rigoroso processo de pré-processamento para assegurar sua qualidade e relevância para o projeto.

Eliminação de Dados Impróprios
O primeiro passo é descartar registros que tenham sofrido qualquer tipo de falha durante a coleta. Isso pode ser manifestado de várias formas: notícias sem conteúdo textual, registros que contêm mensagens de erros ou solicitações de inscrição em conteúdo pago, entre outros. A depuração desses registros garante que o conjunto de dados final contenha apenas informações genuínas e relevantes.

Tokenização do Texto
Uma vez purificado, o próximo passo é desmembrar o texto de cada notícia em palavras ou "tokens". A tokenização é crucial para analisar e compreender o conteúdo em uma granularidade mais detalhada. Utilizamos a biblioteca NLTK e seu tokenizador específico para o idioma português, que é capaz de segmentar o texto com base em espaços, pontuações e outros delimitadores linguísticos. Durante esse processo, caracteres indesejados, como certas pontuações e espaços em branco, podem ser removidos. Por exemplo, a frase "Eu comprei um guarda-chuva ontem." pode ser tokenizada como ('Eu', 'comprei', 'um', 'guarda-chuva', 'ontem', '.', 'Hoje', 'choveu', 'muito')

Remoção de Stopwords
Outra etapa importante é a remoção de stop words, termos que ocorrem com frequência mas que agregam pouco valor semântico a uma frase. Por exemplo, "Eu comprei um guarda-chuva"possui o mesmo significado que "comprei guarda-chuva".

As stopwords escolhidas para serem removidas neste projeto são as mesmas do corpus português do NLTK, e são formados principalmente por:

1. Artigos definidos e indefinidos
2. Preposições e Conjunções
3. Pronomes Pessoais
4. Pronomes Demonstrativos
5. Advérbios e Conjunções Temporais
6. Advérbios de Quantidade

7. Advérbios de Negativa
8. Advérbios de Afirmação
9. Advérbios de Lugar
10. Advérbios de Modo
11. Outras Palavras Comuns (foi, haja, hajam, hajamos, há, havemos, haver, hei, seu, seus, sua, suas, são, ser, será, seriam, seríamos)

5.2.5 Conclusão da Coleta e Preparação dos Dados

A finalização desta etapa resultou em um conjunto de dados textualmente enriquecido e refinado, preparado para a subsequente análise de sentimentos e integração no modelo preditivo do projeto. A metodologia de scraping aqui delineada estabelece uma base para a captura e transformação de grandes volumes de texto em informação para análises financeiras.

5.3 Cálculo de Sentimento

Existem vários modelos e tecnologias para realizar a extração de sentimentos de textos, dentre eles podemos citar o VADER (GILBERT, 2014) e sua adaptação para o português LEiA (ALMEIDA, 2018), o TextBlob (TEXTBLOB. . . ,) ou Flair (AKBIK; BLYTHE; VOLLGRAF, 2018).

Neste trabalho utilizamos o FinBERT pt-br para realizar o cálculo do sentimento dos textos das notícias, devido à sua especificidade para o mercado financeiro brasileiro e pelos resultados alcançados em (SANTOS, 2022).

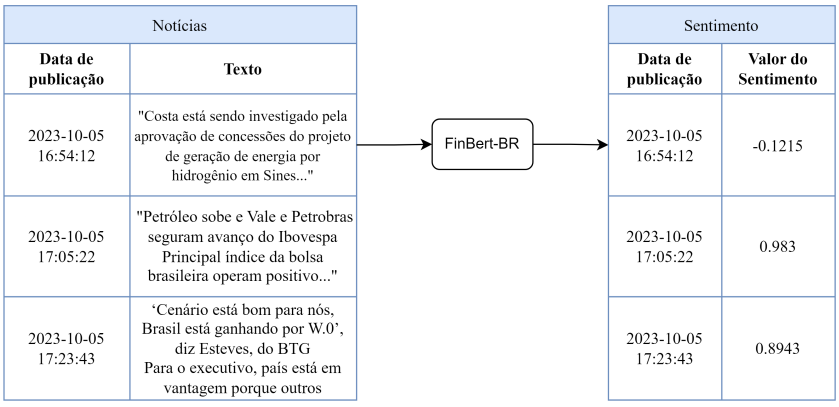


Figura 14 – Utilização do FinBERT PT-BR para cálculo de sentimento.

5.4 Filtragem de Notícias por Empresa e Word Embeddings

Para cada empresa que está sendo analisada neste trabalho, notícias financeiras existem simultaneamente em dois contextos diferentes.

O primeiro é o contexto específico, onde uma notícia referencia diretamente a empresa ou seu setor. Existe também um contexto geral, onde a notícia não tem referência direta a empresa, mas pode ainda ser um indicativo da situação do país como um todo.

Assim, criamos dois dados de entrada a partir do cálculo de sentimentos realizado na seção anterior:

- **Sentimento Geral**: Que possui todas as notícias acumuladas na base de dados, sem distinção de assunto.
- **Sentimento Específico à empresa**: Que inclui e prioriza notícias que possuem uma relação direta com a empresa sendo analisada.

Para filtrar somente as notícias que são relevantes a uma determinada empresa, é usada a técnica de representação vetorial de palavras, descrita na seção 2.8.

Assim, primeiramente é utilizado uma técnica de *Word Embedding* como *Word2Vec* para extrair, de cada notícia, sua representação vetorial segundo a técnica utilizada. Depois, para cada empresa é extraída sua representação vetorial utilizando o mesmo modelo. Um cálculo de semelhança de cossenos informa o grau de similaridade da notícia com a empresa.

Este grau de similaridade é uma das entradas do modelo. Ele é acoplado ao Sentimento Geral na forma de tuplas (Sentimento, Grau de Similaridade), sob o nome **Sentimento Específico**.

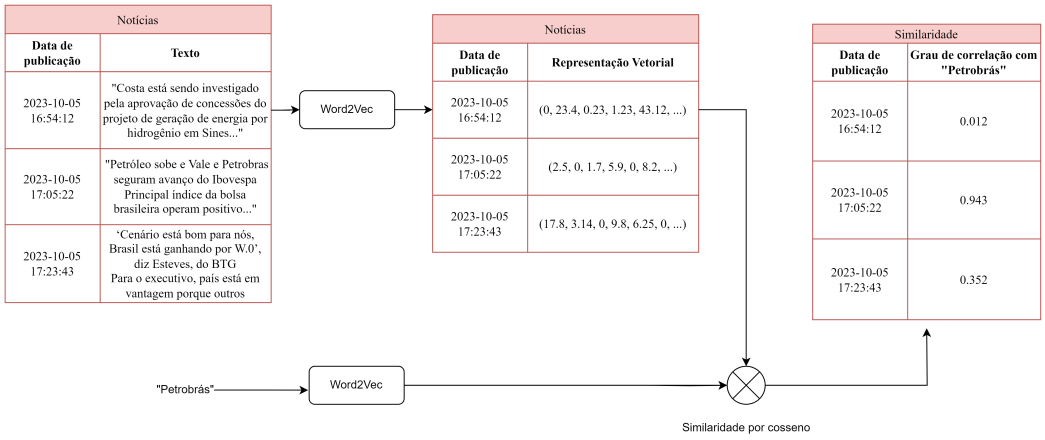


Figura 15 – Exemplo utilizando Word2Vec para cálculo do grau de similaridade.

Modelos pré-treinados de Word Embeddings

O trabalho realizado em (HARTMANN et al., 2017) pelo NILC - (Núcleo Interinstitucional de Linguística Computacional), disponibiliza publicamente recursos vetoriais prontos para serem utilizados nas tarefas de Processamento da Linguagem Natural e Aprendizado de Máquina.

O repositório traz quatro modelos pré-treinados baseados em técnicas de Word2vec, FastText, Wang2vec e Glove.

Para este trabalho são comparados os modelos com 50 dimensões vetoriais.

5.5 Aquisição dos Dados Fundamentalistas da CVM

Para a realização deste estudo, a coleta de dados fundamentalistas de companhias abertas foi efetuada a partir da base de dados disponibilizada pela Comissão de Valores Mobiliários (CVM), que consiste em informações detalhadas sobre o desempenho financeiro das empresas listadas na bolsa de valores brasileira. O intervalo temporal definido para a coleta abrangeu de 2011 até 2023, assegurando uma amostra temporal extensiva para análises financeiras.

5.5.1 Metodologia de Coleta

Inicialmente, procedeu-se com a importação de pacotes essenciais para a manipulação dos dados utilizando a linguagem Python. Os dados foram baixados diretamente da URL base disponibilizada pela CVM, que oferece os arquivos em formato zipado para cada um dos seguintes tipos de documento: Informações Trimestrais (ITR), Demonstrações Financeiras Padronizadas (DFP) e Formulário Cadastral (FCA). Esses documentos foram apresentados na seção 2.4. Um script automatizado foi desenvolvido para iterar sobre os anos especificados, de 2011 até 2023, e para cada tipo de documento, realizando o download e a subsequente extração dos conteúdos dos arquivos zipados para um diretório local.

Após a etapa de download e extração, os dados foram reorganizados e compilados em arquivos separados por ano e por tipo de informação contábil: Balanço Patrimonial Ativo (BPA), Balanço Patrimonial Passivo (BPP), Demonstração do Resultado do Exercício (DRE) e Dados do Fluxo de Caixa (DFC), todos no formato consolidado.

5.5.2 Construção dos Dataframes

Foram gerados dataframes para cada conjunto de informações (BPA, BPP, DRE e DFC), separando-os entre os dados provenientes dos ITR e DFP. Além disso, houve um

filtro para selecionar apenas os dados do último exercício reportado, de modo a manter a consistência temporal e relevância das informações para o estudo.

5.5.3 Cálculo dos Dados Trimestrais

Para a análise dos dados trimestrais, foi necessária a combinação das informações dos ITR, que contemplam os três primeiros trimestres do ano, com os dados anuais dos DFP, que fornecem um panorama completo do exercício fiscal. Implementou-se um cálculo de subtração dos valores acumulados até o terceiro trimestre dos valores anuais reportados nas DFP. Dessa maneira, foi possível construir um conjunto de dados que reflete os resultados de cada trimestre individualmente.

Finalmente, os dataframes foram unificados, consolidando as informações trimestrais, semestrais e de nove meses (9M), disponibilizadas nos formulários ITR com as anuais em uma única estrutura de dados para cada tipo de documento contábil (BPA, BPP, DRE e DFC). Esta estrutura unificada facilita o processo de análise e permite uma visão integrada do desempenho financeiro das empresas ao longo do tempo.

5.5.4 Preparação Final dos Dados

Os dataframes resultantes foram então submetidos a uma limpeza e formatação de dados, incluindo a conversão de datas e a padronização dos formatos numéricos para garantir a precisão das análises subsequentes.

5.6 Aquisição dos dados macroeconômicos através do BACEN

Utilizando a API do Sistema Gerenciador de Séries Temporais (SGS) do Banco Central do Brasil, foram coletados dados macroeconômicos essenciais, incluindo IPCA, taxa SELIC acumulada no mês, PIB trimestral e a taxa de desemprego mensal. Para cada um desses indicadores, os dados obtidos em formato JSON foram convertidos em DataFrames do `pandas`.

Em todos os casos, as colunas de data foram formatadas e as colunas de valores convertidas para ponto flutuante. Os DataFrames foram devidamente rotulados com as colunas ‘Date’ e o respectivo indicador econômico (IPCA, SELIC, PIB, Desemprego), padronizando a estrutura de dados para facilitar a integração e análise futura.

5.7 Obtenção da Série Histórica de Preços

A série de preços é o principal indicador a ser obtido nesse trabalho, pois é dela que será a feita a previsão. A metodologia adotada para a obtenção desses dados históricos

se baseia no uso da biblioteca `yfinance`, que permite o acesso direto às informações de mercado do Yahoo Finance.

Utilizando uma função personalizada em Python, foram extraídos os preços diários de fechamento, abertura e volume das ações das empresas de interesse. Essa coleta foi iniciada a partir do ano de 2011, buscando capturar um amplo histórico que pudesse refletir diferentes ciclos econômicos e eventos de mercado.

A série histórica de preços foi então combinada com outras informações relevantes das empresas, como identificação e dados cadastrais obtidos através do formulário cadastral da CVM. Essa etapa é importante para assegurar que cada cotação de ação esteja corretamente vinculada à sua respectiva empresa.

Adicionalmente, as informações obtidas nas etapas anteriores foram integradas ao conjunto de dados de cotações. Isso possibilitou a associação direta entre os indicadores financeiros das empresas e o comportamento de suas ações no mercado financeiro.

5.7.1 Treinamento, Validação e Teste

O conjunto de todos os dados a serem alimentados para o modelo é separado em três partes com tamanhos proporcionais a 70%, 20% e 10%. São os conjuntos de treinamento, validação e teste respectivamente.

Durante a fase de treinamento, para cada época de um treinamento, o modelo é treinado dia útil a dia útil (que são os dias onde há variação no preço da ação), até o último dia útil do conjunto de treinamento. Uma vez que estamos treinando séries temporais, é importante ressaltar que os dados são fornecidos sequencialmente.

Ao final de cada época de treinamento, é realizada uma avaliação utilizando os 20% dos dados correspondentes ao conjunto de validação. O erro gerado nesta avaliação corresponde ao erro de validação, e é utilizado principalmente para identificação de *overfitting*.

Neste trabalho, para cada sequência de épocas treinados, escolhemos utilizar o método de reter o modelo com o menor erro de validação na sequência.

O conjunto de dados de teste é reservado para o final do trabalho, após todas as etapas de treinamento e finetunning de hiperparâmetros, para avaliar o desempenho final do modelo com dados nunca vistos antes em outras etapas.

5.7.2 Parâmetros e otimização de hiperparâmetros

No campo da aprendizagem de máquina, existe uma distinção entre parâmetros e hiperparâmetros.

Parâmetros são os aspectos que são aprendidos pelo próprio modelo, de forma matemática. Em nosso estudo, os parâmetros incluem elementos como os pesos nas Gated Recurrent Units (GRUs), na Echo State Network (ESN), no Time2Vec, e nos decodificadores, que são redes de múltiplas camadas perceptron (MLP). Esses parâmetros são ajustados automaticamente pelo modelo durante o treinamento para minimizar o erro de previsão.

