Cureus Journal of Computer Science

Open Access Original Article

A Weighted-Average Approach Evaluation of Multi-Class Emotion Analysis Model Using Text Mining and Machine Learning

Received 04/28/2025
Review began 05/21/2025
Review ended 06/20/2025
Published 06/20/2025

DOI: https://doi.org/10.7759/s44389-025-04827-3

Subhash C. Gupta [1] , Noopur Goel [1]

1. Department of Computer Applications, V.B.S. Purvanchal University, Jaunpur, IND

Corresponding author: Subhash C. Gupta, csubhashgupta@gmail.com

Abstract

Introduction

Sentiment analysis is a text mining technique used to extract emotions from textual content into binary classes (positive or negative) or multi-class labels (more than two predefined class labels). In this paper, a multiclass sentiment analysis model is proposed to classify text into one of six predefined class labels.

Methods

The model utilizes the "Emotion Dataset for NLP" from Kaggle, comprising 18,000 samples. These samples underwent preprocessing steps including tokenization, stopword removal, and stemming, before being converted into text features using term frequency and inverse document frequency vectorization. The model was implemented in Python 3.9.12 using the scikit-learn library along with other required packages in Jupyter Notebook. It employed seven machine learning classifiers: k-nearest neighbor, multinomial naive Bayes, decision tree, random forest, logistic regression, support vector machine, and gradient boosting.

Result and analysis

Since the dataset is a multiclass imbalanced dataset, the model's performance has been evaluated using macro-averaged and weighted-average scores of precision, recall, F1 score, and class-wise area under the receiver operating characteristic curves (ROC-AUC) instead of only accuracy. The random forest classifier produced the best results, followed by the logistic regression and support vector machine classifiers. The random forest classifier achieved macro-averaged F1-score, precision, and recall of 82.26%, 83.64%, and 81.25%, respectively, based on optimal hyperparameters tuned via cross-validation. Its weighted-average scores were even better, at 85.87%, 86.06%, and 85.97%, respectively. Furthermore, the class-wise ROC-AUC observations showed that the random forest classifier was the best, scoring the highest AUC for all labels, ranging between 0.97 and 0.99.

Categories: Data Science Methodologies, Machine Learning (ML), Natural Language Processing (NLP)
Keywords: multi-class classification, macro averaged and weighted averaged test score, sentiment analysis, tf-idf, nlp

Introduction

As internet access grows, people are spending a lot of time on the web, social media sites, or blogs to share their experiences, reviews, emotions, thoughts, and frustrations. These expressions are written in natural language, generally in text form, and contain valuable information. These are called sentiments and have attracted interest from organizations, companies, and governments due to the hidden information they contain. The process of extracting hidden information from the textual content of sentiments is called sentiment analysis [1,2]. The technique involves analyzing various sources, such as tweets, reviews, and other forms of user-generated content, to extract valuable information about common attitudes and perceptions. With the exponential growth of social media platforms, the number of publicly expressed opinions and viewpoints has increased dramatically, making sentiment analysis an indispensable tool for gauging public opinion in various fields, including business, politics, and more [3]. Sentiment analysis has proven to be an effective predictor of important events, such as movie box office results or election outcomes. The process involves evaluating opinions expressed about specific entities, such as products, services, people, or places, which can be found on various websites and platforms, such as Amazon and Flipkart. These opinions can be broadly categorized as positive, negative, neutral, or other class labels [1]. The main objective of sentiment analysis is to identify the sentiments conveyed in user-generated text content, enabling organizations and decision-makers to understand public sentiment and respond to it effectively [4].

Sentiment analysis is a field of text mining that uses natural language processing (NLP), machine learning, and artificial intelligence to automatically identify and extract hidden information [5]. Text

mining is a method of extracting data from unstructured or semi-structured sources such as text files, web content, emails, social media sites, blogs, and PDFs. It extracts hidden information from text contents using a combination of data mining, information retrieval, NLP, machine learning, statistics, and computational linguistics, and converts the information into a structured format that can be analyzed by machines [6,7]. Some examples of text mining techniques are text summarization, information extraction, information retrieval, part-of-speech (POS) tagging, spam filtration, document classification, and sentiment analysis. NLP is used to interpret, classify, and analyze human language [8-10]. It converts the text or speech from one language to another.

Sentiment analysis is a model used to classify textual content into positive or negative class labels (binary sentiment analysis) or into multiple target labels (multiclass sentiment analysis). Multiclass sentiment analysis is a supervised machine learning technique used to categorize textual content into one or more predefined class labels. It is also applied to determine the polarity of sentiments [11].

Most of the research on sentiment analysis is based on binary classification and uses accuracy as the main metric to evaluate model performance. However, if the number of samples in each class is not equal and the dataset is imbalanced [12], accuracy alone is not a good metric to evaluate the model. It may happen that the model correctly classifies most of the majority class samples but fails to do the same for the minority class. In this situation, the model may have high accuracy due to the majority class accuracy overshadowing the minority class accuracy. In the case of multiclass classification, the situation may be worse because the data samples of all classes are never the same. Therefore, we cannot be choosy in selecting a balanced dataset for the model. However, we can use metrics that are suitable for multiclass imbalanced dataset models. The well-suited performance metrics for multiclass models are macro-averaged or weighted-average precision, recall, F1 score, Matthews correlation coefficient, and area under the receiver operating characteristic curves (ROC-AUC) [13].

