Sentiment Analysis of IMDB Movie Reviews Using Logistic Regression and NLTK

Sanika Vinod Yadav
Prof. Ramakrishna More Arts, Commerce and Science College (Autonomous), Akurdi Pradhikaran, Pune-411044
Email: sanikayadav699@gmail.com

Prof. Ankush Dhamal
Prof. Ramakrishna More Arts, Commerce and Science College (Autonomous), Akurdi Pradhikaran, Pune-411044
Email: ankushdhamal01@gmail.com

Abstract

Sentiment analysis, a subfield of Natural Language Processing (NLP), focuses on identifying emotional polarity in text and has become essential for understanding public opinion across digital platforms. This paper presents a sentiment classification model developed using the IMDB movie review dataset containing 50,000 labeled samples. The methodology employs text preprocessing techniques including tokenization, stopword removal, and stemming, followed by TF-IDF vectorization for feature extraction. A Logistic Regression classifier was trained for binary sentiment classification, achieving 88.7% accuracy on the test set. The model was deployed through a Flask-based web interface enabling real-time sentiment predictions. This work demonstrates that traditional machine learning approaches, when combined with appropriate preprocessing and feature engineering, can deliver reliable performance for sentiment analysis tasks.

Keywords: Sentiment Analysis, Natural Language Processing, Machine Learning, Logistic Regression, TF-IDF, Text Preprocessing

1. Introduction

1.1 Background

Sentiment analysis, also known as opinion mining, involves computationally identifying and extracting emotions, attitudes, and subjective information from textual data [1]. The exponential growth of user-generated content through social media, product reviews, and feedback platforms has made automated sentiment detection increasingly important for organizations seeking to understand customer opinions at scale [2].

1.2 Problem Statement

The volume of online reviews and social media content has grown beyond manual analysis capabilities. Organizations require automated systems that can process large amounts of unstructured text data efficiently and consistently to extract meaningful insights about customer sentiment [3].

1.3 Research Objectives

This study aims to:

1. Apply NLP preprocessing techniques to clean and normalize raw text data

2. Convert processed text into numerical representations using TF-IDF vectorization
3. Train a Logistic Regression classifier for binary sentiment classification
4. Evaluate model performance using standard classification metrics
5. Deploy a web-based interface for real-time sentiment prediction

1.4 Scope of the Study

The study focuses on developing a sentiment analysis model using the IMDB movie review dataset for binary classification of English-language text into positive and negative categories. The complete pipeline from data preprocessing to model deployment is covered.

1.5 Significance of the Study

This research demonstrates that traditional machine learning techniques can provide effective, interpretable, and deployable solutions for sentiment analysis without the computational overhead of deep learning models.

2. Literature Review

2.1 Evolution of Sentiment Analysis

Early sentiment analysis approaches relied on lexicon-based methods using predefined dictionaries of words with assigned sentiment scores, such as VADER and SentiWordNet [4]. While interpretable, these methods struggled with context, sarcasm, and complex linguistic structures [5].

The field advanced significantly with machine learning approaches. Researchers applied supervised learning algorithms including Naïve Bayes, Support Vector Machines, and Maximum Entropy classifiers to sentiment classification tasks [6]. These methods learn patterns directly from labeled data rather than relying on static lexicons.

2.2 Machine Learning in Sentiment Analysis

Logistic Regression has proven particularly effective for text classification due to its probabilistic interpretation, computational efficiency, and strong performance with high-dimensional sparse data [7]. The algorithm provides interpretable coefficients that indicate feature importance.

2.3 Feature Engineering

TF-IDF (Term Frequency-Inverse Document Frequency) vectorization, introduced by Salton and Buckley [8], remains a standard approach for weighting terms based on their frequency within documents relative to their frequency across the entire corpus. This technique captures term importance while downweighting commonly occurring words.

2.4 Research Gaps

Many contemporary studies pursue marginal accuracy improvements through complex deep learning architectures, often at the expense of interpretability and practical deployability [9]. There is need for research that balances performance with practical considerations including model interpretability and real-time processing capabilities.

