

A Machine Learning-Based Framework for Sentiment Detection from Textual Data

OPEN ACCESS

Volume: 13

Special Issue: 3

Month: February

Year: 2026

P-ISSN: 2321-788X

E-ISSN: 2582-0397

Citation:

Panwar, Sudhanshu, and Pritam Singh Negi. "A Machine Learning-Based Framework for Sentiment Detection from Textual Data." *Shanlax International Journal of Arts, Science and Humanities*, vol. 13, no. 406–413, 2026.

DOI:

<https://doi.org/10.34293/sijash.v13iS3-i2-Feb.10312>

Sudhanshu Panwar

Research Scholar, Department of Computer Science & Engineering, HNB Garhwal University (A Central University), Srinagar Garhwal Uttarakhand, India

Dr. Pritam Singh Negi

Assistant Professor, Department of Computer Science & Engineering HNB Garhwal University (A Central University), Srinagar Garhwal Uttarakhand, India

Abstract

Sentiment detection from textual content plays a vital role in understanding human opinions and behavioral patterns across digital platforms. This study pre-sents a systematic approach for sentiment classification using classical machine learning techniques. The experiments are conducted on the Combined Sentiment dataset, which is publicly available on Kaggle. To improve data quality and mini-mize noise, several text pre-processing steps are applied, including lowercase conversion, punctuation removal, and stopword elimination. After pre-processing, meaningful textual features are extracted using the TF-IDF method, which effectively represents the relevance of terms within the dataset. The pro-cessed data is then partitioned into training and testing sets using an 80:20 ratio. Multiple ML classifiers, including Support Vector Machine (SVM), Logistic Re-gression (LR), Naive Bayes (NB), and Random Forest (RF), are employed to evaluate their effectiveness in emotion classification. The comparative analysis reveals that the SVM classifier achieves superior accuracy compared to other models. Appropriate classifier selection in enhancing emotion classification per-formance.

Keywords: Sentiment Detection, Feature Extraction, TF-IDF, Machine Learning, SVM, LR, NB, RF

Introduction

In recent years, sentiment detection has gained considerable importance across mul-tiple domains, including business intelligence, social media monitoring, e-commerce analytics, and public opinion assessment. Organizations relies on sentiment classifica-tion systems to understand customer feedback, improve decision making processes, and enhance user experience. Similarly, researchers utilize sentiment analysis to study behavioral trends, societal reactions, and communication patterns in digital environ-ments.

Textual data, however, presents unique challenges due to its informal structure, linguistic variability, presence of noise, and contextual ambiguity. Variations in spelling, slang usage,

punctuation, and grammatical inconsistencies can significantly affect classification performance. Therefore, systematic text preprocessing plays a crucial role in preparing raw data for computational analysis. Standardization techniques such as lowercasing, removal of punctuation and stopwords, and elimination of irrelevant symbols help reduce and improve feature consistency.

Feature representation is another essential component in sentiment detection systems. Since machine learning algorithms cannot directly interpret raw text, it must be transformed into a structured numerical format. The Term Frequency-Inverse Document Frequency (TF-IDF) technique is widely used for this purpose, as it captures the importance of words relative to their frequency across documents. By assigning higher weights to discriminative terms and lower weights to common words, TF-IDF enhances the quality of textual representation for classification tasks.

Machine learning-based approaches have demonstrated strong performances in sentiment classification problems. Classical supervised algorithms such as Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes (NB), and Random Forest (RF) are commonly employed due to their robustness, interpretability, and efficiency. These models learn patterns from labelled datasets and predict sentiment categories based on learned decision boundaries or probabilistic estimations. Comparative evaluation of multiple classifiers helps identify the most suitable model for a given dataset and application context.

In this study, a structured machine learning framework is developed for sentiment detection from textual data. The framework integrates systematic preprocessing, TF-IDF based feature extraction, and supervised classification techniques. Performance comparison among different classifiers provides insights into their effectiveness for emotion classification tasks. The proposed methodology contributes to the broader field of text analytics by presenting a clear and replicable pipeline for sentiment analysis using classical machine learning models.

