Sentiment Analysis of IMDB Reviews using SVM & Naïve Bayes

Based on ICRTC-2015 Research Paper

By:

Name: Pragya Tripathi

Enrolment No: 00596303122

Department: Information Technology (IT-E)

TABLE OF CONTENT

Problem Statement

- With the rapid growth of user-generated content on the internet, especially reviews on platforms like IMDB, there is a need to automatically identify whether a review expresses a positive or negative sentiment.
- Manually analyzing such massive data is impractical, and inaccurate classification can affect business decisions and customer perception.
- This project aims to evaluate and compare the performance of Naive Bayes and Support Vector Machine (SVM) algorithms for classifying movie reviews into positive or negative sentiments.

Project Objective

- To implement machine learning techniques for classifying movie reviews into **positive** or **negative** categories.
- To evaluate and compare the effectiveness of **Support Vector Machine (SVM)** and **Naïve Bayes (NB)** classifiers.
- To replicate and analyze the methodology of the ICRTC-2015 paper titled *"Classification of Sentimental Reviews Using Machine Learning Techniques"*.
- To determine the best-performing model based on evaluation metrics such as **accuracy**, **precision**, **recall**, and **F1-score**.

Proposed Methodology

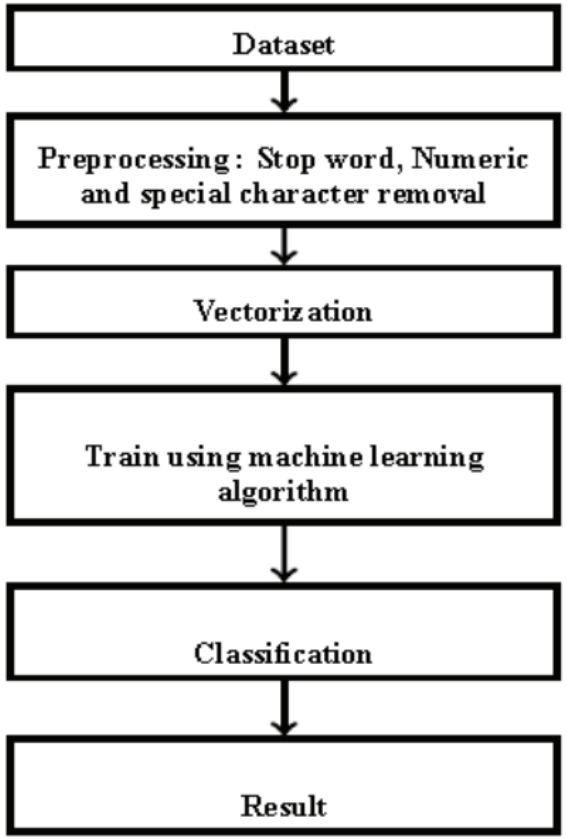

Figure 1: Sentiment Classification Workflow

Source: Tripathy, A., Agrawal, A., & Rath, S. K. (2015). Classification of Sentimental Reviews Using Machine Learning Techniques.

Dataset Description

Name: IMDB Movie Review Dataset

Source: Kaggle / Stanford Large Movie Review Dataset

Total Samples: 50,000 reviews

- 25,000 labeled **positive**
- 25,000 labeled **negative**

Type: Binary classification dataset (balanced)

Data Preprocessing

1. Text Cleaning:

- Convert to lowercase
- Remove punctuation, special characters, and whitespace
- Remove stopwords (e.g., "the", "and"), html tags, urls

2. Tokenization: Split text into individual words.

3. Stemming: Reduce words to their root form (e.g., “running” → “run”) using a stemmer like Porter

4. Feature Extraction:

- **Count Vectorizer:** Converts text to a sparse matrix based on word frequency.
- **TF-IDF Vectorization:** Balances word frequency with importance across documents.

Model Building

Two different machine learning algorithms implemented are as follows:

Naive Bayes

- A probabilistic classifier based on Bayes' Theorem. Assumes independence among features and works well with textual data. It's efficient and requires less training data.

- Naive Bayes uses probabilities to guess the most likely class (like positive/negative). It looks at each word in the text and calculates how likely that word is to appear in each class. Then it combines those probabilities and picks the class with the highest overall score.

- For a given textual review 'd' and for a class 'c' (positive, negative), the conditional probability for each class given a review is P(c|d) . According to Bayes theorem this quantity can be computed using the following equation:

$$P(c \mid d) = \frac{P(d \mid c) * P(c)}{P(d)}$$

Model Building

Support Vector Machine

- A non-probabilistic binary classifier that finds the optimal hyperplane to separate data points. It's powerful for high-dimensional data like text.

