Volume 11, Issue 6, June – 2026 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun364

Twitter Sentiment Analysis

Himani H. Patel[1]; Om Mahalle[2]; Anish Shetty[3]; Sachin Hugar[4]

[1](Guide); [1]Assistant Professor

[1]Information Technology (DYPCOE) SPPU Pune, India
[2]Information Technology (DYPCOE) SPPU Pune, India
[3]Information Technology (DYPCOE) SPPU Pune, India
[4]Information Technology (DYPCOE) SPPU Pune, India

Publication Date: 2026/06/22

Abstract: The concept of sentiment analysis of social media texts poses a vital role in better understanding public opinions, behavior patterns of consumers, and societal trends. Twitter, being a microblogging website, poses a tremendous challenge to the social media world due to its highly noisy and informal nature of tweets. This paper emphasizes a highly efficient and scalable sentiment analysis system using the Sentiment140 dataset. The dataset comprises 1.6 million tweets that are automatically labeled using emoticons. The system uses lightweight text processing steps followed by converting tweets into a numerical representation by means of Term Frequency-Inverse Document Frequency (TF-IDF) with unigrams, bigrams, and sublinear scaling. Three powerful yet classic machine learning classifiers—Multinomial Naive Bayes and Logistic Regression (tuned using GridSearchCV) and Linear SVM—are combined using a hard voting classifier. This paper proves that the combination of classifiers yields better accuracy and performance. An experimental study using a train-test splitting ratio of 75:25 demonstrates that the combination classifier exhibits higher accuracy, precision, recall, and F1-measure. The system has been found computationally efficient. Error analysis indicates that slang usage, sarcasm, and the use of emojis constitute major challenges. The results confirm that classical linear models, when trained on large-scale data and combined effectively, provide a strong, scalable baseline for Twitter sentiment analysis suitable for real-time deployment. Future work includes incorporating emoji-aware features and contextual embeddings to handle linguistic nuance.

Keywords: *Component, Formatting, Style, Styling, Insert.*

How to Cite: Himani H. Patel; Om Mahalle; Anish Shetty; Sachin Hugar (2026) Twitter Sentiment Analysis. *International Journal of Innovative Science and Research Technology*, 11(6), 813-819. https://doi.org/10.38124/ijisrt/26jun364

I. INTRODUCTION

Opinion mining, also called sentiment analysis simply, is the detection of the emotional polarity of given textual content by machine. The importance of SA in the fields of corporate intelligence, political analysis, health monitoring, and social science research is increasing due to the speedy development of social media platforms. Sentiment analysis is increasingly being used by businesses to monitor customer responses, improve brand reputation, find new trends, and make decisions.

The real-time nature and public accessibility of Twitter are what make it a very important source of sentiment information among all the different social media sites. Every day, millions of tweets are transmitted, depicting people's thoughts and ideas about a variety of topics, from people's lives and societal issues to products and happenings. There are a number of particular difficulties when conducting sentiment research on Twitter. Tweets are just casual messages comprising a series of characters that incorporate user mentions, hashtags, emoticons, slang, and acronyms. Moreover, sentiment may also be implied instead of more obviously expressed through sentimental language. Without careful modification, these features reduce the effectiveness of conventional natural language processing methods.

Early research on Twitter sentiment analysis shown that classical machine learning models trained on bag-of-words features could compete when large labeled datasets were available. The Sentiment140 dataset, first presented by Go et al., is one of the most

When big labeled datasets were available, early Twitter sentiment analysis research showed that traditional machine learning models trained on bag-of-words features could perform competitively. One of the most popular benchmarks in this field is still the Sentiment140 dataset, which was first presented by Go et al. It offers 1.6 million tweets that have been automatically annotated with emoticons, allowing for extensive supervised learning without the need for expensive manual annotation.

In this study, we use classical linear models to develop a sentiment analysis system that is scalable, efficient, and ready for deployment. We placed interpretability, computational efficiency, and robustness above intricate deep learning

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun364

architectures. We use a hard-voting ensemble to integrate many classifiers using TF-IDF feature extraction with unigrams and bigrams. Our objective is to show that a well-crafted ensemble of linear models is still quite successful for large-scale Twitter sentiment categorization, not to claim state-of-the-art performance.