The remainder of the paper is organized in a structured manner to ensure clarity and logical flow of discussion, Section 2 presents a comprehensive review of the related work, highlighting previous research contributions and existing approaches in the domain of sentiment detection. Section 3 describes the proposed methodology in detail, including data preprocessing, feature extraction, and model implementation step. Section 4 discusses the experimental results along with a thorough performance analysis and comparative evaluation of the applied classifiers. Finally, Section 5 concludes the study by summarizing the key findings and outlining potential directions for future research.

Related Work

In a past few decades ML based techniques gained researchers interest in the domain of sentiment analysis. In the research conducted by Singh et al.[1] author compared lexicon-based, machine learning, and deep learning methods for sentiment analysis on Manipuri video review comments. Their manually categorized dataset of 4000 comments, collected from social media, revealed that the NB classifier outperformed others.

Similarly, in the work of Ramadhani et al.[2], author performed the classification of sentiments on the tweets of Korean and English languages comprises 4000 tweets. The highest accuracy of 75.03% is obtained using Multilayer Perceptron (MLP) with three hidden layers along with the optimization using Stochastic Gradient Descent (SGD) on the twitter dataset. Also, the typical preprocessing techniques are utilized such as lexical segmentation, lowercasing of words, stemming, stop-word and numerical filtering, noise removal.

In terms of presenting the effectiveness of Convolutional Neural Networks (CNN) with Recurrent Neural Filters (RNF) Yang[3] proposed an effective method of sentiment classification i.e.,

substituting recurrent architectures such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) with the traditional linear CNN filters. Substituting recurrent architecture help model to precisely capture the order of words with compositional meaning and longer dependencies in textual data. This CNN with RNF model obtained the better classification with improvements on the Stanford Sentiment Treebank. Additionally, provides the better answers in selection task while being computation resources efficient.

Another sentiment analysis model is developed in the research conducted by Niu et al.[4]. Author worked on the SVM classifier to analyze the sentiments in medical domain with language model. Traditionally researchers identifying sentiments via dataset comprising personal opinions, Niu et al. extracted polarity of information from the patient record and their medical outcomes. Also, small lexicon of medical words was manually compiled and the incorporation of domain specific knowledge and linguistic features the performance of the model was enhanced.

In one of the research conducted by Prabhakar et al.[5], author presented a senti-ment analysis model to classify sentiments via implementing the bagging and boost-ing fusion in AdaBoost classifier. The dataset comprises the twitter data of US airline and preprocessed into suitable format by eliminating irrelevant information followed by computational data analysis technique. Performance of the model is evaluated in F1-score obtaining 68% with AdaBoost model.

In order to study sentiment analysis addressing violence against women Alzyout et al. [6] worked on Arabic tweet data. For the dataset several ML models are applied including SVM, kNN, Naïve Bayes (NB) and decision tree (DT). After that, typical preprocessing methods were utilized such as tokenization, stemming and stop-word removal. For feature extraction TF-IDF method was used. SVM obtained the highest accuracy of 78.25%.

In the research paper by Garg et al.[7], [8] author outlines the significance of NLP in extracting the linguistic features, context sensitive cues, and sentiment features from textual data. Author demonstrate the optimized feature selection techniques that enables the capability of NLP methods in classification tasks.

Methodology

Sentiment Analysis is a prominent application of ML in today’s digital communica-tion as it allows to understand the sentiments and emotions behind the speaker’s or communicators verbal communication. People in this digital age express their emo-tions, opinions, reviews on social media or digital platform. Sentiments in such type of platform directly impact public opinions, brand reputation, political trends and customer satisfaction. This section presents the methodology of the sentiment analy-sis model. Fig. 1 illustrates the complete workflow our sentiment analysis model. Starting with Data collection phase this methodology comprises five steps that ends with performance evaluation with typical evaluation metrics.

