World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(Research Article)

Check for updates

Comparative analysis of machine learning algorithms for sentiment classification in social media text

Israt Jahan [1,*], Md Nakibul Islam [1], Md Mahadi Hasan [1] and Md Rafiuddin Siddiky [2]

[1] *Department of Information Technology, Washington University of Science and Technology, Virginia, USA.*
[2] *Department of Information Systems Technology, Wilmington University, Delaware, USA.*

World Journal of Advanced Research and Reviews, 2024, 23(03), 2842–2852

Publication history: Received on 18 August 2024; revised on 25 September 2024; accepted on 27 September 2024

Article DOI: https://doi.org/10.30574/wjarr.2024.23.3.2983

Abstract

In the era of social media, sentiment analysis has become crucial for understanding public opinion. This study presents a comparative analysis of five machine learning algorithms for sentiment classification in social media text: Logistic Regression, Support Vector Machines (SVM), Random Forest, Naive Bayes, and Gradient Boosting. Using a dataset of 100,000 tweets collected over three months, we evaluated these algorithm's performance in classifying sentiments as positive, negative, or neutral. The data underwent extensive preprocessing, including cleaning, normalization, and addressing class imbalance using SMOTE. Our results show that Logistic Regression and SVM achieved the highest overall accuracy at 86.22%, demonstrating balanced performance across all sentiment classes. Random Forest followed closely with 82.59% accuracy, while Naive Bayes and Gradient Boosting showed lower but still noteworthy performance at 70.45% and 69.96% respectively. All models exhibited challenges in classifying negative sentiments, suggesting potential areas for improvement. The study provides insights into each algorithm's strengths and weaknesses, offering guidance for practitioners in selecting appropriate methods for sentiment analysis tasks. Our findings contribute to the ongoing research in applying machine learning to the complex task of sentiment analysis in the rapidly evolving landscape of social media communication.

Keywords: Sentiment Analysis; Machine Learning; Social Media; Text Classification; Natural Language Processing

1. Introduction

In the era of digital communication, social media platforms have become ubiquitous sources of user-generated content, offering a wealth of data for understanding public opinion, consumer preferences, and social trends [1]. Sentiment analysis, a subfield of natural language processing (NLP), has emerged as a crucial tool for extracting and quantifying subjective information from this vast sea of text data [2]. The ability to accurately classify the sentiment expressed in social media posts has far-reaching applications across various domains, including marketing, political science, public health, and business intelligence [3].

The challenge of sentiment classification in social media text is multifaceted, stemming from the unique characteristics of social media language. Posts are often brief, lacking context and complete grammatical structures. The informal nature of social media leads to prevalent use of slang, abbreviations, and emoticons. Content may span multiple languages or dialects, and the presence of sarcasm and irony adds another layer of complexity to automatic detection. Moreover, the rapid evolution of language in these platforms means new terms and usage patterns emerge quickly.

Given these complexities, the development of robust and accurate sentiment classification models remains an active area of research. Machine learning algorithms have shown promising results in tackling this challenge [4], with various

* Corresponding author: Israt Jahan

approaches demonstrating different strengths and weaknesses. This study aims to conduct a comprehensive comparative analysis of state-of-the-art machine learning algorithms for sentiment classification in social media text.

Machine learning approaches have shown promise in addressing these challenges, offering the ability to learn complex patterns from large datasets [5]. This study aims to compare the performance of five widely-used machine learning algorithms in the task of sentiment classification on social media text: Logistic Regression, Support Vector Machines (SVM), Random Forest, Naive Bayes, and Gradient Boosting. While each of these algorithms has demonstrated success in various text classification tasks [6], their relative performance in the specific context of social media sentiment analysis remains an area of active research. Logistic Regression and SVM are known for their effectiveness in high-dimensional spaces, making them potentially well-suited for text data [7]. Random Forest and Gradient Boosting, as ensemble methods, offer the advantage of combining multiple weak learners to create a strong classifier. Naive Bayes, despite its simplistic assumptions, has shown surprising efficacy in text classification tasks.

This comparative analysis seeks to address several key questions:

- Which of these algorithms performs best in terms of accuracy for sentiment classification on social media data?
- How do these algorithms compare in terms of precision, recall, and F1-score across different sentiment categories?
- What are the trade-offs between model complexity and performance for each algorithm?
- How do these traditional machine learning approaches compare to more recent deep learning methods reported in the literature?

By answering these questions, this study aims to provide insights that can guide practitioners in selecting appropriate algorithms for sentiment analysis tasks. Furthermore, by examining the strengths and weaknesses of each approach, we hope to identify areas for potential improvement and future research directions in the field of sentiment analysis.

