Pontifícia Universidade Católica
do Rio de Janeiro

Thiago Levis Alambert Rodrigues

Predição de Séries Temporais do Mercado de Ações com Algoritmos de Aprendizado de Máquina

PROJETO FINAL DE GRADUAÇÃO

Relatório de projeto final apresentado como requisito parcial para obtenção do grau de Barechal pelo Programa de Engenharia de Computação, do Departamento de Informática da PUC-Rio.

Orientador: Prof. Augusto Cesar Espíndola Baffa

Rio de Janeiro
Dezembro de 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Thiago Levis Alambert Rodrigues

Predição de Séries Temporais do Mercado de Ações com Algoritmos de Aprendizado de Máquina

Relatório de projeto final apresentado como requisito parcial para obtenção do grau de Bacharel pelo Programa de Engenharia de Computação da PUC-Rio. Aprovada pela Comissão Examinadora abaixo:

Prof. Augusto Cesar Espíndola Baffa
Orientador
Departamento de Informática – PUC-Rio

Rio de Janeiro, 9 de Dezembro de 2024

Thiago Levis Alambert Rodrigues

Ficha Catalográfica

Levis Alambert Rodrigues, Thiago

Predição de Séries Temporais do Mercado de Ações com Algoritmos de Aprendizado de Máquina / Thiago Levis Alambert Rodrigues; orientador: Augusto Cesar Espíndola Baffa. – 2024.

80 f: il. color. ; 30 cm

Dissertação (Graduação) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2024.

Inclui bibliografia

1. Informática – Teses. 2. Aprendizado de Máquina. 3. Predição de preços. 4. Séries temporais. 5. Mercado de ações. I. Baffa, Augusto Cesar Espíndola. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 620.11

Agradecimentos

Agradeço de coração à minha família e parentes pelo apoio incondicional ao longo dessa jornada. Aos meus amigos, especialmente aqueles que conheci durante a faculdade, sou grato pelas experiências compartilhadas e pelas memórias construídas. Um agradecimento especial aos professores e funcionários do departamento de Informática da PUC-RIO, que sempre me trataram com respeito e carinho. Em particular, quero expressar minha gratidão aos professores Augusto Baffa, Marcos Villas e Sergio Lifschitz, cujas orientações e apoio foram fundamentais em meu percurso acadêmico. Por fim, agradeço aos meus amigos do laboratório de informática BioBD, que tornaram essa experiência ainda mais especial.

Resumo

Levis Alambert Rodrigues, Thiago; Baffa, Augusto Cesar Espíndola. **Predição de Séries Temporais do Mercado de Ações com Algoritmos de Aprendizado de Máquina**. Rio de Janeiro, 2024. 80p. Dissertação de Graduação – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho apresenta o desenvolvimento de uma ferramenta para predição de séries temporais do mercado de ações utilizando algoritmos de aprendizado de máquina. A volatilidade e a complexidade do mercado financeiro tornam a análise e previsão de preços de ações um desafio significativo, especialmente para investidores iniciantes. O objetivo deste estudo é aplicar técnicas modernas de aprendizado de máquina para identificar padrões em dados históricos e fornecer previsões confiáveis que auxiliem investidores na tomada de decisões estratégicas.

A metodologia envolve a coleta de dados históricos de preços de ações, o processamento dessas séries temporais e a aplicação de algoritmos como NeuralProphet e LSTM. A avaliação do modelo utiliza métricas como Erro Quadrático Médio (RMSE) e Erro Percentual Absoluto Simétrico Médio (sMAPE) para garantir precisão e robustez.

Os resultados demonstram que os algoritmos de aprendizado de máquina podem capturar padrões complexos no mercado de ações, proporcionando previsões que, em alguns casos, superam as técnicas tradicionais. Como contribuição prática, este trabalho fornece insights valiosos que podem ajudar os investidores a tomar decisões mais informadas.

Palavras-chave

Aprendizado de Máquina; Predição de preços; Séries temporais; Mercado de ações.

Abstract

Levis Alambert Rodrigues, Thiago; Baffa, Augusto Cesar Espíndola (Advisor). . Rio de Janeiro, 2024. 80p. Dissertação de Graduação – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

This work presents the development of a tool for predicting time series in the stock market using machine learning algorithms. The volatility and complexity of the financial market make the analysis and forecasting of stock prices a significant challenge, especially for novice investors. The objective of this study is to apply modern machine learning techniques to identify patterns in historical data and provide reliable forecasts that assist investors in making strategic decisions.

The methodology involves collecting historical stock price data, processing these time series, and applying algorithms such as NeuralProphet and LSTM. The model evaluation uses metrics such as Root Mean Square Error (RMSE) and Symmetric Mean Absolute Percentage Error (sMAPE) to ensure accuracy and robustness.

The results demonstrate that machine learning algorithms can capture complex patterns in the stock market, providing forecasts that, in some cases, outperform traditional techniques. As a practical contribution, this work provides valuable insights that can help investors make more informed decisions.

Keywords

Machine Learning; Price Prediction; Time Series; Stock Market.

Sumário

Lista de figuras

Lista de tabelas

Lista de Abreviaturas

IQR – Interquartile Range

tanh – tangente hiperbólica

sMAPE – Symmetric Mean Absolute Percentage Error

MAPE – Mean Absolute Percentage Error

RMSE – Root Mean Squared Error

RNN – Rede Neural Recorrente

ET – Extra Tree

SMA – Média Móvel Simples

ARMA – Autoregressive Moving Average

AR – Autoregressão

SGD – método do gradiente estocástico

AED – Análise exploratória de dados

OHLC – Open,High,Low,Close

1
Introdução

Nos últimos anos, o número de investidores no mercado de ações tem crescido (B3, 2023), refletindo um interesse crescente da população por essa área. No entanto, iniciar investimentos em um campo novo pode ser desafiador, e muitos investidores não conseguem alcançar seus objetivos, resultando em prejuízos (CINTRA, 2022).

Paradoxalmente, a crescente popularidade das apostas online no Brasil, como o "jogo do tigrinho", que já supera em sete vezes o volume de investimentos na Bolsa de Valores (NC, 2024), revela uma preocupante tendência de gasto improdutivo. Dados da Anbima mostram que 22 milhões de brasileiros, equivalentes a 14% da população adulta, estão investindo seu dinheiro em plataformas de apostas não regulamentadas, frequentemente acreditando erroneamente que estão fazendo investimentos financeiros. Este comportamento é amplamente impulsionado pela falta de educação financeira e pela ilusão de ganhos rápidos promovida pelas plataformas de apostas. Em contraste, apenas uma pequena fração da população investe no mercado de capitais, como ações e títulos.

Esse fenômeno pode ser parcialmente explicado pelas barreiras de acesso percebidas no ambiente de investimentos. O mercado financeiro, embora potencialmente lucrativo, exige um certo nível de conhecimento e compreensão para operar de forma eficaz. Para muitas pessoas, esse requisito inicial pode parecer intimidante e complexo, desencorajando-as a investir tempo e esforço no aprendizado necessário. Assim, a simplicidade aparente das apostas online pode ser mais atraente do que enfrentar os desafios do mercado financeiro.

Diante dessas dificuldades, os investidores frequentemente buscam métodos para auxiliá-los na gestão de seus investimentos e na minimização dos riscos. As abordagens mais conhecidas incluem a busca por influenciadores di-

gitais (PURCHIO, 2021), pesquisas em sites e a análise das oscilações passadas para tentar estimar o preço futuro. No entanto, essas estratégias podem ser consideradas superficiais, especialmente no contexto do mercado de ações, que é altamente complexo e dinâmico (KHAIDEM; SAHA; DEY, 2016). O mercado financeiro é influenciado por uma variedade de fatores, como as tendências do mercado, a saúde de uma empresa, notícias e rumores, questões políticas e o calendário econômico, que devem ser cuidadosamente considerados para tomar decisões informadas.

Atualmente, existem estudos que buscam compreender os melhores métodos de machine learning para aumentar a taxa de acurácia na variação dos preços das ações, podendo assim identificar os melhores momentos para realizar uma operação (STRADER et al., 2020). Em Strader (STRADER et al., 2020), é possível observar que redes neurais são melhores para predizer os valores numéricos de índices de mercado de ações, enquanto a classificação é melhor para identificar se o índice do mercado de ações irá subir ou descer. É relevante observar que já existem papers sobre o tema que demonstram ter melhorias na precisão das previsões, porém eles sempre comentam que existe espaço para aperfeiçoamentos (PARMAR et al., 2018).

Portanto, o trabalho aqui apresentado tem como objetivo desenvolver uma ferramenta eficaz para auxiliar os investidores por meio da inteligência artificial e da análise de dados de séries temporais, permitindo identificar as tendências dos ativos transacionados no mercado de ações do Brasil. Para isso, será realizada a previsão do próximo dia do preço da ação, possibilitando que a ferramenta seja usada como suporte na tomada de decisões dos investidores.

2
Situação atual

O uso de machine learning para prever preços de ações no mercado financeiro está se tornando cada vez mais crucial no cenário econômico atual. Os especialistas indicam que, até 2025, a personalização de recomendações de investimento por IA será um dos principais avanços nas instituições financeiras (PIOVEZAN, 2024a). Um exemplo prático dessa tendência é a inteligência artificial "Diana", desenvolvida pela iniciativa Direto ao Tesouro. Através de um chatbot no WhatsApp, a ferramenta interage com os usuários, traçando o perfil do investidor e sugerindo opções de investimento adequadas. Esse recurso é voltado principalmente para investidores iniciantes com pouco ou nenhum conhecimento sobre o mercado de capitais, facilitando o acesso a estratégias personalizadas e mais seguras (PIOVEZAN, 2024b).

De acordo com o artigo (PARMAR et al., 2018), existe uma grande oportunidade para melhorar as abordagens atuais de predição de ações utilizando algoritmos de machine learning. No cenário financeiro atual, onde essas técnicas estão cada vez mais sendo adotadas para prever tendências, os modelos de machine learning aparecem como ferramentas poderosas. Dada a complexidade e a imprevisibilidade do mercado, a aplicação dessas tecnologias pode ser altamente eficaz na otimização das decisões financeiras.

Além disso, destaca-se que o Aprendizado supervisionado oferece abordagens notáveis para a predição de tendências (RAVIKUMAR; SARAF, 2020)(STRADER et al., 2020). No entanto, há distinções importantes. Os modelos de classificação são mais eficazes na previsão da direção do índice da ação, indicando se irá subir ou descer, enquanto o modelo de regressão proporciona uma estimativa do preço de fechamento da ação.

Dessa forma, a predição de preços para ações pode ser integrada com essas ferramentas como a "Diana"para oferecer previsões mais precisas e per-

sonalizadas para investidores iniciantes. Essas previsões poderiam orientar os usuários a fazer escolhas mais informadas, minimizando riscos e maximizando os retornos, mesmo sem um conhecimento aprofundado do mercado financeiro. Com a automatização dessas análises e recomendações, o acesso a estratégias eficazes se tornaria mais acessível e seguro para o público leigo, democratizando ainda mais o investimento.

Sendo assim, a utilização de modelos de inteligência artificial pode ajudar a enfrentar os desafios presentes para os investidores e empresas. Com a capacidade de analisar os dados e identificar correlações que seriam difíceis de serem percebidas por métodos tradicionais, esses modelos oferecem uma vantagem competitiva, podendo oferecer perceção valiosa para os acionistas. Para as empresas, essa aplicação pode ser utilizada para realizar a sua oferta pública inicial (IPO), para determinar a valorização desejada e a quantidade de ações a serem disponibilizadas.

Visto que já existem diversos estudos sobre o assunto, minha abordagem não visa necessariamente criar algo inédito, mas sim buscar aprimorar os métodos e conceitos já estabelecidos, além de apresentar novas possibilidades de aplicação para conseguir uma acurácia maior na movimentação do mercado de ações.

3
Técnicas e Padrões Definidos

3.1
Técnicas

Nesta seção, são apresentadas as principais técnicas utilizadas neste trabalho, com foco na modelagem de séries temporais. Inicialmente, será abordado o modelo , implementado com a ferramenta NeuralProphet[1], que combina conceitos clássicos de estatística com capacidades modernas de aprendizado de máquina para previsão. Em seguida, será explorada a abordagem baseada em redes neurais recorrentes (RNN), com destaque para as redes de memória de curto e longo prazo (LSTM). Essa técnica é projetada para capturar padrões em dados sequenciais, permitindo o aprendizado de dependências temporais tanto recentes quanto distantes.

3.1.1
NeuralProphet: Integração entre Autoregressão e Redes Neurais

O NeuralProphet é um modelo inovador que combina os princípios da autoregressão tradicional com a flexibilidade e o poder de generalização das redes neurais. Essa abordagem híbrida permite capturar tanto padrões lineares quanto não lineares em séries temporais, oferecendo uma solução robusta e escalável para problemas complexos de previsão.

A seguir, serão abordados os conceitos fundamentais que sustentam o modelo NeuralProphet, começando pela série temporal, autoregressão, passando pelas redes neurais e finalizando com uma análise detalhada do modelo.

3.1.1.1
Série Temporal

Uma série temporal é definida como um conjunto de observações organizadas cronologicamente, podendo ou não ser igualmente espaçadas no

[1] https://neuralprophet.com/

tempo, e que apresentam dependência serial, ou seja, uma relação entre os diferentes pontos temporais. Pode-se representar uma série temporal como $S_1, S_2, S_3, \ldots, S_T$, onde T indica o tamanho total da série. Muitos fenômenos em áreas como física, biologia, economia, entre outros, podem ser modelados como séries temporais. A decomposição tradicional de uma série temporal consiste em quatro componentes principais: tendência, ciclo, sazonalidade e ruído (MORETTIN; TOLOI, 1987).

A tendência descreve o comportamento geral da série ao longo do tempo, refletindo padrões de longo prazo, como crescimento, declínio ou estabilidade. Esses padrões podem ser modelados de forma constante, linear ou até mesmo quadrática, dependendo da natureza do fenômeno.

Os ciclos representam oscilações que se sobrepõem à tendência, caracterizadas por movimentos repetitivos de subida e descida. No entanto, diferentemente da sazonalidade, os ciclos não possuem periodicidade fixa. Exemplos comuns incluem ciclos econômicos ou climáticos.

A sazonalidade corresponde a variações que ocorrem em intervalos regulares e previsíveis, como as vendas em datas específicas do ano ou flutuações diárias. A principal diferença entre sazonalidade e ciclo é que a sazonalidade segue um padrão claro e repetitivo, enquanto os ciclos são mais irregulares.

O ruído, por sua vez, é composto por variações aleatórias e imprevisíveis que não são explicadas pelas outras três componentes. Este elemento reflete fatores externos, erros de medição ou flutuações naturais que não seguem um padrão estruturado. O ruído é essencialmente residual e pode ser tratado como um comportamento estocástico.

3.1.1.2 Autoregressão

A Autoregressão é amplamente utilizada em diversos modelos, eles são notavelmente flexíveis no tratamento de uma ampla variedade de padrões em séries temporais (HYNDMAN; ATHANASOPOULOS, 2011) e são am-

plamente utilizados na prática.

Modelos estatísticos aproveitam as características intrínsecas de uma série temporal para criar representações compactas. Isso é viável devido às premissas rígidas que esses modelos fazem sobre os dados, como a identificação precisa da ordem de um processo autorregressivo (AR). A ordem p de um processo AR(p) refere-se ao número de valores anteriores da série (defasagens) que influenciam diretamente o próximo valor. Processos AR com ordens elevadas são especialmente úteis para analisar dados de alta granularidade (como minutos, segundos ou milissegundos) e para capturar dependências de longo prazo, onde observações passadas ainda impactam os resultados futuros.

Os parâmetros de um modelo AR são tradicionalmente ajustados usando mínimos quadrados (Classic-AR). Ao modelar dependências de longo alcance, o procedimento de ajuste de modelos Classic-AR com uma alta ordem (p pode se tornar impraticavelmente lento (TRIEBE NIKOLAY LAPTEV, 2019).

Em uma série temporal $y_1, ..., y_t$, modelada como um processo autorregressivo, a previsão do próximo valor y_t é obtida ao combinar linearmente os p valores anteriores $(y_{t-1}, y_{t-2}, ..., y_{t-p})$ com pesos w_i que correspondem aos coeficientes do modelo AR aprendidos, como podemos ver na equação 3-1.

$$y_t = c + \sum_{i=1}^{p} w_i y_{t-i} + e_t \tag{3-1}$$

na equação 3-1 temos:

- y_t é p valor da série no tempo t
- w_i corresponde aos coeficientes autoregressivos que representam a influência dos valores passados
- c é uma constante (intercepto)
- e_t é um termo de erro aleatório

O modelo autoregressivo de médias móveis ARMA é um dos principais modelos de previsão de séries temporais. Esse modelo é derivado de um processo AR com um componente de média móvel adicionado (HOLAN

ROBERT LUND, 2010). Um processo ARMA(p, q)) é parametrizado pelas ordens p e q dos componentes AR(p) e MA(q) e seus respectivos pesos w_i e u_i, como visto em 3-2

$$y_t = c + \sum_{i=1}^{i=p} w_i * y_{t-i} + \sum_{i=1}^{i=q} u_i * y_{t-i} + e_t \tag{3-2}$$

3.1.1.3 Redes Neurais

Para superar os desafios de escalabilidade, modelos que utilizam Redes Neurais Recorrentes (RNN) e Redes Neurais Convolucionais (CNN) começaram a ser utilizados. No entanto, no formato atual, RNNs e CNNs foram projetadas para dados ricos de processamento de linguagem natural ou imagens, tornando-as complexas demais para a maioria das aplicações de séries temporais. Além disso, sua adoção é limitada pela dificuldade de tornar os modelos explicáveis para os tomadores de decisão.

Redes neurais são atraentes para a modelagem de séries temporais conforme destacado em (TANG, 1993), devido a dois fatores principais. Elas têm a capacidade de mapear funções não lineares e aproximar qualquer função contínua, permitindo a resolução de problemas complexos desde que haja dados suficientes. Além disso, por serem modelos não paramétricos, não dependem de suposições rígidas sobre o processo gerador dos dados, o que reduz o risco de erros por especificação incorreta. Essa flexibilidade é especialmente vantajosa, pois séries temporais frequentemente apresentam comportamentos únicos que não são capturados por modelos paramétricos (TRIEBE NIKOLAY LAPTEV, 2019).

De forma geral, uma rede neural pode ser entendida como uma rede de neurônios programados e organizados em camadas. As entradas compõem a camada inferior, enquanto as previsões formam a camada superior. Além disso, podem existir camadas intermediárias, conhecidas como camadas ocultas, que contêm neurônios ocultos. (HYNDMAN, 2017).

Uma rede neural mais simples não contém camadas ocultas, tornando-a equivalente a uma regressão linear. A Figura 3.1 ilustra uma rede neural com quatro preditores e seus pesos associados, que são utilizados para obter previsões por meio de uma combinação linear das entradas. Os pesos são ajustados dentro do framework da rede neural, utilizando o método de gradiente descendente para minimizar uma função de custo, como o erro quadrático médio (MSE). (HYNDMAN, 2017).

Quando uma camada intermediária com neurônios ocultos é utilizada, a rede neural torna-se não linear, como mostrado na Figura 3.2. Essa configuração é conhecida como uma rede neural feed-forward multicamadas, na qual cada nó recebe entradas das camadas anteriores. As entradas de cada nó são calculadas através de uma combinação linear ponderada. O resultado dessa combinação é, então, transformado por uma função não linear, como a função Sigmoid (Equação 3-3), antes de se tornar a entrada para a próxima camada.

A transformação em uma função não linear é crucial, pois permite que a rede aprenda representações mais complexas dos dados. Se todas as funções fossem lineares, a rede neural poderia apenas modelar relações lineares, limitando severamente sua capacidade de aprendizado. Com funções não lineares, a rede pode capturar interações complexas e padrões nos dados, aumentando a sua capacidade de generalização. Isso é especialmente vantajoso em tarefas como classificação e regressão em que os dados apresentam comportamentos não lineares, permitindo que a rede forneça previsões mais precisas e robustas.

$$s(x) = 1/(1 + e^{-x}) \tag{3-3}$$

Os pesos das redes neurais, que determinam a força das conexões entre os neurônios, são aprendidos a partir dos dados durante o treinamento. Para evitar que o modelo se ajuste demais aos dados de treinamento (overfitting), utiliza-se a regularização, que impõe restrições aos valores dos pesos.

