An Empirical Analysis of Twitter Reviews Using Optimal Machine Learning Techniques

Rohini G. Khalkar
College of Engineering, Bharati Vidyapeeth (Deemed to be University), Pune, India
rgkhalkar@bvucoep.edu.in (corresponding author)

Mrunal S. Bewoor
College of Engineering, Bharati Vidyapeeth (Deemed to be University), Pune, India
msbewoor@bvucoep.edu.in

Sampat P. Medhane
College of Engineering, Bharati Vidyapeeth (Deemed to be University), Pune, India
spmedhane@bvucoep.edu.in

Received: 8 May 2025 | Revised: 12 June 2025, 2 July 2025, 10 July 2025, 15 July 2025, and 17 July 2025 | Accepted: 19 July 2025

 DOI: https://doi.org/10.48084/etasr.12011

ABSTRACT

Sentiment analysis, a key method in Natural Language Processing (NLP), is used to recognize, extract, and classify subjective material from textual data. With the rapid development of user-generated content on social media platforms, such as Twitter, Facebook, and Instagram, sentiment analysis has gained significant importance. Social media platforms have proven to be useful for gauging customer reviews and brand perception to determine sentiments and thoughts on a variety of topics. Analyzing such data presents challenges due to the brevity of posts, informal syntax, slang, and the heavy use of emojis. This paper proposes a hybrid sentiment analysis framework that integrates textual features with emoji-based sentiment cues. A Support Vector Machine (SVM) classifier is employed along with a balancing factor to ensure vigorous classification. Support Vector Machine (SVM) is the most accurate approach for sentiment classification. Tailored preprocessing significantly improves classification accuracy by comparing traditional and Twitter-specific preprocessing methods. Feature extraction is carried out using Term Frequency-Inverse Document Frequency (TF-IDF), which captures the contextual importance of words. In addition, emoji sentiment mapping assigns predefined sentiment scores to frequently used emojis, thereby enhancing semantic understanding and enlightening the accuracy of sentiment predictions. This study demonstrates that it is important to select appropriate preprocessing and classification techniques for accurate sentiment analysis in a social media context. SVM, Logistic Regression (LR), and Naïve Bayes (NB) classifiers were applied on the sentiment140 dataset, focusing on a sentiment analysis of more than 1.5 million tweets. SVM was found to be more effective when working with a balancing factor that considers the standard deviation.

Keywords-sentiment analysis; support vector machine; classification; opinion mining; logistic regression; standard deviation

I. INTRODUCTION

Social networks are important in communicating people's views. Twitter (now X), a well-known microblog, contains a wealth of rich data in the form of user-entered messages, reviews, and feedback. Processing these formal and informal sentences requires specific methods that can make comprehension and analysis easier. It is difficult to understand these natural language sentences to achieve the intended outcome. By classifying sentiments, such as positive, negative, and neutral, researchers and businesses can gain invaluable insight into public opinions, inform business strategies, and improve customer experiences [1, 2]. The utility of sentiment analysis extends to various digital arenas, from social networks to e-commerce, where it serves as a tool to navigate the complexities of user-generated content [3, 4].

II. LITERATURE REVIEW

Developments in machine learning algorithms, including Support Vector Machines (SVM) and Logistic Regression (LR), have significantly strengthened the success of SA. These methods help to accurately classify the data and adapt to the nuances of language used on diverse platforms, such as Instagram, Facebook, and Twitter [1, 2]. Comparison of SVM and LR highlights the strengths and limitations of each method, which requires ongoing research to improve their predictive competencies [2]. Sentiment analysis is instrumental in the extraction of actionable meaning from data, helping to guide decision-making processes that are crucial to maintaining online reputation and understanding market trends [3]. Social network platforms offer millions of posts daily and provide a rich and real-time source of data. Sentiment analysis on such data can reveal patterns in public opinion, detect emerging trends, and even provide early warnings of crises or shifts in societal attitudes.

