Project 20: IMDB Sentiment Analysis [Mehrdad]

This project will investigate sentiment analysis on a dataset of 50k IMDB movie reviews. The goal is to classify reviews as positive or negative based on the textual content. This is a dataset for binary sentiment classification containing 25,000 highly polar movie reviews for training and 25,000 for testing.

1- Data Exploration: Compute basic statistics (mean, median, min, max) for the length of the reviews (measured by the number of words and characters). Visualize the distribution of review lengths using histograms or box plots. Compare the length of reviews for positive and negative sentiments.

2- Data Cleaning: Perform standard text preprocessing tasks, including: Removing stop words, punctuation, and special characters, Lowercasing the text, Tokenizing the reviews, Stemming or lemmatization.

3- Review Word Frequency Analysis: *1- Most Common Words:* Generate a word frequency count for both the positive and negative reviews. Visualize the most frequent words using word clouds and bar charts, with separate visualizations for positive and negative reviews. and *2- N-gram Analysis*: Explore bi-grams and tri-grams (two- and three-word combinations) to uncover common phrases in the reviews. Separate this analysis for positive and negative reviews to observe any recurring phrases that indicate strong sentiment.

4. Correlation Between Review Length and Sentiment: Calculate the correlation between sentiment and review length and analyze whether longer or shorter reviews tend to be positive or negative. Plot the average review length for positive and negative reviews to explore potential patterns.

5- Review Complexity Analysis: Calculate readability metrics such as the Flesch-Kincaid or Gunning Fog Index for each review to assess its complexity. This analysis explores whether there is a correlation between the complexity of the review and its sentiment (e.g., do more complex reviews tend to be positive or negative?). Also, lexical diversity, which is the ratio of unique words to total words, will be measured for each review. The lexical diversity of positive and negative reviews will then be compared to identify any differences in language richness between the two sentiment classes.

6. Part-of-Speech (POS) Tagging Analysis: Apply POS tagging to the text data to categorize words into grammatical categories such as nouns, verbs, adjectives, etc. The aim is to explore the frequency and distribution of various parts of speech in both positive and negative reviews. For instance, investigate whether adjectives and adverbs, which often convey sentiment, occur more frequently in positive reviews compared to negative ones.

7- Sentiment Analysis on Review Topics: Using topic modeling techniques such as Latent Dirichlet Allocation (LDA), identify the main topics in the reviews. After identifying these topics, analyze the sentiment of the reviews within each topic to determine if specific topics tend to evoke more positive or negative sentiment.

8- Comparison of Pre-Trained Sentiment Models with Ground Truth Labels: Use pre-trained models like VADER and TextBlob to analyze the sentiment of the reviews, assigning a sentiment polarity score (positive, and negative) to each review. To compare these sentiment scores with the ground truth labels (positive or negative) in the dataset, create a confusion matrix to assess how well the pre-trained models align with the true labels. Calculate evaluation metrics such as accuracy, precision, recall, and F1-score for both VADER and TextBlob. Additionally, visualize the distribution of sentiment polarity scores generated by these models for both the positive and negative labeled reviews to identify any discrepancies between the model predictions and the actual labels. Use bar plots or stacked bar plots to show the agreement or disagreement between the pre-trained models' polarity scores and the true sentiment labels. Also, conduct an error analysis to inspect cases where the pre-trained models significantly differ from the ground truth (e.g., reviews labeled as positive but classified as negative by VADER), and discuss patterns or common characteristics of reviews where these models perform poorly, such as sarcasm or ambiguous language.

9- Feature Extraction for Sentiment Classification: Convert the text reviews into numerical representations suitable for machine learning models. First, apply the Bag of Words (BoW) method, which represents the text based on word frequency without considering word order. Next, implement TF-IDF to assign higher importance to less frequent but more meaningful

words in the reviews. Finally, explore word embeddings such as Word2Vec, GloVe, or BERT to capture more advanced and contextual word representations, providing richer semantic information for the sentiment classification models.

10- Sentiment Prediction Using Extracted Features: Build a sentiment classification model using the features extracted in Task 13. Train the model on the training dataset using features extracted via Bag of Words (BoW), TF-IDF, and word embeddings such as Word2Vec, GloVe, or BERT. After training, evaluate the performance of the model on the test dataset. The goal is to predict whether a review is positive or negative based on these numerical representations. You are required to compare the performance of various classifiers, including Logistic Regression, Support Vector Machines (SVM), Random Forest, and Deep Learning models (LSTM or CNN). Each classifier will be applied to BoW, TF-IDF and word embeddings, and the results should be evaluated using metrics such as accuracy, precision, recall, and F1-score.

11- Fine-Tuning NLP Feature Extraction Techniques: In this task, you will focus on optimizing the NLP feature extraction methods used in the sentiment classification models. Begin by exploring different configurations of the BoW, TF-IDF techniques, such as adjusting the n-gram range, using unigrams, bigrams, or trigrams, and varying the maximum number of features to include in the vocabulary. For word embeddings, experiment with fine-tuning Word2Vec, GloVe, or BERT by adjusting parameters like the window size, embedding dimension, and training epochs for Word2Vec, or by applying BERT fine-tuning with different learning rates and batch sizes. Once you havve fine-tuned these feature extraction techniques, apply them to the training dataset and evaluate how these changes impact the model's performance on the test dataset. You are required to compare the optimized feature extraction methods with the baseline configurations used in Task 10. Evaluate the results using metrics such as accuracy, precision, recall, and F1-score, and provide insights into how adjusting the NLP-based features influences the overall sentiment prediction performance.

12- Advanced NLP Techniques and Suggestions: Feel free to explore additional NLP techniques or state-of-the-art models to enhance the sentiment analysis task. You can experiment with transformer-based models such as RoBERTa, DistilBERT, or GPT, comparing their performance to the previously used methods like BERT or traditional word embeddings.

13- Identify appropriate literature to discuss the findings and comment on the strength and weakness of the data processing pipeline.