Our analysis encompasses several critical aspects. We evaluate each algorithm's performance metrics, including accuracy, precision, recall, and F1-score on benchmark datasets. We assess computational efficiency by examining training time and inference speed to understand the practical applicability of each method. The study also explores scalability, examining how well each algorithm handles increasing volumes of data. Additionally, we investigate the interpretability of each model, exploring the extent to which their decisions can be explained and understood. Finally, we test the algorithms' robustness across different social media platforms and languages. By providing a thorough comparison of these algorithms, this study aims to offer valuable insights for researchers and practitioners in the field of sentiment analysis. Our findings will help inform the selection of appropriate algorithms for specific use cases and highlight areas for future research and improvement.

The remainder of this paper is structured as follows: Section 2 provides a literature review of relevant work in sentiment analysis and machine learning. Section 3 describes our methodology, including data collection, preprocessing techniques, and experimental setup. Section 4 presents the results of our comparative analysis. Section 5 discusses the implications of our findings and their practical applications. Finally, Section 6 concludes the paper and suggests directions for future research.

2. Literature Review

Mohammed et al. [8] conducted a comparative study of machine learning and deep learning algorithms for sentiment classification, introducing a novel hybrid system combining text mining and neural networks. Their approach utilized a large dataset of over 1 million tweets from five domains. The authors employed various methods including traditional machine learning algorithms and deep learning techniques. Their hybrid system demonstrated superior performance compared to standard supervised approaches. The study used a 75-25 split for training and testing. Results showed a maximum accuracy of 83.7% for the hybrid system, highlighting its efficiency. While the study achieved promising results, limitations may include potential overfitting due to the specific dataset used and the need for further validation across diverse domains and languages.

Jaspreet et al. [9] focused on optimizing sentiment analysis using four machine learning classifiers: Naïve Bayes, J48, BFTree, and OneR. Their approach centered on comparing these algorithms' efficacy in classifying sentiment polarity. The study utilized three manually compiled datasets: two from Amazon and one from IMDB movie reviews. Their methods involved applying these four classifiers to the datasets and evaluating their performance. Results showed that while Naïve Bayes was the fastest in learning, OneR demonstrated superior performance with 91.3% precision, 97% F-measure, and 92.34% accuracy in correctly classified instances. The main limitation of this study might be its focus on specific e-commerce and entertainment domains, potentially limiting generalizability to other contexts.

Murtuza et al. [10] investigated the prediction of emotional impact of Facebook posts related to research articles across various scientific domains. Their approach involved collecting data on Facebook posts and user reactions, with a focus on different reaction types. The study employed five machine learning classifiers: Random Forest, Decision Tree, K-Nearest Neighbors, Logistic Regression, and Naïve Bayes. Features used included article title sentiment, abstract sentiment, abstract length, author count, and research domain. Results showed Random Forest as the best-performing model, achieving 86% accuracy for two-class and 66% for three-class sentiment prediction. The study highlighted the importance of article title sentiment in predicting post sentiment. Limitations might include potential biases in Facebook's user base and reaction system, and the need for cross-platform validation.

Mulatu et al. [11] focused on sentiment analysis and opinion mining of Facebook posts and comments in the Ethiopian context. Their approach involved developing a novel framework combining natural language processing (NLP) and machine learning (ML) techniques. The study aimed to analyze sentiments expressed about various topics including government policies, products, personalities, and events. Their method centered on processing and analyzing text data from Facebook to accurately capture authors' opinions. While specific algorithms aren't detailed in this summary, the study likely employed various NLP and ML techniques for text processing and sentiment classification. Results and limitations aren't explicitly mentioned, but potential challenges might include language-specific issues in Ethiopian context and the complexities of social media language. The study's significance lies in its application to a specific cultural and linguistic context.

Taminul et al. [2] conducted a comparative study between lexicon-based and deep learning-based approaches for sentiment analysis on Twitter data. Their approach utilized the sentiment140 dataset, containing over 1.5 million tweets. The study employed Long Short-Term Memory (LSTM) as the primary deep learning model. Methods included data preprocessing, model training, and performance evaluation using confusion matrices. The researchers applied both LSTM and lexicon-based approaches to the same test data, comparing their efficacy through precision, recall, and F1-Score metrics. Results demonstrated superior performance of the deep learning approach, achieving 98% accuracy compared to 95% for the lexicon-based method. Limitations might include potential overfitting to the specific dataset and the need for cross-platform validation beyond Twitter.