This study focuses on the Sentiment140 dataset, which is a widely used benchmark that contains 1.6 million labeled tweets [5], to compare and analyze the performance of popular machine learning algorithms in classifying sentiments. These algorithms represent a variety of methods, from the geometric decision boundaries of SVM to the probabilistic modeling of Naïve Bayes (NB), and the linear separability of LR. This study was motivated by the need to identify the most efficient machine learning method, which addresses not only performance metrics such as accuracy, precision, recall, and the F1 score, but also computational cost. The findings can provide practical insights for businesses and researchers seeking to implement sentiment analysis systems in diverse contexts. In addition, this study explores the wide-ranging implications of sentiment analysis, including its potential to influence marketing strategies, predict election results, and improve customer engagement. This study contributes to a deeper understanding of how sentiment analysis can be used effectively and its impact in various domains, paving the way for future innovations in this critical area of data science.

Traditional sentiment analysis practices depend on straightforward lexicon-based approaches, where arguments were tagged as positive or negative within a predetermined vocabulary [1]. With time, advances in machine learning have led to more sophisticated sentiment analysis approaches, allowing more nuanced understanding of text data through Natural Language Processing (NLP) [6]. The study in [7] examined tweets from major UK cities, achieving accuracy of 0.7, 0.63, and 0.71, respectively, using RF, Multinomial NB (MNB), and SVM. In [4], CNN and LSTM were used for sentiment in 100 million distinct unlabeled English tweets. In [8], a variety of techniques were used, including NB, maximum entropy, Decision Tree (DT), RF, XGBoost, SVM, Multi-Layer Perceptron (MLP), CNN, and RNN, to extract sentiment from tweets, achieving an overall accuracy of 83.58%. In [9], a graph-based optimization model achieved 68.3% accuracy. In [10], an impressive accuracy of 98% was reported for the SVM linear model on 520K tweets. In [11], NB was used to analyze Twitter posts about electronic products. In [12], accuracies of 88.2%, 83.8%, 85.5%, and 89.9% were achieved on sentiment classification on Twitter using NB, maximum entropy, SVM, and semantic analysis (WordNet), respectively.

Sentiment analysis has a wide variety of applications, from monitoring political campaigns to market analysis [1, 2]. In consumer markets, sentiment analysis helps companies determine the reactions of the public to goods and services, frequently looking at comments on social media and reviews on the Internet to modify marketing tactics and control brand image, or even reveal trends in public opinion and potentially predict election results in the political sphere [13-21]. Deep learning techniques are among the most recent methodological advances in SA. By efficiently recording context-specific subtleties that impact sentiment interpretation, they outperform traditional techniques in terms of accuracy. However, issues such as accurately recognizing contextually dependent views, humor, and sarcasm elements that automated systems usually misunderstand must be resolved.

III. METHODOLOGY

This study aimed to investigate the sentiment analysis in a large set of tweets. The main objective of sentiment analysis is to determine whether the text data reflects positive, negative, somewhat positive, slightly negative, or neutral sentiment. Studying opinion on social networks can inform businesses, governments, and academics about public views and responses to a range of topics [13-21]. Using the Sentiment140 dataset [5], this study evaluates how different machine learning approaches work, using F1 score, precision, and recall [2]. This dataset is well-suited for sentiment analysis, as it includes labelled data indicating whether tweets are positive, negative, or neutral. There are different types of datasets available for sentiment analysis. The chosen one includes emojis and event-related tweets that could help the analysis in specific contexts.

A. Data Loading and Preprocessing

This study utilized Python libraries, such as pandas and numpy, which are frequently used for reliable information manipulation. The dataset was preprocessed to ensure it was clean and ready for machine learning analysis. Tweets were converted to lowercase to ensure that they were consistent across text data. Noise such as URLs, special characters, numbers, unnecessary punctuation, mentions (@username), and hashtags (#topic) was removed using regular expressions.

Tokenization was performed using NLTK's word_tokenize, separating the text into individual words for further processing. Stopwords, which are common words that do not appreciably contribute to sentiment analysis, e.g., is, the, and, were removed using the predefined NLTK stopwords list. Finally, stemming was applied using the Porter Stemmer to reduce words to their base forms (e.g., "running" became "run") to normalize variations of the same word.

