TECH CHALLENGE FASE 2:
Modelo preditivo do índice IBOVESPA

Sumário

Introdução

Este trabalho tem como objetivo a construção de um modelo preditivo para o índice IBOVESPA, principal indicador de desempenho das ações negociadas na B3. Através da análise de dados históricos diários do índice e da aplicação do algoritmo de Machine Learning XGBoost, buscamos prever se o IBOVESPA fechará em baixa ou alta (0 e 1) no dia seguinte. Neste texto, abordaremos brevemente temas como a análise exploratória de dados, análise de série temporal e modelos de ML. Além disso, serão apresentados os resultados obtidos pelo modelo escolhido, demonstram uma acurácia de 70% na previsão, evidenciando o potencial da abordagem para auxiliar na tomada de decisões.

Ferramentas utilizadas

O Trabalho foi realizado utilizando a linguagem Python por meio da IDE Visual Studio Code. Foram utilizadas algumas bibliotecas da linguagem Python, como Pandas, Matplotlib, Seaborn e Numpy para tratamento e análise dos dados, além das bibliotecas Statsmodels, XGBoost, Scikit-Learn para testes e implementação do modelo de preditivo de Machine Learning.

Base de dados

Os dados analisados neste trabalho foram coletados através do site br.investing.com, onde foi selecionada a periodicidade diária, trazendo dados de 2015 a 2025 por meio do download de um arquivo .csv (Valores Separados por Vírgula).

Preparando os dados

A base de dados original contém as seguintes colunas, que representam informações diárias do índice IBOVESPA:

- **Data:** A data específica de registro das informações do índice.
- **Último:** O valor de fechamento do IBOVESPA no dia.
- **Abertura:** O valor de abertura do IBOVESPA no início do dia de negociação.
- **Máxima:** O valor mais alto atingido pelo IBOVESPA durante o dia.
- **Mínima:** O valor mais baixo atingido pelo IBOVESPA durante o dia.
- **Vol. (Volume):** O volume de contratos negociados no dia, indicando a quantidade de transações.
- **Var% (Variação Percentual):** A variação percentual do IBOVESPA em relação ao valor de fechamento do dia anterior.

Utilizando a biblioteca Pandas do Python, foi realizado o tratamento da base de dados, removendo caracteres especiais dos valores, ajuste nos tipos de dados, além da renomeação das colunas.

Base original:

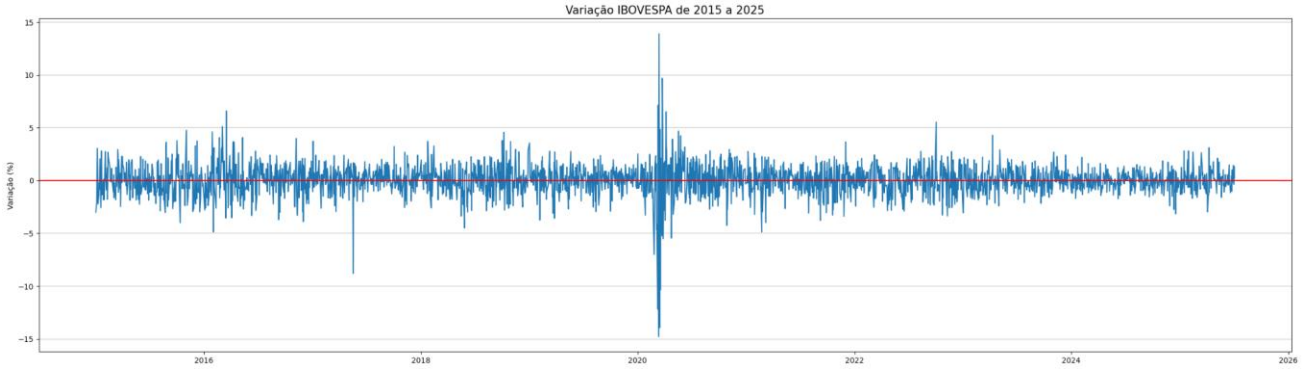
	Data	# Último	# Abertura	# Máxima	# Mínima	Vol.	Var%
0	04.07.2025	141.264	140.928	141.564	140.597	3,31B	0,24%
1	03.07.2025	140.928	139.051	141.304	139.051	6,08B	1,35%
2	02.07.2025	139.051	139.586	140.049	138.384	8,81B	-0,36%
3	01.07.2025	139.549	138.855	139.695	138.855	6,35B	0,50%
4	30.06.2025	138.855	136.865	139.103	136.43	7,68B	1,45%

Base tratada:

	Data	# Fechamento	# Abertura	# Maxima	# Minima	# Volume	# Variacao
0	2025-07-04 00:00:00	141.264	140.928	141.564	140.597	3.31	0.24
1	2025-07-03 00:00:00	140.928	139.051	141.304	139.051	6.08	1.35
2	2025-07-02 00:00:00	139.051	139.586	140.049	138.384	8.81	-0.36
3	2025-07-01 00:00:00	139.549	138.855	139.695	138.855	6.35	0.5
4	2025-06-30 00:00:00	138.855	136.865	139.103	136.43	7.68	1.45

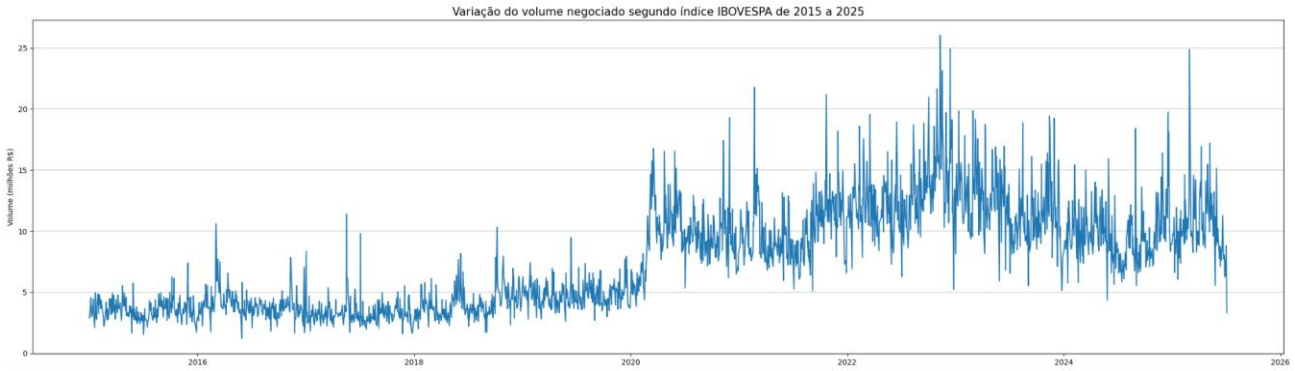
Análise Exploratória dos dados

Variação



Analisando a variação dos valores de fechamento de um dia para o outro, podemos observar os dias que houveram maior oscilação do índice. Notou-se que, os picos máximos e mínimos foram atingidos no período de 2020, muito provavelmente em decorrência da crise ocasionada pela pandemia.

Volume



O volume de negociações do IBOVESPA apresentou um aumento significativo a partir de 2020 devido a uma combinação de fatores. A pandemia de COVID-19 gerou grande volatilidade nos mercados globais, levando a um aumento na atividade de compra e venda por parte de investidores que buscavam tanto proteger seus ativos quanto aproveitar oportunidades de preços. Além disso, a redução das taxas de juros no Brasil para mínimas históricas tornou a renda fixa menos atrativa, impulsionando a migração de investimentos para a bolsa de valores. O avanço da digitalização e a popularização de plataformas de investimento também facilitaram o acesso de novos investidores ao mercado de ações, contribuindo para o crescimento do volume transacionado.Isso demonstra a grande influência de fatores externos nas variações constantes do índice.

FECHAMENTO

Ao analisar o índice de fechamento do IBOVESPA de 2015 a 2025, os principais pontos notados incluem:

- **Volatilidade em 2020:** O ano de 2020 foi marcado por picos de oscilação, tanto máximos quanto mínimos, impulsionados pela crise da pandemia de COVID-19.
- **Tendência de Alta Pós-2020:** Após a forte queda em 2020, o índice demonstrou uma recuperação, com um aumento notável, beneficiado pela redução das taxas de juros no Brasil e a busca por investimentos mais rentáveis.
- **Influência de Fatores Externos:** A análise geral revela que o índice de fechamento do IBOVESPA é altamente sensível a eventos macroeconômicos e geopolíticos, que podem gerar grandes flutuações.
- **Crescimento do Volume de Negociações:** Paralelamente, houve um aumento significativo no volume de negociações, especialmente a partir de 2020, refletindo o maior interesse e participação de investidores no mercado de ações.

Série temporal

O que são Séries Temporais?

Uma série temporal é uma sequência de pontos de dados indexados (ou listados) em ordem cronológica. É utilizada para modelar e analisar dados que variam ao longo do tempo, como preços de ações, temperaturas diárias, vendas mensais etc. O objetivo principal da análise de séries temporais é entender a estrutura subjacente dos dados e fazer previsões futuras.

Série Temporal Estacionária vs. Não Estacionária

A estacionariedade é uma propriedade crucial na análise de séries temporais. Uma série temporal é considerada estacionária se suas propriedades estatísticas (média, variância e autocorrelação) permanecerem constantes ao longo do tempo. Isso significa que a distribuição de probabilidade da série não muda com o tempo.

