THIAGO RAYAM SOUZA SANTOS

Sistema de tomada de decisão no mercado de ações utilizando aprendizado de máquina

São Paulo
2023

THIAGO RAYAM SOUZA SANTOS

Sistema de tomada de decisão no mercado de ações utilizando aprendizado de máquina

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

São Paulo
2023

THIAGO RAYAM SOUZA SANTOS

Sistema de tomada de decisão no mercado de ações utilizando aprendizado de máquina

Versão Corrigida

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

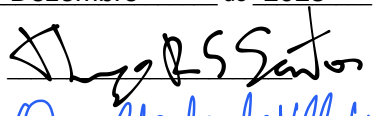
Área de Concentração:
Engenharia de Sistemas

Orientador:
Oswaldo Luiz do Valle Costa

São Paulo
2023

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 18 de Dezembro de 2023

Assinatura do autor: 

Assinatura do orientador: 

Catalogação-na-publicação

Santos, Thiago
Sistema de tomada de decisão no mercado de ações utilizando aprendizado de máquina / T. Santos -- versão corr. -- São Paulo, 2023.
74 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Telecomunicações e Controle.

1.Aprendizado de máquina 2.Sistema de Trading I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Telecomunicações e Controle II.t.

Dedicatória

À minha família, em especial à minha mãe e esposa, agradeço pela compreensão e apoio durante todo o curso.

AGRADECIMENTOS

Ao Professor Dr. Oswaldo Luiz do Valle Costa pela orientação ao longo deste trabalho. Sua paciência e disponibilidade foram essenciais para o desenvolvimento desta dissertação.

Ao Departamento de Engenharia de Telecomunicações e Controle e a toda a Universidade de São Paulo, expresso minha profunda gratidão pela oportunidade de realizar o curso de mestrado. Também desejo agradecer a todos os professores com os quais tive a oportunidade de aprender ao longo desta jornada.

RESUMO

Este trabalho apresenta a aplicação de modelos de aprendizado de máquina baseados em árvores de decisão na avaliação dos momentos ideais para compra e venda de ativos no mercado de ações brasileiro. O aprendizado dos modelos é conduzido utilizando indicadores de mercado, calculados a partir da série histórica de preços. O trabalho apresenta uma aplicação prática, abordando o desafio do tratamento de variáveis não estacionárias, bem como a seleção das melhores variáveis para o modelo. Além de avaliar a capacidade de classificação dos melhores momentos de compra e venda, o estudo também inclui uma análise da aplicação dos modelos na geração de ordens de compra e venda, com a realização do *backtesting* no período de 2017 até 2023. Os resultados obtidos são comparados com a estratégia conhecida como *Moving Average Crossover* e uma estratégia baseada em ordens aleatórias, além de serem comparados com o *Buy and Hold*.

Palavras-Chave – Aprendizado de máquina, Sistema de Trading, Gradient Boosting, Random Forest.

ABSTRACT

This work presents the application of machine learning models based on decision trees in evaluating the optimal moments for buying and selling assets in the Brazilian stock market. The learning of these models is conducted using market indicators calculated from the historical price series. The study introduces a practical application, addressing the challenge of handling non-stationary variables, as well as the selection of the best variables for the model. In addition to assessing the classification capability of the best moments to buy and sell, the study also includes an analysis of the models' application in generating buy and sell orders, with backtesting conducted from 2017 to 2023. The obtained results are compared with the strategy known as Moving Average Crossover and a strategy based on random orders, in addition to being compared with Buy and Hold.

Keywords – Machine Learning, Trading System, Gradient Boosting, Random Forest.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

ADF – *Augmented Dickey-Fuller* (Teste de Dickey-Fuller Aumentado)
ATR – *Average True Range*
BB – *Bollinger Bands*
BBW – *Bollinger Bands Width*
BBP – *Bollinger Bands Percentage*
BBH – *Bollinger Channel Indicator Crossing High Band*
BBL – *Bollinger Channel Indicator Crossing Low Band*
BH – *Buy and hold*
BP – *Buying pressure*
CCI – *Commodity Channel Index*
CMF – *Chaikin Money Flow*
DCP – *Donchian Channel Percentage*
DCW – *Donchian Channel Width*
DLR– *Daily Log Return*
DPO – *Detrended Price Oscillator*
DR– *Daily Return*
EMA – *Exponential Moving Average*
EoM – *Ease of movement*
FI – *Force Index*
FN – Falso negativo
FP – Falso positivo
GPU – *Graphics Processing Unit* (Unidade de Processamento Gráfico)
KCW – *Keltner Channel Width*
KCP – *Keltner Channel Percentage*
KCH – *Keltner Channel Indicator Crossing High Band*
KCL – *Keltner Channel Indicator Crossing Low Band*
KST – *Know Sure Thing Oscillator*
LGBM – *Light Gradient Boosting Machine*
MAC – *Moving Average Crossover* (Cruzamento de Médias Móveis)
MACD - *Moving Average Convergence Divergence*
MAD - *Mean absolute deviation* Desvio Médio Absoluto
MAE – *Mean Absolute Error* (Erro Absoluto Médio)
MFI – *Money Flow Index*
MFR – *Money Flow Ratio*
MFV – *Money Flow Volume*
MDI – *Mean-Decrease Impurity*
MSE – *Mean Squared Error* (Erro Quadrático Médio)
PnL – *Profit and Loss*
PPO – *Percentage Price Oscillator*
PVO – *Percentage Volume Oscillator*
RF – *Random Forest*
ROC – *Rate of Change*

RSI – *Relative Strength Index* (Índice de Força Relativa)
SMA – *Simple Moving Average*
STC – *Schaff Trend Cycle*
TA – *Technical Analysis* (Análise Técnica)
TBM – *Triple Barrier Method*
TP – *Typical Price* (Preço Típico)
TR – *True Range*
TSI – *True strength index*
UI – *Ulcer Index*
UO – *Ultimate Oscillator*
VI – *Vortex Indicator*
VN – Verdadeiro negativo
VP – Verdadeiro positivo
VPT – *Volume-Price Trend*
XGB – *Extreme Gradient Boosting*

LISTA DE NOTAÇÕES E SÍMBOLOS

λ – Janela de tempo
β – Taxa de aprendizado
ϕ – Função sigmoide
θ_m – Parâmetro do nó m da árvore de decisão
$\mathcal{L}_n$ – Amostra de dados com tamanho n
B – Operador atraso
C_i – Preço de fechamento no instante i
$F_M(.)$ – Representação da função de aprendizado da combinação (*boosting*) de M árvores
$G(.)$ – Função de impureza
H_i – Preço máximo no instante i
$h_m(.)$ – Representação da função de aprendizado da árvore m
$I(.)$ – Função indicadora
L_i – Preço máximo no instante i
$Loss(.)$ – Função de perda
O_i – Preço de abertura no instante i
p_k – Proporção de dados com classe igual a k
r_i – Retorno no instante i
V_i – Volume de operações no instante i
$\mathbf{x_i}$ – Vetor de variáveis preditoras no instante i
y_i – Variável alvo no instante i
$\hat{y}_i$ – Previsão da variável alvo no instante i

SUMÁRIO

1 INTRODUÇÃO

Prever o movimento de um ativo no mercado de ações, algo de interesse comum entre os investidores, representa um grande desafio na área de finanças. De acordo com (PATEL et al., 2015) podemos separar a análise de um ativo no mercado de ações em dois tipos.

Um primeiro tipo de análise, conhecida como fundamentalista, explora a situação financeira, econômica e até mesmo o setor na qual a empresa se encontra. Um segundo tipo de análise, conhecida como análise técnica, examina dados do passado, como a variação do preço do ativo ao longo do tempo e o volume de transações.

Na análise técnica busca-se encontrar padrões nos movimentos que ocorreram no passado, visando prever futuros movimentos. Para isso, é comum aplicar métodos matemáticos na tentativa de detectar esses padrões e em seguida avaliar se esses padrões persistem no futuro.

Com o surgimento de alguns algoritmos de aprendizado de máquina e a capacidade destes algoritmos em detectar padrões em problemas com alto grau de complexidade, a aplicação de técnicas de aprendizado de máquina no mercado financeiro ocasionou o surgimento de diversos estudos na área.

Dentre os diversos estudos sobre aplicações de modelos de aprendizado de máquina em estratégias de *trading*, a abordagem com foco em um problema de classificação tem ganhado destaque. Prever o sinal de um resultado muitas vezes é mais importante do que prever o seu valor, e essa é uma razão para favorecer classificadores em vez de métodos de regressão (PRADO, 2020).

O presente trabalho tem o objetivo de avaliar a aplicação de modelos de aprendizado de máquina baseados em árvores de decisão no contexto do desenvolvimento de um sistema de *trading*. Especificamente, foram utilizados três modelos amplamente reconhecidos: o *Random Forest* (RF), o *LightGBM* (LGBM) e o *XGBoost* (XGB). A aplicação de aprendizado de máquina foi empregada na tarefa de classificação, com o propósito de identificar oportunidades de compra e venda no mercado financeiro.

O trabalho está dividido em 5 capítulos, com a inclusão deste capítulo de introdução. No Capítulo 2, apresenta-se uma revisão da literatura, começando com a descrição de algumas técnicas de rotulagem de dados que podem ser utilizadas para definir os momentos ideais para compra e venda e, assim, estruturar o problema de classificação.

Em seguida, são apresentadas possíveis variáveis preditoras, bem como o tratamento necessário nos casos em que estas são consideradas não estacionárias, como a aplicação da técnica de diferenciação fracionária.

Ainda no Capítulo 2, é apresentado o fundamento básico por trás dos modelos de aprendizado de máquina baseados em árvores de decisão, incluindo as técnicas de *bagging* e *boosting*, bem como exemplos de algoritmos baseados nessas técnicas.

Toda a metodologia aplicada no estudo é detalhada no Capítulo 3, onde são apresentados os detalhes da etapa de pré-processamento e treinamento do modelo. Neste capítulo, é apresentada a metodologia utilizada para selecionar as melhores variáveis e definir os hiperparâmetros do modelo. Por fim, o capítulo apresenta a transformação do resultado da aplicação do modelo de aprendizado de máquina na geração de ordens de compra e venda.

Para avaliar o resultado da classificação dos modelos e a aplicação desses resultados na geração de ordens de compra e venda, o Capítulo 4 apresenta os resultados obtidos com cinco importantes ativos listados na bolsa de valores do Brasil (BOVA11, PETR4, VALE3, ITUB4, BBDC4 e ABEV3). Os resultados foram comparados com o resultado de uma estratégia de *trading* conhecida como *Moving Average Crossover* (MAC). Além disso, foram realizadas comparações com uma estratégia criada com ordens aleatórias e a estratégia *Buy and Hold*. E por fim, o Capítulo 5 apresenta a conclusão final.

2 REVISÃO DA LITERATURA

Em um dos primeiros trabalhos a apresentar a aplicação de uma técnica de aprendizado de máquina na predição do movimento de um ativo no mercado de ações, (WHITE, 1988) estudou o uso de uma rede neural na extração de padrões de uma série temporal com os valores de retorno diário. Posteriormente, outros trabalhos apresentaram uma abordagem com uso de alguns indicadores técnicos, além do uso da série temporal com os valores do ativo.

O trabalho de (JANG et al., 1991) utilizou, ainda, 16 indicadores técnicos e também aplicou redes neurais para predição do movimento do mercado de ações da bolsa de valores de Taiwan. Nos anos posteriores, novas técnicas de aprendizado de máquina foram criadas ou aperfeiçoadas. Com isso, surgiram novos trabalhos como (HUANG et al., 2005), aplicando *Support Vector Machine* na predição do movimento do mercado de ações da bolsa de valores de Tóquio, (KUMAR; THENMOZHI, 2006) comparando os resultados entre *Support Vector Machine* e *Random Forest* na predição do movimento do mercado de ações da Índia.

Atualmente, há uma ampla gama de pesquisas relacionadas à aplicação de aprendizado de máquina no mercado financeiro. Alguns estudos empregam o aprendizado de máquina para resolver problemas de regressão, nos quais um modelo é utilizado para prever o preço de um ativo. Por outro lado, uma abordagem diferente, abordada neste trabalho, concentra-se na resolução de problemas de classificação, onde o modelo prevê o movimento do ativo.

Um exemplo disso é o estudo realizado por (NAIK; MOHAN, 2019) na Bolsa Nacional de Valores da Índia (NSE). Eles extraíram 33 indicadores técnicos de ações listadas na NSE e utilizaram algoritmos de aprendizado de máquina para construir modelos de classificação capazes de prever movimentos do mercado de ações.

Recentemente, (FRATTINI et al., 2022) apresentou um estudo que utilizou um dos modelos de aprendizado avaliado neste trabalho. O estudo de (FRATTINI et al., 2022)

aplicou o modelo LGBM na criação de um indicador de tendência. Outro estudo recente na área, realizado por (ANZAHAEI; NIKOOMARAM, 2022), apresentou uma comparação entre o modelo LGBM e o modelo *Catboost* na aplicação de estratégias de *trading*, no qual a estratégia que empregou o modelo LGBM obteve um retorno de 164%, em contraste com 142% alcançados utilizando o modelo *Catboost*.

Para estruturar um problema de classificação no contexto do aprendizado de máquina, é necessário definir a variável alvo que representa o resultado que o modelo de previsão deve alcançar. Isso é realizado por meio da etapa de rotulagem de dados, na qual os exemplos de treinamento são rotulados de acordo com as classes que se deseja prever. A correta definição da variável alvo e a rotulagem adequada dos dados são passos cruciais na construção de modelos de classificação eficazes, pois essas informações servirão como base para o treinamento e avaliação do modelo.

2.1 Rotulagem dos dados

Os trabalhos que visam prever o movimento do mercado com a utilização de uma técnica de classificação apresentam diferentes abordagens para a rotulagem de dados. Algumas dessas abordagens incluem a comparação de um retorno com um valor fixo ou a avaliação de máximos e mínimos da série histórica com o preço atual do ativo. Abaixo, são apresentadas algumas técnicas para a rotulagem dos dados.

2.1.1 Método de horizonte de tempo fixo

Como uma forma simples de classificação, este método se baseia na comparação do retorno de um ativo com um determinado limite. Considerando o preço de fechamento do ativo analisado no instante i como c_i, podemos calcular o retorno da seguinte forma:

$$r_{i+\lambda} = \frac{c_{i+\lambda}}{c_i} - 1 \tag{2.1}$$

sendo λ uma janela de tempo avaliada. Com isso, podemos criar a variável y_i de acordo com a equação (2.2).

$$y_i = \begin{cases} -1 & \text{se } r_{i+\lambda} < -\tau \\ 0 & \text{se } |r_{i+\lambda}| \leq \tau \\ 1 & \text{se } r_{i+\lambda} > \tau \end{cases} \tag{2.2}$$

Dessa forma, rotulamos cada ponto analisado em um instante i de acordo com o

retorno observado no instante $i + \lambda$, classificando i como um momento de compra, caso o retorno observado $r_{i+\lambda}$ seja maior que τ.

Como caso específico deste método, alguns trabalhos, como (TSAI et al., 2011), utilizam o parâmetro τ igual a zero, obtendo apenas duas classes, avaliando apenas se o retorno foi positivo ou negativo.

Este método possui uma limitação ao desconsiderar as oscilações que ocorrem no intervalo entre i e $i + \lambda$. Uma abordagem diferente, apresentada primeiramente por (PRADO, 2018), aborda o problema comentado.

2.1.2 Método das três barreiras

Diferentemente do método anterior, o método das três barreiras não avalia o retorno com base em uma janela de tempo fixa, mas sim o primeiro momento no qual o preço de um ativo ultrapassa uma determinada barreira. São definidas três barreiras, uma barreira superior, uma barreira inferior e uma terceira barreira opcional com base no tempo. A terceira barreira é utilizada quando o preço do ativo não ultrapassa as barreiras superior e inferior por um determinado intervalo de tempo definido.

Utilizando a série temporal com o preço de fechamento do ativo, analisamos o ponto i, definindo o valor de $y_i \in \{-1, 0, 1\}$, com base na criação de três barreiras: uma barreira superior, uma barreira inferior e uma terceira barreira vertical. Após a criação das três barreiras, temos que i representa um momento de compra ($y_i = 1$) caso o primeiro toque da série de fechamento ocorra na barreira superior. Se o primeiro toque ocorrer na barreira inferior, então i representa um momento de venda ($y_i = -1$). Caso o primeiro toque ocorra na barreira vertical, temos ($y_i = 0$).

Na Figura 1, ilustra-se o caso em que o ponto i é definido como um momento de compra.

