Movie Reviews Classification Project

labeledTrainData.tsv
File

so this is my dataset for my project for movie reviews classification Use this dataset containing movie reviews to
build a model that can classify them as
positive or negative help me to build my project by step by step

Memory updated

Let's walk through your movie review classification project step by step. Since you've provided a dataset in `.tsv` format, I'll help you with data preprocessing, model building, and evaluation.

Step-by-Step Plan:

1. **Load and Explore the Data**
    - First, we need to load the dataset and explore its structure, including the columns, data types, and whether it has any missing values.
2. **Data Preprocessing**
    - Clean the text data by removing HTML tags, special characters, etc.
    - Tokenize and convert the text to a numerical format (like Bag of Words or TF-IDF).
    - Split the dataset into training and testing sets.
3. **Build a Model**
    - Choose a machine learning algorithm (e.g., Naive Bayes, Logistic Regression, or a deep learning model like LSTM).
    - Train the model on the preprocessed dataset.
4. **Evaluate the Model**
    - Evaluate the model using metrics like accuracy, precision, recall, F1-score, and a confusion matrix.
5. **Fine-tune the Model**
    - Improve the model's performance by tuning hyperparameters or trying different algorithms.
6. **Deployment (optional)**
    - If you'd like, we can deploy the model via a web app or API (using Flask, for example).

Let's start with **Step 1: Loading and Exploring the Data**. I'll inspect your dataset first to understand its structure.

Your dataset contains 25,000 entries with three columns:

- **id**: Unique identifier for each review.
- **sentiment**: Target label, where `` `1` `` indicates a positive review and `` `0` `` indicates a negative review.
- **review**: The actual text of the movie review.

Step 2: Data Preprocessing

Next, we'll clean the text data in the "review" column:

1. **Remove unwanted characters** like HTML tags, special symbols, etc.
2. **Tokenize** the text into words.
3. **Convert the text into numerical form** (e.g., using TF-IDF or word embeddings).

Would you prefer to use TF-IDF or word embeddings like Word2Vec/Glove for text representation?

ChatGPT can make mistakes. Check important info.