Em contraste, hiperparâmetros são definidos antes do início do processo de treinamento e não são aprendidos ou ajustados durante o treinamento, mas sim determinados pelo projetista do modelo.

No contexto do nosso projeto, os hiperparâmetros incluem as dimensões dos vetores de saída, a quantidade de neurônios em cada componente do modelo, a taxa de aprendizado, entre outros.

Para otimizar esses hiperparâmetros de forma eficiente, é empregado o Optuna. O Optuna é uma ferramenta que através de um processo iterativo, seleciona hiperparâmetros de forma a maximizar ou minimizar uma função alvo, que no nosso caso se trata do erro de validação de um sequência de 50 épocas.

Como o número de iteração necessárias para a otimização de hiperparâmetros cresce exponencialmente com a quantidade de hiperparâmetros sendo treinados ao mesmo tempo, este processo foi realizado em duas etapas distintas.

Na primeira etapa, focamos na otimização das dimensões estruturais do modelo. Isso incluiu ajustar as dimensões dos vetores de saída, a quantidade de neurônios em cada componente.

Na segunda etapa, voltamos nossa atenção para hiperparâmetros operacionais de cada componente do modelo, como a taxa de aprendizado dos neurônios , o *leaky rate* e o *spectral radius* da ESN, entre outros.

Para cada uma destas etapas, são realizados 100 *trials* com o Optuna, cada trial uma sequência de 50 épocas, para um total de 5000 épocas treinadas por etapa. Ao final de cada etapa, são retidos os hiperparâmetros com o menor erro de validação.

5.8 Metodologia para a Construção da Aplicação Web

A metodologia para a construção da aplicação web, conforme descrita na seção 4.4, foi baseada em princípios de modularização e emprego de infraestrutura de nuvem. A aplicação foi concebida em funções específicas e interconectadas.

5.8.1 Desenvolvimento do Frontend

Para o desenvolvimento do frontend, utilizou-se um framework baseado em JavaScript. A interface do usuário foi projetada para ser intuitiva, facilitando a navegação e o acesso às informações. Componentes reativos foram utilizados para gerenciar estados e o ciclo de vida dos componentes.

Recursos de renderização condicional foram implementados para proporcionar feedback visual ao usuário durante operações de carregamento de dados ou falhas. Essa estratégia foi utilizada para garantir uma experiência de usuário responsiva e interativa.

5.8.2 Desenvolvimento do Backend

No backend, a arquitetura foi dividida em serviços distintos, cada um responsável por funcionalidades específicas, como coleta de dados históricos de preços, notícias, dados econômicos, e o serviço de predição. A implementação desses serviços utilizou uma abordagem de computação sem servidor. Durante o desenvolvimento, foram realizados testes iterativos para validar cada componente e serviço.

Cada serviço foi desenvolvido para operar de forma autônoma, com um fluxo de execução definido. A coleta e atualização automática de dados foram asseguradas por meio de integrações com fontes externas, garantindo a atualização contínua das informações. O serviço de predição utilizou o modelo desenvolvido neste trabalho, aplicando-o aos dados coletados.

Para o armazenamento e recuperação de dados, optou-se por um banco de dados NoSQL, dada a baixa normalização da estrutura de dados. As APIs facilitaram a comunicação e o fluxo de dados entre o backend e o frontend. A integração com um gateway de API centralizou o acesso aos serviços do backend, simplificando o gerenciamento das solicitações dos clientes e a comunicação entre os componentes da aplicação.

Um serviço de monitoramento foi utilizado para acompanhar o desempenho da aplicação e rastrear métricas e logs. Para a segurança e gestão de acesso, empregou-se uma solução de gerenciamento de identidade e acesso, restringindo o acesso aos recursos da aplicação. A implantação e hospedagem da aplicação utilizaram uma plataforma de serviços de nuvem, aproveitando a integração contínua e entrega contínua (CI/CD), o que facilitou e automatizou o processo de compilação e implantação do sistema.

6 Desenvolvimento do Trabalho

6.1 Tecnologias Utilizadas

Para a realização desse projeto, foram empregadas diversas tecnologias relacionadas à inteligência artificial e ciência de dados. Devido à expansão dessa área nos últimos anos, várias ferramentas como bibliotecas, plataformas, etc. foram desenvolvidas para facilitar a criação de novos projetos. Aqui abaixo estão elencadas algumas das mais importantes.

6.1.1 Python

Python é uma linguagem de programação de alto nível amplamente utilizada em diversas áreas da computação, incluindo ciência de dados e inteligência artificial (IA). Sua popularidade cresceu significativamente nos últimos anos devido à sua simplicidade, legibilidade de código e uma ampla gama de bibliotecas e frameworks disponíveis, tornando-a uma escolha preferida para muitos profissionais e pesquisadores.

Python possui uma vasta coleção de bibliotecas de código aberto voltadas para ciência de dados e IA. Algumas das mais populares incluem NumPy (para computação numérica), pandas (para manipulação de dados), Matplotlib e Seaborn (para visualização de dados), scikit-learn (para aprendizado de máquina), TensorFlow e PyTorch (para IA profunda) e muitas outras. Essas bibliotecas fornecem um conjunto de ferramentas poderosas para análise, modelagem e implementação de algoritmos.

6.1.2 Jupyter Notebook

O Jupyter Notebook é uma aplicação web de código aberto que permite criar e compartilhar documentos interativos que contêm código, texto formatado, visualizações e equações matemáticas. Ele é amplamente utilizado em ciência de dados, pesquisa acadêmica, ensino e análise exploratória de dados.

Uma das principais vantagens do Jupyter Notebook em relação ao uso do Python normal em um ambiente de script tradicional é que ele permite executar código em blocos, chamados de células. Isso torna o processo de desenvolvimento mais iterativo, pois você pode executar partes do código e ver os resultados imediatamente, o que é especialmente útil em análise de dados e experimentação. Ele também suporta a inclusão de gráficos e visualizações diretamente no documento, o que é extremamente útil para análise de dados e geração de relatórios, e pode-se adicionar texto formatado em Markdown ao lado do código, criando uma narrativa rica no documento.

6.1.3 Anaconda

Anaconda é uma plataforma de código aberto projetada para simplificar a instalação, gerenciamento e distribuição de pacotes e ambientes de desenvolvimento para ciência de dados, análise de dados e computação científica. É uma ferramenta extremamente popular entre cientistas de dados, engenheiros de machine learning e pesquisadores devido à sua facilidade de uso e à vasta coleção de pacotes e bibliotecas que ela oferece.

Algumas das funcionalidades que Anaconda oferece incluem: um gerenciador de pacotes e ambientes que permite instalar, atualizar e gerenciar bibliotecas e pacotes de forma eficiente, criação de ambientes virtuais isolados para isolar projetos e suas dependências, um vasto repositório de pacotes pré-compilados e otimizados para várias plataformas e sistemas operacionais, entre outras facilidades para cientistas de dados que usam Python.

6.1.4 TensorFlow

O TensorFlow é uma plataforma de código aberto amplamente utilizada para aprendizado de máquina e aprendizado profundo. Desenvolvida pela Google Brain team, é conhecida por sua robustez e capacidade de escala, sendo uma opção preferida para produção e implantação de modelos de aprendizado de máquina. O TensorFlow permite a definição e treinamento de modelos de redes neurais com eficiência.

A base do TensorFlow é a manipulação de tensores, que são estruturas de dados que permitem cálculos em grandes volumes de dados com otimização para processamento paralelo, tanto em CPUs quanto em GPUs. Com o TensorFlow, é possível construir redes neurais complexas através de uma API de alto nível, como o Keras. Isso simplifica a definição de camadas, funções de ativação e algoritmos de otimização, como o Adam ou o SGD. O TensorFlow também automatiza o cálculo de gradientes, facilitando o ajuste de pesos durante o treinamento de modelos.

6.1.5 Gensim

Gensim é uma biblioteca de código aberto em Python amplamente utilizada para processamento de linguagem natural (NLP) e modelagem de tópicos. Ela foi projetada especificamente para trabalhar com algoritmos de processamento de texto e criação de modelos de tópicos, tornando-a uma escolha popular para tarefas relacionadas à mineração de textos e análise de documentos. O Gensim é conhecido por sua eficiência e escalabilidade ao lidar com grandes conjuntos de dados textuais.

O Gensim fornece ferramentas para criar representações vetoriais densas de palavras, como o Word2Vec e o Doc2Vec. Essas representações vetoriais são usadas em várias tarefas de processamento de linguagem natural, incluindo similaridade de palavras, classificação

de texto e agrupamento de documentos. Essa é a aplicação principal do Gensim neste projeto, mas ele também é muito usado para a modelagem de tópicos, que é a criação de modelos probabilísticos que identificam tópicos em coleções de documentos. Esses modelos então podem ser usados para extrair insights sobre o conteúdo dos textos e categorizar documentos com base nos tópicos predominantes.

6.1.6 NTLK

O NLTK, ou Natural Language Toolkit, é uma biblioteca de código aberto em Python usada para trabalhar com processamento de linguagem natural (NLP). Ela fornece uma variedade de ferramentas, recursos e bibliotecas de dados que facilitam a análise e o processamento de texto em linguagem natural. O NLTK é amplamente utilizado em tarefas de processamento de linguagem natural, como tokenização, análise sintática, marcação de partes do discurso, análise de sentimentos, classificação de texto e muito mais.

O NLTK oferece métodos para dividir o texto em unidades menores, como palavras ou frases (chamadas de tokens). Isso é útil para preparar texto para análise e extração de informações. Uma vez o texto tokenizado, o NLTK tem ferramentas para realizar análises sintáticas em texto, identificando relações gramaticais entre palavras em uma frase ou sentença. Isso é útil para, por exemplo, remover artigos e preposições de um texto, deixando somente os termos mais relevantes para a compreensão do tema do texto.

6.1.7 Pandas

A biblioteca pandas é uma biblioteca de código aberto em Python amplamente utilizada para análise e manipulação de dados. Ela fornece estruturas de dados flexíveis e ferramentas de análise de alto desempenho que são essenciais para cientistas de dados, analistas de dados e desenvolvedores que trabalham com dados em Python. O pandas é especialmente adequado para lidar com dados tabulares e séries temporais, como as usadas neste projeto.

O pandas oferece duas principais estruturas de dados: o DataFrame e a Series. O DataFrame é uma estrutura bidimensional semelhante a uma tabela ou planilha do Excel, com linhas e colunas rotuladas, ideal para armazenar e manipular dados tabulares. Já a Series é uma estrutura unidimensional que pode ser vista como uma coluna de um DataFrame ou uma série temporal. O pandas fornece métodos para importar dados de diferentes formatos, como CSV, Excel, SQL, JSON e HDF5, e uma ampla gama de funções e operações para manipular esses dados, como filtragem, seleção, agregação, mesclagem, pivoteamento e ordenação de dados.

6.1.8 Dask

A biblioteca Dask é uma biblioteca de código aberto em Python que se concentra em paralelizar e escalonar operações de processamento de dados para lidar com conjuntos de dados maiores do que a memória disponível em um único computador. Ela fornece estruturas de dados e ferramentas que se assemelham às do Pandas, tornando-a uma opção atraente quando se lida com análise de dados e manipulação de dados em escala.

Uma das principais vantagens do Dask é sua capacidade de lidar com dados que não cabem na memória RAM de um único computador. Ele divide tarefas em várias partes menores e distribui essas partes em vários núcleos de CPU ou até mesmo em cluster de máquinas, permitindo que você processe grandes conjuntos de dados de forma eficiente. O Pandas é limitado pelo tamanho da memória disponível e pode enfrentar problemas de desempenho com dados muito grandes.

6.2 Projeto e Implementação

Esta seção aborda o projeto detalhado e a implementação prática dos procedimentos da metodologia e procedimentos empregados nesse trabalho.

6.2.1 Scraping de Notícias

Ao final do processo de Scraping, realizamos uma coleta histórica abrangente, acumulando **1.2 milhões de notícias** de quatro fontes distintas: Valor Econômico, Infomoney, Suno e Brasiljornal. A distribuição detalhada da quantidade de notícias obtidas de cada fonte é ilustrada na figura 18. Vale ressaltar que a abordagem aqui utilizada, contrasta com a técnica implementada na aplicação Web, onde foi utilizada uma API do Google News Feed. Isso ocorre porque na aplicação somente é necessário as notícias mais recentes.

6.2.2 Resultados cálculo do índice de sentimento

Ao final do processo de cálculo de sentimentos, obtivemos duas séries temporais:

O sentimento geral, que considera todas as notícias sem distinção de assunto, e o sentimento específico, que é a composição do sentimento geral com o grau de correlação de cada notícia com o termo da empresa sendo analisado.

Na figura 16 podemos observar a distribuição dos sentimentos das notícias. É possível observar que existem mais notícias com sentimentos nos extremos de positivo (1) e negativo (-1), além de um pico concentrado em notícias com sentimento neutro (0).

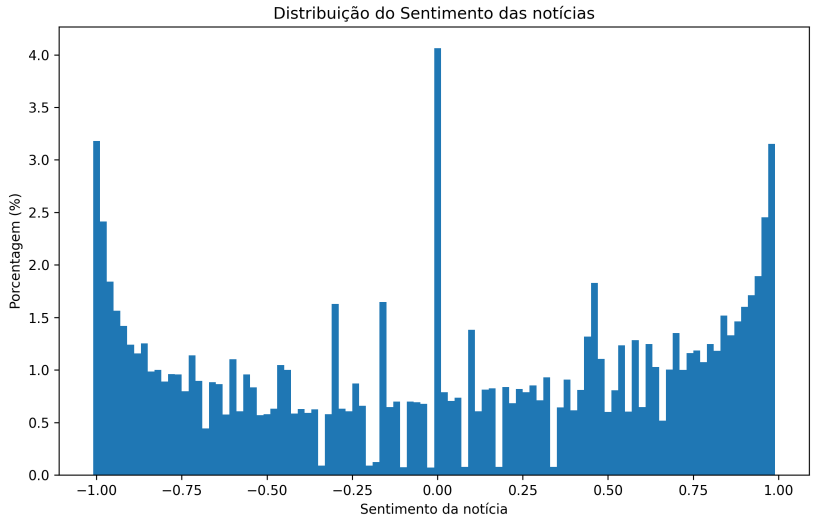


Figura 16 – Distribuição dos sentimentos das notícias: sentimento (de -1, indicando muito negativo a +1, muito positivo) versus a porcentagem de notícias.