No ínicio os pesos são inicializados de forma aleatória com valores da

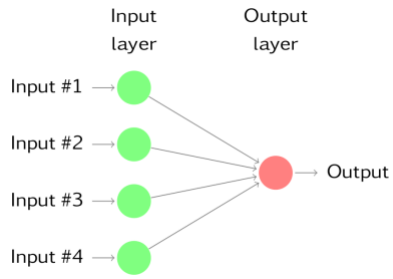

Figura 3.1: Rede neural simples equivalente a regressão linear

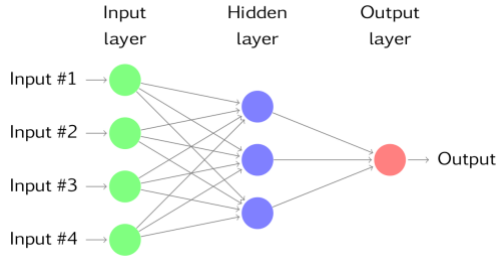

Figura 3.2: Rede neural com uma camada oculta

distribuição e são atualizados com o algoritmo SGD utilizando os dados treinados. Consequentemente existe um fator de aleatoridade nos pesos aprendidos pela rede neural, que é o resultado de cada modelo ajustado encontrando um subótimo.

As RNNs são especialmente adequadas para dados sequenciais, como séries temporais, pois possuem conexões recorrentes que permitem que a informação de passos anteriores influencie as predições. No entanto, RNNs tradicionais sofrem do problema do desaparecimento de gradientes, que dificulta o aprendizado de dependências de longo prazo.

3.1.1.4 AR-Net

O modelo AR-Net, utilizado no NeuralProphet, combina os fundamentos dos modelos estatísticos tradicionais de autoregressão com a flexibilidade de redes neurais. O algoritmo é projetado de forma que os parâmetros de sua primeira camada correspondam aos coeficientes autoregressivos, como é possível ver na figura 3.3. Em sua forma mais simples, o AR-Net é idêntico à regressão linear, ajustado com o método de gradiente estocástico (SGD).

Este modelo se destaca na capacidade de lidar com grandes ordens p (valores passados) e permite a modelagem de dependências de longo alcance.

A fórmula do AR Linear é equivalente à do AR Clássico, conforme apresentado na equação 3-1. A configuração padrão do AR-Net não contém camadas ocultas e é funcionalmente idêntica a um modelo AR clássico, tratando-se de uma rede neural de camada única com p entradas, h saídas, sem vieses e sem função de ativação. Os pesos dessa camada única ajustam cada defasagem específica a um passo de previsão específico, podendo ser associados a coeficientes correspondentes de uma coleção de h modelos AR(p) clássicos, o que torna o modelo simples de interpretar.

Além disso, o AR-Net pode ser configurado para utilizar camadas ocultas. Nesse caso, treina-se uma Rede Neural (NN) totalmente conectada com o número especificado de camadas ocultas e dimensões. Essas características tornam o AR-Net uma solução robusta, escalável e interpretável para problemas de séries temporais. Na figura 3.4, podemos ver uma representação do modelo com n camadas ocultas, de tamanho k, além da camada de saída conectada aos valores passados $y_{t-1}, \ldots, y_{t-p}$.

Para evitar a restrição de conhecer a verdadeira ordem AR, é utilizado os coeficientes AR de forma esparsa. Isso também eliminará a suposição de que os coeficientes AR devem consistir em defasagens consecutivas. Conseguimos isso adicionando um termo de regularização R à perda L que está sendo minimizada em 3-4:

$$\min_{\theta} L(y, \hat{y}, \theta) + \lambda(s) \cdot R(\theta) \tag{3-4}$$

onde

$$\lambda(s) = c_{\lambda} \cdot (s^{-1} - 1)$$

Com os seguintes parâmetros:

$$s = \frac{\hat{p}_{data}}{p_{model}}$$

onde s representa a esparsidade estimada dos coeficientes AR, que é definida pelo usuário. Além disso, temos a força de regularização dada por:

$$c_\lambda \approx \frac{\sqrt{\hat{L}}}{100}$$

Experimentamos com diferentes funções de regularização $R(\theta)$, incluindo a conhecida regularização L1 ("Lasso"). No entanto, nosso objetivo de regularização é diferente da maioria das aplicações. Não queremos desencorajar pesos grandes, como uma norma L1 ou L2 faria. Em vez disso, queremos incentivar o otimizador a definir pesos pequenos como zero, mantendo os outros pesos inalterados. Para nós, é importante que os pesos reais não sejam regularizados para serem menores que seus ótimos não regularizados, pois representam efetivamente os coeficientes AR.

A função de regularização tem como intuito incentivar o otimizador a definir pesos consegue isso ao ter um grande gradiente próximo de zero e, em seguida, decair rapidamente mais perto de um. Assim, os gradientes dos pesos regularizados mais distantes de zero basicamente desaparecem. Esse comportamento é obtido ao fazer uma combinação modificada de uma raiz e uma transformação sigmoidal dos valores absolutos dos pesos, como e visto em 3-5.

É importante que os pesos reais não sejam regularizados para serem menores que seus ótimos não regularizados, pois eles representam os coeficientes AR.

A função de regularização tem como intuito preservar a magnitude dos pesos significativos enquanto anula aqueles que não contribuem para a previsão. Ela consegue isso ao ter um grande gradiente próximo de zero e, em seguida, decair rapidamente mais perto de um. Assim, os gradientes dos pesos regularizados mais distantes de zero basicamente desaparecem. Esse

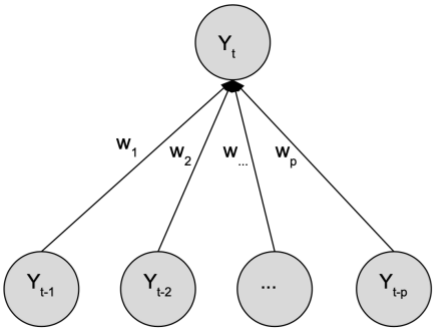

Figura 3.3: Arquitetura de rede neural equivalente ao AR

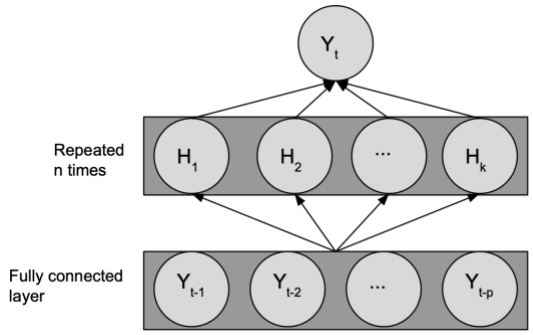

Figura 3.4: Arquitetura de AR-Net com camadas ocultas

comportamento é obtido ao fazer uma raiz de uma função sigmoide modificada dos valores absolutos dos pesos, como é visto em 3-5. Isso permite que o modelo mantenha a interpretabilidade e eficácia na seleção de coeficientes relevantes, promovendo uma representação mais precisa do comportamento dinâmico dos dados.

$$\Lambda_{AR\text{-}Net}(\theta, c_1, c_2) = \frac{1}{p}\sum_{i=1}^{p} 2 \cdot \left(1 + \exp\left(-c_1 \cdot |\theta_i|^{\frac{1}{c_2}}\right)^{-1}\right) - 1 \qquad (3\text{-}5)$$

Os parâmetros da curva de regularização c_1, c_2 dependem da faixa dos coeficientes AR. Para dados normalizados, com coeficientes AR na faixa $[0, 1]$, $c_1 \approx 3$ e $c_2 \approx 3$ funcionam idealmente.

3.1.1.5 Visão geral do NeuralProphet

O NeuralProphet é um modelo que utiliza como característica central a modularidade composicional. Ele é composto por módulos, cada um contribuindo com um componente aditivo para a previsão. A maioria dos componentes também pode ser configurada para ser escalada pela tendência, resultando em um efeito multiplicativo. Cada módulo possui suas entradas individuais e processos de modelagem. No entanto, todos os módulos devem produzir h saídas, onde h define o número de passos que serão previstos no futuro de uma só vez. Esses valores são somados como os valores previstos $\hat{y}_t, \ldots, \hat{y}_{t+h-1}$ para os futuros valores da série temporal $y_t, \ldots, y_{t+h-1}$. Se o modelo depender apenas do tempo, é possível produzir um número arbitrário de previsões. Nas descrições a seguir, esse caso especial será tratado matematicamente como equivalente a uma previsão de um passo à frente, com $h = 1$. Na equação 3-6 podemos ver como o modelo é tratado matematicamente para uma previsão de um passo à frente, com $h = 1$.

$$\hat{y}_t = T(t) + S(t) + E(t) + F(t) + A(t) + L(t) \tag{3-6}$$

onde:

- $T(t)$ = Tendência no tempo t
- $S(t)$ = Efeitos sazonais no tempo t
- $E(t)$ = Efeitos de eventos e feriados no tempo t
- $F(t)$ = Efeitos de regressão no tempo t para variáveis exógenas conhecidas no futuro
- $A(t)$ = Efeitos autorregressivos no tempo t baseados em observações passadas
- $L(t)$ = Efeitos de regressão no tempo t para observações defasadas de variáveis exógenas

Todos os módulos vistos em 3-6 podem ser configurados individualmente e combinados para compor o modelo. A seguir, será explicado o que cada componente faz, detalhando suas funcionalidades e como contribuem para o desempenho geral do modelo. Porém, apenas a tendência, a autoregressão e a sazonalidade serão utilizadas neste projeto.

Tendência A modelagem da tendência parte da fórmula 3-7, que combina um deslocamento m com uma taxa de crescimento k. No entanto, essa abordagem é estendida para permitir que a taxa de crescimento varie em diferentes pontos, possibilitando que a tendência seja representada como uma série linear por partes. Isso resulta em uma modelagem flexível, mas ainda interpretável. Em cada segmento, o efeito da tendência é determinado pela taxa de crescimento constante multiplicada pela variação no tempo. Essa abordagem é generalizada ao introduzir uma taxa de crescimento dependente do tempo, $\delta(t)$, e um deslocamento também dependente do tempo, $\rho(t)$, conforme descrito na equação 3-8.

$$T(t_1) = T(t_0) + k \cdot \Delta t = m + k \cdot (t_1 - t_0) \tag{3-7}$$

$$T(t) = \delta(t) \cdot t + \rho(t) \tag{3-8}$$

A tendência linear por partes ajusta a taxa de crescimento em um número finito de pontos de mudança, definidos como $C = (c_1, c_2, \ldots, c_{n_c})$. Entre os pontos de mudança, a taxa de crescimento é constante, com δ_0 e ρ_0 representando a taxa e o deslocamento iniciais, respectivamente.

Os ajustes na taxa de crescimento são dados pelo vetor $\delta \in \mathbb{R}^{n_c}$, onde δ_j representa a mudança na j-ésima taxa. O deslocamento em t é determinado por ρ_0 somado aos ajustes até t, com $\rho_j = -c_j \delta_j$, garantindo continuidade. A equação para a tendência é apresentada na equação 3-9:

$$T(t) = (\delta_0 + \Gamma(t)^T \delta) \cdot t + (\rho_0 + \Gamma(t)^T \rho) \tag{3-9}$$

Onde:

$$\delta = (\delta_1, \delta_2, \ldots, \delta_{n_c}),$$

$$\rho = (\rho_1, \rho_2, \ldots, \rho_{n_c}),$$

$$\Gamma(t) = (\Gamma_1(t), \Gamma_2(t), \ldots, \Gamma_{n_c}(t)),$$

Com:

$$\Gamma_j(t) = \begin{cases} 1, & \text{se } t \geq c_j, \\ 0, & \text{caso contrário.} \end{cases}$$

O NeuralProphet oferece um mecanismo semi-automático para a seleção dos n_c pontos de mudança, permitindo distribuição equidistante ou ajuste manual (TRIEBE, 2021). Para evitar overfitting nos pontos finais, o último segmento da tendência utiliza uma maior porção dos dados, sendo por padrão 15%.

Sazonalidade A sazonalidade é modelada utilizando termos de Fourier (HARVEY, 1993). Nessa técnica, vários termos de Fourier são definidos para cada sazonalidade, conforme a Equação 3-10, onde k é o número de termos de Fourier definidos para a sazonalidade com periodicidade p. Esses termos, formados por pares de seno e cosseno, permitem modelar múltiplas sazonalidades, incluindo aquelas com periodicidades não inteiras, como sazonalidade anual com dados diários ($p = 365.25$) ou semanais ($p = 52.18$). Em cenários com múltiplas sazonalidades, diferentes valores de n podem ser definidos para cada periodicidade.

$$S_p(t) = \sum_{j=1}^{k} \left[a_j \cdot \cos\left(\frac{2\pi jt}{p}\right) + b_j \cdot \sin\left(\frac{2\pi jt}{p}\right) \right] \tag{3-10}$$

Autoregressão O NeuralProphet emprega o mesmo mecanismo do AR-Net para realizar previsões sem camadas ocultas. No entanto, quando camadas ocultas são introduzidas, o modelo utiliza uma função de regularização diferente daquela proposta pelo AR-Net. Essa variação, que é demonstrada em 3-11, mostrou-se mais eficaz em uma ampla variedade de conjuntos de dados,

conforme demonstrado em (TRIEBE, 2021)

$$\Lambda(\theta, \epsilon, \alpha) = \frac{1}{n} \sum_{i=1}^{n} \log\left(\frac{1}{\epsilon \cdot e} + \alpha \cdot |\theta_i|\right) + \log(\epsilon) + 1 \tag{3-11}$$

A equação 3-11 apresenta uma função de regularização geral baseada em uma transformação logarítmica deslocada e escalada dos valores absolutos dos pesos. A parametrização específica para um módulo com pesos θ permite um ajuste fino da complexidade do modelo através dos parâmetros ϵ e α.

O parâmetro ϵ controla a taxa de crescimento da penalidade para pesos pequenos, influenciando a esparsificação do modelo. Já o parâmetro α determina a taxa de decaimento da penalidade para pesos grandes, afetando a capacidade do modelo de capturar padrões complexos.

A regularização é aplicada na força configurada por módulo e adicionada à função de perda, para ser retropropagada. A regularização só começa após uma porcentagem especificada do treinamento, por padrão, após 50%, e é então aumentada linearmente de zero para sua força total configurada no final do treinamento.

Regressores defasados Regressores defasados, também conhecidos como covariáveis, são utilizados para estabelecer correlações entre outras variáveis observadas e a série temporal alvo. Ao contrário dos regressores futuros, o futuro dos regressores defasados é desconhecido. No momento t da previsão, estão disponíveis apenas os seus valores observados e passados até t - 1, inclusive.

$$L(t) = \sum_{x \in X} L_x(x_{t-1}, x_{t-2}, \ldots, x_{t-p}) \tag{3-12}$$

No NeuralProphet, dado um conjunto de covariáveis $X \in \mathbb{R}^{T \times n_l}$ foi criado um módulo de regressão defasada separado para cada uma das m covariáveis x de comprimento T. Isso permite atribuir individualmente o efeito de cada covariável nas previsões. Cada módulo de regressão defasado é funcionalmente idêntico ao módulo AR, com a única diferença sendo as entradas. Conforme

apresentado na Equação 3-12 as p últimas observações da covariável x são as entradas do módulo, invés da própria série y, como no AR. As saídas têm formato idêntico, com cada módulo produzindo h componentes aditivos $L_t^x(t), L_t^x(t+1), \ldots, L_t^x(t+h-1)$, para as previsões gerais $\hat{y}_t, \hat{y}_{t+1}, \ldots, \hat{y}_{t+h-1}$. Como visto em 3-13

No NeuralProphet, dado um conjunto de covariáveis $X \in \mathbb{R}^{T \times n_l}$, um módulo de regressão defasada é criado separadamente para cada uma das m covariáveis x de comprimento T. Essa abordagem permite isolar o efeito de cada covariável nas previsões. Cada módulo de regressão defasada é funcionalmente idêntico ao módulo AR, diferenciando-se apenas pelas entradas. Conforme apresentado na Equação (3-12), as p últimas observações da covariável x (ou seja, x_{t-1}, x_{t-2}, ..., x_{t-p})) servem como entradas para o módulo, ao invés da própria série temporal y, como ocorre no módulo AR. As saídas desses módulos também possuem formato idêntico, produzindo h componentes aditivos $L_t^x(t), L_t^x(t+1), \ldots, L_t^x(t+h-1)$ que contribuem para as previsões globais $\hat{y}_t, \hat{y}_{t+1}, \ldots, \hat{y}_{t+h-1}$. A relação entre as entradas e as saídas do módulo de regressão defasada é expressa na equação 3-13

$$L_t^x(t), L_t^x(t+1), \ldots, L_t^x(t+h-1) = \text{AR-Net}(x_{t-1}, x_{t-2}, \ldots, x_{t-p}) \quad (3\text{-}13)$$

Regressores futuros Para modelar regressores futuros, tanto os valores passados quanto os futuros desses regressores precisam ser conhecidos. Dado o conjunto de regressores futuros $F \in \mathbb{R}^{T \times n_f}$, onde n_f é o número de regressores, o efeito de todos os regressores futuros no instante de tempo t pode ser representado como $F(t)$, conforme a Equação 3-14. Nessa equação, d_f representa o coeficiente do modelo para a regressor futuro $f \in V$. Por padrão, os regressores futuros têm um efeito aditivo, mas podem ser configurados para ter um efeito multiplicativo.

Para modelar regressores futuros, é necessário conhecer tanto os valores

passados quanto os futuros desses regressores. Dado o conjunto de regressores futuros $F \in \mathbb{R}^{T \times n_f}$, onde n_f representa o número de regressores futuros, o efeito total dos regressores futuros no instante t é denotado por $F(t)$, conforme a Equação (3-14). Nesta equação, d_f representa o coeficiente do modelo associado ao regressor futuro $f \in V$. Por padrão, assume-se que os regressores futuros tenham efeito aditivo sobre a previsão, mas essa configuração pode ser alterada para um efeito multiplicativo.

$$F(t) = \sum_{f \in F} F_f^*(t) \quad \text{(Equação 3-14)} \tag{3-14}$$

onde,

$$F_f(t) = d_f \cdot f(t) \tag{3-15}$$

$$F_f^*(t) = \begin{cases} T(t) \cdot F_f(t), & \text{se } f \text{ for multiplicativo} \\ F_f(t), & \text{caso contrário} \end{cases} \tag{3-16}$$

Eventos e Feriados Os efeitos de eventos especiais ou feriados podem ocorrer esporadicamente. Esses eventos são modelados de forma análoga às regressoras futuras, com cada evento e representado como uma variável binária $e \in [0, 1]$, indicando se o evento ocorre ou não em um dia específico. Para um conjunto de eventos $E \in \mathbb{R}^{T \times n_e}$, onde n_e é o número de eventos e o comprimento da série é T, o efeito de todos os eventos no instante de tempo t pode ser denotado por $E(t)$ na Equação 3-17, onde z_e representa o coeficiente do modelo correspondente ao evento $e \in E$.

$$E(t) = \sum_{e \in E} E_e^*(t) \tag{3-17}$$

onde,

$$E_e(t) = z_e \cdot e(t) \tag{3-18}$$

$$E_e^*(t) = \begin{cases} T(t) \cdot E_e(t), & \text{se } e \text{ for multiplicativo} \\ E_e(t), & \text{caso contrário} \end{cases} \tag{3-19}$$

3.1.2 LSTM

O LSTM é um tipo de rede neural recorrente que se destaca por sua capacidade de capturar dependências de longo prazo em dados sequenciais. Esse modelo é particularmente eficaz no processamento e análise de séries temporais, textos e fala. As LSTMs utilizam células de memória e um conjunto de portas (de entrada, de esquecimento e de saída) que controlam o fluxo de informações, permitindo que o modelo retenha ou descarte informações seletivamente. Isso ajuda a mitigar o problema do gradiente de desaparecimento, comum em RNNs tradicionais, possibilitando um aprendizado mais eficiente em séries temporais mais longas.

