UNIVERSIDADE VILA VELHA

LUCAS CUNHA MISSAGIA

Analise Comparativa de Estrutura de Regressão Linear e Árvore de Regressão aplicada ao Dataset Bovespa

Estrutura de Dados II

Vila Velha - ES

2025

Introdução

O presente relatório tem como objetivo aplicar técnicas de aprendizado de máquina à análise de dados históricos das ações da Petrobras (PETR3 e PETR4), com foco na previsão do preço de fechamento no pregão seguinte. Para isso, foi utilizado um conjunto de dados da B3 (Bovespa) abrangendo o período de 2015 a 2016.
Após um processo completo de pré-processamento, limpeza e engenharia de atributos, foram treinados dois modelos supervisionados: Regressão Linear e Árvore de Decisão Regressora. Ambos os modelos utilizaram as mesmas variáveis técnicas como entrada, incluindo preço de abertura, máxima, mínima, volume e indicadores derivados como média móvel de 5 dias (MM5) e volatilidade (Vol5).
A proposta central é comparar o desempenho e o comportamento de cada modelo, tanto em termos de métricas quantitativas (como MAE e R^2) quanto por meio de análises gráficas e estruturais, identificando as forças e limitações de cada abordagem no contexto de séries temporais financeiras.

Descrição do Conjunto de Dados

O dataset utilizado, nomeado "Bovespa.csv", contém registros diários de negociação de diversas ações da B3 no período de 28 de setembro de 2015 a 28 de setembro de 2016. Cada registro apresenta:

- Date: data da negociação (formato brasileiro DD/MM/AAAA);
- Ticker: código da ação (e.g., PETR3, PETR4);
- Open, High, Low, Close: preços de abertura, máximo, mínimo e fechamento;
- Volume: volume de ações negociadas no dia;

O conjunto original possui cerca de 15 mil registros, dos quais apenas PETR3 e PETR4 foram selecionados para a análise.

Etapas de Processamento e Engenharia de Atributos

- **Filtragem de Dados:**
  Selecionaram-se apenas os registros onde o campo `Ticker` fosse igual a "PETR3" ou "PETR4":

```
df = df[df["Ticker"].isin(["PETR3", "PETR4"])].copy()
```

- **Conversão de Formatos Numéricos:**
  As colunas de preço estavam em formato string com vírgulas (ex: "10,45"). Para permitir análise quantitativa, realizou-se a substituição por ponto e conversão para float:

```
for col in ["Open", "High", "Low", "Close", "Volume"]:
    df[col] = df[col].astype(str).str.replace(",", ".").astype(float)
```

- **Ordenação Temporal:**
  A ordenação por data garante a consistência das janelas temporais usadas em variáveis derivadas:

```
df["Date"] = pd.to_datetime(df["Date"], dayfirst=True)
df = df.sort_values(by=["Ticker", "Date"])
```

- **Criação de Atributos Derivados:**
  Retorno percentual (**`Return`**): variação do fechamento em relação ao dia anterior:

```
df["Return"] = df.groupby("Ticker")["Close"].pct_change()
```

  Média móvel de 5 dias (`MM5`): suaviza oscilações e identifica tendências:

```
df["MM5"] = df.groupby("Ticker")["Close"].transform(lambda x: x.rolling(5).mean())
```

  Volatilidade de 5 dias (Vol5): medida de dispersão dos retornos:

```
df["Vol5"] = df.groupby("Ticker")["Return"].transform(lambda x: x.rolling(5).std())
```

- **Variável Alvo – Target:**
  O `Target` representa o fechamento do dia seguinte:

```
df["Target"] = df.groupby("Ticker")["Close"].shift(-1)
```

- **Limpeza de Dados:**
  Remoção de todas as linhas com valores nulos oriundos das operações de janela:

```
df.dropna(inplace=True)
```

Divisão Temporal em Conjuntos de Treino e Teste

Os dados foram divididos cronologicamente, mantendo 80% para treino e 20% para teste. Essa abordagem evita vazamento de informação futura no conjunto de treino.

```
split_idx = int(len(df) * 0.8)
X_train, X_test = X[:split_idx], X[split_idx:]
y_train, y_test = y[:split_idx], y[split_idx:]
```

Treinamento do Modelo de Regressão Linear

A Regressão Linear é um modelo estatístico supervisionado utilizado para prever valores numéricos com base em variáveis independentes. Seu funcionamento se baseia na construção de uma equação linear, que estima o valor de saída (variável dependente) a partir da combinação ponderada das variáveis de entrada.

