UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ECONOMIA

Roberto Peixoto Fonseca

Predição utilizando a regressão logística com indicadores técnicos como regressores no ETF PIBB11 (período de 2006-2024).

JUIZ DE FORA - MG
2025

Roberto Peixoto Fonseca

Predição utilizando a regressão logística com indicadores técnicos como regressores no ETF PIBB11 (período de 2006-2024).

Monografia apresentada ao curso de Ciências Econômicas da Universidade Federal de Juiz de Fora, como requisito parcial à obtenção do título de bacharel em Ciências Econômicas.

Orientador: Paulo César Coimbra Lisbôa

JUIZ DE FORA - MG
2025

Ficha catalográfica elaborada através do programa de geração
automática da Biblioteca Universitária da UFJF,
com os dados fornecidos pelo(a) autor(a)

Fonseca, Roberto Peixoto.
Predição utilizando a regressão logística com indicadores técnicos como regressores no ETF PIBB11 (período de 2006-2024). / Roberto Peixoto Fonseca. – 2025.
58 p.

Orientador: Paulo César Coimbra Lisbôa
Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Juiz de Fora, Faculdade de Economia, 2025.

1. Regressão logística. 2. Previsão. 3. Indicadores técnicos. 4. PIBB11. 5. Validação cruzada. I. Lisbôa, Paulo César Coimbra, orient. II. Título.

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
REITORIA - FACECON - Depto. de Economia

FACULDADE DE ECONOMIA / UFJF

ATA DE APROVAÇÃO DE MONOGRAFIA II

NA DATA DE 13/03/2025, A BANCA EXAMINADORA, COMPOSTA PELOS PROFESSORES:

1 – PAULO CÉSAR COIMBRA LISBÔA - ORIENTADOR; E

2 – EDUARDO SIMÕES DE ALMEIDA,

REUNIU-SE PARA AVALIAR A MONOGRAFIA DO ACADÊMICO ROBERTO PEIXOTO FONSECA, INTITULADA:

PREDIÇÃO UTILIZANDO REGRESSÃO LOGÍSTICA COM INDICADORES TÉCNICOS COMO REGRESSORES NO ETF PIBB11 (PERÍODO DE 2006-2024).

APÓS PRIMEIRA AVALIAÇÃO, RESOLVEU A BANCA SUGERIR ALTERAÇÕES AO TEXTO APRESENTADO, CONFORME RELATÓRIO SINTETIZADO PELO ORIENTADOR. A BANCA, DELEGANDO AO ORIENTADOR A OBSERVÂNCIA DAS ALTERAÇÕES PROPOSTAS, RESOLVEU APROVAR A REFERIDA MONOGRAFIA.

Documento assinado eletronicamente por **Paulo César Coimbra Lisbôa**, **Professor(a)**, em 17/03/2025, às 16:41, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Eduardo Simoes de Almeida**, **Professor(a)**, em 18/03/2025, às 12:12, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **2296835** e o código CRC **B10175BE**.

RESUMO

Este trabalho utiliza a regressão logística para realizar previsões do ETF PIBB11 no período de janeiro de 2022 a dezembro de 2024, com base em dados de treinamento de 2006 a 2021, utilizando a linguagem de programação Python. As previsões são feitas com base em indicadores técnicos, que atuam como regressores para o modelo. Os indicadores técnicos escolhidos foram: média móvel simples, média móvel exponencial, convergência e divergência de média móvel (MACD), bandas de Bollinger, oscilador estocástico e índice de força relativa. O modelo foi otimizado por meio do algoritmo GridSearchCV, que automatiza a busca pelos melhores hiperparâmetros, e também por meio de métricas como acurácia, precisão, revocação e escore F1. A validação do modelo foi realizada utilizando a técnica de validação cruzada TimeSeriesSplit. Os resultados indicaram que os modelos otimizados para revocação e escore F1 apresentaram os melhores desempenhos, superando ligeiramente a estratégia de comprar e manter o ativo em termos de retorno acumulado no período analisado. Já o modelo otimizado para precisão e acurácia obteve desempenho inferior à estratégia de comprar e manter o ativo. Este estudo contribui para a literatura ao demonstrar a viabilidade da aplicação da regressão logística com indicadores técnicos, além da otimização de hiperparâmetros, na previsão do movimento do ETF PIBB11.

Palavras-chave: Regressão logística; Previsão; Indicadores técnicos; PIBB11; Validação cruzada.

ABSTRACT

This study applies logistic regression to predict the movement of the PIBB11 ETF from January 2022 to December 2024, using training data from 2006 to 2021, and was implemented in Python. The predictions are based on technical indicators that serve as regressors for the model. The selected technical indicators include: simple moving average, exponential moving average, moving average convergence divergence (MACD), Bollinger Bands, stochastic oscillator, and relative strength index. The model was optimized using the GridSearchCV algorithm to automate the search for optimal hyperparameters and evaluated through metrics such as accuracy, precision, recall, and F1-score. Model validation was performed using the TimeSeriesSplit cross-validation technique. The results indicated that models optimized for recall and F1-score achieved the best performance, slightly outperforming the buying and holding strategy in terms of cumulative return over the analyzed period. In contrast, the model optimized for precision and accuracy underperformed compared to the buying and holding strategy. This study contributes to the literature by demonstrating the viability of applying logistic regression with technical indicators, along with hyperparameter optimization, to predict the movement of the PIBB11 ETF.

Keywords: Logistic regression; Prediction; Technical indicators; PIBB11; Cross-validation.

LISTA DE ABREVIATURAS E SIGLAS

ETFs - *Exchanged traded funds*

SMOTE - *Synthetic Minority Over-sampling Technique*

ML – *Machine Learning*

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Prever a direção do mercado financeiro é uma tarefa desafiadora e crucial para os profissionais da área. Apesar de alguns debates acadêmicos sugerirem que a direção dos preços é aleatória, como na Hipótese do Passeio Aleatório, baseada na Hipótese do Mercado Eficiente, que postula que os preços oscilam em torno de seu valor intrínseco, há autores que argumentam que os mercados não são um sistema totalmente aleatório. Murphy (1999), por exemplo, questiona a aleatoriedade do mercado ao citar as tendências que os mercados em baixa apresentam.

Os profissionais do mercado financeiro que estudam, analisam e tentam prever os movimentos do mercado podem ser divididos em dois principais grupos: os fundamentalistas e os técnicos. Segundo Murphy (1999), os fundamentalistas estudam a causa dos movimentos do mercado, enquanto os técnicos analisam o efeito. Os indicadores técnicos fazem parte da análise técnica, e alguns dos mais conhecidos são: médias móveis, bandas de bollinger, osciladores estocásticos e índice de força relativa, entre muitos outros. Realizar previsões no mercado financeiro não é algo novo, os próprios indicadores técnicos dão suporte à tomada de decisão dos investidores e são utilizados para tentar prever o próximo movimento do mercado.

A convergência entre a abordagem tradicional, que utiliza indicadores técnicos para prever os movimentos do mercado, e as técnicas modernas de aprendizado de máquina tem mostrado resultados promissores no mercado financeiro. Enquanto os profissionais do mercado financeiro, como os técnicos, se apoiam em indicadores como médias móveis, bandas de bollinger, osciladores estocásticos, índice de força relativa e outros indicadores para fundamentar suas decisões, estudos recentes têm demonstrado que a aplicação de algoritmos de aprendizado de máquina pode potencializar essas previsões. Para evidenciar boas previsões utilizando modelos de aprendizado de máquina, Faria *et al*. (2008) utilizaram redes neurais artificiais para prever movimentos no mercado de ações, alcançando métricas de previsão satisfatórias. Da mesma forma, Shah *et al*. (2023) combinando indicadores técnicos com técnicas de aprendizado de máquina, aplicaram o algoritmo XGBoost e obtiveram um lucro médio por operação de aproximadamente 4,60% ao ano entre 2011 e 2021.

Este estudo utilizará indicadores técnicos em conjunto com a regressão logística. A regressão logística é uma técnica estatística e um algoritmo de aprendizado de máquina amplamente utilizado para modelar e prever a probabilidade de um evento binário, como a alta ou a queda do preço de uma ação (ou a sinalização de compra e venda) no mercado financeiro. Esse método estima a probabilidade de ocorrência de um evento, permitindo a classificação dos

dados. Neste estudo, a regressão logística será aplicada para prever a direção dos preços do ETF PIBB11, utilizando indicadores técnicos como regressores.

Além disso, o modelo desenvolvido utilizando a regressão logística, juntamente com os indicadores técnicos como regressores, se beneficiará de outras técnicas de aprendizado de máquina, como a validação cruzada e a busca por melhores hiperparâmetros, para otimizar e alcançar melhores resultados. A validação cruzada é útil porque permite que o modelo seja avaliado de forma mais robusta, garantindo que ele seja generalizável para novos dados e menos propenso a sobreajustes (NETTO; MACIEL, 2021)

A técnica de validação cruzada utilizada será a TimeSeriesSplit, disponível na biblioteca scikit-learn da linguagem Python. Essa técnica divide o conjunto de dados de uma série temporal em um número escolhido de K *splits* (PEDREGOSA *et al.*, 2011). Em cada iteração, os primeiros K-1 *splits* são usados para treinar o modelo, enquanto o K-ésimo *split* é usado para testá-lo.

O modelo será desenvolvido utilizando métricas como acurácia, precisão, revocação e o escore F1, que são medidas para verificar o desempenho do modelo, obtidas através do número de vezes que ele acertou ou errou uma previsão (GERÓN, 2021). Neste caso específico, essas métricas serão utilizadas para otimizar ainda mais o modelo. Por exemplo, um modelo otimizado para obter a melhor revocação pode ter resultados melhores do que um modelo otimizado para obter a melhor precisão, pois a revocação prioriza a identificação de todas as instâncias positivas, reduzindo falsos negativos, enquanto a precisão foca em minimizar falsos positivos. O algoritmo que realizará a automação pela busca dos melhores hiperparâmetros será o GridSearchCV.

Consequentemente, os modelos otimizados por técnicas de aprendizado de máquina serão testados, comparando seus retornos acumulados entre si e com o de uma estratégia simples, como a estratégia de comprar e manter o ativo, que servirá como o *benchmark* deste estudo. O objetivo é que o modelo supere essa estratégia básica.

De acordo com Pinto (2024), os ETFs (*Exchange Traded Funds*, ou Fundos Negociados em Bolsa, na tradução livre) são fundos de investimento cujas cotas são negociadas nas bolsas de valores, permitindo que os investidores adquiram uma carteira diversificada de ativos com facilidade e eficiência. O PIBB11 em específico é um ETF que reflete a performance do Índice Brasil 50, também conhecida como IBrX 50. O IBrX 50 é um índice de retorno total com foco no desempenho dos 50 ativos com maior representatividade no mercado de ações brasileiro. Este trabalho então tem como objetivo principal o desenvolvimento de um modelo preditivo para prever o movimento dos preços do ETF PIBB11, utilizando a regressão logística e

indicadores técnicos como regressores A hipótese de pesquisa é que os ajustes específicos nos hiperparâmetros do modelo de regressão logística, como a escolha da penalização, o valor do hiperparâmetro C, o número de iterações e o solver, combinados com a validação cruzada, resultarão em retornos significativamente maiores para o ETF PIBB11 em comparação com a estratégia de comprar e manter o ativo.

Esta monografia está dividida em nove seções, sendo a primeira a introdução. Nas Seções 2, 3, 4 e 5 são apresentados, respectivamente, os objetivos, a pergunta da pesquisa, a hipótese da pesquisa e a justificativa. Na Seção 6, é apresentado o referencial teórico. Na Seção 7, é detalhada a metodologia implementada no projeto de pesquisa. Na Seção 8, discutem-se os resultados obtidos. Por fim, na Seção 9, são apresentados as conclusões e os desafios para o futuro. Após as referências, o código em Python utilizado é apresentado em um apêndice.

2 OBJETIVOS

2.1 Objetivo geral

Desenvolver um modelo preditivo do movimento dos preços do ETF PIBB11 no período de 2022 a 2024, com base em dados treinados de 2006 a 2021, utilizando regressão logística e indicadores técnicos.

2.2 Objetivos específicos

O trabalho tem como objetivos específicos os pontos:

i. Coletar os dados históricos do ETF PIBB11 utilizando a plataforma investing.com;

ii. Obter os indicadores técnicos selecionados utilizando a biblioteca TA (Technical Analysis);

iii. Treinar e validar o modelo de regressão logística através de técnicas de validação cruzada como o TimeSeriesSplit para que o modelo generalize bem para todo o conjunto de dados;

iv. Selecionar um espaço de hiperparâmetros dos modelos de regressão logística e automatizar com o GridSearchCV;

v. Utilizar métricas como revocação, escore F1, precisão e acurácia como critérios de avaliação no GridSearchCV;

vi. Analisar o desempenho dos modelos gerado através de um comparativo entre o retorno acumulado dos modelos contra o retorno acumulado da estratégia de comprar e manter o ativo

vii. Elaborar um relatório final para avaliar e comparar o desempenho dos modelos após os ajustes tendo como base os retornos acumulados, visando identificar quais que foram os modelos que obtiveram os melhores resultados.

3 PERGUNTA DA PESQUISA

Quais são os ajustes específicos nos hiperparâmetros do modelo de regressão logística, utilizando indicadores técnicos como regressores e a validação cruzada, que podem otimizar os retornos do ETF PIBB11 e superar a estratégia de comprar e manter o ativo em termos de rentabilidade?

4 HIPÓTESE DA PESQUISA

Ajustes específicos nos hiperparâmetros do modelo de regressão logística, como a escolha da regularização, o valor do hiperparâmetro C, o número de iterações e o solver, combinados com a validação cruzada, resultarão em retornos significativamente maiores para o ETF PIBB11 em comparação com a estratégia de comprar e manter o ativo.

5 JUSTIFICATIVA

Esta pesquisa é relevante devido à crescente complexidade do mercado financeiro, que surge de fatores como o crescimento e a diversificação de produtos, como os próprios ETFs, além do avanço tecnológico, que acelerou as transações eletrônicas e expandiu os mercados internacionais.

Para evidenciar a crescente complexidade do mercado financeiro, o livro "*Capital Ideas: The Improbable Origins of Modern Wall Street*" (Ideias de Capital: As Origens Improváveis de Wall Street Moderna em português na tradução livre), de Bernstein (2005), destaca que até mesmo os investidores individuais, utilizando poderosos computadores e técnicas quantitativas,

transformaram a análise de investimento. Isso a tornou mais sofisticada e acessível, algo impensável há quase 70 anos. Bernstein afirma que muitos indivíduos lucraram significativamente devido aos novos instrumentos financeiros e estratégias de investimento que surgiram com a revolução nas finanças.