The calculation of a "macro-averaged" metrics score is based on the concept of equality, where all classes are treated equally without considering their size. The macro-averaged precision, recall, or F1 score is the mean of their corresponding class-wise metrics scores. In contrast, a weighted-average metrics score considers the size of class-wise samples and gives more weight to larger classes during the calculation of the overall average score [4]. For an imbalanced dataset, the macro-averaged score may be negatively affected by the worse performance of the minority class. Therefore, weighted-average metrics are suitable for imbalanced and multiclass data analysis. Macro-averaged metrics are chosen when the dataset contains multiple target classes and the difference between minority and majority classes is very low. Conversely, when the demand is to prioritize the majority classes or when data imbalance is very high, weighted-average metrics are applied [4,14].

In this study, a machine learning model is proposed to develop a multiclass sentiment analysis model using seven classifiers: random forest, naive Bayes, logistic regression, k-nearest neighbor (k-NN), decision tree, support vector machine (SVM), and gradient boosting classifiers. The model uses "The Emotion Dataset for NLP", which was downloaded from the public website Kaggle [15]. The model has been evaluated using weighted-average and macro-averaged metric scores as well as the ROC-AUC.

Literature review

Wankhade et al. [1] discussed sentiment analysis, its categorization, and compared its advantages and disadvantages. Subramani et al. [2] proposed a sentiment analysis model based on NLP and machine learning classifiers, including random forest, SVM, and decision tree. Accuracy was considered as the evaluation metric for the model, and the highest accuracy of 90% was achieved by the random forest algorithm.

Suryawanshi [3] conducted a survey on machine learning sentiment analysis models. He noted some recent developments in machine learning and further pointed out its limitations. Furthermore, he made a comparison between machine learning and deep learning. Haque et al. [4] developed a model to perform multiclass sentiment analysis of reviews written in Bengali text. The model is based on deep learning concepts and uses supervised deep learning classifiers based on convolutional neural network and long short-term memory to conduct multiclass sentiment analysis. They achieved an accuracy level of 85.6% with their model.

Verma and Undavia [6] proposed a Python-based machine learning model to rate customer feedback into class labels such as “Happy”, “Satisfied”, or “Not Satisfied”. Shen [11] reviewed past research related to sentiment analysis. In their paper, they focused on the major methods used to analyze sentiments. They discussed the major preprocessing steps of raw data, made a comparative analysis between them, and further discussed different feature extraction methods.

Archana and Velmurugan [12] implemented a machine learning-based sentiment analysis model using naive Bayes, SVM, random forest, and decision tree. The objective of their study was to analyze and classify the reviews of electronic product customers collected from Flipkart. They achieved an accuracy

level of 89.5% with their model.

Materials And Methods

The multiclass sentiment analysis model is implemented in Python using Jupyter Notebook, with the help of essential packages such as Scikit-learn, Matplotlib, Seaborn, and Pandas. The model is based on "The Emotion Dataset for NLP" [15], which was downloaded from the public data source Kaggle. The dataset contains 18,000 samples and two features: one is "Text", representing content written in English, and the other is the target class. The target class is a label that represents one of six predefined emotions: "surprise", "love", "joy", "fear", "anger", and "sadness" [15]. The text class labels, "surprise", "love", "joy", "fear", "anger", and "sadness", are mapped into numeric class labels "1", "2", "3", "4", "5", and "6", respectively, to make calculation easy. The numbers of samples available of each class are shown by a histogram in Figure *1*.

The first step is the preprocessing of the emotion dataset to make it ready for machine learning classifiers. The processes involved in preprocessing are tokenization, removal of digits, special characters, symbols, stopwords, duplicate items, stemming, lemmatization, and POS tagging. Stemming, POS tagging, lemmatization, and stopword handling belong to NLP [8]. Stopwords are words that appear frequently but add no meaning, such as "do", "did not", and "the". Tokenization is the process of breaking a sentence into words [11]. Sometimes a word appears many times in different forms but belongs to a single root word, such as "play", "played", and "playing". Stemming is the process of identifying such words and finding their root meaning [16,17]. Similarly, a single word may appear in many locations but have different POS. POS tagging identifies such words and tags them as different POS [4].