Let $c_j \epsilon \{1,-1\}$ be the class (positive , negative) for a document d_j, the equation for $\vec{w}$ is given by

$$\vec{w} = \sum_j \alpha_j c_j \vec{d}_j, \alpha_j \geq 0$$

Evaluation Metrics

Used train_test_split to split the dataset into training and testing sets (70-30 split) and evaluated the models on accuracy, precision, recall, and F1-score.

- **Accuracy:** Correct predictions / Total predictions.
- **Precision:** TP / (TP + FP) – Correct positive predictions.
- **Recall:** TP / (TP + FN) – Ability to find all positive instances.
- **F1 Score:** Harmonic mean of Precision and Recall.
- **Confusion Matrix:** Visualizes TP, FP, TN, FN for deeper insight.

Result and Findings

Section 1: Summary of Evaluation Metrics

Naïve Bayes

- Accuracy: 86%
- F1 Score: 86%

SVM

- Accuracy: 89%
- F1 Score: 89%

Both models performed well on binary classification task.

SVM slightly outperformed Naïve Bayes in all key metrics.

	Precision	Recall	F1-Score
Positive	0.86	0.87	0.86
Negative	0.87	0.85	0.86
Accuracy	0.86		

Figure 2: Evaluation Metrics Result after Implementation for Naïve Bayes

The performance evaluation parameters obtained for Naive Bayes classifier is shown in table 4.

Table 4: Evaluation parameters for Naive Bayes classifier

	Precision	Recall	F-Measure
Negative	0.80	0.89	0.84
Positive	0.87	0.77	0.82

Maximum accuracy achieved after the cross validation analysis of Naive Bayes classifier is **0.8953**.

Figure 3: Evaluation Metrics Result in Research Paper for Naïve Bayes

	Precision	Recall	F1-Score
Positive	0.9	0.88	0.89
Negative	0.89	0.9	0.89
Accuracy	0.89		

Figure 4: Evaluation Metrics Result after Implementation for SVM

Table 6: Evaluation parameters for Support Vector Machine classifier

	Precision	Recall	F-Measure
Negative	0.87	0.89	0.88
Positive	0.89	0.86	0.88

Maximum accuracy achieved after the cross validation analysis of Support Vector Machine classifier is **0.9406**.