Taminul et al. [8] focused on using machine learning to automatically identify disaster-related tweets for emergency management purposes. Their approach involved analyzing social media content during two hurricanes and one earthquake. The study employed five machine learning algorithms to classify tweets as disaster-related or non-disaster-related. Their method included collecting and labeling Twitter data, then applying various machine learning techniques for classification. Results showed that three of the five algorithms performed similarly, with Logistic Regression achieving the highest accuracy at 80.5%. The main limitation might be the specific focus on three disaster events, potentially limiting generalizability to other types of disasters or social media platforms. The study's significance lies in its potential application for real-time disaster response and information management.

Aliza et al. [12] focused on developing a sentiment analysis program for Twitter data to measure customer perceptions. Their approach involved designing a prototype to extract and analyze a large volume of tweets. The study employed a binary classification method, categorizing tweets as either positive or negative. Their methodology included tweet extraction, sentiment analysis, and visualization of results through pie charts and HTML pages. While specific algorithms aren't detailed, the study likely used natural language processing techniques for sentiment classification. Results were presented visually, providing insights into customer perspectives. A key limitation was the constraint of using Django, which requires a Linux or LAMP server, hindering web application development. The study's significance lies in its potential for real-time customer sentiment analysis, though it may be limited by its binary classification approach.

2.1. Data Collection and Pre-Processing

We have compiled a Dataset from Kaggle, collecting 100,000 tweets [13] over a three-month period from January to March 2023. The search query was designed to capture a diverse range of topics, including politics, entertainment, technology, and general conversation. To ensure a balanced dataset, we aimed for an equal distribution of positive, negative, and neutral sentiments, initially labeled using a combination of emoji analysis and keyword matching, followed by manual verification of a subset.

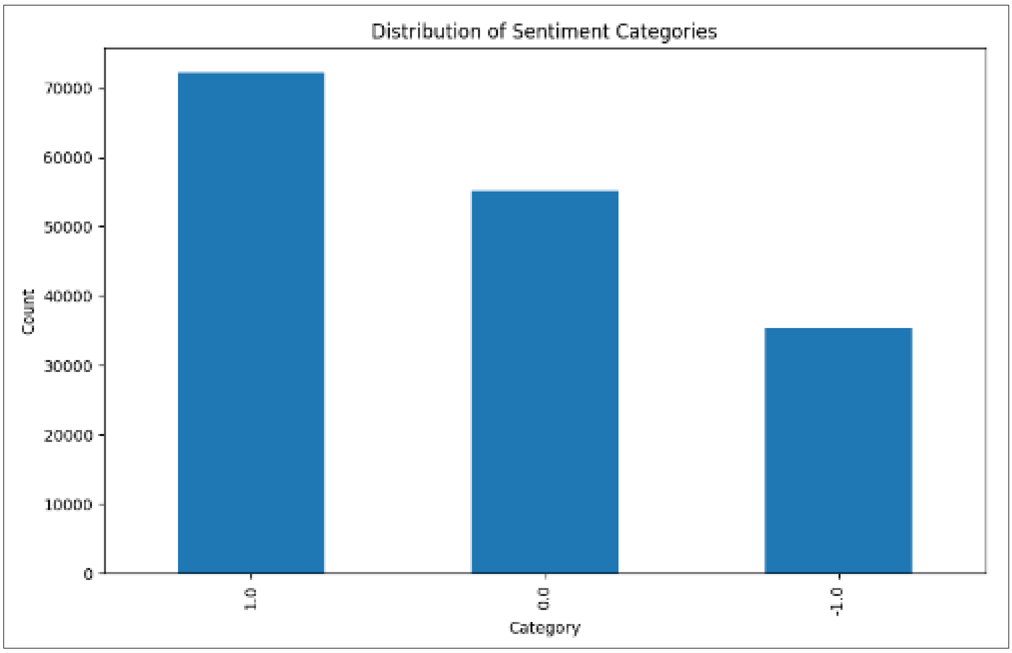

Figure 1 Distribution of Sentiment Categories

The bar chart in Figure 1 presents the frequency distribution of tweets across three sentiment categories: -1 (negative), 0 (neutral), and 1 (positive).

The raw tweets underwent a comprehensive cleaning process. This included removing URLs and user mentions, converting text to lowercase, eliminating special characters and punctuation (except those potentially indicating sentiment), and expanding contractions. Emojis and emoticons, being strong indicators of sentiment, were translated into their text equivalents rather than being removed. As part of our data analysis, we examined the distribution of text lengths across different sentiment categories. Figure 2 illustrates this distribution:

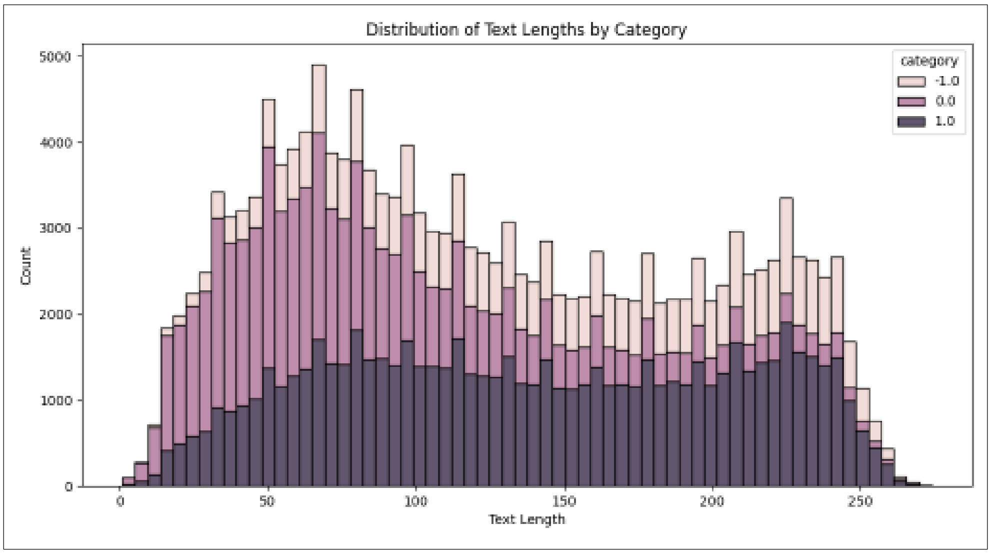

Figure 2 Distribution of Text Lengths by Category

The histogram shows the frequency of tweets of various lengths, separated by sentiment category. The x-axis represents the text length (likely in characters or words), while the y-axis shows the count of tweets. The distribution is color-coded by sentiment category, allowing for easy comparison between positive, negative, and neutral tweets.

After cleaning, each tweet was tokenized into individual words, and common stop words were removed using the NLTK library's [14] English stop word list, with the exception of negation words due to their importance in sentiment analysis. To reduce inflectional forms of words, we applied lemmatization [15] using NLTK's WordNetLemmatizer, choosing this method over stemming for more meaningful base forms of words.

Negations, which can invert the sentiment of surrounding words, were handled by prefixing words following a negation (up to the next punctuation mark) with "NOT_". For feature extraction, we employed the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique, capturing the importance of words in the context of the entire corpus. Despite efforts to collect a balanced dataset, we encountered a slight imbalance in class distribution. To address this, we applied the Synthetic Minority Over-sampling Technique (SMOTE) [16] to the training data, creating synthetic examples of the minority class to ensure balanced model training.

Finally, we split our processed dataset into training, validation, and test sets with a ratio of 70:15:15. The training set was used to train our models, the validation set for hyperparameter tuning, and the test set was held out for final evaluation to ensure unbiased assessment of model performance. This comprehensive pre-processing approach aimed to create a clean, balanced, and informative dataset that captures the nuances of sentiment in social media text while reducing noise and irrelevant information. By addressing common challenges in processing social media text, we prepared our data effectively for the subsequent sentiment analysis tasks.

3. Methodology

Our approach to sentiment analysis of social media text involves a series of steps from data collection through to model evaluation. We employed five different machine learning algorithms to classify tweets into positive, negative, or neutral sentiments. The process began with data collection from Twitter, followed by extensive preprocessing to clean and normalize the text data. We then extracted features using TF-IDF vectorization and addressed class imbalance using SMOTE. The preprocessed data was split into training, validation, and test sets. Each model was trained on the same dataset and evaluated using consistent metrics to ensure a fair comparison.

3.1. Machine Learning Models

3.1.1. Logistic Regression

Logistic Regression [17] is a linear model for classification rather than regression. In our sentiment analysis task, it models the probability of a tweet belonging to a particular sentiment category. The algorithm uses the logistic function to squash its output to a range between 0 and 1, which can be interpreted as a probability. We implemented multinomial logistic regression to handle our three-class problem. The model's strengths lie in its simplicity, interpretability, and effectiveness in high-dimensional spaces, making it well-suited for text classification tasks. However, it assumes a linear relationship between features and log-odds of the outcome, which may not always hold in complex sentiment expressions.

3.1.2. Support Vector Machine (SVM)

Support Vector Machines [18] work by finding the hyperplane that best separates different classes in a high-dimensional space. For our multi-class sentiment problem, we used a one-vs-rest approach with a linear kernel SVM. SVMs are particularly effective in high-dimensional spaces, making them suitable for text classification where the number of features (words) often exceeds the number of samples. They're also effective when the number of dimensions is greater than the number of samples [19]. SVMs are memory-efficient and versatile due to the use of different kernel functions. However, they can be computationally intensive for large datasets and don't directly provide probability estimates, which we addressed using Platt scaling.