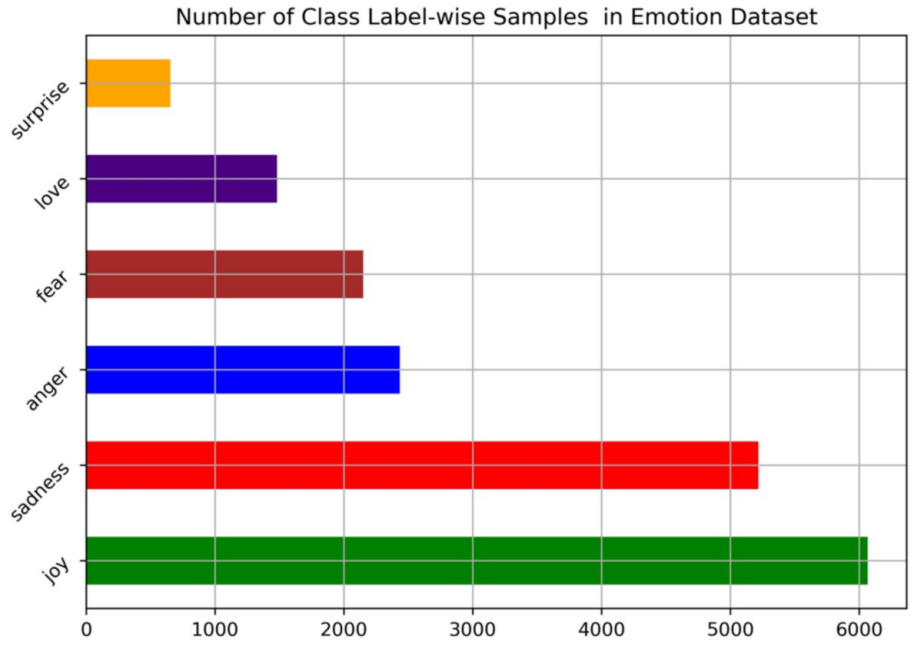

FIGURE 1: Number of samples of each class of multi-class emotion dataset

Feature extraction is the next step of sentiment analysis and is used to extract important features to select significant text and reduce the dimensionality of the dataset [13]. Feature extraction improves classification metrics scores, optimizes utility, and leads to more impactful and meaningful outcomes. It involves converting text into numerical vectors, mostly done using Term-Frequency-Inverse Document Frequency (TF-IDF) [11]. This concept is derived from language modeling theory and is used to calculate the importance of a word in documents, categorizing them based on their importance. The method is based on two terms: Term Frequency (TF) and Inverse Document Frequency (IDF). Term Frequency focuses on the number of times a word appears in a text or document, while Inverse Document Frequency measures its inverse document frequency and provides weight to a word concerning the documents [4,13]. In classification tasks, it helps identify the target class for a word or set of words.

The dataset description and working methodology of the model are shown in Figure *2*.