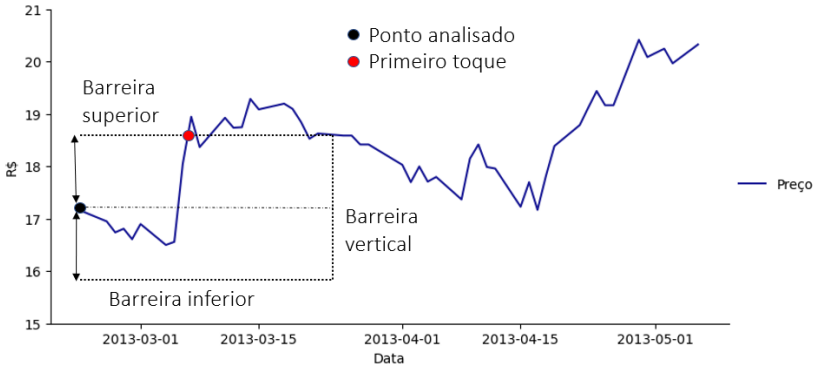


Figura 1: Representação de um ponto de compra com o método das três barreiras

Utilizando um caso particular do método das três barreiras, reduzindo o total de barreiras para dois, (PALAZZO et al., 2023) aplica o modelo de *Random Forest* para previsão do movimento no preço da criptomoeda Litecoin.

2.1.3 Mínimo e máximo

Uma outra técnica de rotulagem dos dados, semelhante à apresentada por (NASCIMENTO et al., 2020), define os momentos de compra, venda e espera com base nos valores de máximo e mínimo do preço do ativo em uma janela móvel. Dentro de cada janela, o algoritmo utilizado calcula o mínimo e o máximo dos valores da série temporal de fechamento. Para o ponto com o valor mínimo da janela é atribuído o valor 1 indicando que esse pode ser um ponto de entrada para comprar um ativo financeiro. Da mesma forma, para o ponto com o valor máximo da janela é atribuído o valor -1 indicando que esse pode ser um ponto de saída para vender um ativo financeiro. Caso contrário, o ponto é rotulado como 0 indicando um momento de espera.

2.1.4 Outros métodos de rotulagem

Foram apresentadas algumas das principais técnicas de rotulagem encontradas na literatura, no entanto, outras abordagens podem ser aplicadas para rotular a variável y_i. Por exemplo, a avaliação da série temporal pode ser realizada utilizando a média móvel, conforme descrito por (DANIEL, 2019). Além disso, a análise da tendência da série pode ser feita por meio da aplicação de um modelo de regressão para o período entre c_i e $c_{i+\lambda}$, como abordado por (PRADO, 2020).

Além dos diversos métodos de rotulagem disponíveis para definir a variável y_i, a

literatura apresenta uma variedade de aplicações de modelos de aprendizado de máquina que utilizam diferentes variáveis preditoras.

2.2 Variáveis preditoras

De acordo com (PRADO, 2018), ao desenvolver uma estratégia de investimento baseada em aprendizado de máquina, geralmente buscamos oportunidades quando ocorre uma convergência de fatores cujas previsões indicam um retorno favorável ajustado ao risco. Na literatura, vários estudos empregaram informações históricas de preços de ativos, bem como indicadores técnicos de mercado derivados desses dados, como entrada para modelos de aprendizado de máquina. Alguns trabalhos, como (WANG et al., 2018), utilizam diretamente os valores de abertura, máxima, mínima, fechamento e volume como variáveis preditoras de um modelo de aprendizado de máquina, enquanto outros, como (NELSON et al., 2017) e (NASCIMENTO et al., 2020), utilizam uma combinação de diversos indicadores técnicos de mercado que podem ser extraídos a partir dos dados históricos de preços. Mais recentemente, estudos como o de (WANG et al., 2023) combinaram indicadores de mercado com análise de sentimentos como variáveis preditoras, onde a análise de sentimentos é realizada através da avaliação de menções sobre o ativo analisado em mídias sociais.

2.2.1 Indicadores técnicos

Os indicadores técnicos são projetados com o objetivo simples de identificar tendências e mudanças nas tendências de preço, sem se aprofundar nas causas e efeitos subjacentes (COLBY, 2003). É por isso que os indicadores técnicos são uma das principais fontes de dados usadas como variáveis de entrada em modelos de aprendizado de máquina. Eles são calculados com base em séries históricas de preços e volumes de transações.

Existem quatro tipos principais de indicadores técnicos: indicadores de tendência, indicadores de momentum, indicadores de volatilidade e indicadores de volume (SALKAR et al., 2021).

1. Indicadores de Tendência:

   Os indicadores de tendência são projetados para identificar e confirmar a direção de uma tendência de preço. Um dos indicadores mais simples utilizado para representar a tendência no preço de um ativo é a média móvel, podendo ser aplicado a média

móvel simples ou a exponencial, que atribui um peso maior no cálculo da média para os valores mais recentes. A média móvel funciona como um filtro para o ruído causado pelas oscilações de curto prazo. Uma análise técnica comum com uso da média móvel é feita identificando pontos de cruzamento entre duas curvas de médias móveis.

2. Indicadores de Momentum:

   Indicam a taxa na qual ocorre uma mudança no movimento do preço de um ativo, como a velocidade na alteração do preço, representando a força de uma tendência. Um dos indicadores mais utilizados é conhecido como Relative Strength Index (RSI). Este indicador, primeiramente apresentado por (WILDER, 1978), é dado por:

$$RSI = 100 - (100/(1 + RS)) \tag{2.3}$$

   onde RS representa a razão entre a média dos movimentos de alta em relação a um determinado período e a média dos movimentos de baixa considerando o mesmo período.

3. Indicadores de Volume:

   Os indicadores de volume analisam o volume de negociação de um ativo financeiro, indicando a intensidade da atividade de compra e venda. Por exemplo, o *Money Flow Index* (MFI) avalia o volume e o preço para determinar a pressão de compra e venda em um ativo.

4. Indicadores de Volatilidade:

   Os indicadores de volatilidade medem a amplitude das flutuações de preços em um ativo financeiro. Eles ajudam a determinar a volatilidade esperada e os níveis de risco. Como exemplo desse tipo de indicador, temos as Bandas de Bollinger, que consistem em três partes principais. A primeira parte é a banda central, que é uma média móvel, geralmente a média móvel simples. Essa banda central atua como uma linha de referência e representa a tendência média de preços ao longo de um período específico. As duas outras partes das Bandas de Bollinger são as bandas exteriores, uma acima e outra abaixo da banda central. Essas bandas exteriores são desenhadas a uma certa distância da banda central e são calculadas com base na variabilidade dos preços passados.

Atualmente existem diversos indicadores conhecidos na literatura. A utilização de um modelo de aprendizado de máquina nos permite a realização de testes com diversos

indicadores, sendo possível encontrar posteriormente quais os indicadores que apresentam maior importância no aprendizado.

Em contraste com dados financeiros brutos, que frequentemente exibem tendências e padrões complexos, além de comportamento não estacionário, os indicadores técnicos são construídos com base em cálculos específicos que têm como objetivo capturar informações relevantes sobre os preços dos ativos. Essa estruturação pode resultar em novas séries temporais que podem ser consideradas estacionárias.

Como exemplo, (WANG; ZHENG, 2014) apresenta um estudo que demonstra que a maioria dos indicadores técnicos pode ser transformada em funções dos retornos logarítmicos. Uma vez que as funções de um processo fortemente estacionário ainda mantêm a estacionariedade, os indicadores técnicos derivados dos retornos logarítmicos também são estacionários. O estudo foi aplicado ao índice CSI 300, o qual é projetado para replicar o desempenho das 300 principais ações negociadas na Bolsa de Valores de Xangai e na Bolsa de Valores de Shenzhen.

Por simplificação, no contexto deste trabalho, o termo "série estacionária" se refere ao conceito de "série fracamente estacionária". Uma série temporal é fracamente estacionária se tanto a média quanto a covariância são invariantes no tempo (TSAY, 2005).

A estacionariedade torna-se uma propriedade importante na aplicação de modelos preditivos, uma vez que, em geral, os modelos de aprendizado de máquina se fundamentam na premissa de que o mecanismo de geração de dados não sofre alterações ao longo do tempo (SUGIYAMA; KAWANABE, 2012).

Ao estudar o impacto de variáveis não estacionárias em modelos de aprendizado de máquina, como o *Random Forest*, (DIXIT; JAIN, 2021) demonstra que os resultados tornam-se extremamente vulneráveis quando as variáveis preditoras utilizadas são não estacionárias.

Atualmente, existem diversos métodos de transformação desenvolvidos para tratar a não estacionariedade em séries temporais. (SALLES et al., 2019) apresenta uma lista de diferentes técnicas, dividindo-as em dois grupos principais. Um grupo de técnicas visa decompor a série não estacionária em diferentes componentes, como demonstrado no trabalho de (CHOWDHURY et al., 2019), que utilizou a técnica conhecida como *Empirical Mode Decomposition* juntamente com o modelo de aprendizado de máquina *Radom Forest* em uma aplicação usando séries temporais financeiras.

Um segundo grupo de técnicas de transformação, conforme apresentado por (SALLES

et al., 2019), baseia-se em um mapeamento. Eles geram uma nova série temporal por meio de um procedimento matemático. Como exemplo, (PRADO, 2018) apresenta a diferenciação fracionária como uma alternativa interessante no contexto da aplicação de modelos de aprendizado de máquina.

2.2.2 Diferenciação Fracionária

A noção de diferenciação fracionária na aplicação de predição de séries temporais foi apresentada por (HOSKING,), com o uso da técnica ARIMA. Para entender o processo de diferenciação fracionária na predição de séries temporais, podemos definir o operador B como um atraso. Aplicando o operador B, por exemplo, no preço de fechamento de uma ação $C_i \in \mathbb{R}$ em um dado instante i, obtemos um valor de C_{i-1} que representa o preço de fechamento da ação em um instante $i-1$, ou seja:

$$BC_i = C_{i-1} \tag{2.4}$$

Podemos utilizar o operador B para obter uma nova série a partir da diferenciação de C_i como:

$$C'_i = C_i - C_{i-1} = (1-B)C_i \tag{2.5}$$

Considerando $i = 1, ..., n$, a equação (2.5) representa a transformação da série com valores de preço, normalmente não estacionária, na série de retorno. Essa transformação pode levar à estacionariedade, mas a memória da série original é apagada (TSAY, 2005). O termo memória refere-se a quão fortemente os valores passados podem influenciar os valores futuros na série (ROBINSON, 2003). Como uma alternativa à diferenciação tradicional, a diferenciação fracionária busca transformar a série com um impacto menor, preservando informações importantes de longo prazo.

De forma geral, a diferenciação na ordem d da série pode ser obtida por

$$(1-B)^d C_i \tag{2.6}$$

que pode então ser resolvida com o auxílio da série binomial, com $d \in \mathbb{R}$, da seguinte forma (PRADO, 2018):

$$(1-B)^d = \sum_{z=0}^{\infty} \binom{d}{z} (-B)^z = 1 - dB + \frac{d(d-1)}{2!}B^2 - ... \tag{2.7}$$

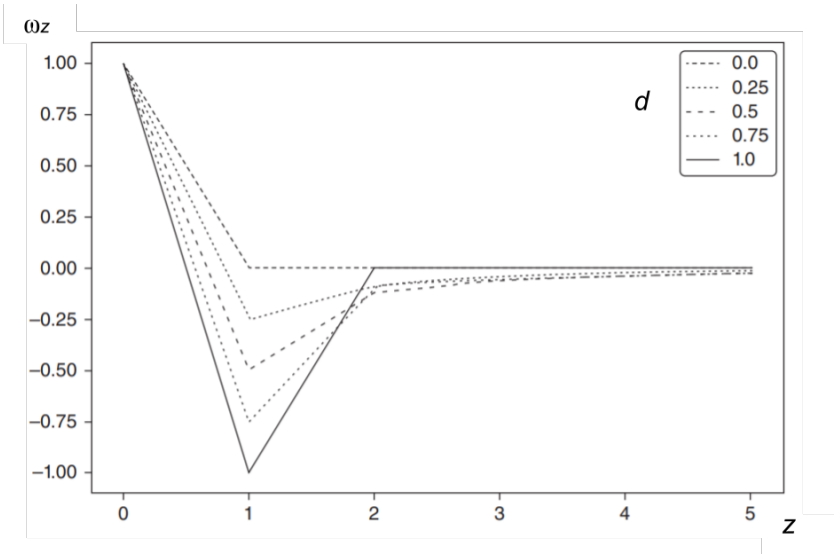
Definindo ω como:

$$\omega = \{1, -d, \frac{(d-1)d}{2!}, ..., (-1)^z \prod_{m=0}^{z-1} \frac{(d-m)}{z!}, ...\} \tag{2.8}$$

e sendo ω_z o z-ésimo termo de ω, a série diferenciada na ordem d pode então ser obtida como:

$$\tilde{C}_i = \sum_{z=0}^{\infty} \omega_z C_{i-z} \tag{2.9}$$

O somatório pode ser aproximado considerando os primeiros termos de ω e, para avaliar o impacto dessa aproximação, a Figura 2 de (PRADO, 2018) apresenta um gráfico comparando o valor do coeficiente ω_z com o aumento do termo z para diferentes valores de d.



Fonte: (PRADO, 2018).

Figura 2: Relação do coeficiente ω_z com o aumento do termo z.

Com o objetivo de preservar a memória da série, deseja-se encontrar o menor valor de d que torne a nova série obtida estacionária. Este processo pode ser realizado de modo iterativo, aumentando o valor de d gradativamente e avaliando a não estacionariedade da série. Para avaliar a estacionariedade da série, é possível utilizar um teste como o teste de Dickey-Fuller Aumentado (ADF) (DICKEY; FULLER, 1979).

Além da técnica apresentada, existem outras abordagens para esta etapa de pré-processamento. A finalidade fundamental deste processo é facilitar o aprendizado do modelo de aprendizado de máquina que será aplicado posteriormente.

2.3 Algoritmos de aprendizado de máquina

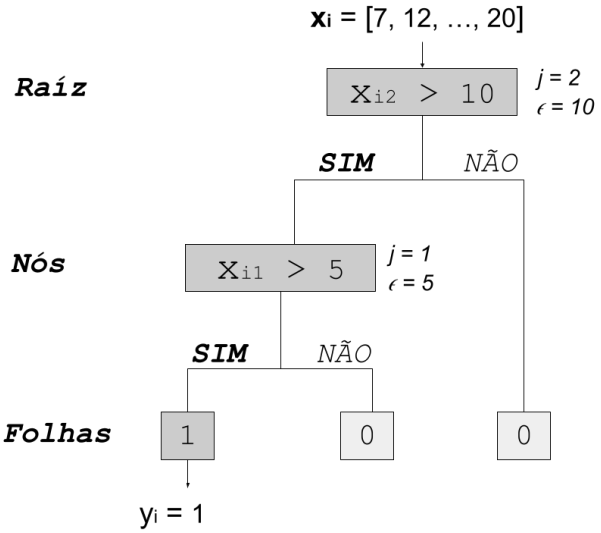
O aprendizado de máquina pode ser definido como "o campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados" (SIMON, 2013). Os algoritmos de aprendizado de máquina são capazes de capturar padrões com uso de dados e possuem aplicações em diversas áreas. Podemos separar estes algoritmos em três grupos, aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço. Neste trabalho será abordado apenas a categoria de aprendizado supervisionado, onde temos um conjunto de variáveis de entrada que se relacionam com uma variável de saída e buscamos encontrar uma regra que mapeie essa relação. Este trabalho se concentra na utilização de modelos que combinam, de diferentes formas, conjuntos de árvores de decisão.

2.3.1 Árvores de decisão

Árvores de decisão é um tipo de algoritmo de aprendizado de máquina bastante utilizado em problemas de classificação. Uma árvore de decisão, de forma resumida, visa mapear os valores de saídas a partir de decisões construídas com base nas variáveis de entrada. A Figura 3 apresenta um exemplo de uma estrutura de árvore de decisão sendo utilizada em um problema de classificação.

Mesmo sendo uma estrutura simples, o exemplo contempla todos os componentes presentes em uma árvore de decisão. Inicialmente temos uma primeira regra que é denominada raíz. As regras que são construídas subsequentemente são conhecidas como nós. Por último, temos a predição da variável de saída dada pelas folhas da árvore.

A Figura 3 ilustra uma árvore já construída, com os parâmetros j e ϵ já definidos em cada nó. O parâmetro j representa o índice da variável escolhida em cada nó enquanto ϵ representa um valor limite associado a variável no nó. Definido os parâmetros, é possível mapear um determinado vetor $\mathbf{x_i}$ formado por l variáveis, como por exemplo l indicadores de mercado, em uma variável de saída y_i, que pode representar a classificação de uma indicação de compra ou venda.



Fonte: Elaborada pelo autor.

Figura 3: Representação de uma árvore de decisão.

A construção de uma árvore de decisão é feita pela escolha da variável e da regra associada a variável, que melhor divide um conjunto de itens em cada passo (BREIMAN et al., 2017). Esse processo de construção ocorre durante a etapa de treinamento do modelo, onde os parâmetros da árvore são definidos.

