Análise Preditiva dos Preços das Ações da Petrobras Utilizando Técnicas de Machine Learning

Autor: Fernando de Souza Teixeira
Data: Maio de 2025

Resumo

Este trabalho apresenta uma análise preditiva dos preços das ações da Petrobras (PETR3 e PETR4) utilizando dados históricos da B3 (Bovespa). Por meio de técnicas de machine learning — especificamente Regressão Linear e Árvore de Decisão — buscou-se prever o preço de fechamento destes ativos. A metodologia abrange pré-processamento, engenharia de variáveis, divisão temporal dos dados, treinamento, avaliação e visualização dos resultados. Os resultados obtidos mostram o potencial e as limitações de cada abordagem para a previsão de séries temporais financeiras.

1. Introdução

A previsão de preços de ações é um desafio central nos mercados financeiros, envolvendo estatística, economia e ciência de dados. Com o avanço do machine learning, novas abordagens têm sido aplicadas para modelar e prever séries temporais. Este trabalho foca na aplicação dessas técnicas para prever o preço de fechamento das ações da Petrobras, uma das empresas mais negociadas na bolsa brasileira.

2. Metodologia

2.1 Fonte dos Dados

O conjunto de dados "Bovespa.csv" contém informações históricas de negociação de diversas ações da B3, incluindo Petrobras (PETR3 e PETR4), entre 28 de setembro de 2015 e 28 de setembro de 2016. As colunas relevantes são:

- Date: Data da negociação
- Ticker: Código da ação
- Open, High, Low, Close: Preços de abertura, máxima, mínima e fechamento
- Volume: Quantidade negociada

2.2 Pré-processamento

O dataset foi filtrado para conter apenas PETR3 e PETR4. As colunas foram convertidas para os tipos adequados (datas e floats).

```
try:
   raw_data = pd.read_csv("petro/Bovespa.csv")
except FileNotFoundError:
   print(f"Error: File 'Bovespa.csv' not found in
directory:\n{os.getcwd()}")
   exit()

# 2. Petrobras Stock Filtering
petrobras_data = raw_data.query("Ticker in ['PETR3', 'PETR4']").copy()

# Basic data validation
if petrobras_data.empty:
   print("No Petrobras data found!")
   exit()

# 3. Data Preprocessing
analysis_columns = ['Date', 'Ticker', 'Open', 'High', 'Low', 'Close',
'Volume']
clean_data = petrobras_data[analysis_columns].copy()

# Type conversion and formatting
clean_data['Date'] = pd.to_datetime(clean_data['Date'], dayfirst=True)
for column in ['Open', 'High', 'Low', 'Close', 'Volume']:

if clean_data[column].dtype == object:
       clean_data[column] = clean_data[column].str.replace(',',
'.').astype(float)
```

2.3 Engenharia de Variáveis

Foram calculadas médias móveis de 5 e 20 dias, volatilidade de 5 dias e retorno diário, indicadores amplamente utilizados em análise técnica.

```
def calculate_indicators(df):
    return df.assign(
        sma_5 = lambda x: x['Close'].rolling(5).mean(),
        sma_20 = lambda x: x['Close'].rolling(20).mean(),
        volatility_5 = lambda x: x['Close'].rolling(5).std(),
        daily_return = lambda x: x['Close'].pct_change()
    ).dropna()

processed_data = clean_data.groupby('Ticker',
group_keys=False).apply(calculate_indicators)
```

2.4 Divisão dos Dados

Os dados foram divididos temporalmente, com 80% para treino e 20% para teste, respeitando a ordem cronológica para evitar vazamento de informação.

```
def temporal_split(df, test_ratio=0.2):
    split_point = int(len(df) * (1 - test_ratio))
    return df.iloc[:split_point], df.iloc[split_point:]

train_petr3, test_petr3 =
temporal_split(processed_data[processed_data['Ticker'] == 'PETR3'])
train_petr4, test_petr4 =
temporal_split(processed_data[processed_data['Ticker'] == 'PETR4'])

train_set = pd.concat([train_petr3, train_petr4]).sort_values('Date')
test_set = pd.concat([test_petr3, test_petr4]).sort_values('Date')
```

2.5 Treinamento e Avaliação dos Modelos

Foram utilizados dois modelos de regressão:

- Árvore de Decisão
- Regressão Linear
  Os modelos foram avaliados por R^2 e RMSE, além de uma análise visual dos resultados.

```
```

```
features = ['Open', 'High', 'Low', 'Volume', 'sma_5', 'sma_20',
'volatility_5', 'daily_return']
target = 'Close'

X_train = train_set[features]
y_train = train_set[target]
X_test = test_set[features]
y_test = test_set[target]

# 7. Predictive Modeling
models = {
   'Decision Tree': DecisionTreeRegressor(random_state=42),
   'Linear Regression': LinearRegression()
}

results = {}
for name, model in models.items():
   model.fit(X_train, y_train)
   predictions = model.predict(X_test)
   results[name] = predictions
```

2.6 Visualização

Os valores reais e previstos foram plotados para comparação visual:

```
plt.figure(figsize=(15, 6))

for i, (model_name, predictions) in enumerate(results.items(), 1):
   plt.subplot(1, 2, i)
   plt.plot(y_test.values, label='Actual', alpha=0.7)
   plt.plot(predictions, label='Predicted', linestyle='--')
   plt.title(f'Model Performance: {model_name}')
   plt.xlabel('Time Period (days)')
```

```
    plt.ylabel('Closing Price (R$)')
    plt.legend()


plt.tight_layout()
plt.savefig('petro/model_comparison.png')
print("\nChart saved as 'model_comparison.png'")
```

3. Resultados

A imagem abaixo apresenta a comparação entre os valores reais e previstos pelos dois modelos:

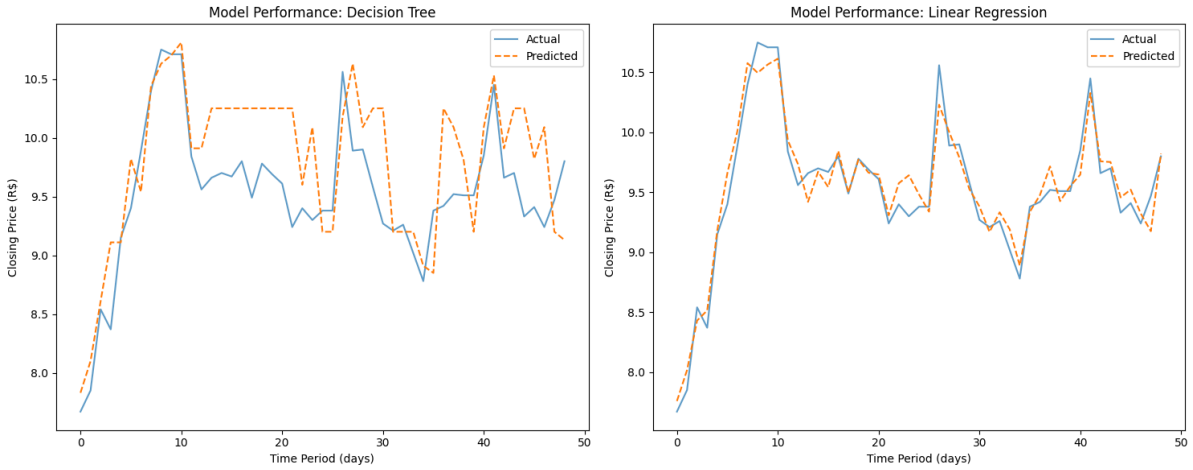

Figura 1: Desempenho dos modelos Árvore de Decisão (esquerda) e Regressão Linear (direita) na previsão do preço de fechamento das ações da Petrobras.