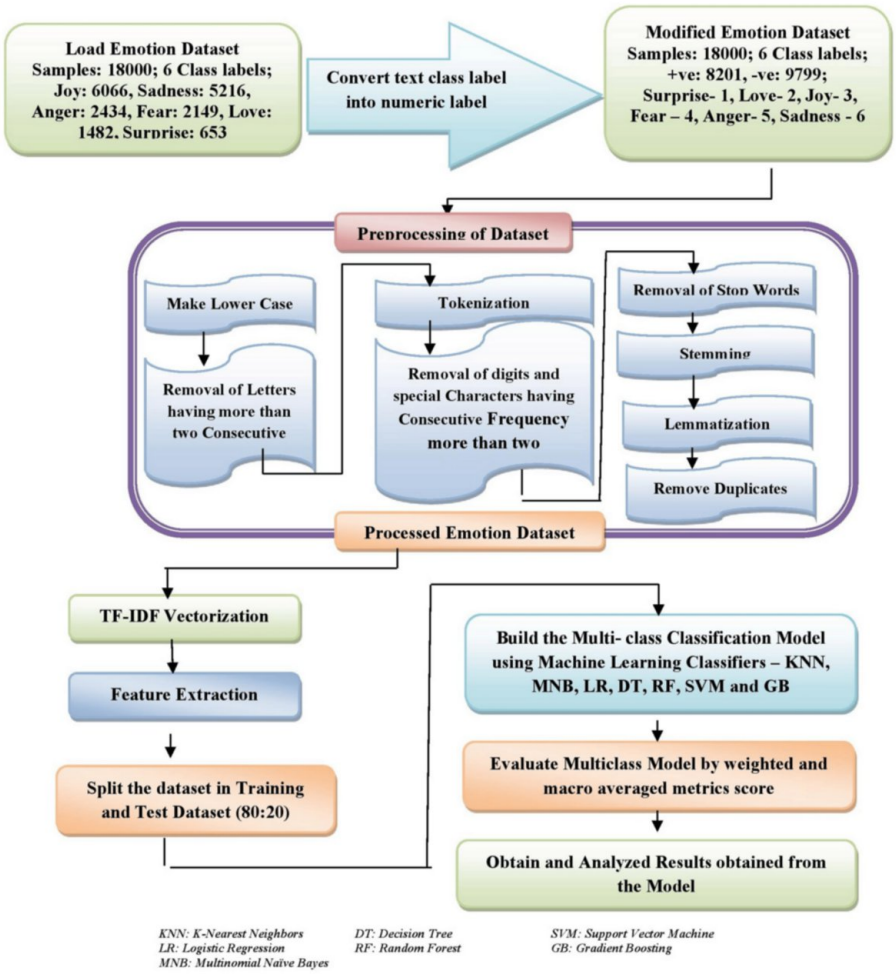

FIGURE 2: Dataset description and working methodology of the model

The dataset is divided into a 70:30 ratio for training and testing. The model is built and trained on the training dataset, while the testing dataset is used to evaluate the model. The model uses the training dataset to learn patterns and trends in the words (dataset) and updates itself accordingly. The model is built using random forest, naive Bayes, logistic regression, decision tree, k-NN, SVM, and gradient boosting classifiers of machine learning. It has been evaluated on unseen data from the test dataset and classifies them into a predefined set of class labels. The evaluation metrics suitable for multiclass classification, such as weighted-average and macro-average precision, recall, and F1 score, were used. The model has also been evaluated on class-wise ROC-AUC and overall accuracy [11].

Results

The model has been evaluated on a number of metrics as mentioned in the previous section. Figure *3* shows the overall accuracy of all classifiers for all class labels. It demonstrates that the model's accuracy ranges from 0.6697 to 0.8597, with the maximum accuracy of 0.8597 (85.97%) achieved by the random forest classifier and the minimum accuracy of 0.6697 (66.97%) by the multinomial naive Bayes classifier. However, as mentioned in the introductory part of the paper, accuracy alone is not a good metric for imbalanced and multiclass datasets. Therefore, other metrics such as macro-averaged and weighted-average F1 score, precision, and recall are also used to evaluate the model.

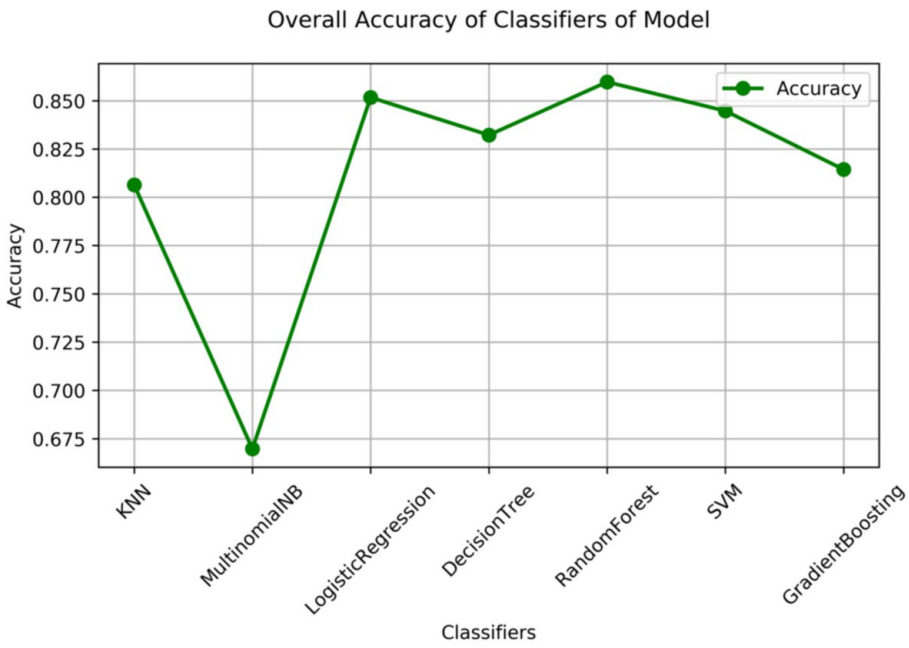

FIGURE 3: Accuracy of classifiers of multiclass sentiment analysis model

k-NN, k-Nearest Neighbor; MultinomialNB, Multinomial Naive Bayes; SVM, Support Vector Machine

Figure *4* shows the classifier's performance class-wise and plots the F1 score, precision, and recall for each class label using Matplotlib in Python. It contains six subplots, each drawn for an individual class label and obtained from the classification reports of each classifier, showing label-wise precision, recall, and F1 score. The observations indicate that for the "surprise" class label, the maximum precision, recall, and F1 scores are achieved by multinomial naive Bayes, gradient boosting, and gradient boosting, respectively; for the "love" class label, they are multinomial naive Bayes, random forest, and random forest; and for the "sadness" class label, they are gradient boosting, logistic regression, and logistic regression.