Por simplificação, representamos os parâmetros j e ϵ do nó m como $\theta_m = (j_m, \epsilon_m)$, onde θ_m define a regra de divisão no nó m. Dada uma amostra de treinamento, a escolha de θ_m é avaliada com base na sua capacidade de separação dos dados no nó m, avaliada através de uma métrica definida como impureza. Representando todo o conjunto de treinamento como $\mathcal{L}_n = \{(\mathbf{x_1}, y_1), ..., (\mathbf{x_n}, y_n)\}$, com x_{ij} o elemento j do vetor $\mathbf{x_i}$, e lembrando que θ_1 representa os parâmetros j e ϵ do nó 1, podemos então realizar a primeira divisão dos dados em dois grupos com tamanhos n_e e n_d:

$$\mathcal{L}_{n_d}^{direita}(\theta_1) = \{(\mathbf{x_i}, y_i) | x_{ij_1} \leq \epsilon_1\} \tag{2.10}$$

$$\mathcal{L}_{n_e}^{esquerda}(\theta_1) = \{(\mathbf{x_i}, y_i) | x_{ij_1} > \epsilon_1\} \tag{2.11}$$

$$n = n_e + n_d \tag{2.12}$$

onde $\mathcal{L}_{n_d}^{direita}$ e $\mathcal{L}_{n_e}^{esquerda}(\theta)$ representam subconjuntos da amostra de treinamento total $\mathcal{L}_n$.

A impureza associada a divisão pode então ser calculada como:

$$G(\mathcal{L}_n, \theta_1) = \frac{n_d}{n} Loss(\mathcal{L}_{n_d}^{direita}(\theta_1)) + \frac{n_e}{n} Loss(\mathcal{L}_{n_e}^{esquerda}(\theta_1)) \tag{2.13}$$

onde $Loss$ representa uma função de perda.

A construção da árvore é realizada de maneira recursiva, primeiramente encontra-se o parâmetro θ_1 que apresenta a menor impureza para o nó 1 (2.14).

$$\theta_1^* = \text{argmin}_{\theta_1} G(\mathcal{L}_n, \theta_1) \tag{2.14}$$

Após a criação de cada nó, ocorre um processo de particionamento dos dados em dois grupos, e, com isso, o conjunto de treinamento usado no nó seguinte representa um subconjunto da amostra de treino. O mesmo procedimento realizado para o primeiro nó ($m = 1$) é repetido para cada nó subsequente.

Esse processo de particionamento dos dados e criação de novos nós é repetido até que cada folha da árvore seja pura, ou até que os critérios de parada sejam atingidos (LOH, 2011). Um dos critérios mais utilizados é a profundidade máxima da árvore, evitando que ela se torne muito complexa.

A função de perda desempenha um papel fundamental na construção e no treinamento de árvores de decisão, pois ela ajuda a avaliar a qualidade das divisões nos nós da árvore. A escolha da função de perda depende da tarefa que está sendo resolvida (classificação ou regressão).

Classificação

Uma das principais funções de perda utilizada para avaliar o erro da divisão em cada nó, em um problema de classificação, é conhecida como índice Gini (TIMOFEEV, 2004). Considerando que após a divisão da amostra de treinamento $\mathcal{L}_n$ de tamanho n, são geradas novas amostras $\mathcal{L}_{n_d}^{direita}$ e $\mathcal{L}_{n_e}^{esquerda}$ com novos tamanhos n_d e n_e, para generalizar a explicação do cálculo da função índice Gini podemos representar de forma geral uma nova amostra gerada como $\mathcal{L}_{n_f}$ de tamanho n_f.

$$Loss(\mathcal{L}_{n_f}) = \sum_k p_k(1 - p_k) \tag{2.15}$$

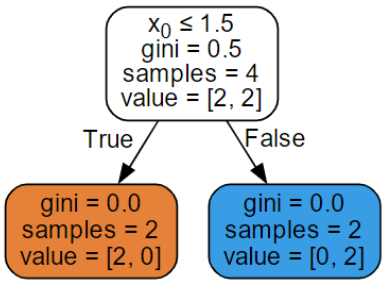
Em que p_k representa a proporção de dados com classe igual a k após a divisão no nó analisado. Sendo $I_i(k) \in \mathbb{I}_{n_f}$ o resultado da função indicadora aplicada a y_i presente no conjunto de dados $\mathcal{L}_{n_f}$, podemos definir p_k conforme as equações abaixo:

$$I_i(k) = \begin{cases} 1 & \text{se } y_i = k \\ 0 & \text{se } y_i \neq k \end{cases} \tag{2.16}$$

$$p_k = \frac{1}{n_f} \sum_{I \in \mathbb{I}_{n_f}} I_i(k) \tag{2.17}$$

Além do índice Gini, é importante destacar que existem outras funções de perda que podem ser úteis, como a entropia e o erro de classificação.

Para facilitar o entendimento, um exemplo numérico é apresentado abaixo. Por simplificação, $\mathbf{x_n} \in \mathbb{R}$ e apenas um nó foi considerado para árvore. Com isso, o treinamento consiste em obter o parâmetro ϵ que minimiza a equação (2.13). Considerando um conjunto de treino $\mathcal{L}_n = \{(0,0),(1,0),(2,1),(3,1)\}$, a Figura 4 apresenta a árvore após o treinamento com um valor $\epsilon = 1,5$.



Fonte: Elaborada pelo autor.

Figura 4: Representação da árvore de classificação após treinamento.

A divisão do conjunto foi feita com $\mathcal{L}_{n_e} = \{(0,0),(1,0)\}$ com tamanho $n_e = 2$ e $\mathcal{L}_{n_d} = \{(2,1),(3,1)\}$ com tamanho $n_d = 2$. Nesse exemplo temos um caso de pureza na divisão e com isso $Loss(\mathcal{L}_{n_e}) = 0$ e $Loss(\mathcal{L}_{n_d}) = 0$.

De modo semelhante, podemos definir uma função de perda para um problema de regressão.

Regressão

Embora o problema abordado neste trabalho seja uma tarefa de classificação, é importante compreender a aplicação das árvores de decisão em problemas de regressão. Algoritmos como o *Gradient Boosting Classifier*, que serão detalhados mais adiante e são utilizados em problemas de classificação, têm como base um conjunto de árvores de regressão.

As árvores de regressão são usadas para variáveis dependentes y_i que assumem valores contínuos ou discretos ordenados, sendo o erro de previsão geralmente medido pela

diferença ao quadrado entre os valores observados e os valores previstos, definido como erro quadrático médio (MSE) (LOH, 2011).

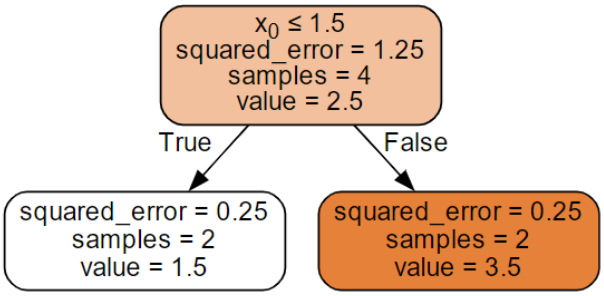
Considerando novamente uma amostra de dados $\mathcal{L}_{n_f}$ de tamanho n_f, definindo Y_f como o conjunto de variáveis dependentes y_i presentes na amostra de treino $\mathcal{L}_{n_f}$, podemos calcular a média da variável alvo nesse conjunto de dados da seguinte forma.

$$\bar{y}_{m_f} = \frac{1}{n_f} \sum_{y_i \in Y_f} y_i \tag{2.18}$$

Dessa forma, podemos definir a função de perda como:

$$Loss(\mathcal{L}_{n_f}) = \frac{1}{n_f} \sum_{y_i \in Y_f} (y_i - \bar{y}_{m_f})^2 \tag{2.19}$$

Da mesma forma que apresentado na árvore de classificação, um exemplo numérico é apresentado abaixo. Para o treino da árvore foi considerado o conjunto $\mathcal{L}_n = \{(0,1),(1,2),(2,3),(3,4)\}$. A Figura 5 apresenta a árvore após o treinamento com um valor $\epsilon = 1,5$.

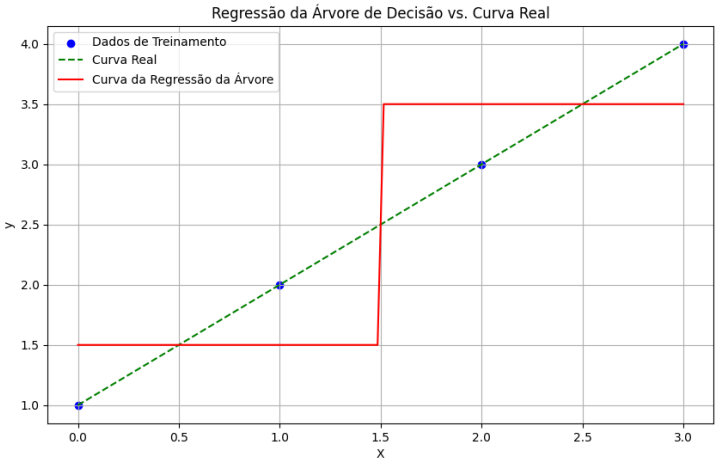


Fonte: Elaborada pelo autor.

Figura 5: Representação da árvore de regressão após treinamento.

Temos então a divisão do conjunto na forma $\mathcal{L}_{n_e} = \{(0,1),(1,2)\}$ com tamanho $n_e = 2$ e $\mathcal{L}_{n_d} = \{(2,3),(3,4)\}$ com tamanho $n_d = 2$. E com isso $\bar{y}_{m_e} = 1,5$ e $\bar{y}_{m_d} = 3,5$. Para a função de perda temos $Loss(\mathcal{L}_{n_e}) = 0,25$ e $Loss(\mathcal{L}_{n_d}) = 0,25$.

A Figura 6 ilustra o resultado da regressão obtida. Em uma aplicação prática, as árvores de decisão costumam ter mais de um nó para melhorar a qualidade das previsões.



Fonte: Elaborada pelo autor.

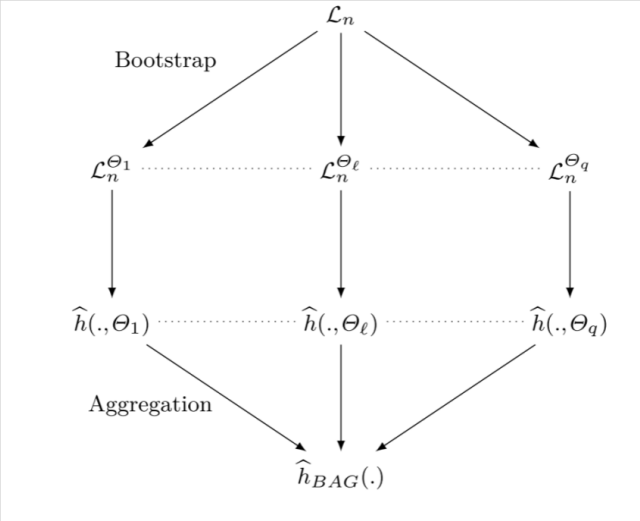
Figura 6: Representação do resultado da regressão.

Como uma forma de melhorar o poder preditivo de uma árvore de decisão, é possível utilizar técnicas que combinam conjuntos de modelos. Duas técnicas amplamente utilizadas são o *bagging* e o *boosting*. Ambas as abordagens aproveitam conjuntos de modelos para aprimorar a capacidade de generalização e a precisão das previsões. Vale destacar que essas técnicas não se limitam apenas ao uso de árvores de decisão como componentes do conjunto.

2.3.2 Bagging

O conceito de *Bagging*, que significa *Bootstrap Aggregating*, é uma técnica que visa melhorar o desempenho dos modelos de predição, criando múltiplas instâncias (*Bootstrap*) de um modelo base e, em seguida, combinando suas previsões (*Aggregating*) para produzir uma previsão final mais precisa e robusta (BREIMAN, 1996).

A figura 7 ilustra o processo de *Bagging*, onde a partir da amostra de treinamento $\mathcal{L}_n$ novas amostras são geradas. Cada modelo base $\hat{h}(.)$ é construído a partir de um novo conjunto de dados gerado. Considerando que cada modelo base criado $\hat{h}(.)$ representa uma árvore, ao final do processo, temos a criação de q árvores de decisão, cada uma com um peso unitário na classificação final (BREIMAN, 2001).



Fonte: (GENUER; POGGI, 2020).

Figura 7: Representação do processo de *Bagging*.

Uma implementação amplamente conhecida do *Bagging* que utiliza árvores de decisão como modelos base é o *Random Forest*. Este modelo, além de adotar a técnica de *Bagging*, incorpora um ajuste adicional durante a criação de cada árvore. Durante a construção de uma árvore $\hat{h}(.)$, sempre que uma divisão na árvore é feita e um novo nó é criado, apenas um subconjunto do conjunto total de variáveis preditoras é avaliado. Em outras palavras, em cada divisão na árvore, o algoritmo *Random Forest* não considera a maioria das variáveis preditoras disponíveis (JAMES et al., 2013). Isso introduz aleatoriedade e diversidade no processo de construção das árvores, o que ajuda a reduzir a correlação entre elas.

A aleatoriedade no *Random Forest* ajuda a garantir que as árvores se concentrem em diferentes aspectos dos dados, capturando padrões que uma única árvore não seria capaz. Outra técnica que pode aprimorar algoritmos preditivos e que pode ser aplicada a modelos de árvores de decisão é conhecida como *Gradient Boosting*, conforme apresentado em (FRIEDMAN, 2001).

2.3.3 Boosting

O processo de *boosting* é uma técnica de aprendizado de máquina em que vários modelos mais simples, como o modelo de árvore de decisão, são utilizados para formar um modelo mais robusto. Esses modelos mais simples são construídos sequencialmente, onde cada modelo subsequente tenta corrigir os erros cometidos pelos modelos anteriores.

Como uma aplicação da técnica de *boosting*, o *Gradient Boosting* (FRIEDMAN, 2002) é uma técnica avançada de aprendizado de máquina que tem se mostrado muito eficaz em muitos problemas de regressão e classificação. Essa técnica utiliza o gradiente descendente para ajustar o modelo iterativamente, minimizando o erro de previsão

De modo geral, a partir de um conjunto de treino $\mathcal{L}_n$, o algoritmo de *Boosting* busca encontrar uma função $F_M(\mathbf{x_i})$, que pode ser representada como uma soma de M funções de aprendizado $h_m(\mathbf{x_i})$, onde cada função de aprendizado significa uma árvore de decisão de regressão.

$$F_M(\mathbf{x_i}) = \sum_{m=1}^{M} \beta_m h_m(\mathbf{x_i}) \tag{2.20}$$

O parâmetro β_m, onde $0 < \beta_m \leq 1$, controla a taxa de aprendizado e é utilizado na regularização para ajustar a contribuição de cada árvore (FRIEDMAN, 2001). Em um problema de regressão o resultado da função $F_M(\mathbf{x_i})$ representa a variável de resposta $\hat{y}_i$, enquanto em um problema de classificação binária ($y_i \in \{0, 1\}$) o valor da função é então mapeado na probabilidade de y_i pertencer a classe dado o valor de $\mathbf{x_i}$

$$p(y_i = 1|\mathbf{x_i}) = \phi(F_M(\mathbf{x_i})) \tag{2.21}$$

onde ϕ representa uma função sigmoide (ou logística). Para ilustrar o mapeamento da probabilidade na classe prevista $\hat{y}_i$, a equação (2.22) apresenta o mapeamento em um caso de classificação binária.

$$\hat{y}_i = \begin{cases} 1 & \text{se } p(y_i = 1|\mathbf{x_i}) \geq 0,5 \\ 0 & \text{se } p(y_i = 1|\mathbf{x_i}) < 0,5 \end{cases} \tag{2.22}$$

Em um problema de classificação com várias classes ($y_i \in \{0, 1, ..., K\}$), ocorre uma redução do problema, transformado cada classe k em uma classificação binária, avaliando $p(y_i = k|\mathbf{x_i})$.

Definida uma função de perda diferenciável $Loss$, a criação do conjunto de árvores ocorre de modo iterativo, na iteração m a árvore h_m busca minimizar a perda $Loss_m$ dada pelo conjunto de árvores que já foram construídos no passo anterior $F_{m-1}(\mathbf{x_i})$ conforme a equação abaixo:

$$h_m = \arg\min_h Loss_m = \arg\min_h \sum_{i=1}^{n} Loss(y_i, F_{m-1}(\mathbf{x_i}) + h(\mathbf{x_i})), \tag{2.23}$$

em um problema de classificação é comum o uso da função de perda *logloss* definida como:

$$Loss(y_i, \hat{y}_i) = -(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)) \tag{2.24}$$

Atualmente, existem diversos modelos de aprendizado de máquina fundamentados na técnica de *Gradient Boosting*. Esses algoritmos são conhecidos por sua eficiência computacional e capacidade de lidar com grandes conjuntos de dados, conforme discutido a seguir:

1. XGBoost (*Extreme Gradient Boosting*):

   Uma das principais contribuições do XGBoost para o campo de aprendizado de máquina é a incorporação de técnicas de regularização para mitigar problemas de *overfitting* (CHEN; GUESTRIN, 2016). Isso o torna uma ferramenta robusta e confiável para construir modelos mais generalizados.

2. LightGBM (*Light Gradient Boosting Machine*):

   O algoritmo LGBM (KE et al., 2017) foi projetado para ser distribuído e eficiente, apresentando vantagens como treinamento mais rápido, maior eficiência e menor uso de memória. Além disso, o LGBM oferece melhor precisão e suporte para aprendizado paralelo, distribuído e em GPU. Ele é capaz de lidar com dados em grande escala, tornando-se uma ferramenta atraente para a resolução de problemas de aprendizado de máquina em larga escala.