A figura 17 mostra como as notícias estão correlacionadas com o termo “Petrobrás”.

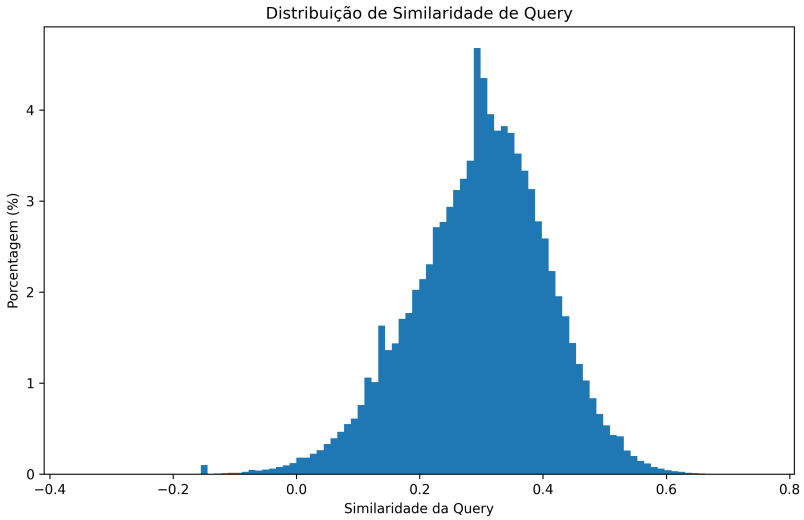


Figura 17 – Distribuição do grau de relação das notícias, com o termo analisado “Petrobrás”.

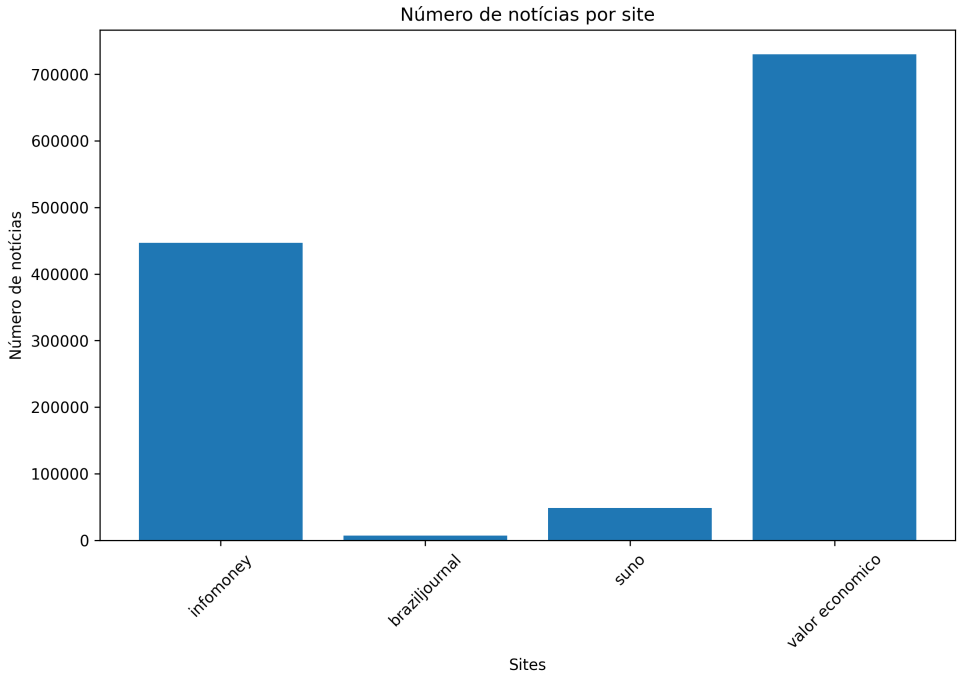


Figura 18 – Número de notícias por fonte utilizada.

6.2.3 Aquisição dos Dados Fundamentalistas da CVM

Ao final da etapa de aquisição de dados fundamentalistas, conforme descrito na seção 5.5, têm-se quatro dataframes principais: *dre*, *dfc*, *bpa*, e *bpp*. A Tabela 1 detalha cada um dos dataframes obtidos durante essa etapa.

Tabela 1 – Descrição dos Dataframes Coletados

Dataframe	**Descrição**
dre	Contém informações detalhadas sobre a Demonstração de Resultado do Exercício (DRE) das empresas.
dfc	Apresenta dados sobre o Fluxo de Caixa das empresas.
bpa	Inclui detalhes do Balanço Patrimonial Ativo das empresas.
bpp	Consiste em dados do Balanço Patrimonial Passivo das empresas.

Cada um desses dataframes contém informações detalhadas sobre as empresas cadastradas na CVM, conforme descrito na Tabela 2.

Tabela 2 – Descrição dos campos nos dataframes

Campo	**Descrição**
CNPJ_Companhia	CNPJ da companhia
CNPJ_CIA	CNPJ da empresa
DENOM_CIA	Denominação da companhia
CD_CVM	Código CVM da empresa
MOEDA	Moeda utilizada nos dados financeiros
ESCALA_MOEDA	Escala da moeda
DT_INI_EXERC	Data de início do exercício
DT_FIM_EXERC	Data de fim do exercício
CD_CONTA	Código da conta
DS_CONTA	Descrição da conta
VL_CONTA	Valor da conta
ANO	Ano do dado financeiro
PERIOD_TYPE	Tipo de período (trimestral, semestral, etc.)

Integração dos Dados do Formulário Cadastral (FCA)

Além dos dataframes financeiros mencionados anteriormente, também foram coletados e processamos dados do Formulário Cadastral (FCA) disponibilizado pela CVM. O resultado foi a construção de um dataframe denominado *fca*, que contém as seguintes colunas principais:

- **CNPJ_Companhia**: CNPJ da companhia, utilizado como identificador único da empresa.
- **Codigo_Negociacao**: Código de negociação da empresa na bolsa, conhecido como ticker, essencial para análises de mercado.

Utilizou-se o campo *CNPJ_Companhia* do *fca* como chave para realizar uma integração com os dataframes *dre*, *dfc*, *bpa*, e *bpp*, integrando assim o dado do ticker a cada registro financeiro correspondente.

Dentre as empresas analisadas nos dataframes, optou-se por focar nas 20 principais empresas listadas na B3, conforme a sua participação no índice Ibovespa em 2023. A Tabela 3 apresenta essa lista, destacando as empresas com maior peso no índice.

Tabela 3 – As 20 Principais Empresas Listadas na B3 em 2023 - Retirado de (Equipe Toro Investimentos, 2023)

Código	**Empresa**	**Participação no índice (%)**
VALE3	Vale	15,546
ITUB4	Itaú Unibanco	6,665
PETR4	Petrobras PN	5,778
PETR3	Petrobras ON	4,232
ABEV3	Ambev	3,571
BBAS3	Banco do Brasil	3,309
ELET3	Eletrobras ON	3,735
B3SA3	B3	3,685
BBDC4	Bradesco	3,951
WEGE3	Weg	2,809
ITSA4	Itaúsa	2,343
RENT3	Localiza	2,447
BBSE3	BB Seguridade	1,257
EQTL3	Equatorial	1,658
RADL3	RaiaDrogasil	1,636
RAIL3	Rumo	1,353
BRKM5	Braskem	0,284
JBSS3	JBS	1,118
PRIO3	PetroRio	1,598
RDOR3	Rede D'Or	1,347

Foco das Informações Contábeis

Das informações contábeis disponíveis, este trabalho foca em cinco principais indicadores: receita, lucro líquido, patrimônio líquido e fluxo de caixa operacional descritos na seção 2.3.1. Essas informações são obtidas respectivamente dos dataframes *dre*, *dre*, *bpp* e *dfc*.

6.2.4 Obtenção da Série Histórica de Preços

Foi realizada a coleta da série histórica de preços das ações das empresas listadas na B3. Utilizou-se uma função como descrita na seção 5.7, que emprega a biblioteca *yfinance* para obter dados históricos de preços desde o início de 2011.

Estruturação dos Dados de Cotação

Após a coleta, os dados foram estruturados em um dataframe denominado *cotacoes*. Em seguida, realizado uma integração dele com os dataframes financeiros e cadastrais anteriormente mencionados, utilizando o *CNPJ_Companhia* como chave. Essa integração

permitiu relacionar as cotações diárias de ações com as informações financeiras e cadastrais das empresas, enriquecendo as informações disponíveis. Os campos do dataframe de cotações estão estruturados na tabela 4 abaixo:

Tabela 4 – Estrutura do Dataframe de Cotações

Coluna	**Descrição**
Date	Data da cotação
Ticker	Código de negociação da empresa na bolsa
Close	Preço de fechamento da ação
Open	Preço de abertura da ação
Volume	Volume de ações negociadas
CNPJ_Companhia	CNPJ da companhia
Valor_Mobilario	Valor mobiliário relacionado
Codigo_Negociacao	Código de negociação da empresa
DENOM_CIA	Denominação da companhia

Este dataframe oferece uma visão abrangente da movimentação diária das ações (para dias em que houve negociação) de cada empresa listada na B3.

6.2.5 Aquisição de Dados Macroeconômicos

Foram coletados dados macroeconômicos, utilizando a API do Sistema Gerenciador de Séries Temporais (SGS) do Banco Central do Brasil conforme mencionado na seção 5.6. Os dados coletados incluem IPCA, SELIC, PIB e Taxa de Desemprego. Cada conjunto de dados foi convertido em um dataframe separado, contendo um campo de data e o respectivo valor para o indicador macroeconômico.

Estruturação e Periodicidade dos Dataframes

Os dataframes foram estruturados com duas colunas principais: *Date* e o valor do indicador econômico. A seguir, detalha-se tais colunas e a periodicidade de cada um dos indicadores:

Tabela 5 – Estrutura e Periodicidade dos Dataframes Macroeconômicos

Indicador	**Descrição da Coluna**	**Periodicidade**
IPCA	Índice de Preços ao Consumidor Amplo	Mensal
SELIC	Taxa do Sistema Especial de Liquidação e Custódia	Mensal Acumulado
PIB	Produto Interno Bruto	Mensal
Desemprego	Taxa de Desemprego (PNAD Contínua)	Mensal

6.2.6 Junção Dados Fundamentalistas, Macroeconômicos e Série de Preços

Após a coleta e estruturação dos dados financeiros, cadastrais e macroeconômicos, procedeu-se com a junção de todas essas informações em um único dataframe. Este processo envolveu a integração dos dados de preço das ações, indicadores financeiros de receita, lucro, patrimônio líquido, fluxo de caixa operacional e os dados macroeconômicos (IPCA, SELIC, PIB e Taxa de Desemprego).

Estratégia de Integração dos Dados

O processo de integração foi realizado de modo a garantir a correta correspondência temporal e empresarial dos dados. Utilizou-se a técnica de integração 'asof' para alinhar as séries de preço das ações com os dados financeiros trimestrais e os indicadores macroeconômicos, com base na data e no ticker da empresa. Isso permitiu a criação de um conjunto de dados históricos para cada empresa.

Dataframe Final

O dataframe final gerado oferece uma visão do desempenho financeiro e de mercado das empresas, enriquecida com contexto macroeconômico. A estrutura do dataframe é detalhada na Tabela 6.

Tabela 6 – Estrutura do DataFrame Final Consolidado

Coluna	**Descrição**
Ticker	Código de negociação da empresa na bolsa
Date	Data da cotação ou do registro financeiro
Close	Preço de fechamento da ação
Open	Preço de abertura da ação
Volume	Volume de ações negociadas
Receita	Receita da empresa no período
Lucro	Lucro da empresa no período
Fluxo_Caixa	Fluxo de caixa operacional da empresa
Patrimonio_Liquido	Patrimônio líquido da empresa
IPCA	Índice de Preços ao Consumidor Amplo (inflação)
SELIC	Taxa do Sistema Especial de Liquidação e Custódia
PIB	Produto Interno Bruto
Desemprego	Taxa de Desemprego

Armazenamento dos Dados

Finalmente, este dataframe final foi consolidado em arquivos CSV, tanto para cada

empresa individualmente quanto para o conjunto total de empresas, abrangendo o período de 2011 a 2023. No apêndice, são apresentados gráficos ilustrando a série histórica de preços da WEG além de dados macroeconômicos. Ela foi escolhida apenas como exemplificação.

6.3 Definição de número de épocas e primeiros treinamentos

Durante as fase iniciais do treinamento dedicamo-nos à definição e experimentação com diferentes hiperparâmetros e configurações de entradas. Esta escolha baseou-se em informações obtidas através da literatura especializada e de testes empíricos conduzidos durante os estágios iniciais dos treinamentos. Os hiperparâmetros iniciais foram, portanto, selecionados com o intuito de estabelecer uma linha de base sólida para os treinamentos subsequentes.

A tabela 7 mostra uma lista de hiperparâmetros sendo configurados manualmente, que incluem as dimensões dos vetores, quantidades de neurônios em redes diferentes e parâmetros operacionais de algumas redes.

Tabela 7 – Tabela com os hiperparâmetros configurados manualmente, e os valores iniciais adotados com base na literatura e em testes empíricos

Hiperparâmetro	**Valor**
Dimensão do vetor Time2Vec	8
Dimensão de saída de cada GRU	32
Neurônios na ESN	100
Neurônios na primeira camada do decodificador	8
Neurônios na segunda camada do decodificador	4
Conectividade dos neurônios na ESN	0.1
Leaky rate da ESN	1
Spectral radius da ESN	0.9
Função de ativação da ESN e decodificador	tanh

Durante os testes empíricos iniciais, foi percebido que o modelo frequentemente alcançava um erro de validação mínimo entre 20 e 30 épocas, antes de começar a apresentar sinais de overfitting. Por conta disto, foram definidos como o número de épocas para realizar os estudos como 50 épocas por estudo. Trata-se de um número que permite o modelo alcançar o erro de validação mínimo, sem aumentar os custos computacionais de forma a impossibilitar as sessões de *finetuning*.