Atualmente, grandes instituições financeiras já utilizam técnicas orientadas a dados para tomadas de decisões no mercado financeiro, como o aprendizado de máquina. A BlackRock, uma companhia norte-americana de investimentos, possui estudos publicados sobre aprendizado de máquina na gestão de ativos e utiliza a plataforma Aladdin para ajudar gestores a tomar decisões. Similarmente, a Goldman Sachs, através da plataforma Marquee, utiliza técnicas de aprendizado de máquina para aprimorar a seleção de riscos e a gestão de ativos. Ambas as plataformas podem ser acessadas via web.

Portanto esta pesquisa pode oferecer contribuições significativas para profissionais do mercado financeiro, pesquisadores e acadêmicos, permitindo o aprimoramento de estratégias e técnicas de previsão e tomada de decisão. Compreender como os indicadores técnicos e o desenvolvimento do modelo de regressão logística influenciam a precisão das previsões pode ter implicações práticas relevantes para investidores e gestores.

6 REFERÊNCIAL TEÓRICO

6.1 Mercado Financeiro e Previsão de Ações

Securato, Securato e Veiga (2009, p. 23) definem o mercado financeiro da seguinte forma:

> Pode-se chamar de mercado financeiro ou bancário o conjunto de instituições e operações ocupadas com o fluxo de recursos monetários entre os agentes econômicos. Basicamente, é o mercado de emprestadores e tomadores de empréstimos, sendo que o valor da remuneração desses empréstimos é chamado de juros ou, em termos percentuais, de taxa de juros. Essa taxa representa, em dado período, a remuneração relativa que os emprestadores obterão e o custo relativo com que os tomadores de empréstimos terão de arcar.

Ainda de acordo com Securato, Securato e Veiga (2009, p. 23), as instituições que desempenham a função de criação e a manutenção do mercado financeiro são chamados de intermediários financeiros. Bancos e corretas são exemplos de intermediários financeiros.

Os intermediários financeiros desenvolvem-se na economia por meio de quatro principais mercados: monetário, crédito, capital e cambial.

Segundo Neto (2021), o mercado monetário tem como principal função o controle da liquidez da economia e das taxas de juros definidas pelas autoridades monetárias. Nesse ambiente, são negociados títulos emitidos pelo Tesouro Nacional, como as Notas do Tesouro Nacional (NTN) e as Letras do Tesouro Nacional (LTN), destinados ao financiamento do orçamento público. Em relação ao mercado de crédito, Neto (2021) afirma que seu objetivo é fornecer recursos, como financiamento, para atender às demandas de curto e médio prazo de diferentes agentes econômicos. Esse financiamento pode ocorrer por meio da concessão de crédito e pela oferta de empréstimo e financiamento. Em relação ao mercado de capitais, Neto (2021) argumenta que esse mercado desempenha um dos papéis mais importantes na economia ao viabilizar transferências de recursos para os agentes que necessitam de capital de longo prazo. Neste mercado há diversas modalidades de financiamento. Os ETFs, por exemplo, objeto de estudo deste trabalho, são fundos negociados no mercado de capitais, que replicam o desempenho de um índice de ações ou de outros ativos. Por fim, no mercado cambial, segundo Neto (2021), ocorrem transações de compra e venda de moedas internacionais.

Os ETFs, um dos principais ativos negociados no mercado de capitais e objeto de estudo deste trabalho, apresentam desafios significativos no que diz respeito à previsão de seus preços. Existe um amplo debate sobre a possibilidade de prever o comportamento do mercado financeiro. Diversos autores defendem que os preços seguem um padrão aleatório, tornando as previsões de curto prazo inviáveis. Um exemplo disso é Malkiel (1973), que, em seu livro "*A Random Walk Down Wall Street – The Time–Tested Strategy for Successful Investing*", introduz a teoria do Passeio Aleatório. Segundo essa teoria, os preços dos ativos se movimentam de maneira imprevisível, sem que seja possível determinar suas direções futuras com base no histórico dos dados.

Por outro lado, há autores que contestam essa visão, argumentando que os mercados não são completamente aleatórios. Murphy (1999), por exemplo, questiona essa hipótese ao destacar fenômenos como a persistência de tendências. Ele argumenta que, se os preços fossem totalmente aleatórios, não haveria uma explicação plausível para a continuidade de determinadas tendências ao longo do tempo. Esse debate evidencia a complexidade do comportamento do mercado de ações e a diversidade de abordagens utilizadas para analisá-lo e prevê-lo.

Apesar do extenso debate sobre os desafios na previsão dos preços de ativos, existem duas grandes abordagens utilizadas para tentar prever o mercado financeiro: a abordagem fundamentalista e a abordagem técnica.

De acordo com Murphy (1999), a abordagem técnica é o estudo da ação do mercado por meio do uso de gráficos, com o objetivo de prever as tendências futuras de preços. Na abordagem técnica, existem diversos indicadores, como, por exemplo, as médias móveis. Já a abordagem fundamentalista examina as forças econômicas de oferta e demanda que influenciam a variação dos preços. Essa abordagem considera todos os fatores relevantes para determinar o valor intrínseco de um ativo e, a partir desse valor intrínseco, informações sobre compra e venda podem ser geradas.

Além da análise técnica e fundamentalista, técnicas estatísticas e de aprendizado de máquina também são empregadas para tentar realizar previsões no mercado financeiro, inclusive utilizando a própria análise técnica e fundamentalista como variáveis nos modelos. Para exemplificar, pode-se citar o estudo desenvolvido por Saud e Shakya (2024), no qual uma série de modelos de aprendizado de máquina é utilizada juntamente com indicadores técnicos.

6.2 Regressão logística e a modelagem

Segundo Favero e Belfiore (2017), a regressão logística é uma técnica estatística utilizada para descrever a relação entre uma variável dependente binária e variáveis independentes (também conhecidas como regressores), que podem ser métricas ou não métricas. A regressão logística é particularmente valiosa quando o objetivo da análise é modelar e compreender a relação entre uma variável dependente com duas categorias possíveis, como, por exemplo, “sim” ou “não”. Essa técnica é projetada para lidar com problemas de classificação binária, sendo amplamente empregada em contextos onde a resposta desejada é uma variável dicotômica.

Segundo Hastie *et al*. (2009), um modelo de regressão linear poderia ser utilizado para estimar a probabilidade de uma variável binária. No entanto, essa abordagem apresenta um problema conceitual, pois não há garantia de que as previsões permaneçam dentro do intervalo de 0 a 1, o que pode resultar em probabilidades negativas ou superiores a 1. Para contornar essa limitação, os autores destacam o uso da regressão logística, que emprega a função logística para garantir que todas as previsões estejam no intervalo entre 0 e 1. A função logística pode ser definida como:

$$P_i = \frac{1}{1 + e^{-z_i}} \text{ (6.1)}$$

Onde Z_i é definido da seguinte forma

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \beta_n X_n \text{ (6.2).}$$

De acordo com Gomes (2024), a saída da função logística é uma probabilidade P(Y = 1) e a relação entre a probabilidade e os regressores pode ser expressa através da fórmula:

$$\log(\frac{P(Y=1)}{1-(P(Y=1)} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \beta_n X_n \text{ (6.3)}$$

Ainda, para estimar os coeficientes da regressão como o $\beta_{0,}, \beta_1, \ldots, \beta_n$, geralmente é utilizado o método da máxima verossimilhança. De acordo com Hastie *et al*. (2009), a abordagem da máxima verossimilhança busca estimar tais coeficientes de modo a maximizar a probabilidade de se observar os dados reais. Isso implica ajustar os coeficientes do modelo de forma que, por exemplo, as previsões de probabilidade do modelo sejam próximas de 1 para os casos em que o mercado suba e próximas de 0 para casos que o mercado caia. De acordo com Gomes (2024), a função de verossimilhança para um conjunto de dados com n observações é dada por:

$$L(\beta) = \prod_{i=1}^{n} P(y_i)^{y_i}(1 - P(y_i))^{1-y_i}$$

Onde

y_i é o valor observado, como por exemplo, 1 para o mercado subir e 0 para o mercado cair

$P(y_i)$ é a probabilidade predita.

A regressão logística tem sido amplamente utilizada em pesquisas na área da saúde, ciências sociais e ciências exatas. De acordo com Netto e Maciel (2021), alguns exemplos de aplicação incluem: determinar se um cliente comprou ou não um determinado produto, identificar se uma transação com cartão de crédito é legítima ou fraudulenta e diagnosticar se um paciente tem diabetes. Sua ampla aplicabilidade a problemas reais a torna uma das técnicas mais valiosas para análise preditiva em diferentes setores.

6.3 Indicadores técnicos com aprendizado de máquina

O estudo que faz a integração do mercado financeiro com o aprendizado de maquinas é algo novo, mas que vem sendo bastante estudado. Gao *et al*. (2024) em seu estudo de revisão de literatura de modelos de aprendizado de máquina aplicado a finanças, demonstra que técnicas

de aprendizado de máquina apresentam um potencial substancial para a tomada de decisões no mercado financeiro. Ainda de acordo com Gao *et al* (2024), quando se pesquisa por publicações cientificas que relaciona o campo financeiro com o campo de aprendizado de máquinas, uma grande parte está concentrada no mercado de ações. Silva (2024, p. 20), em sua dissertação de mestrado, acrescenta que:

> [...] O debate sobre a eficácia da análise técnica versus ML em estratégias de trading é contínuo. Muitos estudos têm mostrado que a combinação de ambas as abordagens pode ser vantajosa, aproveitando tanto padrões históricos quanto análises preditivas para tomada de decisão (Neely *et al.*, 1997; Ready, 2002). Enquanto a análise técnica se concentra em identificar padrões recorrentes nos preços e volumes históricos, o ML oferece a capacidade de descobrir relações não lineares e interações complexas nos dados que não são facilmente detectáveis por métodos tradicionais. Estudos recentes sugerem que a integração de ML com análise técnica pode melhorar significativamente a precisão preditiva e a robustez das estratégias de trading (Hu, Liu, & Zhang, 2020).

Na área que combina então os indicadores técnicos com técnicas de aprendizado de máquinas já surgem grandes estudos, como o estudo de Saud e Shakya (2024), que integram a análise técnica com algoritmos de aprendizado de máquina para prever sinais de negociação. Neste estudo, Saud e Shakya (2024) conduzem uma série de experimentos para aprimorar a previsão do mercado de ações utilizando indicadores técnicos para gerar sinais de compra e venda, alcançando uma performance significativamente melhor do que as estratégias tradicionais. Também é possível citar outro grande estudo, realizado por Aguirre, Medina e Medina (2021), que utiliza a variável convergência e divergência de média móvel juntamente com modelos de algoritmos genéticos para a previsão do índice NASDAQ, alcançando resultados superiores aos das técnicas tradicionais, como o *buy and hold*

6.4 Trabalhos relacionados

Modelos como a regressão logística têm sido amplamente utilizados no campo de previsões, especialmente no mercado financeiro. Além disso, o uso de indicadores técnicos é frequentemente empregado para capturar padrões nos dados e auxiliar na tomada de decisões. Esta seção revisa alguns trabalhos anteriores que utilizaram a regressão logística para previsões no mercado financeiro, incorporando indicadores técnicos.

Um bom trabalho relacionado é o estudo de Jiang, Hu e Jia (2022), no qual foi utilizado um modelo de regressão logística penalizada com indicadores técnicos para prever tendências de alta e baixa. Os autores desenvolveram cinco variações penalizadas da regressão logística e demonstraram que essas abordagens aprimoram as métricas de avaliação dos modelos de classificação, evidenciando que o uso de hiperparâmetros, como a penalização, contribui para um desempenho preditivo superior.

Também se pode citar o estudo de Ayyildz e Iskenderoglu (2024), que realiza uma avaliação da eficácia dos modelos de aprendizado de máquina para fazer previsões no mercado de ações. Os autores utilizam diversos indicadores técnicos, como média móvel simples, oscilador estocástico, índice de força relativa, convergência e divergência de médias móveis, entre outros. Os algoritmos empregados incluem: árvore de decisão, floresta aleatória, K-vizinhos mais próximos, naive bayes, regressão logística, máquinas de vetores de suporte e redes neurais artificiais. A regressão logística, em particular, atingiu taxas de acurácia superiores a 70%, sendo mais eficaz para alguns tipos de índices, enquanto as redes neurais artificiais se mostraram melhores para outros. Os autores atribuem essa diferença às características específicas de cada mercado, como a volatilidade. Eles concluem que os investidores podem otimizar suas estratégias com o auxílio dessas técnicas. No entanto, os autores também apontam algumas limitações no estudo, como a ausência de testes com muitos algoritmos amplamente utilizados, o período de análise limitado, a falta de dados macroeconômicos e a possível exclusão de alguns indicadores técnicos importantes, o que pode ter impactado os resultados.

7 METODOLOGIA DE PESQUISA

7.1 Abordagem da pesquisa

Este estudo adota uma abordagem predominantemente empírica, visando realizar previsões no mercado financeiro, com enfoque específico no ETF PIBB11. A pesquisa será de natureza quantitativa, fundamentada na previsão de dados históricos do ETF PIBB11. O alcance da pesquisa será preditivo, uma vez que busca realizar previsões no mercado financeiro.

A escolha do ETF PIBB11 se deve à sua composição e importância histórica. Em relação à sua composição, o PIBB11 reúne os 50 ativos mais negociados no mercado de ações brasileiro. Já sua relevância histórica se dá pelo fato de ter sido o primeiro ETF lançado no Brasil. O período do primeiro dia útil de janeiro de 2006 ao último dia útil de dezembro de 2024

foi selecionado por cobrir os últimos 18 anos da economia brasileira, oferecendo uma grande quantidade de dados para o treinamento e o teste do modelo. Esse período inclui crises econômicas, como a de 2008, recessões e eventos extraordinários, como a crise sanitária e humanitária da pandemia de coronavírus em 2020. Assim, o modelo será treinado em diversos contextos econômicos, visando aprender com os padrões dos dados

No que diz respeito à revisão bibliográfica, a metodologia incluirá uma seleção criteriosa de livros que tratam de temas relevantes, como aprendizado de máquina e literatura específica do mercado financeiro. Além disso, será feita uma revisão de artigos científicos, acessados por meio de plataformas online. Essa abordagem visa garantir uma fundamentação teórica sólida e abrangente para embasar as análises realizadas.

