Eduardo Coelho

GitHub: https://github.com/eduardoocoelho/ai-lists

Questão 1:

1. **Leitura dos Dados:**
    a. Leitura utilizando o **pandas**. Os parâmetros **escapechar** e **quotechar** são utilizados para lidar com caracteres especiais.
2. **Pré-processamento de Texto:**
    a. A função **preprocess_text** é definida para realizar o pré-processamento de cada texto. Ele remove pontuações e realiza tokenização e lematização usando a biblioteca **NLTK**.
3. **Aplicação do Pré-processamento aos Dados:**
    a. Os textos nos conjuntos de treinamento e teste são pré-processados, convertidos para minúsculas e caracteres de nova linha são removidos. A função **preprocess_text** é aplicada a cada texto.
4. **Divisão em Dados de Treino e Teste:**
    a. Os dados são divididos em recursos (X) e rótulos (y) tanto para treinamento quanto para teste.
5. **Vetorização TF-IDF:**
    a. A biblioteca scikit-learn é utilizada para vetorizar os textos usando o método **TF-IDF** (Term Frequency-Inverse Document Frequency).
6. **Contagem de Valores de Classes:**
    a. As contagens de valores únicos para as classes nos conjuntos de treinamento e teste são exibidas. No entanto, essas linhas não estão atribuindo os resultados a variáveis, então elas não têm impacto direto no restante do código.
7. **Peso de Classes Balanceado:**
    a. Os pesos de classe balanceados são calculados usando a função **compute_class_weight** da biblioteca scikit-learn.
8. **Treinamento do Classificador:**
    a. Um classificador Naive Bayes Multinomial é inicializado com os pesos de classe calculados e é treinado usando os dados de treinamento vetorizados.
9. **Previsão e Avaliação do Modelo:**
    a. O modelo treinado é usado para fazer previsões nos dados de teste, e a acurácia e o relatório de classificação são exibidos.

Questão 2:

1. **Pré-processamento de texto**
    a. Uma função **preprocess_text** é definida para realizar o pré-processamento do texto. Ela remove pontuações e aplica a lematização para reduzir as palavras à sua forma base.
2. **Carregamento dos dados**
    a. Os dados de treinamento e teste são carregados a partir de arquivos CSV. Os rótulos multirrótulo (seis categorias) são extraídos para **y_train**.
3. **Divisão dos dados**
    a. Os dados de treinamento são divididos em conjuntos de treino e validação.
4. **Vetorização TF-IDF**
    a. Os textos são convertidos em representações numéricas usando a vetorização **TF-IDF**.
5. **Treinamento do Modelo**
    a. Um classificador Naive Bayes é treinado separadamente para cada rótulo.
6. **Avaliação do Modelo no Conjunto de Validação**
    a. As previsões são feitas no conjunto de validação, e a precisão e o relatório de classificação são exibidos.
7. **Previsões no Conjunto de Teste**
    a. Previsões são feitas no conjunto de teste.
8. **Combinação das Previsões e Salvar em um CSV**
    a. As previsões no conjunto de teste são combinadas em um DataFrame e salvas em um arquivo CSV chamado 'predictions.csv'.