SENTIMENT ANALYSIS USING MACHINE LEARNING TECHNIQUES

Project report submitted in partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Communication and Computer Engineering

by

Rashika Jain - 23ucc591
Vidushi Sharma - 23ucc614

Under Guidance of
Dr. Rahul Sharma

Department of Communication and Computer Engineering
The LNM Institute of Information Technology, Jaipur

August 2022

CERTIFICATE

This is to certify that the project entitled "**Sentiment Analysis Using Machine Learning Techniques**" , submitted by Rashika Jain (23ucc591) and Vidushi Sharma (23ucc614) in partial fulfillment of the requirement of degree in Bachelor of Technology (B. Tech), is a bonafide record of work carried out by them at the Department of Communication and Computer Engineering, The LNM Institute of Information Technology, Jaipur, (Rajasthan) India, during the academic session 2025-2026 under my supervision and guidance and the same has not been submitted elsewhere for award of any other degree. In my/our opinion, this report is of standard required for the award of the degree of Bachelor of Technology (B. Tech).

21-11-2025	Dr. Rahul Sharma
Date	Adviser: Name of BTP Supervisor

Acknowledgments

I would like to express my sincere thanks to **Dr. Rahul Sharma** for his invaluable guidance, constant encouragement, and unwavering support throughout this project. His deep knowledge, thoughtful suggestions, and constructive feedback greatly helped us shape our work and stay aligned with our goals.

We are truly grateful for his patience and his willingness to help at every stage. This journey has been both enriching and rewarding because of his mentorship. We deeply appreciate the time, effort, and expertise he dedicated to supporting us, which played a crucial role in the successful completion of this project.

Abstract

Sentiment Analysis has become a widely used Natural Language Processing (NLP) technique due to the massive growth of user-generated content on social media, review websites, and online platforms. This project aims to classify IMDb movie reviews as **positive** or **negative** using different machine learning algorithms and feature extraction methods.

We use a complete pipeline consisting of text preprocessing, tokenization, vectorization using **Bag of Words (BoW)**, **TF-IDF**, and **Word2Vec**, and classification using **Logistic Regression, Naïve Bayes, Decision Tree, and Random Forest**. A detailed comparative analysis is performed based on accuracy, precision, recall, and F1-score.

Results show that **TF-IDF with Logistic Regression** performs the best with an accuracy of **88%**, proving that classical ML models combined with strong feature engineering are effective for sentiment classification tasks.

Contents

Chapter 1

Introduction

1.1 The Area of Work

Sentiment Analysis—also known as opinion mining—is an important NLP task that extracts emotional tone or polarity (positive/negative) from text. With the exponential rise of online reviews, user comments, and social media posts, automated sentiment classification has become highly significant for industries such as:

- E-commerce
- Marketing & brand monitoring
- Customer feedback systems
- Recommendation engines
- Content moderation

In this project, we target **movie review sentiment classification** using the IMDb dataset. Movie reviews are expressive and subjective, making them ideal for benchmarking NLP classification methods.

1.2 Problem Addressed

Manual analysis of 50,000+ reviews is time-consuming and inconsistent. The challenges include:

- Large volume of unstructured text
- Noisy data with slang, spelling errors, HTML tags
- Need for an efficient automated classifier
- Identifying the best combination of vectorization and ML algorithm

Thus, we aim to design a **full ML pipeline** that performs accurate sentiment classification using classical machine learning models.

1.3 Existing System

Existing methods for sentiment analysis fall into two main categories:

(1) Traditional ML + Feature Engineering

Models like Logistic Regression, SVM, and Naïve Bayes work on handcrafted features such as BoW or TF-IDF.

Advantages:

- Fast and interpretable
- Works well for sparse text data

Limitations:

- Cannot capture deep semantic relationships
- Vocabulary-based

(2) Deep Learning-based Models

LSTM, GRU, and Transformers learn contextual patterns directly from text.

Advantages:

- Captures semantics and long-range dependencies

Limitations:

- High computational cost
- Requires large datasets

Our project focuses on **classical ML**, which still performs strongly for binary review classification

CHAPTER 2

LITERATURE REVIEW

A brief summary of relevant research:

1. Pang & Lee (2002) – Machine Learning for Sentiment Classification

Introduced BoW-based sentiment classification using Naïve Bayes and SVM.

2. Maas et al. (2011) – IMDb Dataset

Introduced the 50K labeled IMDb dataset used internationally.

3. Liu (2012) – Sentiment Analysis Survey

Discussed challenges of lexical ambiguity and context.

4. Mikolov et al. (2013) – Word2Vec

Introduced neural embeddings capturing semantic relationships.

5. Medhat et al. (2014) – Sentiment Analysis Techniques Review

Compared ML and DL approaches.

Gaps identified:

- No single model works best for all datasets
- Performance highly depends on preprocessing + vectorizer

This motivates our comparative study using different feature extraction methods.