7.2 Coleta de dados

A coleta de dados, incluindo o preço de fechamento diário do ETF PIBB11, será realizada por meio da plataforma investing.com, um portal financeiro global disponível em diversos idiomas, que oferece dados e cotações em tempo real. Segundo a própria plataforma, o site conta com mais de 21 milhões de usuários mensais. Para obter a base de dados do ETF PIBB11, basta pesquisar pelo nome do ETF na plataforma. Os dados históricos podem ser acessados selecionando a opção de exibição dos dados históricos e definindo o período desejado, que, neste estudo, compreende do primeiro dia útil de janeiro de 2006 até o último dia útil de dezembro de 2024. Após essa seleção, a base de dados pode ser baixada no formato CSV para o computador e então utilizada na linguagem Python para visualização e desenvolvimento da pesquisa. A escolha dessa plataforma se justifica pela sua confiabilidade e capacidade de fornecer dados históricos detalhados sobre preços de ativos, um elemento essencial para o escopo deste estudo.

7.3 Variável alvo

A variável alvo será criada utilizando o preço de fechamento do ativo escolhido. Quando a variável alvo for igual a 1, indicará que o preço de fechamento daquele dia (t) foi maior do que o preço de fechamento do dia anterior (t-1), ou seja, o mercado experimentou uma valorização; quando a variável alvo for igual a 0, indicará que o preço de fechamento daquele dia (t) foi menor do que o preço de fechamento do dia anterior (t-1), ou seja, o mercado sofreu

uma desvalorização (ZAINI *et al*., 2019). Assim, com essas informações, será possível realizar as previsões da direção do mercado financeiro. A variável alvo (Y) tem a seguinte formulação:

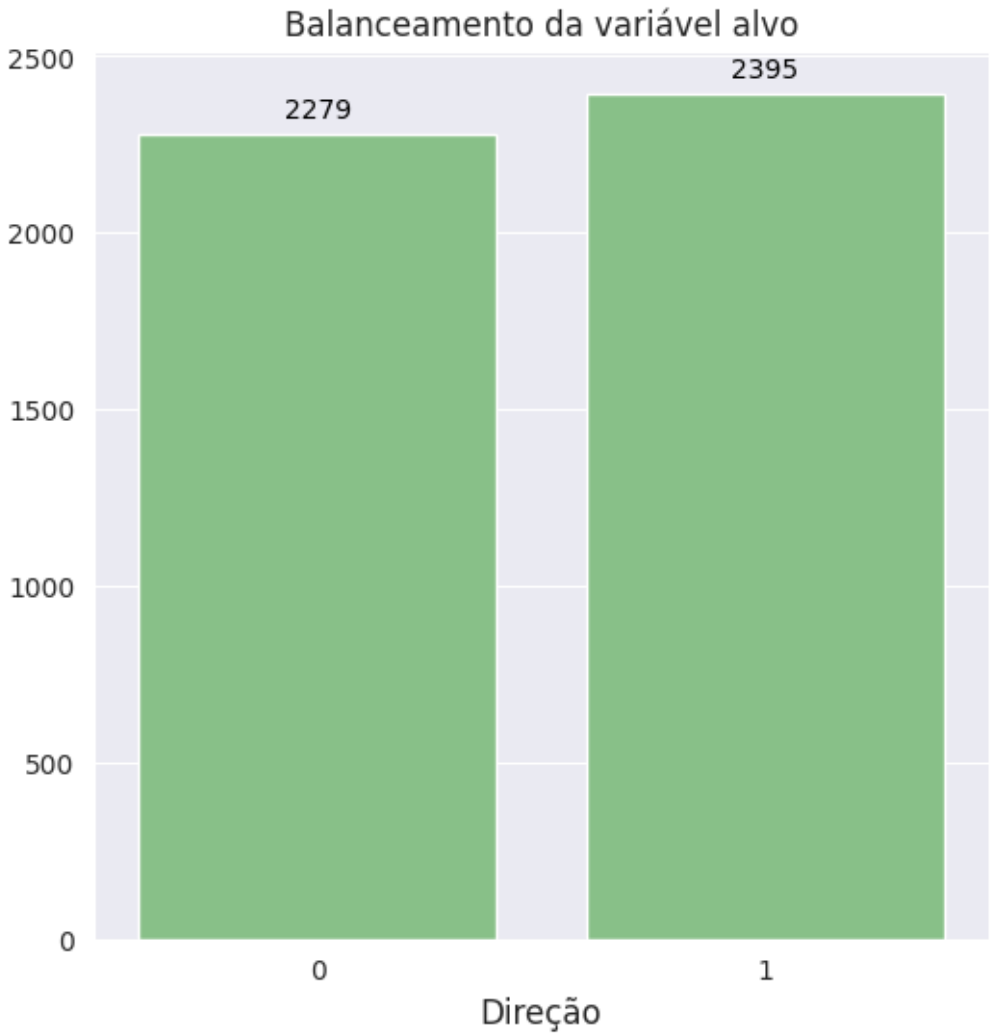
$$Y = \begin{cases} 1, se\ o\ preço\ de\ fechamento\ em\ t\ for\ maior\ do\ que\ em\ \ t-1 \\ 0, se\ o\ preço\ de\ fechamento\ em\ t\ for\ menor\ do\ que\ em\ t-1' \end{cases}$$

Se Y for igual a 1, isso mostrará um movimento de subida. Caso Y seja igual a 0, mostrará um movimento de queda.

Além da criação da variável alvo, em modelos de classificação é necessário verificar como está o seu balanceamento. Dados muito desbalanceados enviesarão as estimativas e o desempenho de modelos de classificação (RHAMAN; WAH; HUAT, 2021). Caso a variável alvo esteja desbalanceada, pode ser necessário utilizar hiperparâmetros disponíveis na linguagem Python que dão suporte para o balanceamento da classe, como o hiperparâmetro class_weight, disponível na biblioteca scikit-learn (PEDREGOSA *et al*., 2011) da linguagem Python e este hiperparâmetro atribui um peso maior à classe desbalanceada. Outra alternativa é utilizar algoritmos disponíveis que realizam balanceamento, como o SMOTE (*Synthetic Minority Over-sampling Technique*) que é amplamente empregado em algoritmos de aprendizado de máquina (CHAWLA *et al*., 2002).

Uma forma de verificar se os dados estão balanceados ou não é utilizando um gráfico de barras. O Gráfico 1 exibe o balanceamento da variável alvo, e nele é possível observar que a direção 0 (movimento de queda) representa 2.279 valores da classe, enquanto a direção 1 (movimento de subida) representa 2.395 valores da classe.

Gráfico 1 - Exibindo o balanceamento da variável alvo



Fonte: Autoria própria.

É possível verificar, então, que os dados estão balanceados previamente (não há discrepância no balanceamento), portanto, nenhuma transformação é necessária neste caso específico.

Além de verificar o balanceamento da variável alvo, é interessante visualizar o comportamento dos preços diários de fechamento ao longo do tempo, para entender como se deu o resultado final do balanceamento. O Gráfico 2 exibe os preços de fechamento do ETF PIBB11 ao longo do período analisado, o que ajuda a entender a tendência dos preços ao longo do tempo.

Gráfico 2 - Exibindo a série histórica

Fonte: Autoria própria

7.4 Modelagem

No contexto do mercado financeiro, a regressão logística será utilizada para prever o movimento do mercado, utilizando indicadores técnicos como regressores. Como este trabalho é típico de problemas de classificação, a regressão logística foi escolhida, pois é amplamente utilizada em problemas de classificação binária, sendo um modelo simples de implementar, entender e com desempenho bastante aceitável (NETTO; MACIEL, 2021).

A regressão logística, um modelo estatístico e de aprendizado de máquina, visa classificar os dados com base em seus regressores. Neste estudo, será utilizada como um modelo de classificação binária (GERÓN, 2021), onde a saída será 1 para indicar uma alta no preço do ETF PIBB11 e 0 para uma queda no preço do ETF PIBB11. A função do modelo de regressão logística é representada pela seguinte fórmula:

$$P_i = \frac{1}{1 + e^{-Z_i}} \quad (7.1)$$

Onde Z_i é representado pela seguinte fórmula:

$$Z_i = B_0 + B_1 X_1 + B_2 X_2 + \cdots + B_n X_n \quad (7.2)$$

E nesta fórmula tem-se:

B_0 que representa o coeficiente do intercepto

$B_1, B_2 \ldots B_n$ que representam os coeficientes do modelo, indicando como a variável alvo muda em resposta a uma unidade de mudança no regressor correspondente

$X_1, X_2 \ldots X_n$ que representam os regressores

Na fórmula da regressão logística, é estabelecido um limiar cujo valor geralmente é de 0,5 (NETTO; MACIEL, 2021). Dada a fórmula da regressão logística, se a probabilidade P_i for maior do que 0,5, então a previsão do modelo será classificada como "1", representando o movimento de subida do mercado. Caso contrário, isto é, P_i for menor do que 0,5, a previsão do modelo será classificada como "0", representando o movimento de queda do mercado.

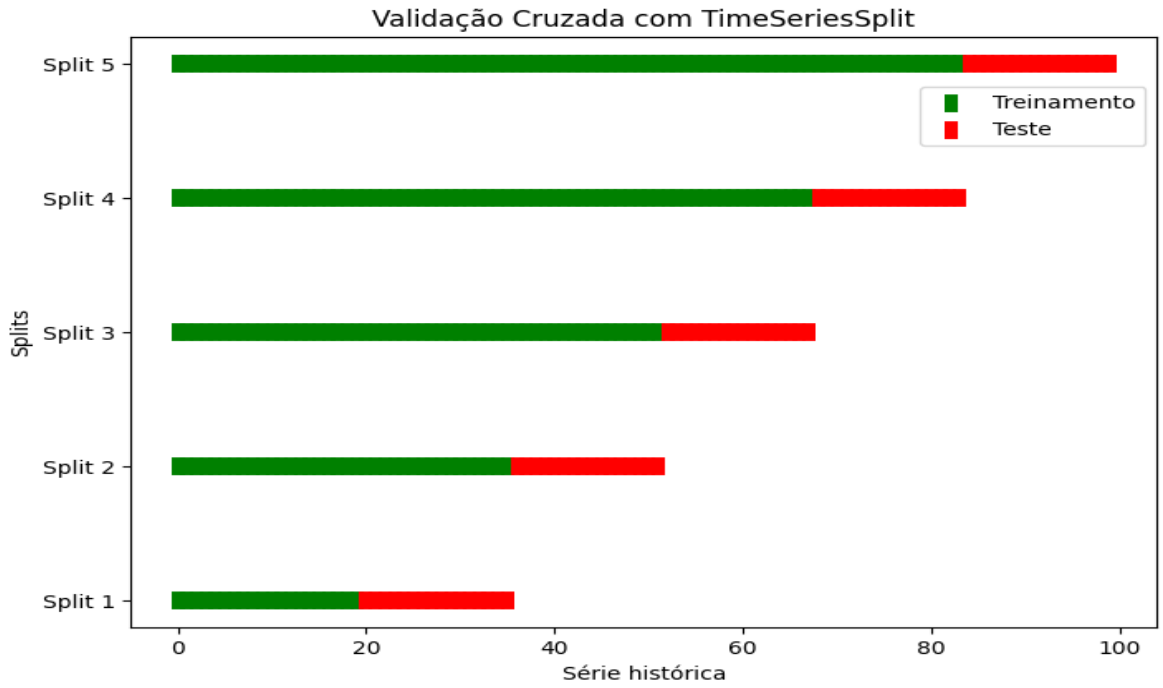
7.5 Validação

De acordo com Netto e Maciel (2021), é comum dividir os dados de um modelo em duas partes: uma para treinar o modelo e outra para testar sua performance. Essa divisão impede que o modelo memorize todos os dados, garantindo que ele não acerte 100% apenas porque viu todos os exemplos durante o treinamento. Se um modelo acertasse 100% nos dados de treinamento, isso poderia indicar que ele não funcionaria bem com dados novos, o que é problemático, pois o ideal é criar um modelo que seja eficaz em prever a direção do mercado, não apenas em ajustar-se perfeitamente aos dados passados. A validação cruzada melhora ainda mais essa abordagem. Em vez de usar uma única divisão dos dados para treino e teste, a validação cruzada divide os dados em múltiplos subgrupos. O modelo é treinado e testado várias vezes, usando diferentes combinações de subgrupos. Isso permite avaliar o desempenho do modelo de forma mais robusta e evita a dependência de uma única divisão dos dados, garantindo que o modelo seja mais generalizável e menos propenso a sobreajuste (NETTO; MACIEL, 2021). Sobreajuste ocorre quando o modelo funciona muito bem para o conjunto de treinamento, mas não generaliza bem para outros conjuntos de dados (GERÓN, 2021).

A validação cruzada escolhida foi o método TimeSeriesSplit, disponibilizado pela biblioteca scikit-learn na linguagem de programação Python (PEDREGOSA *et al*., 2011). A razão para a escolha desse algoritmo se dá pelo fato de os dados estarem dispostos em uma série temporal (preços do ETF PIBB11, do primeiro dia útil de janeiro de 2006 até o último dia útil de dezembro de 2024). Durante a validação cruzada utilizando o TimeSeriesSplit, o conjunto de dados é dividido em um número escolhido de K *splits*, sendo que o número escolhido será K = 5. Em cada iteração da validação cruzada, um dos 5 subconjuntos é designado como conjunto de teste, enquanto os subconjuntos anteriores são utilizados como conjunto de treino.

Essa divisão é realizada sequencialmente, respeitando a ordem temporal dos dados. Cada *split* subsequente avança no tempo em relação ao anterior, garantindo que o conjunto de teste contenha apenas dados futuros em relação ao conjunto de treino. A Figura 1 ilustra este processo:

Figura 1- Validação cruzada com TimeSeriesSplit



Fonte: Autoria própria

A técnica TimeSeriesSplit será utilizada no hiperparâmetro CV do algoritmo GridSearchCV para realizar a validação cruzada em dados dispostos em séries temporais, garantindo que os dados de treino sempre precedam os dados de teste com o objetivo de otimizar o modelo, buscando a melhor performance possível nos retornos do ETF PIBB11

7.6 Automação e otimização

O algoritmo GridSearchCV é uma ferramenta de automação que realiza uma busca exaustiva sobre os hiperparâmetros escolhidos de um modelo (PEDREGOSA *et al.*, 2011). Disponível na biblioteca scikit-learn, ele é amplamente utilizado para otimizar modelos de aprendizado de máquina, ajustando os hiperparâmetros de forma eficiente. O GridSearchCV suporta técnicas de validação cruzada (como explicado anteriormente) e também métricas de

avaliação, como acurácia, precisão, revocação e o escore F1, para obter os melhores resultados de acordo com a característica da métrica escolhida.

