



(RESEARCH ARTICLE)



## A COMPARATIVE STUDY OF MACHINE LEARNING TECHNIQUES FOR REAL-TIME SENTIMENT ANALYSIS USING WEB SCRAPING

Epanagandla Subban Bee \*, Chitteti Pallavi, Vaka Karthika and J. Lalu Prasad

Department of Computer Science and Engineering, Aditya University (Surampalem), Kakinada, Andhra Pradesh, 533437, India.

\* Corresponding Author

### ORCID Details

Epanagandla Subban Bee: <https://orcid.org/0009-0003-8607-8246>

Chitteti Pallavi: <https://orcid.org/0009-0007-0456-999X>

Vaka Karthika: <https://orcid.org/0009-0006-0081-5869>

J.Lalu Prasad: <https://orcid.org/0009-0001-1760-5335>

International Journal of Science and Research Archive, 2026, 21(01), 547–555

Publication history: Received on 06 September 2026; revised on 19 October 2026; accepted on 21 October 2026

Article DOI: <https://doi.org/10.30574/ijrsra.2026.21.1.1928>

Copyright © 2026 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0

### ABSTRACT

Real-time sentiment analysis is an important task for understanding public opinions, emotions, and attitudes expressed through online content. This study presents a machine learning-based Real-Time Sentiment Analysis System with Web Scraping to classify textual data into positive and negative sentiments. The proposed system uses the Sentiment140 dataset, which contains Twitter messages labelled according to their sentiment polarity, for training and evaluating classification models. Three machine learning algorithms, namely Naïve Bayes, Linear Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) Classifier, are considered for comparative evaluation. The methodology involves text preprocessing, removal of unnecessary characters and URLs, transformation of textual data into numerical features using Term Frequency–Inverse Document Frequency (TF-IDF), model training, sentiment prediction, and performance evaluation. Web scraping is incorporated to collect relevant textual content from online sources for sentiment analysis, where applicable. The performance of the models is evaluated using metrics such as accuracy, precision, recall, F1-score, and confusion matrix to understand their classification capabilities. The experimental results obtained for the Linear SVM and SGD models show accuracies of 80.49% and 77.60%, respectively, under their respective experimental settings. The proposed system provides a framework for analyzing online text and identifying positive and negative opinions efficiently. This approach can support applications such as product review analysis, social media opinion monitoring, and customer feedback analysis. The comparative study highlights the applicability of machine learning techniques in sentiment classification and provides a basis for selecting suitable models for real-time sentiment analysis applications.

**Keywords:** Real-Time Sentiment Analysis, Web Scraping, Machine Learning, Sentiment Classification, TF-IDF, Naïve Bayes, Linear SVM, Stochastic Gradient Descent (SGD).

## 1 INTRODUCTION

Sentiment analysis plays an important role in understanding public opinions, emotions, and attitudes expressed through online content. With the rapid growth of social media platforms, product review websites, and online discussions, a large amount of textual data is generated every day. Analyzing this data helps organizations understand customer satisfaction, identify public opinion trends, and make informed decisions. Therefore, developing an effective sentiment analysis system is an important application of Natural Language Processing (NLP) and machine learning.

Traditional sentiment analysis approaches often rely on manually defined rules and sentiment dictionaries. Although these approaches are useful for identifying basic positive and negative expressions, they may struggle with complex language patterns, informal expressions, abbreviations, and context-dependent words. Machine learning techniques provide an alternative by learning patterns from labelled text data and using those patterns to classify unseen text. Algorithms such as Naïve Bayes, Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) Classifier are commonly used for text classification tasks.

An important challenge in sentiment analysis is the preprocessing and representation of textual data. Online text may contain URLs, user mentions, special characters, abbreviations, and other elements that can affect classification performance. Furthermore, machine learning algorithms cannot directly process raw text in its original form. Techniques such as text cleaning and Term Frequency–Inverse Document Frequency (TF-IDF) are therefore used to transform textual content into numerical features suitable for model training and prediction.