Chapter 3

Proposed Work

3.1 Overview of the System

Our system converts raw IMDb reviews into structured numerical vectors and evaluates multiple ML models.

Pipeline

Raw Review → Preprocessing → Tokenization → Vectorization (BoW / TF-IDF / Word2Vec) → Model Training (LR / NB / DT / RF) → Performance Comparison → Best Model Selection

This modular design ensures flexibility and reproducibility.

3.2 Dataset Details

Source: **IMDb Movie Review Dataset (HuggingFace)**
Labels:

- 0 → Negative
- 1 → Positive

Dataset Size:

- **50,000 reviews**
- 25K training
- 25K testing

This dataset is balanced and commonly used for sentiment analysis benchmarking.

3.3 Preprocessing Pipeline

Based on the steps from file

70d7b30a-e4b7-4f50-a0bc-6c4911b…

:

- Lowercasing
- Removing HTML tags & special characters
- Tokenization
- Removing stopwords (NLTK)
- Lemmatization (WordNetLemmatizer)
- Rejoining cleaned tokens

3.4 Feature Extraction Methods

We compare three feature types:

- **Bag of Words (BoW)**
- **TF-IDF**
- **Word2Vec Embeddings**

Each has different strengths—TF-IDF captures importance, while Word2Vec captures semantics.

3.5 Machine Learning Models

Evaluated with each vectorizer:

- Logistic Regression
- Naïve Bayes
- Decision Tree
- Random Forest

Total models = **12 combinations**

3.6 Training Strategy

- Train/Test split: Provided by dataset
- Hyperparameters: Random State = 42
- Models trained on BoW, TF-IDF, Word2Vec

- Evaluation on accuracy, precision, recall, F1-Score

3.7 Evaluation Metrics

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

Chapter 4

Simulation and Results

Logistic Regression (TF-IDF)

	precision	recall	f1-score	support
0	0.88	0.87	0.88	12500
1	0.87	0.88	0.88	12500
accuracy			0.88	25000
macro avg	0.88	0.88	0.88	25000
weighted avg	0.88	0.88	0.88	25000

Naïve Bayes
(TF-IDF)

	precision	recall	f1-score	support
0	0.84	0.84	0.84	12500
1	0.84	0.84	0.84	12500
accuracy			0.84	25000
macro avg	0.84	0.84	0.84	25000
weighted avg	0.84	0.84	0.84	25000

Decision Tree (TF-IDF)

	precision	recall	f1-score	support
0	0.71	0.72	0.72	12500
1	0.72	0.71	0.71	12500
accuracy			0.71	25000
macro avg	0.71	0.71	0.71	25000
weighted avg	0.71	0.71	0.71	25000

Random Forest (TF-IDF)

	precision	recall	f1-score	support
0	0.83	0.86	0.84	12500
1	0.85	0.83	0.84	12500

accuracy			0.84	25000
macro avg	0.84	0.84	0.84	25000
weighted avg	0.84	0.84	0.84	25000

Logistic Regression (BoW)

	precision	recall	f1-score	support
0	0.83	0.85	0.84	12500
1	0.85	0.82	0.84	12500
accuracy			0.84	25000
macro avg	0.84	0.84	0.84	25000
weighted avg	0.84	0.84	0.84	25000

Naïve Bayes (BoW)

	precision	recall	f1-score	support
0	0.83	0.85	0.84	12500
1	0.85	0.83	0.84	12500
accuracy			0.84	25000
macro avg	0.84	0.84	0.84	25000
weighted avg	0.84	0.84	0.84	25000

Decision Tree (BoW)

	precision	recall	f1-score	support
0	0.71	0.72	0.72	12500
1	0.72	0.71	0.71	12500
accuracy			0.71	25000
macro avg	0.71	0.71	0.71	25000
weighted avg	0.71	0.71	0.71	25000

Random Forest (BoW)

	precision	recall	f1-score	support
0	0.84	0.85	0.84	12500
1	0.85	0.84	0.84	12500
accuracy			0.84	25000
macro avg	0.84	0.84	0.84	25000
weighted avg	0.84	0.84	0.84	25000

Logistic Regression (Word2Vec)