Além disso, o GridSearchCV utiliza um param_grid, que é um dicionário ou lista de dicionários que especifica os hiperparâmetros a serem testados. Cada chave no dicionário corresponde a um hiperparâmetro do modelo, e os valores associados são as opções que serão avaliadas. Por exemplo, em um modelo de regressão logística, o param_grid pode incluir diferentes valores para o hiperparâmetro de regularização C ou para o tipo de penalidade (L1 ou L2). O GridSearchCV testa todas as combinações possíveis desses hiperparâmetros, utilizando validação cruzada para garantir que os resultados sejam robustos e generalizáveis.

Ainda, modelos de classificação utilizam métricas como acurácia, precisão, revocação e o escore F1 para avaliar seu desempenho, e essas métricas podem ser empregadas como hiperparâmetros no algoritmo GridSearchCV, possibilitando que o modelo otimize suas previsões com base na métrica especificada. Dessa forma, as métricas mencionadas são integradas ao processo de otimização.

A primeira métrica que será utilizada para otimizar o desempenho do modelo é a métrica de acurácia, que avalia o desempenho global do modelo e tem como objetivo medir a fração de previsões corretas. A acurácia é calculada da seguinte forma (NETTO; MACIEL, 2021):

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (7.3)$$

A segunda métrica que será utilizada para otimizar o desempenho do modelo é a métrica de precisão, que mede a exatidão das previsões positivas (NETTO; MACIEL, 2021). A precisão é calculada da seguinte forma:

$$Precisão = \frac{VP}{VP + FP} \quad (7.4)$$

A terceira métrica que será utilizada para otimizar o desempenho do modelo é a métrica de revocação, que mede a fração de previsões positivas corretamente identificadas (NETTO; MACIEL, 2021). A revocação é calculada da seguinte forma:

$$Revocação = \frac{VP}{VP+FN} \quad (7.5)$$

A quarta métrica que será utilizada para otimizar o desempenho do modelo é a métrica do escore F1, que é uma média harmônica entre a precisão e a revocação (GERÓN, 2021). Devido à sua característica de incorporar tanto a precisão quanto a revocação em seu cálculo, o escore F1 equilibra as duas métricas e, por isso, consegue evitar falsos negativos e minimizar falsos positivos. O escore F1 é calculado da seguinte forma:

$$Escore\ f1 = 2 \times \frac{Precisão \times Revocação}{Precisão + Revocação} \quad (7.6)$$

As métricas de avaliação em modelos de classificação utilizam quatro elementos fundamentais: VP (Verdadeiro Positivo), VN (Verdadeiro Negativo), FP (Falso Positivo) e FN (Falso Negativo). VP refere-se aos casos em que o modelo previu corretamente que o evento positivo ocorreria (por exemplo, uma subida no mercado). VN refere-se aos casos em que o modelo previu corretamente o evento negativo (como uma queda no mercado). FP refere-se ao caso em que o modelo previu erroneamente um evento positivo, enquanto o evento real foi negativo, e FN refere-se ao caso em que o modelo previu erroneamente um evento negativo, enquanto o evento real foi positivo.

Entender como a acurácia, precisão, revocação e escore F1 funcionam é fundamental para compreender não apenas o que elas representam, mas também como refletem o desempenho do modelo em diferentes contextos. A acurácia mede a proporção de previsões corretas em relação ao total, enquanto a precisão indica a confiabilidade das previsões positivas. A revocação, por sua vez, avalia a capacidade do modelo de identificar corretamente os casos positivos, e o escore F1 combina precisão e revocação em uma única métrica, equilibrando ambas. O hiperparâmetro utilizado para otimizar cada uma dessas métricas é o hiperparâmetro “scoring”, disponível no algoritmo GridSearchCV.

Também é possível otimizar o próprio modelo de regressão logística. O hiperparâmetro do GridSearchCV que permite explorar a regressão logística é o param_grid. O param_grid é o espaço de hiperparâmetros disponíveis na regressão logística que serão testados. Embora a regressão logística possua diversos hiperparâmetros, os principais utilizados neste trabalho são penalty, C, solver e max_iter. A escolha desses hiperparâmetros se justifica pela sua importância no controle da complexidade do modelo e no aprimoramento de seu desempenho. O hiperparâmetro penalty define o tipo de regularização a ser aplicada (L1 ou L2), ajudando a evitar o sobreajuste do modelo. O hiperparâmetro C controla a força da regularização, equilibrando o ajuste aos dados de treinamento e a generalização. O solver determina o

algoritmo de otimização do modelo, enquanto o hiperparâmetro max_iter define o número máximo de iterações para o algoritmo convergir, garantindo que o processo de otimização seja eficiente (PEDREGOSA *et al.*, 2011).

O hiperparâmetro penalty utilizará três opções: penalização Lasso (L1), penalização Ridge (L2) e None (nenhuma penalização). As penalizações em uma regressão logística atuam penalizando os coeficientes da regressão. O penalizador Lasso (L1), por exemplo, pode forçar coeficientes redundantes a se tornarem zero. O penalizador Ridge (L2) pode forçar coeficientes redundantes a se aproximarem de zero, mas não os tornará exatamente zero (Hastie *et al.*, 2013). A opção None é quando não há penalização aplicada ao modelo (PEDREGOSA *et al.*, 2011). Essas três opções serão passadas para o algoritmo GridSearchCV.

O hiperparâmetro solver também desempenha um papel importante na otimização do modelo de regressão logística. A biblioteca scikit-learn oferece seis opções de solvers: lbfgs, liblinear, newton-cg, newton-cholesky, sag e saga. Cada um desses solvers possui características específicas, que devem ser consideradas de acordo com as características dos dados e a necessidade do modelo, e é possível encontrá-las em tabelas sumarizadas oferecidas pela própria biblioteca scikit-learn. O solver lbfgs, por exemplo, é adequado para a penalização Ridge e também pode ser utilizado quando não há penalização. No entanto, ele não suporta a penalização Lasso. Além disso, o lbfgs é eficaz para problemas com dados desbalanceados. O liblinear, por sua vez, suporta apenas a penalização Ridge e é capaz de penalizar o intercepto. É especialmente robusto quando os dados não estão escalados, tornando-o útil em muitas situações práticas. Os solvers newton-cg e newton-cholesky compartilham características semelhantes: ambos suportam a penalização Ridge e podem ser usados na ausência de penalização. Ambos também são robustos para dados desbalanceados, mas a principal diferença é que o newton-cg suporta classificação multiclasse, enquanto o newton-cholesky é limitado a problemas binários. Por fim, os solvers sag e saga são recomendados para grandes conjuntos de dados. O sag não suporta a penalização Lasso, enquanto o saga suporta todos os tipos de penalização (L1, L2 e None). Essa flexibilidade torna o saga uma escolha versátil para diferentes tipos de regularização e também em grandes volumes de dados, que é sua principal característica. Na própria documentação da biblioteca scikit-learn, é possível encontrar essas informações sumarizadas em tabelas. A escolha de diferentes solvers no processo de otimização é justificada pela compatibilidade de cada um com tipos específicos de penalização. Cada solver tem características distintas que o tornam adequado para diferentes cenários, como o tipo de penalização, o tamanho do conjunto de dados e a presença de classes desbalanceadas.

O hiperparâmetro C controla a força da regularização, sendo inversamente proporcional à força de penalização (PEDREGOSA *et al.*, 2011). Valores pequenos de C tornam a regularização forte, ou seja, os coeficientes do modelo são fortemente penalizados. Valores grandes de C tornam a regularização fraca, ou seja, os coeficientes do modelo são menos penalizados. Esse hiperparâmetro é útil para controlar o sobreajuste do modelo. Valores comuns para este hiperparâmetro são 0,01, 0,1, 1, 10 e 100. Esses serão os valores utilizados.

O hiperparâmetro max_iter é responsável pelo número de iterações realizadas para que os solvers consigam convergir (PEDREGOSA *et al.*, 2011). Quando um solver converge, significa que ele atingiu uma solução. O valor padrão do max_iter é 100, mas também serão utilizados valores menores para uma convergência mais rápida ou valores maiores caso o modelo demore mais para convergir. Os valores escolhidos, então, serão 50, 100, 150 e 200.

Esses quatro hiperparâmetros serão utilizados para otimizar o modelo de regressão logística, com o intuito de prever melhor a direção do mercado e, consequentemente, melhorar o retorno do ETF PIBB11

7.7 Regressores

A escolha dos indicadores técnicos presentes neste estudo se deu por três grandes motivos: a popularidade, isto é, por serem os indicadores técnicos mais comumente utilizados no mercado financeiro; por serem objetos de estudo no livro de John J. Murphy cujo nome é "*Technical Analysis of the Financial Markets*" (Análise Técnica do Mercado Financeiro em português); e também por suas características.

A criação dos indicadores técnicos se dará pela linguagem de programação Python. Os indicadores técnicos serão criados com o auxílio da biblioteca TA (*technical analysis*), que disponibiliza excelentes funções para a criação dos indicadores, não havendo então necessidade de desenvolver os exaustivos cálculos para cada regressor. Entretanto, para fins didáticos, são apresentados os cálculos para cada indicador técnico no texto a seguir, além de suas características.

7.7.1 Média Móvel Simples

O primeiro indicador técnico presente neste estudo é a média móvel simples. De acordo com Murphy (1999), a média móvel simples é o indicador mais usado e que também é

facilmente testado, sendo base de muitos outros indicadores também. A fórmula da Média Móvel Simples (MMS) é definida como:

$$Média\ Móvel\ Simples = \frac{P_1 + P_2 + \cdots + P_n}{n} \quad (7.7)$$

Onde:

P é o preço do ativo

P_n é o preço do ativo no período n

n é o número de períodos (dias)

Apesar da média móvel simples colocar o mesmo peso para todos os preços, ela é importante para o modelo, pois consegue acompanhar o avanço da tendência (MURPHY, 1999). Será utilizado no trabalho a janela de 20 dias.

7.7.2 Média Móvel Exponencial

O segundo indicador técnico presente neste estudo é a média móvel exponencial. Este indicador também serve como base para o cálculo de outros indicadores técnicos, como a convergência e divergência de média móvel (MACD do inglês).

É vantajoso utilizar a média móvel exponencial como regressor no modelo porque ela atribui mais peso para os preços mais recentes (SAMANT, 2015), algo que a média móvel simples por si só não consegue fazer.

De acordo com Vidotto, Migliato e Zambon (2009), fórmula da média móvel exponencial é definida como:

$$MME = P_t K + MME_{ontem} * (1 - K) \quad (7.8)$$

Em que K representa o fator de suavização, dado por:

$$K = \frac{2}{N + 1}$$

P_t é o preço no período t.

K é o fator de suavização

N é o número de dias da MME

MME_{t-1} é a média móvel exponencial no período t-1.

A MME no trabalho terá uma janela de 20 dias.

7.7.3 Convergência e Divergência de Média Móvel

O terceiro indicador técnico presente neste estudo é o famoso indicador de convergência e divergência de média móvel (também conhecido como MACD do inglês). Convergência e divergência de média móvel é um dos mais simples e mais eficientes indicadores de momentum (SAMANT, 2015), o que será útil na criação do modelo.

Da convergência e divergência de média móvel é possível extrair três informações (SHAH *et al*., 2023):

$$Linha\ CDMM = MME_{12} - MME_{26} \quad (7.9)$$

$$Linha\ de\ Sinal = MME_{9} \quad (7.10)$$

$$Histograma = Linha\ CDMM - Linha\ de\ Sinal \quad (7.11)$$

Para a Convergência e Divergência de Média Móvel, as janelas serão de 12, 26 e 9.

7.7.4 Bandas de Bollinger

O quarto indicador técnico presente neste estudo são as Bandas de Bollinger. De acordo com Williams (2006) em sua dissertação de mestrado, bandas de bollinger são capazes de capturar flutuações súbitas no nível de preços, ou seja, essa característica que a bandas de bollinger oferece é um ganho para o modelo para que ele seja capaz de aprender com as informações que a bandas de bollinger irá disponibilizar e assim aprender a capturar as características de flutuações súbitas no nível de preços. A fórmula para a banda superior e inferior são demonstradas abaixo:

$$Banda\ Superior = MMS + k \times \sqrt{\sum_{i=1}^{n} \frac{(P_i - MMS)^2}{n}} \quad (7.12)$$

$$Banda\ Inferior = MMS - k \times \sqrt{\sum_{i=1}^{n} \frac{(P_i - MMS)^2}{n}} \quad (7.13)$$

Onde

k representa o número de desvios padrão usados para calcular as bandas

P_i são os preços de fechamento nos últimos n períodos.

Será utilizada uma janela de 20 dias para as bandas de bollinger.

7.7.5 Oscilador estocástico

O quinto indicador técnico presente neste estudo é o estocástico. O oscilador estocástico é um indicador que segue o momentum do preço (SAMANT, 2015). De acordo com Murphy (1999), indicadores técnicos como osciladores são úteis principalmente em períodos de mercados instáveis, isto é, os osciladores conseguem rastrear o movimento do preço. Duas linhas são usadas no processo estocástico: %K e %D. A seguir é demonstrado a fórmula para a linha %K:

$$\%K = 100 \times \left(\frac{P_{ult} - L_n}{H_{max} - L_n}\right) \quad (7.14)$$

Onde,

P_{ult} é o preço de fechamento mais recente

H_{max} é o menor preço observado nos últimos n períodos

L_n é o menor preço observado nos últimos n períodos

A linha %D é formada pela média móvel simples de %K com n períodos. Para a linha %K, será utilizada um período de 14 dias, enquanto para a linha %D será uma média móvel de 3 dias.

7.7.6 Índice de Força Relativa.

O índice de força relativa é um dos indicadores técnicos mais populares de momentum. De acordo com Murphy (1999), o índice de força relativa suaviza variações abruptas nos preços e normaliza os resultados em uma escala fixa de 0 a 100 para facilitar a comparação e interpretação do momentum dos preços. De acordo com Shah *et al*. (2023), o índice de força relativa indica forças de compra e venda. É um indicador que agregará valiosamente o modelo. A sua fórmula é dada por:

$$IFR = 100 - \frac{100}{1 + FR} \quad (7.15)$$

Onde FR é igual a média móvel de n dias de fechamento em alta dividido pela média móvel de n dias de fechamento em baixa.

Para o Índice de Força Relativa a janela utilizada será de 14 dias.