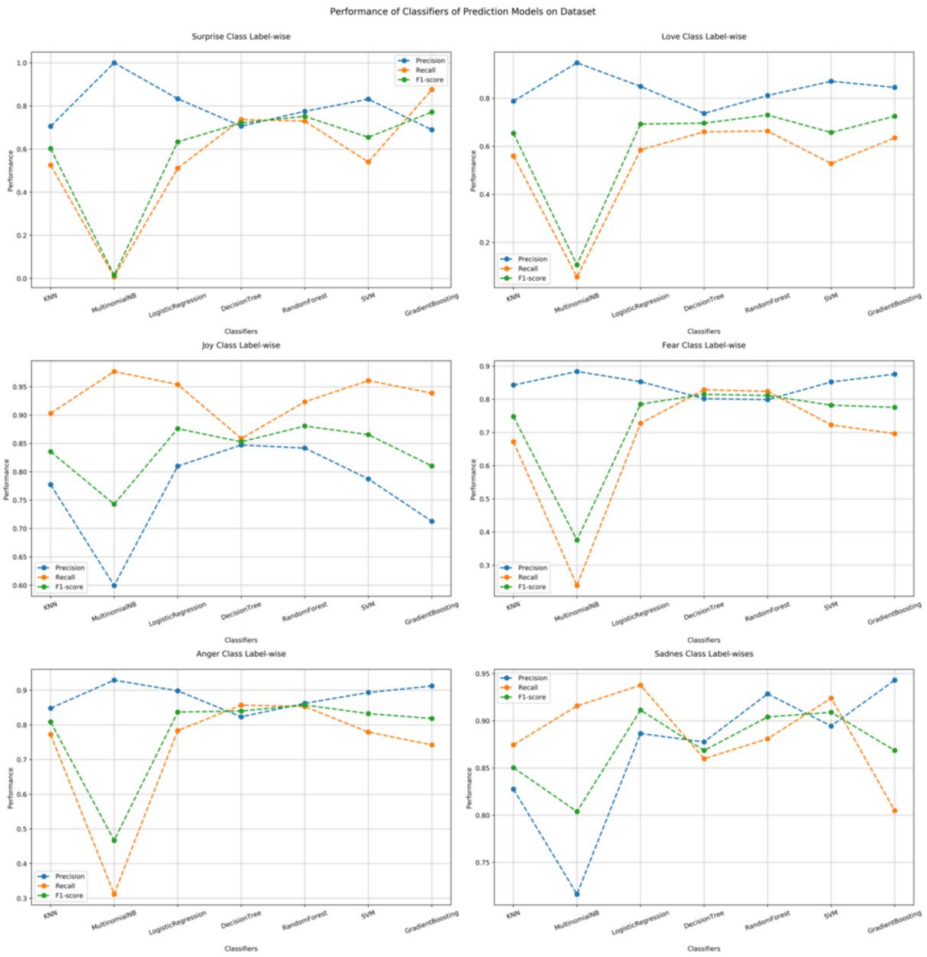

FIGURE 4: Class-wise performance of the classifiers of multiclass sentiment analysis model

k-NN, k-Nearest Neighbor; MultinomialNB, Multinomial Naive Bayes; SVM, Support Vector Machine

Discussion

As mentioned, for multiclass classification, accuracy is not a perfect metric; weighted-average and macro-averaged metric scores are better suited. In Figure *5*, three subplots are drawn, showing comparative line charts of weighted-average and macro-average precision, recall, and F1 score, respectively. The first subplot presents data on the precision of classifiers. It shows that the maximum and minimum macro-average precision scores are 0.8551 (logistic regression) and 0.7975 (decision tree), while for weighted-average precision, they are 0.8606 (random forest) and 0.7559 (multinomial naive Bayes).

Given the multiclass and imbalanced nature of our dataset, weighted-average metrics were chosen to prioritize performance across all classes; accordingly, random forest emerged as the top-performing classifier in terms of precision. The second plot is drawn to show the recall (sensitivity) of classifier calculated by macro-average and weighted average. It again demonstrates that random forest performs better than others, with scores of 0.8125 and 0.8597 for macro-averaged and weighted-averaged recall, respectively. The third plot depicts the F1 score for the same type, and once again, random forest outperforms the others, scoring 0.8226 and 0.8587 for macro-precisions and weighted-average precision, respectively. The figure concludes that weighted-average scores are generally higher compared to their macro-average scores across all classifiers. Random forest is observed to be the best classifier for the multiclass model.