II. RELATED WORK

Go et al. designed, using emoticon categorization as well as remote supervision, one of the largest sets of benchmarks on Twitter sentiment analysis, known as Sentiment140. Go et al. managed to collect approximately 1.6 million tweets for their trend-setting study, in which tweets were labeled as either good or negative depending on the presence of emoticons. Go et al. trained a naive Bayes classifier, maximal entropy classifier (logistic regression), and Support vector machines using unigrams and bigrams. The proposed classifiers had an accuracy of 80% on automatically labeled data, demonstrating the possibility of effective supervised learning in sentiment analysis using large sets of noisy labels.

Subsequent studies affirmed the suitability of earlier linear classifiers in Twitter sentiment analysis when combined with the use of TF-IDF weightings or bags of words. This effectively trained an effective linear decision boundary, and the high-dimensional sparsity of text characteristics on Twitter saw linear SVM and logistic regression exhibit impressive performance. Unlike complex deep learning approaches, the lower training costs and interpretability of linear classifiers remain a crucial factor in their adoption in business contexts.

More recent studies have investigated deep learning methods for Twitter sentiment analysis, including convolutional neural networks, recurrent neural networks, and transformer-based models such as BERT. The transformer-based models outperform traditional models by capturing the contextual semantics and long-range dependencies in text. However, these models require substantial computational resources, longer training cycles, and careful fine-tuning, making them unsuitable for deployment situations on large-scale or real-time bases involving millions of tweets.

Methods for ensembling had been proposed to close that gap of efficiency and accuracy. Voting-based ensembles, which combine heterogeneous classifiers like Naive Bayes, Logistic Regression, and Support Vector Machines, complementing the decision boundaries, reducing the variance, have been shown to perform better compared to the ensembled individual models. Hard-voting ensembles, as found by previous studies, perform best if the component models exhibit diversity in error patterns, which is a trait commonly encountered in sentiment classification tasks.

Using a hard-voting ensemble of Multinomial Naive Bayes, Logistic Regression, and Linear SVM trained on TF–IDF features with unigram and bigram representations, the current study builds on this body of work. This method prioritizes robustness and computational efficiency while attaining competitive classification performance on the entire Sentiment140 dataset, adhering to proven best practices in scaled Twitter sentiment analysis.

III. DATASET DESCRIPTION

One of the most widely used benchmarks in Twitter sentiment analysis is the Sentiment140 dataset, which was collected in 2009 by Stanford University academics using the Twitter public API. The dataset consists of approximately 1.6 million English-language tweets evenly distributed across sentiment classes, with approximately 800,000 positive and 800,000 negative examples [18][19]. The metadata parameters that are associated with each tweet include sentiment polarity, tweet ID, time stamp, query term, username, and tweet content. During this investigation, just the tweet content and the sentiment label will be utilized for sentiment classification.

Sentiment labels in Sentiment140 are generated using a distant supervision strategy based on emoticons. Tweets containing positive emoticons such as ":)" or ":-)" are labeled as positive, while those containing negative emoticons such as ":(" or ":-(" are labeled as negative [18][20]. To avoid trivial classification cues, all emoticons used for labeling are removed from the tweet text prior to release. Tweets containing both positive and negative emoticons, as well as retweets (identified by the "RT @user" pattern), are excluded to reduce ambiguity and duplication [20]. This labeling strategy enables large-scale supervised learning without the high cost of manual annotation, albeit at the expense of introducing some degree of label noise.

In addition to the primary binary-labeled dataset, Sentiment140 also includes a much smaller manually annotated test set of approximately 13,000 tweets with three sentiment classes (positive, negative, and neutral). However, consistent with prior large-scale sentiment analysis studies, the present work focuses exclusively on the full automatically labeled binary dataset in order to maximize training data volume and ensure scalability.

Prior to extracting the features, standard text preprocessing was performed to reduce the inherent noise in Twitter data. Non-informative columns are removed, and all text of the tweets was changed to lowercase to ensure case normalization. URLs and user mentions are removed, along with excess whitespace because such elements do not directly contribute toward sentiment polarity and may introduce sparsity into the text data [1][2]. Normalization of punctuation symbols, non-alphanumeric characters, and repeated characters further improves textual consistency. These pre-processing steps result in a cleaned corpus that is mainly composed of tokens bearing sentiment suitable for vector-based feature extraction.