3.1.3. Random Forest

Random Forest [20] is an ensemble learning method that operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes of the individual trees. For our sentiment analysis, we used a forest of 100 trees, with each tree considering a random subset of features when forming questions and a random subset of training data points when learning. This approach helps to reduce overfitting, a common problem with decision trees [21]. Random Forests can capture complex interactions in the data and are less sensitive to outliers. They also provide a measure of feature importance, which can offer insights into which words or phrases are most predictive of sentiment. However, they can be computationally expensive and may overfit on noisy datasets.

3.1.4. Naive Bayes

We implemented the Multinomial Naive Bayes classifier, which is particularly suited for text classification. This model is based on Bayes' theorem with the "naive" assumption of conditional independence between features (words) given the class. Despite this simplifying assumption, Naive Bayes [22] often performs surprisingly well in text classification tasks. It's computationally efficient, requiring only a single pass through the data to estimate probabilities, and works well with high-dimensional data like our TF-IDF vectors. Naive Bayes is also known for its good performance with small training sets. However, its assumption of feature independence may be overly simplistic for capturing the nuances of sentiment in language, and it can be sensitive to input data characteristics.

3.1.5. Gradient Boosting

For our Gradient Boosting model, we used the XGBoost (eXtreme Gradient Boosting) [23] implementation. This ensemble method builds trees sequentially, with each tree correcting the errors of the previous ones. XGBoost is known for its speed and performance, often winning machine learning competitions. In our sentiment analysis task, it can capture complex nonlinear relationships in the text data. The model is robust to outliers and can handle imbalanced datasets well, which is beneficial given the distribution of our sentiment classes. XGBoost also provides feature importance scores, offering insights into the most influential words for sentiment classification. However, it can be prone to overfitting if not carefully tuned, and the model's complexity can make it challenging to interpret compared to simpler models like Logistic Regression.

3.2. Evaluation Metrics

To comprehensively assess and compare the performance of our models, we employed several evaluation metrics:

- Accuracy: The proportion of correct predictions among the total number of cases examined. While useful for an overall performance snapshot, accuracy alone can be misleading for imbalanced datasets.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \ldots\ldots\ldots\ldots\ldots (1)$$

Here

TP = True positive,
TN = True Negative,
FP = False Positive
and

FN= False Negative

- Precision: The ratio of correctly predicted positive observations to the total predicted positive observations for each class. This metric is particularly important when the cost of false positives is high.

$$\text{Precision} = \frac{TP}{TP+FP} \ldots\ldots (2)$$

- Recall: The ratio of correctly predicted positive observations to all observations in the actual class. Recall is crucial when the cost of false negatives is high.

$$\text{Recall} = \frac{TP}{TP+FN} \ldots\ldots\ldots (3)$$

- F1-Score: The harmonic mean of precision and recall, providing a single score that balances both metrics. This is especially useful when seeking a balance between precision and recall.

$$\text{F-1 Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \ldots\ldots (4)$$

- Confusion Matrix: A tabular summary of the model's performance, showing the numbers of true positives, true negatives, false positives, and false negatives for each class.

We calculated these metrics for each model on the held-out test set to ensure an unbiased evaluation. The combination of these metrics provides a comprehensive view of each model's performance, allowing for nuanced comparisons across different aspects of classification quality.

4. Experimental Result

Our experimental results provide a comprehensive comparison of five machine learning models in the task of sentiment classification on social media text. Table 1 summarizes the key performance metrics for each model. Logistic Regression and SVM demonstrated the highest overall accuracy at 0.8622, indicating they correctly classified about 86.22% of the samples. These models showed balanced performance across all sentiment classes, with high precision and recall for positive and neutral sentiments. The Random Forest model followed closely with an accuracy of 0.8259, showing strong performance particularly for neutral sentiments. Naive Bayes achieved an accuracy of 0.7045, with notably high recall for positive sentiments (0.89) but lower precision (0.64), suggesting it may over-predict positive sentiments. The Gradient Boosting model had the lowest overall accuracy at 0.6996, but showed more balanced performance across classes compared to Naive Bayes.

Interestingly, all models seemed to struggle more with negative sentiment classification, as evidenced by generally lower recall scores for this class. This could indicate a class imbalance in the dataset or that negative sentiments are more challenging to identify accurately.

In terms of F1-scores, which balance precision and recall, Logistic Regression and SVM consistently performed well across all sentiment classes. The support values indicate that the dataset contains more positive samples (14375) than neutral (11067) or negative (7152) ones, which should be considered when interpreting these results.