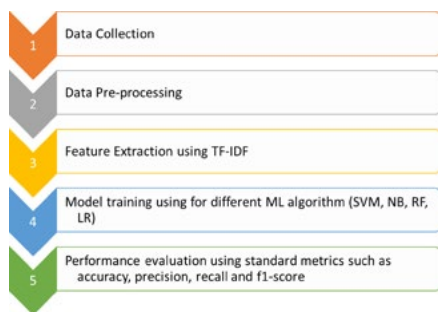


Fig. 1 Systematic Workflow of the Modal

Dataset Collection

This study utilized an open dataset of “Combined Sentiment Dataset” obtained from Kaggle platform. Dataset is composed of 3309 sentence with defined labels and categorized into two categories, positive and negative. Data is harnessed into and organized single structured form, in which sentence feature represent the textual data and sentiment feature represents the corresponding label positive or negative, that makes it suitable for binary classification. These two features served as key features in developing the model.

Data Pre-Processing

Data preprocessing is an essential step in machine learning-based sentiment analysis, where raw textual data is converted into a clean and standardized form so that the model can process it effectively. In this study, the text data were systematically cleaned to reduce noise and extract meaningful patterns. This research utilized Python-based preprocessing techniques. First, missing values were removed from the dataset, and the index was reset to ensure proper data alignment. Then, text cleaning operations were applied to each sentence. The text was converted to lowercase to prevent different cases of the same words (such as “Good” and “good”) from being treated separately and unnecessarily increasing the vocabulary size. Furthermore, URLs, numeric values, and punctuation symbols were removed from the text, as these elements do not contribute meaningfully to sentiment prediction. Additionally, extra whitespace was removed to make the text more compact and consistent. The main objective of these preprocessing steps was to make the text data noise-free and uniform.

Feature Extraction

After preprocess, the data, features are extracted from the dataset using the Term Frequency–Inverse Document Frequency (TF-IDF) vectorization technique to extract statistical features. TF-IDF is a statistical feature extraction technique that calculates the significance of a token based on its frequency in the sentence and its rarity across the entire dataset. This feature extraction approach is typically used in text mining and NLP tasks as it reduces the less impactful words and provides importance to meaningful terms[9]. In this study, TF-IDF is implemented on the “sentence” column to transform sentences into a numerical feature vector. To capture the essential textual patterns from the textual dataset without challenging the complexity of the model, a maximum of 3000 features are selected via setting the `max_features = 3000`.

By utilizing the mathematical formula in the equation 1, the TF-IDF score of a term t for a document d can be calculated:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \quad (1)$$

(TF = Term Frequency, IDF = Inverse Document Frequency).

TF refers to the occurrence of a term in a document, normalized by dividing it by the total number of terms in the document. IDF reduces the weight of words that occur very frequently in a document.

Data Splitting

Subsequent to TF-IDF based statistical feature extraction, dataset is divided into two parts; training and testing. To split the dataset study used the `train_test_split` function from the `scikit-learn` library that split the data into 80:20 ratio. 80 percent of the data is used for training the models, and the remaining 20 percent is used for testing and evaluating the models. Furthermore, the `random_state` is set to 42 that ensures the reproducibility of the experiment by maintaining a consistent result across executions. Through this process, the model was tested on unseen data, enabling a reliable evaluation of its performance.

Classification Model

Completing the feature extraction and data splitting, conventional four classification models were experiment for binary classification. Classification model for experiments are selected on the basis of their extensive utilization in the domain or senti-ment analysis on textual or text-based data.

Decision Tree (DT)

A decision tree[10] It is a simple supervised machine learning classifier that makes predictions by applying a simple set of rules in a tree-like structure. Each branch of the tree represents a condition based on a certain decision, and the leaves determine the predictions. This ML model is effective in interpreting the non-linear relationship in the data.

Random Forest (RF)

Random Forest[11] is an ensemble machine learning technique that works by integrating multiple decision trees. Each decision tree trained slightly different part of data with distinct features. Prediction of RF model is based on the final outcome of each DT based on the majority of voting to particular out-come. It makes the model's prediction more precise and stable along with the re-duced risk factors of overfitting issue.

Logistic Regression (LR)

Logistic Regression[12] is a typical classification algo-rithm generally utilized for binary classification. It calculates the probability of clas-ses by using the sigmoid mathematical function. It is a fast and baseline model for tasks such as text classification.