Durante o treinamento, o modelo ajusta os coeficientes de cada variável para minimizar o erro entre os valores previstos e os valores reais, geralmente por meio do método dos mínimos quadrados.

Foi utilizado o modelo LinearRegression da biblioteca sklearn:

```
model = LinearRegression()
model.fit(X_train, y_train)
```

A predição foi feita sobre todos os dados:

```
df["Predicted"] = model.predict(X)
```

Treinamento do Modelo de Árvore de Decisão Regressora

A árvore realiza divisões sequenciais nos dados com base em regras do tipo “se... então...”, utilizando as variáveis de entrada para segmentar o conjunto de dados em grupos mais homogêneos. Cada nó interno apresenta a condição de corte (ex: Close ≤ 9.82), o número de amostras no nó (samples), o valor médio previsto (value) e o erro quadrático médio local (mse). Nos nós terminais (folhas), são atribuídos os valores previstos finais. Essa estrutura permite interpretar de forma clara como o modelo toma decisões com base nas diferentes faixas de valor das variáveis.

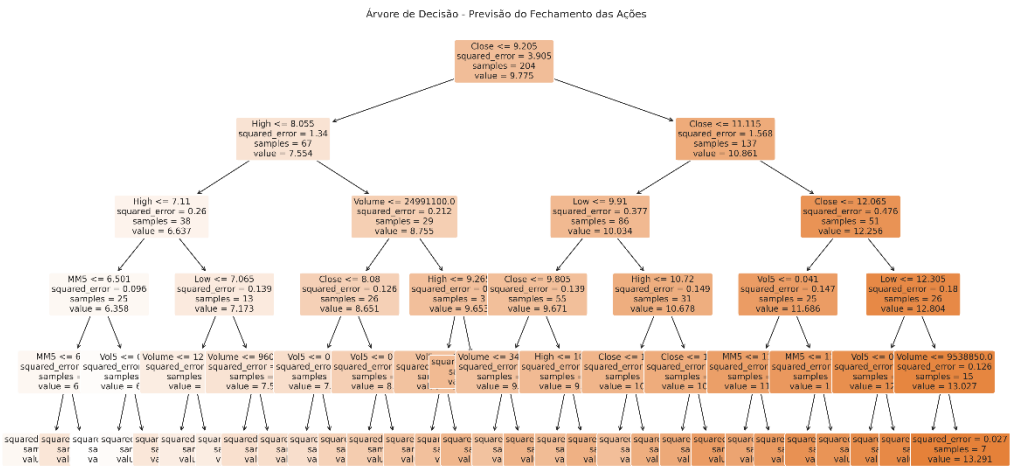
Foi utilizado o modelo DecisionTreeRegressor da biblioteca sklearn:

```
model = DecisionTreeRegressor(max_depth=5, random_state=42)
model.fit(X_train, y_train)
```

A predição foi feita sobre todos os dados:

```
df["Predicted"] = model.predict(X)
```

Estrutura da Arvore de Decisão:



Análise dos Gráficos: Predição x Variáveis de Entrada

Cada conjunto de gráficos apresentado compara uma variável de entrada com as previsões geradas por dois modelos distintos: a Regressão Linear e a Árvore de Decisão Regressora. O objetivo é compreender como cada variável contribui para a construção das predições em cada abordagem e identificar diferenças nos padrões de resposta entre os modelos.

Na regressão linear, espera-se uma relação contínua e suave entre as variáveis e a predição, refletida em gráficos com tendência linear. Já na árvore de decisão, a relação é estabelecida por divisões em faixas (degraus), produzindo gráficos com agrupamentos horizontais, característicos de modelos baseados em regras condicionais.

Embora os gráficos de dispersão sejam tradicionalmente utilizados para interpretar relações contínuas, como no caso da regressão linear, eles também são úteis na árvore de decisão para visualizar como as predições se agrupam em faixas discretas. No entanto, o comportamento segmentado observado nos gráficos reflete a estrutura de divisão hierárquica do modelo, e não uma relação contínua entre variável e saída.

A análise gráfica permite observar quais variáveis têm influência significativa na predição, como os modelos interpretam e utilizam essas variáveis de forma distinta e quais variáveis apresentam baixa ou nenhuma correlação com os valores previstos.