7.8 Comprar e manter o ativo como *benchmark*

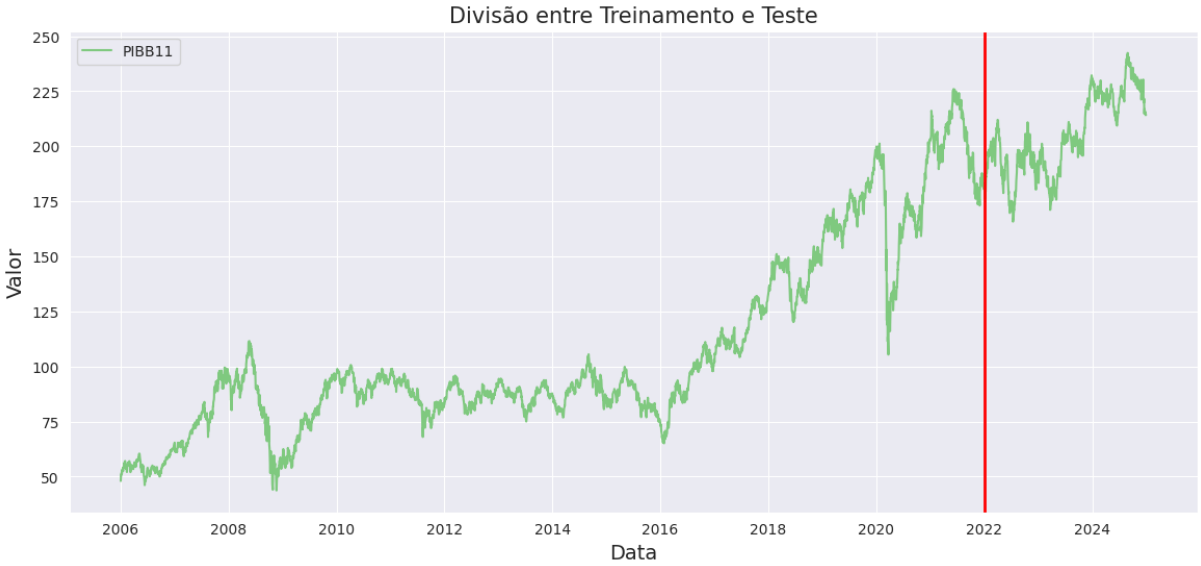
Neste estudo, a estratégia de comprar e manter o ativo será aplicada adquirindo o ETF PIBB11 e avaliando o retorno acumulado ao longo do período analisado. Paralelamente, o modelo de regressão logística será utilizado para gerar sinais de compra e venda, permitindo o cálculo do retorno acumulado das operações baseadas em suas previsões. Dessa forma, será possível comparar a eficácia do modelo em relação à estratégia de comprar e manter, que servirá como *benchmark* desta pesquisa. Considerando que essa estratégia é simples e se baseia apenas na manutenção da posição comprada, espera-se que o modelo preditivo, ao identificar oportunidades mais precisas de entrada e saída do mercado, consiga melhorar o desempenho financeiro em relação a essa abordagem passiva.

8 RESULTADOS

8.1 Treino e teste

Os dados que abrangem o período de 1º de janeiro de 2006 a 31 de dezembro de 2024, foram divididos em um conjunto de treinamento e teste. Após ter criado os regressores, restou então 4674 amostras disponíveis na base de dados e destas amostras, 3922 amostras (aproximadamente 83,91% da base) foram destinadas ao treinamento do modelo, enquanto as 752 amostras restantes (aproximadamente 16,09% da base) restantes foram reservados para teste do modelo. O período compreendido entre o primeiro dia útil de 2006 e o último dia útil de 2021 foi utilizado para treinar o modelo, enquanto o período compreendido entre o primeiro dia útil de 2022 e o último dia útil de 2024 foi utilizado para testar o modelo. O Gráfico 3 ilustra como ficou a separação.

Gráfico 3 - Divisão entre treinamento e teste



Fonte: autoria própria.

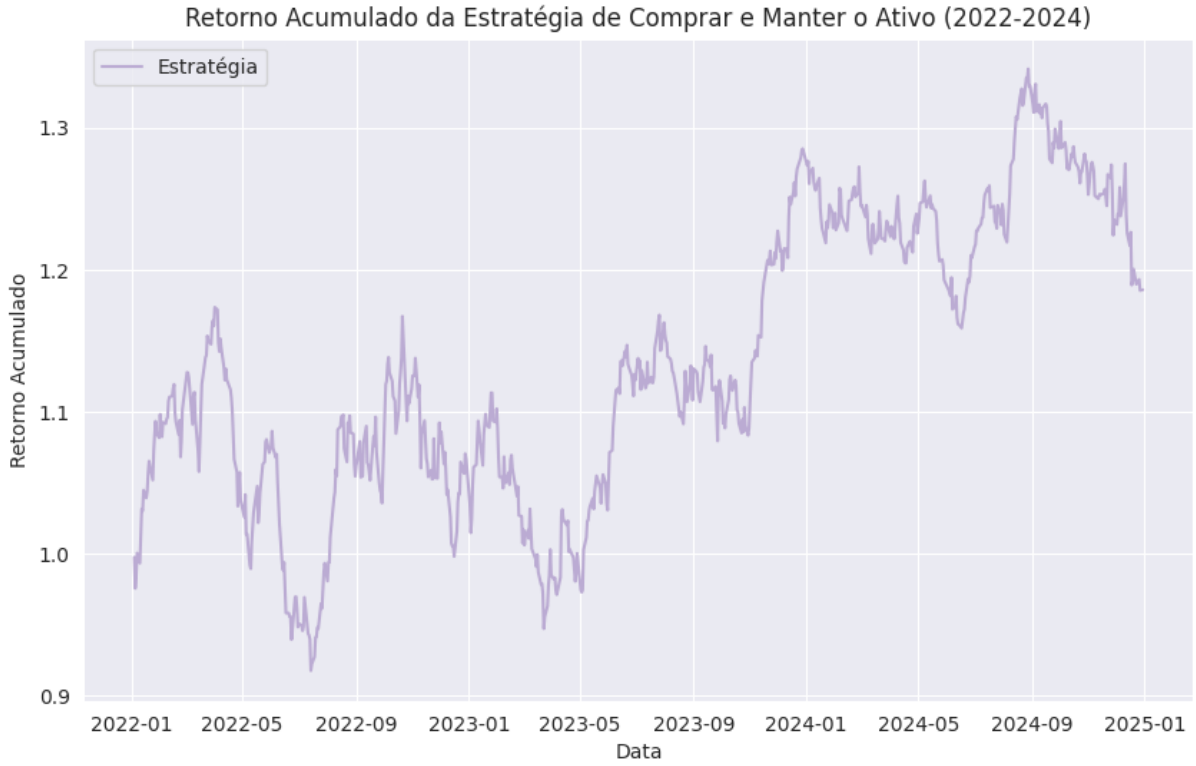
As técnicas como GridSearchCV e a TimeSeriesSplit foram implementadas utilizando somente os dados de treino. Os dados de teste então permanecerão intocáveis, somente utilizados para a validação do modelo.

8.2 Desempenho do modelo

Nesta seção são apresentados os principais resultados obtidos pelo modelo de regressão logística ao prever o movimento dos preços do ETF PIBB11 utilizando a linguagem Python. Os resultados apresentados serão o do retorno acumulado de cada estratégia.

Além disso, há também o retorno acumulado da estratégia de comprar e manter o ativo, que é o *benchmark* do estudo. A estratégia de comprar e manter o ativo obteve um retorno acumulado de 1,18. O Gráfico 4 ilustra como se deu o retorno acumulado desta estratégia.

Gráfico 4- Retorno acumulado da estratégia de comprar e manter o ativo



Fonte: autoria própria

Agora, será testado os modelos contra o retorno acumulado da estratégia de comprar e manter o ativo. O ideal é que os modelos sejam capazes de vencer uma técnica simples como a estratégia de comprar e manter o ativo.

O primeiro modelo avaliado utilizou a revocação como métrica de avaliação. A revocação avalia a capacidade de identificar todas as instâncias positivas presentes nos dados. O GridSearchCV explorou diversas combinações de hiperparâmetros e identificou que os valores ótimos para esta métrica foram: C igual a 0,01, max_iter igual a 150, penalty definida como L1 e o solver como saga. Na Tabela 1 é apresentado os resultados.

Tabela 1 - Resultado da seleção de hiperparâmetros utilizando revocação como critério

Hiperparâmetros	**Valor**
Métrica de avaliação	Revocação
C	0.01
max_iter	150

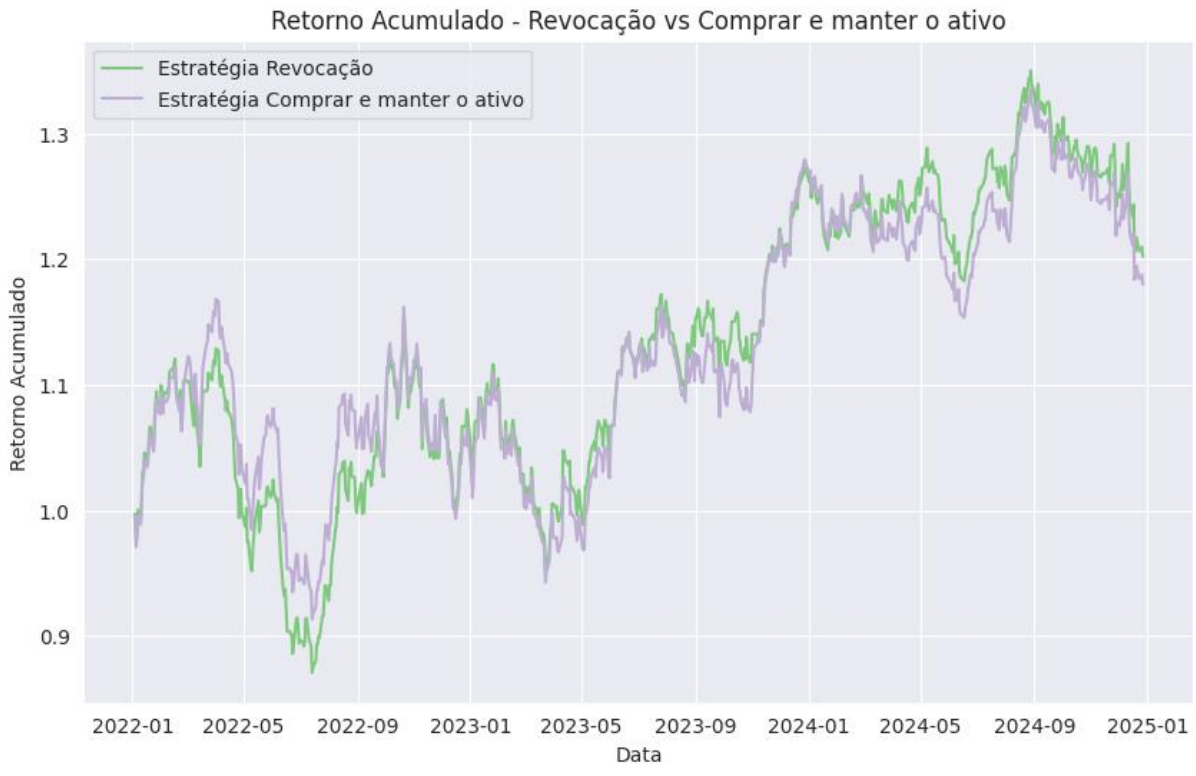
Penalty	L1
Solver	saga

Fonte: autoria própria.

O hiperparâmetro C, que controla a força da regularização do modelo, obteve o valor de 0,01, indicando uma regularização mais forte, o que ajuda a reduzir o risco de sobreajuste do modelo. O hiperparâmetro penalty, que especifica o tipo de regularização a ser aplicada, foi definido como L1, o que indica que o modelo utilizou lasso como penalizador e isso significa que o modelo selecionou apenas os coeficientes mais relevantes, zerando os menos relevantes. O solver saga mostrou-se eficiente para este problema específico, otimizando o modelo e garantindo que o modelo convergisse dentro das 150 iterações definidas em max_iter.

O modelo utilizando revocação como métrica para avaliação obteve um retorno acumulado de 1,20, indicando que o modelo otimizado pela revocação venceu de forma modesta a estratégia de comprar e manter o ativo no período analisado. O Gráfico 5 compara graficamente as duas estratégias.

Gráfico 5 - Retorno acumulado da estratégia revocação contra a estratégia de comprar e manter o ativo



Fonte: autoria própria

O segundo modelo avaliado utilizou o escore F1 como métrica de avaliação. O escore F1 é a média harmônica entre a precisão e a revocação, oferecendo um equilíbrio entre evitar falsos positivos e minimizar falsos negativos. O GridSearchCV explorou diversas combinações de hiperparâmetros e identificou que os valores ótimos para esta métrica foram: C igual a 0,01, max_iter igual a 150, penalty definida como L1 e o solver como saga. Na Tabela 2 é apresentado os resultados

Tabela 2- Resultado da seleção de hiperparâmetros utilizando escore F1 como critério

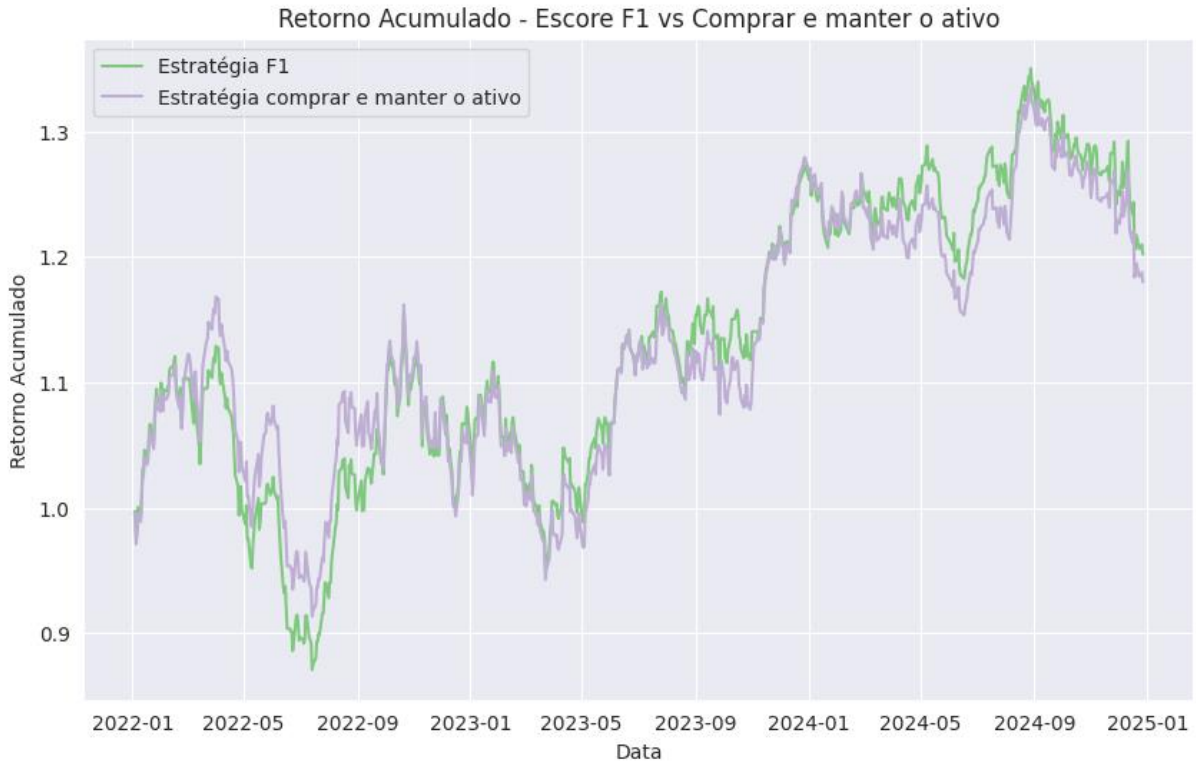
Hiperparâmetros	**Valores**
Métrica de avaliação	Escore F1
C	0,01
max_iter	150
penalty	L1
solver	saga

Fonte: autoria própria

O hiperparâmetro C, que controla a força da regularização do modelo, obteve o valor de 0,01, indicando uma regularização mais forte, o que ajuda a reduzir o risco de sobreajuste do modelo. O hiperparâmetro penalty, que especifica o tipo de regularização a ser aplicada, foi definido como L1, o que indica que o modelo utilizou lasso como penalizador e isso significa que o modelo selecionou apenas os coeficientes mais relevantes, zerando os menos relevantes. O solver saga mostrou-se eficiente para este problema específico, otimizando o modelo e garantindo que o modelo convergisse dentro das 150 iterações definidas em max_iter.

