International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

An Enhanced Comparative Study of Machine Learning Algorithm for Twitter Sentiment Analysis

Devendra D. Gonde
Dr. D.Y. Patil Arts, Commerce and Science College, Pimpri, Pune, Maharashtra , India

Vinay N Ayyagari
Dr. D.Y. Patil Arts, Commerce and Science College, Pimpri, Pune, Maharashtra , India

Abstract - Twitter puts out so much text data every day, all unstructured and full of what people think and feel. I mean, its like a huge stream of opinions just floating around. Sentiment analysis seems pretty key here, pulling out actual useful stuff from all that noise. This work looks at comparing a few machine learning setups, like Logistic Regression, Random Forest, Decision Trees, and that Multinomial Naive Bayes one, all with TF-IDF to handle the features. The experiments show Logistic Regression doing the best with TF-IDF, especially on accuracy and that weighted F1-score.

Keywords - Twitter Sentiment Analysis, TF-IDF, Machine Learning, Text Classification, Natural Language Processing.

INTRODUCTION

Social Media use has changed how people express opinions, share experiences, and react to real world events. Social media platforms like Twitter produce millions of posts containing short messages (tweets) that provide insight into public opinion about a variety of issues including politics, products, social issues, and world events. By analyzing these posts using sentiment analysis, researchers, and companies alike can gain useful insight for making business and trend-related decisions.

Although sentiment analysis on Twitter has important implications, it is difficult to accurately assess sentiment on Twitter because of the many different factors involved. Twitter is a medium that allows for individuals to express themselves through short and informal messages (known as tweets), and they tend to be noisy, as many tweets contain slang, acronyms, emojis, and hashtags.

To address these challenges, this study presents a comparison of various machine learning algorithms to extract features using Term Frequency-Inverse Document Frequency (TF-IDF) representations of the tweet data. TF-IDF is able to effectively represent textual information through the prioritization of meaningful words versus words that simply occur frequently throughout the corpus. The machine learning classifiers evaluated in this study include Logistic Regression, Random Forest, Decision Tree, and Multinomial Naïve Bayes, using accuracy and weighted F1-score as the performance metrics to judge the effectiveness of each classifier. This research aims to develop a robust and efficient machine learning classifier to classify sentiment from Twitter, as well as to provide a strong baseline for future research on Twitter sentiment analysis and advanced natural language processing approaches.

LITERATURE REVIEW

Analysis of sentiment has been a popular research area in natural language processing, especially with the proliferation of social media. An early form of sentiment classification based on dictionary look, up would decide whether the sentiment of text is positive or negative depending on the existence of a list of manually compiled sentiment words. Due to simplicity and interpretability of this method, it was a popular approach with limited contextual understanding and limited success on noisy text such as microblogs.

As machine learning improved, supervised learning methods would ultimately win out in the field. Naive Bayes classifiers, Support Vector Machines, Logistic Regression classifiers and others have all been used widely on sentiment analysis datasets of Twitter. It has been found that machine learning classifiers usually outperform lexicon based ones when labeled data is available, but they are also highly susceptible to the effectiveness of text representation and feature extraction.

The Bag, of, Words and n, gram models have been used as feature representation where all the words are weighed equally which decreases its discriminative ability. Term Frequency, Inverse Document Frequency (TF, IDF) have been introduced as a way to weight discriminative words higher and more common words lower. A few research works have demonstrated improved sentiment classification results with tf, idf combined with linear classifiers such as Logistic Regression.

Tree based models and ensemble approaches have been considered (Decision Trees, Random Forests). These models are capable of representing complex non, linear relations; however, they tend to perform poorly with the high, D sparse

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

text representation and are prone to overfitting. As class, imbalance has been identified as an issue in Twitter sentiment analysis competition data sets, weighted metrics, such as F1, score, were used for evaluation in the competitions.

Despite the advances achieved by deep learning and transformer models, they are not the most practical choice. But they need a lot of computing power and training data, hence the usefulness of the traditional machine learning classifiers along with well, designed features like TF, IDF.. This paper makes use of the work in the literature, but with a comparative study of several machine learning classifiers on Twitter sentiment data in a practical scenario.

PROBLEM STATEMENT

Users' thoughts and feelings on a wide range of subjects are reflected in the vast amount of short, unstructured text data generated by Twitter. These tweets' informal language, noise, lack of context, and uneven sentiment class distributions make it difficult to classify their sentiment effectively. Many of the sentiment analysis methods currently in use rely on simple feature extraction methods or only use a small number of machine learning models, which frequently results in poor generalization and decreased classification performance.

