

Documentação do Código de Predição com Machine Learning

William czpak, matheus pereira

Documentação Completa do Código de Predição com Machine Learning

Introdução

Este documento detalha um código em Python desenvolvido para realizar predições baseadas em dados históricos de países. A solução utiliza algoritmos de aprendizado de máquina para analisar indicadores temporais e gerar previsões de crescimento, variação ou desempenho para os próximos anos. O código é modular, facilitando ajustes e expansão.

Importações

- pandas e numpy: Manipulação de dados estruturados e numéricos.
- sklearn: Algoritmos de aprendizado de máquina (RandomForest, GradientBoost, SVM) e divisão de dados em treino/teste.
- datetime: Geração de carimbos de data/hora para relatórios.
- os: Manipulação de arquivos e diretórios.
- logging: Rastreamento de eventos e erros durante a execução.

Configuração de Logs

A configuração de logging permite rastrear eventos e erros no código. O nível de log configurado como INFO registra informações gerais e mensagens de erro, úteis para depuração.

Função: `get_model`

Esta função retorna uma instância de um modelo de regressão configurado de acordo com o tipo

Documentação do Código de Predição com Machine Learning

especificado. Os tipos suportados incluem:

- random_forest: Utiliza RandomForestRegressor.
- gradient_boost: Utiliza GradientBoostingRegressor.
- svm: Utiliza SVR.

Parâmetros:

- classifier_type: Tipo de modelo (string).
- params: Dicionário com parâmetros adicionais para ajuste fino.

Função: train_and_predict_next_year

Essa função é responsável por treinar um modelo com os dados históricos de um país específico e prever valores futuros para o próximo ano. O fluxo geral inclui:

1. Preparação dos dados: Limpeza e organização.
2. Treinamento e validação: Divisão em treino e teste, ajuste do modelo e cálculo do score.
3. Previsão: Geração de previsões para o próximo ano.

Parâmetros:

- data: DataFrame com os dados completos.
- target_col: Indicador alvo a ser previsto.
- country: País de interesse.
- classifier_type: Modelo a ser utilizado (default: random_forest).
- classifier_params: Parâmetros opcionais para o modelo.

Função: generate_predictions

Documentação do Código de Predição com Machine Learning

Esta função aplica a lógica de previsão a todos os países disponíveis no conjunto de dados, gerando resultados organizados por indicador e país. Também calcula variações esperadas entre valores atuais e previstos.

Parâmetros:

- df: DataFrame com os dados estruturados.
- classifier_type: Tipo de modelo preditivo.
- classifier_params: Parâmetros opcionais.

Função: **save_predictions_report**

Gera e salva um relatório detalhado das previsões em um arquivo de texto, organizado por país e indicador. Inclui valores atuais, previsões, variações e informações do modelo utilizado.

Local de saída padrão: diretório 'output', com o nome do arquivo 'previsoes_sul_asia.txt'.

Conclusão

Este código combina técnicas robustas de ciência de dados e aprendizado de máquina para gerar insights preditivos baseados em séries temporais. A modularidade do código permite fácil adaptação para diferentes cenários e indicadores.

Documentação do Código Flask para Análise de Dados

Este documento explica cada bloco do código fornecido, que implementa uma aplicação Flask para upload, análise e predição de dados baseados em aprendizado de máquina. O sistema também gera visualizações e relatórios.

Imports e Configurações Iniciais

O código inicia importando os módulos necessários, como Flask, Pandas, e bibliotecas de visualização (Plotly, Matplotlib). Também configura logging para registro de erros e informações.

Configuração da Aplicação Flask

A aplicação Flask é criada, configurando pastas para upload, arquivos estáticos, e o tamanho máximo de upload. Uma chave secreta é definida para gerenciamento de sessões.

Funções Auxiliares

- `analyze_missing_data(data)`: Analisa e visualiza dados ausentes.
- `analyze_feature_importance(data)`: Analisa a importância de variáveis para o crescimento populacional.
- `generate_graphs(data)`: Gera gráficos interativos baseados nos dados carregados.
- `load_data()`: Carrega os dados do CSV e prepara para análise.
- `allowed_file(filename)`: Verifica se o arquivo tem a extensão permitida (CSV).
- `validate_csv_structure(file_path)`: Valida se o arquivo CSV contém as colunas esperadas.

Rotas da Aplicação

- `/`: Página inicial para upload de arquivos.

- /upload: Processa o upload e validação do arquivo.
- /predictions: Exibe previsões baseadas nos dados.
- /graphs: Mostra gráficos interativos gerados dos dados.
- /classifier_config: Permite configurar o classificador de ML.
- /save_classifier_config: Salva as configurações do classificador.
- /output/<path:filename>: Serve arquivos gerados.

Configuração do Classificador

Um classificador padrão é definido como Random Forest, mas a aplicação permite alterar os parâmetros ou escolher outros classificadores como SVM ou Gradient Boost.

Execução do Servidor

No final, o servidor Flask é iniciado em modo debug, permitindo testes e desenvolvimento local.