As principais diferenças entre séries temporais estacionárias e não estacionárias são:

- **Série Temporal Estacionária:**
    - **Média Constante:** A média da série não muda ao longo do tempo.
    - **Variância Constante:** A variância (dispersão dos dados) permanece a mesma em diferentes períodos.
    - **Autocorrelação Constante:** A relação entre uma observação e suas observações anteriores (autocorrelação) é consistente ao longo do tempo.
    - **Previsibilidade:** São mais fáceis de modelar e prever, pois, seus padrões estatísticos são estáveis. Modelos como ARIMA (Autoregressive Integrated Moving Average) frequentemente exigem estacionariedade.

- **Série Temporal Não Estacionária:**
    - **Média Variável:** A média da série pode apresentar tendências (crescimento ou declínio) ao longo do tempo.
    - **Variância Variável:** A variância pode mudar, indicando que a volatilidade dos dados aumenta ou diminui com o tempo.
    - **Autocorrelação Variável:** A estrutura de dependência temporal muda ao longo do tempo, tornando difícil a identificação de padrões fixos.
    - **Previsibilidade:** Mais desafiadoras de modelar e prever diretamente. Muitas vezes, é necessário transformá-las em estacionárias (por exemplo, por diferenciação) antes de aplicar modelos tradicionais.

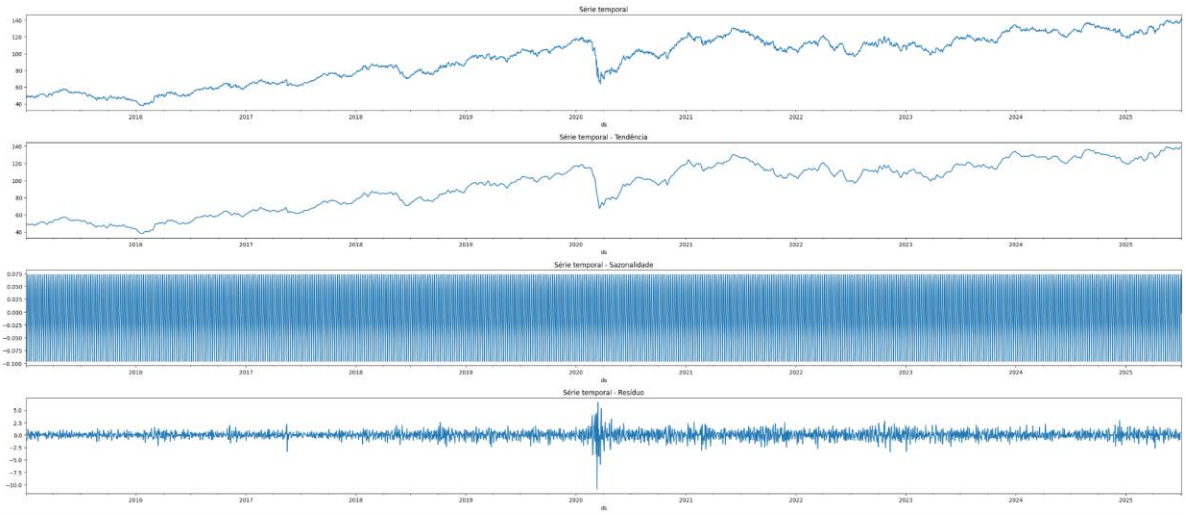
A identificação da estacionariedade é um passo fundamental na análise de séries temporais, pois afeta a escolha e a eficácia dos métodos de modelagem.

Analisando a série temporal

Utilizando a função **seasonal_decompose** da biblioteca **statsmodels**, foi realizada a decomposição da série temporal em três componentes principais:

- **tendência** (o padrão de longo prazo).
- **sazonalidade** (padrões que se repetem em períodos fixos)
- **resíduo** (o ruído aleatório ou o que resta após remover a tendência e a sazonalidade).

Ela ajuda a entender a estrutura subjacente dos dados temporais.



Observamos que a tendência é praticamente o mesmo comportamento da linha original dos dados, a sazonalidade aparentemente não pode ser demonstrada por não possuir um real padrão. Já com o ruído, podemos observar o quão atípico foi o ano de 2020, tornando o período praticamente imprevisível.

Analisando os gráficos, chegamos à hipótese de que a nossa série temporal possui características de uma série não estacionária. Para comprovar a hipótese, aplicamos o método de Teste de Dickey-Fuller Aumentado (ADF)

Teste de Dickey-Fuller Aumentado (ADF)

O Teste de Dickey-Fuller Aumentado (ADF) é um teste estatístico de raiz unitária usado para determinar se uma série temporal é estacionária. Ele testa a hipótese nula de que uma raiz unitária está presente na série (ou seja, a série não é estacionária) contra a hipótese alternativa de que a série é estacionária. Um valor p abaixo de um nível de significância (comumente 0.05) indica que podemos rejeitar a hipótese nula e considerar a série estacionária.

