IMDB Movie Review Sentiment Analysis

Group 8 Team Members:

Name	Email ID	UNT ID
Srilakshmi Parella	SriLakshmiParella@my.unt.edu	11697665
Roopadevi Ambati	Roopadeviambati@my.unt.edu	11702517
Siva Kumar Chaini	SivaKumarChaini@my.unt.edu	11676099
Rohith Kumar Arvapally	RohithKumarArvapally@my.unt.edu	11694004
Srimannarayana Vakalapudi	SrimannarayanaVakalapudi@my.unt.edu	11819166

Abstract:

Sentiment analysis is an important NLP task applied in business intelligence, social media monitoring, and recommendation systems. The research project analyzes the evaluation results of machine learning together with deep learning frameworks which detect positive or negative reviews in IMDB movies. The investigation of Naive Bayes and Support Vector Machines (SVM), Random Forest, Decision Trees and Long Short-Term Memory (LSTM) networks found their performance values through accuracy, precision, recall and F1-score evaluation. The experimental results show that the LSTM model achieves the highest accuracy level of 87.65% while Naive Bayes follows with 84.91% and Random Forest comes in third place with 84.72%. The SVM and Decision Tree models performed worse, indicating the inability to manage high-dimensional text data. The present research contributes to model choice for sentiment analysis tasks.

1.Introduction:

The automated sentiment analysis system extracts subjective opinions from text data so business organizations can conduct large-scale public sentiment measurement. A total of 50,000 reviewed movies from IMDB provides perfect conditions for testing different classification systems. This research performs a comprehensive analysis through which five machine learning approaches of diverse types are evaluated across conventional and deep learning paradigms alongside detailed assessment of preprocessing effects on model performance. Every script and visualization has been released publicly with the aim to enable full reproducibility along with support for additional studies within this domain.

Contributions of this work:

1. Systematic comparison of five machine learning algorithms for sentiment analysis.
2. Analysis of both conventional ML and deep learning methods.
3. In-depth analysis of preprocessing methods and how they affect model performance.
4. Publicly accessible code and
5. Visualizations for reproducibility.

2. Background:

Sentiment analysis generally includes:

1. Text preprocessing (tokenization, stop word removal, stemming)
2. Feature extraction (TF-IDF, word embeddings)
3. Classification with supervised learning techniques

The sentiment analysis process includes standardized workflows which start with text preprocessing and then proceed to feature extraction and finish with supervised learning classification. The first step in text processing phases divides words into tokens while removing unnecessary stop words before converting words back to their base word forms through stemming procedures. The processed text receives numerical form through TF-IDF (Term Frequency-Inverse Document Frequency) method along with more sophisticated word embedding systems that preserve semantic connections in words. The classification algorithms receive these features in order to determine sentiment polarity.

The primary reason Naive Bayes methods persist despite challenges from deep learning methods is their computational speed and strong performance in standard applications though deep learning represents a major field advancement. The Long Short-Term Memory network (LSTM) has established itself as a top choice for pattern recognition and extended context understanding because it solves previous algorithms' word-order failure problem. The field exists between maintaining simpler reliable methods alongside neural architecture development that better identifies complex linguistic patterns which express sentiment in human language.

3. Experiment Methodology:

3.1 Dataset

We employed the IMDB dataset from Kaggle, which includes:

- This dataset contains an equal number of 25,000 positive and 25,000 negative reviews among its total of 50,000 movie assessment examples. The dataset has been divided into training and testing sections where 80 percent corresponds to 40,000 reviews and the remaining 20 percent comprises 10,000 reviews.

3.2 Preprocessing

- Lowercasing

- Removal of HTML tags
- Removal of punctuation
- Removal of stopwords (using NLTK)
- Stemming (Porter Stemmer)

3.3 Feature Extraction

- TF-IDF with 5,000 max features (for classical ML models)
- Word embeddings (for LSTM, sequence length = 200)

3.4 Models & Parameters

Model	Key Parameters
Naive Bayes	Default parameters
SVM	Linear kernel, max_iter=500
Random Forest	n_estimators=100, max_depth=None
Decision Tree	max_depth=5
LSTM	128-unit LSTM, dropout=0.2

3.5 Evaluation Metrics

- Accuracy
- Precision
- Recall
- F1-score

- Confusion matrices

4. Results:

4.1 Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
LSTM	87.65%	87.12%	88.59%	87.85%
Naive Bayes	84.91%	84.58%	85.67%	85.12%
Random Forest	84.72%	85.57%	83.81%	84.68%
SVM	71.60%	71.32%	72.99%	72.15%
Decision Tree	69.68%	64.03%	64.03%	75.13%

Key Observations:

- LSTM outperforms other models, benefiting from sequential text processing.
- Naive Bayes and Random Forest perform similarly, suggesting that simpler models can be effective for this task.
- SVM underperforms, likely due to high-dimensional sparse data.
- Decision Tree has high recall but low precision, indicating overfitting to the majority class.