6.4 Escolha de WordEmbedding

Antes de realizar os processos de Finetuning, foi realizado um teste para saber qual dos quatro modelos pré-treinados pelo NILC em (HARTMANN et al., 2017) traria os

melhores resultados para o trabalho. O gráfico 19 demonstra os resultados da realização de um treinamento por 50 épocas.

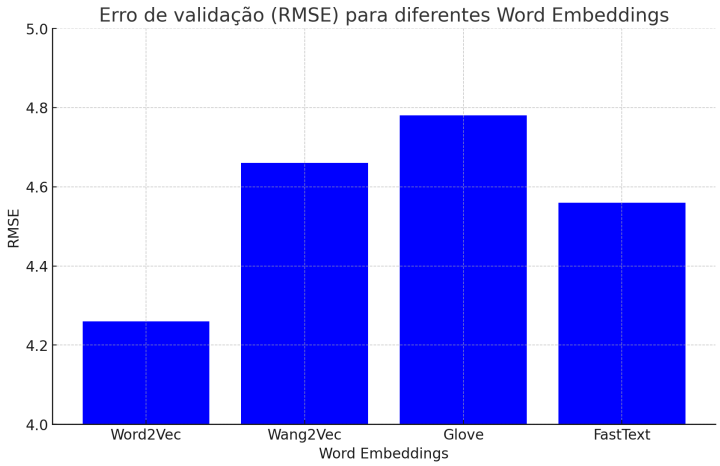


Figura 19 – Erros de validação encontrados para cada Word Embedding testado.

6.5 Ajuste Fino (Finetuning)

Inicialmente, um dos problemas enfrentados durante os primeiros treinamentos do modelo era a ausência de conhecimento do grupo sobre uma configuração inicial de hiperparâmetros efetiva com relação à própria estrutura da rede (o que inclui aspectos como dimensões dos vetores de saída de cada componente do modelo, número de camadas e neurônios em cada rede neural). Para estabelecer uma configuração inicial que fosse satisfatória, utilizou-se a ferramenta Optuna para um estudo automatizado desses hiperparâmetros.

A cada estudo, o Optuna faz uma escolha de hiperparâmetros com base nos resultados de todas as escolhas anteriores, de forma a encontrar rápidamente uma seleção de hiperparâmetros que maximize os resultados desejados. Foram realizadas duas seções de ajuste fino com o Optuna. A primeira sessão teve como objetivo estudar as dimensões apropriadas para a arquitetura do modelo. Uma segunda sessão de estudos do Optuna foi realizada em seguida, utilizando as dimensões encontradas anteriormente, desta vez buscando otimizar hiperparâmetros internos de cada componente utilizado.

Em cada sessão de estudos, o Optuna realizou a tarefa de treinar os modelos por 50 épocas, 200 vezes (um treinamento total de 10.000 épocas por sessão).

6.5.1 Primeiro Ajuste Fino

Durante o primeiro ajuste fino, estamos interessados em otimizar as dimensões dos vetores de saída e da quantidade de neurônios dos componentes do modelo. Durante

essa seção, os intervalos possíveis de busca foram definidos com base em testes empíricos, valores usuais encontrados na literatura ou nas próprias recomendações do TensorFlow.

Os hiperparâmetros sendo otimizados e seus intervalos estão descritos em detalhe na tabela 8, enquanto os resultados obtidos estão descritos na tabela 9.

As resultados desta otimização podem ser observados na figura 20. Observa-se que o estudo começa com um erro RMSE de R$4.19 e é progressivamente reduzido até R$3.39.

Tabela 8 – Janelas de valores para cada hiperparâmetro

Hiperparâmetro	**Intervalo**
Dimensão do Time2Vec (L)	0 a 12
Dimensão de saída de cada GRU	1 a 64
Neurônios na ESN	1 a 200
Neurônios na primeira camada do decodificador	1 a 50
Neurônios na segunda camada do decodificador	1 a 50

Tabela 9 – Valores encontrados para cada hiperparâmetro

Hiperparâmetro	**Valor Encontrado**
Dimensão do Time2Vec (L)	7
Dimensão de saída de cada GRU	9
Neurônios na ESN	40
Neurônios na primeira camada do decodificador	26
Neurônios na segunda camada do decodificador	36

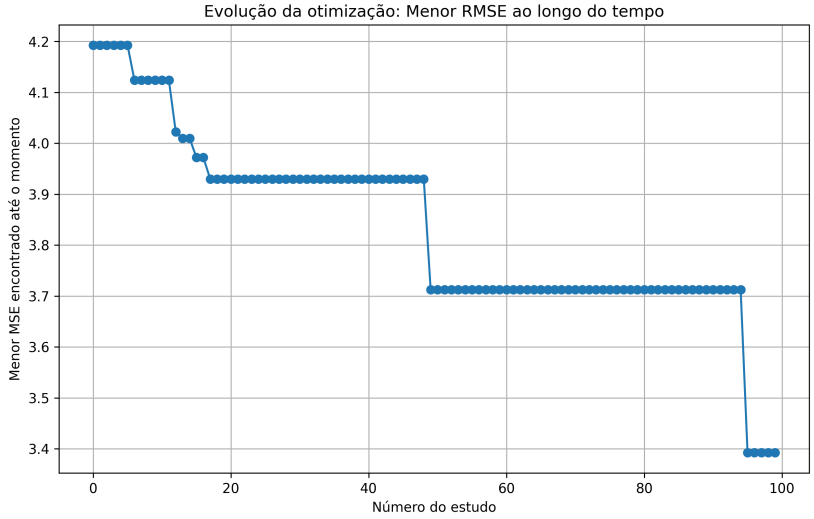


Figura 20 – Evolução do erro mínimo na primeira sessão de ajuste fino.

6.5.2 Segundo Ajuste Fino

Após a realização da primeira sessão de ajuste fino, focada na otimização das dimensões estruturais do modelo, procedeu-se à segunda etapa de otimização utilizando a ferramenta Optuna. Esta fase teve como objetivo aprimorar outros hiperparâmetros numéricos que impactam diretamente no desempenho do modelo. Os hiperparâmetros escolhidos para esta otimização incluíram aspectos relacionados à conectividade neuronal, taxa de vazamento (leaky rate), raio espectral da Echo State Network (ESN) e a função de ativação.

Na tabela 10 são detalhados os hiperparâmetros considerados na segunda sessão de ajuste fino, bem como os intervalos de valores nos quais a busca foi realizada. No caso da Função de ativação, foram consideradas quatro funções possíveis.

Tabela 10 – Intervalos dos hiperparâmetros na segunda sessão de ajuste fino

Hiperparâmetro	**Intervalo**
Conectividade dos neurônios na ESN	0.05 a 0.4
Leaky rate da ESN	0 a 1
Spectral radius da ESN	0 a 1
Função de ativação	tanh, sigmoid, softmax, elu

É importante notar que, embora a otimização dos hiperparâmetros nesta segunda fase não tenha resultado em melhorias tão expressivas quanto na primeira, ela ainda contribuiu para uma melhoria no modelo e ainda garantiu que a escolha de parâmetros iniciais esperados não eram muito distantes dos valores ótimos encontrados.

Os valores encontrados para os hiperparâmetros estão descritos na tabela 11.

Tabela 11 – Valores encontrados para os hiperparâmetros na segunda sessão de ajuste fino

Hiperparâmetro	**Valor Encontrado**
Conectividade dos neurônios na ESN	0.16
Leaky rate da ESN	0.99
Spectral radius da ESN	0.84
Função de ativação	tanh

6.6 Resultados do modelo

Ao final das duas etapas de ajuste fino, chegamos a um ponto onde podemos avaliar os efeitos das diferentes entradas do modelo. Para tal, utilizando os hiperparâmetros encontrados e as bases de dados construídas até esta etapa, foram realizados treinamentos adicionais finais, variando o conjunto de entradas do modelo. Seguindo o mesmo esquema de divisão do conjunto de treinamento, validação e teste em 70%, 20% e 10%, o modelo foi

treinado por cada conjunto de entradas por 200 épocas, e foi retido o ponto no treinamento com o menor erro de validação, afim de evitar possível *overfitting*. O modelo então é avaliado utilizando os 10% dos dados do conjunto de testes para a empresa"Petrobrás" (PETR4).

Os resultados obtidos são apresentados na tabela 12.

Tabela 12 – Comparação do Erro Médio Quadrático (RMSE) para diferentes configurações de entradas do modelo FinSTOCKESN-BR

Configuração do Modelo	**Erro (RMSE)**
Modelo Baseline	8.77
Modelo com todas as entradas	3.08
Modelo sem Time2Vec	3.33
Sem a entrada do sentimento geral	3.01
Sem nenhum dos dois sentimentos	**2.97**
Apenas com o preço de Close	3.19

O modelo baseline, que realiza previsões simplesmente estimando não haver variação do erro de um dia para outro, apresentou um erro médio (RMSE) de R$8.77. A forma mais completa do modelo, contando com todas as entradas, apresentou um erro de R$3.08, que é uma melhora significativa comparada com o Baseline.

Foi possível concluir que a integração do *time encoding* Time2Vec leva a uma melhora considerável no resultado final, com sua inclusão reduzindo o erro de R$3.33 para R$3.08.

Por outro lado, a remoção de entradas de sentimento do modelo mostrou um impacto menos pronunciado do que esperado. Removendo a entrada dos sentimentos o modelo obteve um desempenho levemente superior. No caso o modelo sem o sentimento geral obteve uma pontuação de R$ 3.01 enquanto que removendo os dois sentimentos resultou no melhor valor encontrado, R$2.97.

Ao mesmo tempo, a configuração que utilizou apenas o preço de fechamento das ações apresentou um RMSE de 3.19, sugerindo que, embora os indicadores econômicos adicionais possam enriquecer o modelo, o histórico dos preços de fechamento por si só já é um indicador muito significativo.

6.7 Implementação da Aplicação Web

A aplicação web, como mencionado na especificação em 4.4, é estruturada em duas partes principais: o frontend e o backend. Para a sua construção, optou-se pela utilização da infraestrutura de nuvem da AWS, aproveitando sua robustez, ampla gama de serviços e documentação. A seguir, detalha-se uma descrição dos principais serviços da AWS utilizados na implementação:

- **AWS Lambda:** Este serviço de computação sem servidor foi utilizado para executar o código do backend em resposta a eventos, como solicitações HTTP via API Gateway. A Lambda permite executar funções em resposta a gatilhos específicos, sem a necessidade de gerenciar servidores.
- **Amazon DynamoDB:** é um banco de dados NoSQL que foi utilizado para armazenar e recuperar dados. Na aplicação, foi empregado para guardar informações como preços históricos de ações, dados de notícias e indicadores econômicos.
- **Amazon S3 (Simple Storage Service):** O S3 foi utilizado para armazenar arquivos e dados em formato de objeto, como os modelos treinados, arquivos CSV contendo tickers de ações e outros recursos estáticos necessários pela aplicação.
- **API Gateway da AWS:** Este serviço foi utilizado para criar, publicar, manter, monitorar e proteger as APIs da aplicação web. Ele atua como um "front door"para acessar dados, lógica de negócios e funcionalidade do backend hospedado na AWS. Na aplicação, o API Gateway gerencia as solicitações do cliente para o backend, integrando-se com funções Lambda.
- **AWS IAM (Identity and Access Management):** O IAM foi empregado para gerenciar o acesso aos serviços e recursos da AWS de forma segura. Isso inclui a definição de permissões para quem pode ou não pode utilizar os recursos da AWS, o que é importante para a segurança e a integridade do sistema.
- **AWS CloudWatch:** Este serviço de monitoramento foi utilizado para coletar e rastrear métricas e logs. Além disso, o recurso Event Bridge foi utilizado para a implementação de cron jobs, como a invocação do serviço que faz a coleta de notícias periodicamente.
- **AWS Amplify:** O Amplify foi utilizado para hospedar a aplicação web. Este serviço oferece um ambiente para implantação e hospedagem de aplicações web. Com integração contínua e entrega contínua (CI/CD), facilita e automatiza o processo de compilação e implantação do frontend.

A figura 21 exibe a arquitetura implementada da aplicação web, utilizando a infraestrutura de serviços da AWS. Esta arquitetura, correspondente à figura 12 da especificação, é agora ilustrada em sua forma técnica, destacando a integração prática das ferramentas da AWS. A seguir, será detalhada a funcionalidade de cada serviço envolvido no sistema.

