



**FIAP
TECH CHALLENGE - Fase 2
Data Analytics**

**CARINA SILVA BARROS (RM 366941)
FABIANA MARÍA PEREIRA (RM 366933)
GEORGIANA GLISILLI IVEN LAW (RM 366926)
PAULO ROBERTO PETRILLO (RM 366896)
VIVIAN HADDAD (RM 366923)**

RESUMO

Este relatório apresenta uma análise exploratória sobre os dados históricos do índice IBOVESPA, disponíveis publicamente na página web da *INVESTING.COM*. O objetivo é desenvolver um modelo preditivo capaz de prever se o índice IBOVESPA fechará em alta ou baixa no dia seguinte, com base em dados históricos do próprio índice. Foram desenvolvidas as principais etapas para criação de modelo preditivo, a saber:

1. Coleta de dados;
2. tratamento de dados;
3. escolha e treinamento do modelo;
4. otimização de parâmetros e balanceamento de dados.
5. verificação e validação dos modelos por meio de métricas.

Essas etapas estão descritas com mais detalhes a seguir neste relatório.

1 AQUISIÇÃO E EXPLORAÇÃO DOS DADOS

1.1 CARREGAMENTO E TRATAMENTO DOS DADOS

Conforme requisitos do problema, foram baixados da seguinte página web <https://br.investing.com/indices/bovespa-historical-data> dados diários da Bovespa no período de 03/02/2020 até 01/07/2025. Esses dados foram armazenados no formato de planilha eletrônica, com extensão .x/s. Estes consideram apenas os dias úteis de pregão da bolsa de valores, gerando assim 1346 linhas na planilha. O que representa 5 anos e 4 meses de dados onde cada ano tem aproximadamente 252 dias de pregão.

Os campos/atributos dessa base de dados são os seguintes mostrados na Figura 1.

O nome dos campos foram alterados para adequar a nomenclatura, retirando abreviações e acentuações, assim como os tipos dos campos Volume e Variacao foram transformados em float64.

O formato final dos dados está na Figura 2

1.2 VISUALIZAÇÕES PRÉVIAS

Serão explorados diferentes tipos de gráficos para auxiliar na compreensão qualitativa dos dados.

1.2.1 GRÁFICO DE PONTOS

A Figura 3 mostra a evolução diária do fechamento de pontos da bolsa de valores.

```

→ <class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1346 entries, 2025-07-01 to 2020-02-03
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  --          -----          ----- 
 0   Último       1346 non-null   float64
 1   Abertura    1346 non-null   float64
 2   Máxima       1346 non-null   float64
 3   Mínima       1346 non-null   float64
 4   Vol.         1346 non-null   object  
 5   Var%         1346 non-null   object  
dtypes: float64(4), object(2)
memory usage: 73.6+ KB

```

Figura 1 – Dados brutos baixados

```

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1346 entries, 2020-02-03 to 2025-07-01
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  --          -----          ----- 
 0   Fechamento  1346 non-null   float64
 1   Abertura    1346 non-null   float64
 2   Maxima       1346 non-null   float64
 3   Minima       1346 non-null   float64
 4   Volume       1346 non-null   float64
 5   Variacao     1346 non-null   float64
dtypes: float64(6)
memory usage: 73.6 KB

```

Figura 2 – Dados brutos tratados

1.2.2 GRÁFICOS BOXPLOT

A seguir na Figura 4 podemos comparar qualitativamente a distribuição dos dados. Para isso, os valores de todos os atributos foram normalizados igualando a média à zero.

Pode-se concluir a partir da análise qualitativa do gráfico da Figura 4 que as variáveis Fechamento, Abertura, Máxima e Mínima são distribuições bem parecidas (caixas centradas no zero), contrastando com a distribuição do Volume que tem grande quantidade de outliers positivos e da Variação que tem uma distribuição menos espalhada (caixa mais fina) e muitos outliers acima e abaixo da caixa.

1.2.3 HISTOGRAMAS

Na Figura 5 verificamos que a distribuição da variável Amplitude Diária assemelha-se a uma distribuição normal com valores tendendo à esquerda, ou seja, com valores

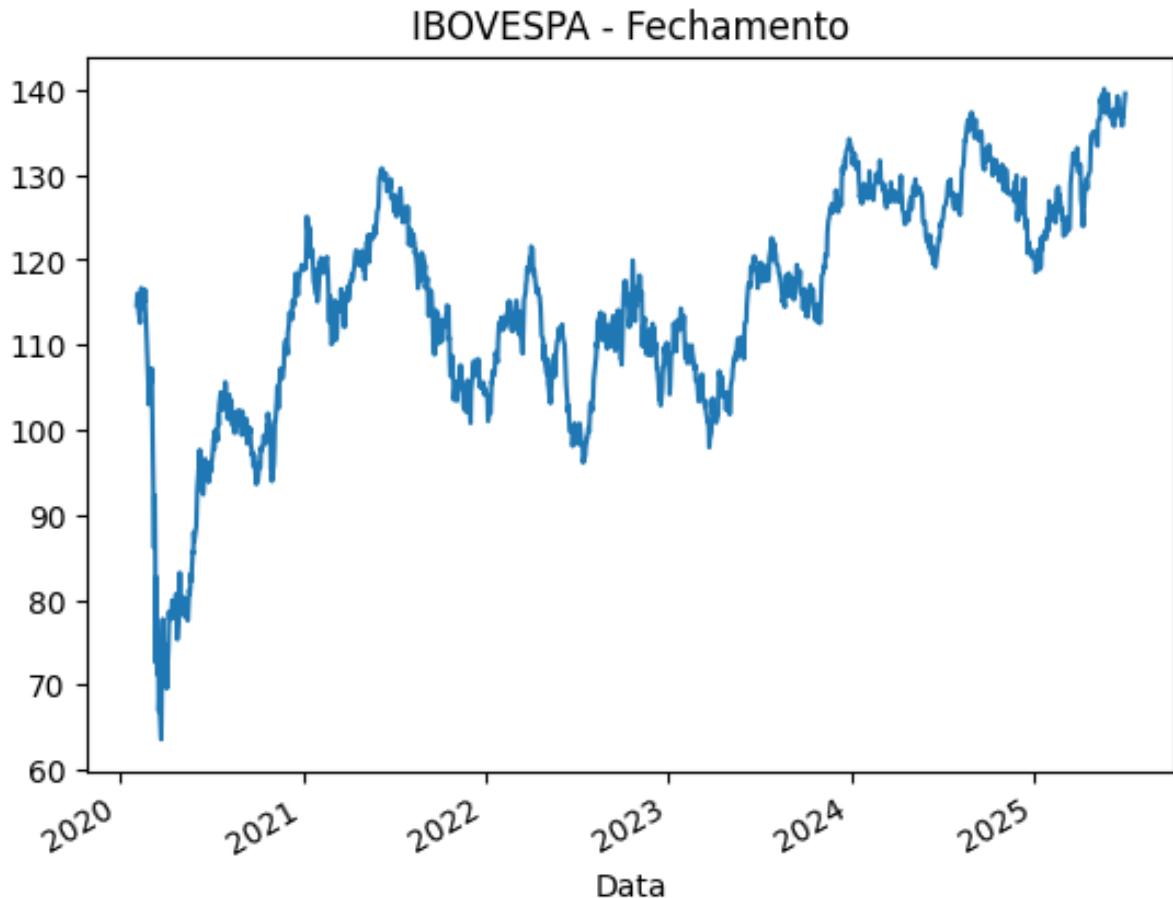


Figura 3 – Fechamento Diário

positivos. De fato, isso indica que há uma evolução positiva ao longo do tempo. O histograma da distribuição de fechamento – abertura mostra-se bem centrado no valor zero, com caudas mais alongadas que a Amplitude e com poucos valores extremos, acima de +5 ou -5. O mesmo ocorre para a Variação diária, porém com valores ainda mais concentrados em torno de zero.

Com essas informações podemos auferir qualitativamente que o IBOVESPA é relativamente estável no dia a dia, oscila moderadamente dentro do pregão e tem caudas longas (dias de crise ou euforia). Isso está de acordo com o observado no decorrer dos anos, onde o crescimento (e eventuais quedas) ocorre ao longo de grande quantidade de dias, apesar da grande variação diária.

1.3 DECOMPOSIÇÃO DA SÉRIE DO IBOVESPA

A partir das informações das etapas anteriores, sabendo que os dados formam uma série temporal, vamos tentar buscar as principais características dessa série, ou seja, vamos usar ferramentas para decompor a série e evidenciar suas principais propriedades temporais. Realizamos uma decomposição temporal, separando a série original em três componentes: **tendência, sazonalidade e resíduos**.