4.2 Visualizations:

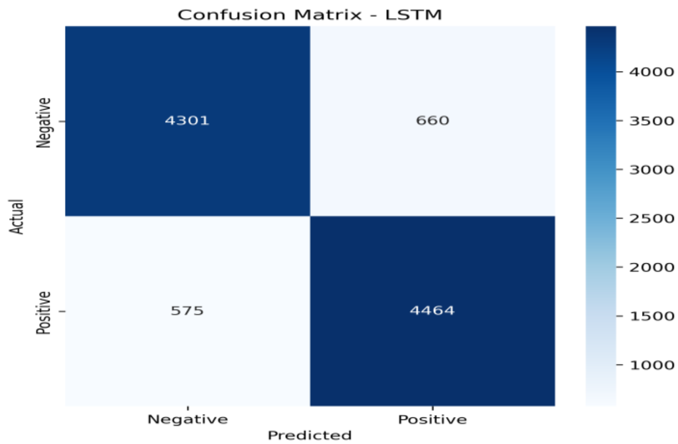

Figure 1: Confusion Matrix for LSTM

A confusion matrix indicates that the LSTM model performs extensively well in sentiment analysis through its 4301 true negatives and 4464 true positives indicating effective classification. The model yielded 660 false positive classifications along with 575 false negatives while doing accurate sentiment detection in most cases. Analysis shows that the model demonstrates proficient performance because its diagonal values remain high which indicates it correctly distinguishes positive and negative sentiment in most situations.

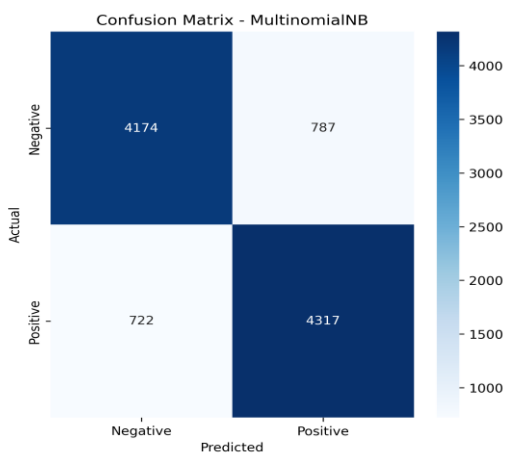

Figure 2: Confusion Matrix for Naive Bayes

The Multinomial Naive Bayes model achieves decent accuracy through its identification of 4174 true negatives and 4317 true positives for both sentiment classes. The Multinomial Naive Bayes model has a higher misclassification rate than LSTM since it produces 787 false positives combined with 722 false negatives. The

Naive Bayes model proves its effectiveness for sentiment analysis tasks despite its basic structure because it competes with traditional approaches and deep learning methods.

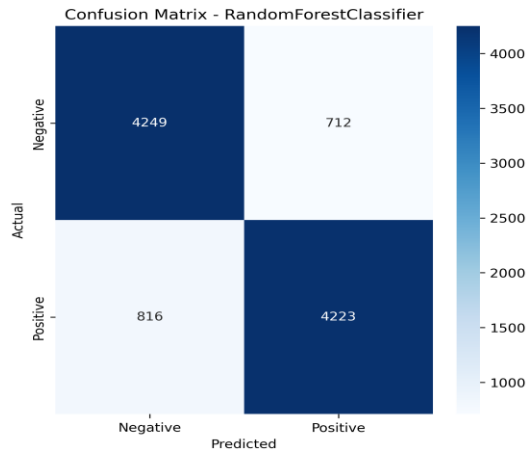

Figure 3: Confusion Matrix for Random Forest Classifier

It shows strong performance through its identification of 4249 true negatives and 4223 true positives equally across both sentiment classes. The model failed to correctly identify 816 negative and 712 positive examples in total while producing slightly fewer errors on positive cases than negative ones. The ensemble method demonstrates performance levels that match what both LSTM and Naive Bayes approaches achieve making tree-based methods suitable for sentiment analysis tasks.