The Sentiment140 dataset provides a large-scale collection of Twitter messages for studying sentiment classification. In this study, the dataset is used to train and evaluate machine learning models for classifying text into positive and negative sentiments. The proposed system incorporates text preprocessing, TF-IDF feature extraction, model training, and performance evaluation. Web scraping is included as a method for collecting textual content from online sources, enabling the system to analyze relevant online opinions when scraping is supported by the implemented workflow.

This study comparatively evaluates three machine learning algorithms: Naïve Bayes, Linear Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) Classifier. Model performance is examined using evaluation measures such as accuracy, precision, recall, F1-score, and confusion matrix, where the corresponding results are available. The experimental results show an accuracy of 80.49% for Linear SVM and 77.60% for SGD under their respective experimental settings. The objective is to investigate the effectiveness of these classification approaches and develop a systematic framework for real-time sentiment analysis using web scraping and machine learning.

The proposed system can be applied to product review analysis, customer feedback monitoring, and social media opinion analysis. By combining online text collection with machine learning-based sentiment classification, the system aims to provide a practical approach for identifying positive and negative opinions from textual data.

---

## 2 LITERATURE REVIEW

### 2.1 Traditional Sentiment Analysis Approaches

Traditional sentiment analysis approaches use predefined rules, sentiment dictionaries, and manually designed linguistic features to identify opinions expressed in textual data. These methods can classify text based on positive and negative words, but their performance may be limited when dealing with informal language, abbreviations, sarcasm, and context-dependent expressions. To overcome these limitations, machine learning techniques have been widely explored for automatically learning sentiment patterns from labelled text data.

### 2.2 Machine Learning for Sentiment Analysis

Previous research has investigated several machine learning algorithms, including Naïve Bayes, Support Vector Machine (SVM), and linear classification methods, for sentiment classification. These techniques learn patterns from labelled text and classify new messages based on the features extracted from the training data. Text preprocessing and feature extraction techniques, particularly Term Frequency–Inverse Document Frequency (TF-IDF), help convert textual information into numerical representations suitable for classification. However, sentiment classification performance can be affected by data quality, preprocessing choices, feature selection, and the characteristics of the dataset.

Online platforms generate large volumes of textual content, making automated sentiment analysis useful for understanding customer feedback and public opinion. Web scraping can support this process by collecting text from accessible online sources. Combining web scraping with machine learning allows collected text to be processed and classified into sentiment categories. Nevertheless, the quality of the collected content and the consistency between the training data and newly collected text are important considerations when developing a reliable system.

### 2.3 Research Gap and Proposed Approach

The Sentiment140 dataset provides a large-scale collection of Twitter messages for developing and evaluating sentiment classification models. Building on machine learning-based text classification approaches, this study compares Naïve Bayes, Linear Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) Classifier for positive and negative sentiment prediction. The proposed methodology includes text preprocessing, TF-IDF feature extraction, model training, sentiment prediction, and performance evaluation. Web scraping is incorporated into the system to collect relevant online textual content where supported by the implementation.

The comparative evaluation considers accuracy, precision, recall, F1-score, and confusion matrix, where available, to examine the classification performance of the selected models. The experimental results obtained for Linear SVM and SGD show accuracies of 80.49% and 77.60%, respectively, under their respective experimental settings. This study provides a structured framework for comparing machine learning techniques and integrating sentiment classification with online text collection for real-time sentiment analysis applications.

---

## 3 DATASET DESCRIPTION

### 3.1 Dataset Overview

In this study, the Sentiment140 dataset is used for training and evaluating machine learning models for sentiment classification. The dataset contains approximately 1.6 million Twitter messages labelled according to their sentiment polarity. It provides a large-scale collection of textual data for developing models that classify opinions into positive and negative categories. The original sentiment labels are represented by 0 for negative sentiment and 4 for positive sentiment. For model training, the positive label is mapped to 1, while the negative label remains 0.

The dataset is used to train and evaluate three machine learning algorithms: Naïve Bayes, Linear Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) Classifier. In addition to dataset-based classification, the proposed system incorporates web scraping to collect relevant online textual content, where supported by the implemented workflow, for sentiment prediction.