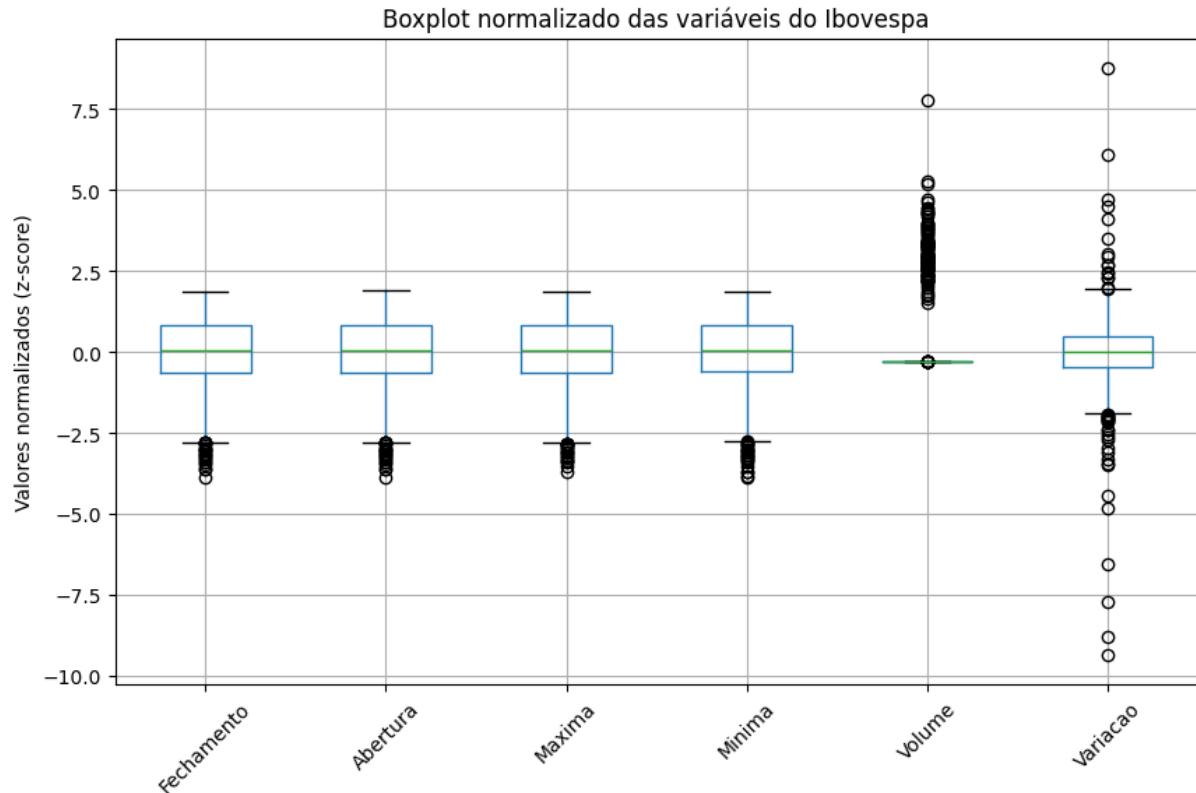


Figura 4 – Boxplot normalizado

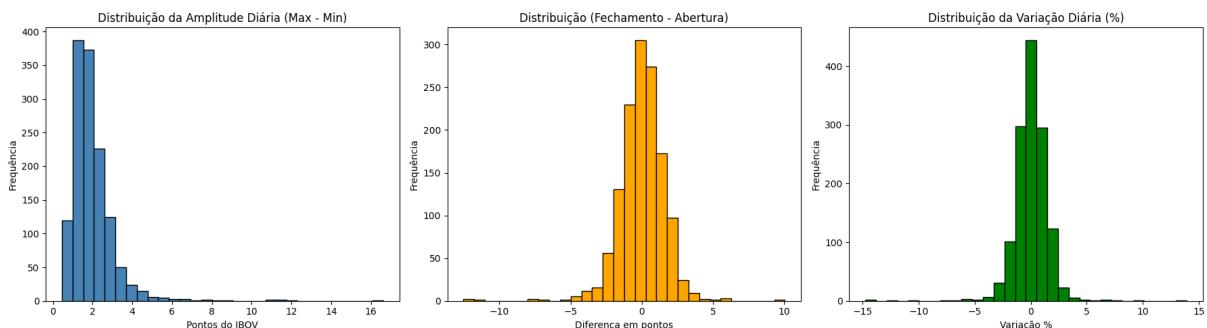


Figura 5 – Histogramas

A decomposição foi feita com periodicidade de 21 dias úteis, aproximadamente equivalente a um mês de negociações. A Figura 6 apresenta os quatro gráficos gerados: a série original de fechamento, a tendência de longo prazo, os padrões sazonais e os resíduos da série.

A análise da tendência mostrou que, ao longo do período observado, o IBOVESPA passou por ciclos de valorização e desvalorização bem definidos – com uma trajetória de alta entre 2020 e 2022, seguida por uma queda e posterior recuperação em 2025. Esse padrão indica que o índice responde a movimentos macroeconômicos e conjunturais de longo prazo.

A componente sazonal evidenciou a presença de padrões cíclicos mensais, com oscilações regulares que se repetem aproximadamente a cada 21 dias úteis. Esses

Decomposição da Série do IBOVESPA

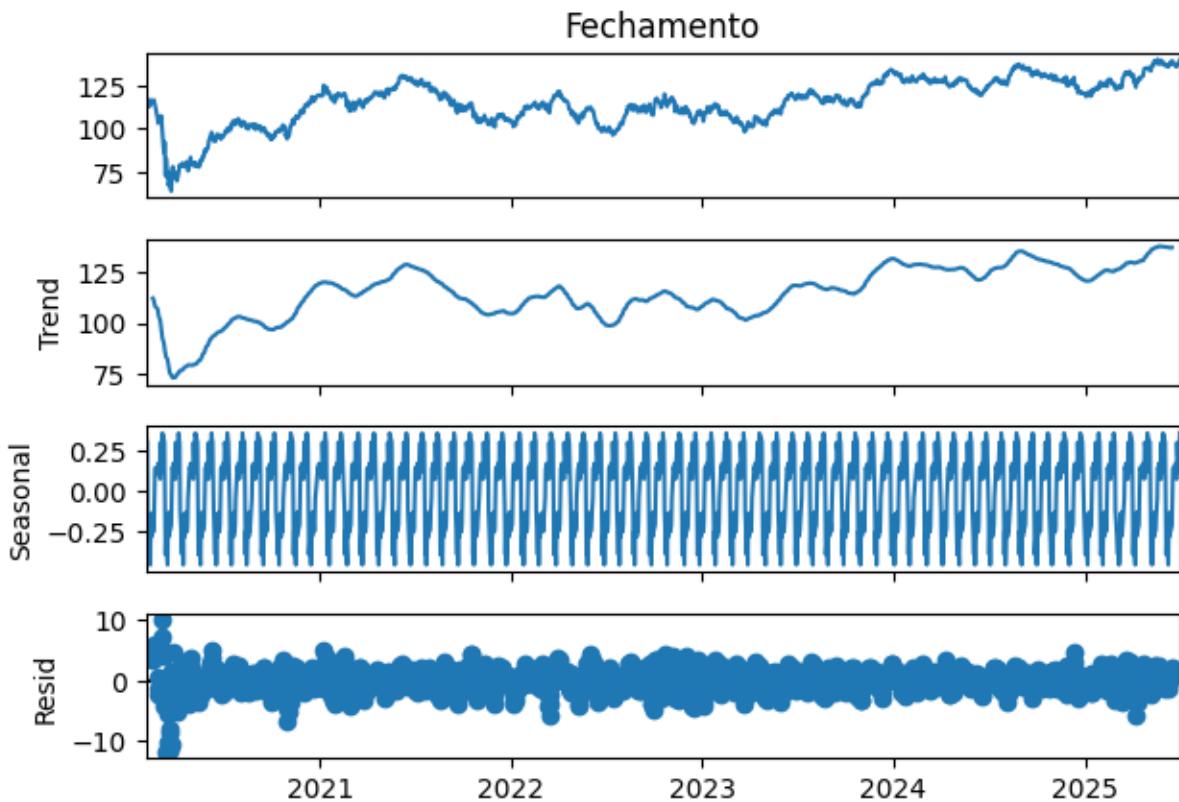


Figura 6 – Decomposição da Série do IBOVESPA

movimentos podem estar associados a eventos recorrentes no mercado financeiro, como vencimentos de contratos, divulgação de indicadores econômicos e ajustes de portfólio por investidores institucionais.

Por fim, os resíduos apresentaram distribuição aleatória e simétrica em torno de zero, o que sugere que os efeitos da tendência e da sazonalidade foram bem capturados pelo modelo. Os ruídos restantes refletem eventos imprevisíveis, como crises políticas, anúncios inesperados ou choques externos, que não seguem padrões regulares. Do ponto de vista teórico sobre séries, esses valores são os responsáveis pelo acréscimo ou decréscimo dos valores da série.

1.4 ANÁLISE DE ESTACIONARIEDADE TEMPORAL – TESTE DICKEY-FULLER AUMENTADO (ADF)

Para avaliar a estacionariedade da série temporal dos preços de fechamento do IBOVESPA, foi aplicado o Teste de *Dickey-Fuller Aumentado*, ADF na sigla em inglês. Esse teste estatístico tem como objetivo verificar a presença de raiz unitária na série, que indicaria um comportamento não estacionário, ou seja, com tendência ou variância não constante ao longo do tempo. Ao aplicar o teste com o comando `adfuller(adfuller(coluna_fechamento))` obteve-se os valores mostrados em Figura 7.

```
Resultado do Teste de Dickey-Fuller (ADF):
ADF Statistic: -1.9995
p-value: 0.2868
Valores críticos:
1%: -3.4352
5%: -2.8637
10%: -2.5679
```

Figura 7 – teste Dickey-Fuller Aumentado (ADF)

A interpretação desses valores segue a lógica das hipóteses estatísticas:

- Hipótese nula(H_0) : a série possui raiz unitária → não estacionária.
- Hipótese alternativa(H_1) : a série é estacionária.

Como o p-valor é superior ao nível de significância de 0,05, não há evidências suficientes para rejeitar a hipótese nula. Além disso, a estatística ADF não é menor que nenhum dos valores críticos apresentados, nem mesmo no nível de 10%. Isso confirma que a série não é estacionária, apresentando variações ao longo do tempo que não se mantêm constantes.

- Se $p\text{-value} \leq 0,05$, não rejeitamos H_0 .
- Se $p\text{-value} > 0,05$, rejeitamos H_0 em favor de H_1 .

Esse resultado é coerente com o comportamento esperado de séries financeiras, como o índice IBOVESPA, que frequentemente seguem um passeio aleatório (random walk). Tal característica implica que os preços se movem de forma imprevisível, influenciados por fatores externos e eventos de mercado, sem apresentar padrões fixos ou repetitivos.

1.5 ANÁLISE DE AUTOCORRELAÇÃO E AUTOCORRELAÇÃO PARCIAL

Para aprofundar a avaliação da estrutura da série temporal dos preços de fechamento do IBOVESPA, foram gerados os gráficos de Autocorrelação (ACF) e Autocorrelação Parcial (PACF), conforme a Figura 8.