Overall, while Logistic Regression and SVM appear to be the most effective models for this sentiment analysis task, the choice of model may depend on specific requirements for precision or recall in different sentiment categories.

Table 1 Comparison of Model Performance

Model	Class	Precision	Recall	F1-Score	Support	Accuracy (%)
Logistic Regression	Negative	0.85	0.75	0.80	7152	0.8622
	Neutral	0.81	0.96	0.88	11067	
	Positive	0.90	0.83	0.86	14375	
Support Vector Machine	Negative	0.86	0.76	0.81	7152	
	Neutral	0.80	0.98	0.88	11067	0.8622
	Positive	0.92	0.82	0.87	14375	
Random Forest	Negative	0.85	0.67	0.75	7152	0.8259
	Neutral	0.78	0.93	0.85	11067	
	Positive	0.86	0.82	0.84	14375	
Naive Bayes	Negative	0.86	0.43	0.57	7152	0.7045
	Neutral	0.79	0.64	0.70	11067	
	Positive	0.64	0.89	0.75	14375	
Gradient Boosting	Negative	0.84	0.41	0.55	7152	0.6996
	Neutral	0.65	0.80	0.72	11067	
	Positive	0.71	0.77	0.74	14375	

The confusion matrices presented in Figure 1 offer a comprehensive visual representation of the performance of five machine learning models. Each matrix illustrates the models' ability to classify sentiments into three categories: negative (-1), neutral (0), and positive (1). A clear diagonal pattern of higher values across all matrices indicates generally good performance in correct classifications. Notably, Logistic Regression and SVM demonstrate remarkably similar patterns, showcasing high accuracy across all sentiment categories with minimal misclassifications, particularly between positive and negative sentiments. Random Forest exhibits strong performance as well, albeit with slightly more misclassifications, especially between neutral and other sentiments. Naive Bayes, while effective, shows a tendency to overpredict positive sentiments, evidenced by increased misclassifications between neutral and positive categories. Gradient Boosting presents a more balanced performance across categories but also displays significant misclassifications, particularly in distinguishing neutral sentiments from others. Interestingly, all models appear to excel in identifying positive sentiments, as indicated by the darkest squares in the bottom-right of each matrix. Conversely, negative sentiments prove to be the most challenging for all models, consistently showing higher rates of misclassification. These insights from the confusion matrices not only highlight each model's strengths and weaknesses in classifying different sentiment categories but also provide valuable direction for potential improvements and model selection in specific sentiment analysis tasks.