Utilizando a função **adfuller** da biblioteca **statsmodels,** realizamos o teste em nossa série:

Teste ADF
Teste estatístico: -1.2941764765333532
P-value: 0.6318323533706444

com o P-value de 0.63, comprovamos que a série não é estacionária

Tentaremos encontrar uma estacionaridade na nossa série utilizando algumas técnicas estatísticas.

Média Móvel e Desvio Padrão

Calculamos a **média móvel/desvio padrão** utilizando o método Rolling() do Pandas para criar "janelas deslizantes" de um tamanho especificado sobre nossa série temporal. Isso permite calcular estatísticas agregadas (como média, soma, desvio padrão) para cada janela, facilitando a análise de tendências e padrões locais nos dados.

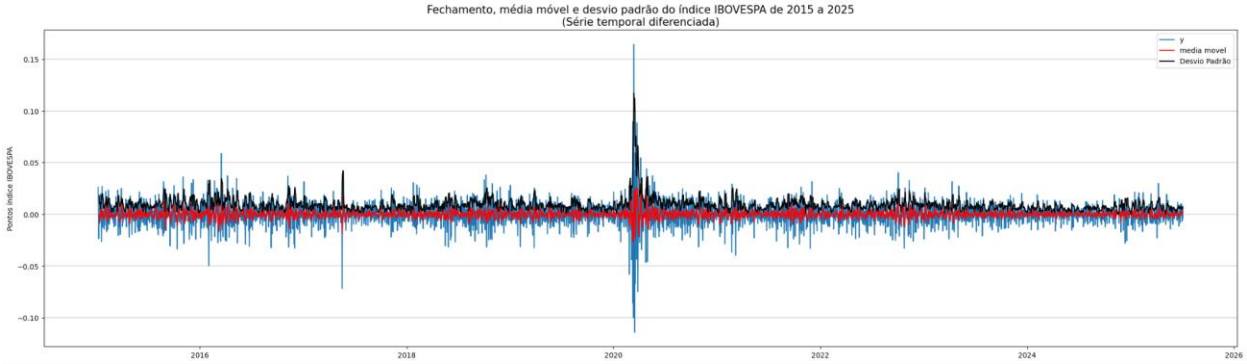
Média móvel:

Transformação logarítmica:

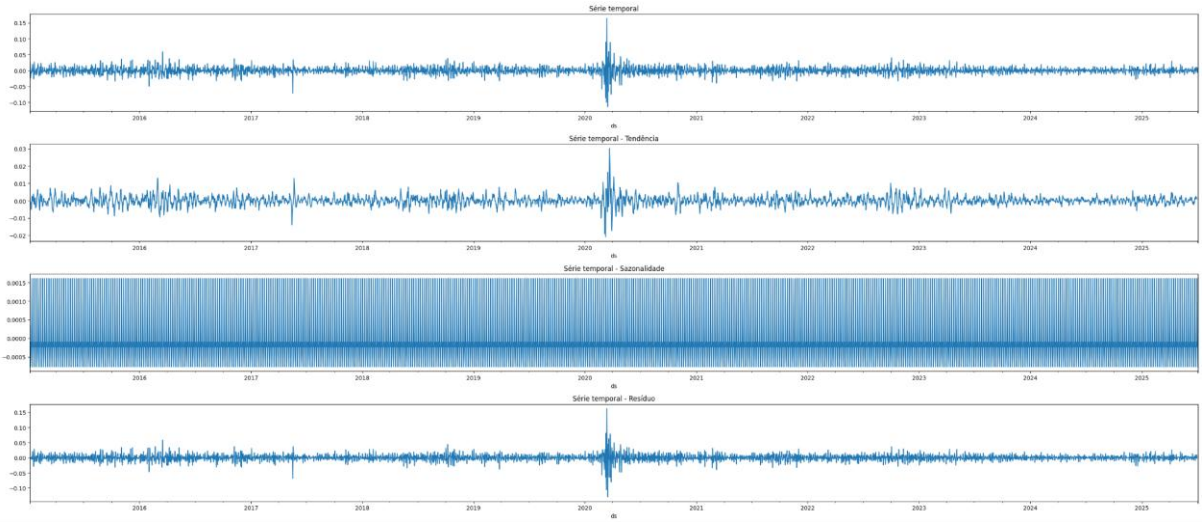
Nota-se que a média móvel mantém a estrutura da linha original, mesmo realizando a transformação logarítmica, o que ainda a caracteriza como não estacionária.