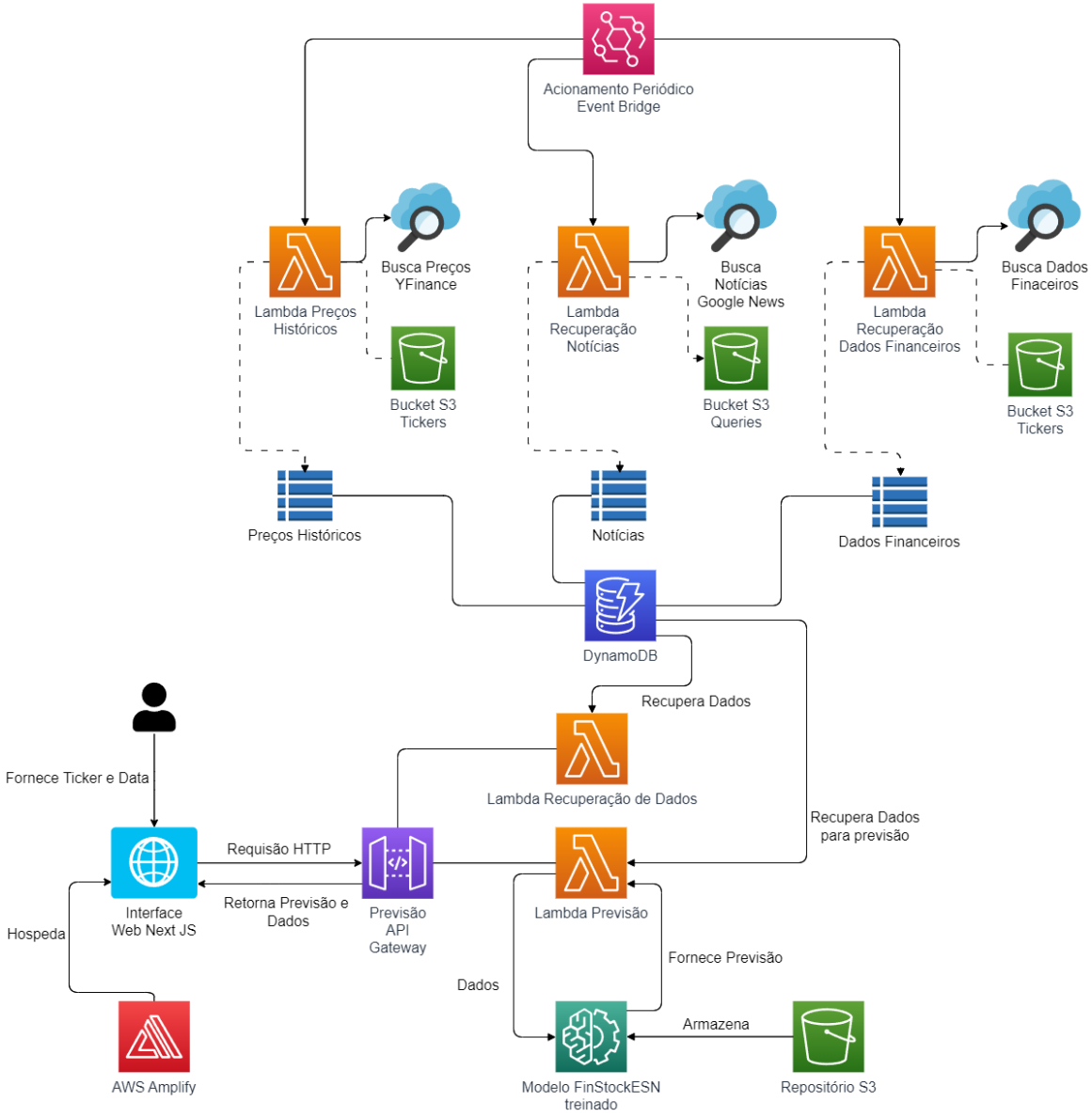


Figura 21 – Arquitetura Implementada na AWS para a Aplicação Web.

6.7.1 Backend

O backend é composto por vários serviços. Abaixo estão detalhados tais serviços sob o ponto de vista da implementação.

Implementação do Serviço de Coleta de Preços Históricos das Ações

A implementação deste serviço foca na coleta de dados históricos de preços de ações, utilizando a biblioteca yfinance e armazenando as informações no AWS DynamoDB. A função é estruturada como uma Lambda Preços Históricos na AWS (mostrada na figura 21), detalhada abaixo:

O serviço segue o seguinte fluxo de execução:

1. **Inicialização e Verificação de Parâmetros**: A função inicia verificando os parâmetros de entrada (num_dias, start_date, end_date) e valida se estão corretos e dentro das regras estabelecidas, como a limitação do intervalo de datas.
2. **Leitura de Tickers do Arquivo CSV no S3**: Utiliza o bucket S3 da AWS para acessar e ler um arquivo CSV, que contém os tickers das ações.
3. **Coleta de Dados Históricos de Ações**: Para cada ticker, a função faz uso da biblioteca yfinance (Yahoo Finance) para obter dados do mercado financeiro, iterando sobre os dados e coletando preços de abertura, fechamento e volume.
4. **Inserção de Dados no DynamoDB**: Os dados coletados são inseridos em uma tabela do DynamoDB, onde cada item contém informações como ticker, data, preços de abertura e fechamento e volume.

Implementação do Serviço de Coleta e Recuperação de Dados Microeconômicos e Macroeconômicos

A função é estruturada como uma Lambda Recuperação de Dados Financeiros na AWS (mostrada na figura 21).

Para a coleta de dados microeconômicos:

1. **Acesso a Lista de Empresas**: O serviço inicia acessando uma lista de empresas armazenada no S3 em um CSV.
2. **Extração de Dados Específicos**: A função coleta informações das empresas utilizando os relatório fornecidos no portal de dados da CVM.
3. **Armazenamento**: Os dados coletados são inseridos em uma tabela Dados Financeiros do DynamoDB.

Para a coleta de dados macroeconômicos:

1. **Seleção de Indicadores**: Definição dos indicadores macroeconômicos relevantes, como taxas de juros, inflação, taxas de desemprego e PIB, indicados em 2.3.1.
2. **Integração com Fontes de Dados**: Conexão com fontes externas, nesse caso a API do Banco Central.
3. **Processamento e Armazenamento**: Os dados coletados são processados e armazenados no DynamoDB.

4. **Retorno dos Dados**: A função Lambda é configurada para responder com os dados no formato JSON. Ela pode ser invocada diretamente para a coleta de notícias ou através de uma integração com a AWS API Gateway para a recuperação dos dados.

Implementação do Serviço de Coleta e Recuperação de Notícias

A lógica do serviço está encapsulada em uma função Lambda Recuperação de Notícias (mostrada na figura 21), que manipula tanto a coleta quanto a recuperação de notícias. A função determina a operação a ser executada com base nos parâmetros de entrada.

A seguir, descrevemos a estrutura e o fluxo de execução da ferramenta. A coleta de notícias é realizada em várias etapas:

1. **Leitura de Arquivo CSV no S3**: A função acessa um bucket S3 para ler um arquivo CSV contendo consultas relacionadas a ações específicas. Exemplos de consulta é *intitle:Petrobras OR intitle:PETR3 OR intitle:PETR4* que busca notícias que possuem em seu título um dos termos Petrobras, PETR3 ou PETR4.

2. **Execução de Consultas no Google News**: Utilizando uma biblioteca que se conecta ao Google News Feed, a função busca notícias com base nas consultas lidas do arquivo CSV.

3. **Salvamento de Notícias no DynamoDB**: As notícias coletadas são armazenadas em uma tabela do DynamoDB, com informações de título, descrição, URL, data de publicação e fonte.

O serviço também implementa um filtro de tempo opcional para limitar a coleta a um período específico.

A recuperação de notícias é processada da seguinte maneira:

1. **Consulta no DynamoDB**: A função realiza consultas no DynamoDB com base no ticker da ação e, opcionalmente, numa data específica.

2. **Formatação dos Dados**: Os dados recuperados são formatados e preparados para serem retornados, incluindo a separação da data de publicação e da fonte da notícia.

3. **Retorno das notícias**: Nesse caso também, a função Lambda é configurada para responder com os dados no formato JSON. Ela pode ser invocada diretamente para a coleta de notícias ou através de uma integração com a AWS API Gateway para a recuperação das notícias.

Implementação do Serviço de Previsão de Preços de Ações

Esta seção detalha a implementação do serviço de previsão de preços de ações, que é executado como uma função Lambda Previsão na AWS (mostrada na figura 21).

São realizadas as seguintes etapas:

1. A função começa recebendo parâmetros via método GET, incluindo o ticker da ação e a data inicial para a previsão. Esses parâmetros são validados, garantindo que sejam fornecidos de forma correta e que a data esteja no formato adequado.

2. Após a validação dos parâmetros, a função estabelece uma conexão com o DynamoDB para buscar os dados históricos do ticker especificado. Isso é feito utilizando consultas que filtram os dados com base no ticker e na data.

3. O serviço utiliza o modelo LiESN (FinStockESN) e Scalers (para normalização dos dados) que são carregados do bucket do S3. Esses modelos foram previamente treinados e armazenados no formato adequado.

4. Com os dados preparados e normalizados, o modelo de previsão é aplicado. As previsões são feitas em lote, e os resultados são transformados de volta para a escala original usando o scaler inverso. Como mencionado na especificação, o scaler é o mesmo utilizado no treinamento, por esse motivo, seu armazenamento.

5. Além de prever os preços com base nos dados históricos, a função estende as previsões para dias futuros, conforme especificado no parâmetro de previsão. Isso envolve a atualização do conjunto de dados com as previsões mais recentes e a realização de novas previsões.

6. Os resultados das previsões são formatados e retornados via API Gateway. Isso inclui informações sobre a data da previsão, o ticker, o preço previsto e, quando disponível, o preço real para comparação.

6.7.2 Implementação do Frontend em Next.js

A implementação do frontend da aplicação foi realizada utilizando o Next.js, um framework baseado em React. O serviço, após sua completa implementação, foi hospedado na AWS Amplify, para torna-se acessível publicamente. A interface do usuário foi dividida em quatro seções principais para facilitar a navegação e a apresentação das informações: Geral, Notícias, Previsões e Dados Financeiros. Cada seção foi idealizada para destacar as principais informações fornecidas pelo serviço.

Geral

Esta seção reúne todas as informações das demais seções em uma única tela, proporcionando uma visão abrangente dos serviços oferecidos pela aplicação.

Figura 22 – Tela Geral - A interface agrega todas as informações essenciais em um único local, permitindo ao usuário um acesso rápido às notícias, previsões e dados financeiros.

Notícias

A seção de Notícias apresenta as últimas atualizações do mercado financeiro para cada ação e data selecionada, com uma interface que permite aos usuários percorrer diferentes artigos. Ao clicar na notícia, ele é redirecionado para o portal que a publicou.

Figura 23 – Tela de Notícias - Exibe as notícias para cada ação e data.

Previsões

A seção de Previsões é dedicada à visualização de dados previstos pelo modelo, com o

gráficos que mostram as tendências do mercado e as previsões de preços de ações. Além disso, também é exibida uma tabela com a previsão de crescimento dos próximos dias.

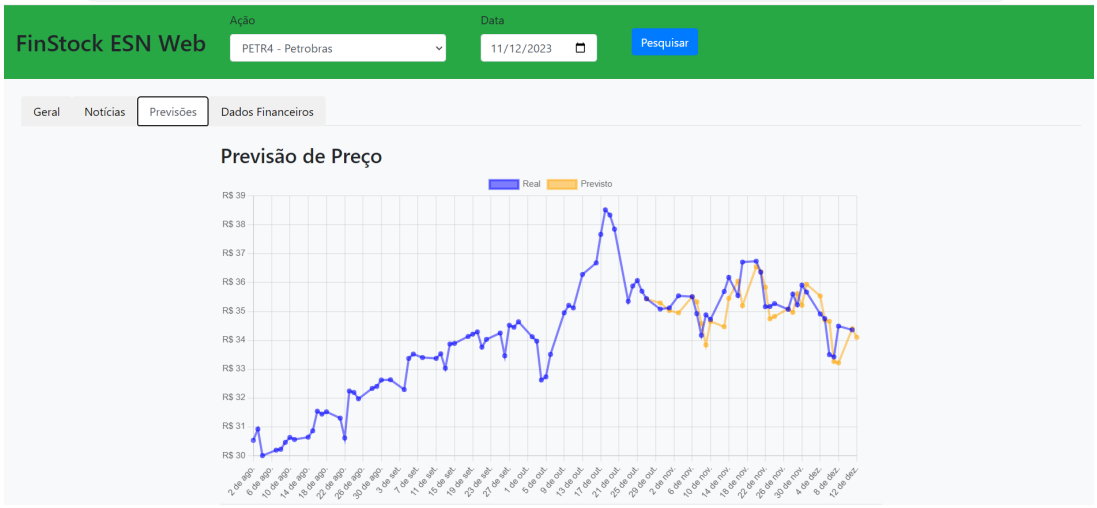
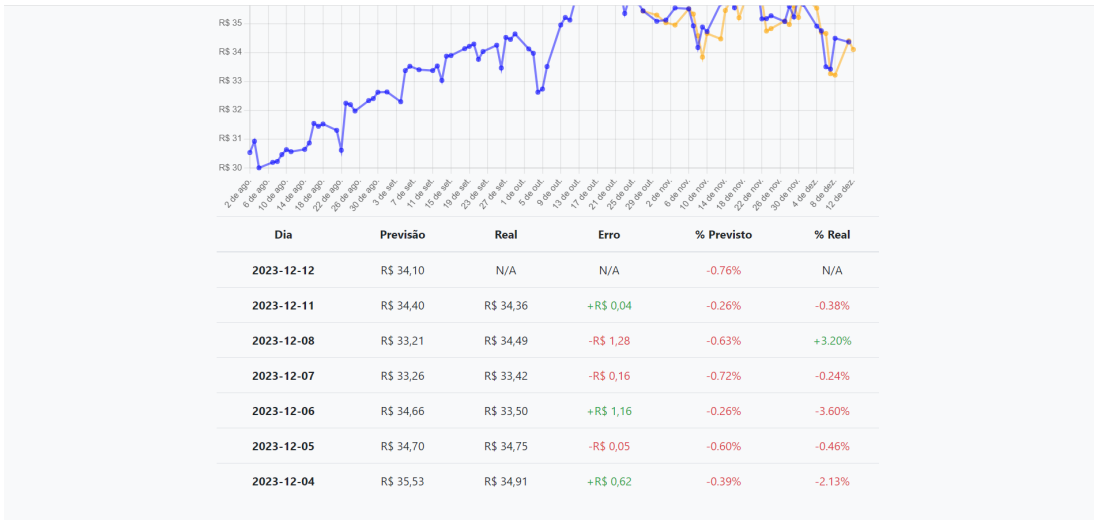


Figura 24 – Tela de Previsões - Oferece uma representação gráfica das previsões de mercado



Dia	Previsão	Real	Erro	% Previsto	% Real
2023-12-12	R$ 34,10	N/A	N/A	-0.76%	N/A
2023-12-11	R$ 34,40	R$ 34,36	+R$ 0,04	-0.26%	-0.38%
2023-12-08	R$ 33,21	R$ 34,49	-R$ 1,28	-0.63%	+3.20%
2023-12-07	R$ 33,26	R$ 33,42	-R$ 0,16	-0.72%	-0.24%
2023-12-06	R$ 34,66	R$ 33,50	+R$ 1,16	-0.26%	-3.60%
2023-12-05	R$ 34,70	R$ 34,75	-R$ 0,05	-0.60%	-0.46%
2023-12-04	R$ 35,53	R$ 34,91	+R$ 0,62	-0.39%	-2.13%

Figura 25 – Continuação Tela de Previsões - Oferece uma tabela de previsão

Dados Financeiros

Por fim, a seção de Dados Financeiros concentra-se em fornecer informações detalhadas sobre aspectos financeiros específicos, como receita, lucro, fluxo de caixa e patrimônio líquido. Além de dados macroeconômicos.