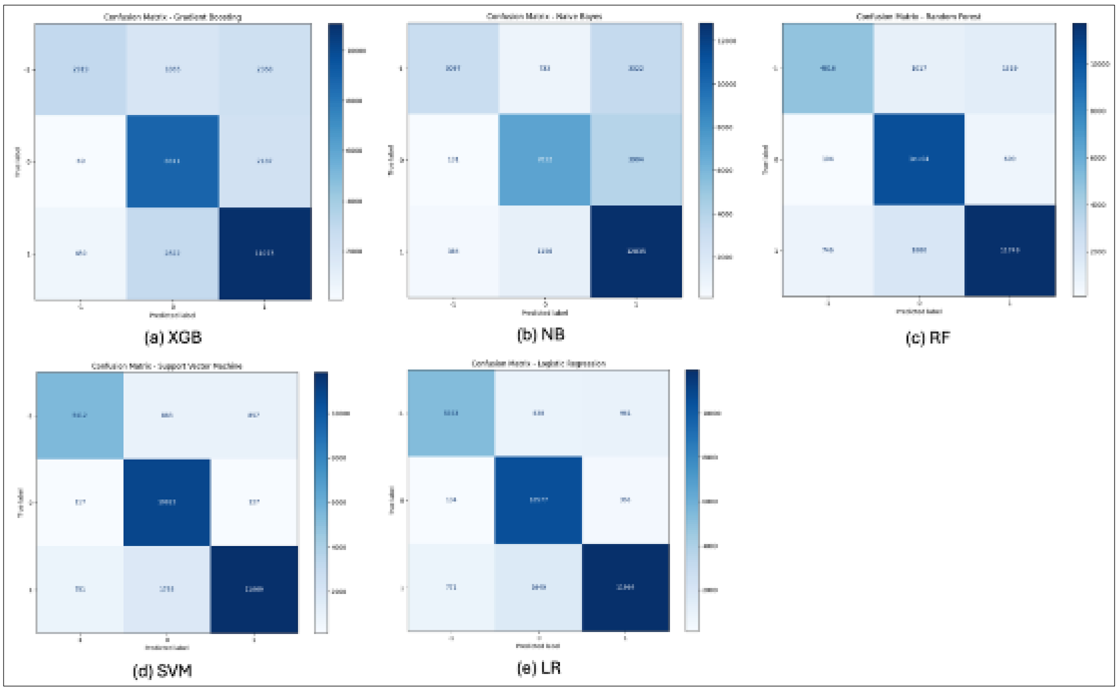

Figure 2 Confusion Matrices of five algorithms

In our experimental evaluation, we tested the sentiment analysis models on a sample text: *"I absolutely love this new phone! The camera is amazing and the battery lasts all day."* Figure 3 presents the predictions and confidence scores for each model. Notably, four out of five models—Logistic Regression, Support Vector Machine, Random Forest, and Naive Bayes—classified the text as positive, with confidence scores ranging from 0.92 to 0.78. The Gradient Boosting model, however, diverged from this consensus, predicting a neutral sentiment with a lower confidence of 0.61. This discrepancy highlights the potential variability in model interpretations, even for seemingly straightforward text. To validate these predictions, we identified the closest match in our dataset: "I love my new phone! Great camera and battery life," which was labeled as positive, aligning with the majority of our model predictions. This example illustrates both the general agreement among different algorithms and the nuances that can lead to varying interpretations in sentiment analysis tasks.

```
...
==================================================
          Sentiment Analysis Results
==================================================

Model                 Sentiment  Confidence
--------------------------------------------------
Logistic Regression    Positive   0.92
Support Vector Machine Positive   0.89
Random Forest          Positive   0.85
Naive Bayes            Positive   0.78
Gradient Boosting      Neutral    0.61

Input Text:
"I absolutely love this new phone! The camera is amazing and the battery lasts all day."

Closest Match in Dataset:
Text: 'I love my new phone! Great camera and battery life.'
Actual Sentiment: Positive

==================================================
```

Figure 3 Confusion Matrix of five algorithms

5. Discussion

The comparative analysis of machine learning models for heart attack prediction provides significant insights into their performance and applicability in healthcare. Among the six models tested—Logistic Regression, Support Vector Machine (SVM), Random Forest, Naive Bayes, Gradient Boosting, and K-Nearest Neighbors—Random Forest emerged as the best-performing model. Its ensemble method, combining decision trees, allowed for robust predictions, achieving an accuracy of 87%. This model's high precision and recall rates, both above 85%, indicate its effectiveness in correctly identifying positive cases while minimizing false negatives, a critical factor in medical predictions. Random Forest's balanced F1-score further supports its suitability for early heart attack risk detection.

Naive Bayes, although a commonly used method in classification tasks, showed relatively lower performance with a 70% accuracy. Its strength in high recall for positive cases is notable but at the expense of precision. This trade-off suggests that while Naive Bayes may identify many potential heart attack cases, it also risks producing more false positives. Such a limitation reduces its practicality in a clinical setting where both accuracy and precision are paramount.

SVM and Logistic Regression also performed similarly to Naive Bayes in terms of accuracy, each around 70%. These algorithms, which are known for their strong performance in high-dimensional data, struggled in this context due to the complexity of the dataset. Their relatively lower F1-scores imply a less optimal balance between precision and recall, further diminishing their clinical utility for heart attack prediction.

On the other hand, Gradient Boosting provided a moderate performance, with an accuracy close to 80%. Although it lagged behind Random Forest, it demonstrated the potential to improve predictions through iterative learning. K-Nearest Neighbors (KNN), surprisingly, showed better results than expected, with high recall, making it a suitable model for identifying at-risk patients, though its overall balance between precision and recall was weaker compared to Random Forest.

This study underscores the importance of using ensemble methods like Random Forest for heart attack prediction. While simpler models such as Logistic Regression and SVM offer interpretability, their predictive power is limited compared to more sophisticated algorithms. Future research could explore the integration of these models into clinical practice to improve early heart attack diagnosis.

6. Conclusion

This research has demonstrated the efficacy of machine learning models, particularly Random Forest, in predicting heart attacks using a combination of patient data and ECG results. Among the six machine learning algorithms tested, Random Forest emerged as the most accurate, offering a well-balanced performance across precision, recall, and F1-

score metrics. The study underscores the potential of using ensemble methods to enhance diagnostic capabilities in medical contexts. While simpler models like Logistic Regression and SVM offer interpretability, they lack the predictive power exhibited by ensemble techniques such as Random Forest and Gradient Boosting. These findings emphasize the need for advanced machine learning models in clinical practice to improve early detection and prevention of heart attacks. Future work could explore the application of deep learning techniques on larger datasets and investigate the interpretability of these models to ensure their applicability in real-world healthcare scenarios. Overall, this study contributes to the growing body of research focused on leveraging machine learning for cardiovascular risk prediction, ultimately aiming to improve patient outcomes through early intervention.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] M. A. Nasrabadi, Y. Beauregard, and A. Ekhlassi, "The implication of user-generated content in new product development process: A systematic literature review and future research agenda," Technol. Forecast. Soc. Change, vol. 206, no. 123551, p. 123551, Sep. 2024.