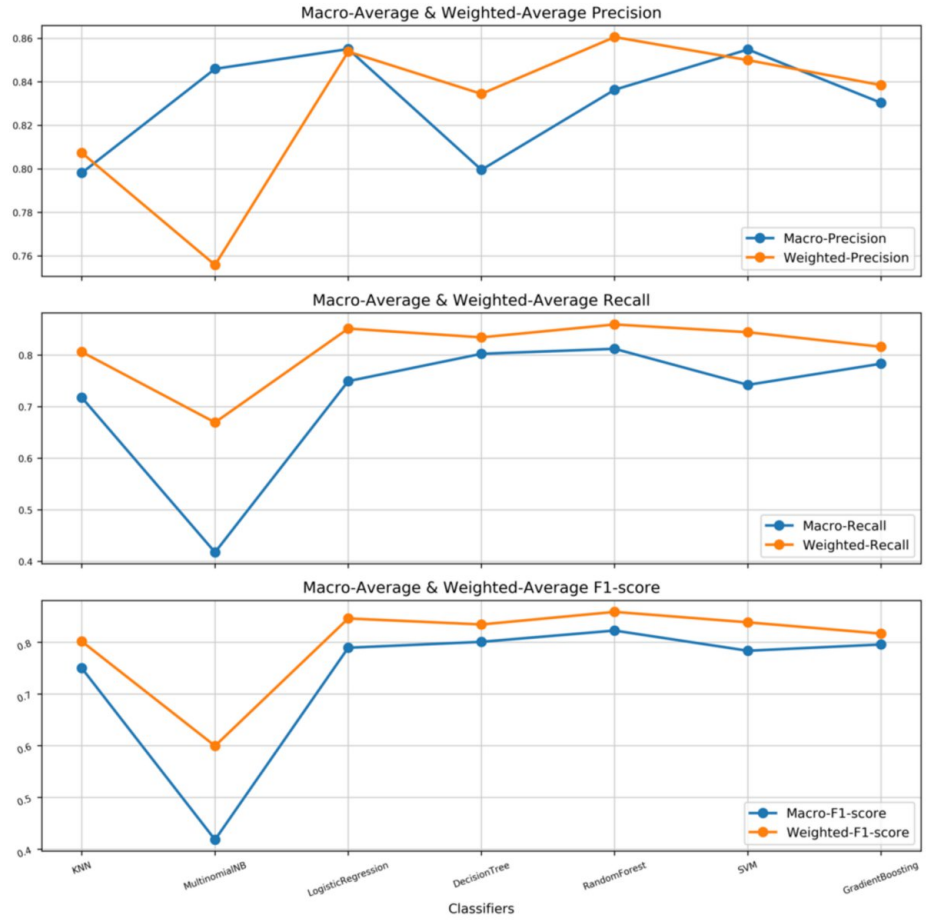

FIGURE 5: The macro-averaged and weighted-averaged performance of the classifiers of multiclass sentiment analysis model

k-NN, k-Nearest Neighbor; MultinomialNB, Multinomial Naive Bayes; SVM, Support Vector Machine

A confusion matrix is a tabular representation of TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) of actual and predicted class labels. For binary confusion matrices, it is of size 2 × 2, while a multiclass confusion matrix is of size N × N, where N is the number of distinct class labels. The diagonal elements show the number of correct positive classifications. The multiclass confusion matrix of the second and third best performing classifiers, logistic regression and SVM, is shown in Figure *6*.

The AUC-ROC curve is a visualization technique for the performance of a classification model. It is one of the best performance metrics for the model at various thresholds, especially when the model is based on an imbalanced dataset. It plots the FP rate on the x-axis and the TP rate on the y-axis, with the curve ranging between 0 and 1. The upper left corner of the ROC curve shows the best result for a classifier. The AUC value is the area under the ROC curve; the more area covered under the curve by AUC, the better the separation between the classes. Generally, the AUC-ROC curve is applied to binary classification problems, but it can also be used for multiclass problems by considering the TPs of a class as the positive target class while others are considered negative target classes [18-21].