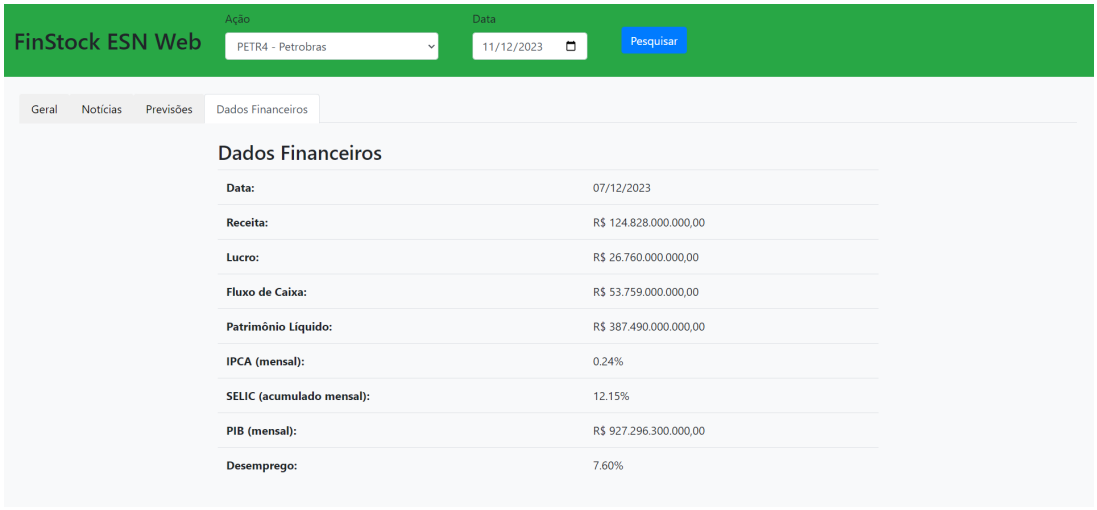


Figura 26 – Tela de Dados Financeiros - Apresenta dados financeiros detalhados, permitindo aos usuários acessar informações sobre a empresa selecionada.

7 Considerações Finais

Este trabalho apresentou o desenvolvimento e a avaliação do modelo FinSTOCKESN-BR, que tem como principal objetivo a avaliação do modelo de previsão proposto e sua influência no mercado de ações brasileiro, integrado com análise de sentimentos, dados financeiros e macroeconômicos. Através de uma série de experimentos, foi avaliado o desempenho do modelo sob diferentes configurações, buscando entender a relevância de cada componente na eficácia da abordagem proposta.

Durante a etapa de avaliação das diferentes entradas do modelo, os experimentos demonstraram a eficácia do Time2Vec como técnica para realizar a codificação temporal das medidas, e foi possível chegar a conclusão de que o melhor conjunto de entradas para o modelo é a junção dos indicadores econômicos da empresa e do país junto com o histórico de preços. Apesar dos esforços despendidos, a agregação dos sentimentos das notícias não levou a uma melhora dos resultados do modelo de forma significativa. Os melhores resultados alcançados contam com a remoção dos dados de sentimento.

Durante este trabalho, conseguimos aplicar efetivamente o conhecimento adquirido ao desenvolver uma aplicação web. Essa aplicação permite que até mesmo usuários sem experiência possam acessar os resultados do modelo de previsão criado para empresas de seu interesse. Com isso, eles podem visualizar de forma intuitiva e clara as previsões de preço do mercado financeiro.

Para tornar a arquitetura da aplicação mais escalável e modular para possíveis expansões futuras, foi usado o paradigma dos microserviços baseados em nuvem. Implementar o sistema dessa forma exigiu uma análise aprofundada dos vários serviços disponíveis na AWS e de como eles interagem entre si.

7.1 Perspectivas de Continuidade

Existem algumas frentes possíveis em que é possível expandir o trabalho realizado.

Para avançar com o desenvolvimento do modelo, explorar o uso de modelos de linguagem em grande escala (LLMs), como GPT-3.5 e GPT-4, é uma direção empolgante. Esses modelos têm potencial para oferecer análises mais detalhadas do que as técnicas tradicionais de análise de sentimentos. No momento, implementar esses modelos enfrenta desafios devido aos altos custos das soluções comerciais e à falta de recursos para execução local eficiente. No entanto, as recentes inovações em recursos computacionais e serviços em nuvem poderão resolver esses problemas em breve.

É possível, também, realizar uma expansão do conjunto de dados no modelo

FinSTOCKESN-BR. Isso envolveria buscar uma forma de melhorar o desempenho dos dados de sentimento através do aumento no volume de dados, talvez até incorporando dados de mercados internacionais. Também seria de particular interesse realizar uma análise aprofundada sobre os efeitos de diferentes fontes de notícias diferentes, como redes sociais, blogues de profissionais financeiros ou fóruns online de finanças e investimento.

Enfim, muitas possibilidades poderão ser exploradas em um trabalho futuro.

Referências

AKBIK, A.; BLYTHE, D.; VOLLGRAF, R. Contextual string embeddings for sequence labeling. In: *COLING 2018, 27th International Conference on Computational Linguistics*. [S.l.: s.n.], 2018. p. 1638–1649. Citado na página 53.

ALMEIDA, R. J. A. *LeIA - Léxico para Inferência Adaptada*. [S.l.]: GitHub, 2018. <https://github.com/rafjaa/LeIA>. Citado na página 53.

ALYOUSEF, H. S. Structure of research article abstracts in political science: A genre-based study. *SAGE Open*, v. 11, n. 3, 2021. Disponível em: <https://doi.org/10.1177/21582440211040797>. Citado na página 32.

ARRATIA, A. et al. Sentiment analysis of financial news: Mechanics and statistics. In: _____. *Data Science for Economics and Finance: Methodologies and Applications*. Cham: Springer International Publishing, 2021. p. 195–216. ISBN 978-3-030-66891-4. Disponível em: <https://doi.org/10.1007/978-3-030-66891-4_9>. Citado na página 34.

B3. *Uma análise da evolução dos investidores na B3*. 2023. Acessado em: 9 de Setembro, 2023. Citado 2 vezes nas páginas 6 e 12.

BAI, Y.-T. et al. Nonstationary time series prediction based on deep echo state network tuned by bayesian optimization. *Mathematics*, v. 11, n. 6, p. 1503, 2023. Disponível em: <https://www.mdpi.com/2227-7390/11/6/1503>. Citado na página 37.

BARONI, M.; DINU, G.; KRUSZEWSKI, G. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, 2014. p. 238–247. Disponível em: <https://aclanthology.org/P14-1023>. Citado na página 29.

BARROS, M. R. de et al. Embracing data irregularities in multivariate time series with recurrent and graph neural networks. In: NALDI, M. C.; BIANCHI, R. A. C. (Ed.). *Intelligent Systems*. Cham: Springer Nature Switzerland, 2023. p. 3–17. ISBN 978-3-031-45368-7. Citado 3 vezes nas páginas 6, 36 e 42.

BOJANOWSKI, P. et al. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016. Citado na página 31.

BOX, G. E. et al. *Time Series Analysis: Forecasting and Control*. 5. ed. New Jersey: Wiley, 2016. Citado na página 17.

BRASIL, B. C. do. *Indicadores Econômicos Selecionados*. 2023. Banco Central do Brasil. Disponível em: <https://www.bcb.gov.br/estatisticas>. Acesso em: 8 out 2023. Citado 2 vezes nas páginas 21 e 22.

CAMBRIA, E.; BROWN, A. Analysis of news sentiments using natural language processing techniques. *Springer*, 2016. Citado na página 34.

CAO, e. a. Stock price movement prediction based on stocktwits investor sentiment. *NCBI*, 2022. Citado na página 15.

DAMODARAN, A. *Avaliação de Investimentos: Ferramentas e Técnicas para a Determinação do Valor de Qualquer Ativo*. 3. ed. Nova York: Wiley, 2012. Citado na página 21.

ECONOMIA, M. da. *Ministério da Economia apresenta nova grade de parâmetros macroeconômicos*. 2023. Governo do Brasil. Disponível em: <https://www.gov.br/economia/pt-br/assuntos/noticias/2023/nova-grade-de-parametros-macroeconomicos>. Acesso em: 8 out 2023. Citado na página 22.

Equipe Toro Investimentos. *Empresas Listadas na Bolsa B3*. 2023. <https://blog.toroinvestimentos.com.br/bolsa/empresas-listadas-bolsa-b3/>. Acessado em 13 de novembro de 2023. Citado 2 vezes nas páginas 8 e 67.

ESEN, F. Classification by keyword search and text mining as a decision analysis method. In: . [S.l.: s.n.], 2022. Citado na página 33.

GILBERT, C. H. E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*. [S.l.: s.n.], 2014. Citado na página 53.

GOLDBERG, Y. *Neural Network Methods in Natural Language Processing*. 1. ed. Toronto: Morgan & Claypool Publishers, 2017. ISBN 1627052984. Citado na página 28.

GOODFELLOW YOSHUA BENGIO, A. C. I. *Deep Learning*. 1. ed. Cambridge, MA: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na página 26.

GRAHAM, J. Z. B. *The Intelligent Investor: The Definitive Book on Value Investing*. Revised edition. New York: HarperCollins, 2006. Citado na página 20.

GUARINO, N. The ontological level. In: _____. *Philosophy and the Cognitive Science*. Vienna: Holder-Pivhler-Tempsky, 1995. p. 443–456. Disponível em: <http://wiki.loa-cnr.it/Papers/OntLev.pdf>. Acesso em: 2 jan. 2012. Citado na página 31.

HARTMANN, N. et al. *Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks*. 2017. Citado 2 vezes nas páginas 55 e 70.

HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: principles and practice*. 3. ed. Melbourne, Australia: OTexts, 2021. Disponível em: <https://otexts.com/fpp3/index.html>. Acesso em: 20 set 2023. Citado na página 17.

HYNDMAN, R. J.; KOEHLER, A. B. Another look at measures of forecast accuracy. *International Journal of Forecasting*, v. 22, n. 4, p. 679–688, 2006. ISSN 0169-2070. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169207006000239>. Citado na página 18.

IBGE. *Painel de Indicadores*. 2023. Instituto Brasileiro de Geografia e Estatística. Disponível em: <https://www.ibge.gov.br/indicadores>. Acesso em: 8 out 2023. Citado na página 21.

JAEGER, H. The "echo state"approach to analysing and training recurrent neural networks - with an erratum note'. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, v. 148, 01 2001. Citado na página 26.

JAIN, A. *Understanding The Basics of Time Series Forecasting*. Analytics Vidhya, 2021. Disponível em: <https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/>. Citado 2 vezes nas páginas 17 e 18.

JIANG, Z. et al. Text classification using novel term weighting scheme-based improved tf-idf for internet media reports. *Mathematical Problems in Engineering*, v. 2021, p. 1–30, 2021. Disponível em: <https://doi.org/10.1155/2021/6619088>. Citado na página 32.

KAZEMI, S. M. et al. *Time2Vec: Learning a Vector Representation of Time*. 2019. Citado na página 37.

KIM, T.; KING, B. R. Time series prediction using deep echo state networks. *Neural Computing and Applications*, v. 32, p. 17769–17787, 2020. Disponível em: <https://doi.org/10.1007/s00521-020-04948-x>. Citado na página 37.

KUMAR, A. *Different types of Time-series Forecasting Models*. Analytics Yogi, 2023. Disponível em: <https://vitalflux.com/different-types-of-time-series-forecasting-models/>. Acesso em: 3 jul. 2023. Citado na página 17.

LEVINE, R.; ZERVOS, S. Stock markets, banks, and economic growth. *American Economic Review*, v. 88, n. 3, p. 537–558, 1998. Citado na página 19.

LIN, X.; YANG, Z.; SONG, Y. Short-term stock price prediction based on echo state networks. *Expert Systems with Applications*, v. 36, n. 3, Part 2, p. 7313–7317, 2009. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417408006519>. Citado na página 14.

LING, W. et al. Two/too simple adaptations of word2vec for syntax problems. In: . [S.l.: s.n.], 2015. Citado 3 vezes nas páginas 6, 30 e 31.

LIU, J. et al. Financial data forecasting using optimized echo state network. In: CHENG, L.; LEUNG, A. C. S.; OZAWA, S. (Ed.). *Neural Information Processing*. Cham: Springer International Publishing, 2018. p. 138–149. ISBN 978-3-030-04221-9. Citado na página 13.

LO, A. The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. 10 2004. Citado na página 13.

LUKOŠEVIČIUS, M. A practical guide to applying echo state networks. In: _____. *Neural Networks: Tricks of the Trade: Second Edition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 659–686. Disponível em: <https://www.ai.rug.nl/minds/uploads/PracticalESN.pdf>. Citado na página 26.

MALKIEL, B. G.; FAMA, E. F. Efficient capital markets: A review of theory and empirical work*. *The Journal of Finance*, v. 25, n. 2, p. 383–417, 1970. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1970.tb00518.x>. Citado na página 13.

MEHTA, P.; PANDYA, S. A review on sentiment analysis methodologies, practices and applications. v. 9, 09 2020. Citado na página 34.

NETO, A. A. *Mercado Financeiro*. 12. ed. São Paulo: Editora Atlas, 2014. Citado na página 20.

PAIVA, F. C. L. Assimilating sentiment analysis in reinforcement learning for intelligent trading. *Tese de Mestrado em Engenharia de Computação – Escola Politécnica, Universidade de São Paulo. São Paulo*, 2023. Citado na página 36.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *Empirical Methods in Natural Language Processing (EMNLP)*. [s.n.], 2014. p. 1532–1543. Disponível em: <http://www.aclweb.org/anthology/D14-1162>. Citado 2 vezes nas páginas 6 e 31.

POVOA, A. *Valuation: Como Precificar Acoes*. 1. ed. Rio de Janeiro: GEN Atlas, 2012. Citado na página 21.

