Meu Processo de Desenvolvimento do Script de Análise de Dados

Data: 04 de Junho de 2025

1. Objetivo Inicial

Decidi criar um script Python para realizar uma análise de dados completa e implementar um modelo básico de Machine Learning. Meu objetivo era ter um código modular, utilizando Pandas, NumPy, Matplotlib, Seaborn e Scikit-learn, que fosse fácil de adaptar e usar em um ambiente Jupyter Notebook. O fluxo planejado incluía carregamento de dados de um CSV, Análise Exploratória (EDA) com visualizações, pré-processamento e um modelo de Regressão Linear.

2. Desenvolvimento Inicial e Dataset

Comecei desenvolvendo as funções modulares para cada etapa: carregamento, EDA, visualização, pré-processamento e modelagem/avaliação. Para testar o fluxo, gerei um dataset sintético (`dados_exemplo.csv`) com colunas numéricas, categóricas e alguns valores ausentes.

Estruturei o script (`analise_ml.py`) com comentários e marcações (`# %%`) para facilitar o uso em notebooks, inicialmente com um bloco `if __name__ == '__main__':` para permitir a execução direta.

3. Organização da Estrutura do Projeto

Para manter o projeto organizado, adotei uma estrutura de pastas padrão para ciência de dados:

- `data/` (com subpastas `raw/` e `processed/`)
- `src/` (para o código Python)
- `notebooks/`
- `models/`
- `reports/`

Movi o `dados_exemplo.csv` para `data/raw/` e o `analise_ml.py` para `src/`. Criei também arquivos essenciais como `.gitignore`, `README.md` (com a descrição e instruções), `requirements.txt` (listando as dependências) e um `__init__.py` vazio em `src/`.

4. Adaptação para Execução Interativa no Notebook

Ao testar o script célula a célula no meu ambiente de notebook, percebi que as chamadas de função não eram executadas como esperado. Identifiquei que o bloco `if __name__ == '__main__':` impedia a execução interativa das chamadas.

Refatorei o script removendo esse bloco e movendo as chamadas de função para células `# %% [code]` separadas no final do arquivo, garantindo que cada etapa pudesse ser executada sequencialmente.

5. Resolução de Erros `NameError`

Durante a execução célula a célula, encontrei erros do tipo `NameError: name '...' is not defined`. Concluí que isso ocorria porque eu estava tentando executar células que usavam funções ou variáveis antes de executar as células onde elas eram definidas.

Para evitar isso no futuro, adicionei um comentário de lembrete no script, antes das células de execução, reforçando a necessidade de executar todas as células de definição (importações e funções) primeiro. Também adicionei uma verificação do diretório de trabalho para ajustar o caminho do CSV corretamente.

6. Resultado Final

Após essas etapas de desenvolvimento e refinamento, cheguei a um projeto de análise de dados bem estruturado e funcional:

- Script Python (`src/analise_ml.py`) modular e adaptado para execução interativa em notebooks.
- Dataset de exemplo (`data/raw/dados_exemplo.csv`).
- Estrutura de projeto organizada.
- Arquivos auxiliares e documentação (`.gitignore` , `README.md` , `requirements.txt`).
- Instruções claras no script sobre a ordem de execução das células.

Este processo iterativo me permitiu construir e depurar o fluxo de análise de forma eficaz.