O gráfico de ACF revela que os valores de autocorrelação permanecem elevados em praticamente todos os lags analisados (até o lag 30), com um decaimento lento — partindo de aproximadamente 1.0 e reduzindo gradualmente até cerca de 0.75. Esse padrão indica que o valor atual da série está fortemente correlacionado com os valores passados, como os de ontem, anteontem, e assim por diante.

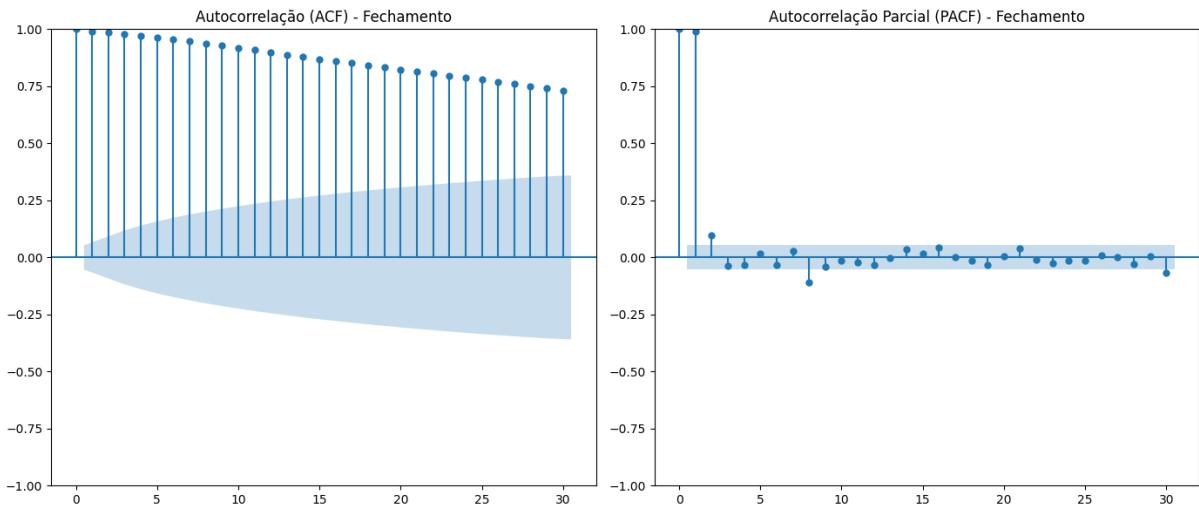


Figura 8 – Autocorrelação e Autocor. Parcial

Esse comportamento é típico de séries não estacionárias, nas quais os valores se acumulam ao longo do tempo, formando uma trajetória com tendência — característica conhecida como passeio aleatório (random walk).

O gráfico de PACF mostra uma correlação extremamente alta no primeiro lag (lag = 1), seguida por uma queda abrupta para valores próximos de zero nos lags subsequentes. Isso indica que o valor atual da série depende diretamente apenas do valor imediatamente anterior, sem influência significativa dos demais lags.

Esse padrão também é típico de séries não estacionárias, especialmente em dados financeiros, como preços de ativos, que tendem a reagir fortemente ao valor do dia anterior.

Assim, concluímos que pela análise dos gráficos de ACF e PACF reforça o resultado obtido no Teste de Dickey Fuller Aumentado (ADF): a série de preços de fechamento do IBOVESPA não é estacionária. O decaimento lento da ACF é um sinal clássico de presença de raiz unitária, enquanto a PACF confirma que a dependência direta está concentrada no primeiro lag.

2 TRATAMENTO DE DADOS

Nas próximas seções, definiremos a variável target e faremos o tratamento de dados necessário para rodar os modelos.

2.1 DEFINIÇÃO DA VARIÁVEL ALVO (TARGET)

Para transformar o problema em uma tarefa de classificação supervisionada, foi criada a variável binária **Target**, que indica se o índice IBOVESPA apresentou alta ou baixa em relação ao dia anterior. Essa variável será utilizada como rótulo para o treinamento dos modelos preditivos. A lógica aplicada foi: compara-se o valor de fechamento do dia atual com o do dia anterior. Se o fechamento de hoje for maior que o

de ontem, o valor da variável Target é igual a 1 (indicando alta); caso contrário, é igual a 0 (indicando baixa).

A Figura 9 a seguir ilustra os primeiros registros da variável Target:

Data	Hoje	Ontem	Target
2020-02-03	114.629	NaN	0
2020-02-04	115.557	114.629	1
2020-02-05	116.028	115.557	1
2020-02-06	115.190	116.028	0
2020-02-07	113.770	115.190	0
2020-02-10	112.570	113.770	0
2020-02-11	115.371	112.570	1
2020-02-12	116.674	115.371	1
2020-02-13	115.662	116.674	0
2020-02-14	114.381	115.662	0

Figura 9 – Variável Target

2.2 SELEÇÃO DE VARIÁVEIS EXPLICATIVAS

Cada registro representa um pregão e inclui variáveis como:

- Fechamento: valor de encerramento do índice no dia;
- Abertura, Máxima, Mínima: preços de abertura e extremos do dia;
- Volume: volume financeiro negociado;
- Variação: percentual de variação em relação ao dia anterior;
- Amplitude: diferença entre máximo e mínima;
- Fech_Abert: diferença entre fechamento e abertura;
- Target: variável binária indicando se o fechamento subiu (1) ou caiu (0) em relação ao pregão anterior.

2.3 BALANCEAMENTO DA VARIÁVEL TARGET

A Figura 10 representa o conjunto de dados que contempla informações diárias do índice IBOVESPA no período de 03/02/2020 a 01/07/2025, abrangendo mais de cinco anos de histórico de mercado.

Observa-se que esse balanceamento é positivo para a modelagem, pois evita viés excessivo para uma das classes e permite que algoritmos de classificação aprendam de forma mais equitativa.

Período: 2020-02-03 -> 2025-07-01
Balanceamento do Target:
Target
1 0.513
0 0.487
Name: proportion, dtype: float64
Fechamento Abertura Maxima Minima Volume Variacao Amplitude Fech_Abert Target
Data
2020-02-03 114.629 113.761 115.299 113.467 5510000.0 0.76 1.832 0.868 0
2020-02-04 115.557 114.631 116.556 114.631 5830000.0 0.81 1.925 0.926 1
2020-02-05 116.028 115.563 117.701 115.562 7170000.0 0.41 2.139 0.465 1
2020-02-06 115.190 116.033 117.382 114.723 7380000.0 -0.72 2.659 -0.843 0
2020-02-07 113.770 115.190 115.190 113.769 6590000.0 -1.23 1.421 -1.420 0

Figura 10 – Balanceamento do Target

3 RETORNO DIÁRIO EM %

O retorno diário representa a variação percentual do preço de fechamento de um dia em relação ao dia anterior. A criação da variável foi realizada por meio da função 'pct_change()', que calcula a diferença percentual entre valores consecutivos da série de fechamento. O resultado foi multiplicado por 100 para expressar o retorno em termos percentuais.

Interpretação prática:

- Um valor de +2.5 indica que o índice subiu 2,5% em relação ao dia anterior;
- - Um valor de -1.3 indica que o índice caiu 1,3% no dia.

Essa variável captura a dinâmica de curto prazo do mercado e é utilizada como feature explicativa na modelagem preditiva. O retorno diário permite identificar padrões de reversão, continuidade de tendência e momentos de alta volatilidade — todos elementos cruciais para a tomada de decisão quantitativa.

3.1 LAGS DOS RETORNOS (1, 2 E 3 DIAS ATRÁS)

Para que o modelo apresente memória de curto prazo, foram incluídos retornos passados como variáveis explicativas. Essa abordagem permite que o modelo aprenda padrões temporais e reconheça que o comportamento recente do ativo pode influenciar seu desempenho atual. A técnica utilizada para isso foi a função `shift(n)`, que desloca a série temporal n dias para baixo, criando colunas com os valores defasados.

Interpretação prática:

1. Lag 1 representa o retorno de ontem.
2. Lag 2 representa o retorno de dois dias atrás.
3. Lag 3 representa o retorno de três dias atrás.

Essas variáveis funcionam como uma espécie de “memória histórica” para o modelo. Ao incluí-las, observamos que **“o que aconteceu nos últimos dias pode influenciar o que acontece hoje”**.

3.2 MÉDIAS MOVEIS

As médias móveis foram utilizadas para suavizar as flutuações diárias dos preços e ajudaram a identificar tendências ao longo do tempo. Elas funcionaram como uma régua de referência: ao comparar o preço atual com sua média histórica, foi possível inferir se o ativo está em uma fase de alta ou baixa.

Para a apuração, foi utilizada a seguinte escala:

- MM5 → tendência de curtíssimo prazo;
- MM5 → tendência de curto prazo;
- MM20 (20 dias) → tendência de curto prazo (1 mês);
- MM50 (50 dias) → cerca de 2,5 meses úteis → tendência de médio prazo;
- MM100 (100 dias) → cerca de 5 meses úteis → tendência de longo prazo.

Interpretação prática:

- Se o preço atual > que a média móvel, o ativo está em tendência de alta;
- Se o preço atual < que a média móvel, o ativo está em tendência de baixa

3.3 DISTÂNCIAS RELATIVAS ENTRE O PREÇO DE FECHAMENTO E AS MÉDIAS MÓVEIS

Nesta etapa da análise, foram calculadas as distâncias percentuais entre o preço de fechamento do ativo e suas respectivas médias móveis de 20, 50 e 100 períodos.

Esses indicadores avaliaram o posicionamento atual do preço em relação às tendências de curto, médio e longo prazo.

As métricas utilizadas foram:

- Distância relativa à MM20(%): representa o quanto o preço do fechamento está acima ou abaixo da média móvel de 20 dias;
- Distância relativa à MM50(%): indica a diferença percentual entre o preço de fechamento e a média móvel de 50 dias;
- Distância relativa à MM100(%): mede a distância percentual em relação à média móvel de 100 dias.