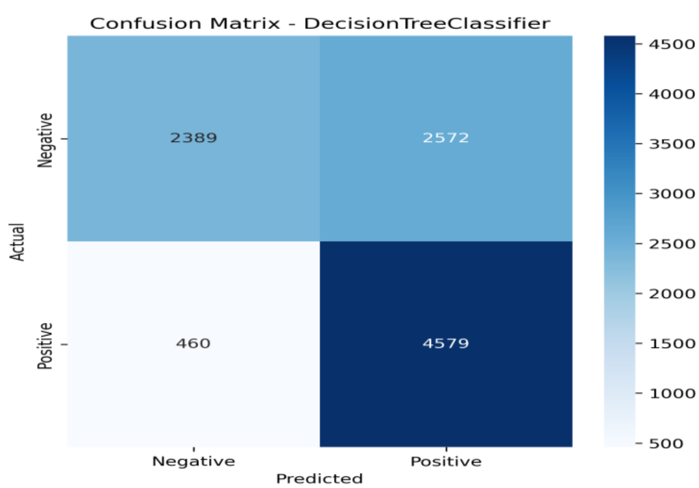

Figure 4: Confusion Matrix for DecisionTreeClassifier

Decision Tree Classifier produces inferior results following previous models because it fails to identify negative sentiments accurately. A total of 4579 correct positive predictions emerges while only 2389 true negative categorizations arise alongside 2572 instances of incorrect positive predictions. Its extreme bias toward positive sentiment prediction indicates this model provides unreliable results compared to the other LSTM, Naive Bayes and Random Forest models in balanced sentiment analysis tasks.

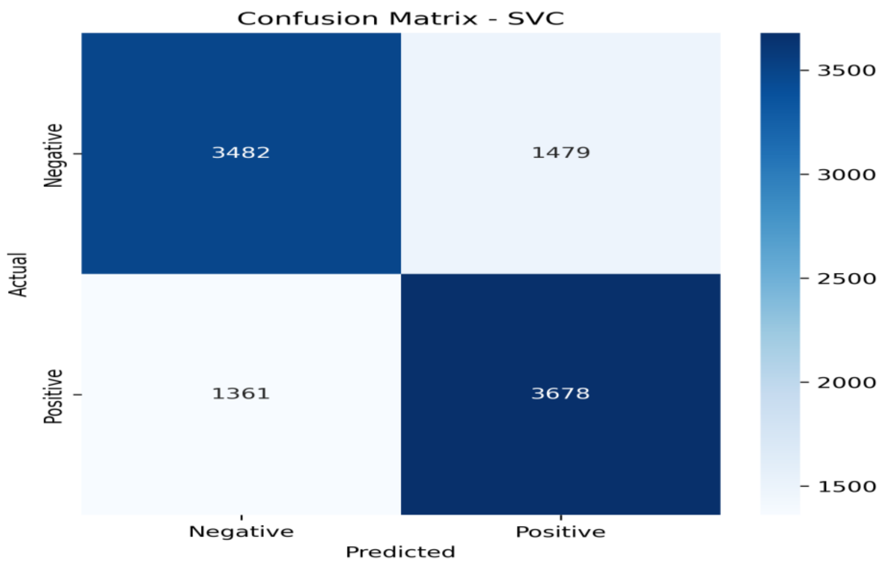

Figure 5: Confusion Matrix for SVC

An evaluation of SVC reveals overall decent accuracy along with 3482 correct negative predictions and 3678 accurate positive classifications however both classes show noticeable misclassification errors. The model demonstrates below average accuracy rates because it produces 1479 false positives coupled with 1361 false negatives in its classifications.

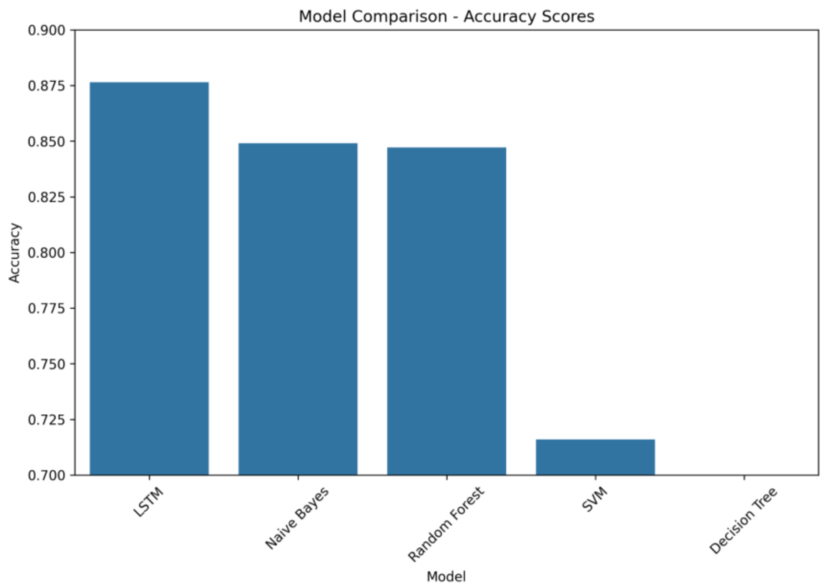

Figure 6: Model Comparison by Accuracy Scores

The accuracy evaluation proves that LSTM delivers better results than competing models by achieving 87.5% accuracy thus demonstrating its excellence for sentiment analysis. The traditional models Naive Bayes alongside Random Forest achieve almost equivalent results at approximately 85% accuracy even though their algorithms remain basic. The results show that SVM obtains 72% accuracy along with Decision Tree reaching below 70% which supports the results in the confusion matrices regarding their struggles to correctly classify unbalanced sentiments.

