

Análise Técnica do Script `manipulador.py` para Modelagem Preditiva de Ações da Petrobras

Felipe Moura, Talles Martins

20 de maio de 2025

Introdução

Este relatório descreve tecnicamente o funcionamento do script `manipulador.py`, criado para manipular dados históricos de ações da Petrobras (PETR3 e PETR4) provenientes do arquivo `Bovespa.csv`, com vistas à aplicação de modelos preditivos de machine learning. O objetivo é preparar os dados para previsão do preço de fechamento utilizando regressão linear.

1 Função `carregar_arquivo`

A função `carregar_arquivo(caminho_arquivo, formato_arquivo, delimitador)` é responsável por importar o conjunto de dados para um `DataFrame` do `pandas`, permitindo a leitura de arquivos nos formatos `csv` ou `excel`.

- `formato_arquivo`: Define o tipo de arquivo a ser carregado ("`csv`" ou "`excel`").
- `delimitador`: No caso de arquivos `CSV`, esse parâmetro especifica o caractere que separa as colunas (ex: vírgula ou ponto e vírgula).

Essa abstração facilita o uso de diferentes fontes de dados com mínima alteração no código.

2 Função `limpar_dados`

Essa função realiza a limpeza inicial do conjunto de dados:

- Converte a coluna `Date` para o formato `datetime`, com `dayfirst=True` para refletir o padrão brasileiro (dia/mês/ano).
- Substitui vírgulas por pontos nas colunas numéricas (`Close`, `Open`, `High`, `Low`, `Adj Close`) e converte-as de `string` para `float`.
- Ordena os dados cronologicamente com `sort_values("Date")`.

Isso garante a integridade temporal da série e o correto tratamento dos dados para análise quantitativa.

3 Função adicionar_colunas_calculadas

Esta função adiciona dois indicadores técnicos:

- **Média Móvel de 5 dias (MA_5):** Média dos últimos 5 preços de fechamento, calculada por ticker.
- **Volatilidade de 5 dias (Volatility_5):** Desvio padrão dos últimos 5 preços de fechamento, também por ticker.

Utiliza-se `groupby("Ticker")` para garantir que os cálculos sejam feitos separadamente para PETR3 e PETR4. O uso de `transform` assegura que as novas colunas mantenham o mesmo tamanho do `DataFrame`. Ao final, remove-se as linhas com valores nulos resultantes das janelas rolantes.

4 Função aplicar_RL

Esta função realiza a modelagem com Regressão Linear:

1. **Seleção de variáveis:** `X` recebe as variáveis explicativas (`Open`, `High`, `Low`, `Volume`, `MA_5`, `Volatility_5`) e `y` o alvo (`Close`).
2. **Divisão treino-teste:** Utiliza `train_test_split` com `shuffle=False` para preservar a ordem temporal.
3. **Treinamento:** Instancia e ajusta um modelo de regressão linear com `model.fit(X_train, y_train)`.
4. **Previsão e avaliação:** Calcula-se o erro quadrático médio (MSE) e o coeficiente de determinação (R^2).
5. **Visualização:** Gera um gráfico comparando os valores reais e previstos do preço de fechamento.

5 Bloco de Execução

Este bloco define o fluxo de execução principal:

1. Carrega o arquivo `Bovespa.csv`.
2. Filtra os registros da PETR3.
3. Limpa os dados.
4. Adiciona indicadores técnicos.
5. Define as variáveis preditoras e alvo.
6. Executa a modelagem com regressão linear.

O uso de `if __name__ == "__main__"` permite a reutilização do código como módulo sem executar o fluxo principal diretamente.

6 Melhorias Sugeridas

- **Tratamento de Outliers:** Verificar e corrigir valores extremos nos dados.
- **Mais indicadores técnicos:** Adicionar outros indicadores financeiros (RSI, MACD) para enriquecer o modelo.

Conclusão

O script `manipulador.py` apresenta uma estrutura sólida para o tratamento e análise preditiva de séries temporais financeiras. Ele utiliza corretamente conceitos fundamentais de ciência de dados, com boa organização modular, clareza e práticas recomendadas. As melhorias sugeridas visam tornar o modelo mais robusto e generalizável em contextos reais de previsão de ativos financeiros.