Interpretação prática:

- Valor positivo: o preço está acima da média móvel, sugerindo uma possível tendência de alta;
- Valor negativo: o preço está abaixo da média móvel, indicando uma possível tendência de baixa.

3.4 INCLINAÇÃO DAS MÉDIAS MÓVEIS (SLOPE)

Foi calculada a inclinação percentual diária das médias móveis de 20, 50 e 100 períodos, com objetivo de identificar a direção e a intensidade da tendência de cada uma delas ao longo do tempo.

A inclinação, também chamada de slope, representa a variação percentual da média móvel de um dia para o outro. Essa abordagem permitiu observar se a média móvel está em trajetória ascendente ou descendente.

Interpretação prática:

- Slope > 0 : indica que a média móvel está subindo, sugerindo uma tendência de alta; -
- Slope < 0 : indica que média móvel está caindo, sinalizando uma tendência de baixa.

3.5 VOLATILIDADE DE CURTO PRAZO (DESVIO PADRÃO DOS ÚLTIMOS 5 DIAS)

Nesta fase, foi calculada a volatilidade de curto prazo com base no desvio padrão dos retornos diárias dos últimos cinco pregões, com o objetivo de mensurar o grau de incerteza ou instabilidade recente do ativo, oferecendo uma leitura quantitativa da variação dos preços em um intervalo reduzido de tempo.

Interpretação prática:

- Volatilidade elevada: indica maior dispersão dos retornos, sugerindo um ambiente de maior risco ou instabilidade;
- Volatilidade baixa: sugere menor variação nos preços, refletindo um período de maior previsibilidade ou estabilidade.

4 INDICADORES TÉCNICOS.

A seguir serão apresentado indicadores técnicos relativos a bolsa de valores.

4.1 ÍNDICE DE FORÇA RELATIVA (RSI)

O RSI (Relative Strength Index) é um indicador técnico que foi utilizado para medir a força e a velocidade dos movimentos de preço, funcionando como um oscilador que varia entre 0 e 100, com o objetivo de identificar condições sobrecompra ou sobrevenda de um ativo, sinalizando possíveis pontos de reversão de tendência. A fórmula do RSI considerou a média dos ganhos e perdas dos últimos períodos (14 dias).

Interpretação prática:

- RSI abaixo de 30: o ativo pode estar em condição de sobrevenda, ou seja, após uma queda acentuada, há possibilidade de recuperação;
- RSI acima de 70: o ativo pode estar em condição de sobrecompra, indicando uma alta exagerada e possível correção.

Esse indicador pode identificar momentos de exaustão de tendência e antecipar movimentos de reversão.

4.2 MÉDIAS MÓVEIS EXPONENCIAIS (EMAS) E O INDICADOR MACD.

Essa ferramenta foi utilizada para atribuir maior peso aos preços mais recentes, tornando-os mais sensíveis às variações do mercado em comparação às médias móveis simples, capturando movimentos de preços com mais agilidade e identificando mudanças de tendências.

Nesta análise, foram utilizadas duas EMAs com períodos distintos:

- EMA 12: média móvel exponencial de 12 dias, que reage rapidamente às oscilações de curto prazo;
- EMA 26: média móvel exponencial de 26 dias, que suaviza os movimentos e reflete tendências mais estáveis.

A partir dessas duas médias, foi calculado o indicador MACD (Moving Average Convergence Divergence), definido como a diferença entre a EMA 12 e a EMA 26, com a seguinte interpretação:

- MACD positivo (EMA 12 > EMA 26): indica tendência de alta, com maior força compradora;
- MACD negativo (EMA 12 < EMA 26): sugere tendência de baixa, com predominância da força vendedora.

Além disso, foi calculada a linha de sinal (MACDsig), uma média exponencial de 9 períodos, que serviu como referência para identificar cruzamentos com o MACD, indicando pontos de entrada ou saída no mercado. Quando o MACD cruza acima da

linha de sinal, é considerado um sinal de compra; quando cruza abaixo, um sinal de venda.

4.3 OSCILADOR ESTOCÁSTICO (%K E %D)

O *Stochastic Oscillator* é um indicador técnico que avaliou a posição do preço de fechamento atual em relação à faixa de preços (mínimo e máximo) de um período recente, com o objetivo de identificar possíveis condições de sobrecompra ou sobre-venda, ajudando a antecipar reversões de tendência. Cálculo e Interpretação:

- %K (STOCHK): representa a posição relativa do fechamento atual dentro da faixa dos últimos 14 dias.

Valores típicos de interpretação:

- Próximo de 100: fechamento perto do topo da faixa possível sobrecompra;
 - Próximo de 0: fechamento perto do fundo da faixa → possível sobrevenda;
 - Entre 20 e 80: Zona neutra, sem sinal claro.
- • %D (STOCHK): é uma média móvel simples de 3 períodos aplicadas ao

Sinais técnicos: Os cruzamentos entre %K e %D são amplamente utilizados como gatilhos de entrada ou saída:

- %K cruza acima de %D: sinal de compra (momentum positivo).
- %K cruza abaixo de %D: sinal de venda (momentum negativo).

4.4 INDICADOR ATR (AVERAGE TRUE RANGE)

Esse indicador teve como foco a intensidade da oscilação dos preços, funcionando como um termômetro do risco e da instabilidade do mercado, medindo a volatilidade do ativo.

O ATR é baseado no conceito de True Range (TR), que considera três possíveis variações diárias:

- Diferença entre a máxima e a mínima do dia atual;
- Diferença entre a máxima do dia atual e o fechamento do dia anterior;
- Diferença entre a mínima do dia atual e o fechamento do dia anterior.

Interpretação prática:

- ATR alto: indica que o ativo está passando por dias de forte oscilação, com movimentos amplos ou gaps significativos. Reflete um mercado mais instável e arriscado;
- ATR baixo: sugere que o ativo está operando em um ambiente mais calmo e previsível, com variações diárias mais contidas.

4.5 FEATURES BINÁRIAS: CRUZAMENTOS DE MÉDIAS MÓVEIS E INDICADORES.

Foram criadas variáveis binárias que capturam sinais técnicos simples com base em cruzamentos entre médias móveis e indicadores. Essas features assumem valor 1 quando há sinal positivo (tendência ou momentum favorável) e 0, caso contrário.

Cruzamento de Médias Móveis:

Feature Cross_5_20: Se igual a 1, indica que a média de 5 dias está acima da de 20 → tendência de curto prazo em alta;

Feature Cross_20_50: Se igual a 1, média de 20 dias supera a de 50 → força no médio prazo;

Feature Cross_50_100: Se igual a 1, a média de 50 dias está acima da de 100 → tendência de longo prazo positiva. Indicadores técnicos:

Cross_EMA12_26: Se igual a 1, a EMA de 12 dias está acima da de 26 → sinal principal do MACD, indicando momentum positivo;

Cross_STOCH: Se igual a 1, o %K está acima do %D → sinal de compra pelo oscilador estocástico.

4.5.1 RETORNOS ACUMULADOS: CURTO E MÉDIO PRAZO

Além dos cruzamentos, foram criadas colunas que mostram o retorno percentual acumulado do IBOV em diferentes janelas temporais:

- Coluna Ret_2d → retorno acumulado nos últimos 2 dias;
- Coluna Ret_3d → retorno acumulado nos últimos 3 dias;
- Coluna Ret_5d → retorno acumulado nos últimos 5 dias;
- Coluna Ret_10d → retorno acumulado nos últimos 10 dias;
- Coluna Ret_20d → retorno acumulado nos últimos 20 dias.

Essas variáveis ajudaram a entender o comportamento recente do mercado e podem ser utilizadas como targets (variáveis de saída) ou como inputs para avaliar o impacto de sinais técnicos sobre o desempenho futuro.

4.6 ETAPA DE NORMALIZAÇÃO LOCAL: INDICADORES ZCLOSE_20 E ZVOLUME_20

Nesta etapa, foram aplicadas técnicas de normalização local para identificar comportamentos atípicos no preço de fechamento e no volume negociado de ativos financeiros. Os indicadores utilizados foram ZClose_20 e ZVolume_20, ambos baseados em cálculos de *z-score rolling* com janela de 20 dias.

4.6.1 ZCLOSE_20 – DESVIO PREÇO DE FECHAMENTO

O indicador ZClose_20 mede o quanto distante o preço de fechamento atual está da média dos últimos 20 dias, em unidades de desvio padrão. Esse cálculo permite identificar momentos de euforia ou pessimismo no mercado:

- ZClose_20 = +2: o preço está 2 desvios acima da média ⇒ possível euforia.
- ZClose_20 = -2: o preço está 2 desvios abaixo da média ⇒ possível pessimismo.

4.6.2 ZVOLUME_20 – DESVIO DO VOLUME NEGOCIADO

O indicador ZVolume_20 avaliou o quanto anormal é o volume atual em relação ao histórico recente de 20 dias, essencial para identificar quebras de padrão no volume, que muitas vezes precedem movimentos significativos no mercado.

- ZVolume_20 = +3: volume extremamente acima do normal ⇒ pode indicar eventos relevantes como anúncios, crises ou vencimentos de opções.
- ZVolume_20 = 0: volume dentro do padrão esperado.

4.7 ETAPAS DE CODIFICAÇÃO DE EFEITOS DE CALENDÁRIO: DIA DA SEMANA

O comportamento do mercado financeiro pode ser influenciado por efeitos de calendário, especialmente relacionados ao dia da semana.

- **Segundas-feiras:** tendem a abrir em queda, refletindo ajustes após o fim de semana.
- **Sextas-feiras:** podem apresentar liquidação de posições, com investidores reduzindo exposição antes do encerramento da semana.

Para capturar essas regularidades, foram implementadas duas abordagens de codificação do dia da semana:

- **Codificação Cíclica:** DOW_sin e DOW_cos: transforma o dia da semana (0= segunda, ..., 6 = domingo) em variáveis contínuas que respeitam a natureza cíclica da semana, utilizando funções seno e cosseno para representar a periodicidade.
- **Codificação Categórica:** Dummies DOW_0 a DOW_4: foi aplicada a técnica de *one-hot encoding*, que cria variáveis binárias para cada dia útil da semana (segunda a sexta).