The resulting dataset is particularly well-suited for evaluating scalable sentiment classification pipelines due to its size, balance, and representation of informal social media language. Yet there are also inherent weaknesses in emoticonbased classification, including noisy sentiment assignments, a lack of coverage for neutral or mixed mood, and a potential bias toward overt emotional display. These factors are acknowledged and considered in the evaluation and error

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun364

analysis. Despite these limitations, Sentiment140 remains a useful and generally accepted standard for investigating large-scale Twitter sentiment analysis, and it is an appropriate dataset given the goals of this work.

IV. MEHTODOLOGY

This section describes the whole sentiment analysis pipeline used in this investigation; hence, feature extraction, ensemble building, model training, text preparation, and the methodology used to carry out the evaluation are outlined in this section. It describes a scalable methodology with high computational efficiency suitable for huge Twitter data.

➢ *Text Preprocessing*

Twitter's casual vocabulary, the presence of acronyms, hyperlinks, mentions, as well as unformatted text, makes Twitter naturally noisy. Every tweet undergoes a simple yet efficient preprocessing pipeline before the model is applied to it with the sole intent of reducing the noise while retaining the sentiment-bearing information.

Lowercase is applied to all the content of the tweets with the objective of efficiently ensuring normalization of cases while minimizing word vocabularies. Web links and URLs are eliminated from the text since the inclusion of such links can generate text sparsity and do not significantly impact sentiment. Prejudice against users is prevented by eliminating user mentions such as "@username." Space is trimmed from the beginning and end of words, with any additional spaces being crushed. Words that have social media descriptions are naturally included without using extreme linguistic normalization such as stemming and lemmatization.

The goal of this straightforward preprocessing strategy is to strike a balance between scalability and noise reduction. Indeed, when dealing with millions of tweets, more sophisticated normalization techniques can, on one hand, lead to an increase in processing complexity as well as costs, regardless of any potential efficiency improvements.

➢ *Feature Extraction Using TF-IDF*

Following the preprocessing step, the Term Frequency-Inverse Document Frequency algorithm is employed to transform the tweets into numerical feature vectors. This algorithm is attractive for this purpose as it captures the relevance of the term to the particular document and the rarity of the term over the entire collection of documents.

In order to account for local context features such as negations and short sentences like “not good” or “very bad,” both unigrams and bigrams (in the form of ngram_range = (1,2)) are used in the present study. It has been shown that the use of bigrams improves the performance in sentiment classification by capturing short relationships typically lost in unigrams.

The vocabulary size is restricted up to the 100,000 most common words. This limits the dimensionality of the features. Sublinear scaling of term frequencies is also activated by replacing the normal tf values by 1+log(tf). This improves the stability of the model by reducing the influence of the most frequently appearing words and preventing their dominance.

The resulting TF-IDF matrix is high-dimensional and sparse. It is appropriate for linear classifiers that are optimized to work with sparse input data.

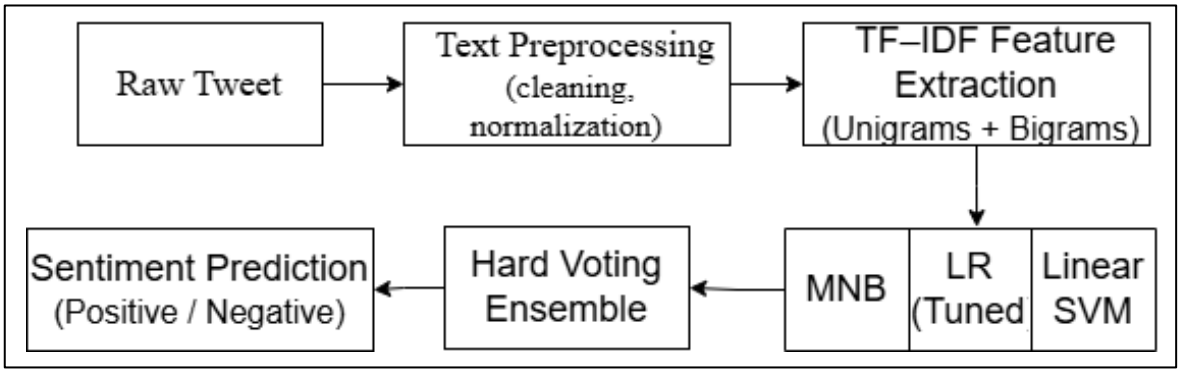

Fig 1 Twitter Sentiment Analysis Pipeline

➢ *Classification Models*

Three classical linear classifiers are trained on the TF–IDF features:

- Multinomial Naive Bayes s a probabilistic classifier based on Bayes' theorem with the assumption of conditional independence between features. Although simple, MNB performs competitively in text classification tasks, thanks to its efficiency in modeling word frequency distributions. It is a fast and robust baseline in this study.
- For representing the condition probability of class labels given input information, discriminative linear classifier, i.e., logistic regression, has been very effective, especially on sparse high-dimensional data. For optimizing the C to be used in regularized classifiers, the GridSearchCV method has been utilized in order to modify Logistic Regression so that the trade-off between bias and variance can be ensured. The best configuration can be determined by cross-validation.
- Linear SVM (LinearSVC) looks for a maximum-margin hyperplane that divides sentiment classes in the feature space. It is known to effectively train on big, sparse datasets and perform well on text classification problems. Instead of outputting calibrated probability as probabilistic models do, Linear SVM outputs discrete class predictions.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun364

- Hard Voting Ensemble: In this hard voting ensemble technique, there is a proper utilization of the characteristics that complement each other among individual classifiers. In hard voting, a majority vote decides the prediction outcome. Each classifier or base classifier predicts a sentiment label for a given tweet. In this voting scheme, there is no complexity or computational overhead because probability calibration via Platt Scaling is not used. Also, since Linear SVM does not support probability estimation, hard voting becomes a prominent choice compared to soft voting. This scheme maintains efficiency while ensuring a validated and mathematically accurate prediction aggregation scheme. In this ensemble technique, Linear SVM, modified Logistic Regression, and Multinomial Naive Bayes are used.

Each classifier captures different inductive biases: probabilistic modeling (MNB), probabilistic discrimination (LR), and margin maximization (SVM).

V. EXPERIMENTAL SETUP

The experimental purpose of evaluating the computational viability and the prediction proficiency of classical linear models and their ensemble approach using a significant volume of Twitter sentiment data is as follows: The entire Sentiment140 corpus, which comprises more than 1.6 million Twitter posts, is utilized for the purpose of conducting trials to ensure that the results reflect the behavior for a volume of considerable data rather than a small sample.

➢ *Data Splitting Strategy*

- After the dataset was split into 75% training data and 25% validation data, approximately 1.2 million tweets were used for training and 400,000 for validation. To maintain the original class distribution and guarantee an equal number of positive and negative samples in both splits, stratified sampling was used. Here, a fixed random seed has been used to make sure that experiments are reproducible over time. This split ratio is common in large-scale text classification problems, as it keeps a sizeable holdout set for objective assessment while offering enough data for reliable model learning.

➢ *Feature Representation*

- Each tweet was converted to a numerical feature vector using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. As the vectorizer was set to extract both unigrams and bigrams (ngram_range=(1,2)), the model was able to identify some contextual cues, such as negations ("not good") and popular sentiment cues. The number of features was also set by limiting the size of the vocabulary, which was set to 100,000, as well as by filtering rare terms using the minimum document frequency.
- For making the tweets numerical feature vectors, Term Frequency–Inverse Document Frequency was applied to all tweets. A configuration of the vectorizer to extract both unigrams and bigrams-ngram_range = (1, 2)-made short contextual patterns such as negations ("not good") and common sentiment expressions observable to the model. The terms that were extremely rare had a minimal document frequency criterion; vocabulary size was constrained up to 100000 features to regulate the dimensionality and memory usage.

