IMDB Movie Review Sentiment Classification

Lameya Islam[*]

[*] Faculty of Science & Engineering, Åbo Akademi University

1. Introduction

Sentiment Analysis is a critical task in Natural Language Processing (NLP) that aims to determine the sentiment expressed in textual data. This project focuses on classifying IMDB movie reviews into positive and negative sentiments using three distinct models: Naïve Bayes (NB), Long Short-Term Memory (LSTM), Support Vector Machine (SVM). The goal is to compare their performance and identify the most suitable approach for this binary classification problem.

2. Methodology

2.1 Dataset Description

The dataset used for this study is the IMDB Sentiment Dataset obtained from Hugging Face. It contains 50,000 movie reviews, evenly split into 25,000 reviews for training and 25,000 for testing. The reviews are labeled as positive and negative, ensuring a balanced dataset.

2.2 Data Preprocessing

Since raw text data cannot be directly fed into models, several preprocessing steps were performed:

a. **Text Cleaning:**
   i. Lowercasing all texts to ensure uniformity.
   ii. Removal of HTML Tags, special characters and extra spaces.

b. Tokenization & Vectorization:
   i. For Naïve Bayes (NB) and SVM, the text was converted into numerical format using TF-IDF Vectorization.
   ii. For LSTM, tokenization was performed using **Keras Tokenizer**, followed by padding sequences to ensure consistent input length.

2.3 Exploratory Data Analysis

1. **Class Distribution**: The histogram analysis shows a balanced distribution with the classes.

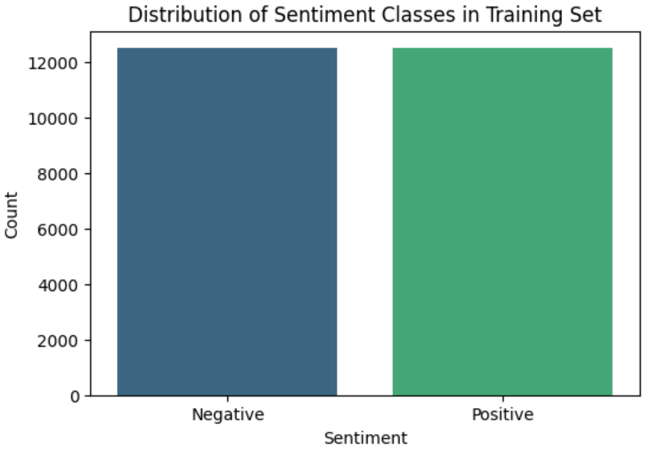


Figure 1: Distribution of classes (Positive and Negative)

2. **Most Common (Top 20) Words:** These words are most frequently found in the training dataset.

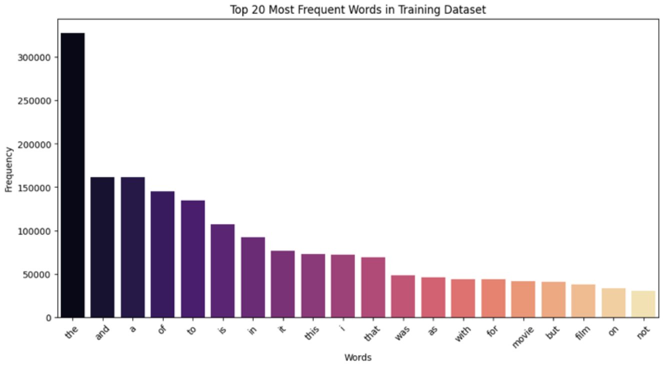


Figure 2: Top 20 Most Frequent Words

3. **Top 20 Bigrams**: Bigrams means two phrase words. The most used two-word phrases to

understand word combinations for sentiment analysis.

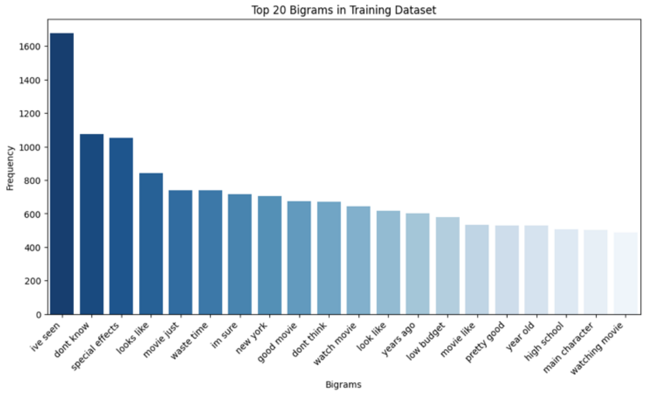


Figure 3: Top 20 Bigrams

4. **Top 20 Trigram**: Trigram means threephrase words. The most used three-word phrases to understand word combinations for sentiment analysis.

