**Sentiment Analysis of Movie Reviews**

**1. Project Overview**

This project aims to classify movie reviews into two categories: positive and negative. By analyzing the sentiment expressed in the text of the reviews, this model can help in understanding consumer reactions and improving recommendation systems. The dataset consists of 50,000 reviews, equally divided between the two sentiment classes.

**2. Data Acquisition**

The data was sourced from a comprehensive dataset containing 50,000 movie reviews. Each review is labeled with either 'positive' or 'negative', indicating the sentiment.

**3. Data Exploration and Preprocessing**

**3.1 Data Cleaning**

- Duplicate Removal: The dataset contained a small number of duplicate entries, which were identified and removed to ensure the integrity of the analysis.

**3.2 Data Visualization & Analysis:**

* Plot a histogram that showed that reviews range from 10 to 14000 characters and generally, it is between 10 to 1500 characters.
* Then we plot histogram for positive review and negative review respectively.
* In general, people comment less word in the positive review to compare with negative review However the range of word for positive review are bigger than the range of negative review. It means in some cases, people give a long comments for excellent movies and people could less critise for bad movies
* a wordcloud graph to show the most used words in large font and the least used words in small font in positive review and negative review
* The wordcloud graphs in both negative and postitive comments don't show meaningful result.

**3.3 Text Preprocessing**

- Preprocess the text data by removing stop-words, punctuation, removing URL links, special characters, emoji, short form and converting text to lowercase.

Then Working with the most Frequent Words and removing it.

* Tokenization and Lemmatization\*\*: reviews were tokenized (breaking text into individual words) and lemmatized (reducing words to their base form).

**4. Feature Engineering**

* Convert the preprocessed text data into numerical representations suitable for machine learning models.
* Use techniques like bag-of-words (BoW) Representation: We initialize a CountVectorizer object. We fit and transform the corpus using the fit\_transform() method of CountVectorizer. This generates a sparse matrix where each row corresponds to a document in the corpus, and each column corresponds to a unique word in the corpus. The cell values represent the frequency of each word in the corresponding document.
* term frequency-inverse document frequency (TF-IDF), or word embed-dings (e.g., Word2Vec, GloVe) to represent textual features.
* Explore the use of n-grams and other text features to capture context and semantics in the movie reviews.

**5. Model Selection and Training**

**5.1 Model Selection**

Choose appropriate machine learning algorithms for sentiment analysis, such as naive Bayes. Split the dataset into training and testing sets for model evaluation. Train the selected models on the training data and fine-tune hyperparameters if necessary.

**5.2 Model Training**

The dataset was split into an 80/20 ratio for training and testing. Models were trained using the training set, and parameters were tuned to optimize performance.

**6. Model Evaluation**

Models were evaluated based on accuracy, precision, recall, and F1-score:

- Accuracy: Measures the overall correctness of the model (85.63 %)

- Precision and Recall: Important in scenarios where false positives and false negatives have different implications.

- F1-Score: Combines precision and recall into a single metric, which is useful when seeking a balance between these two metrics.

#### Trying different n-grams: accuracy decreased with 2 and 3 n-grams

#### TF-IDF: Term Frequency-Inverse Document Frequency: accuracy increased to 86.5%

**7. Conclusion and Future Work**

The sentiment analysis model achieved satisfactory performance, effectively classifying movie reviews. Future work could explore more complex models such as deep learning architectures or ensemble methods. Additionally, expanding the dataset and including reviews in different languages could provide more comprehensive insights.

**8. Challenges and Learnings**

The project highlighted the importance of thorough preprocessing and the impact of feature selection on model performance. The need for computational efficiency was also a key consideration, particularly in processing large text datasets.