**Introduction:**

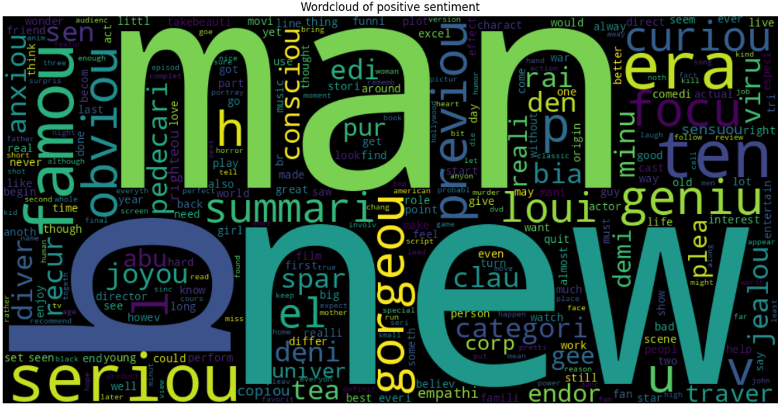
In the contemporary landscape of digital media, understanding audience sentiments towards movies and TV shows is paramount for filmmakers, production companies, and entertainment platforms. Extracting insights from user reviews, particularly from platforms like IMDb, offers valuable perspectives on audience preferences, satisfaction levels, and areas for potential improvement. This document outlines a comprehensive methodology for analyzing movie reviews, focusing on preprocessing, analysis, feature extraction, model selection, and future directions.

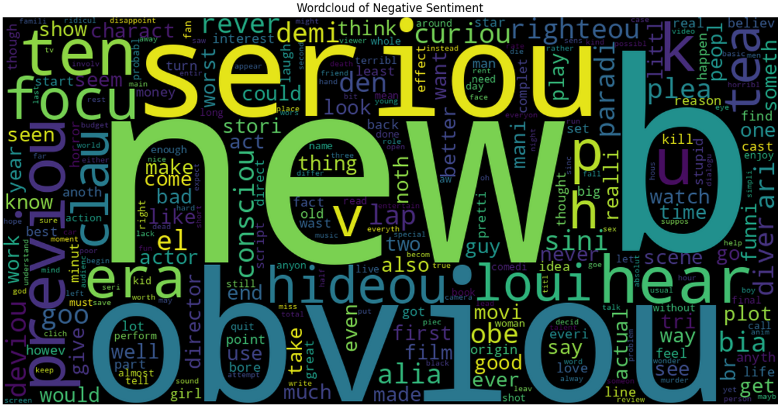
**Dataset Overview:**

The dataset comprises 5000 movie reviews, with each review accompanied by a corresponding sentiment label indicating positive or negative sentiment. Given resource constraints, an initial exploration was conducted on the first 2000 reviews, consisting of 1005 positive reviews and 995 negative reviews.

**Preprocessing and Analysis:**

Preprocessing began with the removal of punctuation, including special characters, HTML tags, and URLs, along with standardizing text to lowercase. Subsequently, tokenization was performed to break down sentences into individual words. Stopwords, common words that add little semantic value, were then eliminated using the NLTK library. Stemming, a technique to reduce words to their root forms, was applied using the Porter Stemmer algorithm. Additionally, word clouds were generated to visually represent high-frequency words in both positive and negative sentiment reviews. After preprocessing, the dataset comprised 239,712 words across 2000 reviews.



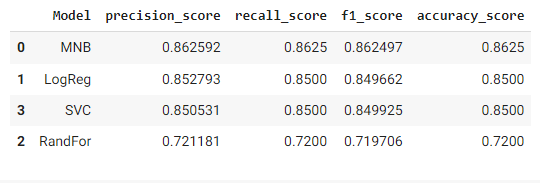


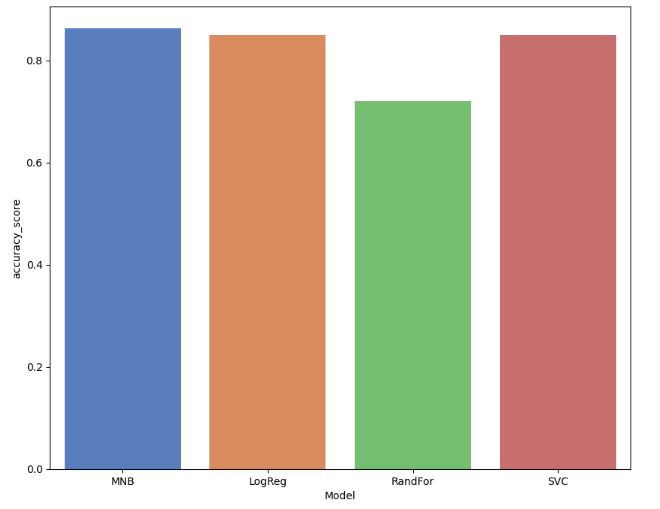
**Feature Extraction:**

TF/IDF vectorization was employed to convert the preprocessed text data into numerical vectors, considering both word occurrence and importance within each document. The dataset was split into 80% for training and 20% for testing.

**Model Selection:**

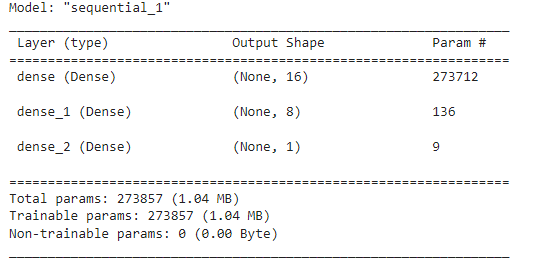
Four machine learning models were experimented with: Multinomial Naïve Bayes, Support Vector Classifier, Random Forest, and Linear Regression. Multinomial Naïve Bayes achieved the highest accuracy at 86%, while Random Forest attained the lowest at 72%. Result of ML models are given below:





Additionally, a deep learning approach using neural networks was explored. Sequential models in Keras were employed, comprising three dense layers with ReLU activation functions for the first two layers and a sigmoid activation function for the output layer. Different optimizers, including Adam and RMSprop, were evaluated. Configuration and summary of model is given below:

|  |  |
| --- | --- |
| **Hyper-parameter** | **value** |
| No. of layers | 3 |
| optimizer | Adam |
| epochs | 15 |
| Batch size | 10 |
| Activation function | sigmoid |



I experimented with same configuration expect changing the optimizer and input layers. The result evaluation on test data of each model is mentioned in the table.

|  |  |  |
| --- | --- | --- |
| **Models** | **Loss** | **Accuracy** |
| RSMprop optimizer (Model1) | 0.602 | 0.812 |
| Adam Optimizer (Model2) | 0.434 | 0.827 |
| LSTM layer (Model3) | 0.693 | 0.502 |

The performance comparison on test data revealed Multinomial Naïve Bayes outperforming the neural network models.

**Future Directions:**

To further enhance model accuracy, future directions include experimenting with word embedding techniques, such as Word2Vec or GloVe, which capture semantic relationships between words. Additionally, incorporating transformer-based models like BERT could provide more contextual understanding of textual data, thereby improving sentiment analysis outcomes.

Top of Form