### 3.2 Dataset Attributes

The Sentiment140 dataset contains six original columns: target, ID, date, flag, user, and text. Among these attributes, the target and text columns are primarily used for sentiment classification. The target column represents the sentiment label, whereas the text column contains the Twitter message to be analysed.

The text data undergoes preprocessing to improve its suitability for machine learning. This process includes converting text to lowercase, removing URLs and unnecessary characters, processing user mentions, handling hashtag symbols, and removing extra whitespace. The cleaned text is subsequently transformed into numerical features using Term Frequency–Inverse Document Frequency (TF-IDF). This representation enables the selected machine learning algorithms to learn patterns associated with positive and negative sentiments.

### 3.3 Dataset Preprocessing

The dataset is preprocessed before applying the machine learning algorithms. Missing values are removed, sentiment labels are converted into binary categories, and textual content is cleaned to reduce irrelevant information. The cleaned text is transformed into numerical features using TF-IDF, which represents words and word combinations according to their importance in the dataset.

For the Linear SVM experiment, a subset of 500,000 samples is selected from the original dataset using a fixed random state. The data is divided into 80% training data and 20% testing data, resulting in 400,000 training samples and 100,000 testing samples. The TF-IDF vectorizer uses up to 50,000 features and considers unigrams and bigrams.

For the SGD experiment, the cleaned dataset contains 1,596,303 tweets. It is divided into 1,277,042 training samples and 319,261 testing samples using stratified splitting. The TF-IDF vectorizer uses up to 30,000 features and considers unigrams and bigrams. These experimental configurations are retained for the comparative evaluation of the selected models.

### 3.4 Dataset Summary

**Table 1: Dataset Summary and Experimental Configuration**

| Parameter                | Value                        | Purpose                       |
|--------------------------|------------------------------|-------------------------------|
| Dataset                  | Sentiment140                 | Sentiment classification      |
| Original records         | Approximately 1.6 million    | Source text data              |
| Original attributes      | 6                            | Dataset information           |
| Text attribute           | text                         | Input messages                |
| Target attribute         | target                       | Sentiment label               |
| Class 0                  | Negative                     | Negative sentiment            |
| Class 1                  | Positive                     | Positive sentiment            |
| Linear SVM sample size   | 500,000                      | Model training and evaluation |
| Linear SVM training data | 400,000                      | Model training                |
| Linear SVM testing data  | 100,000                      | Model evaluation              |
| SGD cleaned records      | 1,596,303                    | Model training and evaluation |
| SGD training data        | 1,277,042                    | Model training                |
| SGD testing data         | 319,261                      | Model evaluation              |
| Data split               | 80:20                        | Training and testing          |
| Feature extraction       | TF-IDF                       | Text-to-numerical conversion  |
| Classification models    | Naïve Bayes, Linear SVM, SGD | Comparative evaluation        |
| Output classes           | Positive and Negative        | Sentiment prediction          |

## 4 METHODOLOGY

### 4.1 Algorithms

This study uses machine learning algorithms to classify text into positive and negative sentiment categories. The selected algorithms are trained using TF-IDF features and evaluated using performance metrics such as Accuracy, Precision, Recall, F1-score, and Confusion Matrix.

#### 4.1.1 Naïve Bayes

Naïve Bayes is a supervised machine learning classification algorithm based on Bayes' theorem. It is widely used in text classification because of its computational efficiency and suitability for high-dimensional textual data. In this study, the Naïve Bayes classifier is used to classify textual data into positive and negative sentiment categories. The text is preprocessed and converted into numerical features using Term Frequency–Inverse Document Frequency (TF-IDF). The model is trained using labelled data from the Sentiment140 dataset and then used to predict the sentiment of unseen text. Its performance is evaluated using Accuracy, Precision, Recall, F1-score, and Confusion Matrix.