Figure 5: Evaluation Metrics Result in Research Paper for SVM

Section 2: Confusion Matrix Visual

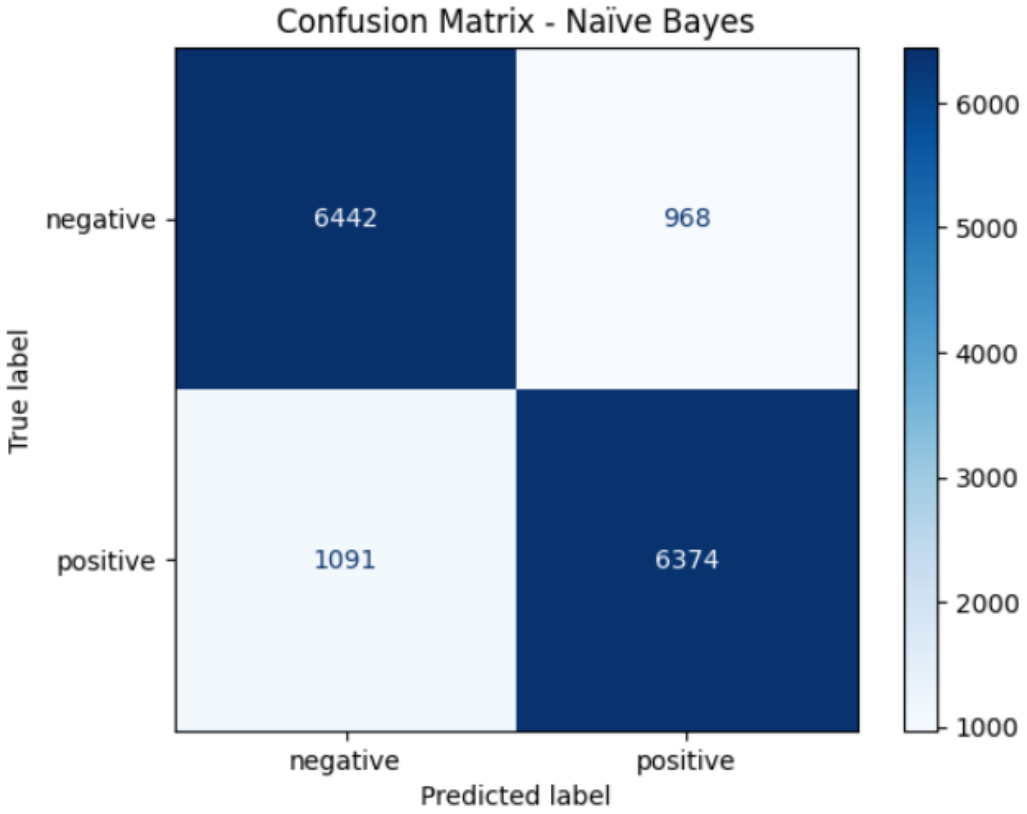

Figure 6: Confusion Matrix for NB

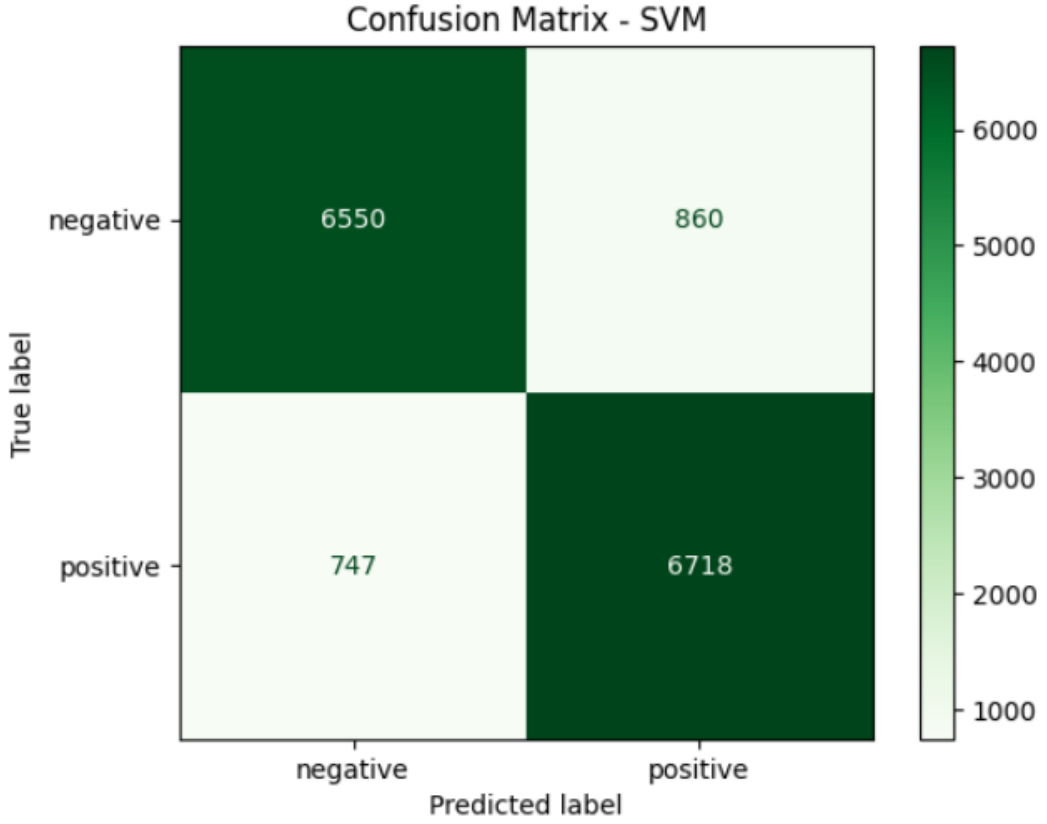

Figure 7: Confusion Matrix for SVM

Section 3: Comparative Analysis

- The results align with the research trends—SVM generally performs better than Naïve Bayes.

Algorithm	Pang & Lee (2002)	Read (2005)	Research Paper (2015)	My Results
Naïve Bayes	86.40%	78.90%	89.50%	86%
SVM	86.10%	81.50%	94%	89%

Figure 8: Final Comparative Analysis Table

POTENTIAL IMPROVEMENTS

- **Model Evaluation Approach:** The accuracy could be further improved by conducting 10-fold cross-validation instead of relying on a single train-test split. This would provide a more reliable estimate of the model's performance across different data partitions.
- **Enhanced Data Preprocessing:** Further refinement of the data preprocessing steps, including advanced cleaning techniques, could potentially lead to better model performance.

REFERENCES

- A. Tripathy, A. Agrawal, and S. K. Rath, “Classification of sentimental reviews using machine learning techniques,” in *International Conference on Recent Trends in Computing (ICRTC)*, 2015, pp. 1–6.
- https://drive.google.com/file/d/1bTHWDqZ4zA7rkKHafvPFY1G7WZFvmeHz/view?usp=drive_link (Implementation Colab Notebook)
- https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/
- https://www.geeksforgeeks.org/naive-bayes-classifiers/

THANK YOU!!