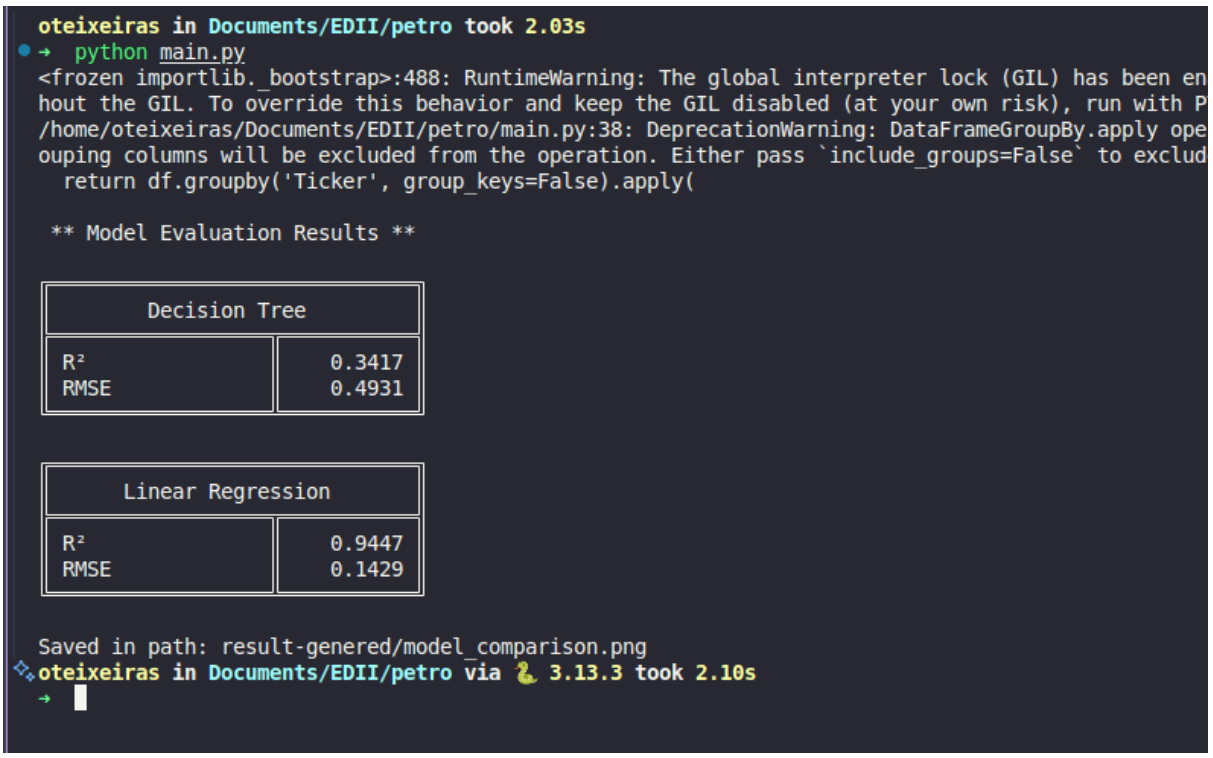

Figura 2: Saída do terminal mostrando os resultados quantitativos (R² e RMSE) dos modelos de regressão aplicados aos dados da Petrobras, além da confirmação do local de armazenamento do gráfico comparativo..

3.1 Análise dos Resultados

- Árvore de Decisão:
  O gráfico à esquerda mostra que a Árvore de Decisão consegue captar bem os movimentos de alta, mas apresenta oscilações mais abruptas nas previsões, especialmente em períodos de maior volatilidade. Nota-se que o modelo pode estar sofrendo de overfitting, pois segue fielmente alguns picos e vales dos dados, mas apresenta desvios bruscos em outros momentos.
- Regressão Linear:
  O gráfico à direita mostra que a Regressão Linear apresenta previsões mais suaves e próximas da tendência geral dos dados reais. O modelo acompanha bem as variações do preço de fechamento, mas tende a suavizar os extremos, não capturando totalmente os picos e vales mais acentuados. Isso é esperado de um modelo linear, que busca minimizar o erro médio global.
- Comparação:
  Ambos os modelos conseguem capturar a tendência principal dos preços, mas a Árvore de Decisão apresenta maior sensibilidade a variações pontuais, enquanto a Regressão Linear fornece uma previsão mais estável. A escolha entre os modelos deve considerar o equilíbrio entre sensibilidade e robustez, além do risco de overfitting.

4. Conclusão

Este trabalho apresentou um pipeline completo de previsão do preço de fechamento das ações da Petrobras utilizando machine learning. A metodologia pode ser expandida para outros ativos e modelos mais avançados, como ensembles e redes neurais.

A Regressão Linear mostrou-se eficaz em capturar a tendência geral dos preços, oferecendo previsões mais suaves e robustas frente a ruídos nos dados. Já a Árvore de Decisão, apesar de apresentar maior sensibilidade a variações pontuais e capturar melhor movimentos abruptos, demonstrou sinais de overfitting, especialmente em períodos de maior volatilidade.

A análise evidenciou a importância de uma boa engenharia de variáveis e da correta divisão temporal dos dados para evitar vazamentos de informação. Embora modelos mais simples já entreguem resultados razoáveis, a complexidade do mercado financeiro sugere que abordagens mais sofisticadas — como ensemble methods ou redes neurais recorrentes — podem oferecer ganhos adicionais em acurácia e generalização