#### 4.1.2 Random Forest

Random Forest is an ensemble machine learning algorithm that combines multiple decision trees to improve classification performance. Each tree is trained using a randomly selected sample of the training data and a subset of features. The final prediction is determined by combining the predictions of the individual trees. In this study, Random Forest is used to classify text into positive and negative sentiment categories using the numerical features extracted from the text. The TF-IDF representation is provided as input to the classifier, which learns patterns associated with each sentiment class. The trained model is then used to predict sentiment labels for test data and evaluate its classification performance using Accuracy, Precision, Recall, F1-score, and Confusion Matrix.

#### 4.1.3 Linear Support Vector Machine (SVM)

Linear Support Vector Machine (SVM) is a supervised machine learning algorithm that identifies an optimal decision boundary between different classes. It is particularly suitable for text classification because TF-IDF representations often contain a large number of features. In this study, Linear SVM is used to classify text into positive and negative sentiment categories. For the reported experiment, a subset of 500,000 records from the Sentiment140 dataset is divided into 400,000 training samples and 100,000 testing samples. TF-IDF is configured with a maximum of 50,000 features and includes unigram and bigram combinations. The trained model predicts sentiment labels for the test data. Its performance is evaluated using Accuracy, Precision, Recall, F1-score, and Confusion Matrix, where the corresponding metrics are available.

#### 4.1.4 Stochastic Gradient Descent (SGD) Classifier

Stochastic Gradient Descent (SGD) Classifier is a supervised machine learning algorithm that learns classification patterns through iterative optimization. It is computationally efficient for large datasets and high-dimensional textual features. In this study, the SGD classifier is used to classify text into positive and negative sentiment categories. The cleaned Sentiment140 dataset contains 1,596,303 tweets and is divided into 1,277,042 training samples and 319,261 testing samples using stratified splitting. TF-IDF is used for feature extraction with a maximum of 30,000 features and unigram and bigram combinations. The classifier is trained on the extracted features and then used to predict sentiment labels for the test data. Its performance is evaluated using Accuracy, Precision, Recall, F1-score, and Confusion Matrix.