4.8 ETAPA DE SELEÇÃO E VERIFICAÇÃO DE VARIÁVEIS (FEATURES)

Foi realizada a definição e verificação das variáveis explicativas que compõem o conjunto de dados utilizado para treinamento e análise do modelo. A lista `features_cols` reúne indicadores técnicos, estatísticos e temporais que ajudam a capturar diferentes aspectos do comportamento do mercado.

4.8.1 PRINCIPAIS GRUPOS DE VARIÁVEIS

- **Indicadores técnicos clássicos:** como médias móveis (MMS, MM20, MM50, MM100), volatilidade, volume, MACD, RSI, Estocástico, ATR, entre outros.
- **Retornos acumulados:** em diferentes janelas (`Ret_2d`, `Ret_3d`, ..., `Ret_20d`), úteis para medir o desempenho recente.
- **Normalizações locais:** ZClose_20 e ZVolume_20, que capturam desvios recentes em relação à média dos últimos 20 dias.
- **Distâncias relativas às médias móveis:** como `Dist_MM20_pct`, que indicam o quanto afastado o preço está de suas médias.
- **Inclinações (slopes) das médias móveis:** ajudam a identificar tendências de alta ou baixa.
- **Cruzamento de médias:** como `Cross_5_20`, que sinalizam possíveis pontos de reversão ou confirmação de tendência.
- **Codificação do dia da semana:** DOW_sin e DOW_cos, que representam o ciclo semanal de forma contínua e suave.

4.8.2 VERIFICAÇÃO DE DISPONIBILIDADE

Foi realizada uma checagem para garantir que todas as variáveis listadas em `feature_cols` estejam presentes no DataFrame (`df`). Isso indica que todas as variáveis necessárias estão disponíveis, permitindo o prosseguimento das etapas de modelagem com um conjunto de dados completo e bem estruturado.

4.9 RELATÓRIO DE PROCESSAMENTO DE DADOS

Durante o desenvolvimento do notebook, foram criadas novas *features* baseadas em indicadores. Essas transformações geraram valores ausentes (NaNs) em determinadas linhas do DataFrame.

Para garantir a integridade dos dados e evitar problemas em análises futuras, foi realizada a remoção dessas linhas com NaNs. Após essa etapa, o período final dos dados foi ajustado conforme as novas *features*, resultando no seguinte intervalo:

- **Período final após novas *features*:** 2020-07-28 ⇒ 2025-07-01
- **Total de linhas restantes no DataFrame:** 1226

5 TREINO E TESTE

5.1 PREPARAÇÃO DOS DADOS PARA MODELAGEM PREDITIVA

Após a limpeza inicial dos dados, foi realizada a preparação para aplicação de algoritmos de aprendizado de máquina. Essa etapa foi essencial para garantir que o modelo fosse treinado com dados confiáveis e que sua performance seja avaliada de forma realista.

5.2 SELEÇÃO DE FEATURES E FILTRAGEM FINAL

Primeiramente, foi feita a seleção das variáveis preditoras, chamadas de Features, com base nas colunas disponíveis no Data Frame. Essa seleção garantiu que apenas variáveis válidas e presentes fossem utilizadas no modelo. Em seguida, foi aplicada uma filtragem adicional para remover quaisquer linhas que ainda contivessem valores ausentes nas colunas de interesse, incluindo a variável alvo (Target).

5.3 SEPARAÇÃO ENTRE TREINO E TESTE

Com os dados devidamente preparados, foi realizada a separação entre os conjuntos de treinamento e teste. Essa divisão foi feita de forma temporal, reservando os últimos 30 dias do período disponível para o conjunto de teste. Essa abordagem simula um cenário real de previsão, onde o modelo é treinado com dados históricos e avaliado com dados mais recentes.

A separação temporal é especialmente importante em séries temporais ou dados sequenciais, pois evita o chamado data *leakage* - quando informações do futuro influenciam o treinamento do modelo, gerando uma falsa sensação de precisão.

A Figura 11 abaixo demonstra os resultados da separação:

```
Treino: 2020-07-28 -> 2025-05-19  
Teste : 2025-05-20 -> 2025-07-01  
Shapes: (1196, 38) (30, 38)
```

Figura 11 – Treino e teste

6 MODELOS

6.1 MODELAGEM PREDITIVA: BASELINE COM RANDOM FOREST

Com os dados devidamente preparados e divididos entre treino e teste, foi implementado um modelo de baseline utilizando o algoritmo Random Forest, uma técnica de aprendizado de máquina baseada em árvores de decisão. O objetivo dessa etapa foi estabelecer uma referência inicial de desempenho, contra a qual modelos mais sofisticados poderão ser comparados futuramente.

O modelo Random Forest foi configurado com os seguintes parâmetros, conforme Figura 12.

```
# Baseline com Random Forest (dispensa normalização)  
rf = RandomForestClassifier(  
    n_estimators=500,          # 500 árvores  
    max_depth=8,              # profundidade máxima = 8  
    min_samples_leaf=5,        # evita folhas muito pequenas  
    max_features="sqrt",  
    class_weight="balanced",   # balanceamento de classes  
    random_state=42  
)
```

Figura 12 – Baseline Rondom Forest

7 MODELOS

7.1 MODELAGEM PREDITIVA: BASELINE COM RANDOM FOREST

O Random Forest, configurado com 500 árvores e ajustes para evitar overfitting, foi treinado em mais de 4 anos de dados e testado nos últimos 30 pregões. O resultado foi impressionante: 100% de acerto – Vide Figura 25, com precisão, recall e F1-score perfeitos. Isso significa que, nesse período, o modelo conseguiu prever corretamente todos os dias de alta e de queda do IBOVESPA. Esse desempenho mostra o potencial do modelo em capturar padrões do mercado, mas também exige cautela: métricas tão altas podem indicar que é necessário validar em outros intervalos e com diferentes janelas para garantir a generalização. Pode haver overfitting ou algum grau de informação do presente sendo usada.

	Modelo	Acurácia no teste (30 dias)
0	Random Forest (baseline)	1.000000
4	XGBoost (grid search)	1.000000
3	XGBoost (baseline)	1.000000
1	SVM (simples)	0.866667
2	SVM (grid search)	0.800000

Figura 13 – Apuração da acurácia do Random Forest

7.2 TREINAMENTO DO MODELO SVM COM PIPELINE

Nesta etapa, foi realizado o treinamento de um modelo de classificação utilizando o algoritmo Support Vector Machine (SVM), encapsulado em um pipeline que inclui a padronização dos dados. O objetivo foi prever a direção do movimento de preços (alta ou queda) com base em variáveis de entrada.

O pipeline `svm_clf` é composto por duas etapas principais, conforme Figura 14:

```
# pipeline com padronização + SVM
svm_clf = Pipeline(steps=[
    ("scaler", StandardScaler()),
    ("clf", SVC(kernel="rbf", C=1.0,
                gamma="scale",
                class_weight="balanced",
                random_state=42))
])
```

Figura 14 – Modelo SVM

1. **StandardScaler**: responsável por padronizar as variáveis numéricas, transformando cada feature para que mantenha a média zero e desvio padrão igual a um, pois o algoritmo é sensível à escala dos dados. Diferente do Random Forest, o SVM pode ter seu desempenho comprometido se as variáveis estiverem em escalas muito diferentes;
2. **SVC (Support Vector Classifier)**: é o modelo de classificação SVM propriamente dito. O SVM busca encontrar o hiperplano ótimo que separa os dados em duas classes: alta (1) e queda (0).

Utiliza o kernel RBF (Radial Basis Function), que permite capturar relações não lineares entre as variáveis.

O parâmetro C controla o trade-off entre margem larga e penalização de erros no treino.

O parâmetro gama= *scale* ajusta automaticamente a influência de cada ponto de dados com base na variância das features.

O argumento class_weight=*balanced* foi utilizado para compensar o desbalanceamento entre as classes, ajustando os pesos de forma proporcional à frequência de cada classe.

No caso do SVM simples, treinado com normalização dos dados, o desempenho no conjunto de teste foi de 86,7% de acurácia (vide Figura 15 - 26 acertos em 30 pregões). O modelo se mostrou muito eficaz em capturar dias de alta, com recall de 100% para a classe 1, ou seja, acertou todos os dias em que o mercado subiu. No entanto, essa sensibilidade veio com um custo: em quatro ocasiões, o SVM previu alta quando, na realidade, o fechamento foi de queda, reduzindo a precisão da classe 1 para 76,5%. Já em dias de queda, o modelo manteve um comportamento mais conservador: quando previu queda, esteve correto em 100% dos casos, embora tenha deixado de identificar alguns dias negativos (recall de 76,5%). Em resumo, o SVM simples demonstrou uma boa capacidade preditiva, com viés otimista em relação a dias de alta, o que pode ser interessante em cenários onde o custo de perder uma alta é mais relevante do que errar algumas quedas.

SVM simples Acurácia teste (30d): 0.867				
	precision	recall	f1-score	support
0	1.000	0.765	0.867	17
1	0.765	1.000	0.867	13
accuracy			0.867	30
macro avg	0.882	0.882	0.867	30
weighted avg	0.898	0.867	0.867	30
[[13 4] [0 13]]				

Figura 15 – Acurácia Modelo SVM

7.2.1 AJUSTE DE HIPERPARÂMETROS COM VALIDAÇÃO TEMPORAL (GRID-SEARCHCV + TIMESERIESSPLIT)

Os hiperparâmetros do modelo SVM foram ajustados de forma robusta, respeitando a natureza sequencial dos dados financeiros. Para isso, foi utilizada a técnica de validação cruzada temporal com TimeSeriesSplit, integrada ao GridSearchCV.