3. CatBoost (*Categorical Boosting*):

   O CatBoost é projetado para lidar bem com variáveis categóricas em problemas de classificação e regressão (PROKHORENKOVA et al., 2018). Ele inclui um mecanismo embutido para codificar automaticamente essas variáveis, tornando o processo de preparação de dados mais eficiente.

4. NGBoost (*Natural Gradient Boosting*)

   O NGBoost é particularmente útil para previsões probabilísticas e é capaz de estimar a incerteza nas previsões, o que o torna relevante para uma variedade de aplicações (DUAN et al., 2020).

Esses modelos baseados na técnica de *Boosting* representam avanços significativos na construção de modelos de aprendizado de máquina de alto desempenho.

3 METODOLOGIA

Um sistema de *trading* automático baseado em aprendizado de máquina é um sistema computacional que utiliza algoritmos de aprendizado de máquina para analisar dados de mercado, identificar padrões, tomar decisões de compra e venda de ativos financeiros e executar essas ordens de forma autônoma, sem intervenção humana direta.

Neste estudo, exploramos a aplicação do aprendizado de máquina para resolver um problema de classificação. Inicialmente, identificamos os momentos ideais de compra e venda ao rotular os dados. Em seguida, treinamos vários modelos de aprendizado de máquina com o objetivo de classificar esses momentos de compra e venda. Para realizar essa classificação, utilizamos variáveis preditoras que incluem os preços de mercado históricos e os indicadores de mercado derivados desses dados.

Como etapa inicial na construção de um sistema de *trading* baseado em aprendizado de máquina, realizou-se a coleta de dados, que inclui informações como os históricos de preços e os volumes de negociação. Após a aquisição desses dados, procedeu-se à sua preparação, que incluiu o cálculo de indicadores técnicos.

3.1 Extração e pré-processamento dos dados

Desde a extração dos dados até a utilização do modelo foi utilizada a linguagem de programação Python. Com a utilização de pacotes como *yfinance* foi possível a coleta de dados históricos de mercado dos ativos analisados. Foram escolhidos alguns dos mais importantes ativos listados na bolsa de valores do Brasil, sendo eles: PETR4, Petróleo Brasileiro SA Petrobras Preference Shares; VALE3, Vale S.A.; ITUB4, Itaú Unibanco; BBDC4, Banco Bradesco SA Preference Shares; ABEV3, Ambev; BOVA11, iShares Ibovespa Fundo de Índice. Os dados obtidos possuem valores diários de abertura O_i, fechamento C_i, alta H_i, baixa L_i e volume V_i, onde i representa o tempo, especificamente o dia.

3.1.1 Rotulagem dos dados

Com os valores diários de C_i, utilizando informações sobre instantes de tempo posteriores a C_i, foi possível definir os momentos ideias de compra e venda, com base em uma regra estabelecida, para o conjunto de treino. Existem diversas técnicas de rotulagem que podem ser aplicadas.

Neste trabalho foi utilizada a técnica de rotulagem conhecida como Mínimo e Máximo, apresentada anteriormente. O algoritmo utilizado para rotulagem está apresentando no Anexo A. A expressão para obtenção de $y_i \in \{-1, 0, 1\}$ pode ser definida conforme 3.1

$$y_i = \begin{cases} -1 & \text{se } C_i \text{ representa o máximo de } C_{i-\lambda} \text{ até } C_{i+\lambda} \\ 1 & \text{se } C_i \text{ representa o mínimo de } C_{i-\lambda} \text{ até } C_{i+\lambda} \\ 0 & \text{caso contrário} \end{cases} \tag{3.1}$$

Para a escolha do parâmetro λ, é importante levar em consideração que, embora o desempenho do modelo possa melhorar com o aumento de λ, conforme mencionado em (HAN et al., 2023), o tamanho dos dados disponíveis para o aprendizado diminui, e o desbalanceamento entre as classes -1, 0 e 1 aumenta. Neste trabalho, foi adotado um valor de λ igual a 7 para todas as análises. Após a rotulagem dos dados, as variáveis preditoras são então definidas.

3.1.2 Cálculo dos indicadores técnicos

Com o auxílio da biblioteca em Python *Technical Analysis* (TA), foi possível obter diretamente diversos indicadores técnicos a partir dos dados extraídos, como as Bandas de Bollinger apresentadas na Figura 8.

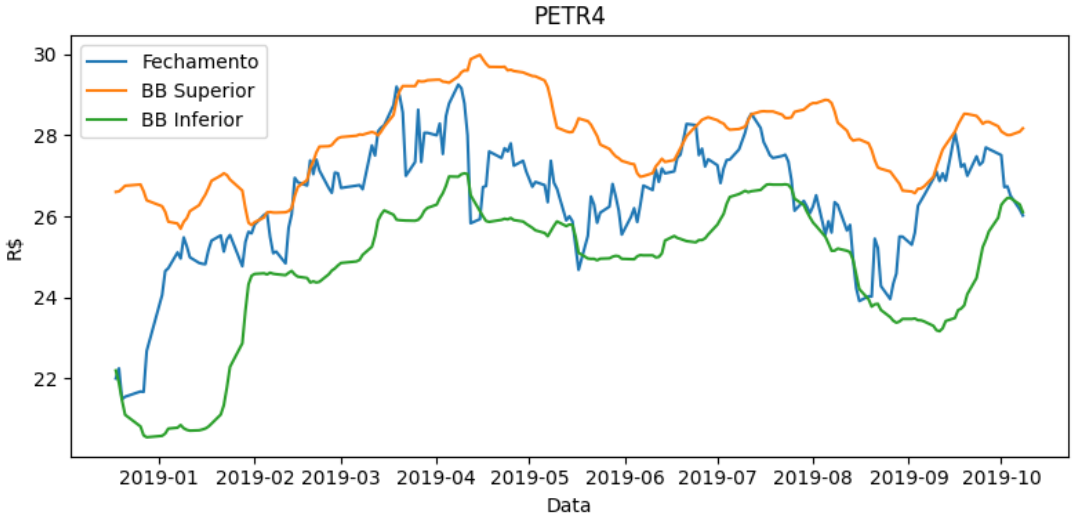


Figura 8: Bandas de Bollinger.

As Bandas de Bollinger mostram a volatilidade do mercado e ajudam a identificar possíveis pontos de virada em uma tendência. Como um primeiro critério na seleção dos indicadores a serem utilizados, entre indicadores semelhantes, escolheu-se aqueles que demonstravam comportamento estacionário com base no teste ADF. Portanto, em vez de utilizar diretamente as Bandas de Bollinger superior BB_{u_i} e inferior BB_{l_i}, como mostrado na Figura 8, por exemplo, optou-se por usar o indicador $BB\%$, calculado conforme a equação (3.2).

$$BB_i\% = \frac{C_i - BB_{l_i}}{BB_{u_i} - BB_{l_i}} \tag{3.2}$$

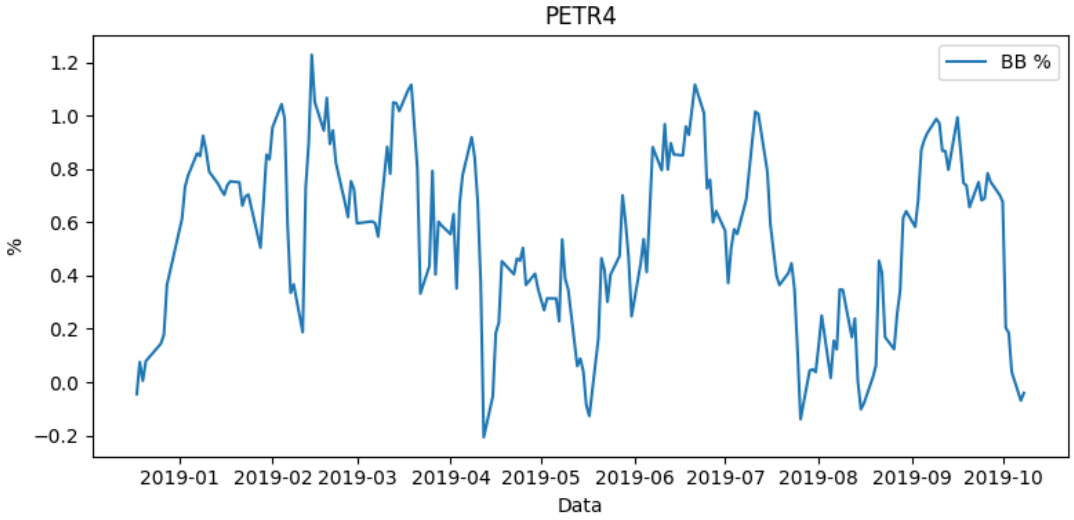


Figura 9: Bandas de Bollinger %.

Após essa seleção, foram obtidos um total de 63 indicadores técnicos de mercado; a lista desses indicadores e os parâmetros utilizados para obtê-los são apresentados no Anexo B.

3.1.3 Diferenciação fracionária

Como uma abordagem para lidar com o problema da não estacionariedade, especialmente nas séries temporais de abertura O_i, fechamento C_i, alta H_i, e baixa L_i, a técnica de diferenciação fracionária foi aplicada. A diferenciação foi realizada usando a equação (2.9), com o somatório truncado até o décimo termo, e a escolha de d foi baseada no teste ADF. A Tabela 1 apresenta um exemplo do resultado obtido para a série de fechamento C_i do ativo PETR4.

d	ADF	5%
0,00	-1,72	-2,86
0,11	-2,26	-2,86
0,22	-3,08	-2,86
0,33	-4,36	-2,86
0,44	-6,35	-2,86
0,56	-9,50	-2,86
0,67	-14,40	-2,86
0,78	-21,10	-2,86
0,89	-27,83	-2,86
1,00	-32,35	-2,86

Tabela 1: Resultado do teste ADF (Augmented Dickey-Fuller)

O resultado do teste é apresentado na coluna ADF, para rejeitar a hipótese nula e concluir que a série é estacionária precisamos utilizar o valor de d tal que o valor ADF seja menor que $2,86$, considerando um nível de significância no teste de 5%. O valor de d é definido como menor valor tal que o teste de estacionariedade seja válido. A Figura 10 representa a transformação da série C_i em $\tilde{C}_i$ para o ativo PETR4, utilizando $d = 0,22$.

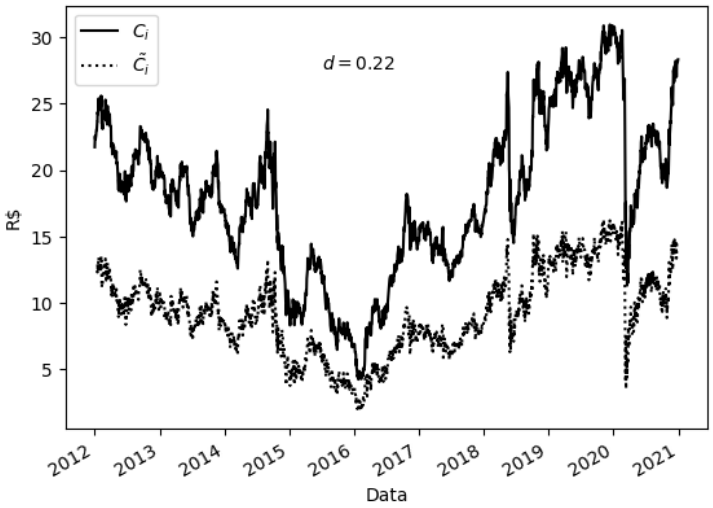


Figura 10: Diferenciação da série de fechamento (PETR4).

A determinação do valor de d está diretamente relacionada ao intervalo da série que está sendo avaliado. Em uma simulação, o intervalo considerado para a escolha do melhor valor de d corresponde ao conjunto de treinamento.

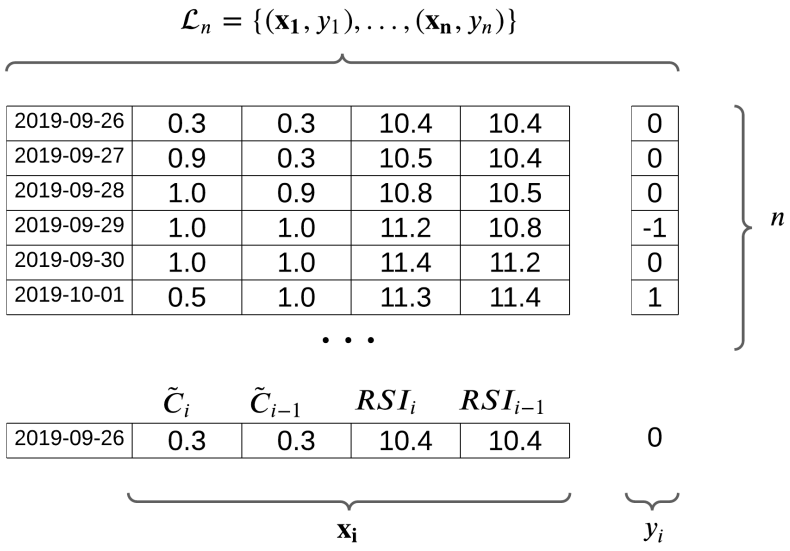
3.1.4 Variáveis preditoras

Para facilitar a identificação de padrões e tendências ao longo do tempo, podemos definir o vetor $\mathbf{x_i} \in \mathbb{R}^l$, que representa todas as variáveis preditoras no instante i, como sendo composto pelas variáveis mencionadas anteriormente, calculadas entre os instantes de tempo i e $i - \eta$, onde η foi definido como 4. Isso implica que as variáveis utilizadas na predição do instante i utilizam indicadores calculados no período compreendido entre o instante i e os quatro instantes de tempo anteriores. A relação é expressa pela equação(3.3).

$$\mathbf{x_i} = \begin{bmatrix} Indicador1_i \\ \vdots \\ Indicador1_{i-\eta} \\ \vdots \\ \tilde{C}_i \\ \vdots \\ \tilde{C}_{i-\eta} \end{bmatrix} \tag{3.3}$$

Como exemplo ilustrativo, podemos considerar o indicador Momentum RSI e o valor da série diferenciada $\tilde{C}_i$. Por simplificação, podemos considerar η igual a 1 para construção

das variáveis de entrada do modelo exemplificado. Na figura 11, temos a representação numérica do vetor $\mathbf{x_i}$ junto com y_i, formando o conjunto de treino.



Fonte: Elaborada pelo autor.

Figura 11: Ilustração das variáveis utilizadas no treinamento.

Com o conjunto de variáveis apresentadas até agora, realizou-se a seleção das melhores variáveis para aplicação no modelo. Esta etapa, conhecida como seleção de variáveis ou *feature selection*, desempenha um papel importante, pois algumas variáveis podem conter informações muito semelhantes ou até redundantes, o que pode não acrescentar valor significativo ao resultado final do modelo. A seleção de variáveis será apresentada com mais detalhe adiante no texto, abordando os critérios e métodos utilizados para escolher as variáveis mais relevantes para o modelo.

3.2 Treinamento do modelo

O processo de treinamento foi realizado com o uso de uma técnica de janela deslizante, na qual os dados foram divididos em um conjunto de treinamento/validação e um conjunto de dados utilizados para simular a aplicação do modelo treinado, conforme ilustrado na Figura 12. Para cada ano na simulação, foram utilizados os 6 anos anteriores para o treinamento e validação do modelo de aprendizado de máquina. A simulação abrangeu os anos de 2011 até 2023.

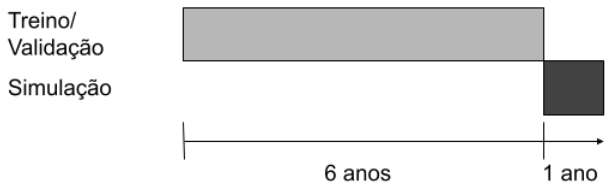


Figura 12: Representação da divisão dos dados.

Para otimizar os resultados obtidos na classificação, os conjuntos de treinamento e validação foram empregados tanto na seleção das variáveis mais relevantes quanto na otimização dos hiperparâmetros do modelo de aprendizado.

3.2.1 Seleção de variáveis

Essa etapa envolve a escolha das variáveis que serão utilizadas para treinar o modelo, o que contribui para a redução de sua complexidade. Existem várias técnicas disponíveis para realizar essa avaliação, uma delas é conhecida como *Mean-Decrease Impurity* (MDI). O MDI é uma técnica comumente usada em árvores de decisão e algoritmos baseados em árvores.

A ideia principal por trás do MDI é medir o quanto cada variável preditora contribui para a redução da impureza nos nós da árvore durante o processo de divisão. Considerando a equação da impureza para um nó 1 conforme a equação (2.13), podemos calcular a variação entre a impureza antes da divisão e após utilizando a equação (3.4).

$$\Delta G(\mathcal{L}_n, \theta_1) = Loss(\mathcal{L}_n(\theta_1)) - (\frac{n_d}{n} Loss(\mathcal{L}_{n_d}^{direita}(\theta_1)) + \frac{n_e}{n} Loss(\mathcal{L}_{n_e}^{esquerda}(\theta_1))) \quad (3.4)$$

Quanto maior for a diminuição na impureza, mais importante é a variável. A importância de uma variável pode ser calculada como a média da variação na impureza, considerando todos os nós nos quais a variável foi selecionada. Com o uso da métrica MDI, o número de variáveis foi reduzido, selecionando apenas alguns dos indicadores técnicos de mercado disponíveis inicialmente. A Figura 13 apresenta um exemplo de avaliação da métrica MDI para o ativo PETR4, onde são exibidos apenas os 10 indicadores com os melhores resultados.

