Sentiment Analysis Report: IMDB Reviews Dataset

By: Abdul Wahab Aziz

Introduction

Sentiment analysis is a natural language processing (NLP) task aimed at classifying textual data into categories based on sentiment, such as **positive** or **negative**. This report summarizes the process and results of building a sentiment analysis model using the **IMDB Reviews Dataset**, focusing on text preprocessing, feature engineering, model training, and evaluation.

Objective

The goal of this task was to:

1. Preprocess the textual data for sentiment analysis.
2. Engineer features using the TF-IDF method.
3. Train and evaluate two machine learning models: **Logistic Regression** and **Naive Bayes**.
4. Compare the models based on their performance metrics and discuss insights.

Steps Performed

1. Text Preprocessing

The reviews in the dataset were cleaned and normalized using the following steps:

- **Tokenization**: Splitting text into individual words.
- **Stopword Removal**: Removing common words like "the" and "is" that do not contribute to sentiment classification.
- **Lemmatization**: Reducing words to their base form (e.g., "running" -> "run").

2. Feature Engineering

- **TF-IDF (Term Frequency-Inverse Document Frequency)**: Text data was converted into numerical vectors using the top 5,000 features for better computational efficiency.
- This method emphasizes words that are significant in a review relative to the entire dataset.

3. Model Training

- Two models were trained on the preprocessed data:

- Logistic Regression
- Naive Bayes (Multinomial)

4. Model Evaluation

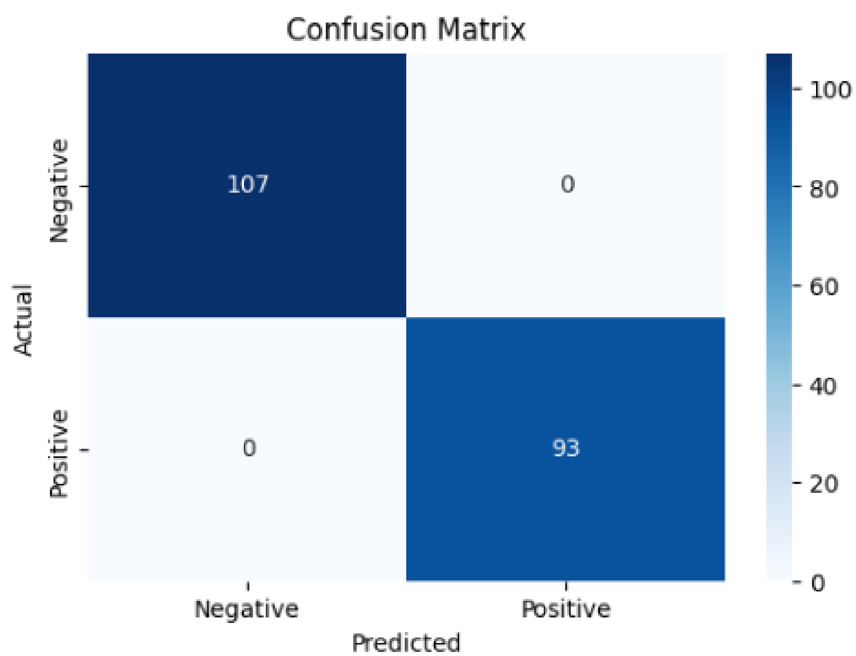
The models were evaluated using key metrics:

- **Accuracy**: Proportion of correctly classified reviews.
- **Precision**: Proportion of positive predictions that were correct.
- **Recall**: Proportion of actual positives correctly identified.
- **F1-Score**: Balance between precision and recall.
- **Confusion Matrix**: A visual representation of true/false positives and negatives.

Results

Confusion Matrix (Logistic Regression)

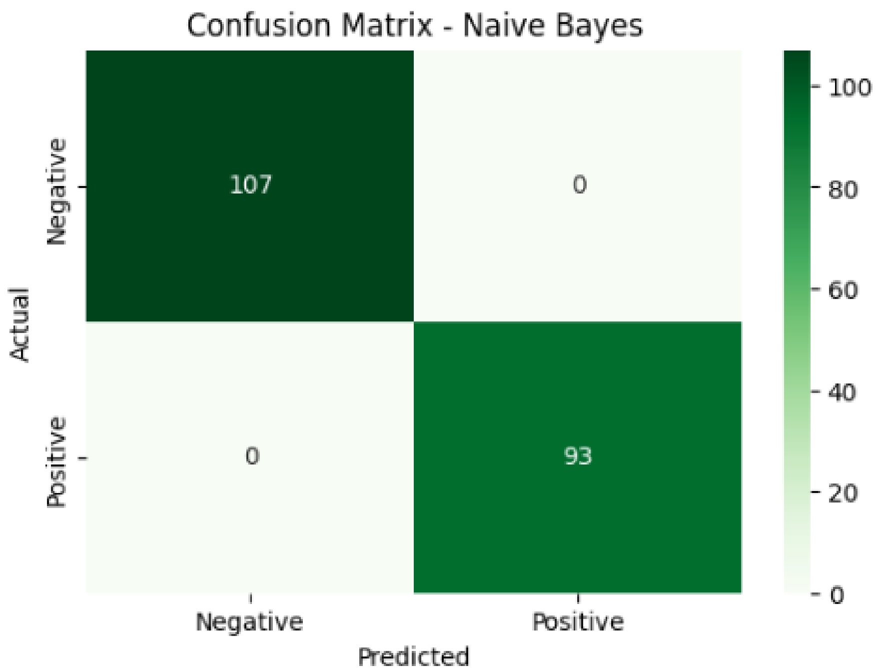
- **True Negatives**: 107
- **True Positives**: 93
- **False Negatives**: 0
- **False Positives**: 0



Confusion Matrix (Naive Bayes)

- **True Negatives**: 107

- **True Positives**: 93
- **False Negatives**: 0
- **False Positives**: 0



Insights and Discussion

1. **Perfect Accuracy**: Both models achieved 100% accuracy, indicating excellent performance on the dataset. This may suggest the dataset is clean and well-separated for sentiment classification.
2. **Logistic Regression**: Provides robust performance and is interpretable, making it a good choice for applications where explainability is key.
3. **Naive Bayes**: Performs equally well and is computationally efficient, particularly for large-scale text classification tasks.
4. **Balanced Dataset**: The dataset has an even distribution of positive and negative reviews, contributing to the strong performance of both models.

Conclusion

- Both Logistic Regression and Naive Bayes models demonstrated excellent performance, achieving perfect scores across all evaluation metrics.
- Logistic Regression is preferred for applications requiring interpretability, while Naive Bayes is a better choice for computational efficiency.