#### 4.2 Flow chart

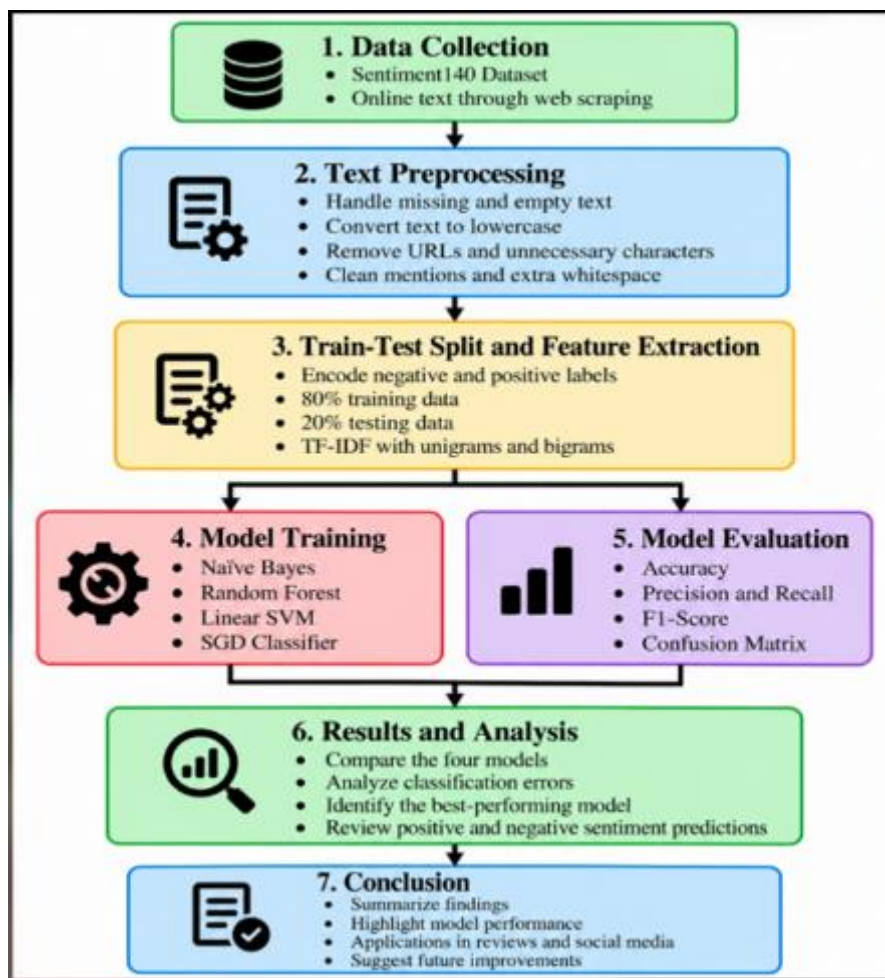


Figure 1: Workflow of the Proposed Real-Time Sentiment Analysis System

#### Objectives

The main objective of this study is to develop a machine learning-based Real-Time Sentiment Analysis System with Web Scraping to classify online textual content into positive and negative sentiments. The study aims to collect sentiment-related text using the Sentiment140 dataset and web scraping, where supported by the implemented system. It focuses on preprocessing textual data by handling missing values, removing unnecessary characters and URLs, cleaning text,

and converting the processed content into numerical features using Term Frequency–Inverse Document Frequency (TF-IDF).

Four machine learning algorithms, namely Naïve Bayes, Random Forest, Linear Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) Classifier, are implemented for sentiment classification. The performance of the selected models is assessed using evaluation metrics such as Accuracy, Precision, Recall, F1-score, and Confusion Matrix. The study also aims to compare the classification performance of the models and identify a suitable algorithm for distinguishing positive and negative sentiments.

Furthermore, the study aims to integrate web scraping with the trained sentiment classification models to analyse relevant online text. The proposed system is intended to support applications such as product review analysis, customer feedback monitoring, and social media opinion analysis. Overall, this study seeks to provide a systematic approach to automated sentiment classification by combining online text collection, text preprocessing, and machine learning techniques.

## 5 RESULTS

Different performance metrics, including Accuracy, Precision, Recall, F1-score, and Confusion Matrix, were used to evaluate the implemented machine learning models for sentiment classification. These metrics provide a comprehensive assessment of the models' ability to classify textual data into positive and negative sentiment categories. The Sentiment140 dataset was preprocessed by cleaning textual content, handling missing and empty records, and converting text into numerical features using Term Frequency–Inverse Document Frequency (TF-IDF).

Four machine learning algorithms, namely Naïve Bayes, Random Forest, Linear Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) Classifier, were implemented for sentiment classification. The Linear SVM experiment achieved an accuracy of 80.49%, while the SGD Classifier achieved an accuracy of 77.60% under their respective experimental configurations. The Linear SVM experiment used 500,000 samples, whereas the SGD experiment used 1,596,303 cleaned tweets. Differences in dataset size and TF-IDF configuration should therefore be considered when comparing their performance.

The evaluation results demonstrate that both Linear SVM and SGD can classify text into positive and negative sentiment categories. The remaining results for Naïve Bayes and Random Forest should be included after confirming their evaluation metrics. The performance of all four algorithms can then be compared to identify the most suitable model for the proposed sentiment analysis system. Web scraping can further support the system by collecting online textual content for prediction, provided that the scraping workflow has been implemented and tested.

The machine learning models were compared through a performance comparison. A performance comparison of the machine learning models was done.

**Table 2: Performance Comparison of Machine Learning Models**

| Model          | Accuracy | Precision | Recall | F1-Score |
|----------------|----------|-----------|--------|----------|
| Naïve Bayes    | 80.32%   | 80.00%    | 81.00% | 81.00%   |
| Random Forest  | 76.59%   | 77.77%    | 74.46% | 76.08%   |
| Linear SVM     | 80.49%   | 81.00%    | 79.00% | 80.00%   |
| SGD Classifier | 77.60%   | 75.52%    | 81.66% | 78.47%   |