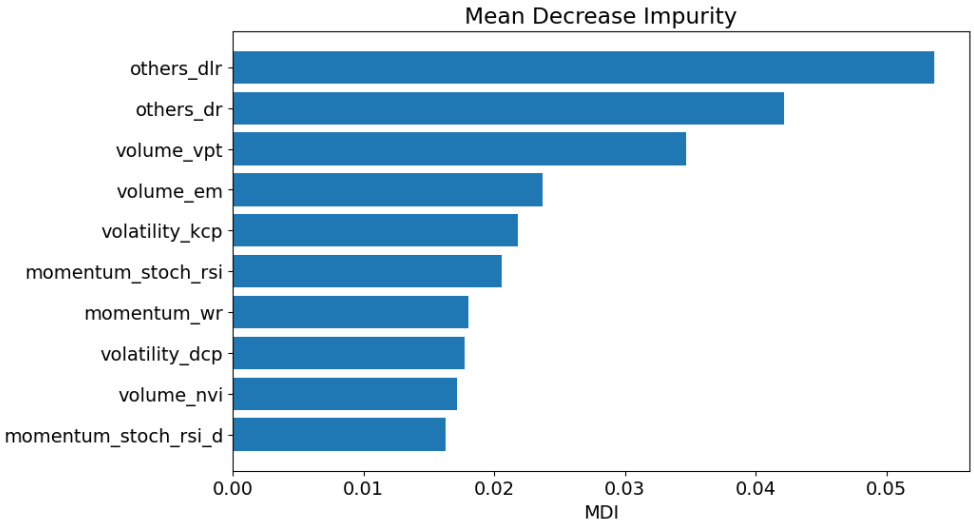


Figura 13: Representação da avaliação da métrica MDI.

Todos os indicadores apresentados na Figura 13 estão detalhados no Anexo B. A partir do gráfico, destacam-se os quatro principais indicadores que demonstraram maior contribuição em relação à diminuição da impureza no modelo, em um experimento com o ativo PETR4.

1. *Daily Log Return*:

   O retorno logarítmico diário é uma medida que expressa a variação percentual no preço de um ativo ao longo de um dia, normalmente utilizado para avaliar a volatilidade e o desempenho histórico.

2. *Daily Return*:

   O retorno diário, similar ao retorno logarítmico diário, é uma medida que indica a variação percentual no preço de um ativo em um único dia de negociação. Ambos os retornos são fundamentais para entender a performance do ativo no curto prazo.

3. *Volume-Price Trend* (VPT):

   O VPT é um indicador que combina o volume de negociação com a mudança no preço do ativo. Ele busca identificar a força por trás dos movimentos de preços, considerando o volume de negociações. Um aumento no VPT pode indicar uma tendência mais forte.

4. *Ease of Movement*:

Este indicador avalia a facilidade com que o preço de um ativo se move, levando em consideração o volume de negociações. Ele é especialmente útil para identificar se os movimentos de preços são apoiados por um volume significativo, indicando maior convicção no mercado.

3.2.2 Hiperparâmetros do modelo de classificação

Hiperparâmetros são configurações ajustáveis que não são aprendidos diretamente pelo modelo, mas desempenham um papel crucial em como o modelo é treinado e como faz previsões. O objetivo da otimização de hiperparâmetros é encontrar a combinação ideal desses parâmetros para maximizar o desempenho do modelo.

Dois dos hiperparâmetros mais comuns em modelos baseados em árvores são o número de estimadores, que determina quantas árvores compõem o modelo, e a profundidade máxima da árvore, que controla o nível de complexidade das árvores individuais. O número de estimadores afeta a capacidade do modelo de generalizar e sua estabilidade, enquanto a profundidade máxima é essencial para regular a complexidade das árvores. A escolha adequada desses hiperparâmetros é fundamental para equilibrar o desempenho e a capacidade de generalização do modelo.

Existem diversas abordagens para encontrar os melhores hiperparâmetros. Como exemplo de técnicas utilizadas, temos uma técnica conhecida como *Grid Search*, que varre sistematicamente um conjunto pré-definido de valores para cada hiperparâmetro. Isso é útil quando se tem uma ideia clara do espaço de busca, no entanto, a aplicação dessa técnica pode demandar um alto custo computacional. A cada vez que diferentes hiperparâmetros são avaliados, é necessário treinar o modelo, fazer previsões e, em seguida, calcular a métrica usada para avaliar o desempenho do modelo.

Uma abordagem alternativa, conhecida como *Bayesian Optimization*, oferece uma estratégia mais eficiente e inteligente para encontrar os melhores hiperparâmetros (DEWANCKER et al., 2016). Nessa abordagem, um modelo probabilístico é construído para estimar a relação entre os hiperparâmetros e a métrica de desempenho do modelo de aprendizado de máquina.

Neste trabalho, foi utilizada a biblioteca em Python *scikit-optimize* para a busca de melhores hiperparâmetros com a técnica de *Bayesian Optimization*. A Figura 14 apresenta um exemplo de busca para o hiperparâmetro máxima profundidade, com o uso do modelo *Random Forest* aplicado aos dados do ativo PETR4.

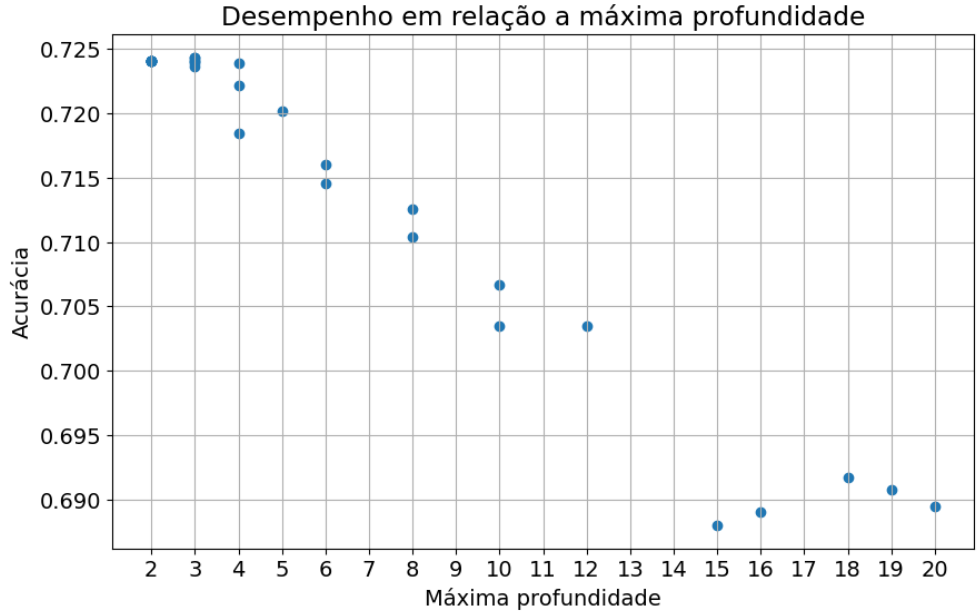


Figura 14: Acurácia em relação a máxima profundidade.

Na busca pelos melhores hiperparâmetros, a métrica de acurácia foi utilizada para avaliar o desempenho do modelo. A seguir, serão apresentadas esta e outras métricas comuns para avaliação do desempenho de modelos de classificação.

3.2.3 Avaliação do modelo de aprendizado

Para o resultado da classificação de cada classe k, podemos definir VP como o total de casos em que o modelo previu corretamente a classe k; VN como o total de casos em que o modelo previu corretamente a ausência da classe k; FP como o total de casos em que o modelo previu incorretamente a classe k; FN como o total de casos em que o modelo previu incorretamente a ausência da classe k.

Com base nessa definição, estão listadas abaixo algumas das principais métricas usadas para avaliação do modelo de classificação.

1. Acurácia: mede a proporção de previsões corretas em relação ao total de previsões:

$$\text{Acurácia} = \frac{(VP) + (VN)}{Total} \tag{3.5}$$

Em um problema com K classes, podemos calcular a acurácia como:

$$\text{Acurácia} = \frac{\sum_{i=1}^{K} VP_k}{Total} \tag{3.6}$$

Nesta fórmula, VP_k representa o número de verdadeiros positivos na classe k.

2. Precisão: avalia a proporção de previsões positivas corretas em relação ao total de previsões positivas feitas pelo modelo:

$$\text{Precisão} = \frac{(VP)}{(VP) + (FP)} \tag{3.7}$$

3. Revocação: mede a proporção de previsões positivas corretas em relação ao total de exemplos verdadeiramente positivos:

$$\text{Revocação} = \frac{(VP)}{(VP) + (FN)} \tag{3.8}$$

4. F1-Score: média harmônica entre precisão e revocação:

$$\text{F1-Score} = \frac{2 \times \text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}} \tag{3.9}$$

5. Média Ponderada:

   Para agrupar as métricas de Precisão, Revocação e F1-Score de todas as classes em um problema com múltiplas classes, é possível utilizar a média ponderada. No exemplo da Revocação, em um problema com K classes, é possível usar a fórmula a seguir, levando em consideração o total ponderado de cada classe:

$$\text{Revocação(Média)} = \frac{\sum_{k=1}^{K} \text{Revocação}_k \times Total_k}{Total} \tag{3.10}$$

   Nesta fórmula $Total_k$ representa o total de exemplos na classe k. Em especial para a Revocação, o resultado da média ponderada 3.10 se resume a equação (3.6):

$$\frac{\sum_{k=1}^{K} \frac{VP_k}{Total_k} \times Total_k}{Total} = \frac{\sum_{i=1}^{K} VP_k}{Total} \tag{3.11}$$

3.3 Simulação

Nesta seção, a simulação apresentada representa a aplicação de um modelo previamente treinado. Para uma compreensão mais completa do processamento de dados, desde a extração até a geração de ordens de compra e venda, o fluxograma apresentado na Figura 15 resume a metodologia abordada neste trabalho. Os dados obtidos na etapa de extração consistem em: abertura O, fechamento C, máxima H, mínima L e volume V.

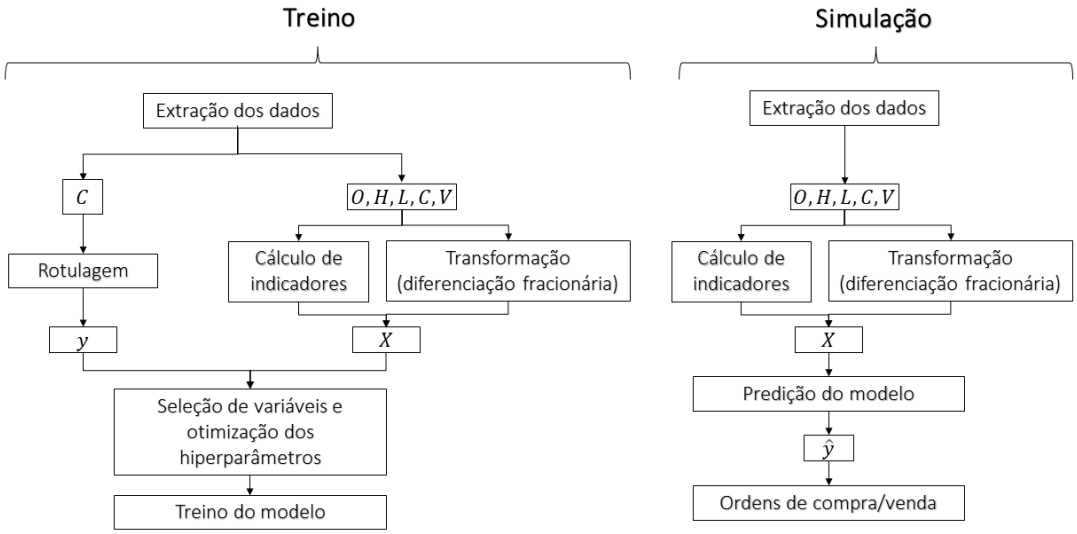


Figura 15: Fluxograma do processamento de dados.

Vale destacar que, para realizar a transformação apresentada na equação (2.9) nos dados do conjunto de simulação, foi utilizado o mesmo parâmetro d obtido no conjunto de treino, pois em uma aplicação real não seria possível ter acesso aos dados do futuro para avaliar a estacionariedade da série.

Com o modelo treinado, foi possível realizar a predição $\hat{y}_i$ com base nas variáveis preditoras previamente apresentadas. Para avaliar a aplicação do modelo no mercado financeiro, a predição $\hat{y}_i$ é então transformada em ordens de compra e venda. Com posições de compra e venda, torna-se possível realizar a avaliação do desempenho da estratégia proposta.

3.3.1 Ordens de compra e venda

A transformação da predição $(-1, 0, 1)$ em posições de compra e venda foi realizada com a aplicação de um filtro, no qual a entrada de uma nova posição é impedida enquanto uma posição anterior igual ainda estiver aberta. Os filtros de sinal são condições ou critérios adicionais que podem ser aplicados à estratégia de negociação antes de permitir a execução de uma ordem de compra ou venda. Esses filtros são utilizados para aprimorar a qualidade das decisões de negociação e para evitar a execução de operações em determinadas condições indesejadas.

O filtro utilizado auxilia na prevenção da acumulação de várias posições simultâneas

pela estratégia. Esse tipo de filtro ajuda a controlar o risco, limitando a exposição ao mercado a uma única posição por vez. Se uma posição de compra estiver aberta e um sinal de compra subsequente for gerado sem que a posição anterior tenha sido fechada, o filtro impede a execução da nova ordem de compra. Definindo $PosCompra_i \in \{0, 1\}$ como uma variável que define se existe uma posição de compra aberta no instante de tempo i, $PosCompra_i = 1$, a ordem de Compra ou Venda definida pela estratégia pode então ser expressa como:

$$\begin{cases} \text{Compra} & \text{se } PosCompra_i = 0 \text{ e } \hat{y}_i = 1 \\ \text{Venda} & \text{se } PosCompra_i = 1 \text{ e } (\hat{y}_i = -1 \text{ ou } |retorno_i| \geq lim) \end{cases} \tag{3.12}$$

onde $|retorno_i|$ representa o retorno percentual obtido no instante de tempo i com relação a última operação.

A variável *lim* representa o *Take Profit* e *Stop Loss* quando aplicado. O *Take Profit* refere-se a um ponto predefinido em que uma estratégia decide encerrar uma posição para realizar lucro, ajudando a garantir que os ganhos sejam capturados em momentos oportunos. Em contraste, o *Stop Loss* é um nível de preço pré-estabelecido em que uma estratégia opta por encerrar uma posição, limitando assim as perdas caso o mercado se mova desfavoravelmente.

3.3.2 *Backtesting*

Com as ordens definidas, foi realizado o *backtesting* da estratégia com o auxílio da biblioteca vectorbt. O *backtesting* envolve a simulação do desempenho da estratégia com base em seu comportamento histórico usando dados passados. A Figura 16 representa a simulação de compra e venda utilizando como base o ativo PETR4.

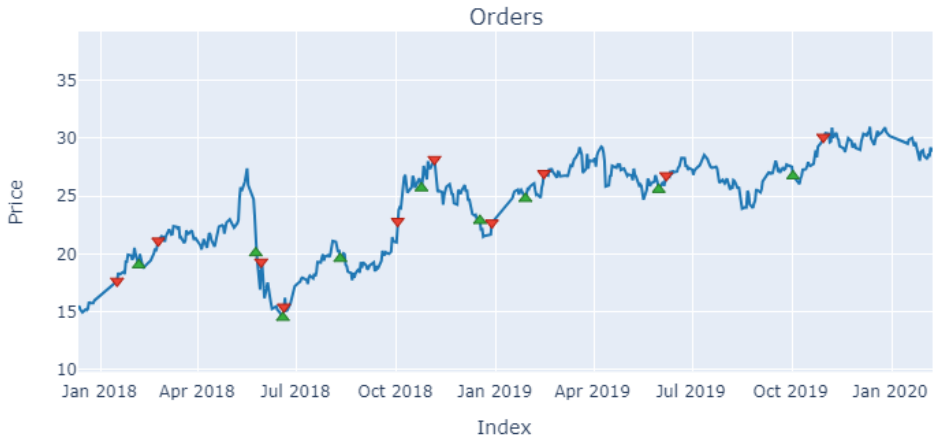


Figura 16: Representação das ordens de compra e venda.

Para cada par $(Compra/Venda)$ representando na Figura 16 é possível associar informações de Lucro e Perda (PnL) percentual conforme a Figura 17. Considera-se que a compra envolve a aplicação de todo o capital em uma posição de compra, e a venda implica em zerar completamente essa posição de compra.