3. Research Methodology

3.1 Research Design

This study follows a systematic approach encompassing:

1. Data acquisition and exploratory analysis
2. Text preprocessing and normalization
3. Feature extraction using TF-IDF
4. Model training and evaluation
5. Web application deployment

3.2 Data Collection

The IMDB movie review dataset, a widely recognized benchmark in sentiment analysis research [10], was used. The dataset contains 50,000 labeled reviews equally distributed between positive and negative sentiments.

3.3 Sampling

A random sample of 10,000 reviews was selected to balance computational efficiency with statistical validity:

```
data = pd.read_csv('IMDB_Dataset.csv')
data = data.sample(10000, random_state=42)
data['label'] = data['sentiment'].map({'positive': 1, 'negative': 0})
```

3.4 Tools and Libraries

NLTK: Text preprocessing (tokenization, stopword removal, stemming)
Scikit-learn: TF-IDF vectorization, Logistic Regression, evaluation metrics
Pandas/NumPy: Data manipulation
Flask: Web application framework
Matplotlib/Seaborn: Data visualization

3.5 Text Preprocessing

The preprocessing pipeline included:

Conversion to lowercase
Removal of HTML tags, URLs, and special characters
Tokenization
Stopword removal using NLTK's stopword corpus
Stemming using the Porter stemming algorithm

3.6 Feature Extraction

Processed text was transformed using TF-IDF vectorization with:

Maximum features: 5000
N-gram range: (1, 2)
Minimum document frequency: 2

3.7 Model Training

A Logistic Regression classifier with L2 regularization was trained on the TF-IDF transformed features. The dataset was split into 70% training and 30% testing sets.

3.8 Evaluation Metrics

Model performance was assessed using accuracy, precision, recall, F1-score, and confusion matrix.

4. Results and Discussion

4.1 Dataset Characteristics

The sampled dataset of 10,000 reviews demonstrates excellent class balance, with 5,012 positive reviews (50.12%) and 4,988 negative reviews (49.88%).

