Análise de sentimentos em reviews de filmes no Letterboxd

Tipo de Atividade: Prático

Área de Estudo: Processamento de linguagem natural

Estimativa de Prazo: Médio

Quantidade de Pessoas: 3 - 4

Descrição da Proposta de Atividade:

Através de diversas reviews de filmes na rede social “Letterboxd”, a proposta é treinar um modelo capaz de identificar o sentimento da review, se foi positivo ou negativo, expresso pelo autor da crítica em relação ao filme. Será um projeto de Processamento de Linguagem Natural, utilizando técnicas como pré-processamento de texto e modelos supervisionados, e, a princípio, terá essa categorização binária de “positivo ou negativo”, porém, pode ser expandido para análises mais sutis em relação ao sentimento de quem escreveu, ou comparação entre sentimento e a nota atribuída.

Prompt usado: "Pegando diversas reviews de filmes na rede social “Letterboxd”, a proposta é treinar um modelo capaz de identificar o sentimento da review, se foi positivo ou negativo, e potencialmente identificar a opinião da pessoa em relação ao filme com base na crítica escrita. Preciso da descrição de um breve direcionamento do projeto, as etapas e os assuntos necessários para estudar como pré requisito para participar ativamente do projeto"

Direcionamento do Projeto: Análise de Sentimentos de Reviews de Filmes

Este projeto visa desenvolver um sistema que automatiza a classificação de reviews de filmes do Letterboxd, determinando o sentimento expresso (positivo ou negativo) e, por consequência, a opinião do usuário. Utilizaremos técnicas de Processamento de Linguagem Natural (PLN) e Machine Learning para construir e treinar um modelo robusto, capaz de lidar com a complexidade da linguagem humana em contextos de crítica cinematográfica.

Etapas do Projeto

Para alcançar nosso objetivo, o projeto será dividido nas seguintes etapas principais:

1. Coleta e Pré-processamento de Dados

- **Coleta de Reviews:** Extração de um grande volume de reviews de filmes da plataforma Letterboxd. Isso pode envolver o uso de APIs, web scraping (observando os termos de serviço da plataforma) ou conjuntos de dados pré-existentes.
- **Limpeza de Dados:** Remoção de ruídos (emojis, caracteres especiais, URLs, tags HTML, etc.), padronização de texto (minúsculas, remoção de acentos), tratamento de contrações e gírias.
- **Tokenização:** Divisão do texto em unidades menores (palavras ou sentenças).
- **Remoção de Stop Words:** Exclusão de palavras comuns que não agregam valor significativo ao sentimento (e.g., "o", "a", "de", "para").
- **Lematização/Stemização:** Redução das palavras à sua forma base (e.g., "correndo" vira "correr", "casas" vira "casa").

2. Análise Exploratória de Dados (AED)

- Entender a distribuição dos sentimentos (se já tivermos dados rotulados) ou a necessidade de rotulagem.
- Identificar as palavras mais frequentes em reviews positivas e negativas.
- Visualizar a extensão das reviews e outras características que possam influenciar o modelo.

3. Rotulagem dos Dados (se necessário)

- Se os dados coletados não tiverem o sentimento já atribuído, será necessário um processo de rotulagem manual ou semi-automática para criar um conjunto de treinamento supervisionado. Isso envolve classificar cada review como positiva ou negativa.

4. Engenharia de Features

- **Vetorização de Texto:** Transformar o texto em representações numéricas que os modelos de Machine Learning possam entender. As técnicas comuns incluem:
    - **Bag-of-Words (BoW):** Representa a frequência de cada palavra em um documento.
    - **TF-IDF (Term Frequency-Inverse Document Frequency):** Pondera a importância de uma palavra em relação a um documento e ao corpus.
    - **Word Embeddings (Word2Vec, GloVe, FastText):** Representações densas de palavras que capturam relações semânticas e sintáticas.
    - **Embeddings Contextuais (BERT, GPT, RoBERTa):** Representações que levam em consideração o contexto da palavra na frase, oferecendo resultados mais sofisticados.