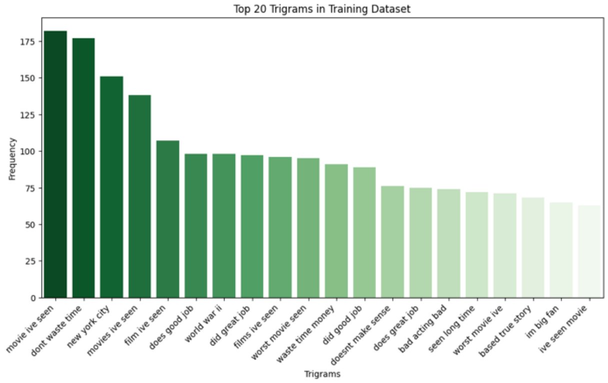


Figure 4: Top 20 Trigrams

5. **Top 20 words based on TF-IDF:** Words that are most relevant which terms are most influential in sentiment prediction.

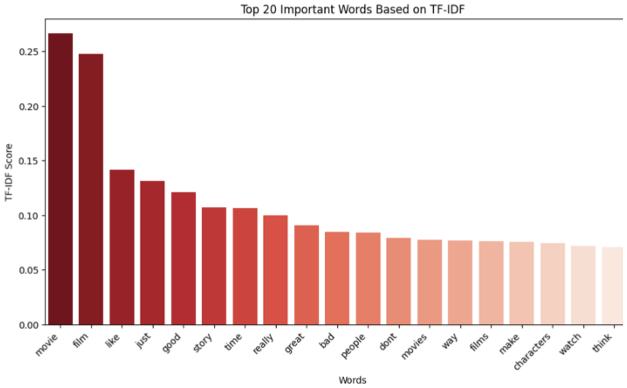


Figure 5: Top 20 Important Words based on TF-IDF

6. **Correlation between Word (Top 20) and Sentiment:** This correlation helps understand which words significantly impact sentiment classification and can be useful for improving sentiment analysis models.

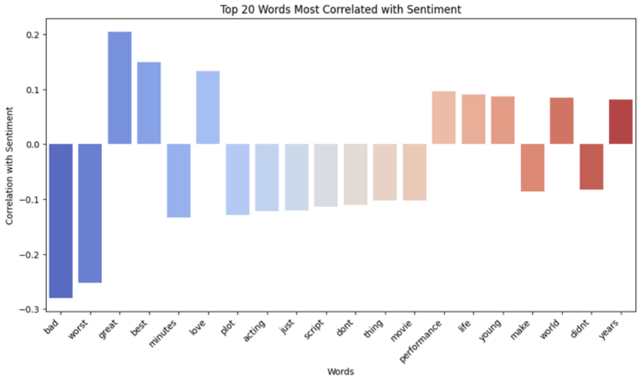


Figure 6: Correlation between Word and Sentiment

3. Modeling

3.1 Selected Algorithms

Three models were selected based on their ability to capture different patterns in the data:

1. **Naïve Bayes (NB):** A probabilistic classifier based on Bayes' Theorem. It's implemented using Multinomial Naïve Bayes with TF-IDF vectorization. It is suitable for text classification due to its simplicity and efficiency.

2. **Long Short-Term Memory (LSTM):** A Recurrent Neural Network (RNN) variant capable of capturing long-term dependencies in text. It's implemented using TensorFlow/Keras, with an embedding layer followed by LSTM layers and a dense output layer. It is trained using Adam optimizer with a binary cross-entropy loss function.