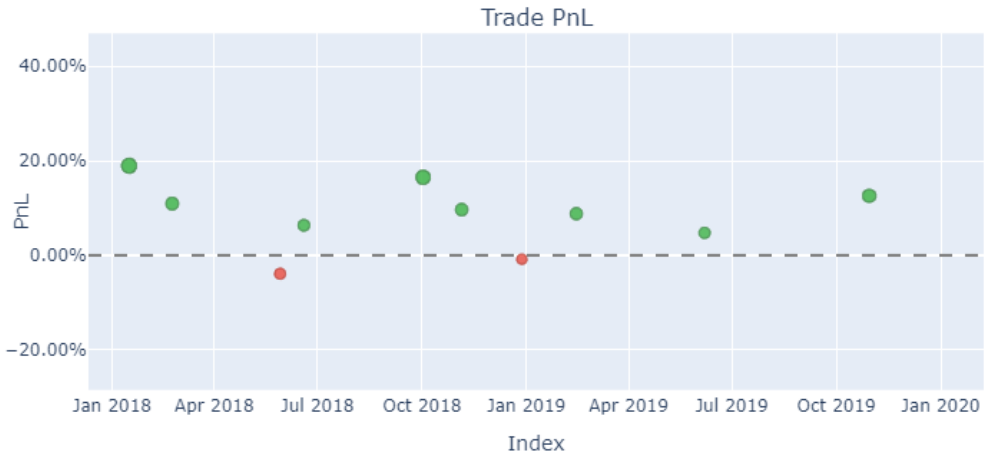


Figura 17: *Trade PnL %*.

O cálculo do *Trade PnL %* é uma métrica fundamental para avaliar a rentabilidade de operações de compra e venda em termos percentuais, o que facilita a comparação entre diferentes operações e estratégias. O *Trade PnL %* é calculado da seguinte forma:

$$\text{Trade PnL \%} = \left(\frac{\text{Preço de Venda} - \text{Preço de Compra}}{\text{Preço de Compra}} \right) \times 100 \tag{3.13}$$

Considerando os lucros ou perdas acumulados de várias operações e as oscilações

no preço do ativo analisado, é possível calcular o retorno acumulado. Dado um valor inicial para o portfólio, podemos atualizar esse valor em cada instante i utilizando o fator $(1 + \text{Retorno}_i)$, de forma geral:

$$\text{Valor Portfólio}_{i+1} = (1 + \text{Retorno}_i) \times \text{Valor Portfólio}_i \tag{3.14}$$

Para avaliar o fator $(1 + \text{Retorno}_i)$ de forma acumulada desde o valor inicial $(i = 0)$, podemos utilizar $\text{Retorno Acumulado}_i$ da seguinte forma:

$$(1 + \text{Retorno Acumulado}_i) = (1 + \text{Retorno Acumulado}_{i-1}) \times (1 + \text{Retorno}_i) \tag{3.15}$$

Essa métrica permite medir o crescimento ou decréscimo do capital ao longo do tempo. Na Figura 18, a curva superior representa o retorno acumulado da estratégia, enquanto a curva inferior representa o valor do ativo.

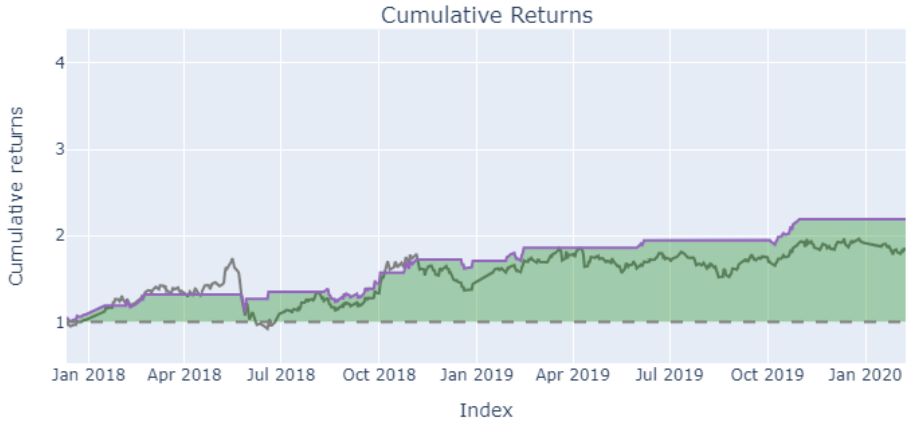


Figura 18: Retorno acumulado

Neste trabalho, analisou-se a métrica de retorno acumulado como uma razão, sem levar em consideração as limitações do capital inicial. Como uma limitação, não foram considerados os custos associados às operações de compra e venda. Além disso, não foi considerado nenhum rendimento adicional durante o período em que não havia nenhuma posição aberta.

3.3.3 Métricas

Para avaliar a estratégia de *trading* proposta, foram utilizadas três métricas: retorno financeiro, taxa de acerto das operações e índice de Sharpe. O retorno financeiro mede a porcentagem de lucro obtido em relação ao capital investido. Enquanto a taxa de acerto das operações mede a proporção de operações bem-sucedidas em relação ao total de operações realizadas. Apresentado primeiramente por (SHARPE, 1966), o índice de Sharpe é uma medida que avalia o desempenho de uma estratégia de investimento em relação ao risco assumido.

Os resultados obtidos no *backtesting* foram comparados com a estratégia *Buy and Hold* e com uma estratégia de *trading* popular conhecida como *Moving Average Crossover* (MAC), que se baseia no cruzamento de duas curvas, uma que representa a tendência de curto prazo e uma segunda curva representando uma tendência de longo prazo. Além disso, foi realizada uma comparação do resultado obtido ao substituir o resultado da classificação do modelo por uma atribuição de classes de modo aleatório.

4 RESULTADOS

Os resultados apresentados refletem o desempenho obtido por meio da aplicação do modelo previamente treinado, uma etapa representada como "simulação" na Figura 15. Nesta análise, os dados do período de 2017 até agosto de 2023 foram empregados, sendo aplicados os modelos LGBM e XGBoost como exemplos de técnicas de *boosting*, e o modelo *Random Forest* como exemplo da aplicação da técnica de *bagging*.

4.1 Avaliação da classificação

Inicialmente, foram avaliados os resultados quanto à capacidade de cada modelo em classificar corretamente a variável alvo y_i. Nessa fase, foram comparados os valores preditos $\hat{y}_i$, obtidos por meio da aplicação de cada modelo, com a variável y_i.

A Tabela 2 apresenta a Acurácia para cada ativo e modelo. Os resultados indicam a proporção de previsões corretas em relação ao total. Observa-se que, em geral, todos os modelos obtiveram desempenhos semelhantes.

Ativo	**LGBM**	**RF**	**XGB**
ABEV3	0,68	0,69	0,68
BBDC4	0,66	0,66	0,64
BOVA11	0,65	0,68	0,66
ITUB4	0,68	0,69	0,65
PETR4	0,67	0,68	0,69
VALE3	0,68	0,70	0,68

Tabela 2: Acurácia por Modelo

Além da acurácia, foram obtidas as métricas de Precisão e *F1 Score*. Essas métricas são calculadas individualmente para cada classe, conforme as equações 3.7 e 3.9, e o resultado apresentado na Tabela 3 representa a média ponderada considerando todas as

classes -1, 0 e 1.

Ativo	**Precisão**			**F1 Score**		
	LGBM	**RF**	**XGB**	**LGBM**	**RF**	**XGB**
ABEV3	0,64	0,64	0,67	0,65	0,65	0,67
BBDC4	0,66	0,63	0,65	0,66	0,64	0,64
BOVA11	0,66	0,63	0,66	0,66	0,64	0,66
ITUB4	0,67	0,64	0,66	0,67	0,66	0,66
PETR4	0,66	0,64	0,68	0,66	0,65	0,68
VALE3	0,66	0,66	0,67	0,66	0,67	0,67

Tabela 3: Precisão e F1 Score por Modelo

A métrica de Revocação não foi apresentada neste contexto, uma vez que, no cálculo da média ponderada de todas as classes, seu resultado é equivalente ao da acurácia, conforme equação (3.11).

Medir apenas a acurácia de um modelo pode ser enganoso, pois essa métrica não leva em consideração o desequilíbrio entre as classes. Ao avaliar a média ponderada dos valores de Precisão e F1 Score de cada classe, temos também uma forte influência da classe com maior volume, que nesse exemplo seria o valor de $y_i = 0$, valores não definidos como máximo ou mínimo na etapa de rotulagem.

Outra métrica utilizada para avaliar a classificação, visando reduzir o efeito do desequilíbrio entre as classes na avaliação dos resultados, consiste na média simples da Revocação de cada classe. Essa métrica é também conhecida como Acurácia Balanceada (GRANDINI et al., 2020).

Ativo	**LGBM**	**RF**	**XGB**
ABEV3	0,44	0,43	0,51
BBDC4	0,53	0,43	0,53
BOVA11	0,52	0,42	0,52
ITUB4	0,51	0,44	0,54
PETR4	0,49	0,43	0,52
VALE3	0,47	0,44	0,52

Tabela 4: Acurácia Balanceada

Observando os resultados da Tabela 4, podemos notar que o modelo XGB geralmente

superou os outros dois modelos para a maioria dos ativos.

É essencial avaliar não apenas a precisão geral de um modelo de classificação, mas também como e onde ocorrem os erros de classificação. A natureza desses erros pode ter um impacto significativo nas decisões tomadas com base nas previsões do modelo.

Considerando que a classificação do modelo é utilizada para definir momentos de compra e venda no mercado financeiro, o erro de classificar um ponto de venda (-1) como um ponto de compra (1) é potencialmente o mais crítico, podendo resultar em perdas substanciais. Por outro lado, o erro de classificar um ponto de compra (1) como um ponto de espera (0) é menos crítico. Nesse cenário, o modelo está apenas deixando uma oportunidade de compra, o que pode resultar em oportunidades perdidas de lucro, mas geralmente não implica perdas substanciais.

Para avaliar detalhadamente os erros de classificação de cada modelo em relação a cada ativo, foi utilizado o conceito da matriz de confusão. Essa matriz oferece uma visão detalhada de como as previsões do modelo se comparam aos valores reais. A Figura 19 apresenta o resultado da matriz de confusão para o ativo PETR4 com o uso do modelo XGB.

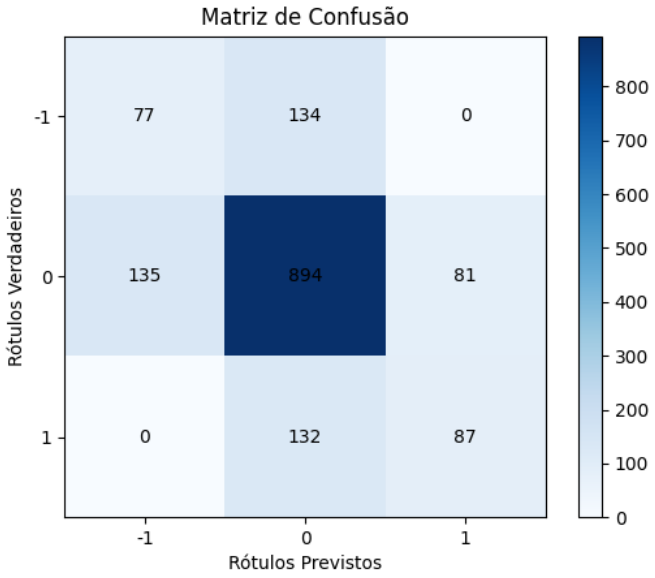


Figura 19: Matriz de confusão (PETR4 - XGB).

A matriz de confusão mostra o número de observações classificadas corretamente (verdadeiros positivos e verdadeiros negativos) e as classificações incorretas (falsos positivos e falsos negativos) em relação às classes reais. A diagonal principal da matriz contém os

valores que foram previstos corretamente para cada classe.

Apesar de observarmos uma alta taxa de erro envolvendo a classe (0), ou seja, dados que foram rotulados como (-1) ou (1) e são previstos erroneamente como (0), é importante ressaltar que a taxa de erro entre as classes (-1) e (1) está zerada. Este comportamento foi evidenciado através da matriz de confusão apresentada para o ativo PETR4 e o modelo XGB. No entanto, é importante destacar que essa tendência se mantém consistente ao avaliar todos os modelos e ativos, o que sugere que os modelos são especialmente eficazes em distinguir entre as classes extremas (-1) e (1).

4.2 Avaliação do *Backtesting*

Com o resultado do *Backtesting*, a avaliação dos modelos foi iniciada com base em suas taxas de acerto nas estratégias de negociação, expressas como razões. Antes de analisar as taxas de acerto, a Tabela 5 apresenta os valores absolutos de volume de trades. É importante ressaltar que o volume de trades envolve custos associados que podem ter um impacto significativo no resultado financeiro de uma estratégia. No entanto, por simplificação, os resultados apresentados neste trabalho não levam em consideração o custo de cada transação, embora esse seja um fator relevante a ser considerado em uma aplicação prática.

Como comparação contra os modelos de classificação, foi avaliado o resultado de uma estratégia de *trading* baseada no cruzamento de duas médias móveis para o mesmo período (MAC). Essa estratégia considera uma curva com a média de 10 dias e outra com a média de 20 dias, sendo uma abordagem comum em análise técnica, e oferece uma perspectiva adicional sobre a eficácia das estratégias em comparação com os modelos preditivos.

Além disso, também foi avaliado o desempenho de uma estratégia de negociação aleatória, denominada como "Random". Nessa abordagem, os trades foram executados de forma completamente aleatória, sem qualquer base em análise técnica ou modelos preditivos. Para cada ativo, foi considerado um volume de trades igual a 50.

Ativo	MAC	Random	LGBM	RF	XGB
ABEV3	38	50	45	26	58
BBDC4	38	50	70	43	67
BOVA11	42	50	72	32	71
ITUB4	40	50	65	40	74
PETR4	44	50	61	46	58
VALE3	37	50	52	33	63

Tabela 5: Tabela de Trades por Modelo

A partir da avaliação do total de Trades que são apresentados na tabela 5, é possível calcular a taxa de acerto, calculada dividindo o número de operações vencedoras pelo número total de operações.

Ativo	MAC	Random	LGBM	RF	XGB
ABEV3	0,47	0,56	0,62	0,65	0,64
BBDC4	0,45	0,40	0,63	0,60	0,51
BOVA11	0,43	0,44	0,58	0,66	0,69
ITUB4	0,40	0,42	0,66	0,68	0,69
PETR4	0,39	0,42	0,72	0,74	0,64
VALE3	0,54	0,50	0,69	0,67	0,70

Tabela 6: Tabela de Taxa de Acerto por Modelo

Para reduzir o impacto do número de negociações na métrica apresentada na Tabela 6, também foi calculada a taxa de acerto acumulada por modelo, considerando todas as negociações de todos os ativos avaliados. A Figura 20 apresenta o resultado dessa taxa acumulada em função do número de negociações. Isso proporciona uma perspectiva mais abrangente do desempenho de uma estratégia ao longo do tempo.

Quando consideramos exclusivamente a taxa de acerto, observamos que, em todos os ativos, os modelos avaliados apresentaram um desempenho superior na identificação de oportunidades de negociação bem-sucedidas. Embora seja amplamente utilizada na avaliação de estratégias de investimento ou negociação, essa métrica não fornece uma visão abrangente da rentabilidade. Isso ocorre porque ela não leva em consideração o tamanho das vitórias e perdas em cada negociação, ou seja, não considera a magnitude dos resultados individuais.

Além da taxa de acerto, outra métrica amplamente utilizada para avaliar a eficácia

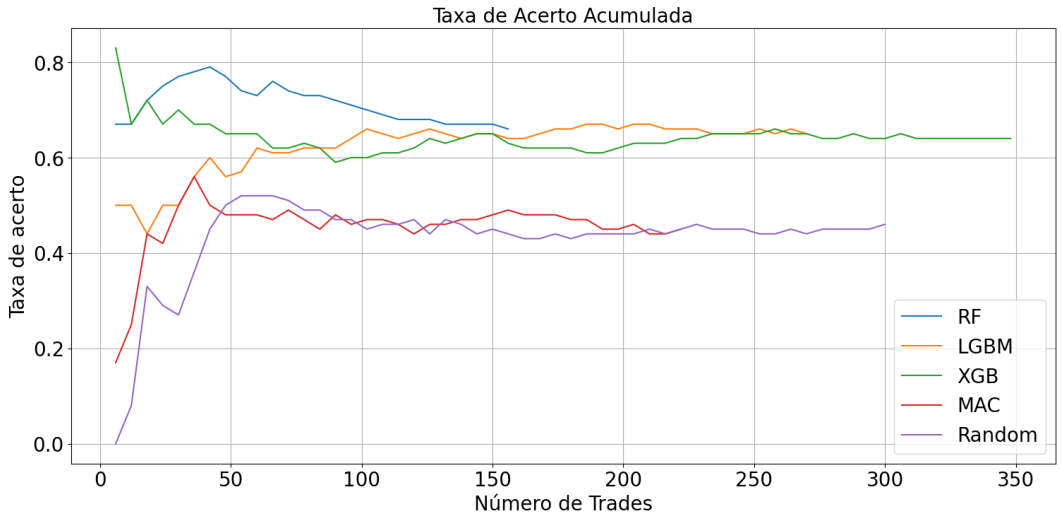


Figura 20: Taxa de Acerto Acumulada.

de estratégias de investimento ou negociação é o Índice de Sharpe. O cálculo do Índice de Sharpe envolve a divisão do excesso de retorno médio do portfólio em relação ao retorno da renda fixa pela volatilidade, que é representada pelo desvio padrão dos retornos. Essa divisão busca determinar a eficiência da estratégia, medindo o retorno obtido em relação ao risco assumido.