5. Seleção e Treinamento do Modelo

- **Escolha do Algoritmo:** Seleção de modelos adequados para classificação de texto. Opções incluem:
    - **Modelos Tradicionais de Machine Learning:** Naive Bayes, Support Vector Machines (SVM), Regressão Logística.
    - **Redes Neurais:** Redes Neurais Recorrentes (RNNs, LSTMs, GRUs), Redes Neurais Convolucionais (CNNs) para texto, e principalmente **Modelos Pré-treinados de Linguagem (Transformers como BERT, DistilBERT, mBERT)** que são o estado da arte para PLN.
- **Divisão dos Dados:** Separar o conjunto de dados em treino, validação e teste.
- **Treinamento do Modelo:** Alimentar o modelo com os dados de treinamento para que ele aprenda a mapear as features textuais aos sentimentos.
- **Otimização de Hiperparâmetros:** Ajustar os parâmetros do modelo para melhorar seu desempenho.

6. Avaliação do Modelo

- Utilizar métricas como **Acurácia, Precisão, Recall, F1-Score e Curva ROC** para avaliar o desempenho do modelo no conjunto de dados de teste.
- Análise de erros para identificar onde o modelo está falhando.

7. Implantação (Opcional, mas recomendado)

- Criar uma interface simples (ex: API Flask/FastAPI, Streamlit) para que o modelo possa receber novas reviews e classificar seu sentimento em tempo real.

Assuntos Necessários para Estudar (Pré-requisitos)

Para participar ativamente e extrair o máximo deste projeto, os seguintes conhecimentos são fundamentais:

1. Fundamentos de Programação

- **Python:** É a linguagem padrão para projetos de Ciência de Dados e Machine Learning. Domínio de sintaxe, estruturas de dados (listas, dicionários, tuplas), funções, classes e manipulação de arquivos.

2. Matemática e Estatística

- **Álgebra Linear:** Compreensão básica de vetores, matrizes, operações matriciais, que são a base para representações de dados e operações em redes neurais.
- **Cálculo (Básico):** Derivadas e gradientes são essenciais para entender como os modelos otimizam seus pesos.
- **Estatística Descritiva e Inferencial:** Média, mediana, moda, desvio padrão, variância, distribuição de probabilidade, testes de hipóteses. Isso é crucial para a AED e a avaliação dos modelos.

3. Machine Learning (Aprendizado de Máquina)

- **Conceitos Fundamentais:** Tipos de aprendizado (supervisionado, não supervisionado, por reforço), overfitting, underfitting, viés e variância.

- **Algoritmos de Classificação:** Entendimento de como funcionam algoritmos como Regressão Logística, SVM, Naive Bayes, Árvores de Decisão e Random Forests.
- **Métricas de Avaliação:** Compreensão profunda de acurácia, precisão, recall, F1-Score, matriz de confusão.
- **Ciclo de Vida de um Projeto de ML:** Desde a coleta de dados até a implantação.

4. Processamento de Linguagem Natural (PLN/NLP)

- **Conceitos Básicos de PLN:** Tokenização, stop words, stemização, lematização, n-grams.
- **Representação de Texto:** Bag-of-Words, TF-IDF.
- **Word Embeddings:** Entendimento do conceito de Word2Vec, GloVe e FastText.
- **Arquiteturas de Redes Neurais para PLN (Desejável):** Introdução a RNNs, LSTMs, GRUs, e o funcionamento básico da arquitetura Transformer.
- **Modelos Pré-treinados de Linguagem (Desejável):** Conhecimento de BERT e como usá-lo para fine-tuning.

5. Bibliotecas Python Essenciais

- **Numpy:** Para operações numéricas e arrays multidimensionais.
- **Pandas:** Para manipulação e análise de dados (DataFrames).
- **Scikit-learn:** Para pré-processamento, seleção de modelos de ML tradicionais e avaliação.
- **NLTK ou SpaCy:** Para tarefas de PLN como tokenização, stemização, lematização.
- **Matplotlib / Seaborn:** Para visualização de dados.
- **TensorFlow ou PyTorch:** Para construir e treinar modelos de Redes Neurais, especialmente se for usar Transformers.
- **Hugging Face Transformers:** Biblioteca essencial para trabalhar com modelos Transformer pré-treinados.