O LSTM foi escolhido devido a diversos fatores, tais como à sua ampla aplicação no campo do machine learning, ideais para capturar relações a longo prazo nos dados, eficazes para aprender padrões sequenciais complexos, como tendências e sazonalidades, e também devido à extensa comprovação da sua eficácia na predição de ações, como nos exemplos (ZHANG, 2023)(S; D; RAJAN, 2022a)(S; D; RAJAN, 2022b).

3.1.2.1 Conceitos Fundamentais

Todas as redes neurais recorrentes têm a forma de uma cadeia de módulos repetitivos de redes neurais. Em RNNs padrão, este módulo repetitivo terá uma estrutura muito simples, como uma única camada tanh, como visto na figura 3.5

LSTMs também apresentam uma estrutura semelhante a uma cadeia, mas o módulo repetitivo possui uma estrutura diferente. Em vez de uma única camada de rede neural, há quatro, interagindo de maneira diferente, como visto

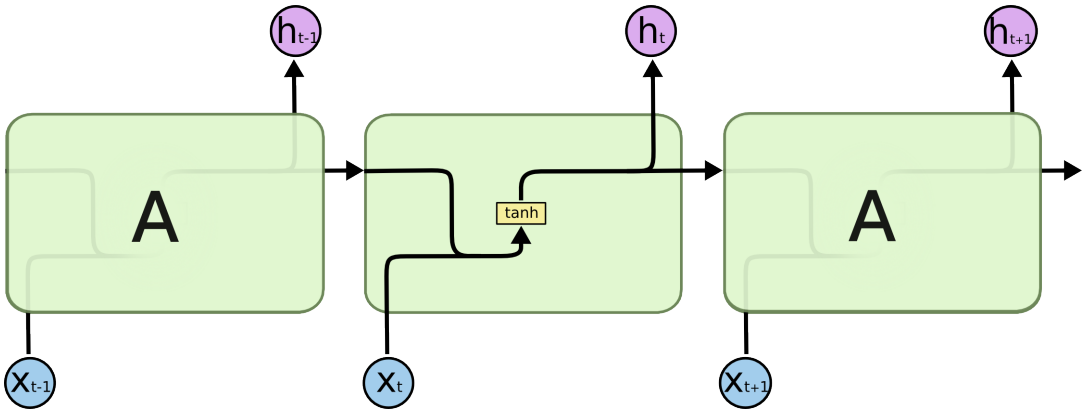

Figura 3.5: O módulo repetitivo em uma RNN padrão

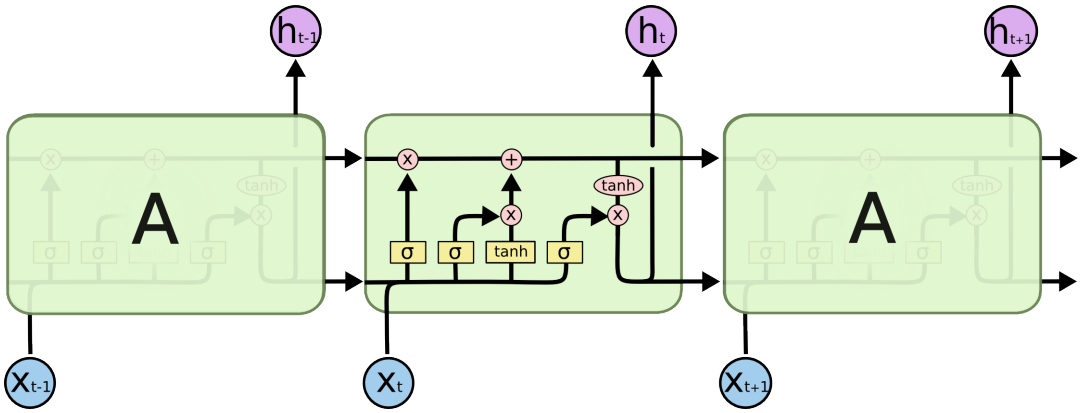

Figura 3.6: O módulo repetitivo em uma LSTM

na figura 3.6.

O principal componente das LSTMs é o estado da célula, representado pela linha horizontal que atravessa o topo do diagrama na Figura 3.9. A LSTM é capaz de adicionar ou remover informações do estado da célula, com esse processo sendo controlado por suas portas.

As portas funcionam como mecanismos que decidem, de forma seletiva, quais informações podem passar. Elas são compostas por uma camada de rede neural sigmoide e uma operação de multiplicação ponto a ponto. A camada sigmoide gera valores entre 0 e 1, indicando a proporção de cada componente que deve ser permitido passar. Um valor de 0 bloqueia completamente a passagem, enquanto um valor de 1 permite a passagem total. A LSTM utiliza três portas principais para proteger e regular o estado da célula.

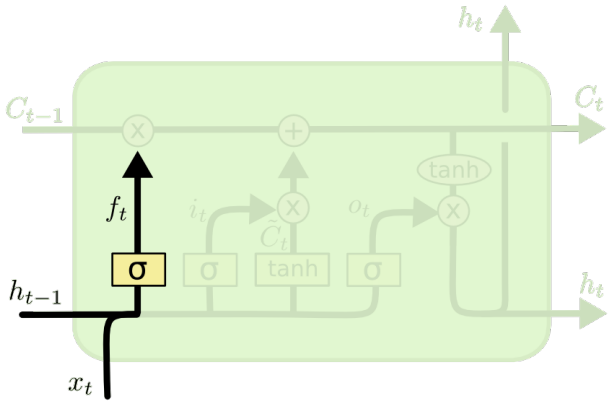

Figura 3.7: Processo responsável por descartar informações do passado.

3.1.2.2 Funcionamento

O primeiro passo da LSTM é determinar quais informações devem ser descartadas do estado da célula. Essa decisão é feita por uma camada sigmoide, chamada de "camada de porta de esquecimento", que analisa h_{t-1} e x_t e gera um valor entre 0 e 1 para cada elemento do estado da célula C_{t-1}. Um valor de 1 indica que a informação deve ser mantida, enquanto 0 sinaliza que deve ser descartada. Essa lógica pode ser observada na Figura 3.7 e na equação correspondente 3-20, onde:

- σ representa a função de ativação;
- $[h_{t-1}, x_t]$ é a entrada aumentada, combinando a entrada atual com a saída anterior;
- $W_f \cdot [h_{t-1}, x_t]$ indica um vetor de estados;
- b_f denota o bias.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3-20}$$

O próximo passo é decidir quais novas informações serão armazenadas no estado da célula. Primeiramente, uma camada sigmóide conhecida como "camada de porta de entrada"decide quais valores serão atualizados. Em

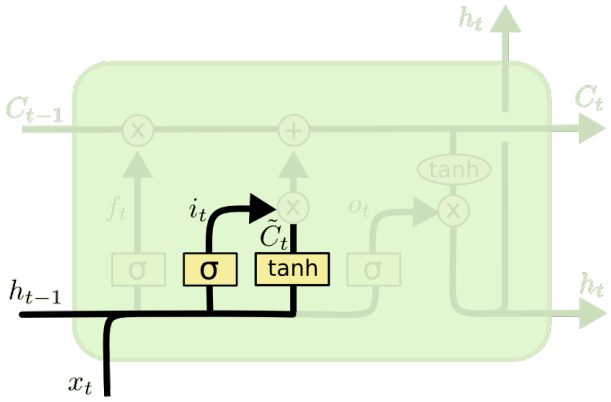

Figura 3.8: Processo responsável por inserir novas informações na célula.

seguida, uma camada tanh cria um vetor de novos valores candidatos $\tilde{C}_t$ que podem ser adicionados ao estado. Essa lógica pode ser observada na figura 3.8

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3-21}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3-22}$$

O próximo passo é atualizar o estado da célula anterior C_{t-1}, para o novo estado C_t. As etapas anteriores determinaram quais informações serão modificadas. O estado antigo é multiplicado por f_t, descartando as informações que foram selecionadas para serem esquecidas. Em seguida, são adicionados novos valores candidatos, representados por $i_t \cdot \tilde{C}_t$, que são escalados de acordo com a decisão de atualização de cada valor do estado. Esse passo pode ser visto na figura 3.9

A etapa final do processo consiste na geração da saída da rede neural. A saída é derivada do estado interno da célula, porém submetida a um processo de filtragem. Uma camada sigmóide é utilizada para determinar quais componentes do estado da célula contribuirão para a saída. Em seguida, o estado da célula é normalizado pela função tanh, assegurando que os valores estejam dentro do intervalo de -1 a 1. A combinação dos resultados da função tanh com a saída da camada sigmóide resulta na saída final, composta exclusivamente pelas informações selecionadas, como é possível observar na

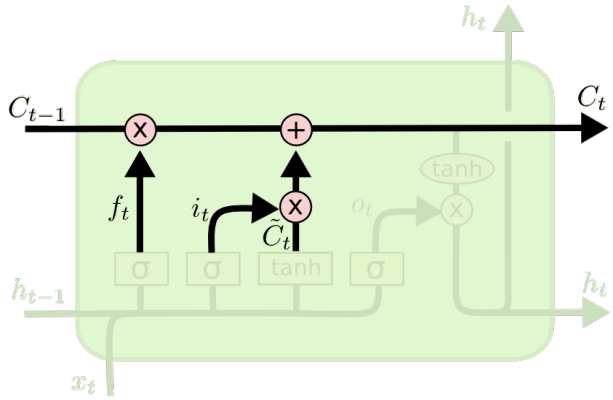

Figura 3.9: Processo responsável pela atualização do estado da célula.

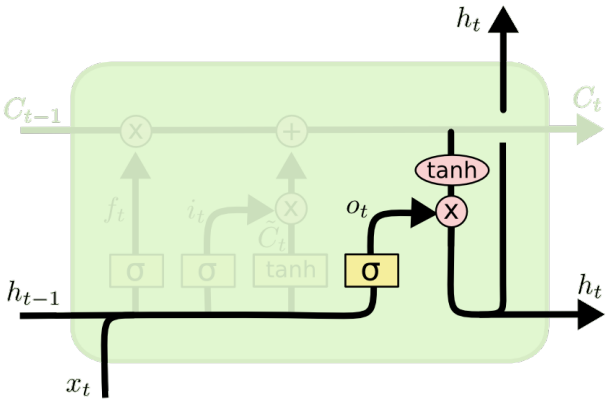

Figura 3.10: Processo responsável pela geração da saída da rede neural.

figura 3.10 e descrito matematicamente na equação 3-24.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3-23}$$

$$h_t = o_t * \tanh(C_t) \tag{3-24}$$

Ao longo do processo descrito, as LSTMs realizam diversas operações fundamentais para gerenciar informações ao longo do tempo, conforme detalhado abaixo:

- 3-20: Determina quais informações devem ser descartadas do estado da célula com base na relevância dos dados anteriores.
- 3-21: Avalia se as novas informações calculadas devem ser incorporadas ao estado da célula.

– 3-22: Atualiza o estado interno, decidindo se o contexto atual deve ser armazenado na memória da rede, acessando o circuito superior,
– 3-23: Seleciona as informações do estado da célula que serão enviadas como saída, garantindo sua adequação por meio de filtragem e normalização.

3.2 Padrões

Em contextos como análise de dados, inteligência artificial e engenharia de software, técnicas referem-se a métodos e abordagens específicos aplicados para alcançar objetivos concretos, seja na análise de informações, otimização de processos ou desenvolvimento de modelos preditivos. Essas técnicas formam a base das operações analíticas e são essenciais para transformar dados brutos em informações significativas que podem orientar decisões estratégicas.

3.2.1 Coleta de Dados

Para a coleta de dados, utilizou-se a biblioteca Python TvDataFeed, que permite o acesso direto às informações do site TradingView. Com ela, é possível obter dados históricos em intervalos configuráveis, como diário, semanal e mensal. No projeto, definiu-se um período de 10 anos com intervalo diário. O TradingView foi escolhido por oferecer uma ampla gama de dados precisos e atualizados para diferentes ativos financeiros. Além disso, destaca-se pela confiabilidade, flexibilidade e facilidade de uso, o que torna o processo de extração de dados simples e eficaz.

Foi decidido criar o portfólio inicial utilizando 52 ativos (Tabela3.1), representando as maiores empresas e fundos negociados no Brasil por setor. Todas as informações foram extraídas do provedor TradingView através de coleta diária de valores. Além disso, os ativos foram escolhidos com base em sua liquidez e representatividade no mercado, garantindo uma diversificação inicial adequada para análise.

Tabela 3.1: Principais ativos por setor

Ativos Utilizados					
ABEV3	AMER3	AZUL4	BBAS3	BBDC3	BEEF3
BLAU3	BPAC3	BRFS3	CPLE3	CRFB3	CSNA3
ELET3	EMBR3	ETHE11	FLRY3	GGBR3	GOLL4
HAPV3	HASH11	ITUB3	JBSS3	KEPL3	KLBN3
MELI34	MRFG3	MGLU3	MLAS3	NTCO3	OIBR3
ODPV3	PETR3	PETR4	PNVL3	POSI3	PRIO3
QBTC11	QETH11	RADL3	RANI3	RDOR3	RRRP3
SUZB3	TASA3	TAEE3	TIMS3	USIM3	VALE3
VIVT3	WEGE3	IBOV11			

3.2.2 Análise exploratória de dados

Na análise exploratória de dados e diagnóstico, foram utilizadas as informações do livro (ATWAN, 2022). Com base no estudo, foram definidas as seguintes técnicas para a análise:

- **Gráfico de Linha**: Utilizado para exibir os dados ao longo do tempo, permitindo visualizar tendências e padrões de forma contínua.
- **Estatísticas Básicas**: Ferramenta essencial para obter uma visão geral dos dados, permitindo a análise de medidas descritivas e identificação de padrões ou possíveis anomalias. As principais informações fornecidas são:
  - ▷ **count**: Número de observações não nulas.
  - ▷ **mean**: Média aritmética.
  - ▷ **std**: Desvio padrão.
  - ▷ **min**: Valor mínimo.
  - ▷ **25% (primeiro quartil)**: Valor abaixo do qual estão 25% dos dados.
  - ▷ **50% (mediana)**: Valor central dos dados.
  - ▷ **75% (terceiro quartil)**: Valor abaixo do qual estão 75% dos dados.

▷ **max**: Valor máximo.

– **Gráfico de Caixa (Boxplot)**: O boxplot é uma ótima ferramenta para se utilizar quando queremos ter um resumo visual rápido a respeito da distribuição dos dados, fornecendo uma visão detalhada da dispersão dos valores. O diagrama é composto pelos seguintes items:

  – Caixa é dividida por 3 items diferentes:

    * Quartil Inferior (Q3) repesentado na figura 3.11 com o mesmo nome, essa parte indica o ponto onde os 25% dos dados são iguais ou inferiores a esse valor
    * Mediana (Q2) é a linha de dentro da caixa, que representa o valor central dos dados, ou seja, onde 50% dos dados são iguais ou inferiores a esse valor
    * Quartil Superior (Q3) representa o ponto onde 75% dos dados são iguais ou inferiores a esse valor Além disso, podemos calcular o Intervalo interquartil (IQR) que representa a altura da caixa, esse valor pode ser obtido através do (Q3 - Q1)

  – os bigodes também conhecidos como whiskers, representados na figura 3.11 como máximo e mínimo retratam a extensão da caixa até o menor e maior valor dentro de:

$$\begin{aligned} \text{IQR} &= Q_3 - Q_1, \\ \text{Limite inferior}] &= Q_1 - 1.5 \times \text{IQR}, \\ \text{Limite superior} &= Q_3 + 1.5 \times \text{IQR} \end{aligned} \tag{3-25}$$

  – Outliers são valores que ficam fora do alcance dos bigodes. Eles são representados como pontos individuais no gráfico e indicam variações que podem ser anomalias, erros de medição ou eventos raros significativos na distribuição dos dados.

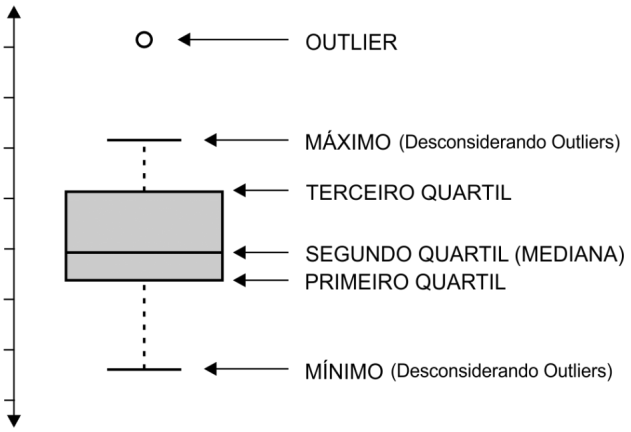

Figura 3.11: Diagrama boxplot.

Utilizado para visualizar a distribuição dos dados, destacando a mediana, quartis e possíveis outliers, fornecendo uma visão detalhada da dispersão dos valores.

- **Histograma**: Ferramenta para observar a distribuição de frequência dos dados, permitindo entender a frequência de ocorrência em intervalos específicos.
- **Gráfico de Tendência, Sazonalidade e Ruído**: Componentes da análise de séries temporais. A tendência identifica direções de longo prazo, a sazonalidade capta padrões periódicos, e o ruído representa variações aleatórias não explicadas.
- **Matriz de Correlação**: Utilizada para reduzir a quantidade de ativos iniciais, a matriz de correlação prioriza a seleção de ativos que apresentam séries temporais diferentes. O objetivo é escolher ativos de áreas diversificadas, que não se correlacionam entre si, promovendo assim uma maior diversificação das séries temporais.

3.2.3 Pre-Processamento

As técnicas e padrões foram definidos a partir de diversos artigos e estudos a respeito do tema.

O critério para selecionar as técnicas e padrões definidos prioriza métodos amplamente testados e reconhecidos, recomendados tanto pela comunidade científica quanto pela indústria, como MATLAB, Google e Scikit, entre outros (ZHENG; CASARI, Apr)(GARCíA; LUENGO; HERRERA, 2015). Embora as opções escolhidas possam não ser consideradas estado da arte, elas são reconhecidas como eficazes e impactantes na prática, pois são amplamente utilizadas por acadêmicos, pesquisadores e profissionais e se aplicam a dados de série temporal.

O pré-processamento de dados é amplamente discutido na literatura como um conjunto de operações que transforma dados brutos em dados de maior qualidade. Esse processo abrange etapas como normalização, detecção de outliers e redução de dimensionalidade (ALEXANDROPOULOS; KOTSIANTIS; VRAHATIS, 2019)(GARCíA; LUENGO; HERRERA, 2015). O pré-processamento de dados é essencial para obter entradas de alta qualidade e, consequentemente, saídas de qualidade, especialmente em modelos de machine learning.

3.2.3.1 Limpeza e Sincronização de Datas

Esta seção aborda dois aspectos essenciais. O primeiro é a limpeza da série temporal, que envolve a remoção de dados correspondentes a finais de semana e feriados, assegurando assim a integridade e a confiabilidade das informações. O segundo aspecto é a identificação das datas comuns a todos os ativos, o que possibilita comparações precisas e análises conjuntas. Essas etapas são fundamentais para a modelagem de séries temporais, a comparação de desempenho entre ativos e a geração de previsões mais precisas. Dados limpos e íntegros formam a base para decisões mais informadas no mercado financeiro.