O modelo utilizando escore F1 como critério para a avaliação obteve um retorno acumulado de 1,20, indicando que o modelo otimizado pelo escore F1 venceu de forma modesta a estratégia de comprar e manter o ativo no período analisado. O Gráfico 6 abaixo compara graficamente as duas estratégias.

Gráfico 6 - Retorno acumulado da estratégia escore F1 contra a estratégia de comprar e manter o ativo.



Fonte: autoria própria

O terceiro modelo avaliado utilizou a precisão como métrica de avaliação. A precisão mensura a exatidão das previsões positivas. O GridSearchCV explorou diversas combinações de hiperparâmetros e identificou que os valores ótimos para maximizar essa métrica foram: C igual a 0,01, max_iter igual a 150, penalty definida como L2 e o solver como lbfgs. Na Tabela 3 são apresentados os resultados.

Tabela 3- Resultado da seleção de hiperparâmetros utilizando precisão como critério

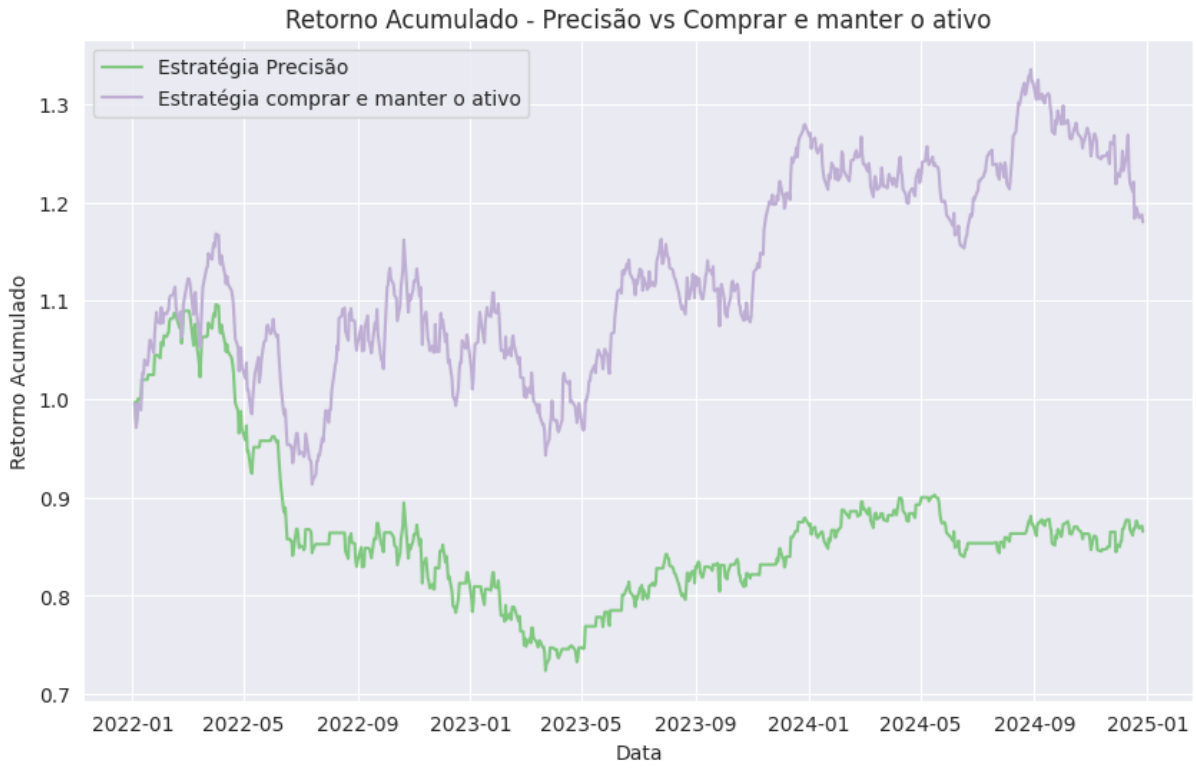
Hiperparâmetros	**Valores**
Métrica de avaliação	Precisão
C	0,01
Max_iter	150
Penalty	L2
Solver	lbfgs

Fonte: autoria própria.

O hiperparâmetro C, que controla a força da regularização do modelo, obteve o valor de 0,01, indicando uma regularização mais forte, o que ajuda a reduzir o risco de sobreajuste do modelo. O hiperparâmetro penalty foi definido como L2, o que indica que o modelo utilizou o penalizador ridge e isso significa que o modelo penalizou os coeficientes, mas sem zera-los, o que tende a suavizar os coeficientes. O solver lbfgs mostrou-se eficiente para este problema específico, otimizando o modelo e garantindo que o modelo convergisse dentro das 150 iterações definidas em max_iter.

O modelo utilizando precisão como critério obteve um retorno acumulado de 0,87, indicando que o modelo otimizado pela precisão perdeu para a estratégia de comprar e manter o ativo no período analisado. O Gráfico 7 compara graficamente as duas estratégias.

Gráfico 7- Retorno acumulado da estratégia precisão contra a estratégia de comprar e manter o ativo.



Fonte: autoria própria

O quarto modelo avaliado utilizou a acurácia como métrica de avaliação. A acurácia mensura o desempenho global do modelo ao calcular a fração de previsões corretas. O GridSearchCV explorou diversas combinações de hiperparâmetros e identificou que os valores

ótimos para maximizar essa métrica foram: C igual a 0,01, max_iter igual a 50, penalty definida como L2 e o solver como newton-cg. Na Tabela 4 são apresentados os resultados.

Tabela 4 - Resultado da seleção de hiperparâmetros utilizando acurácia como critério

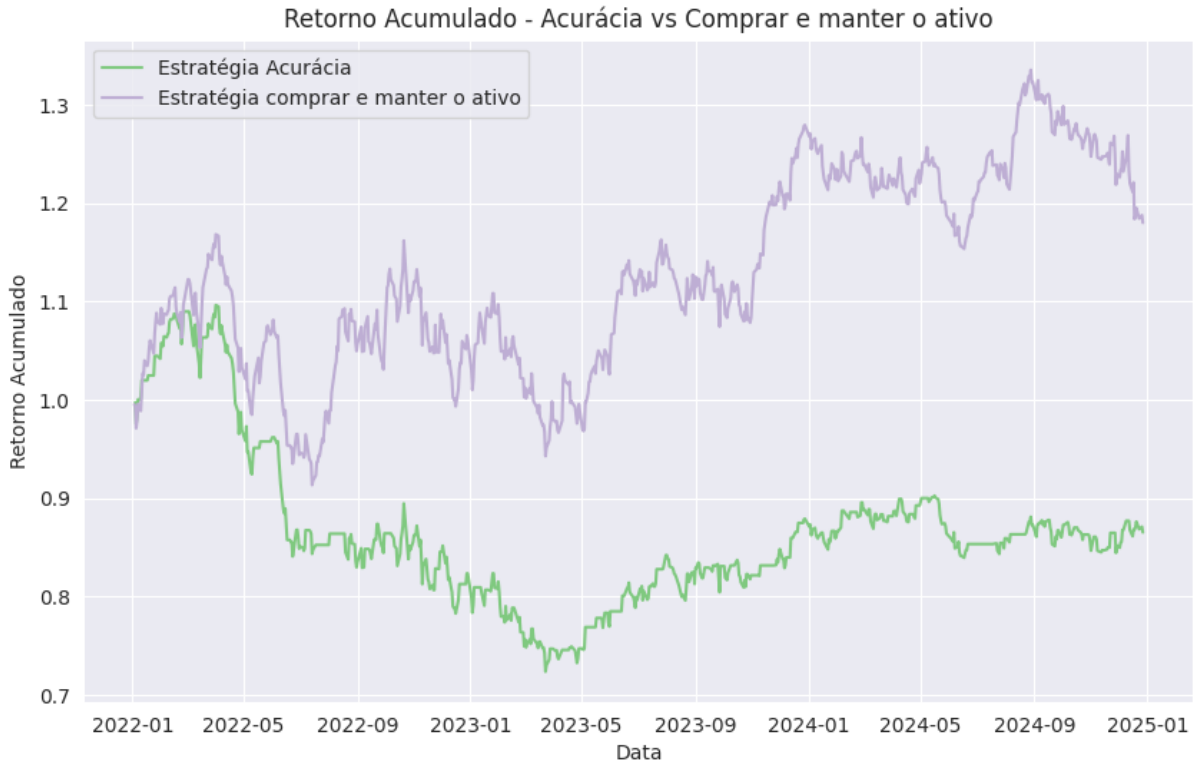
Hiperparâmetros	**Valores**
Scoring	Acurácia
C	0,01
Max_iter	50
Penalty	L2
Solver	newton-cg

Fonte: autoria própria

O hiperparâmetro C, que controla a força da regularização do modelo, obteve o valor de 0,01, indicando uma regularização mais forte, o que ajuda a reduzir o risco de sobreajuste do modelo. O hiperparâmetro penalty, que especifica o tipo de regularização a ser aplicada, foi definido como L2, o que indica que o modelo utilizou o penalizador ridge e isso significa que o modelo penalizou os coeficientes, mas sem zera-los, o que tende a suavizar os coeficientes. O solver saga mostrou-se eficiente para este problema específico, otimizando o modelo e garantindo que o modelo convergisse dentro das 50 iterações definidas em max_iter.

O modelo utilizando acurácia como critério para a avaliação obteve um retorno acumulado de 0,87, indicando que o modelo otimizado pela precisão perdeu para a estratégia de comprar e manter o ativo no período analisado. O Gráfico 8 compara graficamente as duas estratégias.

Gráfico 8- Retorno acumulado da estratégia acurácia contra a estratégia de comprar e manter o ativo.



Fonte: autoria própria

Após a execução e otimização de todos os modelos, foi possível analisar os resultados obtidos por cada um. Os modelos que priorizaram a revocação e o escore F1 apresentaram os melhores desempenhos, ambos com um retorno acumulado de 1,20, seguidos pelos modelos otimizados para acurácia e precisão, que obtiveram um retorno acumulado de 0,87. A estratégia de comprar e manter o ativo registrou um retorno acumulado de 1,18, superando os modelos otimizados para acurácia e precisão, mas ficando modestamente abaixo dos modelos otimizados para revocação e escore F1. A Tabela 5 resume a estratégia e o retorno acumulado, respectivamente.

Tabela 5- Retorno acumulado das estratégias

Estratégias	**Retorno acumulado**
Revocação	1,20
Escore F1	1,20
Precisão	0,87

Acurácia	0,87
Comprar e manter o ativo	1,18

Fonte: autoria própria

9. CONCLUSÃO E DESAFIOS PARA O FUTURO

Este trabalho teve como objetivo utilizar a regressão logística para realizar previsões no ETF conhecido como PIBB11 no período de 2022 a 2024.

A metodologia proposta neste estudo foi aplicada ao longo de diversas otimizações do modelo, priorizando métricas como revocação, escore F1, precisão e acurácia, e comparando seu desempenho com a estratégia de comprar e manter o ativo, que serviu como o *benchmark* do estudo. Os resultados indicaram que modelos que priorizaram a revocação e o escore F1 apresentaram retornos modestos superiores à estratégia de comprar e manter o ativo, enquanto a otimização das métricas de acurácia e precisão obteve um desempenho inferior. A análise dos retornos evidenciou a importância da escolha das métricas de otimização no desempenho do modelo, destacando o papel da regressão logística e de seus hiperparâmetros para previsões no mercado financeiro. Apesar da revocação e do escore F1 apresentarem um retorno acumulado de 1,20, superior aos 1,18 da estratégia de comprar e manter o ativo, a diferença de retorno acumulado foi modesta, reforçando a dificuldade de realizar previsões consistentes no mercado financeiro, conforme discutido anteriormente neste estudo.

Para o futuro, sugere-se utilizar outros modelos de aprendizado de máquinas, como árvore de decisão, floresta aleatória e também modelos de redes neurais e compará-los entre si, além da incorporação também de variáveis fundamentalistas como regressores. Sugere-se também estudar o impacto de cada regressor para entender melhor o modelo por trás dos panos e escolher o conjunto ideal de regressores que maximizam os retornos.

REFERÊNCIAS:

AYYILDIZ, N.; ISKENDEROGLU, O. How effective is machine learning in stock market predictions?, **Heliyon**, [S. *l.*], v. 10, Issue 2, Disponível em: https://www.sciencedirect.com/science/article/pii/S2405844024001543. Acesso em: 18 fev. 2025.

AGUIRRE, A. A.; MEDINA, N. A; MEDINA, R. A., Artificial intelligence applied to investment in variable income through the MACD (moving average convergence/ divergence). **Journal of Economics, Finance and Administrative Science indicator**, Lima,

Peru, v. 26, n. 52, 2021. DOI: http://dx.doi.org/10.1108/jefas-06-2020-0203. Disponível em: http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S2077-18862021000200268. Acesso em: 01 fev. 2025.

BLACKROCK. **ARTIFFICIAL intelligence and machine learning in asset management**. Nova Iorque, EUA. Disponível em: https://www.blackrock.com/viewpoint-artificial-intelligence-machine-learning-asset-management.pdf Acesso em: 10 jun. 2024.