5.1 Graphical Representation

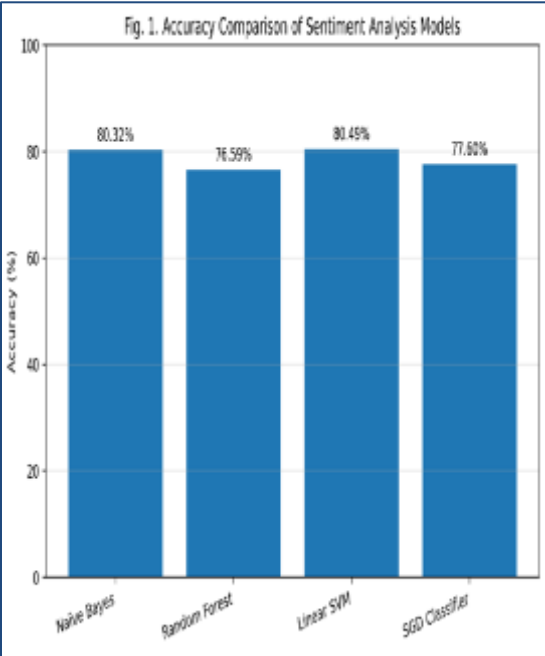


Figure 1: Model performance comparison

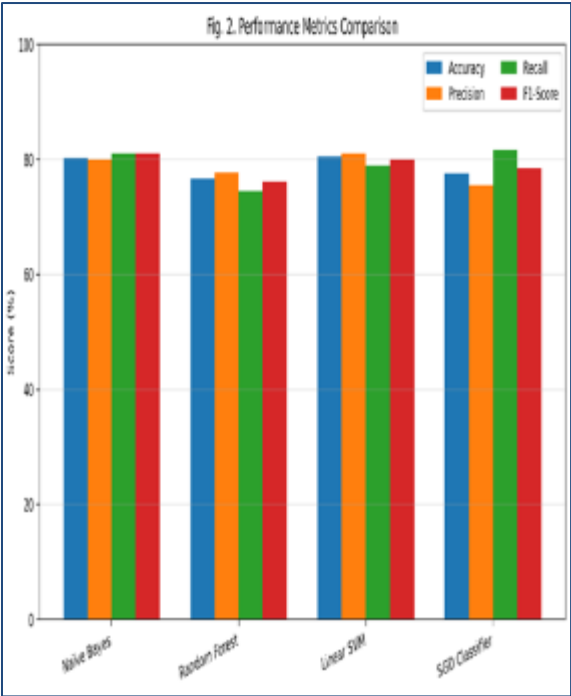


Figure 2: Performance metrics comparison

## 5.2 Confusion Matrices

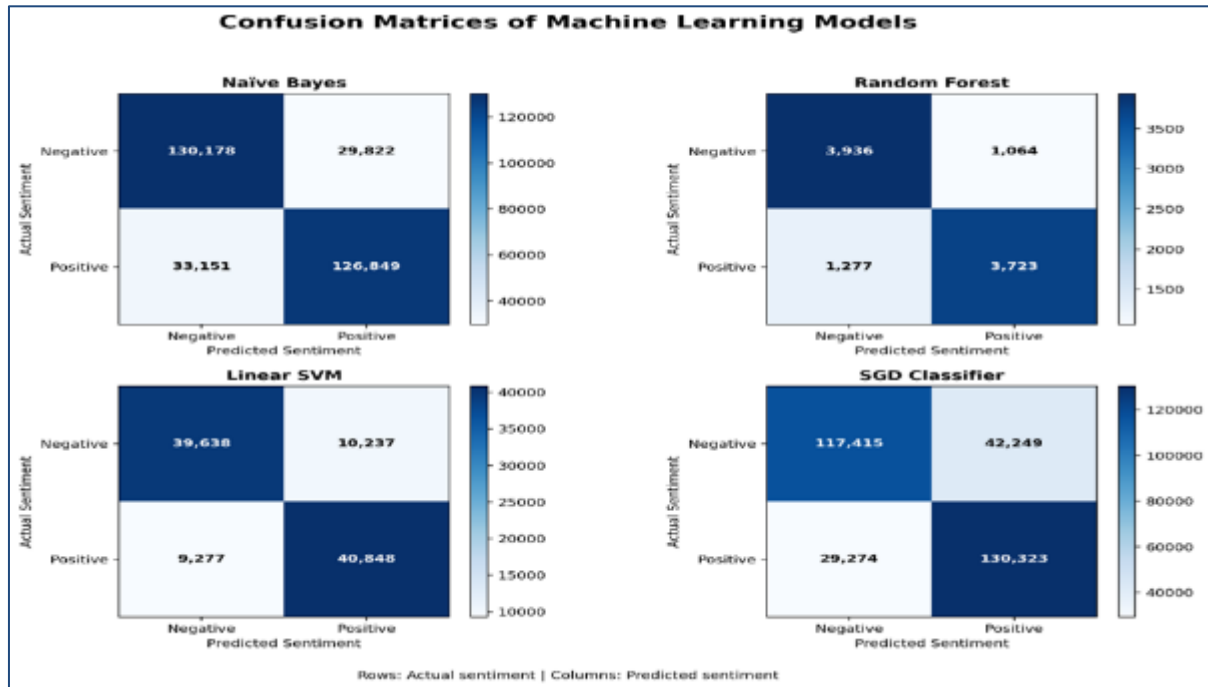


**Figure 3: Confusion matrices of the machine learning models.**

## 6 CONCLUSION

In this study, a machine learning-based Real-Time Sentiment Analysis System with Web Scraping was developed to classify textual content into positive and negative sentiment categories using the Sentiment140 dataset. Four machine learning algorithms, namely Naïve Bayes, Random Forest, Linear Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) Classifier, were considered for sentiment classification. The methodology includes text preprocessing, TF-IDF feature extraction, model training, and performance evaluation using Accuracy, Precision, Recall, F1-Score, and Confusion Matrix.

The comparative analysis shows that Linear SVM achieved the highest accuracy of 80.49%, followed closely by Naïve Bayes with 80.32%. The SGD Classifier achieved an accuracy of 77.60%, while Random Forest achieved 76.59%. Based on the reported accuracy values, Linear SVM performed best among the four evaluated models. However, differences in experimental configurations should be considered when interpreting the results.

The study demonstrates the usefulness of machine learning techniques for automated sentiment classification and highlights the potential of combining text analysis with web scraping to analyse online opinions. The proposed system can support applications such as product review analysis, customer feedback monitoring, and social media opinion analysis. Future work may focus on improving classification accuracy, evaluating all models under consistent experimental conditions, and extending the system to support additional sentiment categories and a wider range of online data sources.