3.2.3.2 Detecção de outliers

A técnica de detecção de outliers será realizada utilizando uma janela móvel em dois modelos distintos, e os outliers serão definidos como a interseção dos resultados obtidos por esses modelos, permitindo maior robustez à análise. (YU et al., 2014)(TAWAKULI et al., 2024)(ATWAN, 2022). Para isso, foram escolhidos o Interquartile Range e o z-score, ambos métodos clássicos para a detecção de anomalias. A proposta é identificar e remover os outliers técnicos, que são erros humanos, como erros de inserção, que não representam eventos reais da série temporal. Ao diferenciá-los dos outliers funcionais, que refletem ocorrências genuínas, é possível aprimorar a qualidade dos dados. Isso facilita uma interpretação mais precisa dos padrões e tendências da série, contribuindo para a construção de modelos preditivos mais robustos e confiáveis. Portanto, a estratégia para detectar outliers combinará uma média móvel de 30 dias com os métodos de detecção Z-score e IQR. A fórmula do IQR é representada na equação 3-25

A fórmula do z-score é dada por

$$\mathrm{Z} = \frac{(X - \mu)}{\sigma} \tag{3-26}$$

onde X é o valor, μ é a média da amostra e σ é o desvio padrão da amostra. z é o valor crítico comumente 2 ou 3 que define a distância em desvios padrão do valor médio para classificar um ponto como outlier sendo, neste projeto, definido como 3.

3.2.3.3 Tratamento de outliers

O foco principal são os outliers técnicos, que são diferenciados dos outliers funcionais, os quais representam eventos reais. Considerando a natureza dos outliers técnicos, caracterizados por serem valores discrepantes e não representativos da série, optou-se por uma abordagem de suavização local (GARCíA;

LUENGO; HERRERA, 2015). A média móvel, com uma janela de 7 períodos, mostrou-se adequada para essa finalidade, pois permite substituir o valor do outlier por uma estimativa mais robusta, baseada nos valores vizinhos. Essa técnica é particularmente útil para outliers isolados, garantindo que a correção não afete significativamente a tendência geral dos dados.

3.2.4 Engenharia de Atributos

A engenharia de atributos é um processo crucial para criar, selecionar, e transformar variáveis ou características a partir de dados brutos para melhorar o desempenho de modelos de machine learning, sendo utilizadas como variáveis exógenas nos modelos de predição. Como discutido por Li e Li (LI; LI, 2018), a seleção e avaliação de atributos são etapas cruciais nesse processo, e por tanto, elas serão utilizadas nesta análise.

3.2.4.1 Análise Técnica

Para a análise das técnicas, foram selecionados 59 indicadores técnicos, escolhidos entre os principais disponíveis que podem ser calculados exclusivamente com base nas informações de abertura, máxima, mínima e fechamento (OHLC). Essa restrição foi adotada devido à limitação das informações obtidas durante a coleta de dados, garantindo que todas as análises estejam alinhadas a esse conjunto específico.

Essas ferramentas desempenham um papel essencial na identificação de padrões e tendências nos dados e serão utilizadas como variáveis exógenas nos modelos de machine learning (PHUOC et al., 2024), fornecendo insights valiosos para melhorar a precisão das previsões. A seleção dos indicadores tem como objetivo otimizar o desempenho dos modelos ao proporcionar uma compreensão mais detalhada dos comportamentos do mercado.

Todos os indicadores selecionados estão listados na Tabela 3.2, organizados em quatro grupos principais. A seguir, serão apresentados e detalhados

alguns dos indicadores escolhidos.

- **Volatilidade**: Indicadores de volatilidade medem a intensidade das variações de preços em um determinado período. Eles ajudam a identificar os momentos de mercado calmo ou agitado, permitindo prever possíveis movimentos de ruptura. Exemplos

  - **Bollinger Bands (BBands)**: Utilizadas para medir a volatilidade de um ativo e ajudar a identificar condições de sobrecompra ou sobrevenda. As bandas consistem em três linhas (TRADINGVIEW, 2024a)

    * **Média Móvel Simples (SMA)**: Representa uma linha central, calculada como a média móvel de n períodos dos preços. A fórmula é:

$$\text{SMA} = \frac{1}{n}\sum_{i=1}^{n}\text{Preço}_i \tag{3-27}$$

    * **Desvio Padrão (DP)**: Mede a dispersão dos preços em relação à SMA ao longo dos n períodos, sendo calculado como:

$$\text{DP} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{Preço}_i - \text{SMA})^2} \tag{3-28}$$

    * **Banda Superior**: Representa o limite superior das Bandas de Bollinger, calculado como:

$$\text{Banda Superior} = \text{SMA} + k \times \text{DP} \tag{3-29}$$

    * **Banda Inferior**: Representa o limite inferior das Bandas de Bollinger, calculado como:

$$\text{Banda Inferior} = \text{SMA} - k \times \text{DP} \tag{3-30}$$

    Essas bandas refletem o quão dispersos os preços estão ao redor da média, com o desvio padrão ajustando a largura das bandas para refletir a volatilidade.

– **Avarage True Range (ATR)**: O ATR é um indicador utilizado para analisar a variação dos preços ao longo de diferentes intervalos de tempo em gráficos. Ele mede a amplitude das flutuações no preço dos ativos e calcula a média dessas amplitudes, proporcionando uma visão clara da volatilidade do mercado (TRADINGVIEW, 2024b).

$$TR_t = \max\left(\text{High}_t - \text{Low}_t, |\text{High}_t - \text{Close}_{t-1}|, |\text{Low}_t - \text{Close}_{t-1}|\right) \tag{3-31}$$

para a fórmula do True Range 3-31 temos

* **High**$_t$: O preço máximo do ativo no período t.
* **Low**$_t$: O preço mínimo do ativo no período t.
* **Close**$_{t-1}$: O preço de fechamento do ativo no período anterior $t-1$.

O True Range (TR) pode ser interpretado da seguinte maneira:

* A diferença entre o preço máximo (High_t) e o preço mínimo (Low_t) no período t.
* A diferença entre o preço máximo do período t e o preço de fechamento do período anterior (Close_{t-1}).
* A diferença entre o preço mínimo do período t e o preço de fechamento do período anterior (Close_{t-1}).

O maior valor entre essas três variações é considerado o True Range (TR) para o período t. Essa abordagem captura a variação total do preço, incluindo gaps ou movimentos significativos entre o fechamento de um período e a abertura do próximo.

Uma vez que o True Range (TR) é calculado, a fórmula do Average True Range (ATR) 3-32 é usada para calcular a média dos valores de TR ao longo de um número específico de períodos n:

$$ATR_t = \frac{\sum_{i=1}^{n} TR_i}{n} \tag{3-32}$$

    - $\mathbf{ATR}_t$: O Average True Range para o período t.
    - $\sum_{i=1}^{n} TR_i$: A soma dos valores de True Range (TR) ao longo dos últimos n períodos.
    - n: O número de períodos usados para calcular a média.

- **Tendência**: Indicadores de tendência são usados para identificar a direção predominante dos preços ao longo do tempo. Eles ajudam a reconhecer se o mercado está em uma tendência de alta, baixa ou lateral.

  - **Average Directional Movement Index (ADX)**: É um indicador técnico utilizado para medir a força de uma tendência em um mercado, independentemente de sua direção. Ele é calculado com base no movimento direcional +DI (Directional Indicator positivo) e -DI (Directional Indicator negativo)(TRADINGTECHNOLOGIES, 2024).

$$+DM = \begin{cases} \text{High} - \text{High}_{\text{prev}}, & \text{se } (\text{High} - \text{High}_{\text{prev}}) > (\text{Low}_{\text{prev}} - \text{Low}) \\ & \text{e } (\text{High} - \text{High}_{\text{prev}}) > 0, \\ 0, & \text{caso contrário.} \end{cases} \tag{3-33}$$

$$-DM = \begin{cases} \text{Low}_{\text{prev}} - \text{Low}, & \text{se } (\text{Low}_{\text{prev}} - \text{Low}) > (\text{High} - \text{High}_{\text{prev}}) \\ & \text{e } (\text{Low}_{\text{prev}} - \text{Low}) > 0, \\ 0, & \text{caso contrário.} \end{cases} \tag{3-34}$$

$$+DI = 100 \times \frac{+DM}{\text{ATR}} \tag{3-35}$$

$$-DI = 100 \times \frac{-DM}{\text{ATR}} \tag{3-36}$$

$$DX = 100 \times \frac{|+DI - -DI|}{+DI + -DI} \tag{3-37}$$

$$ADX = \frac{\mathrm{DX}_1 + \mathrm{DX}_2 + \cdots + \mathrm{DX}_n}{n} \tag{3-38}$$

* 3-33 e 3-34 representam, respectivamente, o movimento positivo e negativo do preço.
* 3-35 e 3-36 correspondem à média suavizada desses movimentos em relação ao intervalo (True Range), definido em 3-31.
* 3-37 calcula a diferença percentual entre +DI e -DI, indicando que, quanto maior o valor de DX, mais forte é a tendência, seja de alta ou de baixa.
* 3-38 define o ADX como a média móvel do DX, usada para medir a força geral da tendência. O ADX não considera a direção da tendência, mas apenas sua intensidade: valores altos indicam uma tendência forte, enquanto valores baixos sugerem um mercado lateral.

– **Aroon**: O indicador mede a força e a direção de uma tendência com base no tempo decorrido desde os pontos mais altos e mais baixos dentro de um período específico. Ele é composto por dois componentes Aroon Up e Aroon Down(TRADINGVIEW, 2024).

$$A_{\text{up}} = \frac{N - \text{Número de períodos desde o maior preço}}{N} \times 100 \tag{3-39}$$

$$A_{\text{down}} = \frac{N - \text{Número de períodos desde o menor preço}}{N} \times 100 \tag{3-40}$$

Nas equações 3-39 e 3-40, N representa o número total de períodos na análise, como, por exemplo, 14 dias em um cálculo típico.

A fórmula do Aroon Up 3-39 calcula o tempo passado desde o maior preço do período analisado, enquanto a fórmula do Aroon Down 3-40 avalia o tempo desde o menor preço do mesmo período.

Valores próximos de 100 em ambos os casos indicam que o preço está próximo do extremo correspondente, sendo o máximo para o Aroon Up e o mínimo para o Aroon Down.

- **Superposição**: Indicadores de overlap são utilizados para analisar gráficos de preços, onde um indicador é sobreposto diretamente sobre o gráfico principal, frequentemente para suavizar o ruído do mercado.

  - **Exponential Moving Average (EMA)**: A EMA é uma média móvel que atribui maior peso aos valores mais recentes, tornando-a mais sensível a mudanças recentes nos dados. Essa característica permite que a EMA reaja mais rapidamente às variações no preço do ativo em comparação à Simple Moving Average (SMA), que distribui os pesos de forma uniforme(BIO, 2024).

$$\alpha = \frac{2}{N+1} \tag{3-41}$$

Para a equação 3-41 temos:

* α: Conhecido como fator de suavização determina o peso atribuído aos valores mais recentes na fórmula da EMA
* N: Número de períodos

. O fator (α) depende do número total de períodos (N) considerados. Quanto menor o valor de N, maior será α, tornando a EMA mais responsiva às mudanças nos dados.

$$\text{EMA}_t = \alpha \cdot P_t + (1 - \alpha) \cdot \text{EMA}_{t-1} \tag{3-42}$$

Para a equação 3-42 apresenta a fórmula recursiva da Exponential Moving Average, que calcula o valor atual da média (EMA_t) como uma combinação ponderada entre:

* O valor atual (P_t), ajustado pelo fator de suavização (α);
* O valor da EMA do período anterior (EMA_{t-1}), ajustado pelo peso complementar $(1 - \alpha)$.

- **Supertrend (supertrend)**: O Supertrend é um indicador de tendência baseado no Average True Range (ATR), utilizado para identificar tendências e potenciais pontos de entrada e saída no gráfico de preços. Seu cálculo combina detecção de tendência e volatilidade, funcionando com base em duas variáveis principais: o fator multiplicador e o período ATR, que mede a volatilidade do ativo.

  Traçado como uma linha no gráfico, o Supertrend alterna entre estar acima ou abaixo do preço, dependendo da direção da tendência. Ele utiliza o preço médio e o ATR para ajustar dinamicamente os níveis de suporte ou resistência, sendo também útil para detectar mudanças na direção da tendência e posicionar stops.

$$\text{UpperBand}_t = \text{MedianPrice}_t + (\text{Multiplier} \times \text{ATR}_t) \tag{3-43}$$

$$\text{LowerBand}_t = \text{MedianPrice}_t - (\text{Multiplier} \times \text{ATR}_t) \tag{3-44}$$

$$\text{Supertrend}_t = \begin{cases} \text{LowerBand}_t, & \text{se o preço estiver em tendência de alta} \\ \text{UpperBand}_t, & \text{se o preço estiver em tendência de baixa} \end{cases} \tag{3-45}$$

- **Momento**: Indicadores de momento medem a taxa de mudança nos preços. Eles ajudam a identificar a força de uma tendência e possíveis pontos de reversão, mostrando se um ativo está sendo comprado ou vendido excessivamente.

  - **Stochastic Oscillator (STOCH)**: é um oscilador de momento limitado em um intervalo. Ele mede a posição do preço de fechamento em relação ao intervalo entre o preço mais alto e o mais baixo de um número definido de períodos. Normalmente, o STOCH é usado para três finalidades: identificar níveis de sobrecompra

e sobrevenda, detectar divergências e identificar sinais de alta e baixa(TRADINGVIEW, 2024a).

O Stochastic Oscillator pode ser dividido em duas linhas: %K e %D. A linha %K é a porcentagem do preço de fechamento atual (K) em relação ao intervalo de preço dentro do número de barras usadas no período de análise:

$$\%K = \text{SMA}\left(100 \cdot \frac{\text{Close}_{\text{atual}} - \text{Lowest Low}}{\text{Highest High} - \text{Lowest Low}}, \text{smoothK}\right) \tag{3-46}$$

A linha %D é uma média suavizada da %K para reduzir ruídos enquanto mantém a tendência geral:

$$\%D = \text{SMA}(\%K, \text{periodD}) \tag{3-47}$$

Onde:

- Lowest Low: O menor preço observado dentro do número de barras recentes no período de análise (parâmetro periodK).
- Highest High: O maior preço observado dentro do mesmo intervalo de análise (periodK).
- smoothK: Número de períodos usados para suavizar a linha %K.
- periodD: Número de períodos usados para calcular a média da linha %K e gerar a linha %D.

– **Moving Average Convergence Divergence (MACD)**: O MACD é um indicador popular usado na análise técnica para identificar aspectos da tendência de um ativo, como momento, direção e duração. Ele combina dois tipos de indicadores: duas médias móveis de períodos diferentes, que identificam a direção e a duração da tendência, e a diferença entre elas (linha MACD) e uma média exponencial dessa diferença (linha de sinal). O resultado dessa di-

ferença é mostrado em um histograma, que indica o momento do ativo (TRADINGVIEW, 2024b).

$$\text{MACD} = \text{EMAcurta} - \text{EMAlonga} \tag{3-48}$$

Para a equação 3-48, temos:

* **EMA curta**: Média Móvel Exponencial de um período curto (geralmente 12 períodos).
* **EMA longa**: Média Móvel Exponencial de um período longo (geralmente 26 períodos).

$$\text{Sinal} = \text{EMA}_{\text{MACD}} \tag{3-49}$$

Para a equação 3-49, temos:

* **EMA MACD**: EMA da linha MACD (normalmente 9 períodos).

$$\text{Histograma} = \text{MACD} - \text{Sinal} \tag{3-50}$$

O histograma do MACD é a diferença entre a linha MACD e a linha de sinal, representando visualmente o momento de alta ou baixa.

Componentes principais:

* **Linha MACD**: Representada pela equação 3-48, é a diferença entre uma EMA de longo prazo (geralmente 26 dias) e uma EMA de curto prazo (geralmente 12 dias).
* **Linha de sinal**: É uma EMA da linha MACD, normalmente de 9 períodos.
* **Histograma MACD**: Representa a diferença entre a linha MACD e a linha de sinal, oscilando acima e abaixo de uma linha central (linha zero), o que indica a força do movimento.

Os indicadores são fundamentais para uma análise técnica robusta, sendo cruciais para detectar padrões e tendências nos dados, e os melhores serão

utilizados como variáveis exógenas nos modelos de machine learaning (PHUOC et al., 2024), fornecendo informações valiosas que visam melhorar a precisão das previsões. A partir disso, serão utilizados 61 indicadores através de 4 grupos: Volatilidade, tendência, Sobreposição e Momento

3.2.5 Seleção de atributos

A abordagem propõe o uso do Extra Tree Regressor para a seleção de atributos. O ET é um modelo que visa identificar os atributos mais informativos a partir de um conjunto de variáveis potenciais, permitindo assim a identificação das características mais relevantes (TALUKDER et al., 2023)(TALUKDER, 2023). O ET é capaz de calcular uma pontuação para cada variável e, com base nesses escores, realiza-se um filtro para selecionar apenas os indicadores mais significativos para cada ação. As variáveis selecionadas são então empregadas nos modelos de aprendizado de máquina, garantindo que apenas informações úteis sejam utilizadas.

O algoritmo de Extra Trees, assim como o algoritmo de Random Forests, cria várias árvores de decisão. No entanto, a amostragem para cada árvore é aleatória, sem reposição. Isso significa que cada árvore recebe um conjunto único de amostras para o treinamento. Além disso, um número específico de atributos é selecionado aleatoriamente, a partir do conjunto total de features, para cada árvore.

A característica mais importante e única do Extra Trees é a seleção aleatória do valor de divisão para cada feature. Em vez de calcular um valor de divisão ideal localmente, o algoritmo escolhe aleatoriamente um valor para dividir os dados.

Essa abordagem resulta em árvores altamente diversificadas e menos correlacionadas entre si, o que aumenta a robustez e generalização do modelo.

A abordagem proposta utiliza o modelo Extra Tree Regressor para a seleção de atributos. O Extra Trees (ET) é uma técnica que identifica os atri-

butos mais informativos de um conjunto de variáveis potenciais, permitindo a seleção das características mais relevantes (TALUKDER et al., 2023)(TALUKDER, 2023). Esse modelo calcula uma pontuação para cada variável e, com base nesses escores, será aplicado um filtro que seleciona apenas os indicadores mais significativos. Os atributos selecionados são então utilizados nos modelos de aprendizado de máquina, garantindo que somente informações úteis sejam consideradas.

Assim como o Random Forest, o Extra Trees constrói várias árvores de decisão. No entanto, diferencia-se pela amostragem aleatória sem reposição para criar cada árvore, garantindo que cada uma receba um conjunto único de amostras. Além disso, um subconjunto de atributos é escolhido aleatoriamente para cada árvore.

A principal distinção do ET é a seleção aleatória do valor de divisão para cada atributo, ao invés de calcular localmente o valor ideal. Isso resulta em árvores altamente diversificadas e menos correlacionadas, o que aumenta a robustez e capacidade de generalização do modelo.

Dado um conjunto de vetores de treinamento $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \ldots, l$, e um vetor de rótulos $\mathbf{y} \in \mathbb{R}^l$, uma árvore de decisão particiona recursivamente o espaço de características de modo que as amostras com os mesmos rótulos ou valores-alvo semelhantes sejam agrupadas.

Sejam os dados no nó m representados por Q_m, contendo n_m amostras. Para cada divisão candidata $\theta = (j, t_m)$, composta por uma característica j e um limiar t_m, os dados são particionados nos subconjuntos $Q_m^{\text{left}}(\theta)$ e $Q_m^{\text{right}}(\theta)$, conforme a seguinte definição:

$$\begin{aligned} Q_m^{\text{left}}(\theta) &= \{(x, y) \mid x_j \leq t_m\}, \\ Q_m^{\text{right}}(\theta) &= Q_m \setminus Q_m^{\text{left}}(\theta). \end{aligned} \tag{3-51}$$

A qualidade de uma divisão candidata no nó m é então calculada

utilizando uma função de impureza ou função de perda $H(\cdot)$, cuja escolha depende da tarefa a ser resolvida (classificação ou regressão):

$$G(Q_m, \theta) = \frac{n_m^{\text{left}}}{n_m} H(Q_m^{\text{left}}(\theta)) + \frac{n_m^{\text{right}}}{n_m} H(Q_m^{\text{right}}(\theta)).$$

Os parâmetros que minimizam a impureza são então selecionados:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, G(Q_m, \theta).$$

O processo é recursivo para os subconjuntos $Q_m^{\text{left}}(\theta^*)$ e $Q_m^{\text{right}}(\theta^*)$, até que a profundidade máxima permitida seja atingida, $n_m < \min_{\text{samples}}$, ou $n_m = 1$.