Também calculamos a diferença entre a série logarítmica e a média móvel logarítmica e, a partir dos resultados, utilizamos a função **.diff()** para calcular a diferença entre as observações consecutivas da série temporal. Essa técnica é comumente empregada para remover tendências e sazonalidades, ajudando a tornar a série em uma série estacionária, tornando-a mais adequada para modelagem.

Série:



decomposição da série diferenciada:



Nota-se agora um comportamento estacionário na série, o que se comprova através do teste ADF.

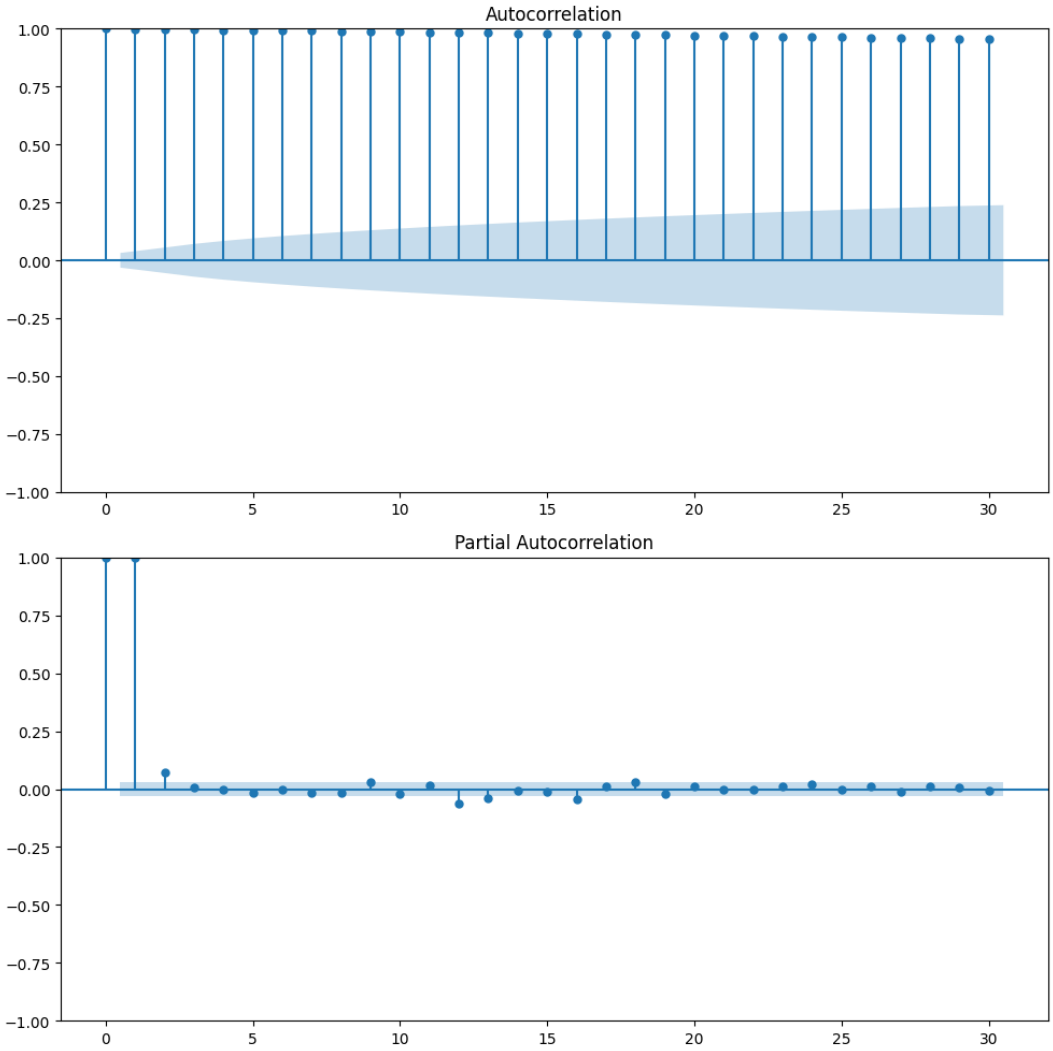
Teste ADF

Teste estatístico: -18.504429954015507

P-value: 2.1184938677057453e-30

Com o P-value abaixo de 5%, confirmamos que agora a série é estacionária.

Com as séries temporais estacionárias obtidas a partir da diferenciação da série original, será verificada a autocorrelação dos dados através dos métodos ACF (Autocorrelation Function) e PACF (Partial Autocorrelation Function) da biblioteca **statsmodels**.



analisando os gráficos onde, o valor 1 representa uma alta correlação, podemos observar que na autocorrelação parcial, apenas a primeira defasagem é estatisticamente significativa, indicando que a correlação entre a série e suas defasagens anteriores se torna insignificante após a primeira defasagem.