Essa abordagem comparativa visual é essencial para avaliar o desempenho numérico dos modelos e sua capacidade de interpretação, generalização e comportamento estrutural diante dos dados históricos.

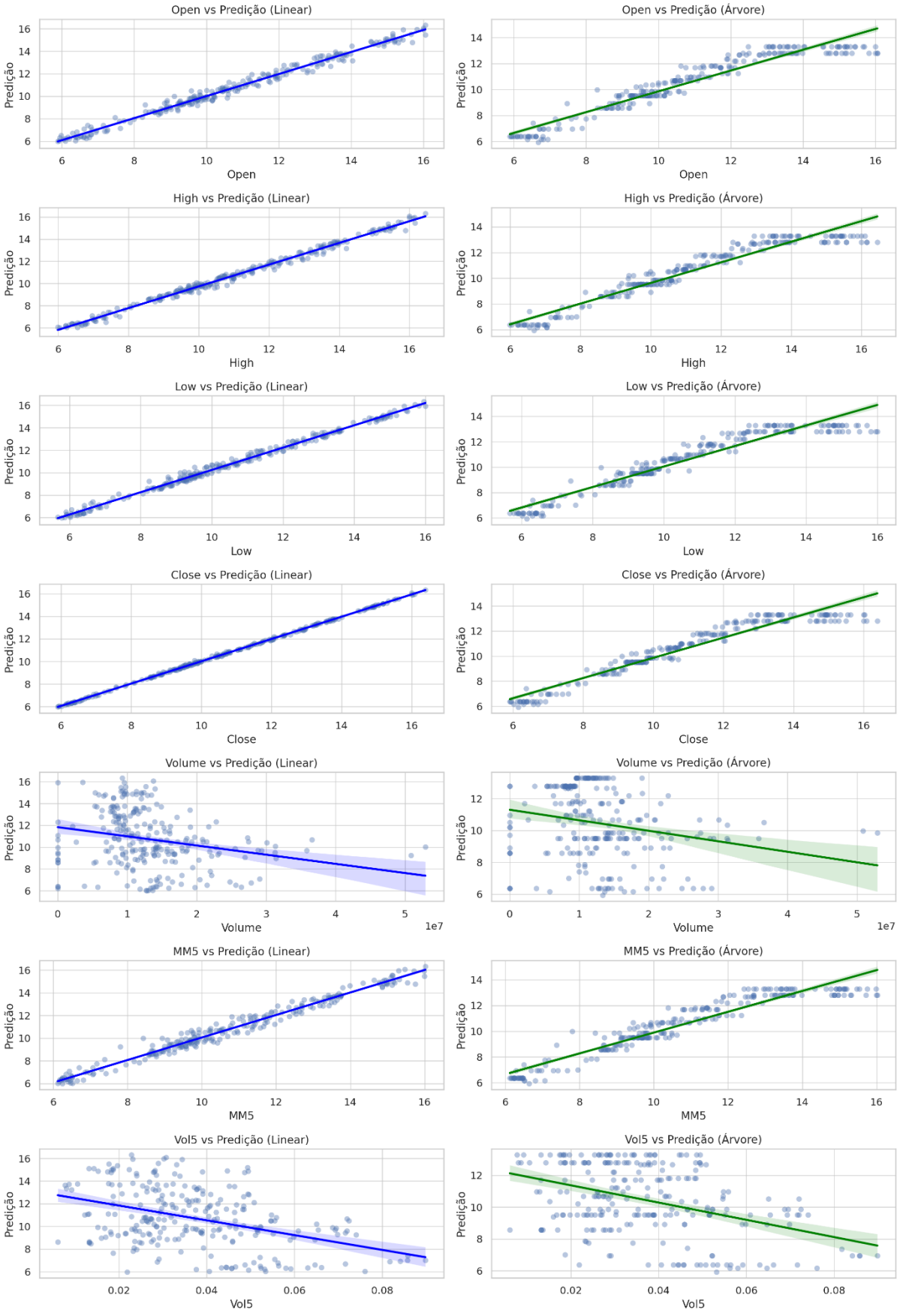
Open vs Predicted:

Este gráfico compara o preço de abertura do dia com o valor previsto para o fechamento do dia seguinte. Observa-se uma tendência linear moderada, indicando que o modelo considera o preço de abertura como uma referência inicial. No entanto, ele não é um fator determinante isoladamente. Essa variável contribui com o contexto do início do pregão, mas sua precisão depende da combinação com outras variáveis.