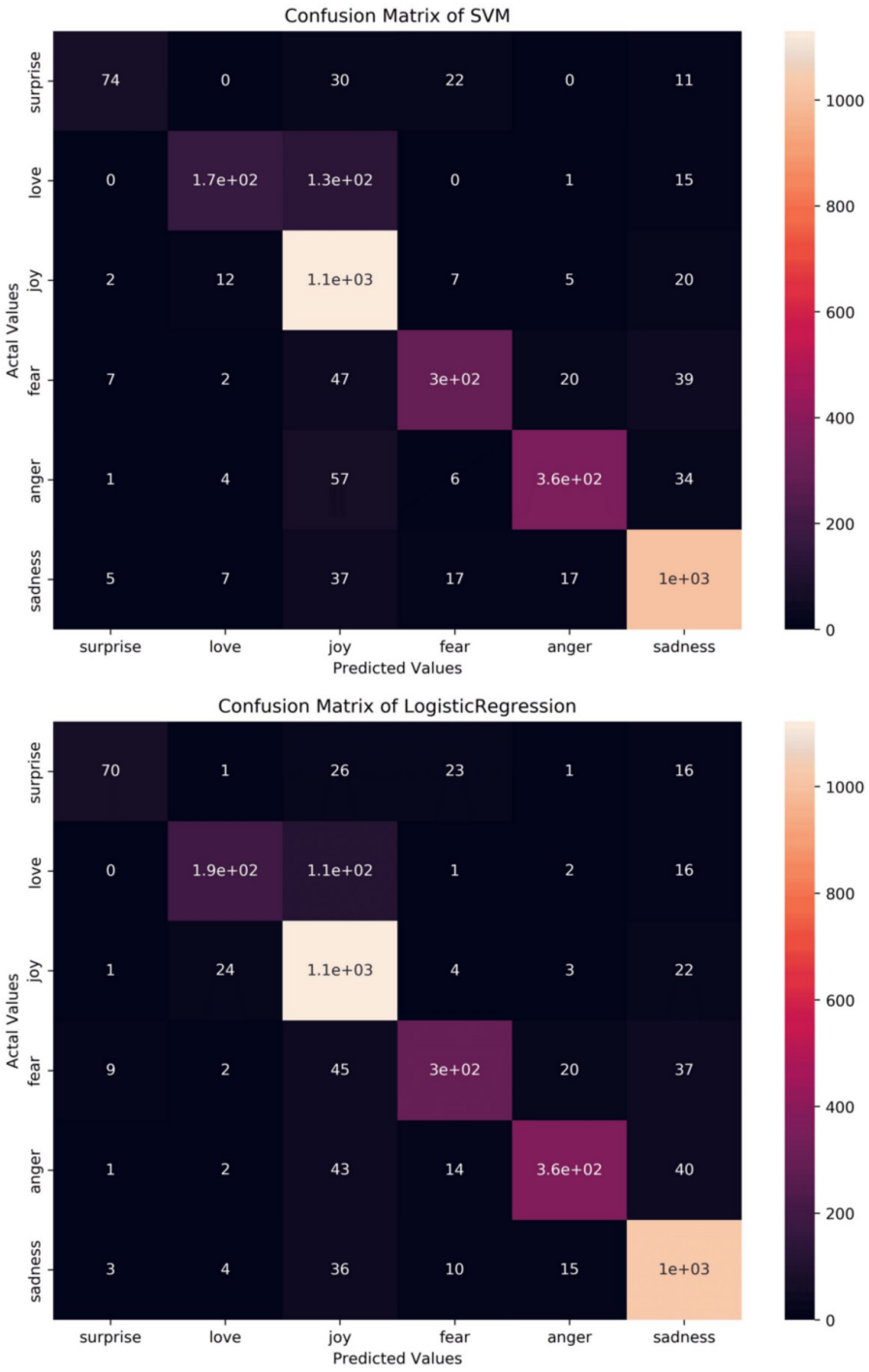

FIGURE 6: Confusion matrix of logistic regression and SVM classifiers of multiclass sentiment analysis model

SVM, Support Vector Machine

For each classifier, we calculated the AUC-ROC curve using the one-vs-rest approach for multiclass classification, plotting the TP rate against the FP rate for each class. The calculated AUC-ROC curve is shown in Figure 7. Each AUC-ROC curve plots one classifier's performance for each individual class. The plot also shows the AUC score of the classifier for every class. If we observe the AUC-ROC curve of all classifiers, logistic regression is the best performing classifier, with random forest being the second best and very close to logistic regression. Logistic regression, random forest, and k-NN classifiers demonstrated superior performance, with AUC scores ranging from 0.95 to 0.99 across all class labels, indicating excellent class separation capabilities. If we consider the AUC class-wise, the top two classifiers are logistic regression and random forest for all "surprise", "love", "joy", "fear", "anger", and "sadness" class

labels.

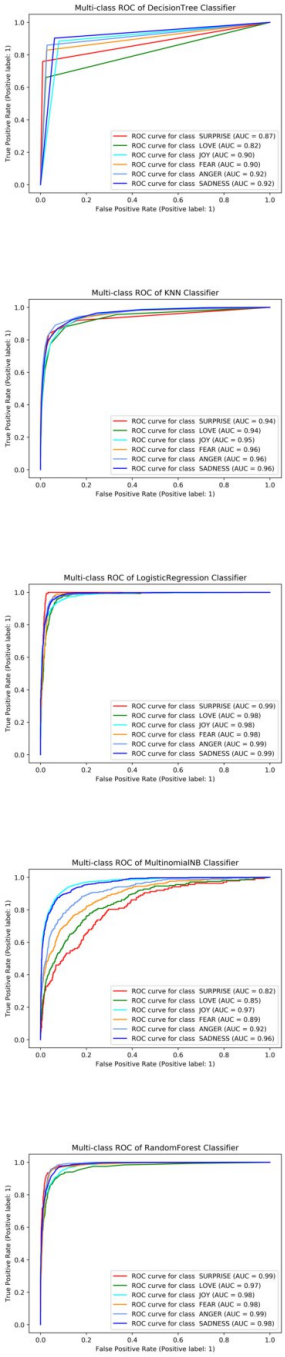

FIGURE 7: The ROC-AUC chart of classifiers of multiclass sentiment analysis model showing their AUC score for each class label

AUC, Area Under the Curve; ROC, Receiver Operating Characteristic; k-NN, k-Nearest Neighbor; MultinomialNB, Multinomial Naive Bayes

Conclusions

The machine learning multiclass sentiment analysis model discussed in the paper demonstrated its performance on six different class labels of the "Emotion Dataset for NLP" and was evaluated using the most suitable metrics, such as weighted-average precision, recall, and F1 score, as well as macro-averaged precision, recall, and F1 score. Random forest is the best classifier of the model for both when evaluated

using either weighted-average or macro-average metrics scores. Its weighted-average scores and macro-average scores are (0.8606, 0.8597, and 0.8587) and (0.8364, 0.8125, and 0.8226), respectively, for precision, recall, and F1 score. The performance of random forest is followed by logistic regression and SVM classifiers. Performance is also evaluated using the multiclass AUC-ROC curve, where logistic regression wins by a very small margin over random forest for all class labels. Both scored AUCs between 0.97 and 0.98 for different class labels.