Com a análise, podemos concluir que a série possui uma autocorrelação baixa e que a influência de observações passadas diretas é mais forte e predominante.

Os resultados da análise de série temporal confirmam que a série do IBOVESPA não é estacionária, conforme demonstrado pelo alto P-value no Teste de Dickey-Fuller Aumentado (ADF) na série original. A decomposição da série revelou uma tendência clara e a ausência de sazonalidade perceptível, com o resíduo destacando a atipicidade e imprevisibilidade do período de 2020.

Conclusão da análise

A aplicação de técnicas como a média móvel e a transformação logarítmica não foi suficiente para estacionarizar a série, evidenciando a complexidade dos dados. Somente a diferenciação da série original conseguiu alcançar a estacionariedade, com um P-value significativamente baixo no teste ADF subsequente.

A análise das funções de autocorrelação (ACF) e autocorrelação parcial (PACF) na série diferenciada mostrou que apenas a primeira defasagem é estatisticamente significativa na autocorrelação parcial, indicando uma baixa autocorrelação geral e que a influência de observações passadas diretas é a mais predominante.

Dada a não estacionariedade inicial e a baixa autocorrelação após a diferenciação, métodos de análise e previsão de séries temporais que pressupõem estacionariedade, como modelos **ARIMA** (Autoregressive Integrated Moving Average) sem diferenciação ou com ordens de médias móveis (MA) e autorregressivas (AR) mais elevadas, seriam inadequados ou exigiram transformações significativas para serem aplicados de forma eficaz. A complexidade e a baixa autocorrelação após a diferenciação também sugerem que modelos mais simples podem ser insuficientes para capturar toda a dinâmica da série. Portanto, a escolha de um modelo como o **XGBoost**, que não se baseia em pressupostos de estacionariedade e pode lidar com a complexidade dos dados de mercado, é justificada.

Aplicando o modelo de Machine Learning

Modelos de Machine Learning

Modelos de Machine Learning são algoritmos matemáticos e estatísticos que permitem que computadores "aprendam" a partir de dados, sem serem explicitamente programados para cada tarefa. O objetivo é que o modelo identifique padrões nos dados, faça previsões ou tome decisões. Esse aprendizado pode ser supervisionado (com dados rotulados), não supervisionado (com dados não rotulados) ou por reforço (aprendendo através de tentativa e erro).

XGBoost

O XGBoost (eXtreme Gradient Boosting) é um algoritmo de aprendizado de máquina otimizado e escalável que pertence à família dos algoritmos de "boosting". Ele é amplamente utilizado em problemas de classificação e regressão devido à sua eficiência, flexibilidade e alta performance.

Características do XGBoost:

- **Algoritmo de Boosting:** O XGBoost constrói uma sequência de modelos de aprendizado, onde cada novo modelo tenta corrigir os erros dos modelos anteriores.

Ele combina previsões de vários "modelos fracos" (geralmente árvores de decisão) para criar um "modelo forte".

- **Regularização:** Inclui termos de regularização (L1 e L2) para evitar overfitting. Isso significa que o modelo é penalizado por ser muito complexo, o que ajuda a generalizar melhor para novos dados.
- **Tratamento de Dados Ausentes:** O XGBoost possui uma forma integrada de lidar com valores ausentes, aprendendo a melhor direção (ramificação) para instâncias com dados faltantes.
- **Otimização para Desempenho:** É otimizado para ser rápido e eficiente em termos de computação. Ele suporta computação paralela e distribuída, o que o torna adequado para grandes conjuntos de dados.
- **Flexibilidade:** Pode ser usado para problemas de classificação, regressão e ranqueamento.
- **Feature Importance:** Permite a visualização da importância de cada característica (feature) no modelo, auxiliando na interpretabilidade.

Como o XGBoost funciona:

O funcionamento do XGBoost baseia-se na ideia de "gradient boosting", que constrói modelos aditivos de forma sequencial.

1. **Modelo Inicial:** Começa com um modelo inicial simples (geralmente uma previsão de valor constante).
2. **Cálculo dos Resíduos:** Calcula os "resíduos" (diferença entre as previsões do modelo atual e os valores reais). Esses resíduos representam os erros que o modelo precisa corrigir.
3. **Treinamento de uma Nova Árvore:** Treina um novo modelo fraco (uma árvore de decisão) para prever esses resíduos. Ou seja, a árvore tenta aprender a corrigir os erros do modelo anterior.
4. **Adição da Árvore ao Modelo:** A previsão da nova árvore é adicionada ao modelo existente, mas com um "fator de aprendizado" (learning rate) para controlar a contribuição de cada nova árvore. Isso ajuda a prevenir o overfitting e permite que o modelo aprenda gradualmente.
5. **Iteração:** Os passos 2 a 4 são repetidos por um número pré-definido de iterações, ou até que a performance do modelo pare de melhorar. A cada iteração, o modelo se torna mais preciso ao corrigir os erros acumulados.
6. **Previsão Final:** A previsão final do XGBoost é a soma das previsões de todas as árvores individuais construídas durante o processo.