Thus, this study aims to ascertain the efficacy of various machine learning algorithms, including Random Forest, Decision Tree, Logistic Regression, and Multinomial Naïve Bayes, in Twitter sentiment classification when paired with TF-IDF feature extraction. The goal is to determine the best model based on accuracy and weighted F1-score.

Objective

The primary objective of this research is to perform a comparative analysis of machine learning algorithms for Twitter sentiment classification using TF-IDF-based feature representation.

The specific objectives of the study are as follows:

1. To preprocess and clean Twitter text data in order to reduce noise and improve data quality.
2. To apply Term Frequency–Inverse Document Frequency (TF-IDF) for effective numerical representation of tweet text.
3. To implement and evaluate multiple machine learning classifiers, including Logistic Regression, Random Forest, Decision Tree, and Multinomial Naïve Bayes.
4. To compare model performance using accuracy and weighted F1-score, considering the imbalanced nature of Twitter sentiment data.
5. To identify the most effective machine learning model for Twitter sentiment analysis and provide a reliable baseline for future research.

Methodology and Dataset:

METHODOLOGY

1. **Data Preprocessing**

The raw tweet text was Pre-processed to remove noise and improve model performance. The preprocessing steps included:

- Conversion of text to lowercase
- Removal of URLs, user mentions, hashtags, punctuation, and special characters
- Removal of Stop-words
- Text normalization and tokenization

2. **Feature Extraction using TF-IDF**
Textual data was converted into numerical feature vectors using Term Frequency–Inverse Document Frequency (TF-IDF). By giving terms that appear frequently in a tweet but infrequently throughout the corpus higher weights, TF-IDF suppresses common words while boosting sentiment-relevant features. Bigrams and unigrams were thought to be useful for capturing context. Because the resultant feature vectors are sparse and high-dimensional, they can be used with conventional machine learning classifiers.
3. **Model Implementation**
The following machine learning algorithms were implemented and evaluated:
    - Logistic Regression
    - Random Forest Classifier
    - Decision Tree Classifier
    - Multinomial Naïve Bayes
4. **Performance Evaluation**
Model performance was evaluated using accuracy and weighted F1-score to address the issue of class imbalance. Confusion matrix analysis was also performed to analyse classification behaviour across sentiment classes. Comparative visualizations were generated to illustrate performance differences among the models.

RESULTS AND ANALYSIS

A different way to check how well the models worked involved looking at accuracy alongside the weighted F1-score. Confusion matrices also played a role in making sense of results across unevenly sized sentiment groups. Because tweets weren't evenly split by feeling, these methods helped balance the evaluation. Each measure added clarity where class sizes varied too much.

One look at Figure X shows how Logistic Regression, Random Forest, Decision Tree, besides Multinomial Naïve Bayes stack up when trained on TF-IDF features. Top marks for accuracy? That goes to Logistic Regression. So does the best weighted F1-score across the board. Its edge suggests it copes well with the quirks of high-dimensional, sparse data built from TF-IDF.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

	Model	Accuracy	F1_Weighted
0	Random Forest	0.881121	0.881195
1	Decision Tree	0.765440	0.764989
2	Logistic Regression	0.746745	0.745259
3	Multinomial NB	0.692959	0.679808