High vs Predicted:

Relaciona o maior preço do dia com a previsão do fechamento futuro. A dispersão dos pontos indica uma correlação fraca a moderada. O valor máximo do dia (High) representa a volatilidade intradiária, mas não é utilizado como um fator decisivo pelo modelo. Ainda assim, pode ser útil em dias com fortes oscilações de mercado.

Comparação: Regressão Linear vs Árvore de Decisão



Low vs Predicted:

Compara o menor preço do dia com a previsão do fechamento subsequente. Assim como High, o Low descreve a amplitude do pregão, mas apresenta baixa influência direta na predição. A alta dispersão dos pontos sugere que o modelo não se baseia fortemente nessa variável para gerar estimativas consistentes.

Close vs Predicted:

Este é o gráfico mais relevante da análise, pois relaciona o preço de fechamento atual com a previsão do fechamento do dia seguinte. Os pontos seguem uma linha fortemente ascendente, indicando alta correlação. Isso evidencia que o modelo utiliza o fechamento anterior como principal referência para a previsão seguinte — o que é esperado em séries temporais financeiras, onde o último preço costuma ser o melhor preditor do próximo.

Volume vs Predicted:

Compara o volume de ações negociadas no dia com o valor previsto de fechamento. A alta dispersão dos pontos indica baixa correlação, demonstrando que o modelo não depende significativamente dessa variável. Embora o volume seja um bom indicador de interesse de mercado, isoladamente não define a direção dos preços.

MM5 vs Predicted (Média Móvel de 5 dias):

Relaciona a média dos últimos cinco fechamentos com a previsão. Observa-se uma forte correlação linear positiva, reforçando que a MM5 ajuda a suavizar oscilações pontuais e indicar tendências de curto prazo. O modelo parece utilizar essa variável como uma âncora de tendência, evitando reações excessivas a variações diárias isoladas.

Vol5 vs Predicted (Volatilidade de 5 dias):

Compara a volatilidade dos retornos dos últimos cinco dias com o valor previsto. A alta dispersão dos pontos indica baixa ou nenhuma correlação direta com a previsão. Isso mostra que o modelo não ajusta suas estimativas com base na volatilidade recente. Contudo, a variável Vol5 ainda pode ajudar a quantificar o grau de incerteza, mesmo que não afete diretamente o valor previsto.

Avaliação do Modelo

- **MAE – Mean Absolute Error:**
  Mede o erro médio absoluto entre predições e valores reais:
```
mae = mean_absolute_error(y_test, model.predict(X_test))
```

- **R² - Coeficiente de Determinação**
  Indica a proporção da variância explicada pelo modelo:
```
r2 = r2_score(y_test, model.predict(X_test))
```

Os resultados foram:

- **Regressão Linear:**
  MAE: 0.3252
  R^2: 0.7771

- **Árvore de Decisão Regressora:**
  MAE: 1.5121
  R^2: -3.0145

Conclusão

A análise comparativa entre a Regressão Linear e a Árvore de Decisão Regressora evidenciou diferenças significativas no desempenho dos modelos. A Regressão Linear obteve resultados expressivamente melhores, com um Erro Absoluto Médio (MAE) de 0.3252 e um coeficiente de determinação (R^2) de 0.7771, indicando boa capacidade de ajuste e previsão dos preços de fechamento com base nos atributos técnicos utilizados.

Em contrapartida, a Árvore de Decisão Regressora apresentou desempenho insatisfatório, com um MAE de 1.5121 e um R^2 negativo (-3.0145), o que sugere que o modelo foi incapaz de generalizar adequadamente os padrões dos dados. Um R^2 negativo indica que as previsões foram piores do que simplesmente usar a média dos valores reais como predição, caracterizando um modelo com baixo poder explicativo nesse contexto.

Esse contraste reforça que, para este problema específico — com atributos majoritariamente lineares e baixa complexidade de interação —, a Regressão Linear é a abordagem mais apropriada. Além disso, a análise destacou a importância de variáveis como Close e MM5, enquanto Volume e Vol5 apresentaram baixa relevância preditiva em ambos os modelos.

Como proposta de continuidade, recomenda-se a inclusão de variáveis externas (como notícias) e a experimentação com modelos mais robustos e não lineares, como florestas aleatórias, boosting ou redes neurais, que podem oferecer maior capacidade de generalização e ajuste em contextos mais complexos.