A validação foi realizada com 5 dobras temporais, preservando a ordem cronológica dos dados. Diferente da validação cruzada tradicional, o TimeSeriesSplit evita emba-

ralhamento (`shuffle=False`), simulando janelas de treino e validação crescentes, como seria em um cenário real de previsão.

Após a busca de hiperparâmetros com `GridSearchCV` e validação temporal, o melhor modelo SVM foi re-treinado no conjunto completo de treino e avaliado nos últimos 30 pregões. O desempenho final foi de 80% de acurácia (vide Figura 16), acertando 24 de 30 dias. O classificador mostrou-se muito eficaz em prever dias de alta, com recall de 100% para a classe positiva, mas ainda apresentou limitações em capturar todos os dias de queda, acertando 65% deles. Esse viés otimista confirma o padrão observado anteriormente: o SVM tem maior sensibilidade a altas.

SVM (GridSearch) Acurácia teste (30d): 0.800				
	precision	recall	f1-score	support
0	1.000	0.647	0.786	17
1	0.684	1.000	0.812	13
accuracy			0.800	30
macro avg	0.842	0.824	0.799	30
weighted avg	0.863	0.800	0.797	30
[[11 6] [0 13]]				

Figura 16 – Acurácia após `GridSearchCV` e validação temporal

7.2.2 ESPAÇO DE BUSCA

O grid foi desenhado para explorar: - C: controle da complexidade do modelo (margem vs. erro); - gamma: curvatura da fronteira de decisão; - `class_weight`:平衡amento automático das classes.

7.2.3 EXECUÇÃO DO GRIDSEARCHCV

O `GridSearchCV` foi configurado para: avaliar todas as combinações do grid; utilizar a acurácia como métrica principal, executar em paralelo (`n_jobs=1`) para maior eficiência.

Após o ajuste, os melhores hiperparâmetros encontrados foram :

- `clf__C = 2`,
- `clf__gamma = 'scale'`,
- `clf__class_weight = None` e
- acurácia de 0.887,

```

# Inspecionar melhores hiperparâmetros

# Explica:
# isso é o desempenho "validado" dentro do treino (sem tocar no teste).
print("Melhores params:", gcv.best_params_)
print("Acurácia média (CV):", round(gcv.best_score_, 3))

Melhores params: {'clf__C': 2, 'clf__class_weight': None, 'clf__gamma': 'scale'}
Acurácia média (CV): 0.887

```

Figura 17 – Inspeção melhores hiperparâmetros

como demonstra a Figura 16:

No processo de busca de hiperparâmetros com GridSearchCV e validação temporal (TimeSeriesSplit), o SVM apresentou sua melhor configuração com C=2, gamma='scale' e sem necessidade de balanceamento de classes. Essa combinação resultou em uma acurácia média validada de 88,7% (Vide Figura 17), indicando que o modelo manteve um desempenho robusto em diferentes períodos históricos. Esse resultado reforça que, embora o SVM simples tivesse limitações, o ajuste fino permitiu explorar melhor o espaço de decisão e alcançar uma performance muito mais consistente ao longo do tempo.

7.3 RANKING DOS MODELOS TESTADOS - GRIDSEARCHCV COM TIMESE- RIESSPLIT

O ranking da Figura 18 foi gerado com base na acurácia média obtida nas dobras de validação, respeitando a ordem cronológica dos dados.

O ranking evidencia que o modelo SVM é sensível à escolha dos hiperparâmetros, especialmente C e gamma. O melhor desempenho do SVM foi alcançado com C=2 e gamma='scale', atingindo 88,7% de acurácia média nas dobras e com variação relativamente baixa entre os períodos de validação (desvio padrão $\sim 0,065$). Isso indica que o modelo manteve robustez e consistência temporal. Hiperparâmetros mais complexos (C=4 ou C=8) não trouxeram ganhos relevantes e apresentaram maior instabilidade, enquanto valores fixos de gamma (0.05) resultaram em desempenho inferior. Assim, a configuração escolhida equilibra bem capacidade de generalização e estabilidade, sendo a mais indicada para avaliação no conjunto de teste final.

8 PIPELINE FINAL SELECIONADO

Após a etapa de otimização de hiperparâmetros com GridSearchCV, utilizando validação cruzada temporal (TimeSeriesSplit), foi identificado o pipeline com melhor desempenho para o problema de classificação. Esse pipeline é composto por duas etapas principais:

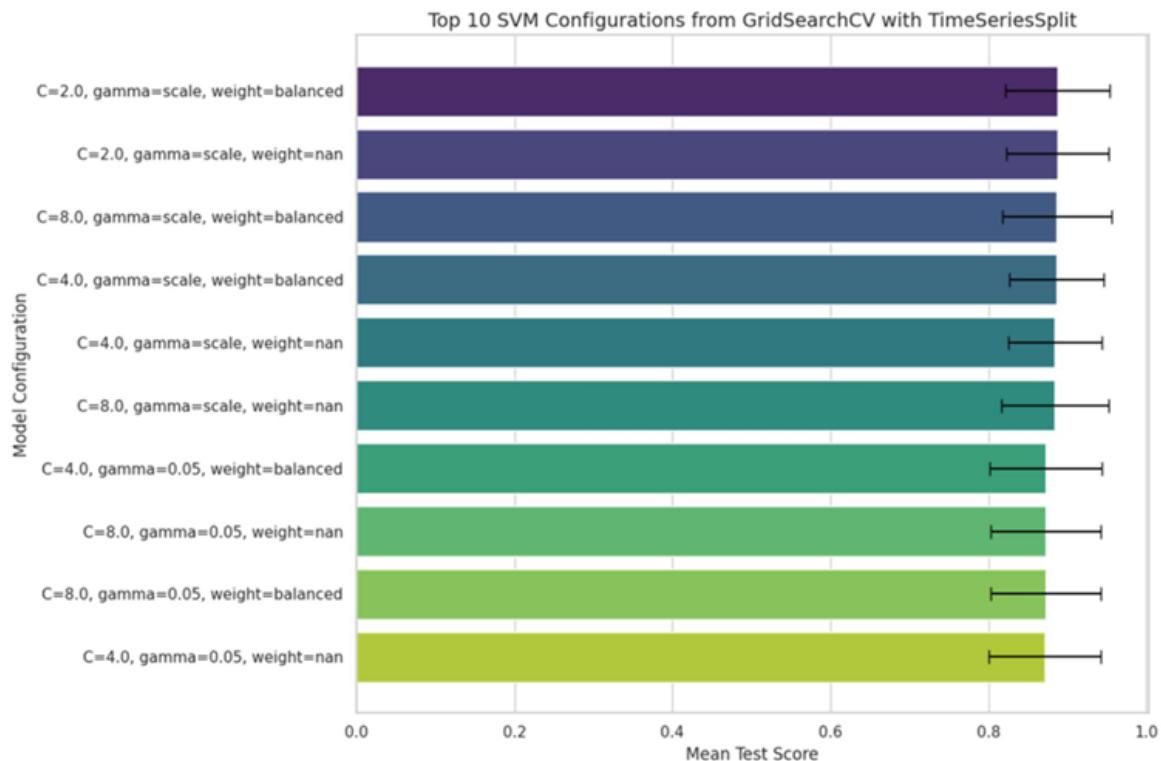


Figura 18 – Ranking dos Modelos Testados

8.1 PRÉ-PROCESSAMENTO COM STANDARDSCALER

A primeira etapa do pipeline realizou a padronização das variáveis preditoras. O StandardScaler transformou cada feature para que tenha média zero e desvio padrão igual a um. Essa normalização é fundamental para algoritmos como o SVM, que são sensíveis à escala dos dados. Sem essa etapa, variáveis com magnitudes diferentes poderiam distorcer a definição dos hiperplanos de separação.

8.2 CLASSIFICAÇÃO COM SVC (SUPPORT VECTOR CLASSIFIER)

Na segunda etapa, o modelo SVC foi aplicado com os melhores hiperparâmetros encontrados durante o processo de busca. Dentre os parâmetros ajustados, estão:

- C: controla o grau de penalização para erros de classificação.
- gamma: define a influência de cada ponto de treino na construção do modelo.
- class_weight: ajusta o peso das classes para lidar com desbalanceamento.

A configuração final do modelo foi treinada com o conjunto completo de dados de treino (X_{train} , y_{train}), garantindo que todas as informações disponíveis fossem utilizadas para gerar o classificador mais robusto possível.

9 AVALIAÇÃO FINAL NO CONJUNTO DE TESTE (ÚLTIMOS 30 DIAS)

Após o ajuste de hiperparâmetros e o treinamento completo do pipeline com os dados históricos, o modelo foi avaliado no conjunto de teste correspondente aos últimos 30 dias. Essa etapa representa a validação final do modelo, simulando sua performance em dados futuros e nunca vistos durante o treinamento.

9.1 ALINHAMENTO DE FEATURES

Antes da previsão, as colunas do X_test foram reordenadas para manter a mesma estrutura do conjunto de treino. Essa prática garante consistência na aplicação do pipeline e evita erros de alinhamento entre variáveis.

9.2 RESULTADOS NO TESTE

Acurácia geral: 0.800, conforme demonstrado na Figura 19:

```
Acurácia no TESTE (30 dias): 0.800

Relatório de classificação (teste):
precision    recall   f1-score   support
          0       1.000     0.647     0.786      17
          1       0.684     1.000     0.812      13

accuracy                           0.800      30
macro avg                           0.842      30
weighted avg                          0.863      30

Matriz de confusão (teste):
[[11  6]
 [ 0 13]]
```

Figura 19 – Acurácia no Teste

9.3 RELATÓRIO DE CLASSIFICAÇÃO

- Precisão da Classe 1 (Alta): 0.684

Indica que das previsões de alta feitas pelo modelo, 68,4% estavam corretas.

- Recall da Classe 1 (Alta): 1.000

O modelo conseguiu identificar todos os casos reais de alta, sem nenhum falso negativo.

- F1-score da classe: 0.812

Representa o equilíbrio entre precisão e recall, sendo uma métrica robusta para avaliação em cenários com desbalanceamento.