Fig X

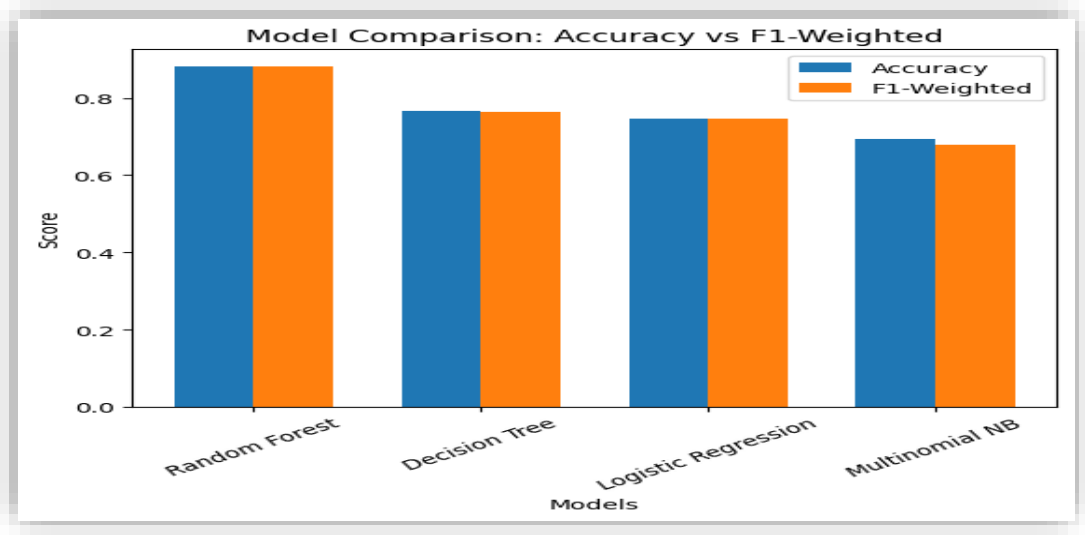

Fig Y

Though Random Forest handled non-linear trends well, it struggled because word-based data tends to lack density. Efficiency came naturally to Multinomial Naïve Bayes, yet context-heavy examples exposed its limits. Where understanding nuance mattered, results dipped noticeably. Overfitting likely dragged down the Decision Tree, making it react too strongly to irregular inputs. Its accuracy fell furthest behind, possibly thanks to fragile logic paths in messy datasets.

Outcomes from testing show feature extraction works best when matched with the right classifier. When it comes to TF-IDF text formats, something like Logistic Regression tends to deliver stronger outcomes. Tree methods can handle organized datasets well - yet they falter when faced with large, sparse textual inputs. Picking models wisely matters a lot, especially since accuracy depends heavily on how success is measured. What stands out is that choosing both algorithm and metric thoughtfully shapes how trustworthy sentiment conclusions really are.

CONCLUSION

One look at machine learning methods for sorting Twitter emotions began with turning words into numbers using TF-IDF. Instead of grouping techniques together, each one stood alone - Logistic Regression, then Random Forest, followed by Decision Tree and lastly Multinomial Naïve Bayes - all tested on cleaned-up tweets. To judge how well they worked, scores came not just from overall correctness but also balance-aware F1 measures plus detailed error patterns. Because some feelings appeared far more often than others, results leaned heavily on where mistakes happened most. Though every method had its moment, none skipped the struggle tied to uneven emotion counts.

Tests showed Logistic Regression using TF-IDF worked better than others when judging accuracy and balanced F1 scores. That happens because straight-line methods handle big, spread-out word data well - exactly what TF-IDF creates. Tree systems tried hard yet still fell short on this mood-sorting job. Even groups of models didn’t catch up despite their usual strength elsewhere.

This research offers a solid starting point for analyzing emotions on Twitter through standard machine learning tools. Researchers and those working in the field might find these findings helpful when choosing methods to sort sentiments. Looking ahead, exploring neural network models along with

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 02, NCRTCS - 2026

context-aware word representations could push accuracy higher while uncovering subtle language features in online conversations.

FUTURE SCOPE

While this work shows how well TF-IDF models handle Twitter sentiment tasks, room remains for growth. Instead of stopping here, researchers might test richer word representations - ones that see meaning in context, not just frequency. Moving beyond basics, tools like Word2Vec or FastText could reveal deeper links between terms. Another path lies in models built on transformers: BERT, say, or RoBERTa, which adjust their understanding based on surrounding words. These shifts may sharpen how systems detect emotion in short, messy social media posts.

Figuring out sarcasm, irony, emojis, or mixed-language posts on Twitter still trips up systems - something later research could dig into. Watching live reactions by analysing tweets as they flow might open doors to tracking shifting opinions. Then again, balancing uneven categories through methods like synthetic data tweaks or weighted penalties could make models handle tough cases better.

REFERENCES

[1] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment classification using machine learning techniques,” Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79–86, 2002.

[2] 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.

[3] J. Eisenstein, “What to do about bad language on the internet,” Proceedings of NAACL-HLT, pp. 359–369, 2013.

[4] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988.

[5] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[6] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” European Conference on Machine Learning, pp. 137–142, 1998.

[7] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python, O’Reilly Media, 2009.

[8]] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[9] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” Proceedings of NAACL-HLT, pp. 4171–4186, 2019.