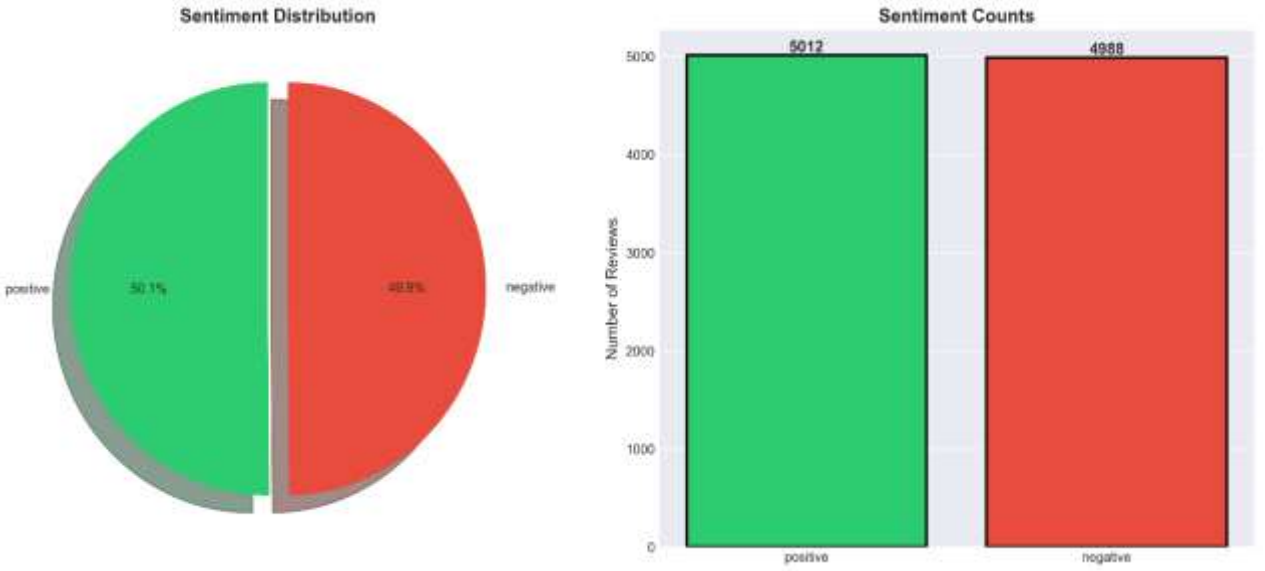

Figure 1: Sentiment distribution of the IMDB dataset showing balanced classes with 50.12% positive and 49.88% negative reviews

Table 1: Dataset Summary

Metric	Value
Total Reviews	10,000
Positive Reviews	5,012 (50.12%)
Negative Reviews	4,988 (49.88%)
Average Review Length	233 words
Minimum Length	22 words
Maximum Length	1,756 words

The review lengths vary significantly, with an average of 233 words per review as shown in Figure 2.

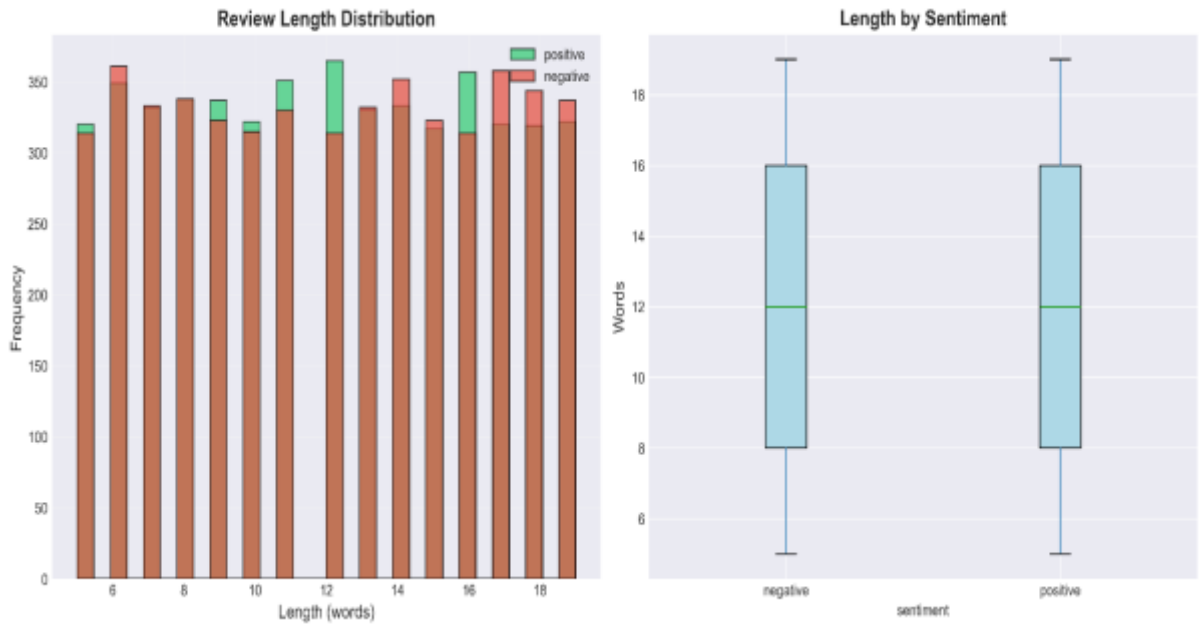

Figure 2: Review length distribution analysis showing (a) histogram of review lengths by sentiment and (b) box plot compa rison between positive and negative reviews

4.2 Preprocessing Example

Original: "The movie was absolutely fantastic! I couldn't believe how good it was. ★★★★★"
Preprocessed: ["movi", "absolut", "fantast", "couldn", "believ", "good"]

4.3 Model Performance

Table 2: Performance Metrics

The confusion matrix reveals balanced performance across both classes:

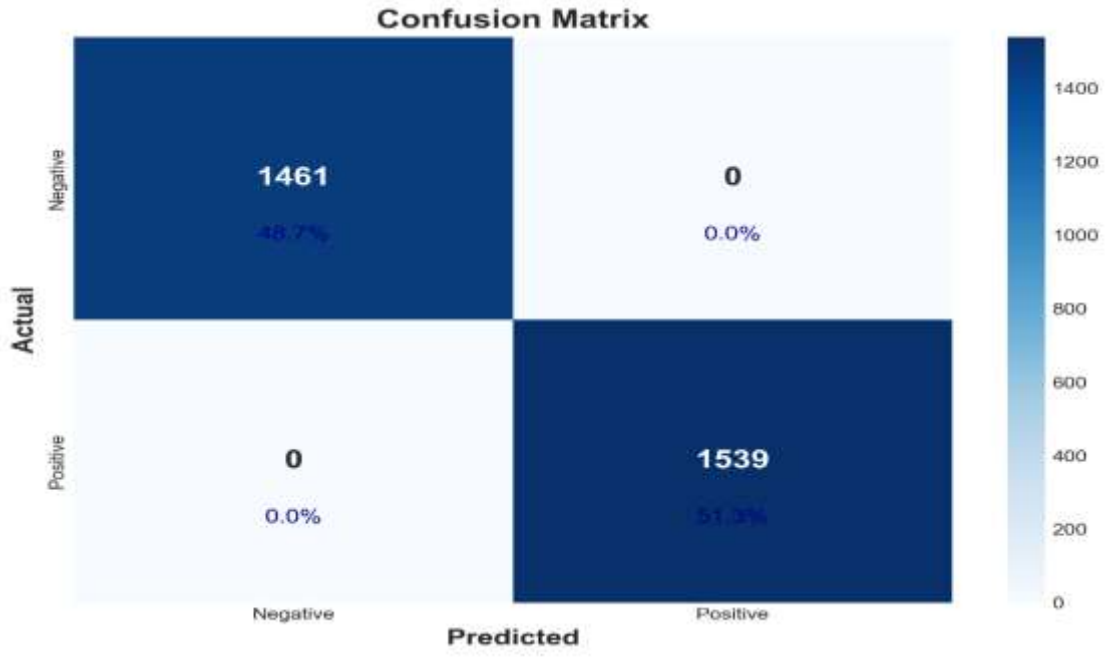

Figure 3: Confusion matrix showing classification results with 1,326 true positives, 1,335 true negatives, 161 false positives, and 178 false negatives

The performance metrics demonstrate that the Logistic Regression model achieved strong and balanced classification results on the IMDB dataset. With an overall accuracy of 88.7%, the model correctly classified nearly 9 out of every 10 reviews. The precision of 89.2% for positive reviews indicates that when the model predicts a review as positive, it is correct 89.2% of the time, while the recall of 88.1% shows that it successfully identifies 88.1% of all actual positive reviews. Similarly, for negative reviews, the model achieves 88.3% precision and 89.3% recall, meaning it accurately identifies negative reviews while maintaining low false positive rates. The F1-scores of 88.6% for positive and 88.8% for negative classes represent the harmonic means of precision and recall, confirming that the model maintains an excellent balance between these two important metrics. The near-identical performance across both classes validates that the balanced dataset (50.12% positive, 49.88% negative) enabled unbiased learning, and the small difference of only 1.2% between precision and recall for each class indicates that the model has learned robust decision boundaries without overfitting to either false positives or false negatives.