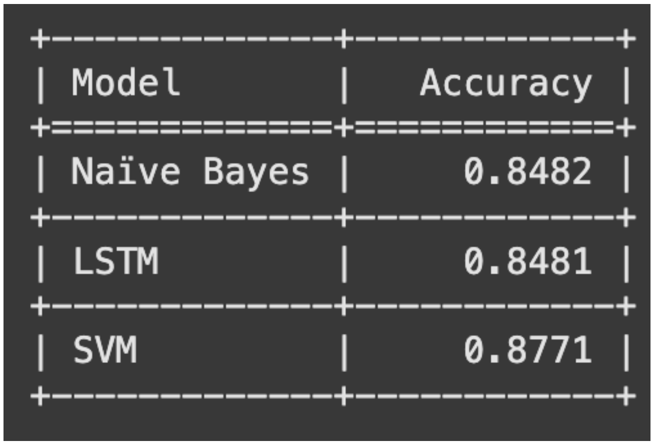
3. **Support Vector Machine (SVM):** A **supervised learning model** that finds an optimal hyperplane for classification. It's implemented using **TF-IDF vectorization** followed by an SVM classifier with

an RBF kernel. It is Effective for high-dimensional text classification.

3.2 Model Evaluation

The dataset was 25000 for **training (50%)** and 25,000 for **testing (50%)** sets.

Model's accuracy:



```
+-------------+------------+
| Model       |   Accuracy |
+=============+============+
| Naïve Bayes |     0.8482 |
+-------------+------------+
| LSTM        |     0.8481 |
+-------------+------------+
| SVM         |     0.8771 |
+-------------+------------+
```