BERNSTEIN, P. L. The view from the Top of the Tower. **Capital Ideas:** The Improbable Origins of Modern Wall Street. New Jersey: John Wiley & Sons, 2005. p. 225-227

CHAWLA, N. V. *et al*. SMOTE: synthetic minority over-sampling technique**, Journal of artificial intelligence research**, p. 321-357, 2002. Disponível em: https://www.jair.org/index.php/jair/article/view/10302/24590 Acesso em: 1 jun. 2024.

FÁVERO, L. P.; BELFIORE, P. **Manual de análise de dados:** estatística e modelagem multivariada com EXCEL, SPSS e STATA. Rio de Janeiro: Elsevier, 2017.

FARIA, E. L. *et al*. **Previsão do Mercado de Ações Brasileiro utilizando Redes Neurais Artificias**. CBPF-NT-002/2008. Rio de Janeiro: CBPF, 2008. Disponível em: https://cbpfindex.cbpf.br/publication_pdfs/NT00208.2011_01_04_11_01_14.pdf Acesso em: 5 abr. 2024.

GAO, H. *et al*. Machine learning in business and finance: a literature review and research opportunities. **Financ Innov 10**, n. 86, 2024. DOI: https://doi.org/10.1186/s40854-024-00629-z. Disponível em: https://jfin-swufe.springeropen.com/articles/10.1186/s40854-024-00629-z Acesso em: 8 abr. 2025

GERÓN, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow:** Concepts, Tools, and Techniques to Build Intelligent Systems. 2. ed. atual. [*S. l.*]*:* O'Reilly Media, 2021.

GOLDMAN SACHS. **Quant Insight Now on Marquee**. Nova Iorque, EUA. Disponível em: https://marquee.gs.com/welcome/campaigns/quant-insight-on-marquee Acesso em: 1 jun. 2024.

GOMES, P. C. Regressão Logística: Um Guia Detalhado. *In*: GOMES, Pedro Cesar. **Datageeks**. [*S. l.*], 10 abr. 2024. Disponível em: https://www.datageeks.com.br/regressao-logistica/. Acesso em: 19 fev. 2025.

HASTIE *et al*. **An Introduction to Statistical Learning with Applications in Python.** [*S. l.*]: Springer, 2023. p. 138-144

INVESTING.COM. **About Us**. Disponível em: https://br.investing.com/about-us/. Acesso em: 15 mar. 2025.

JIANG, H.; HU, X.; JIA, H. Penalized logistic regressions with technical indicators predict up and down trends. **PubMed Central**, [S. *l.*], p. 1-12, 16 ago. 2022. DOI: 10.1007/s00500-022-07404-1. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9379894/ Acesso em: 13 jun. 2024.

MALKIEL, B. G. **A Random Walk down Wall Street:** The Time-tested Strategy for Successful Investing. 11. ed, Nova Iorque: W. W. Norton & Company, 2006.
MURPHY, J. J. **Technical Analysis of the Financial Markets**: A Comprehensive Guide to Trading Methods and Applications**.** Nova Iorque: New York Institute of Finance, 1999.

NETO, A. S., **Mercado Financeiro**, São Paulo: Atlas. 2021.

NETTO, A.; MACIEL, F. **Python Para Data Science e Machine Learning Descomplicado**. São Paulo: Alta Books. 2021.

PEDREGOSA, F. *et al*. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v.12, p.2825-2830, 2011. Disponível em: https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html Acesso em: 21 jun. 2024.

PINTO, L. O que é ETF: esse pode ser o seu 1º investimento na Bolsa? **Expert XP**, São Paulo, Brasil, 12 jun. 2024. Disponível em: https://conteudos.xpi.com.br/aprenda-a-investir/relatorios/o-que-e-etf/ Acesso em: 15 mar. 2025.

RHAMAN, H. A. A.; WAH, Y. B.; HUAT, O. S. Predictive Performance of Logistic Regression for Imbalanced Data with Categorical Covariate. **Pertanika Journal of Tropical Agricultural Science**, Kuala Lumpur, Malasya, p. 181-197, 2021. Disponível em: http://www.pertanika.upm.edu.my/resources/files/Pertanika%20PAPERS/JST%20Vol.%2029%20(1)%20Jan.%202021/10%20JST-1911-2020.pdf Acesso em: 1 jun. 2024.

SAMANT, S. Prediction of Financial Performance Using Genetic Algorithm and Associative Rule Mining**. International Journal of Engineering Research and General Science,** [S. *l*.], v. 3, Issue 1, 2015. Disponível em: https://pnrsolution.org/Vol3/Issue1/135.pdf Acesso em: 10 jun. 2024.

SAUD, A. S.; SHAKYA, S. Technical indicator empowered intelligent strategies to predict stock trading signals, **Journal of Open Innovation: Technology, Market, and Complexity**, v. 10, Issue 4, 2024. DOI: https://doi.org/10.1016/j.joitmc.2024.100398. Disponível em: https://www.sciencedirect.com/science/article/pii/S2199853124001926 Acesso em: 17 fev. 2025.

SECURATO, J. R.; SECURATO, J. C.; VEIGA, R. P, Introdução ao mercado financeiro. *In:* SECURATO, J.R; SECURATO, J.C. (org.). **MERCADO FINANCEIRO**: Conceitos, cálculos e análise de investimento. São Paulo: Saint Paul Editora. 2023, p. 23-52.

SILVA, M. V. **ALGORITMOS DE MACHINE LEARNING EM ESTRATÉGIAS DE TRADING UMA ANÁLISE DA EFICIÊNCIA E APLICABILIDADE NO MERCADO FINANCEIRO**. 2024. Dissertação (Mestrado Profissional em Economia e Finanças) – Escola de Economia de São Paulo, Fundação Getúlio Vargas, São Paulo, 2024.

SHAH, A. *et al*. Identifying Trades Using Technical Analysis and ML/DL models. **International Journal Of Innovative Research**, [S. *l*.], v. 11, Issue 4, 2023. Disponível em: https://arxiv.org/abs/2304.09936 Acesso em: 20 maio 2024.

VIDOTTO, R. S.; MIGLIATO, A. L.; ZAMBON, A. C., O Moving Average Convergence-Divergence como Ferramenta para a Decisão de Investimentos no Mercado de Ações. **Anpad RAC**, Curitiba, v. 13, n. 2, art. 7, p. 291-309, 2009. Disponível em: https://www.scielo.br/j/rac/a/QhsbBQSzyMR75VtJP5chtbm/. Acesso em: 10 fev. 2025.

WILLIAMS, O. D. **Empirical Optimization of Bollinger Bands for Profitability**. Dissertação (Mestrado em Gestão de Risco), Simon Fraser University, Burnaby, Canada, 2006. DOI: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2321140 Disponível em: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2321140 Acesso em: 18 jun. 2024.

ZAINI, B. J. et al. Classify Stock Market Movement Based on Technical Analysis Indicators Using Logistic Regression. **Journal of Advanced Research in Business and Management Studies, School of Quantitative Sciences**, Malaysia, v. 14, issue 1, p.35-41, 2019. Disponível em: https://www.akademiabaru.com/doc/ARBMSV14_N1_P35_41.pdf. Acesso em: 20 abr. 2024.