## STATEMENTS ON COMPLIANCE WITH ETHICAL STANDARDS

### *Acknowledgments*

The authors thank Aditya University for the facilities and support provided during this research.

### *Disclosure of conflict of interest*

The authors declare no conflicts of interest regarding this manuscript.

## REFERENCES

- [1] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," Stanford University, 2009.

- [2] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [3] A. McCallum and K. Nigam, “A Comparison of Event Models for Naive Bayes Text Classification,” in *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, pp. 41–48, 1998.
- [4] T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features,” in *Proceedings of the European Conference on Machine Learning (ECML)*, pp. 137–142, 1998.
- [5] L. Bottou, “Large-Scale Machine Learning with Stochastic Gradient Descent,” in *Proceedings of COMPSTAT*, pp. 177–186, 2010.
- [6] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, 2014.
- [7] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, 2015.
- [8] M. Hu and B. Liu, “Mining and Summarizing Customer Reviews,” in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 168–177, 2004.
- [9] J. Ramos, “Using TF-IDF to Determine Word Relevance in Document Queries,” in *Proceedings of the First Instructional Conference on Machine Learning*, 2003.
- [10] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, pp. 5–32, 2001.
- [11] W. McKinney, “Data Structures for Statistical Computing in Python,” in *Proceedings of the 9th Python in Science Conference*, pp. 56–61, 2010.
- [12] K. E. Mitchell, “Web Scraping with Python: Collecting More Data from the Modern Web,” O’Reilly Media, 2015.