Ativo	**MAC**	**Random**	**LGBM**	**RF**	**XGB**
ABEV3	0.30	-0.11	0.20	0.23	0.37
BBDC4	0.47	-0.60	0.31	0.61	0.26
BOVA11	0.80	0.17	0.22	0.55	0.21
ITUB4	0.53	-0.43	0.54	0.58	0.70
PETR4	0.23	-0.75	1.05	1.10	0.72
VALE3	0.73	0.17	0.53	0.77	1.05

Tabela 7: Tabela de índice Sharpe

Apesar de uma diferença mais acentuada na taxa de acerto, a comparação dos Índices de Sharpe apresentados na Tabela 7 entre as estratégias que utilizam os modelos LGBM, RF e XGB em comparação com a estratégia MAC demonstra resultados semelhantes. É importante ressaltar que, em comparação com o resultado obtido pela geração de ordens aleatórias, torna-se evidente a desvantagem da estratégia que se baseia exclusivamente em informações aleatórias.

Outra importante medida considerada na avaliação dessas estratégias de investimento foi o retorno financeiro. Essa métrica avalia o desempenho financeiro geral de cada estra-

tégia, medindo o lucro ou prejuízo acumulado ao longo do tempo. O retorno financeiro fornece uma perspectiva fundamental sobre o impacto real das estratégias no patrimônio do investidor, sendo uma métrica crítica para avaliar a eficácia a longo prazo. Para a análise dessa métrica, também foram levados em consideração os resultados obtidos com a estratégia de *Buy and Hold*, a qual envolve a aquisição simples de cada ativo, seguida pela espera ao longo do tempo.

A Tabela 8 apresenta uma visão do retorno obtido para cada ativo com a utilização de diferentes estratégias, incluindo a aplicação dos modelos de classificação.

Ativo	**MAC**	**Random**	**LGBM**	**RF**	**XGB**	**BuyHold**
ABEV3	0.20	-0.23	0.06	0.10	0.33	-0.14
BBDC4	0.49	-0.65	0.22	0.75	0.15	0.02
BOVA11	0.76	0.04	0.10	0.60	0.08	0.89
ITUB4	0.57	-0.49	0.64	0.75	1.02	0.17
PETR4	0.05	-0.85	3.75	2.85	1.50	0.94
VALE3	1.14	-0.02	0.74	1.69	2.16	1.15

Tabela 8: Tabela de Retorno por Modelo

Para ilustrar o desempenho geral de cada modelo, a Figura 21 apresenta o retorno acumulado de uma carteira composta com todos os ativos que foram avaliados, considerando que o valor inicialmente alocado em cada ativo seja exatamente o mesmo.

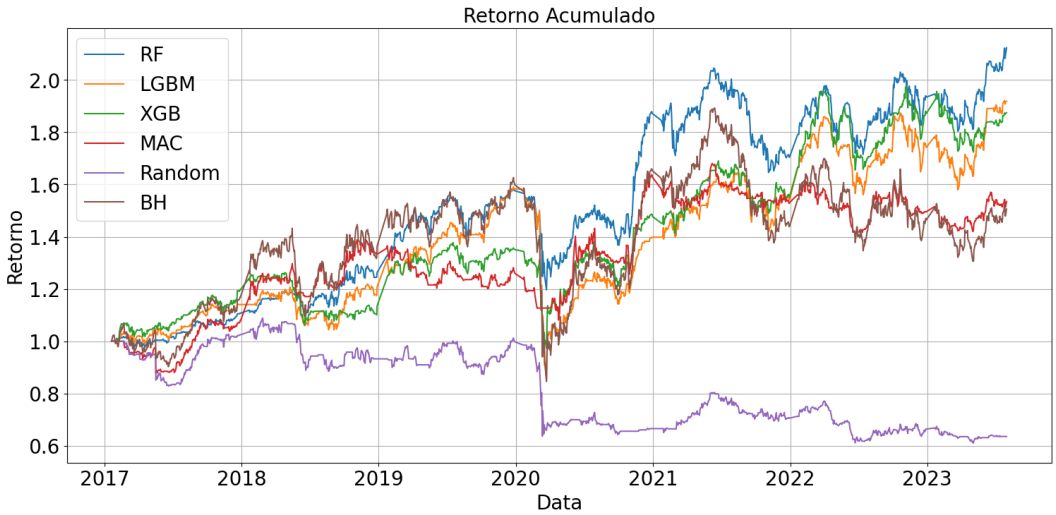


Figura 21: Retorno acumulado da carteira (%).

O resultado apresentado na Figura 21 consolida o resultado individual de cada ativo apresentado na Tabela 8.

Ao avaliar o retorno financeiro, com exceção do resultado obtido com dados aleatórios, ambas as estratégias apresentaram, no geral, um resultado positivo, embora tenham sido bastante dispersos entre os ativos avaliados. Analisando o gráfico da Figura 21, apesar de os modelos de classificação terminarem o período avaliado com um resultado superior, é importante considerar a volatilidade apresentada no gráfico. Em determinados períodos, esses modelos apresentaram desempenho inferior ao MAC. Portanto, não é possível concluir que houve uma superioridade clara.

Para validar a diferença entre os modelos que utilizaram aprendizado de máquina com as demais estratégias, foi realizado um teste estatístico t de Student. Esse teste é conduzido para comparar as médias entre dois grupos e determinar se há uma diferença estatisticamente significativa entre eles.

O teste foi conduzido utilizando os valores de retorno ao longo de todo o período avaliado e para todos os ativos. Inicialmente, procedeu-se à análise da distribuição dos retornos diários para cada estratégia, como exemplificado na Figura 22.

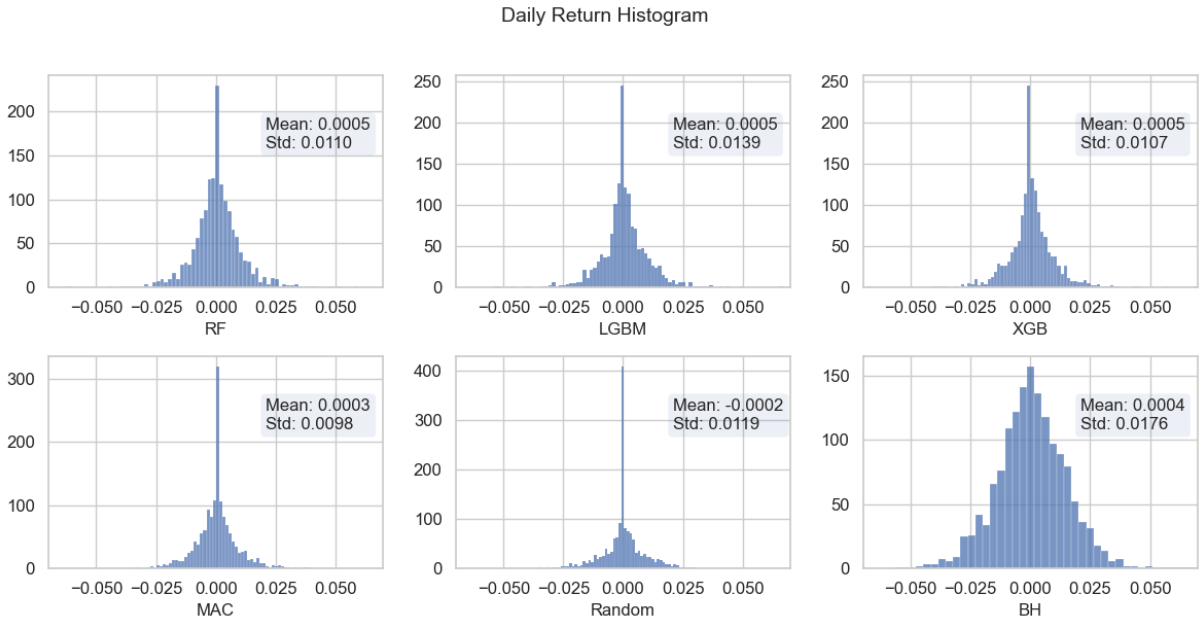


Figura 22: Histograma do retorno diário.

Com exceção do histograma que representa o *Buy and Hold*, ambos exibem uma notável concentração no retorno igual a zero. Essa concentração decorre de períodos nos quais não há posições de compra, resultando em um valor constante para a carteira. A fim de mitigar essa influência, o gráfico da Figura 23 apresenta o retorno diário considerando apenas os momentos em que o retorno não é zero, proporcionando uma visualização mais clara dos padrões e movimentos relevantes.

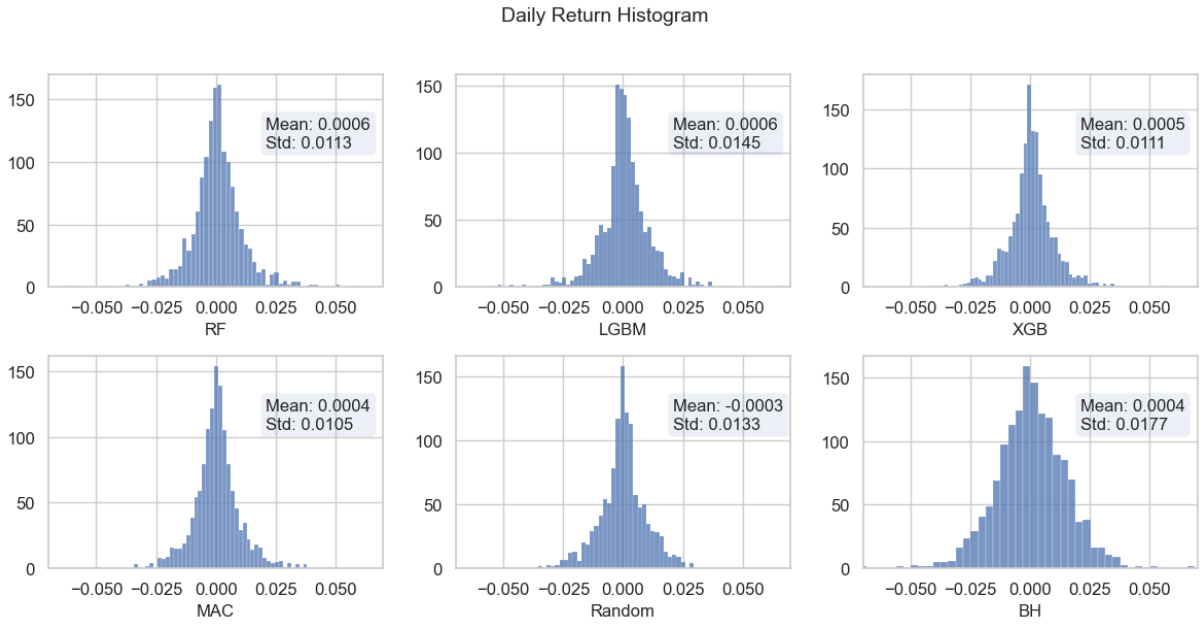


Figura 23: Histograma do retorno diário.

Com base nos histogramas apresentados, foi aplicado o teste t de Student para analisar se existem diferenças significativas nas médias dos retornos diários entre os modelos baseados em aprendizado de máquina e as estratégias de referência (*Buy and Hold* e MAC) conforme a Tabela 9.

Model	MAC	BH
RF	0.619	0.780
LGBM	0.690	0.815
XGB	0.760	0.894

Tabela 9: p-values.

O *p-value*, ou valor p, é uma medida estatística que indica a probabilidade de observar resultados tão extremos quanto os obtidos em um teste estatístico, sob a hipótese nula, que postula a igualdade de médias entre os dois grupos analisados. Os resultados sugerem que, no contexto desta análise, não há evidências convincentes para afirmar que as médias dos retornos diários dos modelos RF, XGB e LGBM são estatisticamente diferentes das médias das estratégias de referência MAC e BH.

5 CONCLUSÃO

Neste estudo, foi abordada a aplicação de um modelo de classificação para o desenvolvimento de uma estratégia de *trading*. Os algoritmos LGBM, XGB e *Random Forest* foram adotados como modelos de aprendizado de máquina. Para estruturar a abordagem de classificação, utilizou-se a técnica de rotulagem conhecida como Min-Max. Como variáveis preditoras, foram empregados os valores de abertura, fechamento, alta, baixa e volume, bem como os indicadores extraídos a partir desses dados.

A avaliação da capacidade de classificação do modelo envolveu a análise dos resultados de acurácia, precisão, revocação e F1 Score. Além disso, para avaliar a eficácia do modelo como estratégia de negociação, foram comparados os índices de Sharpe, as taxas de acerto e os retornos financeiros do modelo em relação à estratégia MAC e em relação à geração aleatória de ordens de compra e venda. Todos os resultados apresentados foram obtidos por meio da análise dos anos de 2017 até 2023.

No que diz respeito às métricas empregadas na avaliação da capacidade de classificação dos modelos, ambos apresentaram resultados comparáveis aos encontrados em estudos semelhantes na literatura.

Na comparação realizada após o *backtesting*, a adoção de um modelo de classificação com a utilização de indicadores técnicos de mercado demonstra o potencial de desenvolver estratégias com uma taxa de acerto superior, em comparação com abordagens populares também baseadas unicamente em registros históricos, como a estratégia MAC.

Em relação a outras métricas, como o retorno acumulado, embora as estratégias que utilizam modelos de classificação tenham apresentado um resultado ligeiramente superior ao final do período avaliado, a Figura 21 ilustra a dependência direta desse resultado em relação à janela de tempo considerada no *backtesting*. Isso torna difícil afirmar que haja um ganho real independente do período avaliado. Adicionalmente, o teste estatístico realizado indica que não podemos concluir de forma robusta uma diferença significativa na média de retornos entre as estratégias analisadas.

Além disso, na comparação com o *Buy and Hold*, é crucial reiterar que, na prática, uma implementação real dessas estratégias envolveria considerações adicionais, tais como os custos de transação. Por outro lado, em uma aplicação real, seria viável otimizar a alocação dos recursos disponíveis. Durante os períodos em que não há posição de compra em um ativo específico, $PosCompra_i = 0$, esses recursos poderiam ser realocados para outra aplicação ou mesmo aumentar a posição de compra em outros ativos da carteira.

Apesar do aumento na taxa de acerto com o uso de aprendizado de máquina, garantir a rentabilidade de uma estratégia requer a consideração do dimensionamento do valor a ser alocado em cada ordem. Este tópico não foi abordado neste trabalho, mas é possível estender a utilização da classificação do modelo de aprendizado de máquina para ajudar a determinar o valor a ser alocado em cada operação. Por exemplo, é viável extrair dos modelos não apenas a previsão de cada classe, mas também a probabilidade associada a essa previsão, conforme a equação (2.21). Isso permite dimensionar o capital que deve ser investido em cada operação como uma função de $p(y_i = 1|\mathbf{x_i})$, sendo o valor investido proporcional ao valor de $p(y_i = 1|\mathbf{x_i})$. Em (PRADO, 2020) são apresentados mais detalhes dessa aplicação.

Em estudos futuros, além de considerar o dimensionamento dos investimentos em cada operação, seria interessante incorporar novas variáveis preditoras, como aquelas derivadas da análise de sentimentos. Essa análise pode ser realizada ao avaliar menções sobre os ativos em textos de mídias sociais. Além disso, seria importante avaliar outros períodos, ampliando a compreensão do desempenho das estratégias em diferentes condições de mercado.

ANEXO A – ALGORITMOS

Todos os algoritmos utilizados neste trabalho estão disponíveis no seguinte repositório do GitHub: <https://github.com/thiagorayam/ml_finance>

A.1 Rotulagem (Mínimo e Máximo)

Input: Fechamento, TamanhoJanela, TotalDias

Output: Vetor y

Function `Rotulagem(`*input*`):`

$Inicio \leftarrow 0$

$Fim \leftarrow Inicio + TamanhoJanela$

$Contador \leftarrow 0$

$y[:] = 0$

while $Contador \leq TotalDias$ **do**

$valor_{min} \leftarrow min(Fechamento[Inicio : Fim])$

$valor_{max} \leftarrow max(Fechamento[Inicio : Fim])$

$idx_{valor_{min}} \leftarrow idxmin(Fechamento[Inicio : Fim])$

$idx_{valor_{max}} \leftarrow idxmax(Fechamento[Inicio : Fim])$

if $valor_{min} < Fechamento[Inicio]$ AND $valor_{min} < Fechamento[Fim]$ **then**

$y[idx_{valor_{min}}] \leftarrow 1$

end

if $valor_{max} > Fechamento[Inicio]$ AND $valor_{max} > Fechamento[Fim]$ **then**

$y[idx_{valor_{max}}] \leftarrow -1$

end

$Inicio \leftarrow int((Fim + Inicio)/2)$

$Fim \leftarrow Inicio + TamanhoJanela$

$Contador \leftarrow Fim$

end

return y

Algorithm 1: Algoritmo de rotulagem dos dados

ANEXO B – INDICADORES TÉCNICOS

Neste anexo, estão apresentados os indicadores técnicos utilizados neste trabalho. Esses indicadores foram calculados com o auxílio da biblioteca Python Technical Analysis Library (TALib), que oferece uma ampla gama de ferramentas para análise técnica.