PéREZ-ENCISO, Z.; LAURA. A guide for using deep learning for complex trait genomic prediction. *Genes*, v. 10, p. 553, 07 2019. Citado 3 vezes nas páginas 6, 24 e 25.

SANTOS, L. L. FINBERTPTBR: Análise de Sentimentos de Textos em Português Referente ao Mercado Financeiro. *TCC (Graduação em Engenharia de Computação) – Escola Politécnica, Universidade de São Paulo. São Paulo*, 2022. Citado 4 vezes nas páginas 34, 36, 38 e 53.

SEBASTIAN, W.; ISA, S. M. Stock price prediction using bert and word2vec: Sentiment analysis. *International Journal of Engineering Trends and Technology (IJETT)*, v. 8, n. 9, 2020. Disponível em: <http://www.warse.org/IJETER/static/pdf/file/ijeter85892020.pdf>. Citado na página 30.

SOH, H.; DEMIRIS, Y. Spatio-temporal learning with the online finite and infinite echo-state gaussian processes. *IEEE transactions on neural networks and learning systems*, v. 26, 06 2014. Citado 2 vezes nas páginas 6 e 27.

SUTOR, P. et al. Metaconcepts: Isolating context in word embeddings. In: . [S.l.: s.n.], 2019. p. 544–549. Citado 2 vezes nas páginas 6 e 29.

TEXTBLOB: Simplified Text Processing. <https://textblob.readthedocs.io/en/dev/>. Accessed: December 6, 2023. Citado na página 53.

Trierweiler Ribeiro, G. et al. Novel hybrid model based on echo state neural network applied to the prediction of stock price return volatility. *Expert Systems with Applications*, v. 184, p. 115490, 2021. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417421009003>. Citado na página 13.

YADAV, A. et al. Sentiment analysis of financial news using unsupervised approach. *ScienceDirect Procedia Computer Science*, v. 167, p. 589–598, 2020. Disponível em: <https://doi.org/10.1016/j.procs.2020.03.325>. Citado na página 34.

Apêndices

APÊNDICE A – Exemplificação Dados com Gráficos

Para ilustrar os dados coletados e consolidados, mostra-se os dados da empresa WEG, uma das principais do índice Ibovespa.

A.1 Dados Fundamentalistas e Preço da WEG

Abaixo estão os gráficos representando o preço e dados fundamentalistas da WEG, como receita, lucro, patrimônio líquido e fluxo de caixa operacional. Estes gráficos oferecem uma visão da performance financeira da empresa ao longo do tempo.