Figure 7: Result of the Models

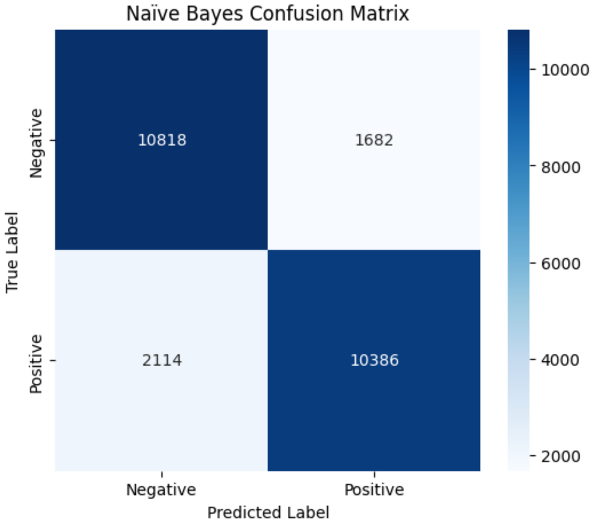


Figure 8: Confusion Matrix of Naïve bayes

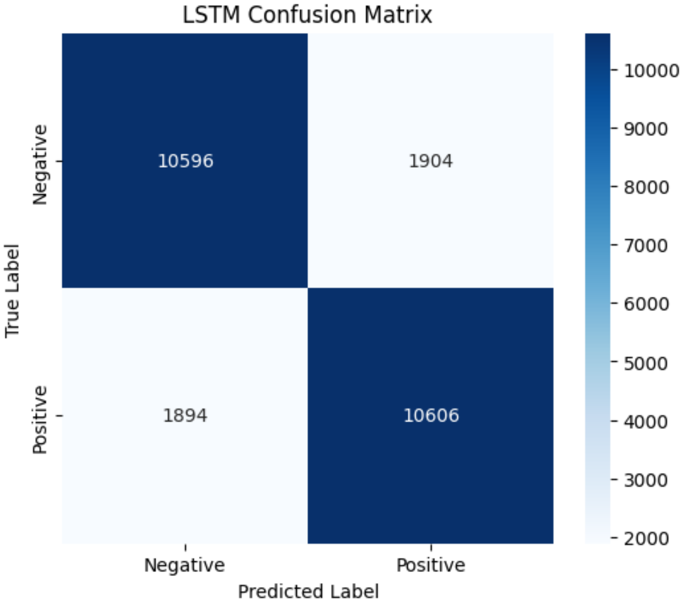


Figure 9: Confusion Matrix of LSTM

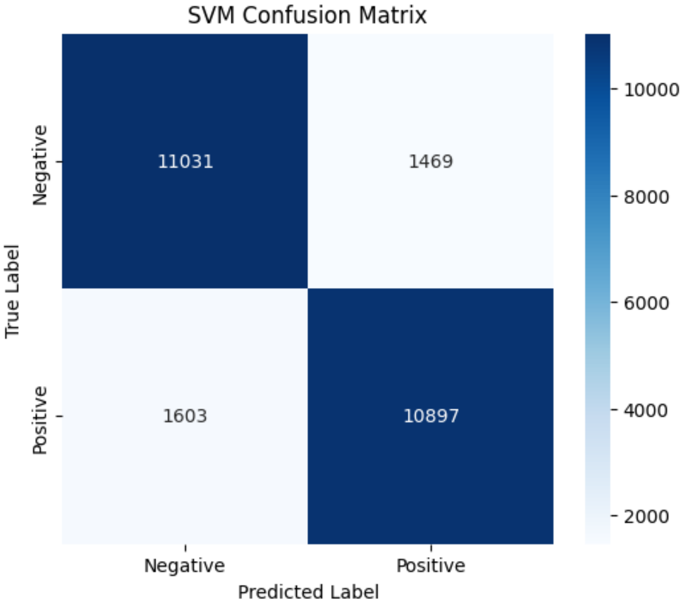


Figure 10: Confusion Matrix of SVM

3.3 Learning Curve Analysis

- **Naïve Bayes:** The Naïve Bayes model shows extreme overfitting at small training sizes, with training accuracy at **1.0** while validation accuracy remains low (~0.5). As training size increases, validation accuracy improves steadily, while training accuracy slightly decreases, indicating better generalization. However, the persistent gap between the two suggests **high bias**, meaning the model's assumptions (e.g., feature independence) may not fully capture the dataset's complexity. Increasing data helps, but the model may still struggle with intricate patterns.

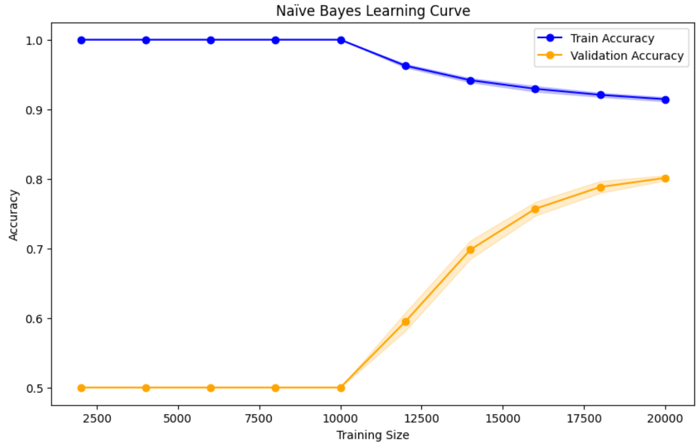


Figure 11: Naïve Bayes Learning Curve

- **LSTM**: The LSTM model learns quickly in the first few epochs, with both training and validation accuracy rising sharply. However, after a few epochs, validation accuracy starts fluctuating while training accuracy continues increasing, indicating **overfitting**. So, I have used **L2 regularization techniques** like dropout and early stopping to prevent memorization of training data.

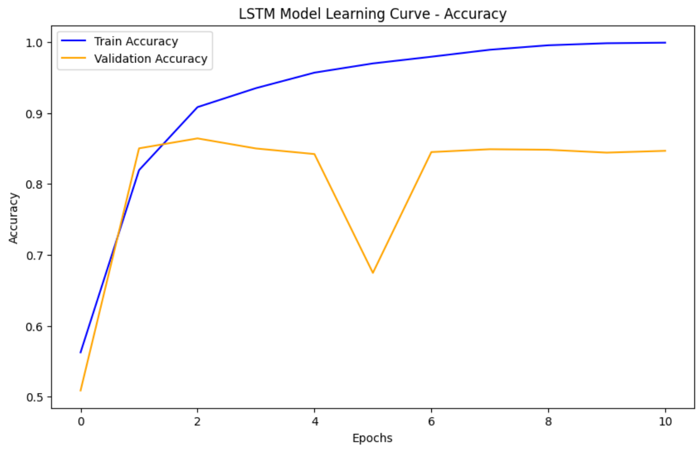


Figure 12: Learning Curve of LSTM

4. Conclusion

This study demonstrated that machine learning models can effectively classify and analyze sentiment through texts enabling operational optimization. Here, SVM emerged as the best-performing model, achieving an accuracy of 87.7%, offering a strong balance between efficiency and performance. LSTM and Naïve Bayes followed closely, demonstrating deep text understanding but requiring higher computational resources.

4.1 Bottlenecks

There are a few challenges that I faced:

1. **Computational Complexity:** LSTM required high processing power; GPU acceleration was used and time too.
2. **Overfitting in LSTM:** Implemented L2 regularization, dropout layers and early stopping.

After overcoming the bottlenecks, SVM turns out to be the best model with higher accuracy.