Feature Engineering

Foram criadas as seguintes variáveis derivadas (features):

1. **Retorno_Diario**: Variação percentual diária
2. **MM_3, MM_5, MM_10, MM_15**: Médias móveis dos fechamentos
3. **Volatilidade_5**: Desvio padrão dos últimos 5 fechamentos
4. **Spread_Day**: Diferença entre máxima e mínima do dia
5. **Fechamento_Abertura**: Diferença entre o fechamento e a abertura
6. **Target**: 1 se o fechamento do dia seguinte for maior que o atual, 0 caso contrário.

Testando o modelo

Para otimizar o desempenho do XGBoost, foi utilizada a técnica GridSearchCV, que realiza uma busca exaustiva pelas melhores combinações de hiperparâmetros. O modelo foi testado com diferentes valores para learning_rate': 0.05, 'max_depth': 10, 'n_estimators': 150, bem como regularização L1 (alpha: 0) e L2 (lambda: 1). A validação cruzada foi realizada com cv=3 e a métrica de avaliação escolhida foi a acurácia.

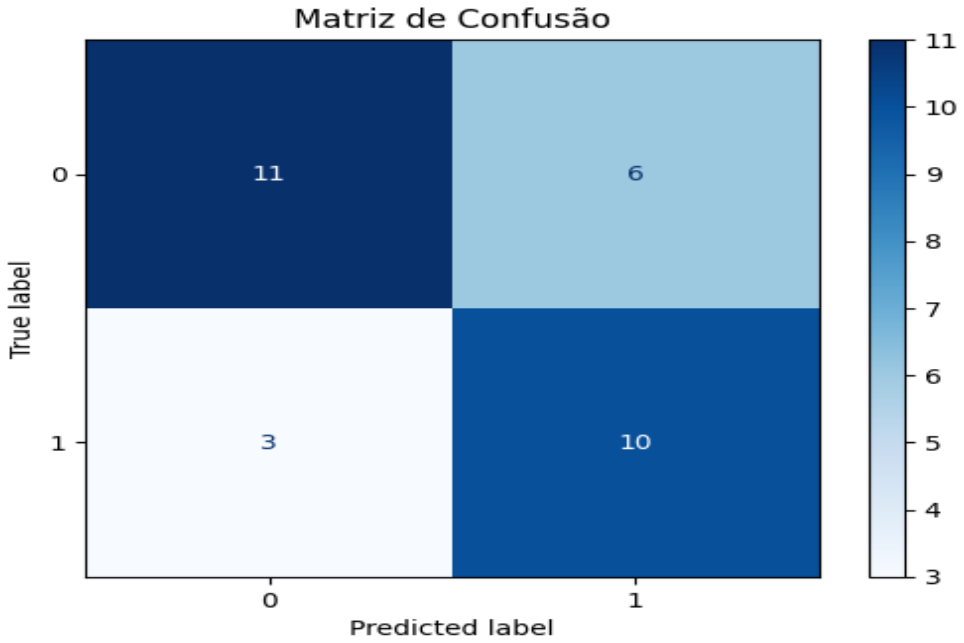
L1 e L2 são técnicas de regularização usadas em modelos de Machine Learning para evitar o overfitting (quando o modelo se ajusta demais aos dados de treino e perde capacidade de generalização). Elas adicionam penalidades aos coeficientes do modelo durante o treinamento, forçando o modelo a ser mais simples e robusto.

O modelo final, com os parâmetros ideais, foi então testado sobre os últimos 30 dias de dados, conforme exigido no desafio. A escolha por utilizar os dados mais recentes como conjunto de teste também se alinha à prática comum em séries temporais, já que, quanto mais distante a previsão, maior a incerteza envolvida — fenômeno conhecido como “efeito de cone”, onde a variabilidade dos possíveis cenários futuros aumenta com o horizonte de tempo.