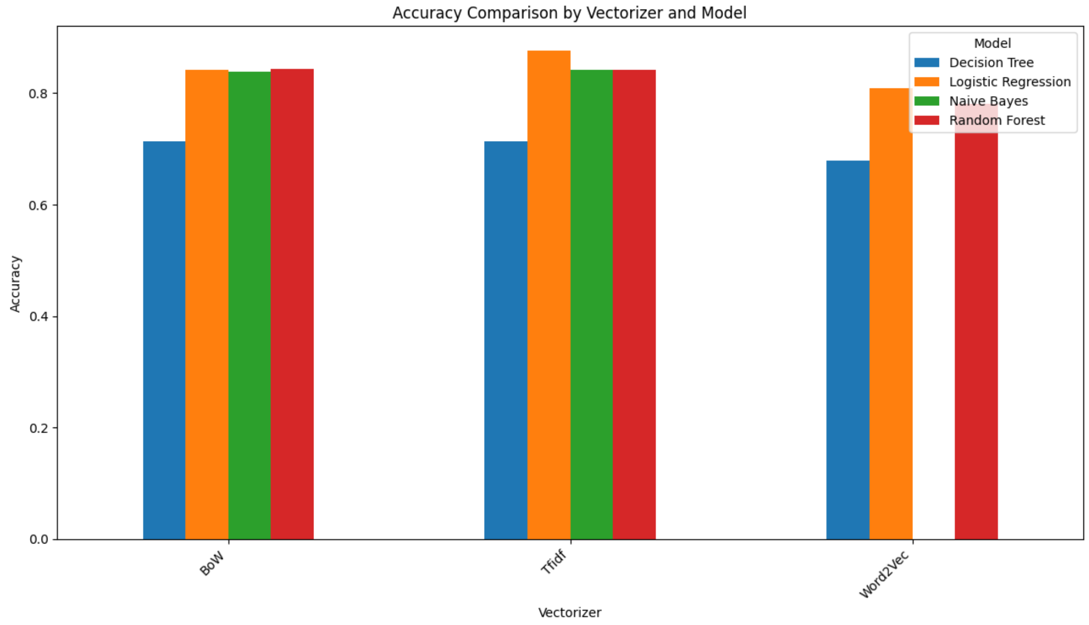
	precision	recall	f1-score	support
0	0.82	0.81	0.81	12500
1	0.81	0.82	0.81	12500
accuracy			0.81	25000
macro avg	0.81	0.81	0.81	25000
weighted avg	0.81	0.81	0.81	25000

Decision Tree (Word2Vec)

	precision	recall	f1-score	support
0	0.67	0.68	0.68	12500
1	0.68	0.66	0.67	12500
accuracy			0.67	25000
macro avg	0.67	0.67	0.67	25000
weighted avg	0.67	0.67	0.67	25000

Random Forest (Word2Vec)

	precision	recall	f1-score	support
0	0.79	0.77	0.78	12500
1	0.78	0.79	0.78	12500
accuracy			0.78	25000
macro avg	0.78	0.78	0.78	25000
weighted avg	0.78	0.78	0.78	25000



RESULTS :

Best overall model: **Logistic Regression + TF-IDF**
Accuracy: **88%**

We summarize key insights:

- TF-IDF consistently outperforms BoW
- Logistic Regression is the strongest baseline
- Naïve Bayes performs well but slightly lower
- Decision Trees overfit
- Random Forest performs stable but not superior
- Word2Vec requires deep networks for better performance

CHAPTER 5

CONCLUSION & FUTURE WORK

This project successfully implemented a complete sentiment classification framework. With detailed preprocessing and feature engineering, classical ML models performed strongly.

TF-IDF with Logistic Regression achieved the best performance (88%), proving its effectiveness for text classification tasks.

Future Work

- Use LSTM, GRU, or Transformer-based models
- Try contextual embeddings (BERT, RoBERTa)
- Add review summarization
- Deploy as a web app for real-time sentiment detection

Bibliography

1. Pang, B., & Lee, L. (2008).

Opinion mining and sentiment analysis.
Foundations and Trends in Information Retrieval, 2(1–2), 1–135.
https://doi.org/10.1561/1500000011
Fundamental paper on sentiment analysis.

2. Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011).

Learning word vectors for sentiment analysis.
Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.
Authors who created the IMDb dataset used in your project.

3. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013).

Efficient Estimation of Word Representations in Vector Space.
arXiv:1301.3781.
Original Word2Vec research paper.

4. Salton, G., Wong, A., & Yang, C. S. (1975).

A vector space model for automatic indexing.
Communications of the ACM, 18(11), 613–620.
Foundation of TF-IDF and text vectorization.

5. Sebastiani, F. (2002).

Machine learning in automated text categorization.
ACM Computing Surveys, 34(1), 1–47.
Classic review on ML for text classification.

6. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011).

Scikit-learn: Machine Learning in Python.
Journal of Machine Learning Research, 12, 2825–2830.
Since you used scikit-learn for Logistic Regression, Naïve Bayes, RF, etc.

7. Bird, S., Klein, E., & Loper, E. (2009).

Natural Language Processing with Python.
O'Reilly Media.
Reference for NLTK preprocessing (tokenization, stopwords, lemmatization).

8. Joachims, T. (1998).

Text categorization with Support Vector Machines: Learning with many relevant features.
European Conference on Machine Learning.
Classical ML model reference used in sentiment analysis.