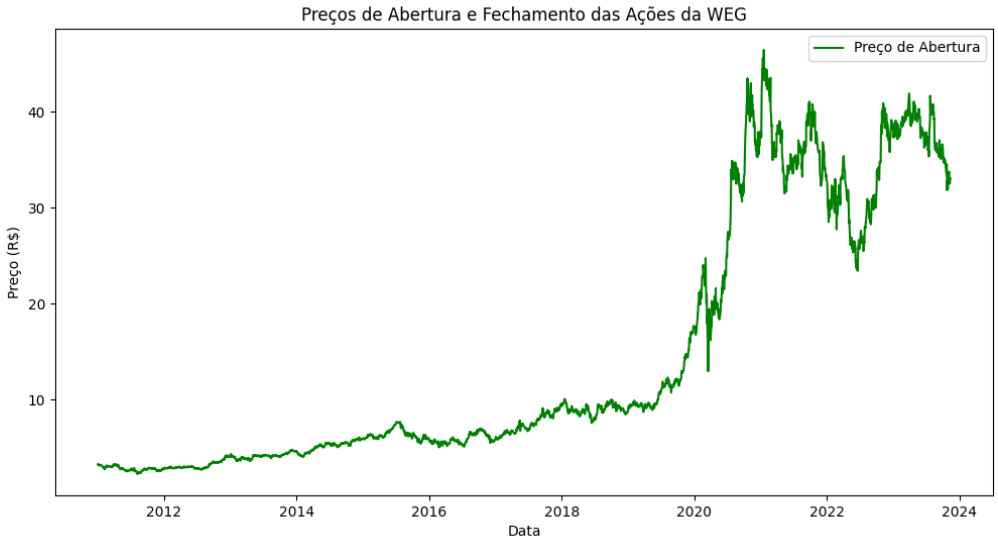


Figura 27 – Evolução do Preço da WEG - R$ Mil

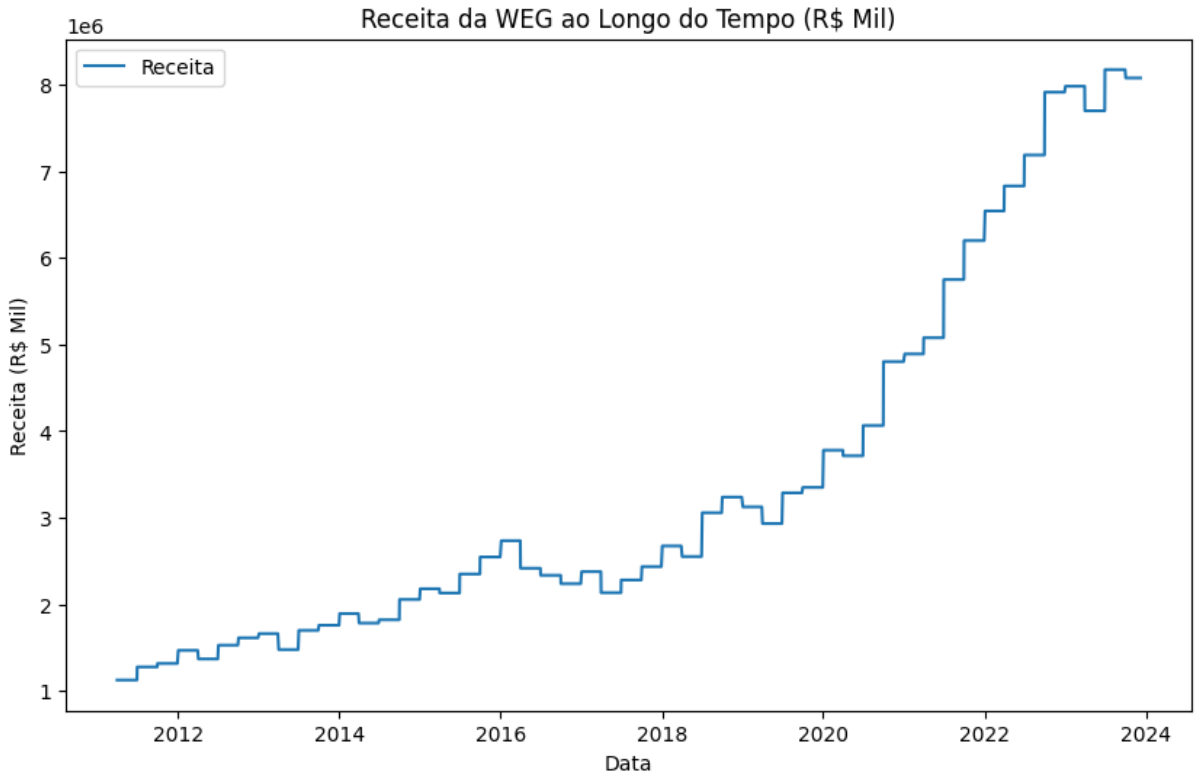


Figura 28 – Evolução da Receita da WEG - R$ Mil

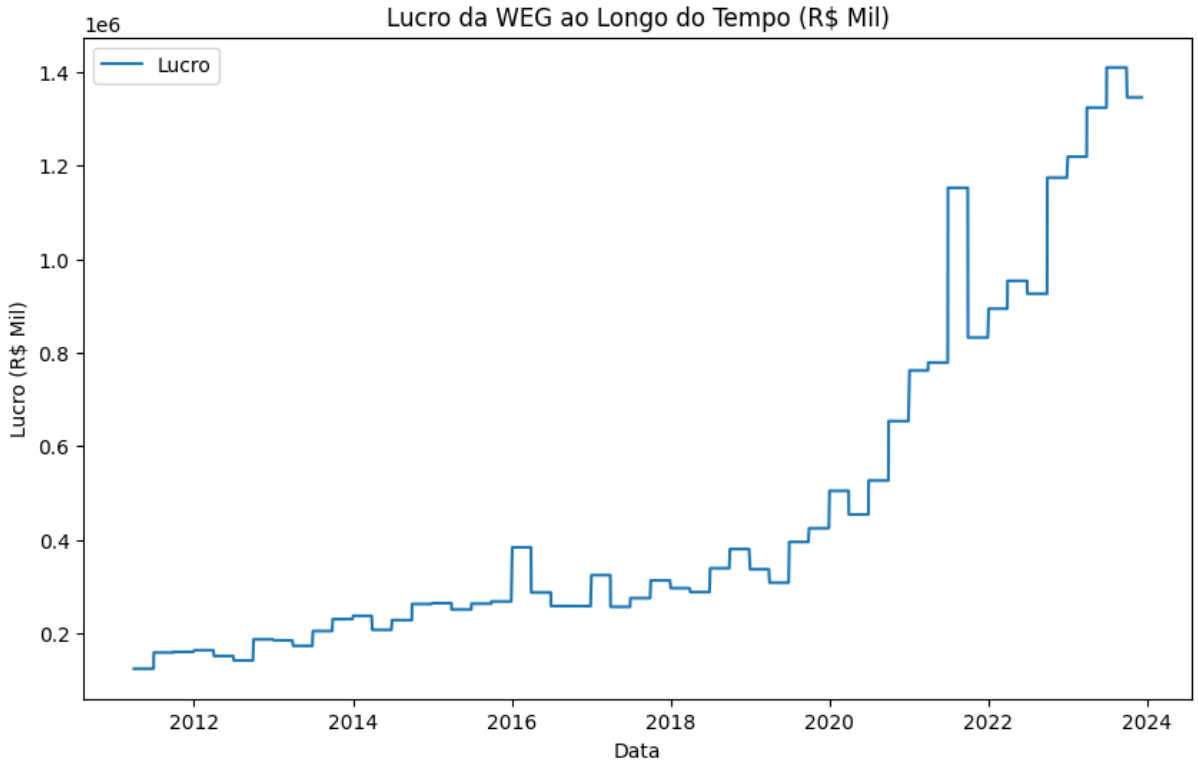


Figura 29 – Evolução do Lucro da WEG - R$ Mil

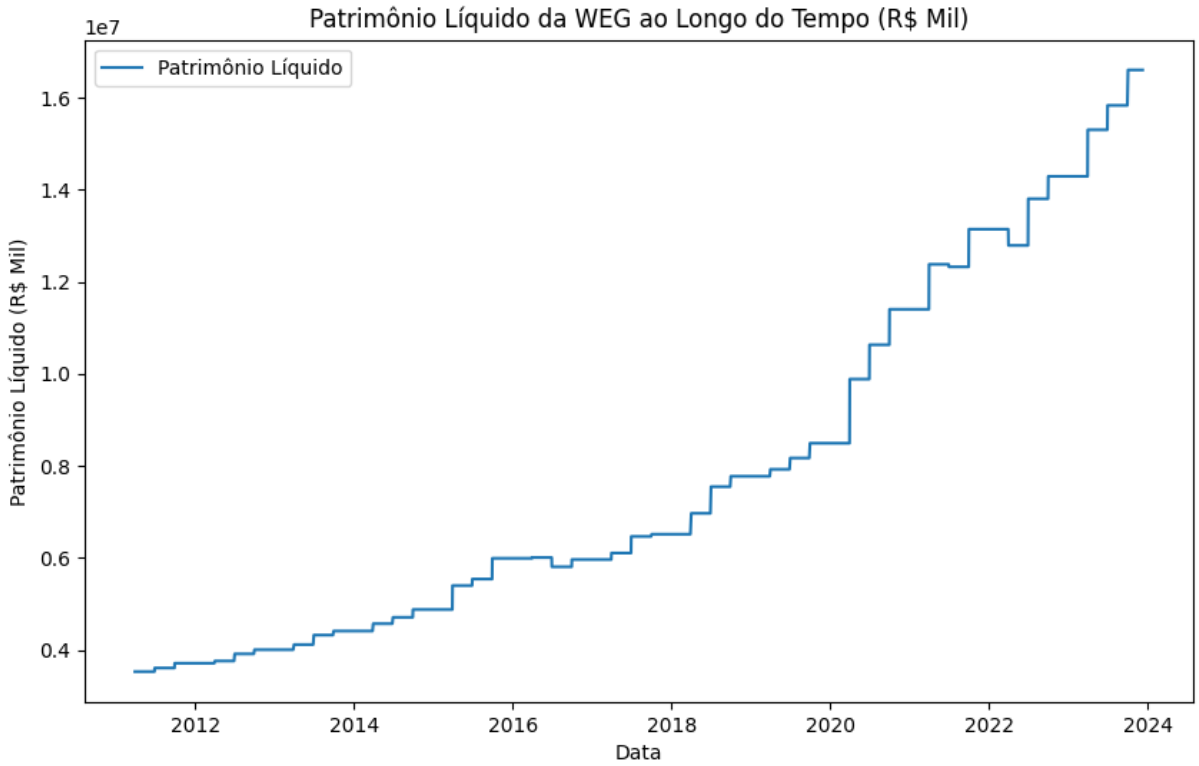


Figura 30 – Evolução do Patrimônio Líquido da WEG - R$ Mil

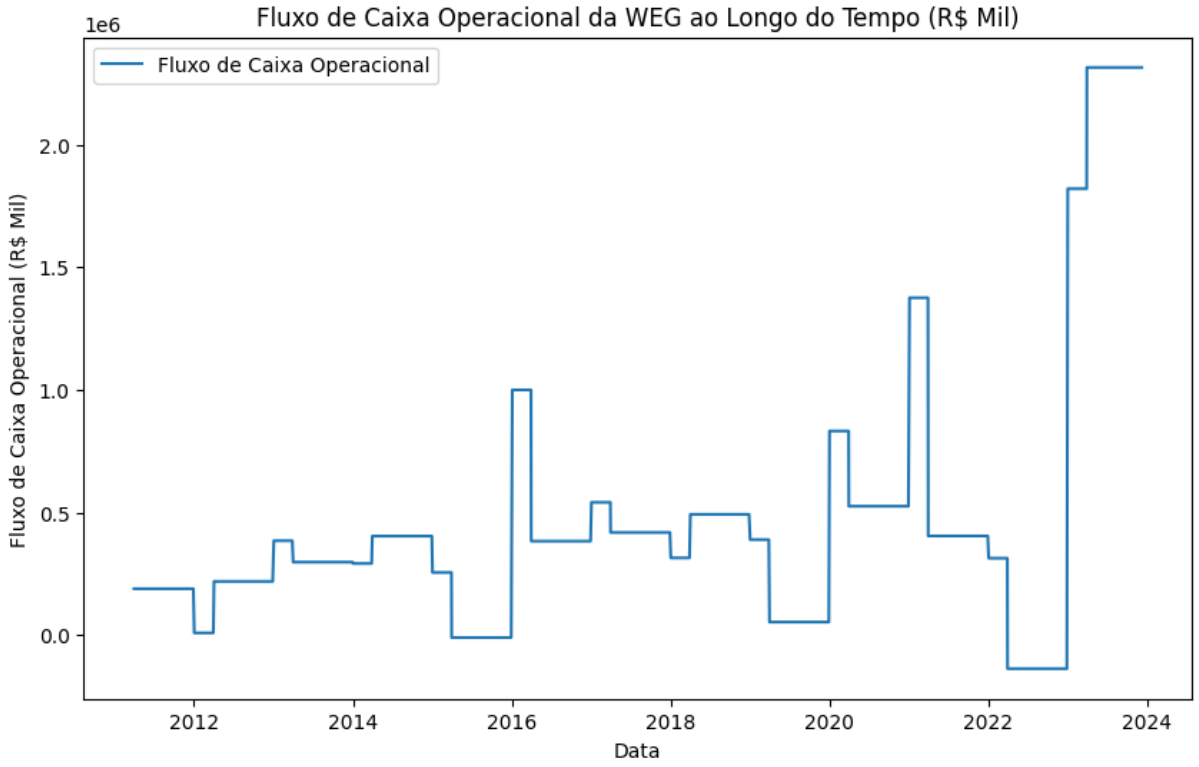


Figura 31 – Evolução do Fluxo de Caixa Operacional da WEG - R$ Mil

A.2 Dados Macroeconômicos

Abaixo estão os gráficos representando os indicadores macroeconômicos, IPCA, SELIC, PIB e Taxa de Desemprego

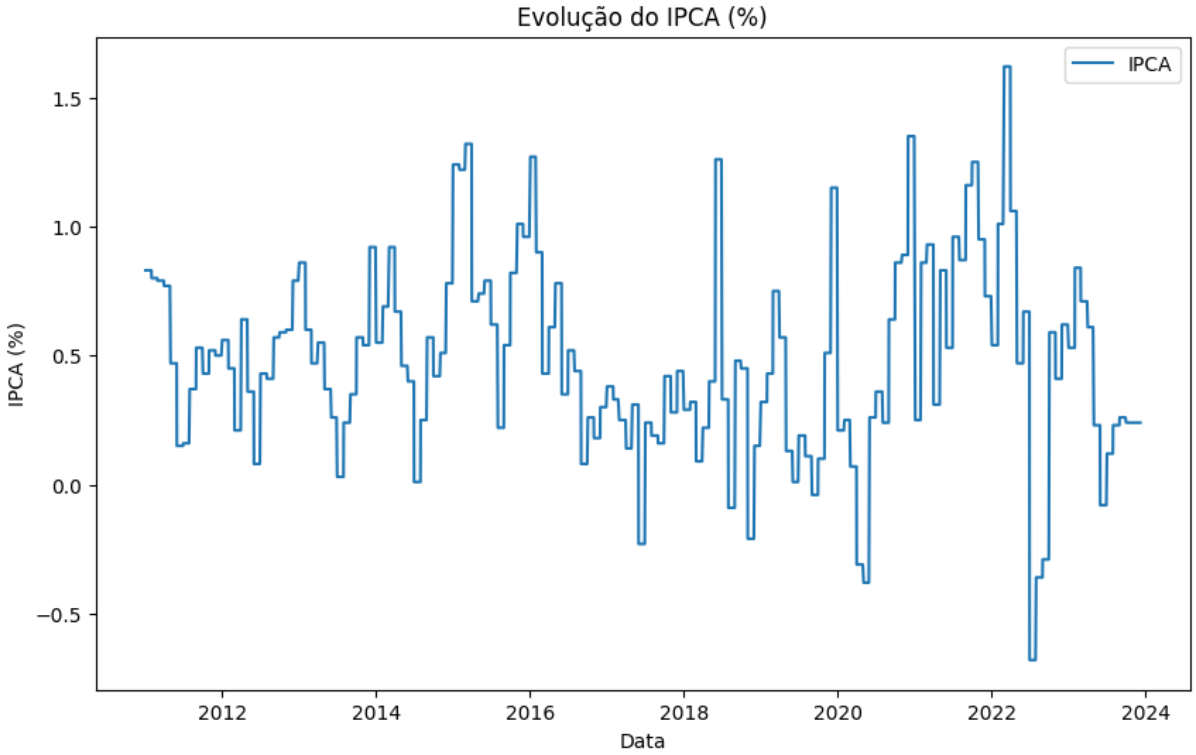


Figura 32 – Evolução do IPCA - Mensal

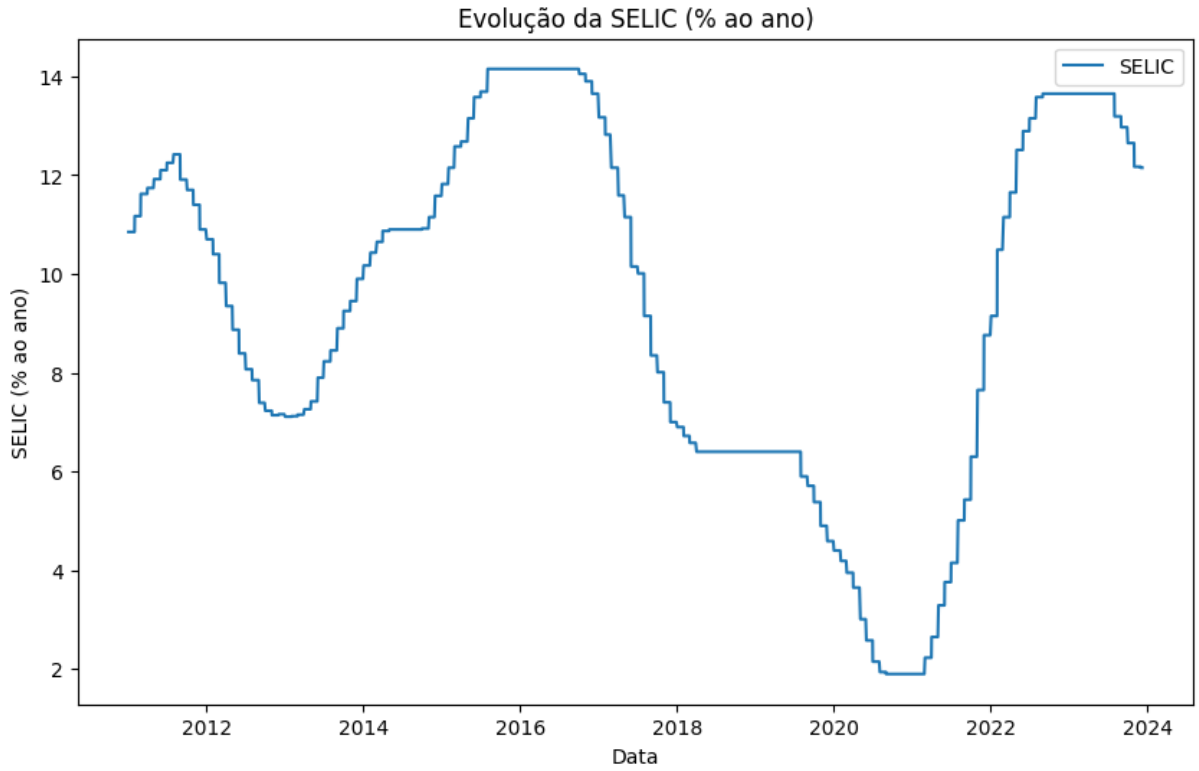


Figura 33 – Evolução da SELIC - Mensal Acumulado

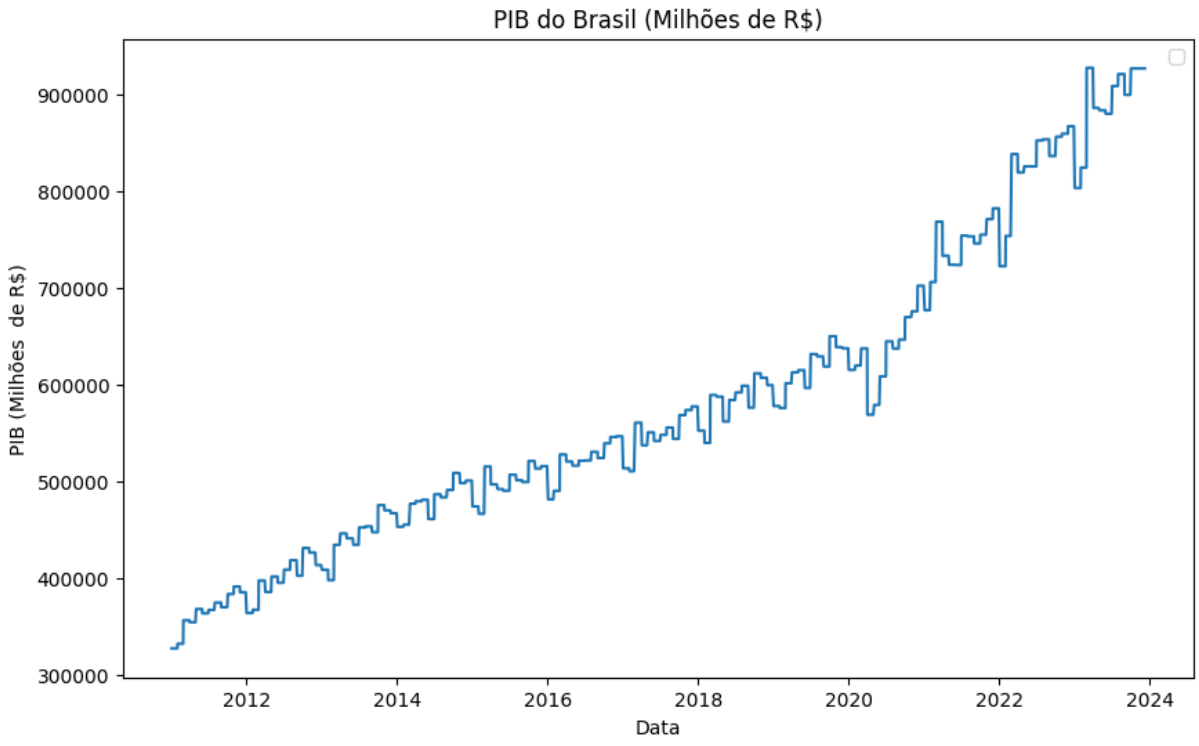


Figura 34 – Evolução do PIB - Mensal - R$ Milhão

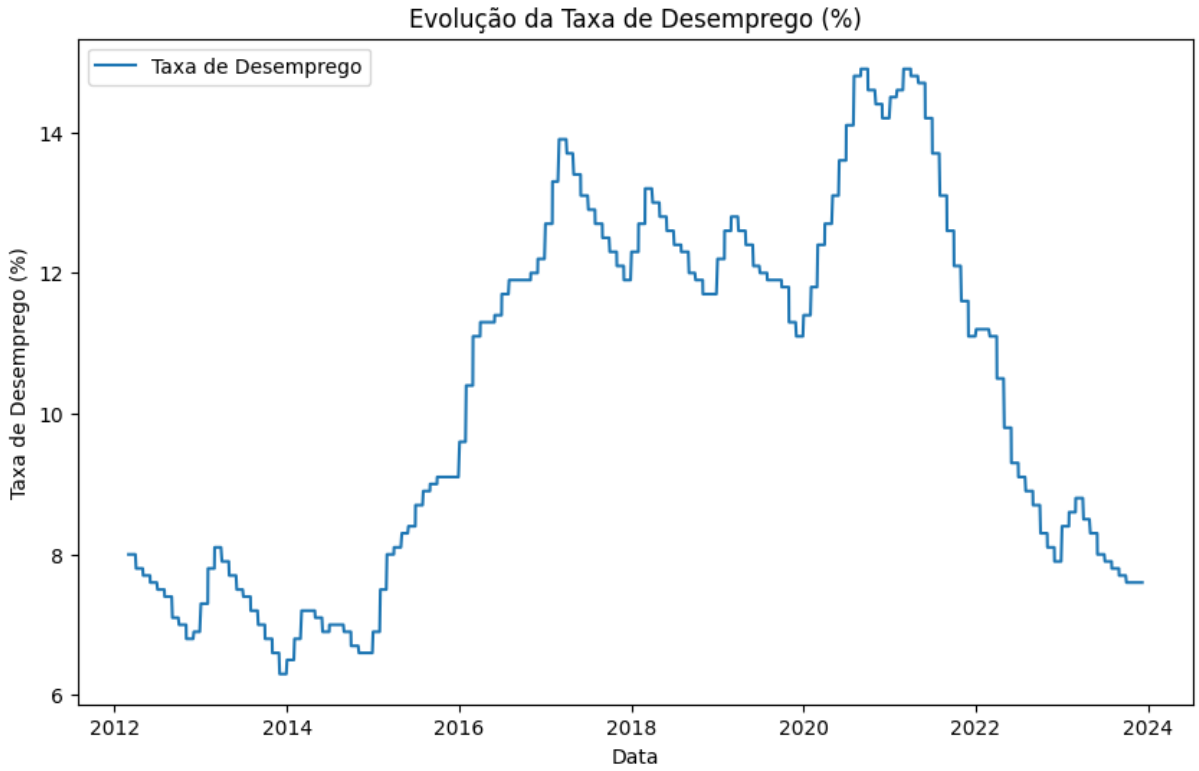


Figura 35 – Evolução da Taxa de Desemprego - Mensal