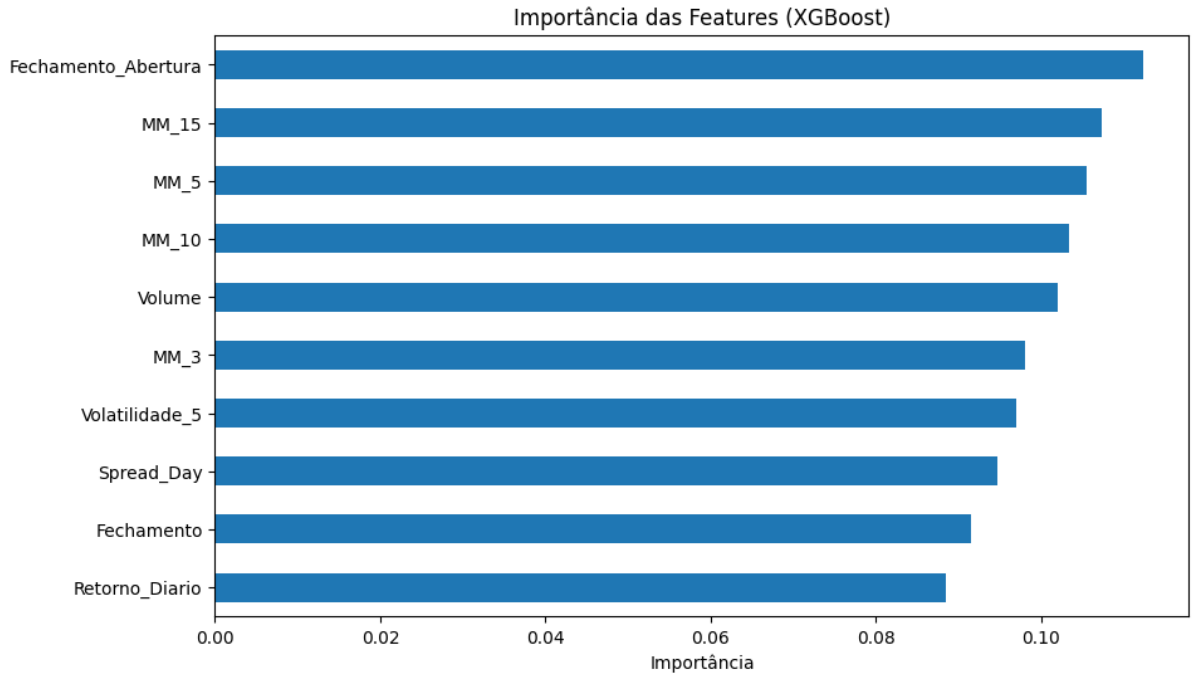
Resultados do modelo XGBoost:
- Acurácia nos últimos 30 dias: 70%

Métricas detalhadas:

Classe 0 (queda): Precision 0.79, Recall 0.65, F1-Score 0.71
Classe 1 (alta): Precision 0.62, Recall 0.77, F1-Score 0.69



Demonstração gráfica da importância das Features do modelo



A imagem abaixo mostra o comportamento real do IBOVESPA no dia **07/07/2025**, com fechamento em queda de **-1,26%**.

07.07.2025	139.490	141.265	141.342	139.295	6,12B	-1.26%
04.07.2025	141.264	140.928	141.564	140.597	3,31B	+0.24%

O modelo previu corretamente a direção da movimentação com base nas informações do dia anterior (04/07/2025), com o treinamento sendo realizado nos últimos 30 dias para teste, assim, validando sua eficácia mesmo em cenários atuais.

Conclusão

Através deste estudo, foi possível construir um modelo preditivo robusto para o índice IBOVESPA, aplicando conceitos de séries temporais e aprendizado de ML. Apesar da complexidade dos dados e da não estacionariedade da série original, o uso de técnicas adequadas como a diferenciação, Média Móvel, Feature Engineering, GridSearchCV para os melhores hiperparâmetros e o modelo XGBoost permitiu alcançar uma acurácia de 70% na previsão de movimento diário do índice, com resultados satisfatórios em métricas como precisão e recall. Outros modelos utilizados na análise como: Regressão Logística, Random Forest, KNN e Regressão Linear não performaram de forma positiva ficando abaixo de 60% de acurácia.

A escolha do XGBoost mostrou-se acertada frente à volatilidade do mercado e à natureza não linear dos dados financeiros, além de fornecer insights importantes por meio da análise de importância das variáveis (features). O modelo demonstrou capacidade de generalização ao prever corretamente o comportamento do mercado em datas recentes.

Como trabalho futuro, seria interessante incorporar variáveis macroeconômicas externas (como câmbio, juros, inflação e indicadores globais) e testar modelos de deep learning, como redes neurais recorrentes (RNNs ou LSTMs), que podem capturar ainda mais os padrões sequenciais dos dados.

O projeto evidencia o potencial da inteligência artificial aplicada ao mercado financeiro, oferecendo ferramentas para apoiar investidores na tomada de decisão com base em dados e algoritmos inteligentes.