Neste projeto, foi utilizado o modelo Extra Tree como método de classificação, implementado da seguinte forma:

Se o alvo for um valor contínuo, então para o nó m, os critérios comuns para minimizar e para determinar os locais das futuras divisões são o Erro Quadrático Médio (MSE ou erro L2), a Deviância de Poisson, bem como o Erro Absoluto Médio (MAE ou erro L1). O MSE e a Deviância de Poisson definem o valor predito dos nós terminais como o valor médio aprendido $\bar{y}_m$ do nó, enquanto o MAE define o valor predito dos nós terminais como a mediana $\text{median}(y)_m$.

Erro Quadrático Médio:

$$\begin{aligned} \bar{y}_m &= \frac{1}{n_m} \sum_{y \in Q_m} y \\ H(Q_m) &= \frac{1}{n_m} \sum_{y \in Q_m} (y - \bar{y}_m)^2 \end{aligned} \tag{3-52}$$

Deviância de Poisson:

$$H(Q_m) = \frac{2}{n_m} \sum_{y \in Q_m} \left(y \log \frac{y}{\bar{y}_m} - y + \bar{y}_m \right) \tag{3-53}$$

Erro Absoluto Médio:

$$\begin{aligned} \text{median}(y)_m &= \underset{y \in Q_m}{\text{median}}(y) \\ H(Q_m) &= \frac{1}{n_m} \sum_{y \in Q_m} |y - \text{median}(y)_m| \end{aligned} \tag{3-54}$$

$$\begin{aligned} median(y)_m &= \underset{y \in Q_m}{\text{median}}(y) \\ H(Q_m) &= \frac{1}{n_m} \sum_{y \in Q_m} |y - median(y)_m| \end{aligned} \tag{3-55}$$

3.2.5.1 Normalização e escalonamento dos dados

Para a normalização e o escalonamento dos dados, optou-se por utilizar a função da tangente hiperbólica. Essa escolha se baseia em estudos que demonstraram que a tangente hiperbólica apresenta resultados interessantes e eficazes(NAYAK; MISRA; BEHERA, 2012) em diversas aplicações de normalização e escalonamento de dados(NAYAK; MISRA; BEHERA, 2012)(BHANJA, 2018). A escolha da técnica de normalização é crucial, pois métodos inadequados podem prejudicar a acurácia da previsão. No artigo referenciado (NAYAK; MISRA; BEHERA, 2012), foram analisados vários métodos de normalização, como Escalonamento Decimal, Mediana, Z-Score, Min-Max, Sigmoid, Tanh e MAD, em modelos de previsão, e o modelo escolhido demonstrou um impacto positivo na normalização dos dados na previsão de ações.

Podemos observar a fórmula da tangente hiperbólica 3-56, onde X_i representa o i-ésimo valor da variável X. O X_i^S é o resultado da normalização do dado X_i pela fórmula fornecida.

$$X_i^S = 0.5 \left(\tanh \left(\frac{0.01 \cdot (X_i - \mu)}{\sigma} \right) + 1 \right) \tag{3-56}$$

Na qual μ é o valor médio e σ é o desvio padrão do conjunto de dados, respectivamente.

3.2.6
Análise dos Resultados

Para avaliar a precisão dos modelos Neural Prophet e LSTM na previsão dos preços de ações, foram utilizadas três métricas: sMAPE, MAPE e RMSE. Essas métricas fornecem uma avaliação abrangente do desempenho dos modelos, considerando diferentes aspectos dos erros de previsão.

Nas equações abaixo A_t é o valor observado, F_t é o valor previsto, e n representa o número total de previsões.

O sMAPE é uma métrica simétrica que mede a diferença percentual absoluta média entre os valores reais e previstos, normalizada pela média dos valores absolutos de cada par. O sMAPE é menos sensível a valores extremos do que o MAPE, tornando-o uma opção mais robusta em muitos casos. Um valor de sMAPE próximo de zero indica uma alta precisão, enquanto um valor próximo de 100% indica uma baixa precisão.

$$sMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|A_t - F_t|}{(|A_t| + |F_t|)/2} \tag{3-57}$$

O MAPE calcula a média dos desvios percentuais absolutos entre os valores previstos e os valores reais. Ele é uma medida popular devido à sua simplicidade e interpretabilidade. Um valor baixo de MAPE indica que, em média, as previsões estão próximas dos valores reais. No entanto, o MAPE apresenta algumas limitações. Ele é sensível a valores reais próximos de zero, o que pode levar a erros muito altos.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{3-58}$$

O RMSE é uma métrica que calcula a raiz quadrada da média dos quadrados das diferenças entre os valores previstos e os reais. Ao elevar os erros ao quadrado, o RMSE atribui um peso maior aos grandes erros, tornando-o mais sensível a outliers. Isso significa que o RMSE penaliza mais as previsões que estão muito distantes dos valores reais. Expresso na mesma unidade que

a variável dependente, o RMSE facilita a interpretação prática dos resultados. Por exemplo, se estamos prevendo o preço de ações, um RMSE de 2 reais indica que, em média, nossas previsões estão erradas em 2 reais.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(A_t - \hat{F}_t)^2} \tag{3-59}$$

A escolha dessas três métricas permite uma análise mais completa do desempenho dos modelos. Ao considerar a combinação das métricas sMAPE, MAPE e RMSE, é possível obter uma visão mais precisa da qualidade das previsões e identificar os pontos fortes e fracos de cada modelo.

Tabela 3.2: Indicadores selecionados

Indicadores Selecionados					
Aberration	Acceleration Bands	Average Directional Index	Directional Movement	Arnaud Legoux Moving Average	Archer Moving Averages Trends
Awesome Oscillator	Absolute Price Oscillator	Aroon	Average True Range	Bollinger Bands	Bias
Balance of Power	Commodity Channel Index	Candlestick Patterns	Chande Forcast Oscillator	Center of Gravity	Choppiness Index
Chande Kroll Stop	Chande Momentum Oscillator	Coppock Curve	Correlation Trend Indicator	Double Exponential Moving Average	Donchian Channel
Detrended Price Oscillator	Exponential Moving Average	Efficiency Ratio	Fisher Transform	Gann High-Low Activator	Heiken Ashi
Ichimoku Cloud	Inertia Indicator	Jurik Moving Average	Kaufman's Adaptive Moving Average	Keltner Channel	KDJ Indicator
Know Sure Thing	Moving Average Convergence Divergence	Pretty Good Oscillator	Percentage Price Oscillator	Parabolic SAR	Quantitative Qualitative Estimation
Rate of Change	Relative Strength Index	Relative Strength Xtra	Relative Vigor Index	Stochastic Oscillator	Super Trend
Triple Exponential Moving Average	TRIX Indicator	TTM Trend	True Strength Index	Ultimate Oscillator	Variable Index Dynamic Average
Vortex Indicator	Weighted Moving Average	Williams %R	Zero Lag Moving Average	Z-Score	HLC3
OHLC4	Linear Decay	Weighted Closing Price	Midprice	Super Smoother Filter	Midpoint
Fibonacci's Weighted Moving Average	HL2	Hull Moving Average	Simple Moving Average		

4
Implementação Técnica

Neste capítulo, será apresentada a implementação técnica do projeto, descrevendo o fluxo que guia a transformação dos dados brutos em informações úteis. Este fluxo, ilustrado na Figura 4.1, compreende um processo contínuo e iterativo, iniciando-se na coleta dos dados relevantes, que servem como matéria-prima para todo o trabalho. Em seguida, esses dados passam por uma fase de análise exploratória, onde buscamos entender suas características e identificar padrões. O próximo passo consiste no pré-processamento, uma etapa crucial para limpar, transformar e preparar os dados para modelagem. A engenharia de atributos entra em cena para refinar ainda mais os dados, criando novas variáveis para otimizar o desempenho dos modelos. Finalmente, os dados processados são consumidos pelos modelos de machine learning, resultando em uma série de predições do preço das ações do mercado financeiro, concretizando o objetivo central do projeto.

4.1
Coleta de dados

Para a coleta de dados, foram estudados e testados diferentes provedores de informações, levando em consideração diversos critérios, como exemplo, a

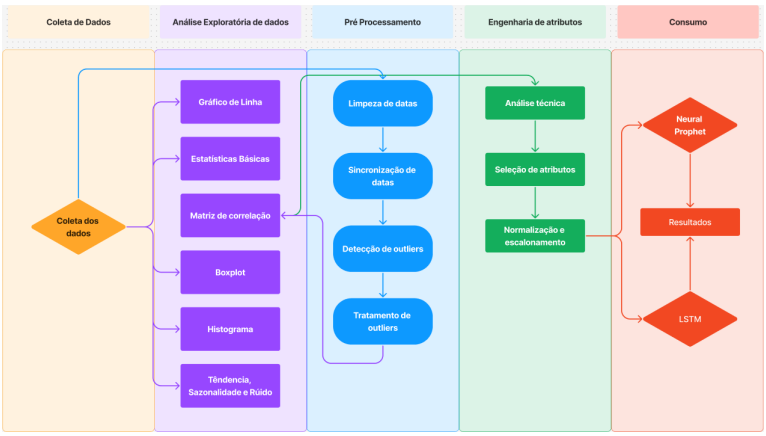

Figura 4.1: Fluxo de transformação de dados: da coleta ao consumo.

facilidade de integração com Python. Foi determinado como essencial que os provedores oferecessem uma API de fácil utilização para permitir a integração dos dados com a linguagem, facilitando assim o desenvolvimento do projeto.

Além disso, para o treinamento dos modelos, foi verificada a necessidade de dados históricos abrangentes, com pelo menos 10 anos de informações diárias disponíveis, e preferencialmente com opções de intervalos menores, como de 1 hora ou 1 minuto.

Outro critério importante foi a qualidade dos dados brutos, visto que em algumas APIs apresentavam dados repetidos em diferentes dias, o que poderia distorcer análises temporais.

Também foram consideradas as limitações das APIs, uma vez que algumas apresentavam restrições quanto ao número de solicitações que poderiam ser feitas dentro de um determinado período de tempo, o que poderia impactar negativamente a capacidade de coletar dados.

Considerando esses critérios , os provedores que tiveram maior destaque foram Yahoo Finance, TvDataFeed, Tiigo e Morningstar. Dentre eles, o TvDataFeed foi escolhido como o preferencial devido ao cumprimento de todos os pré-requisitos estabelecidos e à sua capacidade de fornecer dados do TradingView, uma plataforma renomada no fornecimento e análise de informações do mercado financeiro.

4.2 Análise Exploratória de Dados

A Análise Exploratória de Dados é uma etapa fundamental para o início deste projeto, proporcionando uma compreensão profunda e inicial do conjunto de dados utilizados. Essa abordagem não apenas revela padrões, tendências e anomalias nos dados, mas também verifica a qualidade e integridade das informações disponíveis, sendo essencial antes de passar para os passos de modelagem de dados e aprendizado de máquina. O processo para gerar cada componente da Análise, como ilustrado na Figura 4.2, é independente. Isso

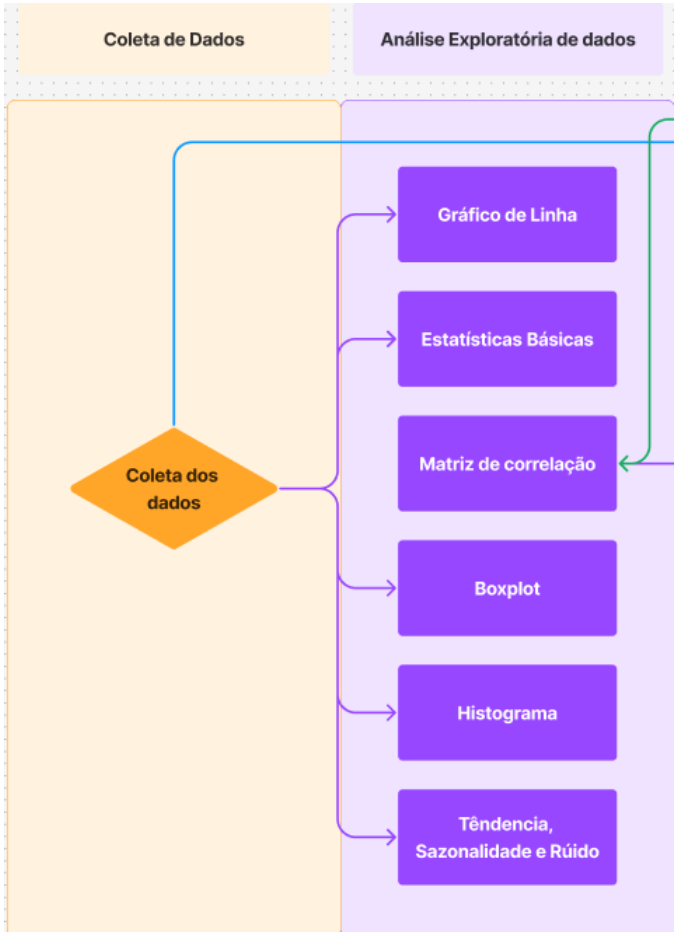

Figura 4.2: Análise exploratória de dados.

significa que a análise de cada técnica pode ser realizada separadamente, permitindo uma investigação mais detalhada e facilitando a identificação de relações e interações específicas sem interferência dos demais procedimentos.

É importante ressaltar que as séries dos ativos devem corresponder de forma boa de acordo com a AED, pois, caso contrário, elas podem ser eliminadas desse processo e não serão passadas para frente.

é possível observar pelo diagrama que a matriz de correlação não apresenta relação direta com a coleta de dados, isso porque ela vai receber as informações pós pré-processamento, e irá escolher os ativos menos correlacionados que irão fazer parte do processo final

A Análise Exploratória de Dados (AED) é uma etapa crucial no início deste projeto, proporcionando uma compreensão aprofundada do conjunto de

dados utilizado. Essa abordagem revela padrões, tendências e anomalias, além de avaliar a qualidade e integridade das informações disponíveis. A AED é essencial antes de avançar para as etapas futuras. O processo de geração de cada componente da análise, conforme ilustrado na Figura 4.2, é independente, permitindo que cada técnica seja explorada separadamente. Isso facilita uma investigação mais detalhada e a identificação de relações e interações específicas sem a interferência de outros procedimentos.

É fundamental que as séries dos ativos apresentem uma boa correspondência de acordo com a AED; caso contrário, elas poderão ser excluídas deste processo e não avançarão para as etapas seguintes.

O diagrama mostrado na figura 4.2 ilustra que a matriz de correlação não está diretamente relacionada à coleta de dados, pois será alimentada com informações após o pré-processamento. Essa matriz será utilizada para selecionar os ativos menos correlacionados, que serão utilizados no processo final.

4.3 Pré-Processamento

Diferentemente AED, o pré-processamento segue um fluxo estruturado, conforme ilustrado na Figura 4.5. O primeiro passo é a limpeza das datas, que envolve a remoção de feriados e finais de semana. Em seguida, realiza-se a normalização das datas, garantindo que todas as séries apresentem as mesmas datas. Séries que não cumprem essa regra são descartadas, conforme demonstrado no Algoritmo 4.3. Após essa etapa, procede-se com a identificação de outliers, descrita no Algoritmo 4.4. Os outliers identificados são então tratados por meio da aplicação da regra da média móvel com uma janela de 7 dias, conforme discutido na Subsubseção 3.2.3.3.

```
função validar_series_das_datas(séries):
    # Cria um conjunto para armazenar todas as datas
    todas_as_datas = conjunto vazio

    # Coleta todas as datas de cada série
    para cada série em séries:
        para cada data em série:
            todas_as_datas.adicionar(data)

    # Verifica quais séries têm todas as datas
    séries_validas = lista vazia

    para cada série em séries:
        se todas_as_datas.contém todas as datas em série:
            séries_validas.adicionar(série)

    # Retorna a lista de séries válidas
    retornar séries_validas
```

Figura 4.3: Função para validar séries de datas

4.4 Engenharia de atributos

Assim como no pré-processamento, a engenharia de atributos também segue um fluxo interdependente, conforme ilustrado na Figura 4.7. O primeiro passo envolve a transformação dos dados de preços de abertura, máxima, mínima e fechamento (OHLC) em indicadores técnicos, utilizando a biblioteca Pandas TA. Após essa transformação, os dados resultantes passam por um processo de seleção, onde os melhores atributos são escolhidos com base em sua relevância e impacto na performance do modelo. Esse processo de seleção é detalhado no algoritmo apresentado na Figura 4.6, que utiliza o Extra Tree Regressor para identificar os indicadores mais significativos.

Por fim, os dados selecionados são normalizados e escalados. Essas técnicas garantem que os diferentes atributos estejam em uma mesma escala, evitando que atributos com magnitudes maiores dominem o aprendizado dos modelos de machine learning. Uma vez que esses passos são concluídos, os dados processados estão prontos para serem alimentados nos modelos de

```
função identificar_outliers(serie, Periodo):
    # série: dados a serem analisados
    # Periodo: tamanho da janela da busca em dias

    # Inicializa lista para armazenar os outliers
    outliers = lista vazia

    # Para cada janela de tamanho Periodo na série
    para i de 0 até tamanho(serie) - Periodo:
        sub_serie = serie[i:i + Periodo]

        # Calcula IQR e limites para a sub-série
        limite_inferior, limite_superior = calcular_IQR(sub_serie)

        # Para cada ponto na sub-série
        para cada valor em sub_serie:
            # Verifica se o valor é um outlier pelo método IQR
            se valor < limite_inferior ou valor > limite_superior:
                outliers.adicionar(valor)

            # Calcula o Z-Score
            z_score = calcular_z_score(valor, sub_serie)

            # Verifica se o valor é um outlier pelo Z-Score
            se z_score > 3 ou z_score < -3:
                outliers.adicionar(valor)

    # Marca os outliers na série
    para cada outlier em outliers:
        marcar(outlier como outlier)

    retornar outliers
```

Figura 4.4: Função para identificar outliers em uma série de dados

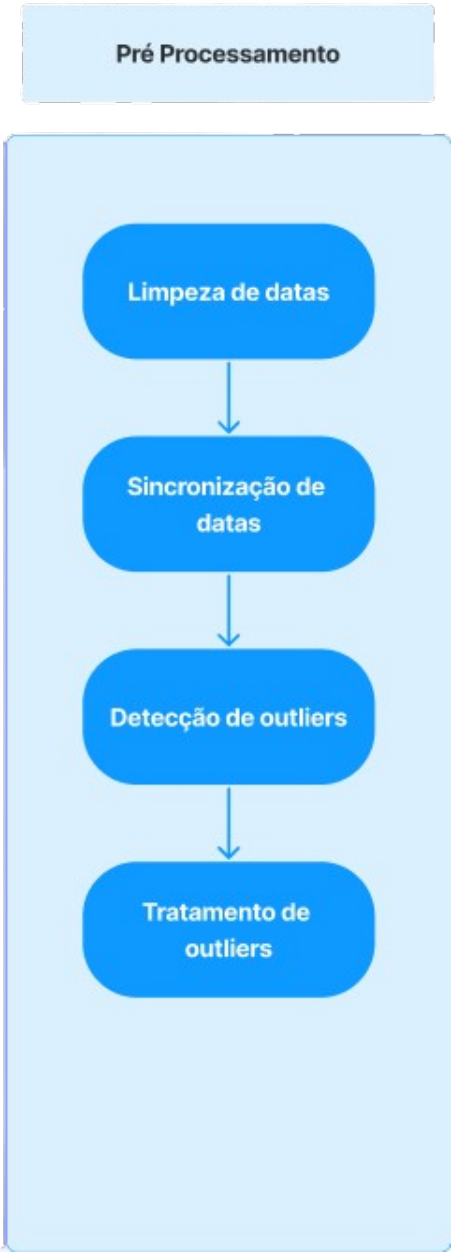

Figura 4.5: Fluxo pré-processamento dos dados.

machine learning, possibilitando uma análise mais eficaz e precisa.

