IMDb Movie Review Sentiment Analysis

1) **Overview -** Sentiment analysis is a natural language processing (NLP) task that involves determining whether a given text expresses a positive or negative sentiment. In this project, we will analyze movie reviews from the IMDb dataset and predict the sentiment (positive or negative) based on the text of the reviews. By leveraging various text preprocessing techniques, feature extraction methods, and classification algorithms, this project will develop a machine learning model capable of accurately predicting the sentiment of movie reviews. The insights derived from this analysis can be useful for movie producers, critics, and platforms like IMDb to understand public opinion and tailor marketing or content strategies accordingly.
2) **Problem Statement -** The primary objective of this project is to build a machine learning classification model that can predict the sentiment of IMDb movie reviews. The dataset contains a collection of movie reviews, and each review is labeled as either positive or negative. Using text preprocessing, feature extraction techniques (such as TF-IDF), and various classification algorithms, the project will aim to develop a model that can effectively classify the sentiment of movie reviews. The model's performance will be evaluated using standard classification metrics, such as accuracy, precision, recall, and F1-score.
3) **Dataset Information** - The IMDb dataset contains a large number of movie reviews, each labeled with either a positive or negative sentiment.
    i) Text of the review: The actual review provided by the user.
    ii) Sentiment label: The sentiment of the review, either "positive" or "negative."

Dataset: Imdb

4) **Deliverables**

a) Data Exploration and Preprocessing

i) Analyze the dataset for trends, missing values, and outliers.
    (a) Perform basic data exploration, such as checking for missing values, identifying imbalanced classes (positive/negative), and analyzing the length of reviews.

ii) Perform data cleaning and text preprocessing.
    (1) Steps will include:
        (a) Removing stop words, punctuation, and special characters.
        (b) Tokenization of text (splitting text into words).
        (c) Lemmatization and stemming.
        (d) Vectorization using techniques like Bag-of-Words and TF-IDF.

b) Feature Engineering

i) Feature extraction using techniques like TF-IDF, Word2Vec, or embeddings.
    (1) Transform the textual data into numerical features that can be used by machine learning models.

ii) Textual features: Word count, character count, average word length, etc.

c) Model Development

i) Build and train classification models to predict the sentiment of reviews.

(a) Experiment with various classification algorithms such as Logistic Regression, Naive Bayes, Support Vector Machine (SVM), Random Forest, and Neural Networks (e.g., LSTM, BERT, etc.).

d) Model Evaluation

i) Evaluate the model's performance using appropriate metrics.

e) Final Report and Presentation

(1) **Create a final report** that documents the entire process, from data exploration and preprocessing to model evaluation.

(2) **Video presentation** (maximum 5 minutes) summarizing the key findings, model development process, and insights derived from the project.

f) Success Criteria

The project will be deemed successful if:

(1) The classification model achieves an acceptable performance on the test data using metrics like accuracy, F1-score, and ROC-AUC.

(2) Insights regarding the factors influencing sentiment (such as word frequency, review length, etc.) are clearly communicated.

(3) Predictions for new movie reviews can be made with a reasonable degree of accuracy.

(4) The final presentation effectively communicates the results and analysis of the project.

(5) Visualizations: Use plots, such as bar charts, confusion matrices, and word clouds, to clearly present data trends and model results.

g) Tools Required

i) **Python Libraries:** pandas, numpy, scikit-learn, matplotlib, seaborn, NLTK, spaCy, TensorFlow/Keras, XGBoost, etc.

ii) **Jupyter Notebook:** For developing and documenting the code.

iii) **Text Preprocessing Libraries:** NLTK, spaCy, scikit-learn (for vectorization), etc.

iv) **Visualization Libraries:** matplotlib, seaborn (for visualizing data distributions, model performance, etc.).