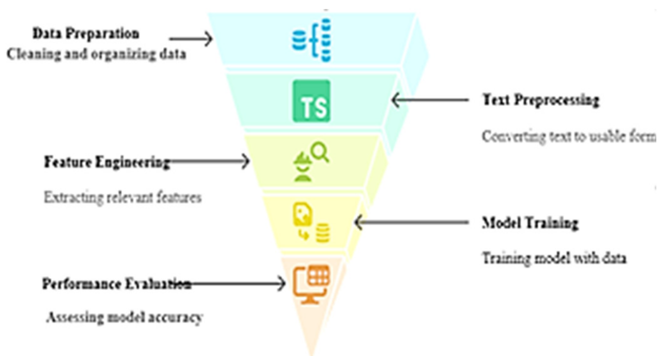

Fig. 1. Process of review analysis.

B. Feature Engineering

The Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer was employed to convert textual data into a machine-readable format. This approach works effectively with assigned weights to words, emphasizing those of advanced importance while reducing the consequence of common words across the dataset. Additional numeric features were considered. The polarity of each word was calculated using TextBlob, a Python library for NLP, assembled for every tweet/review to find a *polarity_sum* feature. This provides an additional layer of sentiment in the text. By adding TF-IDF features to the *polarity_sum*, the feature set ensured that both textual and numeric representations of the data are operated on, potentially increasing model performance.

C. Data Splitting/Partitioning

The dataset was divided into 70% training data and 30% testing data using sklearn's train_test_split. This partition allowed the model to be trained on the majority of the data while being tested on unseen examples to authenticate its simplification capability. The split was stratified to maintain the distribution of target labels across training and testing datasets, ensuring balanced representation.

D. Numeric Feature Integration

The numeric feature (*polarity_sum*) was standardized using the standard scaler to bring it onto a comparable scale with other features. Standardization is a significant step to avoid any single feature disproportionately influencing the model's training. These scaled numeric features were then united with the TF-IDF textual geographies using scipy's hstack, creating a unified feature matrix for both the training and testing datasets.

IV. ALGORITHMS

This study selected only six features from the dataset [5], namely target, id, date, flag, user, and text, for further implementation and analysis. Three machine learning algorithms were implemented to analyze the dataset.

A. Support Vector Machine (SVM)

SVM is a powerful and commonly used supervised learning algorithm for classification tasks. SVM aims to find a hyperplane that best splits the data into distinct classes by making best use of the margin between them. In the context of sentiment analysis, SVM classifies tweets into positive and negative sentiment classes based on the features extracted. The key advantage of SVM is its ability to perform well even with multidimensional data, which is common with text data, and handle non-uniform boundaries with the help of the kernel. SVM aims to discover the hyperplane that exploits the boundary between two classes. The optimization objective is:

$$\max\left(\frac{2}{\|w\|}\right), \text{ subject to } y_i(w \cdot x_i + b) \geq 1, \ \forall i \quad (1)$$

where w is the weight vector, x_i is the feature vector, y_i is the class label (+1 or -1), and b is the bias term. The aim is to maximize $\frac{2}{\|w\|}$ as it represents the margin that is the distance between the separating hyperplane and the nearest data points (support vectors). Maximizing this aims to find the widest possible margin to correctly classify data.

$y_i(w \cdot x_i + b) \geq 1$ ensures that every data point x_i is correctly classified and is at least one unit away from the decision boundary. SVM achieved a training accuracy of 94.51% and a testing accuracy of 71.46%

B. Logistic Regression (LR)

LR is an algebraic method used for binary classification. It predicts the probability of a binary outcome (such as positive or negative sentiment) based on the input features. The algorithm uses the logistic function (sigmoid function) to convert linear outputs into probabilities [14]. LR is known for its simplicity and efficiency, making it a reasonable choice for sentiment analysis. It works well for linearly separable data, but may struggle with non-linear relationships unless extended to more complex models, such as multinomial LR. The formula for LR is:

$$P(y = 1 \mid X) = \frac{1}{1+e^{(\beta_0+\beta_1 x_1+\cdots+\beta_n x_n)}} \quad (2)$$

where $\beta_0, \beta_1, \ldots, \beta_n$ are coefficients, $x_1, x_2, \ldots, x_n$ are feature values, and y is the outcome.