Após a seleção de atributos, os dados são normalizados e escalonados utilizando a equação do estimador tangente hiperbólica, conforme descrito em 3.2.5.1. Esse processo resulta em dois arquivos CSV finais: um contendo os dados normalizados e escalonados, e outro que armazena as informações de média e desvio padrão, permitindo a reversão da normalização e do escalonamento posteriormente.

4.5 Consumo

A fase de consumo envolve a obtenção dos dados processados e sua aplicação de forma independente dos dois modelos, o NeuralProphet e o LSTM. Cada modelo consome os dados separadamente, realizando suas próprias análises e gerando previsões individuais para o preço de fechamento do ativo. O fluxograma dessa etapa, apresentado na Figura 4.8, ilustra esse fluxo, destacando como os dados seguem caminhos distintos para cada modelo. Essa abordagem independente permite comparar os resultados e obter insights complementares a partir de diferentes perspectivas analíticas.

```
função selecionar_indicadores_tecnicos_com_extra_tree(
    indicadores_tecnicos,
    preco_fechamento):

    # indicadores_tecnicos: matriz de atributos (features)
    # preco_fechamento: vetor de rótulos (target)

    # Cria o classificador Extra Tree
    regressor = ExtraTreeRegressor()

    # Treina o classificador com os dados
    regressor.treinar(indicadores_tecnicos, preco_fechamento)

    # Obtém a importância dos atributos
    importancias = regressor.importancia_dos_atributos()

    # Cria um DataFrame para armazenar a importância
    df_importancias = criar_data_frame(
        indicadores_tecnicos.colunas,
        importancias
    )

    # Ordena os atributos pela importância (do maior para o menor)
    df_importancias.ordenar_por_importancia()

    # Define um limite para seleção de atributos
    limite_importancia = 0.05  # Exemplo: 5% de importância mínima

    # Seleciona os atributos com importância acima do limite
    attr_selecionados = df_importancias.selecionar_acima(limite)


    retornar attr_selecionados
```

Figura 4.6: Função para obter os melhores indicadores da série

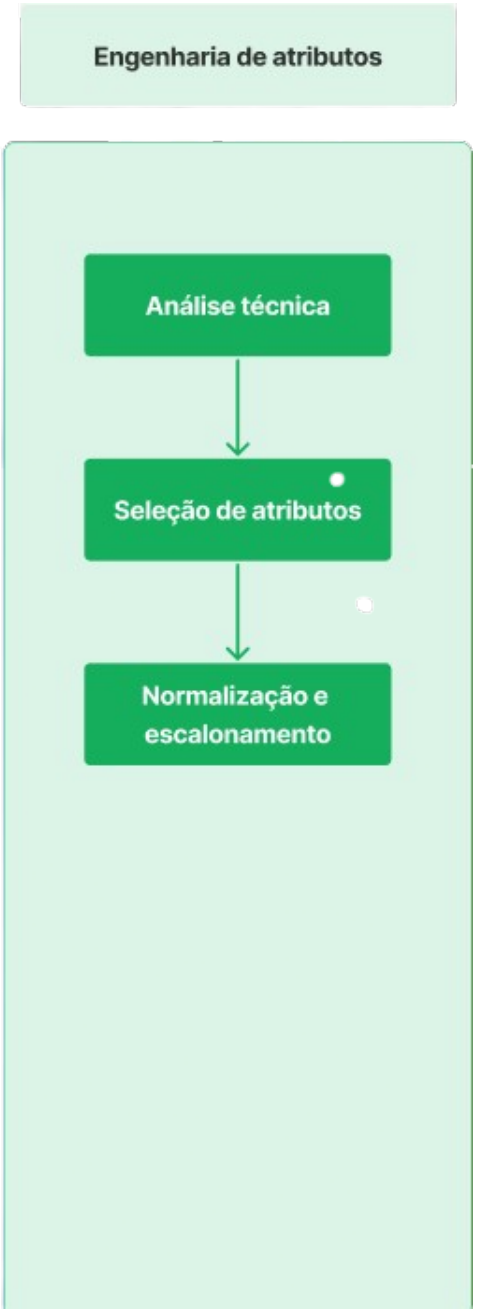

Figura 4.7: Fluxo engenharia de atributos.

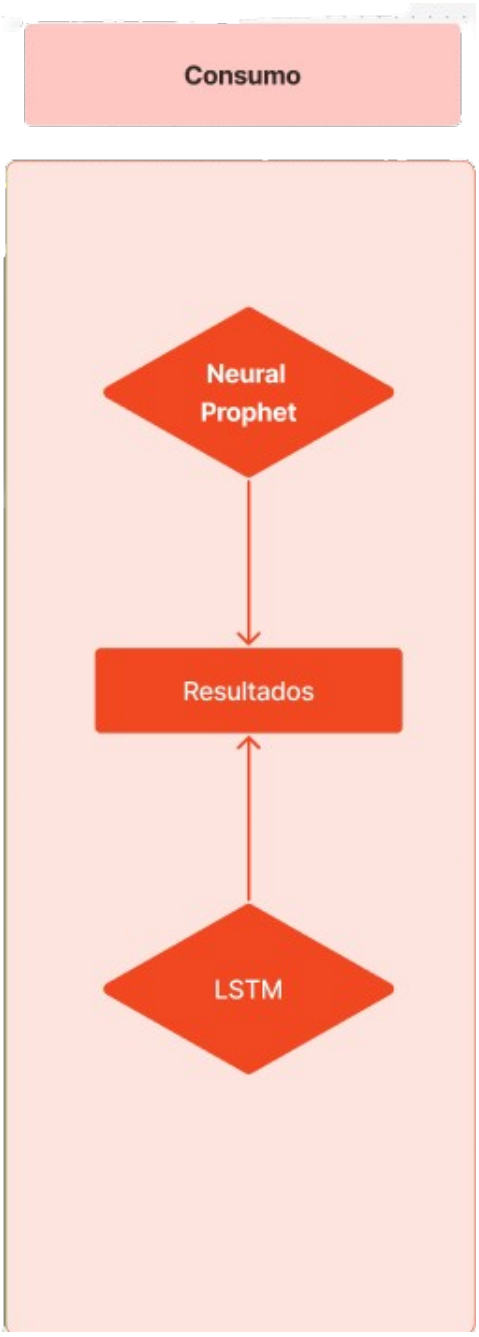

Figura 4.8: Fluxo de consumo.

5
Resultados

Este capítulo apresenta os resultados obtidos na predição de séries temporais, detalhando os principais achados e interpretações. Os dados foram analisados conforme descrito na metodologia, e os resultados são expostos a seguir.Inicialmente, são apresentados os aspectos gerais das análises realizadas, seguidos pela descrição do pré-processamento dos dados e da engenharia de atributos. Por fim, os resultados dos modelos aplicados são avaliados, com destaque para os padrões identificados e suas implicações.

5.1
Análise exploratória de dados

5.1.1
Resumo Estatístico

Tabela 5.1: Tabela de estatísticas descritivas do ativo MELI34

	Open	High	Low	Close	Volume
Count	1256	1256	1256	1256	1256
Mean	49.709	50.625	48.698	49.665	701949.940
Std	18.420	18.701	18.054	18.374	766898.296
Min	16.433	16.433	15.833	16.208	1200.000
25%	37.350	38.240	36.418	37.435	221361.500
50%	50.895	51.845	49.785	50.860	485242.000
75%	64.440	65.483	63.610	64.453	902577.750
Max	89.900	92.440	86.500	89.900	6805730.000

Conforme visto em 3.2.2 A tabela de estatísticas 5.1 apresenta as principais métricas descritivas das colunas coletadas. Dessa forma, é possível analisar dados importantes, como o tamanho do conjunto de dados, identificar possíveis outliers observando os valores mínimos e máximos, a distribuição geral dos valores, além de avaliar o desvio padrão para entender o comportamento dos dados da ação.

Na tabela 5.1, observamos que a ação MELI34 não possui dados suficientes para cobrir um período de pelo menos 10 anos, uma vez que foram coletados

apenas 1.256 registros, correspondendo a aproximadamente 5 anos de dados. Além disso, o elevado desvio padrão indica uma alta volatilidade dessa ação. Com base nas análises realizadas, concluímos que este ativo não é a melhor opção para previsões, devido à escassez de dados históricos disponíveis e à significativa volatilidade dos preços.

5.1.2 Boxplot

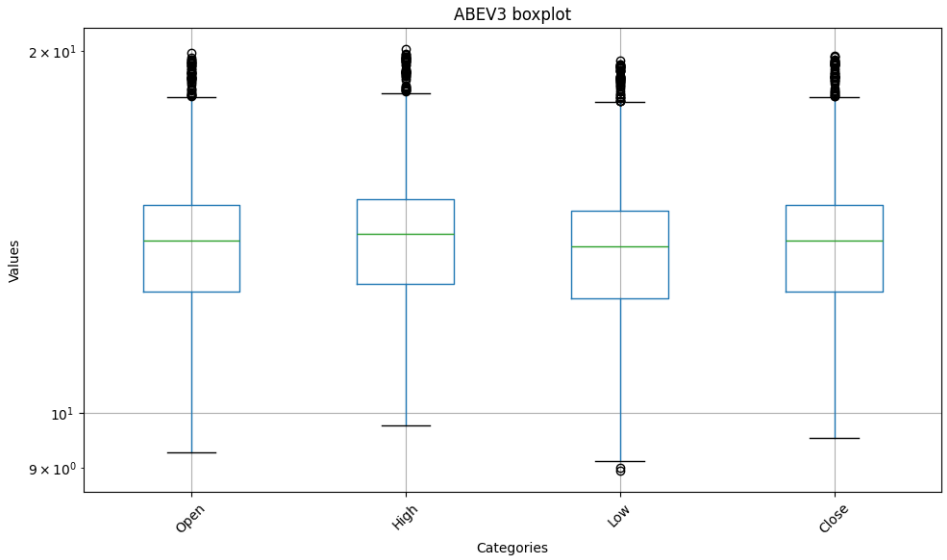

Figura 5.1: Boxplot representando a distribuição dos dados do ativo ABEV3, destacando medianas, quartis e outliers.

No gráfico de caixas que ilustra as ações do ativo ABEV3, da figura 5.1, é possível identificar uma quantidade significativa de valores atípicos, representados pelos círculos acima das caixas e na parte inferior da caixa "Low". Essa situação ressalta a necessidade de um cuidado redobrado ao analisar essa ação, em função do elevado número de anomalias observadas.

5.1.3 Histograma

O gráfico de histograma 5.2 é uma ferramenta útil para avaliar a distribuição de dados numéricos e identificar padrões inesperados no intervalo de dados.Em séries temporais, que geralmente possuem valores contínuos, o

histograma divide a extensão dos dados em intervalos e conta a quantidade de observações da série que se encontram em cada um desses intervalos.

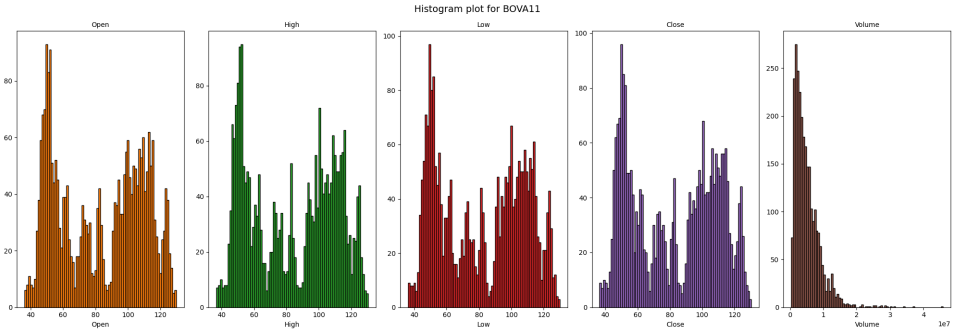

Figura 5.2: Histograma representando a distribuição dos dados do ativo BOVA11.

5.1.4 Gráfico de linha

O gráfico 5.3 destaca tendências significativas nas ações analisadas, evidenciando flutuações notáveis no preço ao longo do tempo. É possível observar tanto períodos de crescimento quanto de retração. Em particular, quando o preço apresenta quedas abruptas, como ocorreu em março de 2020 e novamente entre maio de 2022 e junho de 2023, essa situação pode indicar a presença de anomalias associadas a eventos ou notícias relevantes sobre a ação.

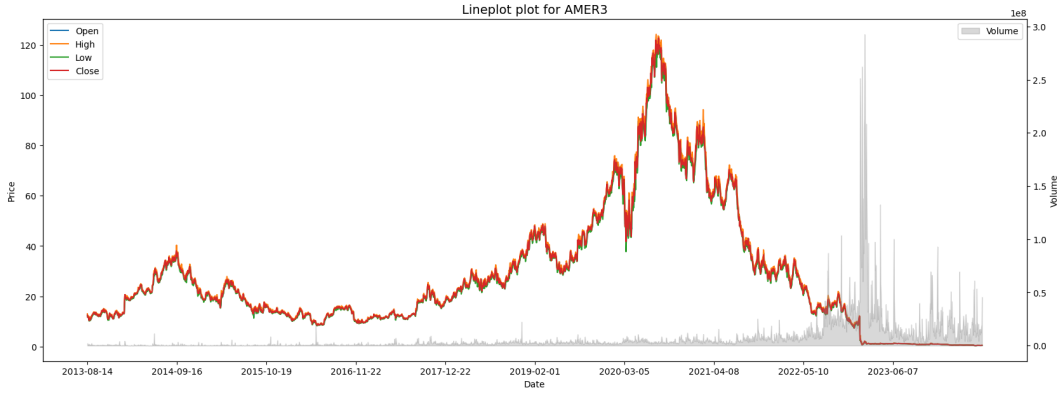

Figura 5.3: Gráfico de linha representando a variação histórica do ativo AMER3 ao longo do tempo.

5.1.5
Tendência, Sazonalidade e Ruído

Ao analisarmos os gráficos de tendência, sazonalidade e ruído, podemos extrair diversas informações importantes. Como ilustrado na figura 5.4 da ação ABEV3, a sazonalidade revela um padrão forte nos dados, com repetições anuais. Isso indica que o preço da ação segue um padrão previsível a cada ano. Observando o gráfico de tendências, percebe-se um movimento geral de alta entre 2014 e 2018, seguido por um pico e uma subsequente queda até 2020. Após 2020, a tendência se estabiliza com pequenas flutuações. Quanto ao componente de ruído, ele oscila em torno de zero, sugerindo que a maior parte das variações nos preços foi explicada pelos componentes sazonal e de tendência. Nota-se um aumento na volatilidade por volta de 2020, possivelmente devido à pandemia.

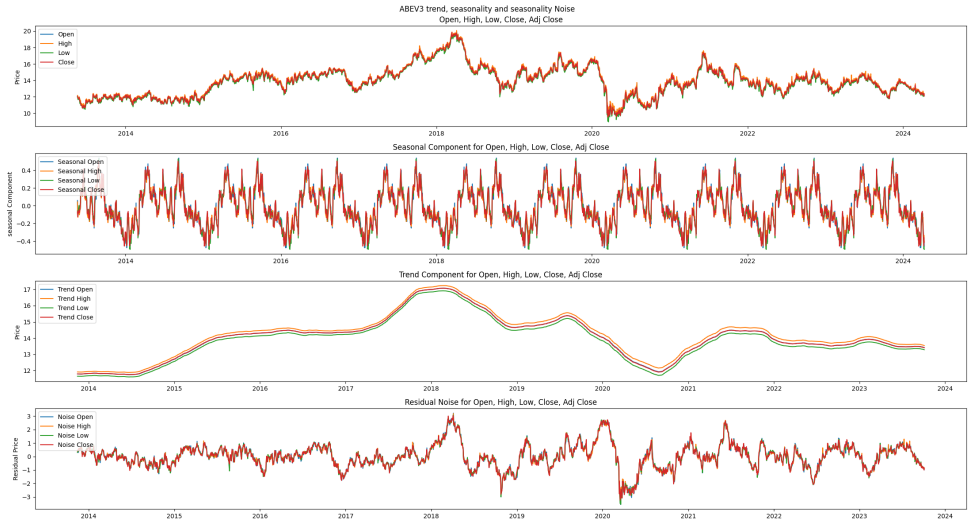

Figura 5.4: Figura representando os componentes de tendência, sazonalidade e ruído na série temporal do ativo ABEV3.

5.1.6
Matriz de correlação

A Matriz de correlação (Figura 5.5) é utilizada para analisar o comportamento geral dos ativos e suas interações, oferecendo uma visão abrangente das correlações entre eles. Além disso, ela permite identificar as ações com menor

correlação (Figura 5.6), que serão selecionadas para compor a carteira, assegurando maior independência entre os ativos e contribuindo para uma carteira diversificada.

Na figura 5.6, podemos observar que os ativos escolhidos para compor a carteira final estão listados na tabela 5.2.

Tabela 5.2: Ativos selecionados

Ativos selecionados			
ABEV3	BBAS3	BBDC3	BEEF3
CSNA3	MGLU3	PETR3	VALE3

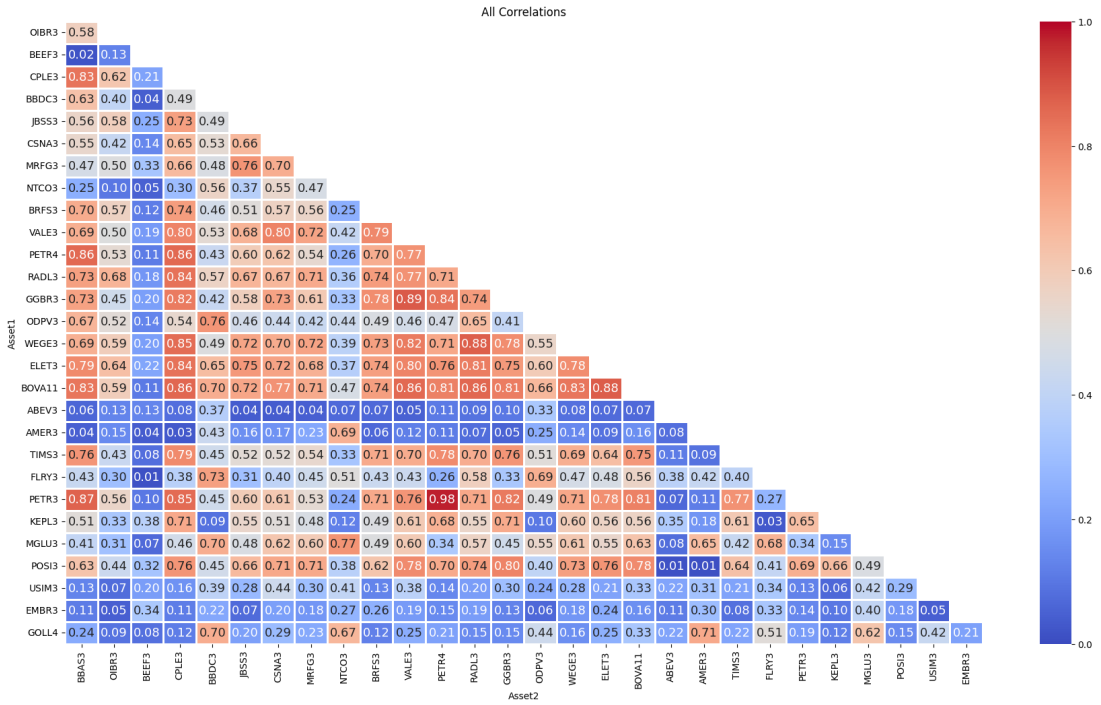

Figura 5.5: Matriz de correlação dos ativos, mostrando as correlações em valor absoluto entre variáveis

5.2
Pré Processamento

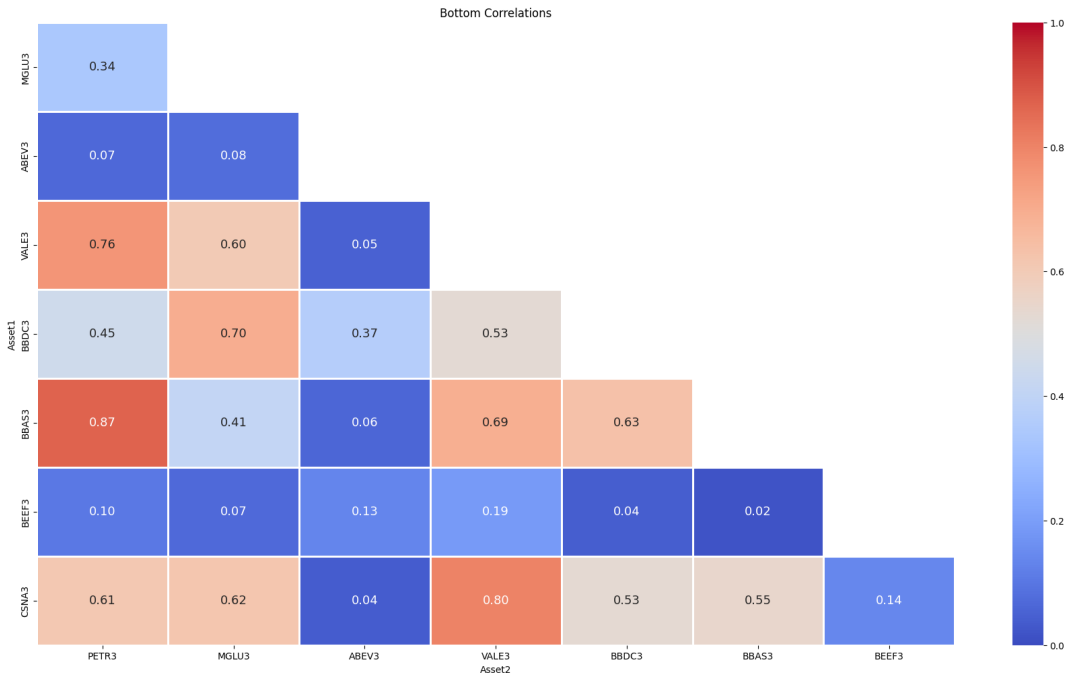

Figura 5.6: Menores correlações em valor absoluto na matriz de correlação dos ativos, evidenciando as relações mais fracas ou inversas entre variáveis.