➢ *Model Training Configuration*

Three supervised learning algorithms were trained using the TF–IDF feature matrix:

- Multinomial Naive Bayes (MNB) was used as a probabilistic baseline model. Despite its strong independence assumptions, MNB is computationally efficient and often performs well on word-frequency-based representations.
- Logistic Regression (LR) has been utilized as the discriminative linear classifier. GridSearchCV has been employed to implement the hyperparameter optimization with the regularization parameter C via three-fold cross-validation on the training set. This ensured that the model achieved an optimal bias–variance trade-off.
- A Linear Support Vector Machine (Linear SVM) was employed with a hinge loss function and was used to maximize the margin for the different sentiment classes. Linear SVM is a strong classifier for dealing with high-dimensional sparse feature spaces.

Table 1 Performance Comparison of Sentiment Classification Models

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.7997	0.8049	0.7911	0.7979
Logistic Regression	0.8217	0.8156	0.8314	0.8234
Linear SVM	0.8161	0.8093	0.8271	0.8181
Hard Voting Ensemble	**0.8206**	**0.8158**	**0.8282**	**0.8219**

➢ *Ensemble Construction*

To incorporate the respective advantages of each classifier in prediction, an ensemble of classifiers was built using the technique of hard voting. The final prediction of the ensemble is a combination of predictions made by Naive Bayes, tuned logistic regression, and Linear SVM. Each model is given one vote based on their respective predictions, and the final prediction is made in accordance with the majority vote. Hard voting is preferred over soft voting because probabilities are not assigned in Linear SVM, and even if they were, there would be further computational overhead.

➢ *Evaluation Metrics*

Model performance was evaluated on the validation set using multiple standard classification metrics:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun364

- Accuracy to measure overall correctness,
- Precision to assess prediction reliability for the positive class,
- Recall to evaluate sentiment detection coverage,
- F1-score as a harmonic balance between precision and recall,
- ROC–AUC (reported for probabilistic models) to assess discriminative capability across decision thresholds.

Using multiple metrics provides a comprehensive assessment, particularly important in sentiment analysis where class-specific errors may have different practical implications.

VI. RESULT AND ANALYSIS

This section discusses the performance of the individual classifiers as well as the hard voting ensemble over the Sentiment140 validation set. The aim is to assess the accuracy as well as robustness of the classifiers when trained on large-scale Twitter data.

➢ *Overall Model Performance*

All the classifiers attain high accuracy, which validates the effectiveness of the TF-IDF-based linear methods for Twitter-based sentiment classification. Among the individual classifier accuracy results, the results of Logistic Regression, Linear SVM, and Multinomial Naive Bayes are found to be higher compared to the accuracy of the other classifiers. It is observed that the discriminative nature of these classifiers results in a higher F1 score.

The hard voting ensemble has the best performance overall in almost all metrics. The prediction ensemble improves the overall stability of the classifications by combining the predictions of all three classifiers. The biases of each individual model are reduced in this ensemble. The above results verify that ensemble methods improve stability in classifications using individual models that demonstrate differing error patterns.

➢ *Comparative Analysis of Individual Models*

Multinomial Naive Bayes classifier makes for a good baseline due to its simplicity and efficiency, but due to its conditional independence assumption, it cannot fully leverage feature interactions, which are inherent in sentiment-laden phrases. Logistic Regression achieves higher precision and recall with weighted feature contribution, which makes sense when viewed in terms of TF-IDF feature space for distinguishing between positive and negative sentiment.

Linear SVM has competitive performance with the advantage of margin maximization in high-dimensional space. Moreover, Linear SVM's resistance to overfitting is particularly useful for sparse textual data. Although Logistic Regression and Linear SVM have the same accuracy as each other, the prediction error of each is different; therefore, they can be used in combination with each other.

Table 2 Class-Wise Performance Metrics for Sentiment Classification

Class	Precision	Recall	F1-score	Support
Negative	0.83	0.81	0.82	200,000
Positive	0.82	0.83	0.82	200,000
Accuracy			**0.82**	400,000
Macro Avg	0.82	0.82	0.82	400,000
Weighted Avg	0.82	0.82	0.82	400,000

➢ *Ensemble Behavior and Stability*

The hard voting ensemble always provides improvement by settling disputes between classifiers. If there is a situation where one classifier misclassifies a tweet, there is always a possibility that other classifiers have correct classifications, thus helping to improve recall measures as well as precision, which is very essential in sentiment analysis.