Multinomial LR is a statistical technique for implementing the probabilities of various categorical outcomes. Binary LR estimates outcomes based on specific input features and examines correlations with effect likelihoods using a linear mixture of independent variables. LR achieved a training accuracy of 81.01% and a testing accuracy of 77.80%. Multinomial LR achieved a training accuracy of 93.15 % and a testing one of 69.26%

C. Naive Bayes (NB)

The NB algorithm is a classification process grounded on Bayes' theorem. This algorithm assumes that the features are independent of each other and estimates the possibility of a sample belonging to a specific class based on the probabilities of its features. In Bayesian classification, the main aim is to find the subsequent prospects, i.e., the probability of a label given some observed features, $P(L|features)$. With the help of Bayes' theorem, this can be expressed in a quantitative form as follows [1]:

$$P(L|features) = P(L)P(features|L)/P(features) \quad (3)$$

where $P(L|features)$ is the posterior probability of a class, $P(L)$ is the prior probability of a class, $P(features|L)$ is the likelihood of a predictor given class, and $P(features)$ is the prior probability of a predictor. Here, NB estimates the possibility of each feature of the example given the class and multiplies them to get the possibility of the sample belonging to the class. Then multiply the likelihood is multiplied by the prior probability of the class to get the posterior probability of the sample belonging to the class. This process is repeated for each class, and the class with the maximum possibility is selected.

In MNB, the model finds out how many times each word appears in messages from different classes. Gaussian Naive Bayes (GNB) is a prevailing classification algorithm based on Bayes' theorem, assuming that features are distributed according to a Gaussian distribution and are conditionally independent given the class label.

D. SVM with Balancing Factor (Optimal Model)

TF-IDF has two components: Term Frequency (TF) and Inverse Document Frequency (IDF). TF shows how frequently a word appears in a document/text, with a higher frequency indicating greater importance. If a term looks frequently in a document, it is likely pertinent to its content [2]. For IDF, if a term appears rarely in documents, it is more likely to be expressive and precise.

$$TF - IDF\ (t,d)\ = TF(t,d) * IDF(t) \quad (4)$$

where $TF(t,d)$ denotes the number of occurrences in d, and $IDF(t)$ is the inverse occurrence of t in all documents.

$$IDF(t) = N/DF(t) = N/N\ (t)$$

Simple TF-IDF does not justify the widespread position of a term across the whole corpus. Common words, such as "the" or "and", may have high term frequency scores but are not expressive in distinguishing documents. The modified TF-IDF formula is given by:

$$\text{Modified } TF - IDF = TF - IDF\ +\ z =$$

$$= (TF(t,d) * IDF(t)) + (\frac{X-\mu}{\sigma}) \quad (5)$$

where z is a balancing factor, X is the value of the element, μ is the population mean, and σ is the standard deviation of the population.

Thus, a balancing factor is incorporated with TF-IDF by finding the standard deviation as a balancing factor into TF-IDF, which helps in finding the correct polarity and class of a tweet. The SVM model with the balancing factor achieved training and testing accuracy of 94.97% and 80.4%, which was greater than the respective LR model.

E. Challenges Faced During Preprocessing the Twitter Dataset

Finding negative speech on Twitter through SA is fraught with several challenges, such as data noise, context sensitivity, labelling complexities, class imbalance, content in multiple languages, slang and abbreviations, variability in negative speech, and emojis and emoticon interpretation.

V. RESULTS

The evaluation of the models provided insights into the performance of the three algorithms: SVM, LR, and NB. Each algorithm is tested on the Sentiment140 dataset [5], which contains tweets representing positive, negative, and neutral sentiments. The models were trained on 70% of the dataset and tested on the remaining 30% to ensure a fair and unbiased evaluation. Performance was measured using accuracy, precision, recall, and F1 score for both training and testing datasets. Table I and Figure 2 show that using SVM with the balancing factor increases accuracy in both training and testing sets, achieving 94.97% and 80.4%, respectively, which is greater than that of LR.