Support Vector Machine (SVM)

Support Vector Machine[13] is a robust super-vised machine learning algorithm that separates the samples by drawing the best possible boundary, known as a hyperplane. Hyperplane separates the samples of distinct classes in such way that the distance between classes is as largest as possible.

Performance Evaluation

Confusion matrix-based performance metrics were used to evaluate the effectiveness of the machine learning models. The confusion matrix provides a detailed overview of the model's prediction outcomes. It consists of four main components: True Positives (TP), which represent correctly predicted positive sentiments; False Positives (FP), where the model incorrectly predicted a negative sentiment as positive; True Negatives (TN), which represent correctly predicted negative sentiments; and False Negatives (FN), where a positive sentiment was incorrectly classified as negative. Based on these values, various evaluation metrics were calculated to clearly understand the model's sentiment classification capabilities.

1. **Accuracy:** Accuracy can be calculated by taking the sum of true positive classes and true negative and divide them with the total number of samples. It presents how well the model performs using simple and intuitive measures as defined in Equation 2.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / \text{Total Samples} \quad (2)$$

2. **Precision:** Precision denotes the model performance by predicting that the positive sentiments as truly positive. By using equation 3 precision can be calculated.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (3)$$

3. Recall: Recall evaluates the model’s capability to accurately identify the actual positive predicted classes/instances. It can be evaluated using the formula given in equation 4.

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (4)$$

4. F1-Score: F1-score is a harmonic mean of precision and recall which is calculated by the equation 5. F1-score defines the balanced measurement of a model’s score, ranging from 0 to 1.

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (5)$$

5. Support: Total number of instances of distinct classes in the dataset is known as support, which indicates the no of true samples belonging to the specific class.

Experimental Results & Analysis

Over 3309 samples from Combined Sentiment Dataset were experimented after split-ting the data into 80:20 ratio. Dataset comprises labelled sentences which include two sentiment categories: positive or negative. Each sentence in the dataset has a senti-ment label, so the model must classify it into one of two classes. After preprocessing steps, namely removing URLs, numbers, punctuation, and extra spaces, we applied TF-IDF vectorization for feature extraction.

A maximum of 3000 features were identified to capture the most relevant terms in the dataset. Finally, we split the data into 80% for training and 20% for testing sets to evaluate the model’s performance. Four supervised machine learning algorithms: Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes (NB), and Random Forest (RF) were trained and evaluated using the dataset. To measure their performance, we used standard evaluation metrics including accuracy, precision, recall, and F1-score, all derived from the confusion matrix.

A comparative analysis of the results shows that the Naïve classifier achieved the highest overall classification accuracy of 82% among the implemented models. Followed by SVM and LR, obtaining 81%, while RF the least accuracy of 77%. Based on the precision performance evaluation, NB achieved the highest score of 0.84, and RF still had the lowest score with 0.78. Table 1 presents the comparative results of the experiment with accuracy, precision, recall, and F1-score of classifiers, including SVM, LR, NB, and RF.

Table I. Performance measure of ML models to detect sentiments

Model	Accuracy	Precision	Recall	F1-Score
SVM	0.81	0.82	0.79	0.81
LR	0.81	0.83	0.78	0.80
NB	0.82	0.84	0.78	0.81
RF	0.77	0.78	0.76	0.77

These observations suggest that linear models, particularly NB, are highly effective when working with high-dimensional TF-IDF feature representations in binary senti-ment classification tasks. Overall, the experimental results demonstrate that careful preprocessing, combined with effective feature extraction techniques, plays a crucial role in improving sentiment prediction performance.

Fig. 2. present the comparative analysis of all four utilized classifiers, demonstrat-ing the clustered distribution of each performance metric.