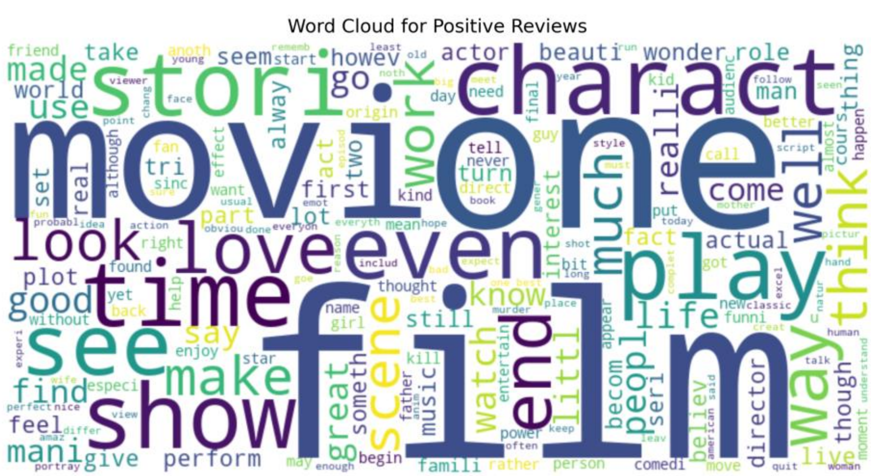

Figure 7: Word Cloud - Positive

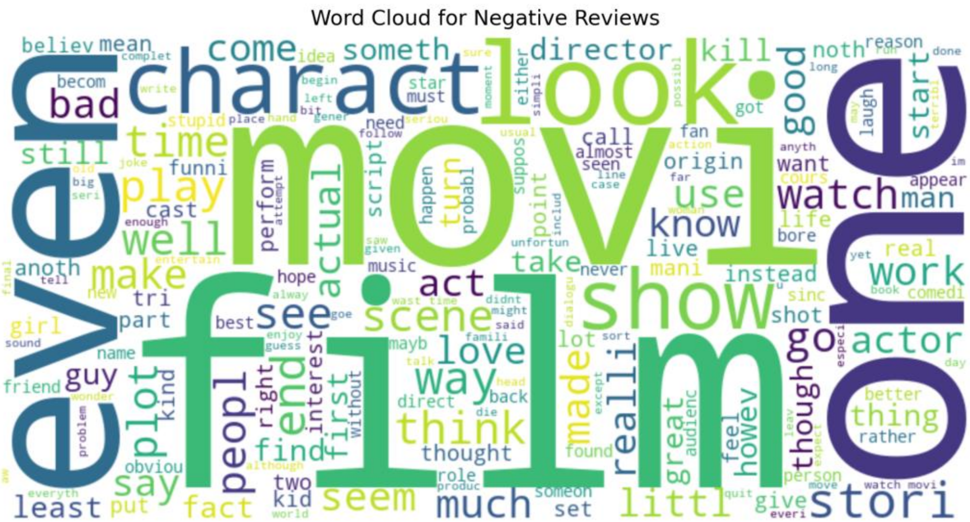

Figure 8: Word Cloud - Negative

5. Related Work:

1. Pang et al. (2002) – Early work on sentiment classification using Naive Bayes and SVM.
2. Kim (2014) – Demonstrated CNNs for sentence classification.
3. Wang & Manning (2012) – Compared Naive Bayes, SVM, and MaxEnt on sentiment tasks.
4. Hochreiter & Schmidhuber (1997) – Introduced LSTM networks.
5. Mikolov et al. (2013) – Proposed word embeddings (Word2Vec).

6. Conclusion:

Our experiments indicate that LSTMs perform best (87.65% accuracy) in IMDB sentiment analysis, then Naive Bayes and Random Forest. SVM and Decision Trees performed very badly, probably because they cannot process sparse text data efficiently.

Limitations & Future Work:

- Try BERT or Transformer-based models for better accuracy.
- Fine-tune hyperparameters for enhanced SVM performance.
- Investigate ensemble methods of combining LSTM and conventional classifiers.

7. References:

1. Pang, B., Lee, L., & Vaithyanathan, S. (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques."

2. Kim, Y. (2014). "Convolutional Neural Networks for Sentence Classification."

3. Wang, S., & Manning, C. D. (2012). "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification."

4. Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory."

5. Mikolov, T., et al. (2013). "Efficient Estimation of Word Representations in Vector Space."