

Documentação do Projeto em Flask - Análise de Emoções

Sumário

- 1.Introdução
- 2.Requisitos do Sistema
- 3.Estrutura de Diretórios
- 4.Dependências Externas
- 4.Configuração do Ambiente de Desenvolvimento
- 5.Arquitetura do Código
- 5.Pré-processamento de Dados
- 6.Modelagem de Dados
- 7.Rotas e Funcionalidades
- 8.Interface de Usuário
- 9.Exemplos de Uso
- 10.Considerações Finais
- 11.Referências

1. Introdução

O projeto "Análise de Emoções" tem como objetivo classificar emoções em texto usando um modelo de Machine Learning. Este projeto é uma aplicação web Flask que permite aos usuários inserir texto e prever a emoção associada.

2. Requisitos do Sistema

Python 3.6 ou superior

Flask 2.0.1

pandas 1.3.3

scikit-learn 0.24.2

3. Estrutura de Diretórios

app.py: O arquivo principal do aplicativo Flask.

index.html: Página inicial do aplicativo.

resultado.html: Página de resultado da previsão de emoções.

basetreinamento1.csv: Arquivo CSV contendo dados de treinamento.

baseteste1.csv: Arquivo CSV contendo dados de teste.

4. Dependências Externas

Flask: Framework web utilizado para desenvolver a aplicação.

pandas: Biblioteca para manipulação de dados em formato CSV.

scikit-learn: Biblioteca para Machine Learning e processamento de texto.

5. Configuração do Ambiente de Desenvolvimento

Clone o repositório do projeto.

Crie um ambiente virtual usando `python -m venv venv`.

Ative o ambiente virtual com `source venv/bin/activate` (Linux/macOS) ou `venv\Scripts\activate` (Windows).

Instale as dependências com `pip install -r requirements.txt`.

Execute o aplicativo com `python app.py`.

6. Arquitetura do Código

O código do projeto está organizado da seguinte maneira:

app.py: Contém a lógica principal do aplicativo Flask.

templates/: Diretório que contém os arquivos HTML das páginas.

static/: Diretório opcional para arquivos estáticos como CSS e JavaScript.

7. Pré-processamento de Dados

Os dados de treinamento e teste são carregados a partir dos arquivos CSV, e quaisquer valores nulos são preenchidos com strings vazias. Em seguida, as frases e emoções são extraídas dos dados.

8. Modelagem de Dados

Usamos o CountVectorizer para converter o texto em vetores numéricos e treinamos um modelo Naive Bayes para prever emoções com base nos vetores.

9. Rotas e Funcionalidades

/: Página inicial do aplicativo.

/prever: Rota para prever a emoção com base no texto inserido pelo usuário.

10. Interface de Usuário

As páginas HTML index.html e resultado.html contêm os elementos da interface de usuário, permitindo aos usuários inserir texto e visualizar os resultados.

11. Exemplos de Uso

Acesse a página inicial.

Insira um texto e clique em "Prever".

Visualize a emoção prevista e as probabilidades associadas.

12. Considerações Finais

Este projeto demonstra a capacidade de classificar emoções em texto usando técnicas de Processamento de Linguagem Natural (PLN) e Machine Learning. Possíveis melhorias incluem aprimoramento do modelo e expansão da interface de usuário.

13. Referências

Documentação do Flask: <https://pypi.org/project/Flask/>

Documentação do pandas: <https://pandas.pydata.org/>

Documentação do scikit-learn: <https://scikit-learn.org/stable/>