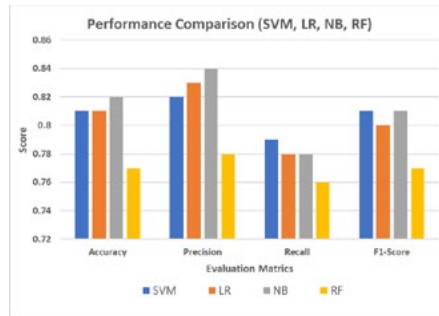


Fig. 2 Comparative Analysis of ML Models on Distinct Performance Measures

Conclusion

In this study, we developed a sentiment classification model based on the textual dataset and applied four distinct ML algorithms SVM, RF, LR and NB. For precise classification we have trained our model with 3309 labeled sentence dataset i.e., Combined Sentiment Dataset from Kaggle. TF-IDF feature extraction techniques is applied to extract linguistic based features in corpus. Among all the classifiers exper-imented, Naïve bayes obtained the highest performance with 82% accuracy followed by SVM and LR achieving the accuracy of 81%. On the other evaluation metrics like precision and F1-score Naïve Bayes classifier outperform all the classifiers, denoting the best suitable algorithm for labeled sentence or textual data for binary classifica-tion of sentiments. For further research in the domain of sentiment analysis, instead relying only on TF-IDF feature extraction method, researchers must seek the experi-ment on Word2Vec, Glove or more complex architecture such as BERT embedding. Also, Deep Learning models should be experimented as they are capable to identify hidden pattern in data and capture non-linear relationship.

References

1. T. D. Singh, T. J. Singh, M. Shadang, and S. Thokchom, "Review Comments of Manipuri Online Video: Good, Bad or Ugly," 2021, pp. 45–53. doi: 10.1007/978-981-33-4084-8_5.
2. A. M. Ramadhani and H. S. Goo, "Twitter sentiment analysis using deep learning methods," in 2017 7th International Annual Engineering Seminar (InAES), IEEE, Aug. 2017, pp. 1–4. doi: 10.1109/INAES.2017.8068556.
3. Y. Yang, "Convolutional Neural Networks with Recurrent Neural Filters," Aug. 2018.
4. Y. Niu, X. Zhu, J. Li, and G. Hirst, "Analysis of Polarity Information in Medi-cal Text."
5. E. Prabhakar, M. Santhosh, A. H. Krishnan, T. Kumar, and R. Sudhakar B B Student, "Sentiment Analysis of US Airline Twitter Data using New Adaboost Approach." [Online]. Available: www.ijert.org
6. M. Alzyout, E. AL Bashabsheh, H. Najadat, and A. Alaiad, "Sentiment Analysis of Arabic Tweets about Violence Against Women using Machine Learning," in 2021 12th International Conference on Information and Com-munication Systems (ICICS), IEEE, May 2021, pp. 171–176. doi: 10.1109/ICICS52457.2021.9464600.
7. N. Garg, "Context-Aware Multilingual Fake News Detection Using Machine Learning and Genetic Algorithm-Based Feature Selection," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 13, no. 11, pp. 1502–1510, Nov. 2025, doi: 10.22214/ijraset.2025.75362.
8. N. Garg and P. Singh Negi, "Effective Fake News Detection through Feature Optimization and Model Performance Analysis," 2025.
9. N. Garg and P. S. Negi, "Integrating grey wolf optimiser and artificial bee colony for efficient feature selection in fake news detection," *International Journal of Computing and Artificial Intelligence*, vol. 6, no. 2, pp. 287–292, Jul. 2025, doi: 10.33545/27076571.2025.v6.i2d.208.

10. "1.10. Decision Trees — scikit-learn 1.8.0 documentation." Accessed: Feb. 14, 2026. [Online]. Available: <https://scikit-learn.org/stable/modules/tree.html>
11. "What Is Random Forest? | IBM." Accessed: Feb. 14, 2026. [Online]. Available: <https://www.ibm.com/think/topics/random-forest>
12. "12.1 - Logistic Regression | STAT 462." Accessed: Feb. 14, 2026. [Online]. Available: <https://online.stat.psu.edu/stat462/node/207/>
13. "What Is a Support Vector Machine? - MATLAB & Simulink." Accessed: Feb. 14, 2026. [Online]. Available: <https://www.mathworks.com/discovery/support-vector-machine.html>