5.2.1
Limpeza e normalização das Datas

A coleta de dados foi realizada em um intervalo de 10 anos. A limpeza das séries garantiu que apenas os dias úteis fossem mantidos. Durante o processo de normalização das datas, foi identificado que alguns ativos não possuíam registros completos para todo o período analisado. De acordo com a análise realizada no processo de AED, foi decidido manter essas séries, pois apresentavam dados brutos de boa qualidade. Como resultado, perderam-se 6 meses de informações, mas essa decisão foi necessária para garantir a qualidade dos resultados.

5.2.2
Detecção de Outliers

A Figura 5.7 ilustra as anomalias detectadas na série temporal do ativo BBAS3. É fundamental destacar que nem todas as anomalias identificadas serão consideradas outliers e, portanto, tratadas. As que refletem eventos significativos, como a pandemia de COVID-19 em 2020, devem ser preservadas,

pois representam o comportamento natural da série.

Ao examinar a distribuição dos outliers, observa-se que os outliers técnicos costumam exibir um padrão mais isolado, como os dois pontos extremos no início do gráfico. Em contraste, os outliers ocorridos em 2020 podem ser atribuídos a eventos externos, fornecendo uma compreensão mais ampla da dinâmica da série temporal.

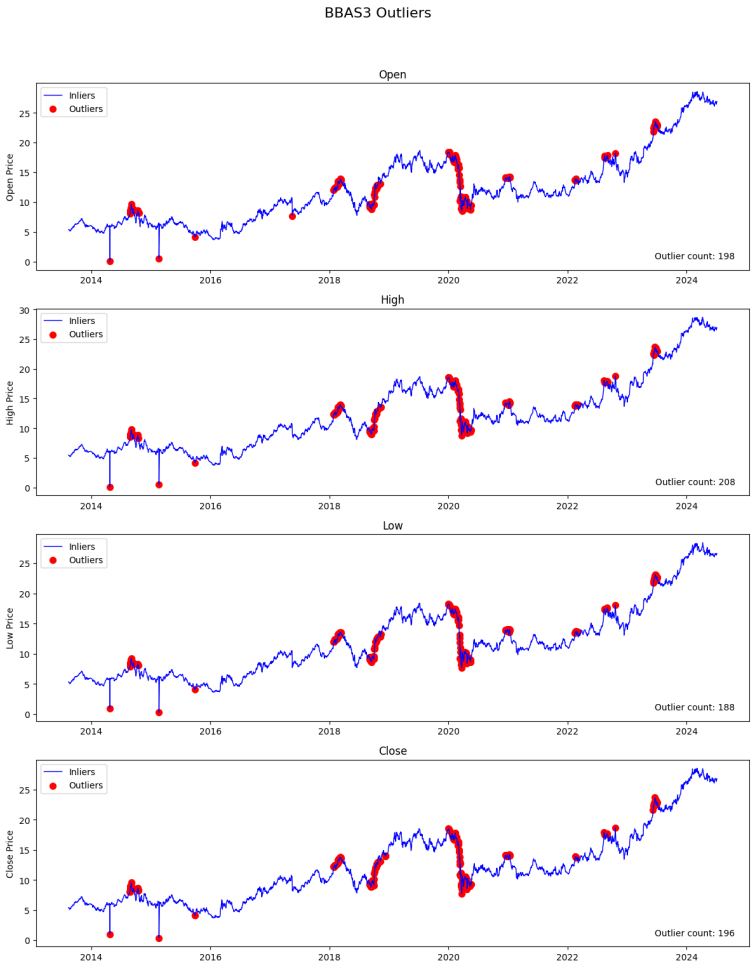

Figura 5.7: Gráfico de Anomalias do ativo BBAS3.

Na Tabela 5.3, apresenta-se a relação da quantidade de outliers dos ativos com base nos valores de open, high, low e close. Ressalta-se que nem todos os outliers identificados foram necessariamente tratados, sendo considerados apenas os outliers técnicos.

Tabela 5.3: Quantidade de outliers por ativo e tipo de valor.

Ativo	Open	High	Low	Close
ABEV3	210	207	205	210
BBAS3	198	208	188	196
BBDC3	233	261	225	228
BEEF3	201	196	202	201
CSNA3	239	244	241	244
MGLU3	235	227	237	244
PETR3	186	189	182	186
VALE3	219	224	227	227

5.2.3 Tratamento de outliers

Considerando a natureza dos outliers técnicos, caracterizados por serem valores discrepantes e não representativos da série, optou-se por uma abordagem de suavização local. A média móvel, com uma janela de 7 períodos, mostrou-se adequada para essa finalidade, pois permite substituir o valor do outlier por uma estimativa mais robusta, baseada nos valores vizinhos. Essa técnica é particularmente útil para outliers isolados, como os observados no início da série temporal do ativo BBAS3, garantindo que a correção não afete significativamente a tendência geral dos dados.

5.3 Engenharia de Atributos

5.3.1 Análise Técnica

Ao processar os dados de abertura, alta, baixa e fechamento utilizando a biblioteca Pandas Technical Analysis (Pandas TA)[1], os indicadores técnicos gerados são adicionados a uma planilha. Cada indicador técnico é representado como uma nova coluna, enquanto as linhas correspondem aos períodos de tempo dos dados.

Após a geração dos indicadores, a planilha resultante é salva em um arquivo CSV, que é amplamente usado para armazenar dados de forma

[1] https://github.com/twopirllc/pandas-ta

estruturada. Isso facilita o acesso e a análise posterior.

5.3.2 Seleção de Atributos

O objetivo desse processo é eliminar features irrelevantes e reduzir a dimensionalidade dos dados, atribuindo uma pontuação de importância a cada uma delas. Apenas as features com uma relevância superior a 3% serão mantidas, conforme ilustrado na Figura 5.8.

Ressalta-se que a figura 5.8 apresentada não contempla necessariamente todos os indicadores utilizados na análise. Além disso, os valores atribuídos a cada indicador podem variar de acordo com suas configurações. Por exemplo, o indicador SMA pode ser configurado com diferentes períodos, como SMA 10 ou SMA 20, o que influencia sua importância relativa no conjunto de dados.

A Tabela 5.4 exibe os indicadores selecionados para cada ativo, de acordo com o processo de seleção descrito nesta seção. Esses indicadores foram selecionados com base nos critérios apresentados anteriormente.

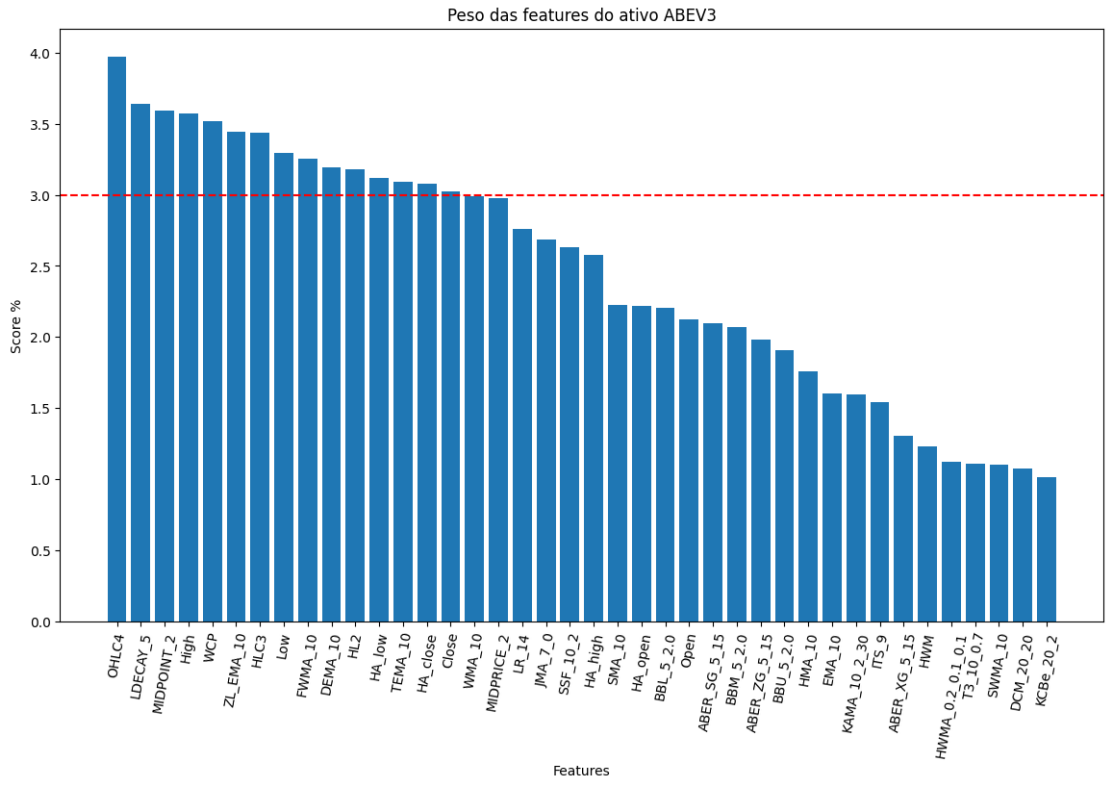

Figura 5.8: Importância das Features do Ativo ABEV3.

Tabela 5.4: Indicadores selecionados por ativos

Indicadores	ABEV3	BBAS3	BBDC3	BEEF3	CSNA3	MGLU3	PETR3	VALE3
OHLC4	X							
Linear Decay	X		X	X		X		
Midpoint	X	X		X	X	X	X	X
High	X		X	X	X		X	X
Weighted Closing Price	X			X			X	X
Exponential Moving Average	X			X		X	X	
HLC3	X	X	X	X	X		X	X
LOW		X		X			X	X
Fibonacci's Weighted Moving Average	X							
Double Exponential Moving Average	X			X				
HL2	X	X		X		X		X
Heiken Ashi	X	X	X	X	X	X	X	
Triple Exponential Moving Average	X	X		X		X		
Weighted Moving Average	X			X		X		
Bollinger Bands Middle Band						X	X	
Simple Moving Average						X	X	
Aberration		X					X	
OPEN							X	
Super Smoother Filter			X	X		X		
Hull Moving Average			X			X		
Jurik Moving Average			X					
Supertrend								X
Midprice				X				

5.3.3 Normalização e escalonamento

O resultado da normalização e escalonamento da série utilizando a função tanh foi como esperado, pois os dados foram transformados para um intervalo

entre -1 e 1. Essa transformação permitiu que as variações da série fossem mais uniformes e proporcionou uma melhor estabilidade nas análises subsequentes.

5.3.4 Avaliação e Comparação de Modelos

Neste projeto, foi utilizada como referência principal a métrica sMAPE devido à sua capacidade de oferecer uma avaliação mais equilibrada da precisão das previsões. O sMAPE, ao considerar a média dos valores reais e previstos, permite tratar de forma simétrica os erros de superestimação e subestimação, resultando em uma análise mais robusta.

O RMSE, embora amplamente utilizado, possui algumas limitações. Ele dá maior peso a erros maiores, pois os erros são elevados ao quadrado, o que pode distorcer a avaliação da precisão das previsões. Além disso, o MAPE, apesar de ser uma métrica comum para avaliar previsões, é indefinido quando o valor real é zero e tende a ser tendencioso em relação a valores baixos. Sua sensibilidade a outliers também pode comprometer a análise em dados voláteis.

Dessa forma, embora a escolha principal para este projeto seja o sMAPE, os outros indicadores, como RMSE e MAPE, não devem ser descartados. Cada métrica oferece uma perspectiva única sobre o desempenho do modelo, e uma análise mais abrangente pode ser obtida ao considerar todas elas em conjunto.

De acordo com (CHAI, 2014), valores de MAPE e sMAPE ideais estão em torno de 10%, indicando previsões precisas. Um RMSE ideal é aquele mais próximo de zero; entretanto, em cenários onde há maior dispersão nos valores dos preços, um RMSE mais elevado pode ser considerado aceitável (HOLAN ROBERT LUND, 2010).

A comparação entre os modelos Neural Prophet e LSTM mostrou que o Neural Prophet apresentou resultados consistentemente superiores. Dado o desempenho superior em todas as métricas analisadas, não foi considerada necessária a combinação dos dois modelos para tentar melhorar o desempenho, já que o Neural Prophet se destacou em todos os cenários.

Na figura 5.9, observamos o melhor desempenho geral de ambos os modelos, que ocorreu no ativo ABEV3. Nesse caso, as métricas apresentaram os seguintes resultados, respectivamente, para o NeuralProphet e para o LSTM:

- . O RMSE do modelo Prophet (0.33) é aceitável, enquanto o LSTM (0.49) sugere um desempenho inferior, porém também aceitável
- MAPE Ambos os modelos apresentam resultados muito baixos, sendo o Prophet 1.92% superior ao LSTM 2.85%.
- O Prophet 1.92% atende a esse critério, enquanto o LSTM 2.90% está um pouco acima, mas ainda competitivo.

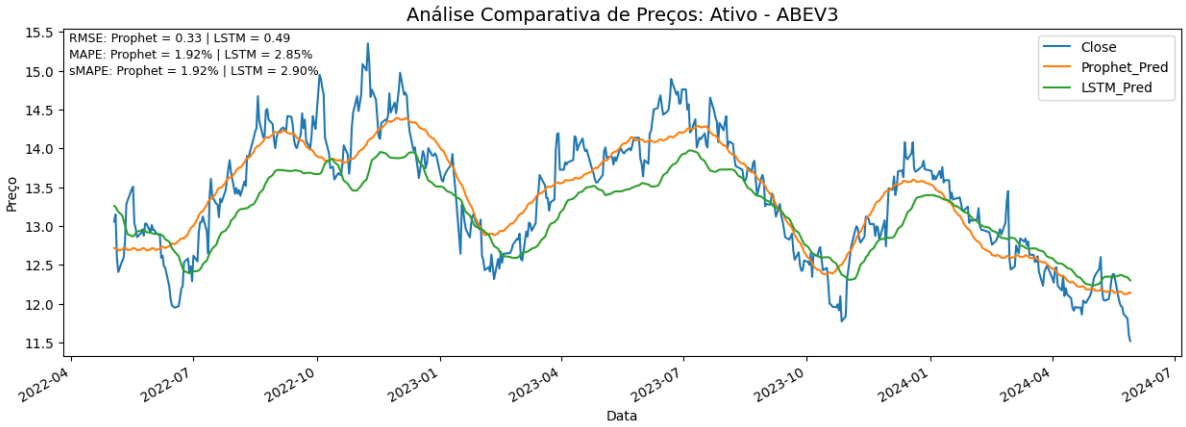

Figura 5.9: Análise dos modelos na série temporal do ativo ABEV3

Na figura 5.9, podemos observar o melhor desempenho geral de ambos os modelos, que foi no ativo ABEV3. O preço do ativo varia entre 11,5 e 15,5. Isso significa que:

Esses resultados destacam a eficácia do modelo NeuralProphet em prever o ativo ABEV3 em comparação ao LSTM, especialmente em termos de precisão e erro absoluto.

Na figura 5.10, podemos observar que o erro é maior do que o observado na figura anterior. Os resultados das métricas de desempenho para os modelos Prophet e LSTM são os seguintes:

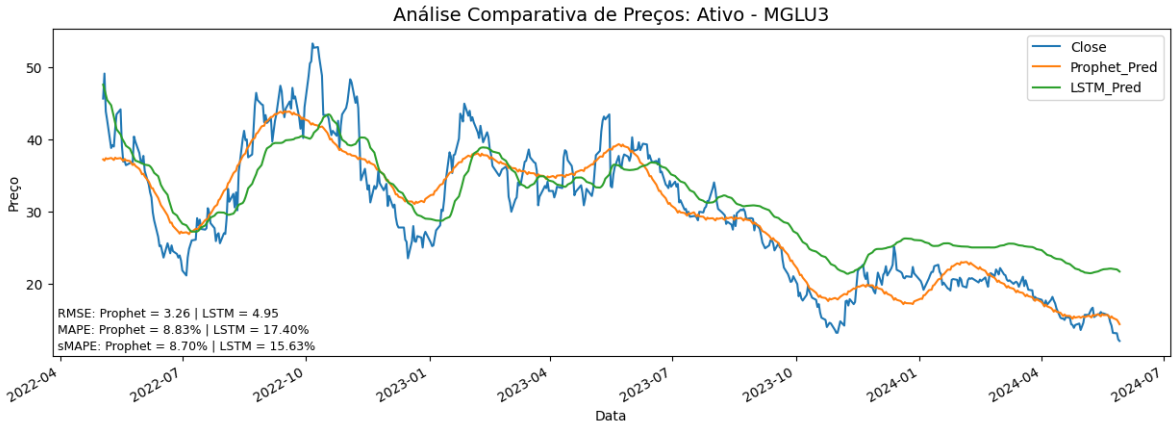

Figura 5.10: Análise dos modelos na série temporal do ativo MGLU3

- RMSE Prophet 3.26 e LSTM 4.95, indicando que ambos os modelos apresentam erros absolutos significativos, sendo o modelo Prophet relativamente mais preciso.
- MAPE: Prophet 8.83% e LSTM 17.40%, refletindo uma precisão que passa um pouco do ótimo nas previsões, sendo valores consideravelmente maiores em comparação aos resultados do ativo ABEV3.
- sMAPE Prophet 8.70% e LSTM 15.63%, sugerindo que, embora o Prophet tenha um desempenho superior, ambos os modelos têm um erro percentual significativo em relação à média dos valores reais e previstos.

A tabela 5.5 apresenta todos os resultados das métricas para os modelos e os ativos analisados.