Tabela 10: Indicadores de Volume

Indicador	**Expressão**	**Observação**
FI	$(C_i - C_{i-1})V_i$	
EoM	$(H_i - L_i)\frac{(H_i - H_{i-1}) + (L_i - L_{i-1})}{2V_i}$	Considerado o indicador direto e a sua média móvel 14 dias
MFI	$100 - \frac{100}{1+\text{MFR}_i}$	MFR representa a razão entre variações positivas/-negativas do preço típico $\frac{H_i+L_i+C_i}{3}$ nos últimos 14 dias
VPT	$VPT_{i-1} + V_i(\frac{C_i - C_{i-1}}{C_{i-1}})$	
CMF	$\frac{\sum_{j=i-20}^{i} MFV_i}{\sum_{j=i-20}^{i} V_j}$	MFV_i é dado por $\frac{(C_i - L_i) - (H_i - C_i)}{H_i - L_i} V_i$

Tabela 11: Indicadores de Volatilidade (*Bollinger Bands*)

Indicador	**Expressão**	**Observação**
BBW	$100\frac{UpperBand_i - LowerBand_i}{MiddleBand_i}$	
BBP	$\frac{C_i - LowerBand_i}{UpperBand_i - LowerBand_i}$	
BBH	1 se $C_i > UpperBand_i$	variável binária $(0, 1)$
BBI	1 se $C_i < LowerBand_i$	variável binária $(0, 1)$

Notas: $MiddleBand_i$ representa a média móvel (20 dias) do fechamento C_i, enquanto $UpperBand_i$ e $LowerBand_i$ são somados e subtraídos de dois desvios padrão.

Tabela 12: Indicadores de Volatilidade (*Keltner Channel*)

Indicador	**Expressão**	**Observação**
TR	$\max(H_i - L_i, \lvert H_i - C_{i-1}\rvert, \lvert L_i - C_{i-1}\rvert)$	
ATR	$\frac{\sum_{l=0}^{14}(TR_{i-l})}{15}$	média móvel (15 dias)
KCW	$100\frac{UpperBand_i - LowerBand_i}{MiddleBand_i}$	
KCP	$\frac{C_i - LowerBand_i}{UpperBand_i - LowerBand_i}$	
KCH	1 se $C_i > UpperBand_i$	variável binária $(0, 1)$
KCI	1 se $C_i < LowerBand_i$	variável binária $(0, 1)$

Notas: $MiddleBand_i$ representa a média móvel exponencial (20 dias) do fechamento C_i, enquanto $UpperBand_i$ e $LowerBand_i$ são somados e subtraídos de dois ATR.

Tabela 13: Indicadores de Volatilidade (outros)

Indicador	**Expressão**	**Observação**
DCW	$100\frac{UpperBand_i - LowerBand_i}{MiddleBand_i}$	
DCP	$\frac{C_i - LowerBand_i}{UpperBand_i - LowerBand_i}$	
UI	$\sqrt{\frac{\sum_{j=i-14}^{i}(PercentDrawdown_i)^2}{14}}$	

Notas: $MiddleBand_i$ é a média móvel (20 dias) do preço de fechamento C_i, enquanto $UpperBand_i$ é o máximo dos preços mais altos H_i dos últimos 20 dias e $LowerBand_i$ é o mínimo dos preços mais baixos L_i dos últimos 20 dias; $PercentDrawdown_i$ é calculado subtraindo o preço de fechamento atual do preço máximo de fechamento dos últimos 14 dias e dividindo esse resultado pelo preço máximo de fechamento dos últimos 14 dias.

Tabela 14: Indicadores de Tendência

Indicador	**Expressão**	**Observação**
MACD	$EMA_{12i} - EMA_{26i}$	média movel exponencial de C_i (12 e 26 dias)
VI^+	$\frac{\sum_{l=0}^{14}(VM^+{}_{i-l})}{\sum_{l=0}^{14}(TR_{i-l})}$	$VM^+{}_i = H_i - L_{i-1}$
VI^-	$\frac{\sum_{l=0}^{14}(VM^-{}_{i-l})}{\sum_{l=0}^{14}(TR_{i-l})}$	$VM^-{}_i = L_i - H_{i-1}$
TRIX	$100\frac{EMA^3{}_i - EMA^3{}_{i-1}}{EMA^3{}_{i-1}}$	EMA^3 representa a média móvel exponencial (15 dias) sendo aplicada 3 vezes em C_i
Mass	$\sum_{l=0}^{24} \frac{EMA(H_i - L_i)}{EMA(EMA(H_i - L_i))}$	EMA representa a média móvel exponencial (9 dias)
DPO	$C_{i-11} - SMA(C_i)$	SMA representa a média móvel (20 dias)
KST	$\sum_{l=1}^{4} l \times \text{RCMA}_l$	
STC	$100\frac{Stoch_d - Stoch_{d\,min}}{Stoch_{d\,max} - Stoch_{d\,min}}$	$Stoch$ é a média móvel móvel exponencial de $100\frac{MACD - MACD_{min}}{MACD_{max} - MACD_{min}}$.
CCI	$\frac{TP_i - SMA(TP_i)}{0.015 \times MAD(TP_i)}$	
AUp	$100\frac{25 - (\text{Dias desde o Último Máximo})}{25}$	
ADown	$100\frac{25 - (\text{Dias desde o Último Mínimo})}{25}$	

Notas: Com $MACD$ foram calculados 3 indicadores: $MACD$, sua média móvel e a diferença entre as duas medidas; SMA e MAD representam a média e o desvio médio (20 dias); Além de $VM^+{}_i$ e $VM^-{}_i$ também foi utilizado a diferença; $TP_i = (H_i + L_i + C_i)/3$; $RCMA$ é a média móvel centrada acumulada da taxa de variação de C_i (centrada em $i - 10, i - 15, i - 20$ e $i - 30$).

Tabela 15: Indicadores de Momentum

Indicador	**Expressão**	**Observação**
RSI	$100 - \frac{100}{1+\frac{\sum_{l=0}^{13} UP_{i-l}/14}{\sum_{l=0}^{13} DW_{i-l}/14}}$	
StochRSI	$\frac{(RSI_i - \min(RSI_i))}{(\max(RSI_i) - \min(RSI_i))}$	Mínimo e Máximo dos últimos 14 dias
StochRSI$_k$	SMA$(StochRSI)$	Média móvel simples (3 dias)
StochRSI$_d$	SMA$(StochRSI_k)$	Média móvel simples (3 dias)
TSI	$100\frac{EMA_{25}(EMA_{13}(C_i - C_{i-1}))_i}{EMA_{25}(EMA_{13}(\lvert C_i - C_{i-1}\rvert))_i}$	EMA representa a média móvel exponencial (25 dias e 13 dias)
Williams %R	$-100\frac{\max_{l=0}^{13}(H_{i-l}) - C_i}{\max_{l=0}^{13}(H_{i-l}) - \min_{l=0}^{13}(L_{i-l})}$	
$BP_{avg}(L)$	$\frac{\sum_{l=0}^{L} \min(C_{i-l-1}, L_{i-1})}{\sum_{l=0}^{L} TR_{i-l}}$	
UO	$100\frac{4\times BP_{avg}(7) + 2\times BP_{avg}(14) + BP_{avg}(28)}{7}$	
ROC	$100\frac{C_i - C_{i-12}}{C_{i-12}}$	
PPO	$100\frac{EMA_{12}(C_i) - EMA_{26}(C_i)}{EMA_{26}(C_i)}$	Outros indicadores são dados pela média móvel (9 dias) do PPO e sua diferença
PVO	$100\frac{EMA_{12}(V_i) - EMA_{26}(V_i)}{EMA_{26}(V_i)}$	Outros indicadores são dados pela média móvel (9 dias) do PVO e sua diferença
Stoch$_s$	$\frac{(C_i - \min(L_i))}{(\max(H_i) - \min(L_i))}$	Mínimo e Máximo dos últimos 14 dias
Stoch$_s$	SMA$(StochK)$	Média móvel simples (3 dias)

Notas: UP representa movimentos de subida no preço de fechamento; DW representa movimentos de queda no preço de fechamento;

Além dos indicadores de Volume, Volatilidade, Tendência e Momentum apresentados, foram utilizados os indicadores *Daily Log Return* ($DLR = ln(\frac{C_i}{C_{i-1}})$) e *Daily Return* ($DR = \frac{C_i}{C_{i-1}} - 1$).

REFERÊNCIAS

ANZAHAEI, S. M. M.; NIKOOMARAM, H. A comparative study of the performance of stock trading strategies based on lgbm and catboost algorithms. **International Journal of Finance & Managerial Accounting**, Iranian Financial Engineering Association (IFEA), v. 7, n. 26, p. 63–75, 2022.

BREIMAN, L. Bagging predictors. **Machine learning**, Springer, v. 24, p. 123–140, 1996.

_____. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.

BREIMAN, L.; FRIEDMAN, J. H.; OLSHEN, R. A.; STONE, C. J. **Classification and regression trees**. [S.l.]: Routledge, 2017.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. [S.l.: s.n.], 2016. p. 785–794.

CHOWDHURY, U. N.; CHAKRAVARTY, S. K.; HOSSAIN, M. T.; AHMAD, S. Empirical mode decomposition based ensemble random forest model for financial time series forecasting. 2019.

COLBY, R. W. **The encyclopedia of technical market indicators**. [S.l.]: McGraw-Hill, 2003.

DANIEL, F. Financial time series data processing for machine learning. **arXiv preprint arXiv:1907.03010**, 2019.

DEWANCKER, I.; MCCOURT, M.; CLARK, S. Bayesian optimization for machine learning: A practical guidebook. **arXiv preprint arXiv:1612.04858**, 2016.

DICKEY, D. A.; FULLER, W. A. Distribution of the estimators for autoregressive time series with a unit root. **Journal of the American statistical association**, Taylor & Francis, v. 74, n. 366a, p. 427–431, 1979.

DIXIT, A.; JAIN, S. Effect of stationarity on traditional machine learning models: Time series analysis. In: **2021 Thirteenth International Conference on Contemporary Computing (IC3-2021)**. [S.l.: s.n.], 2021. p. 303–308.

DUAN, T.; ANAND, A.; DING, D. Y.; THAI, K. K.; BASU, S.; NG, A.; SCHULER, A. Ngboost: Natural gradient boosting for probabilistic prediction. In: PMLR. **International conference on machine learning**. [S.l.], 2020. p. 2690–2700.

FRATTINI, A.; BIANCHINI, I.; GARZONIO, A.; MERCURI, L. Financial technical indicator and algorithmic trading strategy based on machine learning and alternative data. **Risks**, Multidisciplinary Digital Publishing Institute, v. 10, n. 12, p. 225, 2022.

FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. **Annals of statistics**, JSTOR, p. 1189–1232, 2001.

_____. Stochastic gradient boosting. **Computational statistics & data analysis**, Elsevier, v. 38, n. 4, p. 367–378, 2002.

GENUER, R.; POGGI, J.-M. Random forests. In: **Random Forests with R**. [S.l.]: Springer, 2020. p. 33–55.

GRANDINI, M.; BAGLI, E.; VISANI, G. Metrics for multi-class classification: an overview. **arXiv preprint arXiv:2008.05756**, 2020.

HAN, Y.; KIM, J.; ENKE, D. A machine learning trading system for the stock market based on n-period min-max labeling using xgboost. **Expert Systems with Applications**, Elsevier, v. 211, p. 118581, 2023.

HOSKING, J. M.(1981),". **Fractional differencing. Biometrika**, v. 68, n. 1, p. 165–76.

HUANG, W.; NAKAMORI, Y.; WANG, S.-Y. Forecasting stock market movement direction with support vector machine. **Computers & operations research**, Elsevier, v. 32, n. 10, p. 2513–2522, 2005.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. et al. **An introduction to statistical learning**. [S.l.]: Springer, 2013. v. 112.

JANG, G.-S.; LAI, F.; JIANG, B.-W.; CHIEN, L.-H. An intelligent trend prediction and reversal recognition system using dual-module neural networks. In: IEEE COMPUTER SOCIETY. **Proceedings First International Conference on Artificial Intelligence Applications on Wall Street**. [S.l.], 1991. p. 42–43.

KE, G.; MENG, Q.; FINLEY, T.; WANG, T.; CHEN, W.; MA, W.; YE, Q.; LIU, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. **Advances in neural information processing systems**, v. 30, 2017.

KUMAR, M.; THENMOZHI, M. Forecasting stock index movement: A comparison of support vector machines and random forest. In: **Indian institute of capital markets 9th capital markets conference paper**. [S.l.: s.n.], 2006.

LOH, W.-Y. Classification and regression trees. **Wiley interdisciplinary reviews: data mining and knowledge discovery**, Wiley Online Library, v. 1, n. 1, p. 14–23, 2011.

NAIK, N.; MOHAN, B. R. Stock price movements classification using machine and deep learning techniques-the case study of indian stock market. In: SPRINGER. **Engineering Applications of Neural Networks: 20th International Conference, EANN 2019, Xersonisos, Crete, Greece, May 24-26, 2019, Proceedings 20**. [S.l.], 2019. p. 445–452.

NASCIMENTO, D.; COSTA, A.; BIANCHI, R. Stock trading classifier with multichannel convolutional neural network. In: SBC. **Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional**. [S.l.], 2020. p. 282–293.

NELSON, D. M.; PEREIRA, A. C.; OLIVEIRA, R. A. D. Stock market's price movement prediction with lstm neural networks. In: IEEE. **2017 International joint conference on neural networks (IJCNN)**. [S.l.], 2017. p. 1419–1426.

PALAZZO, G.; SBRUZZI, E. F.; NASCIMENTO, C. L.; LELES, M. C. Predicting litecoin price movement in a pre-defined trading volume window using random forest model. In: IEEE. **2023 IEEE International Systems Conference (SysCon)**. [S.l.], 2023. p. 1–6.

PATEL, J.; SHAH, S.; THAKKAR, P.; KOTECHA, K. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. **Expert systems with applications**, Elsevier, v. 42, n. 1, p. 259–268, 2015.

PRADO, M. L. D. **Advances in financial machine learning**. [S.l.]: John Wiley & Sons, 2018.

_____. **Machine learning for asset managers**. [S.l.]: Cambridge University Press, 2020.

PROKHORENKOVA, L.; GUSEV, G.; VOROBEV, A.; DOROGUSH, A. V.; GULIN, A. Catboost: unbiased boosting with categorical features. **Advances in neural information processing systems**, v. 31, 2018.

ROBINSON, P. M. **Time series with long memory**. [S.l.]: Advanced Texts in Econometrics, 2003.

SALKAR, T.; SHINDE, A.; TAMHANKAR, N.; BHAGAT, N. Algorithmic trading using technical indicators. In: IEEE. **2021 International Conference on Communication information and Computing Technology (ICCICT)**. [S.l.], 2021. p. 1–6.

SALLES, R.; BELLOZE, K.; PORTO, F.; GONZALEZ, P. H.; OGASAWARA, E. Nonstationary time series transformation methods: An experimental review. **Knowledge-Based Systems**, Elsevier, v. 164, p. 274–291, 2019.

SHARPE, W. F. Mutual fund performance. **The Journal of business**, JSTOR, v. 39, n. 1, p. 119–138, 1966.

SIMON, P. **Too big to ignore: the business case for big data**. [S.l.]: John Wiley & Sons, 2013. v. 72.

SUGIYAMA, M.; KAWANABE, M. **Machine learning in non-stationary environments: Introduction to covariate shift adaptation**. [S.l.]: MIT press, 2012.

TIMOFEEV, R. Classification and regression trees (cart) theory and applications. **Humboldt University, Berlin**, v. 54, 2004.

TSAI, C.-F.; LIN, Y.-C.; YEN, D. C.; CHEN, Y.-M. Predicting stock returns by classifier ensembles. **Applied Soft Computing**, Elsevier, v. 11, n. 2, p. 2452–2459, 2011.

TSAY, R. S. **Analysis of financial time series**. [S.l.]: John wiley & sons, 2005.

WANG, J.; SUN, T.; LIU, B.; CAO, Y.; WANG, D. Financial markets prediction with deep learning. In: IEEE. **2018 17th IEEE international conference on machine learning and applications (ICMLA)**. [S.l.], 2018. p. 97–104.

WANG, Z.; HU, Z.; LI, F.; HO, S.-B.; CAMBRIA, E. Learning-based stock trending prediction by incorporating technical indicators and social media sentiment. **Cognitive Computation**, Springer, v. 15, n. 3, p. 1092–1102, 2023.

WANG, Z.; ZHENG, W. **High-frequency trading and probability theory**. [S.l.]: World Scientific, 2014. v. 1.

WHITE, H. Economic prediction using neural networks: The case of ibm daily stock returns. In: **ICNN**. [S.l.: s.n.], 1988. v. 2, p. 451–458.

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