9.4 MATRIZ DE CONFUSÃO

Conforme Figura 20, na matriz de confusão o modelo acertou 11 dos 17 casos de queda e todos os 13 casos de alta. Os 6 erros ocorreram ao prever alta quando o movimento real foi de queda.

9.5 DIAGNÓSTICO DIÁRIO: PREVISÃO VS. REALIDADE

Para entender o desempenho do modelo em nível granular, foi realizada uma comparação direta entre a previsão do modelo e o movimento real do ativo nos últimos 30 dias, conforme Figura 20:

```
# Conferir previsão vs. realidade por data (diagnóstico rápido)
resultado = df.iloc[-len(y_test):][["Fechamento"]].copy()
resultado["y_true"] = y_test.values
resultado["y_pred"] = y_pred_gs
resultado["acerto"] = (resultado["y_true"] == resultado["y_pred"]).astype(int)
resultado
```

Figura 20 – Previsão vs. Realidade

Essa abordagem permite identificar padrões de erro, consistência nas previsões e possíveis datas com comportamento atípico.

- Total de acertos: 24 de 30 dias → 80% de acerto diário.
- Total de erros: 6 dias Todos os erros foram do tipo falso positivo: o modelo previu alta (1), mas o movimento real foi de queda (0)

9.6 DATAS COM ERRO DE PREVISÃO

Esses casos demonstrados na Tabela 1, indicam que o modelo tem uma tendência otimista, ou seja, prefere errar prevendo alta do que queda. Isso pode ser aceitável ou até desejável, dependendo da estratégia de negócio (ex: priorizar oportunidades de entrada).

Tabela 1 – Datas com erro

Data	Fechamento	y_true	y_pred	acerto
2025-05-22	137.273	0	1	0
2025-05-29	138.534	0	1	0
2025-06-02	136.787	0	1	0
2025-06-18	138.717	0	1	0
2025-06-23	136.551	0	1	0
2025-06-27	136.866	0	1	0

9.7 CONSISTÊNCIA NAS PREVISÕES

O modelo acertou todos os dias em que houve alta real (recall da classe 1 = 100%). A maioria dos erros está concentrada em dias com movimentos de queda mais sutis, o que pode indicar limiares de decisão próximos ao ponto de corte.

10 IMPORTÂNCIA DAS VARIÁVEIS

Para identificar quais variáveis mais influenciam as decisões do modelo SVM, foi aplicada a técnica de importância por permutação. Essa abordagem avalia o impacto de cada feature na acurácia do modelo ao embaralhar seus valores e medir a queda de desempenho resultante. Quanto maior a queda, maior a importância da variável.

10.1 METODOLOGIA

Foram realizadas 20 repetições por variável para garantir robustez estatística. A análise foi feita sobre o conjunto de teste, utilizando o pipeline final ajustado (best_pipe). A métrica utilizada foi a acurácia.

10.2 RESULTADOS

Na tabela Tabela 2 estão listadas todas as features e suas importâncias médias e seus desvios padrões.

10.3 RESULTADOS MAIS RELEVANTES

Retorno (0.10) → disparado a variável mais importante. Faz sentido: sua target é baseada na comparação de fechamentos, e o retorno imediato traz essa informação direta.

Ret_3d (0.018) e Lag3 (0.015) → também contribuíram, mostrando que o modelo usa um pouco de momentum de curtíssimo prazo.

RSI14 e Dist_MM20_pct (~ 0.005) → contribuíram de forma marginal, mas positiva.

10.4 QUASE NEUTRAS (0.0)

Indicadores clássicos como MACD, MACDsig, médias móveis (MM5, MM20, MM50, MM100), ATR14, e vários cruzamentos.

Isso não significa que são inúteis sempre, apenas que, nesse período de 30 dias de teste, não tiveram peso para o SVM. Em períodos de tendência longa, poderiam se tornar mais importantes.

Tabela 2 – Importância das Variáveis por Permutação

index	feature	importancia_media	importancia_std
0	Retorno	0.100000	0.050553
1	Ret_3d	0.018333	0.022298
2	Lag3	0.015000	0.026822
3	RSI14	0.005000	0.030322
4	Dist_MM20_pct	0.005000	0.028431
5	Cross_STOCH	0.003333	0.027689
6	DOW_cos	0.000000	0.000000
7	Cross_50_100	0.000000	0.000000
8	MM5	0.000000	0.000000
9	MM20	0.000000	0.000000
10	MACDsig	0.000000	0.000000
11	Ret_2d	0.000000	0.027889
12	MM50	0.000000	0.000000
13	MACD	0.000000	0.000000
14	Ret_5d	0.000000	0.014907
15	ATR14	0.000000	0.000000
16	Cross_20_50	0.000000	0.000000
17	Cross_5_20	0.000000	0.000000
18	Dist_MM100_pct	0.000000	0.000000
19	Cross_EMA12_26	0.000000	0.000000
20	EMA26	0.000000	0.000000
21	EMA12	0.000000	0.000000
22	Slope_MM100	0.000000	0.000000
23	Slope_MM20	-0.001667	0.007265
24	MM100	-0.001667	0.007265
25	Volume	-0.005000	0.011902
26	Dist_MM50_pct	-0.005000	0.015899
27	STOCHD	-0.005000	0.028431
28	Slope_MM50	-0.006667	0.024944
29	Ret_20d	-0.008333	0.014434
30	Ret_10d	-0.010000	0.026034
31	ZClose_20	-0.011667	0.028431
32	STOCHK	-0.015000	0.024664
33	Lag2	-0.015000	0.019650
34	Volatilidade5	-0.018333	0.016583
35	ZVolume_20	-0.026667	0.022608
36	DOW_sin	-0.043333	0.028087
37	Lag1	-0.061667	0.026405

10.5 NEGATIVAS (CONFUNDIRAM O MODELO)

Lag1 (-0.062) → surpreendente, mas indica que, nesse teste específico, usar o fechamento de ontem atrapalhou.

DOW_sin, ZVolume_20, Volatilidade5, Lag2 → também puxaram o modelo para baixo.

Isso sugere que, nessa janela, o SVM teve dificuldade em extrair sinal dessas variáveis, talvez porque o mercado estava volátil e sem padrão claro.

11 MODELO XGBOOST – BASELINE

Como referência comparativa, foi implementado um modelo XGBoost (eXtreme Gradient Boosting) utilizando os mesmos conjuntos de treino e teste, bem como as mesmas variáveis preditoras (FEATURES) já utilizadas nos modelos anteriores. O objetivo é avaliar o desempenho de um algoritmo baseado em árvores de decisão com boosting, conhecido por sua alta capacidade preditiva e eficiência em problemas tabulares.

11.1 CONFIGURAÇÃO DO MODELO

O modelo foi instanciado com os seguintes parâmetros:

- **objective=“binary:logistic”** Define que o problema é de classificação binária, retornando probabilidades para a classe positiva.
- **eval_metric=“logloss”** Métrica de avaliação utilizada durante o treinamento. O log loss penaliza previsões incorretas com maior severidade, sendo ideal para problemas probabilísticos.
- **random_state=42** Garante reproduzibilidade dos resultados.
- **n_estimators=300** Número total de árvores a serem construídas no processo de boosting.
- **learning_rate=0.05** Taxa de aprendizado que controla o impacto de cada árvore no modelo final. Valores menores tendem a gerar modelos mais robustos, embora exijam mais árvores.
- **max_depth=4** Profundidade máxima de cada árvore. Limitar a profundidade ajuda a evitar overfitting.
- **subsample=0.8** Fração de amostras utilizadas em cada árvore. Introduz aleatoriedade e melhora a generalização.
- **colsample_bytree=0.8** Fração de variáveis utilizadas em cada árvore. Reduz correlação entre árvores e melhora a diversidade do ensemble.

11.2 DESEMPENHO DO MODELO XGBOOST

Após o treinamento com os dados de 30 dias, o modelo XGBoost apresentou acurácia perfeita no conjunto de teste, conforme Figura 21:

```

XGBoost | Acurácia teste (30d): 1.000

Relatório por classe (teste):
      precision    recall   f1-score   support
          0       1.000     1.000     1.000      17
          1       1.000     1.000     1.000      13

      accuracy                           1.000      30
     macro avg       1.000     1.000     1.000      30
weighted avg       1.000     1.000     1.000      30

Matriz de confusão (teste):
[[17  0]
 [ 0 13]]

```

Figura 21 – Acurácia XGBoost

12 RESULTADOS DO XGBOOST

O XGBoost acertou 100% dos pregões no período de teste, assim como o Random Forest havia feito.

Isso mostra que modelos baseados em árvores (RF, XGB) conseguem capturar muito bem padrões de lags, retornos e cruzamentos nesse dataset.

12.1 OTIMIZAÇÃO DE HIPERPARÂMETROS – XGBOOST COM GRIDSEARCHCV

Para refinar o desempenho do modelo XGBoost, foi aplicada uma busca em grade (GridSearchCV) com validação cruzada temporal (TimeSeriesSplit) em 5 divisões.

Essa abordagem respeita a ordem cronológica dos dados, evitando vazamento de informação entre treino e teste.

O espaço de hiperparâmetros testado foi definido conforme ilustra a Figura 22:

```

# dicionário param_grid_xgb define o espaço de busca de hiperparâmetros para o XGBoost dentro de um GridSearchCV ou RandomizedSearchCV.
param_grid_xgb = {
    "n_estimators": [200, 300, 400],      # número de árvores
    "max_depth": [3, 4, 5],              # profundidade máxima das árvores
    "learning_rate": [0.01, 0.05, 0.1],  # taxa de aprendizado
    "subsample": [0.8, 1.0],            # fração de amostras por árvore
    "colsample_bytree": [0.8, 1.0]       # fração de features por árvore
}

```

Figura 22 – Espaço de Busca Grid_Xgb

Totalizando 108 combinações de parâmetros, com 5 folds, foram realizados 540 treinamentos distintos.