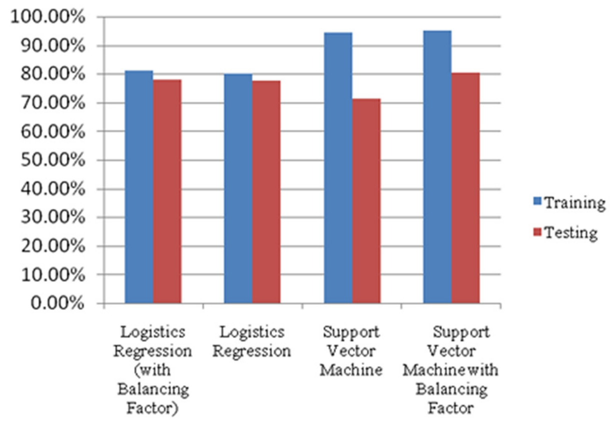

Fig. 2. Training and testing accuracies of SVM and LR with and without the balancing factor.

TABLE I. ACCURACY FOR SVM AND LR

Model	LR with balancing factor	LR	SVM	SVM with balancing factor
Training	81.01%	80.10%	94.51%	94.97%
Testing	77.80%	77.79%	71.46%	80.4%

TABLE II. COMPARATIVE ANALYSIS OF MNB AND GNB RESULTS

Model	Data	Accuracy	Precision	Recall	F1 score
BF_MNB (without Emoji)	Training	92. 55%	92.56%	92.55%	92.55%
	Testing	69.46 %	69.51%	69.46%	69.45%
BF_MNB (with Emoji)	Training	89.88%	91.29%	89.89%	89.47%
	Testing	78.26 %	80.41%	78.27%	76.39%
BF_GNB (without Emoji)	Training	95.2%	95.49%	95.20%	95.19%
	Testing	59.48%	60.23%	59.48%	58.88%
BF_GNB (with Emoji)	Training	91.54%	93.10%	91.54%	91.67%
	Testing	58.4 %	64.35%	58.40%	58.77%

Fig. 3. Graph of training and testing accuracy for Multinomial (MNB) and Gaussian (GNB) Naïve Bayes classifiers with and without emoji.

VI. CONCLUSION

The results show that SVM outperformed the other algorithms in terms of training accuracy and precision, making it the most suitable choice for this sentiment analysis task. The higher F1 score for SVM reflects its ability to balance precision and recall, which is crucial for applications that require reliable classification. NB, although efficient, showed limitations in seizing the relationships between words in the tweets. This highlights the importance of choosing algorithms that align with the precise features of the dataset and the requirements of the analysis. Future work could involve exploring ensemble methods or deep learning models to further enhance accuracy and scalability.

REFERENCES

[1] G. Rao, D. G. Pujari, R. G. Khalkar, and V. A. Medhe, "Study on Software Defect Prediction Based on SVM and Decision Tree Algorithm," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 7s, pp. 90–95, Jul. 2023, https://doi.org/10.17762/ijritcc.v11i7s.6980.

[2] R. G. Khalkar, M. S. Bewoor, S. P. Medhane, and G. Rao, "Comprehensive study of reviews from social media using machine learning techniques," in *Advances in AI for Biomedical Instrumentation, Electronics and Computing*, 1st ed., London: CRC Press, 2024, pp. 288–293.

[3] S. Sorare and M. Kamble, "Ternary Classification of Product Based Reviews: Survey, Open Issues and New Approach for Sentiment Analysis," *Indian Journal of Artificial Intelligence and Neural Networking*, vol. 1, no. 2, pp. 1–8, Apr. 2021, https://doi.org/10.54105/ijainn.B1008.041221.

[4] M. Cliche, "BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs." arXiv, 2017, https://doi.org/10.48550/ARXIV.1704.06125.

[5] "Sentiment140 dataset with 1.6 million tweets." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/kazanova/sentiment140.

[6] S. Kumar, R. Sanasam, and S. Nandi, "IndiSentiment140: Sentiment Analysis Dataset for Indian Languages with Emphasis on Low-Resource Languages using Machine Translation," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Mexico City, Mexico, 2024, pp. 7689–7698, https://doi.org/10.18653/v1/2024.naacl-long.425.