A partir dos resultados apresentados na Tabela 5.6, observa-se que o modelo NeuralProphet apresentou um desempenho superior em todas as métricas avaliadas em comparação ao modelo LSTM. O RMSE do NeuralProphet foi significativamente menor (1.195 contra 2.615), indicando que este modelo conseguiu manter erros absolutos mais baixos. Além disso, o MAPE do NeuralProphet 4.013% também foi inferior ao do LSTM 9.865%, demonstrando maior precisão relativa nas previsões. Por fim, o sMAPE, que é uma métrica mais robusta para avaliar erros de previsão, reforça a superioridade do NeuralProphet, com um valor de 3.989% contra 9.836% do LSTM. Esses resultados

Tabela 5.5: Desempenho das Métricas RMSE, MAPE e sMAPE para os Modelos Neural Prophet e LSTM nos Ativos

Ação	Métrica	Neural Prophet	LSTM
ABEV3	RMSE	0.33	0.49
	MAPE(%)	1.92	2.85
	sMAPE(%)	1.92	2.90
BBAS3	RMSE	0.57	3.42
	MAPE(%)	2.47	13.50
	sMAPE(%)	2.46	14.67
BBCD3	RMSE	0.59	0.82
	MAPE(%)	3.51	4.92
	sMAPE(%)	3.51	5.09
BEEF3	RMSE	0.53	1.47
	MAPE(%)	4.19	15.09
	sMAPE(%)	4.18	13.50
CSNA3	RMSE	0.73	1.26
	MAPE(%)	4.43	6.70
	sMAPE(%)	4.42	6.98
MGLU3	RMSE	3.26	4.95
	MAPE(%)	8.83	17.40
	sMAPE(%)	8.70	15.63
PETR3	RMSE	1.16	5.23
	MAPE(%)	3.82	14.66
	sMAPE(%)	3.79	16.10
VALE3	RMSE	2.39	3.28
	MAPE(%)	2.93	3.80
	sMAPE(%)	2.93	3.82

indicam que o NeuralProphet foi mais eficaz em capturar os padrões da série temporal analisada.

A análise dos resultados indica que os modelos de previsão, com destaque para o Neural Prophet, são eficazes em capturar as tendências e os padrões sazonais da série temporal, conseguindo prever valores dentro de uma margem de erro aceitável. No entanto, podem surgir dificuldades na previsão precisa de picos, resultando em discrepâncias entre os valores reais e previstos. Isso ocorre porque os picos geralmente representam eventos inesperados ou variações súbitas, que são mais difíceis de prever e podem não seguir os padrões previamente observados na série.

Tabela 5.6: Resultados das Métricas para os Modelos Neural Prophet e LSTM

Modelo	RMSE	MAPE (%)	sMAPE (%)
Neural Prophet	1.195	4.013	3.989
LSTM	2.615	9.865	9.836

Além disso, o modelo LSTM também apresenta um certo atraso ao prever a tendência em relação ao modelo Neural Prophet, o que pode impactar sua capacidade de responder rapidamente a mudanças nos dados.

6
Conclusão

Este trabalho teve como objetivo aplicar técnicas de machine learning em um conjunto de dados para prever os preços das ações, visando auxiliar potenciais investidores. Os objetivos específicos foram alcançados, com a busca por previsões dentro do erro aceitável definido pelas métricas estabelecidas.

Foram conduzidos diversos experimentos utilizando algoritmos de aprendizado de máquina para prever os preços das ações, resultando em modelos com alta acurácia. Vários desses modelos apresentaram uma média de erro inferior a 10% no sMAPE, que foi estabelecido como a métrica principal do projeto, evidenciando um elevado nível de precisão nas previsões.

Os modelos desenvolvidos podem ser utilizados para gerar recomendações de compra ou venda para investidores. Além disso, este projeto pode servir como uma ferramenta educacional, ajudando os investidores a interpretar previsões e tomar decisões fundamentadas em dados. Por meio da visualização dos dados e dos resultados das previsões, os investidores podem aprender sobre as dinâmicas do mercado, identificar tendências e compreender melhor os fatores que influenciam os preços das ações. Esse conhecimento pode capacitar os investidores a se tornarem mais autônomos e críticos em suas escolhas.

Através do desenvolvimento deste estudo e dos resultados obtidos, identificou-se a oportunidade de expandir o trabalho com projetos futuros. Um desses projetos é a automatização de trades, que poderia conectar o modelo a uma corretora, permitindo a execução de negociações baseadas em previsões e regras predefinidas.

Outro projeto viável é a implementação de uma plataforma de simulação de investimento, que permitiria aos usuários praticar estratégias de compra e venda sem risco financeiro, utilizando as previsões do modelo. Essa plataforma ofereceria um ambiente seguro para investidores iniciantes testarem suas deci-

sões e aprenderem a interpretar as previsões em diferentes cenários de mercado. Com essa ferramenta, os usuários teriam a oportunidade de experimentar e refinar suas abordagens de investimento, aumentando sua confiança e habilidades antes de entrar no mercado real.

Para aprimorar o desempenho na previsão, é fundamental abordar algumas limitações e desafios inerentes a esse tipo de análise. As principais questões que podem comprometer a precisão das previsões incluem:

1. **Instabilidade da série temporal**: As séries temporais de ações são frequentemente influenciadas por uma variedade de eventos, como crises econômicas, mudanças regulatórias e resultados financeiros. Essas flutuações podem causar instabilidades nos dados, dificultando a previsão. Eventos imprevistos podem levar a oscilações bruscas nos preços, resultando em erros significativos nas previsões.

2. **Ajustes finos no modelo**: Modelos de machine learning frequentemente necessitam de ajustes cuidadosos, como a seleção de hiperparâmetros e a aplicação de técnicas de validação cruzada. Esses ajustes são cruciais para otimizar o desempenho do modelo, mas podem ser complexos e exigir uma compreensão profunda dos dados e do comportamento do mercado.

3. **Quantidade de dados**: A precisão das previsões tende a aumentar com a disponibilidade de um conjunto de dados mais robusto. Ter acesso a dados históricos extensos permite que os modelos aprendam padrões mais complexos e sutis. Portanto, a escassez de dados pode limitar a capacidade do modelo de generalizar e prever com acurácia.

4. **Falta de volume como variável**: Ao utilizar apenas os dados de preços de abertura, alta, baixa e fechamento (OHLC), a análise pode ser significativamente enriquecida com a inclusão do volume de negociações. O volume serve como uma variável crucial, pois fornece insights sobre

a liquidez do ativo e o interesse dos investidores. Além disso, pode ser utilizado para calcular indicadores técnicos, que podem contribuir para a melhoria das previsões.

5. **Análise de Sentimento do Mercado**: A integração de dados de notícias financeiras e redes sociais permite analisar o impacto do sentimento do mercado nos preços das ações. Esses dados podem ser utilizados como variáveis no modelo, aprimorando a precisão das previsões ao capturar a psicologia dos investidores e suas reações a eventos relevantes. Além disso, é importante considerar informações sobre eventos reais, como desastres naturais e acontecimentos corporativos (ex: fusões, aquisições), que também influenciam o sentimento do mercado e podem afetar significativamente os preços das ações.

Em conclusão, os resultados obtidos foram satisfatórios, demonstrando um desempenho promissor na previsão. No entanto, para aprimorar ainda mais essa eficácia, é essencial considerar os fatores mencionados que podem ter dificultado os resultados dos modelos. As limitações apresentadas podem impactar a precisão das previsões, indicando que há espaço para melhorias e ajustes nos métodos utilizados. A necessidade de ajustes finos nos modelos, a quantidade de dados disponíveis e a falta de volume como variável são questões que podem ser abordadas. Ao reconhecer e trabalhar essas limitações, será possível refinar os modelos preditivos, potencializando sua eficácia. Portanto, futuras análises devem se concentrar em superar esses desafios para otimizar os resultados obtidos.

7
Referências bibliográficas

B3, R. **Número de investidores na B3 cresce 34% em renda fixa e 23% em renda variável em 12 meses**. 2023. Disponível em: https://www.b3.com.br/pt_br/noticias/numero-de-investidores-na-b3-cresce-34-em-renda-fixa-e-23-em-renda-variavel-em-12-meses.htm. Citado na página 1.

CINTRA, L. A. **Prejuízo abate 90% de quem tenta viver como day trader, indica estudo: Levantamento usou dados de 20 mil investidores durante seis anos; menos de 1% tem ganhos relevantes**. 2022. Disponível em: https://www1.folha.uol.com.br/mercado/2022/04/prejuizo-abate-90-de-quem-tenta-viver-como-day-trader-indica-estudo.shtml. Citado na página 1.

NC, R. **Aposta no 'jogo do tigrinho' é 7 vezes maior que investimento na bolsa no país**. 2024. Publicado em: 01/05/2024; Atualizado em: 01/05/2024; Visto em: 12/05/2024 – 17h41. Disponível em: https://noticiasdocentro.com.br/realidade/brasil/aposta-no-jogo-do-tigrinho-e-7-vezes-maior-que-investimento-na-bolsa/. Citado na página 1.

PURCHIO, L. **Jovens e conectados: novos investidores da Bolsa buscam informação na web: Estudo da B3 mostra que entre os 2 milhões de novos investidores da bolsa, cerca de 60% acompanham influencers digitais**. 2021. Disponível em: https://veja.abril.com.br/economia/jovens-e-conectados-novos-investidores-da-bolsa-buscam-informacao-na-web. Citado na página 2.

KHAIDEM, L.; SAHA, S.; DEY, S. R. Predicting the direction of stock market prices using random forest. **arXiv preprint arXiv:1605.00003**, 2016. Disponível em: https://arxiv.org/pdf/1605.00003.pdf. Citado na página 2.

STRADER, T. J. et al. Machine learning stock market prediction studies: Review and research directions. **Journal of International Technology and Information Management**, John M. Pfau Library, California State University San Bernardino, v. 28, p. 63–83, JAN 2020. ISSN 1941-6679. Disponível em: https://dx.doi.org/10.58729/1941-6679.1435. Citado 2 vezes nas páginas 2 e 3.

PARMAR, I. et al. Stock market prediction using machine learning. **2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)**, IEEE, Not available, p. Not available, DEC 2018. ISSN Not available. Disponível em: https://dx.doi.org/10.1109/icsccc.2018.8703332. Citado 2 vezes nas páginas 2 e 3.

PIOVEZAN, R. **Recomendação de investimento feita por IA será próximo boom, mas é preciso cautela, dizem especialistas**. 2024. Publicado em: 02/10/2024;. Disponível em: https://borainvestir.b3.com.br/objetivos-financeiros/

recomendacao-de-investimento-feita-por-ia-sera-proximo-boom-nas-instituicoes-financeiras-dizem-especia
Citado na página 3.

PIOVEZAN, R. **Chatbot do Tesouro Direto e Claude IA ajudam a avaliar investimentos — mas com cuidado**. 2024. Publicado em: 26/09/2024;. Disponível em: https://borainvestir.b3.com.br/objetivos-financeiros/investir-melhor/chatbot-do-tesouro-direto-e-claude-ia-ajudam-a-avaliar-investimentos-mas-com-cuidado/. Citado na página 3.

RAVIKUMAR, S.; SARAF, P. Prediction of stock prices using machine learning (regression, classification) algorithms. **2020 International Conference for Emerging Technology (INCET)**, IEEE, Not available, p. Not available, JUN 2020. ISSN Not available. Disponível em: https://dx.doi.org/10.1109/incet49848.2020.9154061. Citado na página 3.

MORETTIN, P.; TOLOI, C. **Previsão de séries temporais**. 2nd. ed. São Paulo: Atual Editora, 1987. Citado na página 6.

HYNDMAN, R.; ATHANASOPOULOS, G. fpp: Data for "forecasting: principles and practice". **CRAN: Contributed Packages**, The R Foundation, Jun 2011. Disponível em: https://doi.org/10.32614/cran.package.fpp. Citado na página 6.

TRIEBE NIKOLAY LAPTEV, R. R. O. J. Ar-net: A simple auto-regressive neural network for time-series. **arXiv preprint arXiv:1911.12436**, 2019. Acesso em: 15 out. 2024. Disponível em: https://arxiv.org/pdf/1911.12436.pdf. Citado 2 vezes nas páginas 7 e 8.

HOLAN ROBERT LUND, G. D. S. H. The arma alphabet soup: A tour of arma model variants. **Statistics Surveys**, Institute of Mathematical Statistics, v. 4, n. none, Jan 2010. ISSN 1935-7516. Disponível em: https://doi.org/10.1214/09-ss060. Citado 2 vezes nas páginas 8 e 68.

TANG, P. A. F. Z. Feedforward neural nets as models for time series forecasting. **ORSA Journal on Computing**, Institute for Operations Research and the Management Sciences (INFORMS), v. 5, n. 4, p. 374–385, Nov 1993. ISSN 0899-1499. Disponível em: https://doi.org/10.1287/ijoc.5.4.374. Citado na página 8.

HYNDMAN, R. fpp2: Data for "forecasting: Principles and practice"(2nd edition). **CRAN: Contributed Packages**, The R Foundation, Feb 2017. Disponível em: https://doi.org/10.32614/cran.package.fpp2. Citado 2 vezes nas páginas 8 e 9.

TRIEBE, O. Neuralprophet: Explainable forecasting at scale. **arXiv preprint arXiv:2111.15397**, NOV 2021. https://arxiv.org/abs/2111.15397. Disponível em: https://arxiv.org/abs/2111.15397. Citado 2 vezes nas páginas 16 e 17.

HARVEY, N. S. A. C. 10 structural time series models. **Handbook of Statistics**, Elsevier, p. 261–302, 1993. ISSN 0169-7161. Disponível em: https://doi.org/10.1016/s0169-7161(05)80045-8. Citado na página 16.

ZHANG, Y. Stock price prediction using lstm model. **Highlights in Science, Engineering and Technology**, Darcy & Roy Press Co. Ltd., v. 44, p. 302–306, apr 2023. ISSN 2791-0210. Disponível em: https://dx.doi.org/10.54097/hset.v44i.7352. Citado na página 20.

S, S. K.; D, C.; RAJAN, S. Stock price prediction using deep learning lstm (long short-term memory). **2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)**, IEEE, Not available, p. 1787–1791, APR 2022. ISSN Not available. Disponível em: https://dx.doi.org/10.1109/icacite53722.2022.9823639. Citado na página 20.

S, S. K.; D, C.; RAJAN, S. Stock price prediction using deep learning lstm (long short-term memory). **2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)**, IEEE, Not available, p. 1787–1791, APR 2022. ISSN Not available. Disponível em: https://dx.doi.org/10.1109/icacite53722.2022.9823639. Citado na página 20.

ATWAN, T. A. **Time Series Analysis with Python Cookbook**. [S.l.]: Packt Publishing, 2022. 0 p. ISBN 9781801075541. Citado 2 vezes nas páginas 26 e 30.

ZHENG, A.; CASARI, A. **Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists**. [S.l.]: O'Reilly Media, Apr. 218 p. Citado na página 29.

GARCíA, S.; LUENGO, J.; HERRERA, F. Data preprocessing in data mining. **Intelligent Systems Reference Library**, Springer International Publishing, Not available, p. Not available, undefined 2015. ISSN 1868-4394, 1868-4408. Disponível em: https://dx.doi.org/10.1007/978-3-319-10247-4. Citado 2 vezes nas páginas 29 e 31.

ALEXANDROPOULOS, S.-A. N.; KOTSIANTIS, S. B.; VRAHATIS, M. N. Data preprocessing in predictive data mining. **The Knowledge Engineering Review**, Cambridge University Press (CUP), v. 34, p. Not available, undefined 2019. ISSN 0269-8889, 1469-8005. Disponível em: https://dx.doi.org/10.1017/s026988891800036x. Citado na página 29.

YU, Y. et al. Time series outlier detection based on sliding window prediction. **Mathematical Problems in Engineering**, Hindawi Limited, v. 2014, p. 1–14, undefined 2014. ISSN 1024-123X, 1563-5147. Disponível em: https://dx.doi.org/10.1155/2014/879736. Citado na página 30.

TAWAKULI, A. et al. Survey:time-series data preprocessing: A survey and an empirical analysis. **Journal of Engineering Research**, Elsevier BV, Not available, p. Not available, MAR 2024. ISSN 2307-1877. Disponível em: https://dx.doi.org/10.1016/j.jer.2024.02.018. Citado na página 30.

LI, Y.; LI, T. Feature selection and evaluation. **Feature Engineering for Machine Learning and Data Analytics**, CRC Press, Not available, p. 191–220, MAR 2018. ISSN Not available. Disponível em: https://dx.doi.org/10.1201/9781315181080-8. Citado na página 31.

PHUOC, T. et al. Applying machine learning algorithms to predict the stock price trend in the stock market – the case of vietnam. **Humanities and Social Sciences Communications**, Springer Science and Business Media LLC, v. 11, p. Not available, MAR 2024. ISSN 2662-9992. Disponível em: https://dx.doi.org/10.1057/s41599-024-02807-x. Citado 2 vezes nas páginas 31 e 40.

TRADINGVIEW. **Bollinger Bands (BB)**. 2024. https://www.tradingview.com/wiki/Bollinger_Bands_(BB) [Accessed: (2024-09-23)]. Citado na página 32.

TRADINGVIEW. **Average True Range (ATR)**. 2024. https://www.tradingview.com/support/solutions/43000501823-average-true-range-atr/ [Accessed: (2024-09-23)]. Citado na página 33.

TRADINGTECHNOLOGIES. **Documentation | Trading Technologies**. 2024. https://www.tradingtechnologies.com/help/x-study/technical-indicator-definitions/average-directional-movement-adx/ [Accessed: (2024-09-23)]. Citado na página 34.

TRADINGVIEW. **Aroon**. 2024. https://www.tradingview.com/support/solutions/43000501801-aroon/ [Accessed: (2024-09-23)]. Citado na página 35.

BIO, F. **How Is the Exponential Moving Average (EMA) Formula Calculated?** 2024. https://www.investopedia.com/ask/answers/122314/what-exponential-moving-average-ema-formula-and-how-ema-calculated.asp. Citado na página 36.

TRADINGVIEW. **Stochastic (STOCH)**. 2024. https://www.tradingview.com/support/solutions/43000502332-stochastic-stoch/. Citado na página 38.

TRADINGVIEW. **MACD (Moving Average Convergence/Divergence)**. 2024. https://www.tradingview.com/support/solutions/43000502344-macd-moving-average-convergence-divergence/. Citado na página 39.

TALUKDER, S. H. et al. Heart disease risk assessment and prediction: A robust ensemble approach with extra tree classifier. **2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI)**, IEEE, Not available, p. 1–6, DEC 2023. ISSN Not available. Disponível em: https://dx.doi.org/10.1109/eiceeai60672.2023.10590147. Citado 2 vezes nas páginas 40 e 41.

TALUKDER, M. S. H. **Unleashing the Power of Extra-Tree Feature Selection and Random Forest Classifier for Improved Survival Prediction in Heart Failure Patients**. 2023. https://arxiv.org/abs/2308.05765. Citado 2 vezes nas páginas 40 e 41.

NAYAK, S. C.; MISRA, B. B.; BEHERA, H. S. Evaluation of normalization methods on neuro-genetic models for stock index forecasting. **2012 World Congress on Information and Communication Technologies**, IEEE, Not available, p. Not available, OCT 2012. ISSN Not available. Disponível em: https://dx.doi.org/10.1109/wict.2012.6409147. Citado na página 43.

BHANJA, A. D. S. Impact of data normalization on deep neural network for time series forecasting. **arXiv preprint arXiv:1812.1812**, 2018. Disponível em: https://arxiv.org/pdf/1812.05519. Citado na página 43.

CHAI, R. R. D. T. Root mean square error (rmse) or mean absolute error (mae)? – arguments against avoiding rmse in the literature. **Geoscientific Model Development**, Copernicus GmbH, v. 7, n. 3, p. 1247–1250, Jun 2014. ISSN 1991-9603. Disponível em: https://doi.org/10.5194/gmd-7-1247-2014. Citado na página 68.