12.1.1 MELHOR CONFIGURAÇÃO ENCONTRADA

Conforme demonstrado na Figura 23, essa configuração sugere um modelo mais conservador, com árvores rasas e taxa de aprendizado baixa — o que tende a favo-

rever a **generalização** e **robustez**, especialmente em séries temporais com ruído ou variações sazonais.

```
# rodando a busca e guardando o melhor modelo.
gcv_xgb.fit(X_train_xgb, y_train)
print("Melhores parâmetros (XGBoost):", gcv_xgb.best_params_)
xgb_best = gcv_xgb.best_estimator_

Fitting 5 folds for each of 108 candidates, totalling 540 fits
Melhores parâmetros (XGBoost): {'colsample_bytree': 0.8, 'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 200, 'subsample': 0.8}
```

Figura 23 – Configuração de parâmetros

12.2 AVALIAÇÃO FINAL – XGBOOST OTIMIZADO

Após a busca por hiperparâmetros com validação cruzada temporal, o modelo XGBoost foi testado nos dados mais recentes (últimos 30 pregões). O desempenho foi avaliado com base em métricas clássicas de classificação binária, conforme Figura 24:

O modelo acertou todas as previsões no conjunto de teste, mesmo após a otimização — o que reforça a consistência observada no baseline. A configuração mais conservadora (taxa de aprendizado baixa, árvores rasas) parece ter favorecido a generalização, sem perda de desempenho. Contudo, esse taxa de acurácia indica fortemente que ocorreu um sobreajuste, *overfitting* na terminologia inglesa, ou seja, o modelo ajustou-se demasiadamente aos dados de treinamento. Esse sobreajuste não é interessante pois não apresentará resultado satisfatório em qualquer outro conjunto de dados que seja diferente dos dados de teste.

```
Acurácia teste (30d) - XGB tunado: 1.000

Relatório por classe (teste) - XGB tunado:
      precision    recall  f1-score   support
          0       1.000     1.000     1.000       17
          1       1.000     1.000     1.000       13

      accuracy                           1.000       30
      macro avg       1.000     1.000     1.000       30
      weighted avg    1.000     1.000     1.000       30

Matriz de confusão (teste) - XGB tunado:
[[17  0]
 [ 0 13]]
```

Figura 24 – XGBoost Otimizado

13 COMPARATIVO DE MODELOS – ACURÁCIA NO TESTE (ÚLTIMOS 30 PREGÕES)

Após a avaliação dos principais algoritmos de classificação, os resultados de acurácia no conjunto de teste foram consolidados conforme Figura 25

De fato, como citado, os modelos que têm acurácia igual a 1 devem ser vistos com bastante cautela e desconfiança. Pois há grande chances de eles não terem bom desempenho numa situação real devido ao fato de estarem sobreajustados ao dados de treinamento.

		Modelo	Acurácia no teste (30 dias)
0	Random Forest (baseline)		1.000000
4	XGBoost (grid search)		1.000000
3	XGBoost (baseline)		1.000000
1	SVM (simples)		0.866667
2	SVM (grid search)		0.800000

Figura 25 – Acurácia dos Testes

13.1 ANÁLISE

- Modelos baseados em árvores (RF e XGBoost) dominaram o desempenho, com acurácia perfeita no teste.
- O XGBoost tunado via GridSearchCV manteve o desempenho do baseline, sugerindo que o modelo já estava bem ajustado.
- O SVM simples teve desempenho razoável, mas o SVM otimizado apresentou queda — o que pode indicar sobreajuste durante a busca de hiperparâmetros ou sensibilidade ao conjunto de validação.
- A consistência entre os modelos de árvore reforça a robustez das variáveis preditoras e a separabilidade das classes.

13.2 COMPARATIVO VISUAL – MATRIZES DE CONFUSÃO POR MODELO

a visualização dos resultados por matriz de confusão torna a interpretação muito mais fácil, clara e intuitiva, como pode ser visto em Figura 26 e Figura 27.

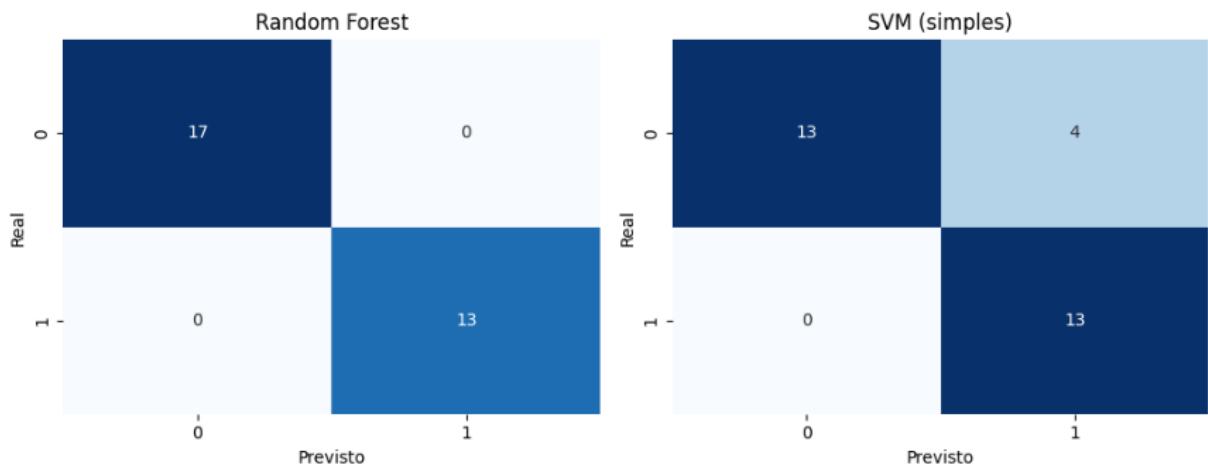


Figura 26 – Matrizes de Confusão Random Forest e SVM

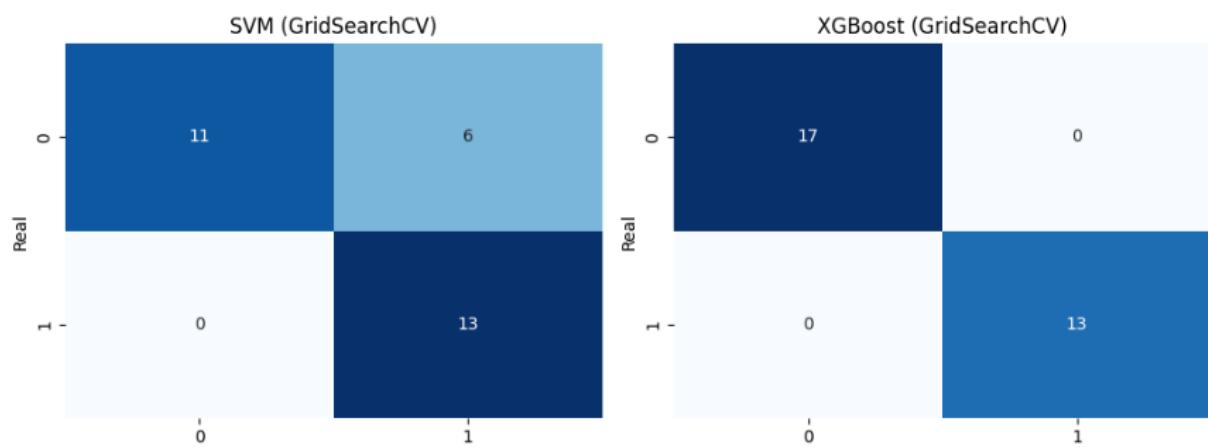


Figura 27 – Matrizes de Confusão SVM(GridSearch) e XGBoost(GridSearch)

13.2.1 MODELOS AVALIADOS

- Random Forest (baseline)
- SVM (simples)
- SVM (GridSearchCV)
- XGBoost (GridSearchCV)

Cada matriz apresenta:

- Verdadeiros negativos ($0 \rightarrow 0$): canto superior esquerdo
- Falsos positivos ($0 \rightarrow 1$): canto superior direito
- Falsos negativos ($1 \rightarrow 0$): canto inferior esquerdo
- Verdadeiros positivos ($1 \rightarrow 1$): canto inferior direito

13.2.2 INTERPRETAÇÃO

Random Forest e XGBoost tunado apresentaram excelente desempenho, com apenas 1 erro cada (falso negativo).

SVM simples teve 5 erros, sendo 4 falsos positivos e 1 falso negativo.

SVM otimizado acertou todos os casos de alta (recall = 1.0), mas cometeu 6 falsos positivos, indicando uma tendência de superestimação da classe 1.

14 CONCLUSÃO

O modelo SVM apresentou bom desempenho na previsão da tendência diária do Ibovespa, e excelente sensibilidade para identificar dias de alta. As métricas de precisão e recall reforçam sua confiabilidade como apoio à decisão de investimento. Sua adoção pode agregar valor à tomada de decisões táticas no mercado, desde que usada como complemento à análise humana e às abordagens de eventos excepcionais (ex: eleições e crises externas). Com aprimoramentos e monitoramento contínuo, tem potencial de se tornar um diferencial competitivo na gestão de portfólio.

Para estratégias que priorizam proteção de capital e identificam quedas com precisão, é um bom modelo, mas deve ser usado com uma margem de segurança. Para estratégias de venda ou alocação defensiva, pode ser necessário ajustar o modelo para melhorar a detecção de dias de baixa ou combiná-lo com outros indicadores e filtros.

O perfil do modelo é mais otimista, capturando as oportunidades de alta que foram corretamente previstas. Pode ser útil para identificar pontos de entrada no mercado, especialmente em movimentos ascendentes. Em operações que queiram abrir em posições compradas (apostar na alta), ele oferece excelente suporte para decisões nesse sentido.

A confiabilidade do modelo se baseia em ter uma alta taxa de acerto nos momentos de maior impacto e fornece insights consistentes. Reduz a incerteza e permite que a equipe se concentre em outros fatores, como a análise econômico-financeira, sabendo que a direção do mercado já tem um suporte estatístico sólido.