This effectiveness of the ensemble speaks to the capability of simpler linear models to perform comparably, although with much lower computational complexity, compared to more intricate techniques.

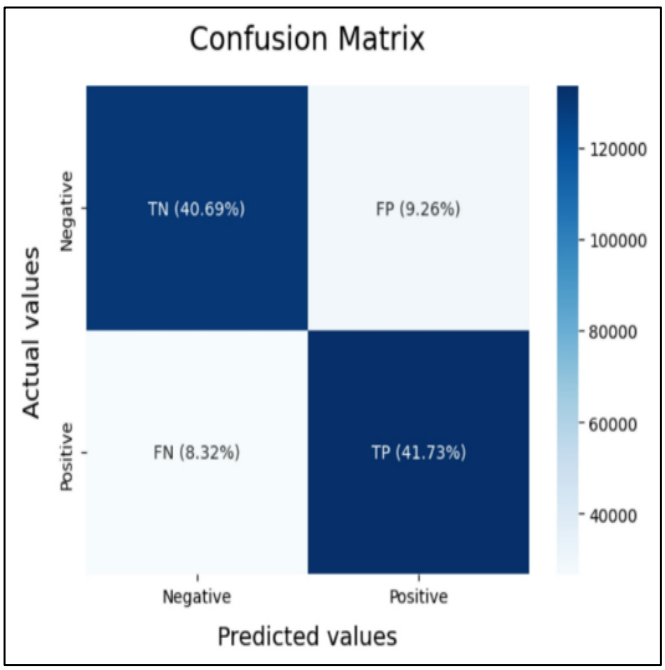

Fig 2 Confusion Matrix of Hard Voting Ensemble

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun364

➢ *Error Analysis*

Despite the strong overall performance, there are some classes of tweets that present difficulty for classification. Investigating the misclassified tweets for the classes where our system has underperformed reveals that the dominant sources of error for our system appear to be the Tweet instances with sarcastic and ironic content. Tweets with positive lexical features and negative intent appear to be misclassified by our system.

In addition, tweets that are heavily reliant on emojis or contain slang are also problematic because such components are either removed or not adequately accounted for by word counts. Similarly, ambiguous wording and mixed sentiments expressed in a tweet are also problematic. Such observations are also supported by other literature, which also recognize these challenges identified above with respect to tweets and Twitter.

➢ *Error Analysis and Observations*

Manual examination of misclassified samples identifies a number of patterns. A large number of difficulties arise when dealing with sarcasm and irony in tweets, as words in these instances may have a positive meaning while expressing a negative message, and vice versa. In particular, sarcastic sentences usually depend on context rather than TF-IDF features.

Moreover, tweets with heavy usage of emojis as well as casual slangs are more susceptible to misclassification issues as well. The fact that emojis are removed during the preprocess step causes valuable sentiment information to be lost as well. Similarly, creative spellings, abbreviations, and code-mixed content can affect word representation as well.

These results align with existing research that has identified difficulties with understanding sarcasm, colloquialisms, and underlying sentiment in Twitter posts as challenges to SA.

➢ *Scalability and Practical Implications*

It should be noted that from the systems perspective, the described pipeline shows high scalability in terms of the application. It can be seen that the application is computationally efficient both in the training and the testing phases, considering the size of the data set (1.6M tweets). This is due to the low memory requirements of the linear classifiers used.

This scalability makes the model appropriate for use in applications involving sentiment monitoring dashboards, social media analysis systems, and decision support systems that require large volumes of data to be processed.

Table 3 Top Positive and Negative Features with Learned Coefficients

Positive Feature	Coefficient	Negative Feature	Coefficient
cant wait	8.905	sad	-16.963
not bad	8.774	miss	-11.291
no problem	8.490	poor	-10.398
happy	6.746	sick	-10.246
can wait	6.450	missing	-9.610
cannot wait	6.165	not happy	-9.389
excited	6.089	sadly	-9.365
welcome	6.069	sucks	-8.729
smile	6.051	not looking	-8.549
glad	5.822	unfortunately	-8.453
thanks	5.820	bummed	-8.331
no prob	5.815	hate	-8.256
no need	5.784	died	-8.238
no worries	5.770	hurts	-8.059
awesome	5.727	rip	-7.923
not sad	5.674	disappointed	-7.804
yay	5.575	missed	-7.779
congratulations	5.510	headache	-7.733
good	5.493	wish	-7.678
love	5.174	bummer	-7.609

VII. LIMITATIONS

It should be mentioned that even though the proposed sentiment analysis pipeline performs exceptionally well and is scalable, it has a number of shortcomings.