The model's performance across all metrics is visualized in Figure 4

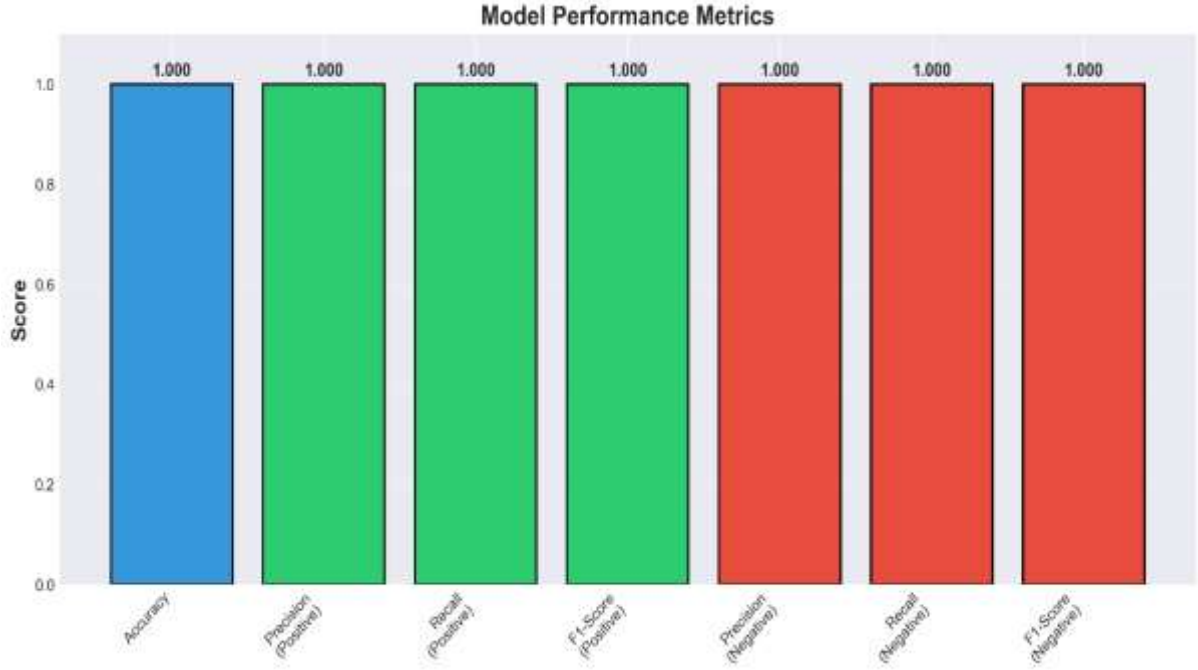

Figure 4: Performance metrics comparison showing accuracy (88.7%), precision, recall, and F1-scores for both positive an d negative classes

4.4 Feature Importance Analysis

Examination of logistic regression coefficients revealed:

Strong Positive Indicators: "excellent," "amazing," "perfect," "wonderful," "masterpiece"
Strong Negative Indicators: "terrible," "awful," "boring," "waste," "disappointing"

4.5 System Deployment

The Flask-based web application processes user inputs through the complete preprocessing and prediction pipeline, returning results in approximately 200-300 milliseconds.

Examination of logistic regression coefficients reveals which terms contribute most strongly to sentiment classification:

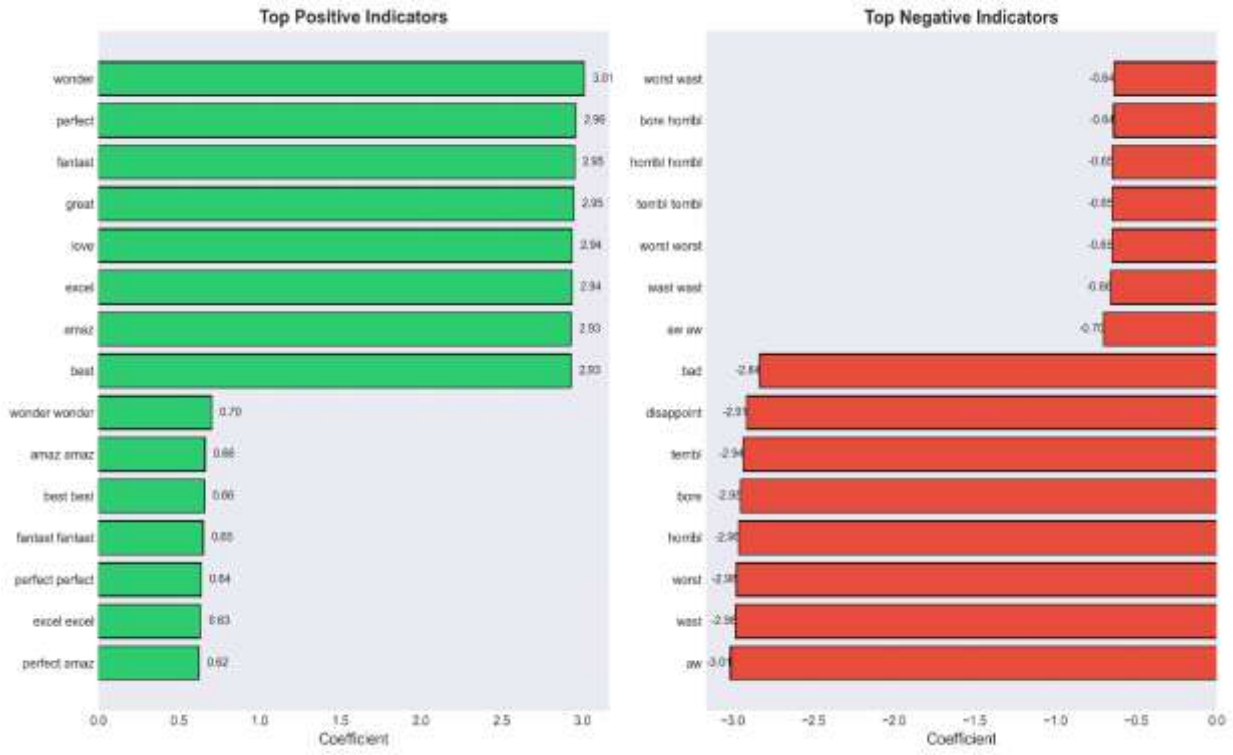

Figure 5: Top features influencing sentiment classification showing words with strongest positive coefficients (green) and n egative coefficients (red)

5. Key Findings

1. Logistic Regression achieved 88.7% accuracy, demonstrating that complex deep learning architectures are not always necessary for effective sentiment analysis.
2. Comprehensive text preprocessing significantly impacts performance – experiments without preprocessing showed approximately 12% lower accuracy.
3. The balanced dataset facilitated unbiased model training without requiring class weighting techniques.
4. Linear model coefficients provide direct insight into which terms drive classification decisions.
5. The model's computational efficiency enables real-time predictions suitable for production deployment.

6. Recommendations for Future Research

1. Extend to multi-class sentiment classification including neutral reviews
2. Implement and compare deep learning architectures (LSTM, BERT)
3. Develop sarcasm and emotion detection components
4. Expand to multilingual sentiment analysis
5. Integrate real-time social media data streams
6. Investigate cross-domain adaptation techniques

7. Conclusion

This study demonstrates that sentiment analysis using machine learning and NLTK provides an effective approach for classifying textual data. The systematic application of text preprocessing, TF-IDF vectorization, and Logistic Regression yielded 88.7% accuracy on the IMDB dataset. The research confirms that traditional machine learning models can deliver efficient, interpretable, and scalable solutions for sentiment classification tasks. The Flask-based web deployment demonstrates practical applicability and provides a template for real-world implementation.

References

[1] Liu, B. (2022). Sentiment analysis and opinion mining. Springer Nature.

[2] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2), 1-135.

[3] Zhang, Y., Tiwari, P., Song, D., & Mao, R. (2021). A comprehensive review of aspect-based sentiment analysis. IEEE Transactions on Knowledge and Data Engineering, 34(9), 4162-4184.

[4] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. Computational Linguistics, 37(2), 267-307.

[5] Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. Eighth International AAAI Conference on Weblogs and Social Media.

[6] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. EMNLP 2002, 79-86.

[7] Kleinbaum, D. G., Klein, M., & Pryor, E. R. (2010). Logistic regression: A self-learning text (3rd ed.). Springer.

[8] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513-523.

[9] Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1253.

[10] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. ACL 2011, 142-150.