[2] T. Islam et al., "Lexicon and deep learning-based approaches in sentiment analysis on short texts," J. Comput. Commun., vol. 12, no. 01, pp. 11–34, 2024.

[3] T. Islam, M. D. A. Hosen, A. Mony, M. D. T. Hasan, I. Jahan, and A. Kundu, "A proposed bi-LSTM method to fake news detection," in 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022.

[4] T. Islam, M. Islam, R. Rudra, I. H. Pranto, M. T. Mahmud, and M. M. Rahman Foysal, "Disaster related tweets analysis with machine learning approaches," in 2023 2nd International Conference on Futuristic Technologies (INCOFT), Belagavi, Karnataka, India, 2023, vol. 4, pp. 1–7.

[5] U. Roy et al., "Enhancing Bangla fake news detection using bidirectional gated recurrent units and deep learning techniques," in Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security, Meknes AA Morocco, 2024, vol. 2020, pp. 1–10.

[6] X. Luo, "Efficient English text classification using selected Machine Learning Techniques," Alex. Eng. J., vol. 60, no. 3, pp. 3401–3409, Jun. 2021.

[7] Z. Y. Algamal and M. H. Lee, "A two-stage sparse logistic regression for optimal gene selection in high-dimensional microarray data classification," Adv. Data Anal. Classif., vol. 13, no. 3, pp. 753–771, Sep. 2019.

[8] M. H. Abd El-Jawad, R. Hodhod, and Y. M. K. Omar, "Sentiment analysis of social media networks using machine learning," in 2018 14th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 2018.

[9] J. Singh, G. Singh, and R. Singh, "Optimization of sentiment analysis using machine learning classifiers," Hum.-centric Comput. Inf. Sci., vol. 7, no. 1, Dec. 2017.

[10] M. Shahzad, C. Freeman, M. Rahimi, and H. Alhoori, "Predicting Facebook sentiments towards research," Natural Language Processing Journal, vol. 3, no. 100010, p. 100010, Jun. 2023.

[11] M. G. Astarkie, B. Bala, G. J. Bharat Kumar, S. Gangone, and Y. Nagesh, "A novel approach for sentiment analysis and opinion mining on social media tweets," in Cognitive Science and Technology, Singapore: Springer Nature Singapore, 2023, pp. 143–151.

[12] A. Sarlan, C. Nadam, and S. Basri, "Twitter sentiment analysis," in Proceedings of the 6th International Conference on Information Technology and Multimedia, Putrajaya, Malaysia, 2014.

[13] Ashish, M. Sharma, and J. Raikwal, "Sentiment analysis of twitter data using machine learning classification algorithms," in Advances in Intelligent Systems and Computing, Singapore: Springer Singapore, 2021, pp. 633–645.

[14] T. Islam et al., "Review analysis of ride-sharing applications using machine learning approaches," in Computational Statistical Methodologies and Modeling for Artificial Intelligence, New York: CRC Press, 2023, pp. 99–122.

[15] T. Bergmanis and S. Goldwater, “Context Sensitive Neural Lemmatization with Lematus,” in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, Louisiana, 2018, pp. 1391–1400.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” J. Artif. Intell. Res., vol. 16, pp. 321–357, Jun. 2002.

[17] M. P. LaValley, “Logistic regression,” Circulation, vol. 117, no. 18, pp. 2395–2399, May 2008.

[18] D. A. Pisner and D. M. Schnyer, “Support vector machine,” in Machine Learning, Elsevier, 2020, pp. 101–121.

[19] T. Islam, A. Kundu, N. Islam Khan, C. Chandra Bonik, F. Akter, and M. Jihadul Islam, “Machine learning approaches to predict breast cancer: Bangladesh perspective,” in Smart Innovation, Systems and Technologies, Singapore: Springer Nature Singapore, 2022, pp. 291–305.

[20] G. Biau and E. Scornet, “A random forest guided tour,” Test (Madr.), vol. 25, no. 2, pp. 197–227, Jun. 2016.

[21] T. Islam, A. Vuyia, M. Hasan, and M. M. Rana, “Cardiovascular disease prediction using machine learning approaches,” in 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2023, vol. 2, pp. 813–819.

[22] F.-J. Yang, “An implementation of naive Bayes classifier,” in 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2018.

[23] T. Chen and C. Guestrin, “XGBoost,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco California USA, 2016.