Final Report: Sentiment Analysis of IMDB Movie Reviews

Task 1: Data Exploration and Preprocessing - I worked with a dataset containing IMDB movie reviews labelled as either positive or negative. I began by exploring the dataset to understand its structure. I verified that there were no missing values and found that the class distribution between positive and negative reviews was fairly balanced.

To gain more insight into the nature of the reviews, I created a new feature called review_length, which measures the length of each review. I plotted the distribution of review lengths and found that most reviews fell within a consistent range, with a few outliers.

For preprocessing, I cleaned the text by converting it to lowercase, removing HTML tags, special characters, and punctuation using regular expressions. I also normalized whitespace. I did not include tokenization, stop word removal, stemming, or lemmatization in this version, but I recognize that these steps could improve model performance and plan to include them in future work.

Task 2: Feature Engineering - I transformed the cleaned text into numerical features using TF-IDF vectorization with a limit of 5000 features. This allowed me to capture the importance of each word in the corpus relative to other documents.

While TF-IDF was effectively implemented, I did not include additional textual features such as word count, character count, or average word length. I also did not use embedding-based techniques like Word2Vec or GloVe at this stage. These could be considered for future improvement, especially when training deep learning models.

Task 3: Model Development - I developed several machine learning models using the TF-IDF features. These included Logistic Regression, Multinomial Naive Bayes, Support Vector Machine (SVM), and Random Forest. All models were trained and evaluated on the same train-test split to ensure consistency.

In addition to the traditional models, I implemented a deep learning model using LSTM. I tokenized and padded the review texts, then built a sequential model using an embedding layer, an LSTM layer, and a dense output layer with sigmoid activation. I trained the LSTM model for 3 epochs.

Task 4: Model Evaluation - I evaluated all models using accuracy, precision, recall, F1 score, confusion matrix, and classification report.

Here are the evaluation results:

- Logistic Regression achieved 81% accuracy.
- Naive Bayes achieved 81.5% accuracy.
- SVM achieved the highest accuracy at 83.5%.
- Random Forest achieved 77.5% accuracy.
- LSTM achieved approximately 73.5% accuracy.

The SVM model performed the best overall. The LSTM model did not perform as well, likely due to the limited number of training epochs and the absence of advanced preprocessing steps.

Conclusion - This project covered the complete sentiment analysis process, from data exploration to model evaluation. Traditional machine learning models, especially SVM and Logistic Regression, performed well using TF-IDF features. The LSTM model showed potential but would benefit from additional training and preprocessing improvements.

For future work, I plan to:

- Include stop word removal, stemming, and lemmatization.
- Add engineered features such as word and character counts.
- Use pre-trained embeddings like GloVe or Word2Vec.
- Train the LSTM model with more epochs and deeper layers.
- Explore transformer-based models like BERT to improve accuracy further.

Video Link - Here