[7] Y. Qi and Z. Shabrina, "Sentiment analysis using Twitter data: a comparative application of lexicon- and machine-learning-based approach," *Social Network Analysis and Mining*, vol. 13, no. 1, Feb. 2023, Art. no. 31, https://doi.org/10.1007/s13278-023-01030-x.

[8] V. Priyanka, "Twitter Sentiment Analysis using Deep Learning," *SSRN Electronic Journal*, 2021, https://doi.org/10.2139/ssrn.3885137.

[9] L. Jiang, M. Yu, M. Zhou, X. of the 49th A. M. of the A. for C. L. Liu, and T. Zhao, "Target-dependent Twitter Sentiment Classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, OR, USA, Jun. 2011, pp. 151–160.

[10] J. Ranganathan and A. Tzacheva, "Emotion Mining in Social Media Data," *Procedia Computer Science*, vol. 159, pp. 58–66, Jan. 2019, https://doi.org/10.1016/j.procs.2019.09.160.

[11] M. S. Neethu and R. Rajasree, "Sentiment analysis in twitter using machine learning techniques," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, Tiruchengode, Jul. 2013, pp. 1–5, https://doi.org/10.1109/ICCCNT.2013.6726818.

[12] D. Gurkhe, N. Pal, and R. Bhatia, "Effective Sentiment Analysis of Social Media Datasets using Naive Bayesian Classification," *International Journal of Computer Applications*, vol. 99, no. 13, pp. 1–4, Aug. 2014, https://doi.org/10.5120/17430-8274.

[13] M. Zouidine and M. Khalil, "Large Language Models for Arabic Sentiment Analysis and Machine Translation," *Engineering, Technology & Applied Science Research*, vol. 15, no. 2, pp. 20737–20742, Apr. 2025, https://doi.org/10.48084/etasr.9584.

[14] Y. D. Al-Otaibi, "Enhancing e-Commerce Strategies: A Deep Learning Framework for Customer Behavior Prediction," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 15656–15664, Aug. 2024, https://doi.org/10.48084/etasr.7945.

[15] K. Divya and Mrs P. Menaka, "Twitter Sentiment Analysis," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 11, no. 2, pp. 1305–1310, Mar. 2025, https://doi.org/10.32628/CSEIT25112465.

[16] O. Y. Adwan, M. Al-Tawil, A. Huneiti, R. Shahin, A. Abu Zayed, and R. Al-Dibsi, "Twitter Sentiment Analysis Approaches: A Survey," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, no. 15, Aug. 2020, Art. no. 79, https://doi.org/10.3991/ijet.v15i15.14467.

[17] E. Deniz, H. Erbay, and M. Coşar, "Multi-Label Classification of E-Commerce Customer Reviews via Machine Learning," *Axioms*, vol. 11, no. 9, Aug. 2022, Art. no. 436, https://doi.org/10.3390/axioms11090436.

[18] K. Sudheer and B. Valarmathi, "Real time sentiment analysis of e-commerce websites using machine learning algorithms," *International Journal of Mechanical Engineering and Technology*, vol. 9, pp. 180–193, Feb. 2018.

[19] R. G. Khalkar, M. S. Bewoor, S. P. Medhane, and G. Rao, "Comprehensive study of reviews from social media using machine learning techniques," in *Advances in AI for Biomedical Instrumentation, Electronics and Computing*, 1st ed., London, UK: CRC Press, 2024, pp. 288–293.

[20] S. Gouthami and N. P. Hegde, "Automatic Sentiment Analysis Scalability Prediction for Information Extraction Using SentiStrength Algorithm," in *Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems*, vol. 612, A. B. Reddy, S. Nagini, V. E. Balas, and K. S. Raju, Eds. Springer Nature Singapore, 2023, pp. 21–30.

[21] A. Badgaiyya, P. Shankarpale, R. Wankhade, U. Shetye, and S. Pande, "An Application of Sentiment Analysis Based on Hybrid Database of Movie Ratings," *International Research Journal of Engineering and Technology*, vol. 08, no. 01, 2021.