The multiclass classification model has been applied to an imbalanced dataset, which may negatively impact its performance. In the future, the same model will be applied by balancing the dataset using some oversampling or undersampling algorithms, such as Synthetic Minority Oversampling Technique. The results of the model will also be validated and optimized using cross-validated hyperparameter-tuned classifiers.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Subhash C. Gupta, Noopur Goel

Acquisition, analysis, or interpretation of data: Subhash C. Gupta, Noopur Goel

Drafting of the manuscript: Subhash C. Gupta

Critical review of the manuscript for important intellectual content: Subhash C. Gupta, Noopur Goel

Supervision: Noopur Goel

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

References

1. Wankhade M, Rao ACS, Kulkarni C: A survey on sentiment analysis methods, applications, and challenges. Artificial Intelligence Review. 2022, 55:5731-5780. 10.1007/s10462-022-10144-1
2. Subramani K, Nivedha U, Jabasheela L, Divya S, Dhuneesha E: Secured text-based emotion classification using machine learning with NLP. Educational Administration: Theory and Practice. 2024, 30:901-910.
3. Suryawanshi NS: Sentiment analysis with machine learning and deep learning: A survey of techniques and applications. International Journal of Science and Research Archive. 2024, 12:5-15. 10.30574/ijsra.2024.12.2.1205
4. Haque R, Islam N, Tasneem M, Das AK: Multi-class sentiment classification on Bengali social media comments using machine learning. International Journal of Cognitive Computing in Engineering. 2021, 4:21-35. 10.1016/j.ijcce.2023.01.001
5. Swati R, Sangeet S, Barkha B, Gupta G: Sentiment analysis using text mining: a review. International Journal on Data Science and Technology. 2018, 4:49-53. 10.11648/j.ijdst.20180402.12
6. Verma J, Undavia JN: Sentiment analysis through text mining to rate the e-commerce products based on the customer feedback. Journal of Emerging Technologies and Innovative Research. 2022, 9:579-585.
7. Gupta SC, Goel N: Emotion analysis of textual contents using natural language processing and text mining. Journal of Theoretical and Applied Information Technology. 2024, 102:7998-8012.
8. Jyothi C, Devi BK: Hotel reviews on sentiment analysis for using NLP algorithms. Journal of Critical Reviews. 2016, 3:115-120.
9. Boorugu R, Ramesh G: A survey on NLP based text summarization for summarizing product reviews. 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India. 2020, 352-356. 10.1109/ICIRCA48905.2020.9183355
10. Raut SP, Chaudhari SD, Waghole VB, Jadhav PU, Saste AB: Automatic evaluation of descriptive answers using NLP and machine learning. International Journal of Advanced Research in Science Communication and Technology. 2022, 2:735-745. 10.48175/ijarsct-3030
11. Shen X: Sentiment analysis of modern Chinese literature based on deep learning. Journal of Electrical Systems. 2024, 20:1565-1574. 10.52783/jes.3075
12. Archana M, Velmurugan T: Enhancing sentiment analysis in electronic product reviews using machine learning algorithms. International Journal of Computer Information Systems and Industrial Management

Applications. 2024, 16:547-565.
13. Semary NA, Ahmed W, Amin K, Pławiak P, Hammad M: Enhancing machine learning-based sentiment analysis through feature extraction techniques. PLoS ONE. 2024, 19:e0294968. 10.1371/journal.pone.0294968
14. Ho E, Schneider M, Somanath S, Yu Y, Thuvander L: Sentiment and semantic analysis: Urban quality inference using machine learning algorithms. iScience. 2024, 27:110192. 10.1016/j.isci.2024.110192
15. Kaggle: Emotions dataset for NLP. https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp.
16. Delmonte R, Pallotta V: Opinion mining and sentiment analysis need text understanding. Advances in Distributed Agent-Based Retrieval Tools. Pallotta V, Soro A, Vargiu E (ed): Springer, Berlin, Heidelberg; 2011. 361:81-95. 10.1007/978-3-642-21384-7_6
17. Abdullah EF, Alasadi SA, Al-Joda AA: Text mining based sentiment analysis using a novel deep learning approach. The International Journal of Nonlinear Analysis and Applications. 2021, 12:595-604.
18. Scikit-learn: Multiclass Receiver Operating Characteristic (ROC). Accessed: February 15, 2025: https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html.
19. Trevisan V: Interpreting ROC Curve and ROC AUC for Classification Evaluation. (2022). Accessed: February 15, 2025: https://towardsdatascience.com/interpreting-roc-curve-and-roc-auc-for-classification-evaluation-28ec3983f077/.
20. Scikit-learn: roc_auc_score. Accessed: February 15, 2025: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html.
21. Bhandari A: Guide to AUC ROC Curve in Machine Learning. (2024). Accessed: February 15, 2025: https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/..