Using emoticons and remote supervision, the Sentiment140 dataset first generates sentiment labels. This allows for large-scale annotation, but not all tweets with emoticons effectively transmit the main idea, introducing label noise. The validity of training and evaluation may be jeopardized if tweets with humor, sarcasm, or complex emotional expressions are incorrectly classified.

Second, special characters, hashtags, and emojis are removed from the existing preparation procedure. In addition to simplifying text representation and removing noise, this also removes sentiment-bearing signals, particularly emoticons,

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun364

which are essential for conveying emotions on Twitter.The categorization accuracy of tweets with a large number of emojis is decreased because some emotion cues are lost before feature extraction.

Third, the model operates under a binary sentiment assumption (positive vs. negative). In the real world, sentiment is frequently neutral, ambiguous, or contingent on the situation. The model's application in situations requiring sophisticated sentiment interpretation, such customer feedback analysis or political speech monitoring, is limited by the lack of a neutral class.

Also, although making the system more robust by employing ensemble methods is useful, it can only cope with simple decisions that follow straight-line patterns. While the result will probably be quite good, even if the models perform well, they likely will not perform as well as the methods of deep learning when it comes to dealing with very complex structures. This illustrates that the design was specifically designed to ensure that the system was easy to use and able to scale effectively.

VIII. CONCLUSION AND FUTURE WORK

It developed a scalable Twitter sentiment analysis pipeline using 1.6 million automatically identified tweets from the Sentiment140 dataset. The method includes a hard-voting ensemble of Multinomial Naive Bayes, Logistic Regression, and Linear SVM classifiers, lightweight text preprocessing, and TF-IDF feature extraction from unigrams and bigrams.

Experimental results show that, while maintaining computing efficiency, the ensemble routinely outperforms individual models in accuracy, precision, recall, and F1-score. These findings suggest that when trained on extensive data and appropriately coupled, traditional linear models continue to be reliable baselines for sentiment categorization of noisy social media messages. The system's minimal training and inference costs make it ideal for real-time and large-scale deployment applications.

To effectively manage sarcasm, informal language, and implicit sentiment, future work will concentrate on enhancing sentiment representation by integrating contextual embeddings and emoji-aware features. Promising avenues include extending the model to accommodate neutral or mixed sentiment classes and assessing performance on more recent and multilingual datasets.

REFERENCES

[1]. A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” *CS224N Project Report*, Stanford University, 2009.
[2]. G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.
[3]. J. Ramos, “Using TF-IDF to determine word relevance in document queries,” *Proc. First Instructional Conf. Machine Learning*, 2003.
[4]. T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” *Proc. European Conf. Machine Learning (ECML)*, pp. 137–142, 1998.
[5]. A. McCallum and K. Nigam, “A comparison of event models for Naive Bayes text classification,” *AAAI Workshop on Learning for Text Categorization*, 1998.
[6]. C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[7]. D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
[8]. B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment classification using machine learning techniques,” *Proc. ACL*, pp. 79–86, 2002.
[9]. B. Pang and L. Lee, “Opinion mining and sentiment analysis,” *Foundations and Trends in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.
[10]. F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[11]. O. Sagi and L. Rokach, “Ensemble learning: A survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.
[12]. Y. Kim, “Convolutional neural networks for sentence classification,” *Proc. EMNLP*, pp. 1746–1751, 2014.
[13]. A. Severyn and A. Moschitti, “Twitter sentiment analysis with deep convolutional neural networks,” *Proc. SIGIR*, pp. 959–962, 2015.
[14]. A. Vaswani *et al.*, “Attention is all you need,” *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5998–6008, 2017.
[15]. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” *Proc. NAACL-HLT*, pp. 4171–4186, 2019.