APÊNDICE - Código em Python

```
pip install ta # Uma biblioteca interessante conhecida como "Technical Analysis" que possui formas de cálcular os mais diversos indicadores técnicos.

############### IMPORTANDO AS BIBLIOTECAS PARA O AMBIENTE ###############################

import pandas as pd # Importando a biblioteca para fazer a exploração dos dados
pd.set_option('display.float_format', lambda x: '%.3f' % x) # Para limitar os valores de pontos flutuantes em 3 casas decimais
import numpy as np # Importando a biblioteca para auxiliar na exploração dos dados
import datetime # Importando a biblioteca para trabalhar com datas, horários e etc SE necessário
import yfinance as yf # Importando a biblioteca para pegar o preço das ações após ter instalado-a.
import matplotlib.pyplot as plt # Importando a biblioteca gráfica para visualização
%matplotlib inline
import seaborn as sns # Importando a biblioteca gráfica para visualização
from IPython.core.display import display, HTML # Biblioteca para controlar a exibição dos conteúdos em diferentes formatos de textos, tabelas e etc.
display(HTML("<style>.container { width:100% !important; }</style>")) # SE necessário
import ta # Importando a biblioteca TA para fazer as análises técnicas

# Modelo e treinamento
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split, TimeSeriesSplit,
GridSearchCV, cross_val_score

# Métricas
from sklearn.metrics import precision_score, accuracy_score,
recall_score, f1_score

# Importando o arquivo CSV para o ambiente de desenvolvimento
dados_pibb11 = pd.read_csv('/content/PIBB11.csv')

# Renomeando as colunas para tirar os acentos e etc.

colunas_renomeadas = {
    'Último': 'Close',
    'Abertura': 'Open',
    'Máxima': 'High',
    'Mínima': 'Low'
}

dados_pibb11.rename(columns=colunas_renomeadas, inplace=True)


# Tratando os dados para que fique no formato correto

colunas_para_float = ['Close', 'Open', 'High', 'Low']


for col in colunas_para_float:
    dados_pibb11[col] = dados_pibb11[col].str.replace(',',
'.').astype(float)

dados_pibb11['Data'] = pd.to_datetime(dados_pibb11['Data'])

dados_pibb11.set_index('Data', inplace=True)

print(f'O DataFrame do PIBB11 tem o formato de {dados_pibb11.shape}
linhas e colunas, respectivamente.')

dados_pibb11.sort_index(inplace=True)

# Criando uma função que irá me retornar uma coluna binária.
# Quando o preço de fechamento (t) for maior que o preço de fechamento
anteriot (t-1), a coluna direcao receberá o valor 1
# Quando o preço de fechamento (t) for menor que o preço de fechamento
anteriot (t-1), a coluna direcao receberá o valor 0

def criacao_binaria(dataframe):
```

```
    dataframe['fechamento_dia_anterior'] = dataframe['Close'].shift(1)
    dataframe['direcao'] = 0
    dataframe.loc[dataframe['Close'] >
dataframe['fechamento_dia_anterior'], 'direcao'] = 1
    dataframe.drop(columns=['fechamento_dia_anterior'], inplace=True)

criacao_binaria(dados_pibb11)

dados_pibb11_backup = dados_pibb11.copy()


# Criando uma função para ver gráfico de linhas sem ter que ficar
repetindo toda hora os mesmos códigos
def grafico_lineplot(x, y, dataset, label, xlabel, ylabel, titulo):
  sns.set_palette('Accent')
  sns.set_style('darkgrid')
  ax = sns.lineplot(x = x, y = y, data = dataset, label = label)
  ax.set_xlabel(xlabel, fontsize = 14)
  ax.set_ylabel(ylabel, fontsize = 14)
  ax.figure.set_size_inches(14,6)
  ax.set_title(titulo, loc = 'center', fontsize = 15)

def grafico_barplot(dataframe, xlabel, ylabel, titulo):
    contagem = dataframe['direcao'].value_counts().sort_index()  #
Ordena os índices para manter a coerência
    ax = sns.barplot(x=contagem.index, y=contagem.values)

    ax.set_xlabel(xlabel, fontsize=12)
    ax.set_ylabel(ylabel, fontsize=12)
    ax.figure.set_size_inches(6, 6)
    ax.set_title(titulo, loc='center', fontsize=12)

    for i, v in enumerate(contagem.values):
        ax.text(i, v + v * 0.02, str(v), color='black', ha='center',
fontsize=10)

# lineplot do PIB11
grafico_lineplot(dados_pibb11.index, 'Close', dados_pibb11,
'fechamento', 'Data', 'Valor (R$)', 'Série histórica dos preços -
PIBB11')


# Gráfico de barras do PIBB11
grafico_barplot(dados_pibb11, 'Direção', '', 'Balanceamento da variável
alvo')

# Criação de uma função que irá criar minhas variáveis independentes
para quaisquer datasets que eu passar.
def variaveis_independentes(dataframe):
```

```
  #Criando a média móvel #
  dataframe['mm_20'] = dataframe['Close'].rolling(window=20).mean() #
Criação da coluna média móvel = 20 para variações de tendências de
curto prazo.
  #dataframe['mm_200'] = dataframe['Close'].rolling(window=200).mean()
# Criação da coluna média móvel = 200  para variações de tendências de
longo prazo.

  # Criando a variável média móvel exponencial de 20 dias.
  dataframe['ema_20'] = dataframe['Close'].ewm(span=20,
adjust=False).mean()  # EMA de 20 períodos

  # Criando a váriavel bb que vai receber as Bandas de Bollinger com a
janela sendo 20.
  bb = ta.volatility.BollingerBands(dataframe['Close'], window=20)
  dataframe['bb_meio'] = bb.bollinger_mavg()  # Criação da coluna que
vai ter a média móvel da Bandas de Bollinger
  dataframe['bb_cima'] = bb.bollinger_hband()  # Criação da coluna que
vai ter a média móvel da  Bandas de Bollinger
  dataframe['bb_baixo'] = bb.bollinger_lband()  # Criação da coluna que
vai ter a média móvel da  Bandas de Bollinger

  # Criando outro indicador técnico, o dos osciladores estocásticos,
com as configurações padrões de 14, 3 e 3.
  stoch = ta.momentum.StochasticOscillator(high=dataframe['High'],
low=dataframe['Low'], close=dataframe['Close'])
  dataframe['%k'] = stoch.stoch()  # Representa a posição atual do
preço em relação ao intervalo entre o valor mínimo e máximo durante um
determinado período
  dataframe['%d'] = stoch.stoch_signal()  # Média móvel do valor %K
para ajudar a suavizar os movimentos do indicador.

  # Criando outro indicador técnico, INDICE DE FORÇA RELATIVA, com a
janela sendo 14.
  dataframe['rsi'] = ta.momentum.rsi(dataframe['Close'], window=14)

  # Criando outro indicador técnico, a Moving Average Convergence
Divergence (MACD) com as janelas de 12, 26 e 9.
  macd = ta.trend.MACD(dataframe['Close'])
  dataframe['macd_rapida'] = macd.macd()
  dataframe['macd_linha_sinal'] = macd.macd_signal()
  dataframe['macd_histograma'] = macd.macd_diff()

variaveis_independentes(dados_pibb11)

dados_pibb11.drop(['Open', 'High', 'Low', 'Close','Vol.', 'Var%'], axis
= 1, inplace = True)
```

```
# CRIAÇÃO DO MODELO

dias = 1

dados_pibb11['shift_direcao'] = dados_pibb11[['direcao']].shift(-dias)

dados_pibb11.dropna(inplace = True)

dados_pibb11['shift_direcao'] =
dados_pibb11['shift_direcao'].astype('int64')

# Lista das variáveis target do modelo
variaveis_target = ['mm_20', 'ema_20', 'bb_cima', 'bb_baixo', '%k',
'%d', 'rsi', 'macd_rapida', 'macd_linha_sinal', 'macd_histograma']
dados_pibb11_variaveis = dados_pibb11[variaveis_target]

# Dropando as linhas com valores nulos, lembrando que quando se usa
médias móveis, as linhas anteriores ficam com valores nulos.
dados_pibb11.dropna(inplace=True)

# Transformando o indice para datetime por objetividade.
dados_pibb11.index = pd.to_datetime(dados_pibb11.index)

# Separando os dados em treino e teste manualmente.
# Poderia ter sido feito utilizando o Shuffle = True do
train_test_split? Claro, mas
# Eu prefiro fazer manualmente, de forma com que eu consiga separar a
data especificamente.
dados_treino = dados_pibb11.loc[:'2021-12-31']
dados_teste = dados_pibb11.loc['2022-01-01':]

X_treino = dados_treino[variaveis_target]
y_treino = dados_treino['shift_direcao']

X_teste = dados_teste[variaveis_target]
y_teste = dados_teste['shift_direcao']


modelo_shift = LogisticRegression(random_state=42)

tscv = TimeSeriesSplit(n_splits=5)

dados_teste['retorno'] = dados_pibb11_backup['Close'].pct_change()

dados_teste['estrategia_comprar_e_manter'] = (1 +
dados_teste['retorno']).cumprod()

print(f"Retorno da estratégia:
{dados_teste['estrategia_comprar_e_manter'].iloc[-1]:.2f}")
```

```
plt.figure(figsize=(10, 6))
plt.plot(dados_teste['estrategia_comprar_e_manter'],
label='Estratégia', color = '#bbabd3')
plt.legend()
plt.title('Retorno Acumulado da Estratégia de Comprar e Manter o
Ativo')
plt.xlabel('Data')
plt.ylabel('Retorno Acumulado')
plt.show()


#### RECALL ####

param_grid = [
    {'C': [0.01, 0.1, 1, 10, 100], 'max_iter': [50, 100, 150, 200],
'penalty': ['l2', None], 'solver': ['newton-cg', 'lbfgs', 'sag',
'newton-cholesky']},
    {'C': [0.01, 0.1, 1, 10, 100], 'max_iter': [50, 100, 150, 200],
'penalty': ['l1', 'l2'], 'solver': ['liblinear']},
    {'C': [0.01, 0.1, 1, 10, 100], 'max_iter': [50, 100, 150, 200],
'penalty': ['l1', 'l2', None], 'solver': ['saga']}
]



grid_search_recall = GridSearchCV(
    estimator=modelo_shift,
    param_grid=param_grid,
    cv=tscv,
    scoring='recall',
    n_jobs=-1,
    verbose=2
)

melhor_modelo_recall = grid_search_recall.fit(X_treino, y_treino)
melhor_modelo_recall = melhor_modelo_recall.best_estimator_
previsoes_melhor_modelo_recall = melhor_modelo_recall.predict(X_teste)

print('-'*100)
print('Recall score')
print(recall_score(y_teste, previsoes_melhor_modelo_recall))
melhores_parametros_recall = grid_search_recall.best_params_
print('Melhores parâmetros encontrados Recall Score:',
melhores_parametros_recall)
print('-'*100)

dados_teste['retorno'] = dados_pibb11_backup['Close'].pct_change()
```

```
# Convertendo previsoes_shift para uma Series do pandas para usar o
método shift
previsoes_melhor_modelo_recall =
pd.Series(previsoes_melhor_modelo_recall, index=dados_teste.index)

# Aplicando a estratégia
# Se a previsão for 1 (subida), comprar e capturar o retorno do dia.
# Se a previsão for 0 (descida), não comprar (retorno = 0).
dias = 1
dados_teste['retorno_estrategia'] = dados_teste['retorno'] *
previsoes_melhor_modelo_recall.shift(dias)

dados_teste['retorno_acumulado'] = (1 +
dados_teste['retorno_estrategia']).cumprod()

dados_teste['estrategia_comprar_e_manter'] = (1 +
dados_teste['retorno']).cumprod()

# Exibindo o retorno final da estratégia
print(f"Retorno final da estratégia com recall:
{dados_teste['retorno_acumulado'].iloc[-1]:.2f}")
print(f"Retorno final Comprar e manter o ativo:
{dados_teste['estrategia_comprar_e_manter'].iloc[-1]:.2f}")



plt.figure(figsize=(10, 6))
plt.plot(dados_teste['retorno_acumulado'], label='Estratégia
Revocação')
plt.plot(dados_teste['estrategia_comprar_e_manter'], label='Estratégia
Comprar e manter o ativo')
plt.legend()
plt.title('Retorno Acumulado - Revocação vs Comprar e manter o ativo')
plt.xlabel('Data')
plt.ylabel('Retorno Acumulado')
plt.show()


### escore f1 ###

param_grid = [
    {'C': [0.01, 0.1, 1, 10, 100], 'max_iter': [50, 100, 150, 200],
'penalty': ['l2', None], 'solver': ['newton-cg', 'lbfgs', 'sag',
'newton-cholesky']},
    {'C': [0.01, 0.1, 1, 10, 100], 'max_iter': [50, 100, 150, 200],
'penalty': ['l1', 'l2'], 'solver': ['liblinear']},
    {'C': [0.01, 0.1, 1, 10, 100], 'max_iter': [50, 100, 150, 200],
'penalty': ['l1', 'l2', None], 'solver': ['saga']}
]
```

```
grid_search_f1 = GridSearchCV(
    estimator=modelo_shift,
    param_grid=param_grid,
    cv=tscv,
    scoring='f1',
    n_jobs=-1,
    verbose=2
)

melhor_modelo_f1 = grid_search_f1.fit(X_treino, y_treino)
melhor_modelo_f1 = melhor_modelo_f1.best_estimator_
previsoes_melhor_modelo_f1 = melhor_modelo_f1.predict(X_teste)
print('-'*100)
print("f1 score")
print(f1_score(y_teste, previsoes_melhor_modelo_f1))
print('-'*100)
melhores_parametros_f1 = grid_search_f1.best_params_
print("Melhores parâmetros encontrados F1 Score:",
melhores_parametros_f1)
print('-'*100)

dados_teste['retorno'] = dados_pibb11_backup['Close'].pct_change()

previsoes_melhor_modelo_f1 = pd.Series(previsoes_melhor_modelo_f1,
index=dados_teste.index)


dias = 1

dados_teste['retorno_estrategia'] = dados_teste['retorno'] *
previsoes_melhor_modelo_f1.shift(dias)

dados_teste['retorno_acumulado'] = (1 +
dados_teste['retorno_estrategia']).cumprod()

dados_teste['estrategia_comprar_e_manter'] = (1 +
dados_teste['retorno']).cumprod()

print(f"Retorno final da estratégia F1 Score:
{dados_teste['retorno_acumulado'].iloc[-1]:.2f}")
print(f"Retorno final Comprar e manter o ativo:
{dados_teste['estrategia_comprar_e_manter'].iloc[-1]:.2f}")

plt.figure(figsize=(10, 6))
plt.plot(dados_teste['retorno_acumulado'], label='Estratégia F1')
```

```
plt.plot(dados_teste['estrategia_comprar_e_manter'], label='Estratégia
comprar e manter o ativo')
plt.legend()
plt.title('Retorno Acumulado - Escore F1 vs Comprar e manter o ativo')
plt.xlabel('Data')
plt.ylabel('Retorno Acumulado')
plt.show()

### precisão ###

param_grid = [
    {'C': [0.01, 0.1, 1, 10, 100], 'max_iter': [50, 100, 150, 200],
'penalty': ['l2', None], 'solver': ['newton-cg', 'lbfgs', 'sag',
'newton-cholesky']},
    {'C': [0.01, 0.1, 1, 10, 100], 'max_iter': [50, 100, 150, 200],
'penalty': ['l1', 'l2'], 'solver': ['liblinear']},
    {'C': [0.01, 0.1, 1, 10, 100], 'max_iter': [50, 100, 150, 200],
'penalty': ['l1', 'l2', None], 'solver': ['saga']}
]



grid_search_precision = GridSearchCV(
    estimator=modelo_shift,
    param_grid=param_grid,
    cv=tscv,
    scoring='precision',
    n_jobs=-1,
    verbose=2
)

melhor_modelo_precision = grid_search_precision.fit(X_treino, y_treino)
melhor_modelo_precision = melhor_modelo_precision.best_estimator_
previsoes_melhor_modelo_precision =
melhor_modelo_precision.predict(X_teste)
print('-'*100)
print("Precision score")
print(precision_score(y_teste, previsoes_melhor_modelo_precision))
melhores_parametros_precision = grid_search_precision.best_params_
print("Melhores parâmetros encontrados Precision Score:",
melhores_parametros_precision)
print('-'*100)

dados_teste['retorno'] = dados_pibb11_backup['Close'].pct_change()

previsoes_melhor_modelo_precision =
pd.Series(previsoes_melhor_modelo_precision, index=dados_teste.index)
```

```
dias = 1
dados_teste['retorno_estrategia'] = dados_teste['retorno'] *
previsoes_melhor_modelo_precision.shift(dias)

dados_teste['retorno_acumulado'] = (1 +
dados_teste['retorno_estrategia']).cumprod()

dados_teste['estrategia_comprar_e_manter'] = (1 +
dados_teste['retorno']).cumprod()

print(f"Retorno final da estratégia:
{dados_teste['retorno_acumulado'].iloc[-1]:.2f}")
print(f"Retorno final da estratégia de comprar e manter:
{dados_teste['estrategia_comprar_e_manter'].iloc[-1]:.2f}")

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.plot(dados_teste['retorno_acumulado'], label='Estratégia Precisão')
plt.plot(dados_teste['estrategia_comprar_e_manter'], label='Estratégia
comprar e manter o ativo')
plt.legend()
plt.title('Retorno Acumulado - Precisão vs Comprar e manter o ativo')
plt.xlabel('Data')
plt.ylabel('Retorno Acumulado')
plt.show()


### acurácia ###

param_grid = [
    {'C': [0.01, 0.1, 1, 10, 100], 'max_iter': [50, 100, 150, 200],
'penalty': ['l2', None], 'solver': ['newton-cg', 'lbfgs', 'sag',
'newton-cholesky']},
    {'C': [0.01, 0.1, 1, 10, 100], 'max_iter': [50, 100, 150, 200],
'penalty': ['l1', 'l2'], 'solver': ['liblinear']},
    {'C': [0.01, 0.1, 1, 10, 100], 'max_iter': [50, 100, 150, 200],
'penalty': ['l1', 'l2', None], 'solver': ['saga']}
]



grid_search_accuracy = GridSearchCV(
    estimator=modelo_shift,
    param_grid=param_grid,
    cv=tscv,
    scoring='accuracy',
    n_jobs=-1,
    verbose=2
```

```
)

melhor_modelo_accuracy = grid_search_accuracy.fit(X_treino, y_treino)
melhor_modelo_accuracy = melhor_modelo_accuracy.best_estimator_
previsoes_melhor_modelo_accuracy =
melhor_modelo_accuracy.predict(X_teste)
print('-'*100)
print("Accuracy score")
print(accuracy_score(y_teste, previsoes_melhor_modelo_accuracy))
melhores_parametros_accuracy = grid_search_accuracy.best_params_
print("Melhores parâmetros encontrados Accuracy Score:",
melhores_parametros_accuracy)
print('-'*100)


dados_teste['retorno'] = dados_pibb11_backup['Close'].pct_change()

previsoes_melhor_modelo_accuracy =
pd.Series(previsoes_melhor_modelo_accuracy, index=dados_teste.index)

dias = 1

dados_teste['retorno_estrategia'] = dados_teste['retorno'] *
previsoes_melhor_modelo_accuracy.shift(dias)

dados_teste['retorno_acumulado'] = (1 +
dados_teste['retorno_estrategia']).cumprod()

dados_teste['estrategia_comprar_e_manter'] = (1 +
dados_teste['retorno']).cumprod()

print(f"Retorno final da acurácia:
{dados_teste['retorno_acumulado'].iloc[-1]:.2f}")
print(f"Retorno final da estratégia de comprar e manter:
{dados_teste['estrategia_comprar_e_manter'].iloc[-1]:.2f}")


plt.figure(figsize=(10, 6))
plt.plot(dados_teste['retorno_acumulado'], label='Estratégia Acurácia')
plt.plot(dados_teste['estrategia_comprar_e_manter'], label='Estratégia
comprar e manter o ativo')
plt.legend()
plt.title('Retorno Acumulado - Acurácia vs Comprar e manter o ativo')
plt.xlabel('Data')
plt.ylabel('Retorno Acumulado')
plt.show()


### gráficos monografia e etc"
```

```
# GRAFICO MONO TIMESERIESSPLIT

# Número de dados e número de splits (folds)
n_samples = 100
n_splits = 5

X = np.arange(n_samples)

tscv = TimeSeriesSplit(n_splits=n_splits)

plt.figure(figsize=(8, 6))


for i, (train_index, test_index) in enumerate(tscv.split(X)):
    plt.scatter(train_index, [i + 0.5] * len(train_index), c='green',
marker='_', lw=10, label='Treinamento' if i == 0 else "")
    plt.scatter(test_index, [i + 0.5] * len(test_index), c='red',
marker='_', lw=10, label='Teste' if i == 0 else "")

plt.legend(loc='upper right', bbox_to_anchor=(1, 0.93))
plt.xlabel('Série histórica')
plt.ylabel('Splits')
plt.title('Validação Cruzada com TimeSeriesSplit')
plt.yticks(np.arange(n_splits) + 0.5, [f'Split {i+1}' for i in
range(n_splits)])
plt.show()

def grafico_lineplot(dataset, y, label, xlabel, ylabel, titulo,
data_corte):
    sns.set_palette('Accent')
    sns.set_style('darkgrid')
    ax = sns.lineplot(x=dataset.index, y=dataset[y], label=label)
    plt.axvline(x=pd.to_datetime(data_corte), color='red', linestyle='-
', linewidth=2)
    ax.set_xlabel(xlabel, fontsize=14)
    ax.set_ylabel(ylabel, fontsize=14)
    ax.figure.set_size_inches(14, 6)
    ax.set_title(titulo, loc='center', fontsize=15)
    plt.show()
dados_pibb11_backup.index = pd.to_datetime(dados_pibb11_backup.index)
data_corte = "2022-01-01"

grafico_lineplot(dataset=dados_pibb11_backup, y='Close',
label='PIBB11',
                 xlabel='Data', ylabel='Valor', titulo='Divisão entre
Treinamento e Teste',
                 data_corte=data_corte)
```