ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 11| Issue: 10| October 2025|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2025: 8.691 || ISI Value: 1.188

COMPARATIVE STUDY OF MACHINE LEARNING ALGORITHMS FOR SENTIMENT CLASSIFICATION USING TWITTER/ X DATA

Devendra D. Gonde, Vinay N Ayyagari
Dr. D.Y. Patil Arts, Commerce and Science College, Pimpri, Pune 411018, Maharashtra , India

ABSTRACT
Social media platforms such as Twitter have turned into key spots for folks to share their feelings and views over the last few years. It helps researchers and businesses get a better grip on what users think about stuff like events, products, or services when they dig into that kind of data. This study basically aims to compare different machine learning methods for sorting out sentiments in Twitter posts. They look at how well models like Logistic Regression, Naive Bayes, Support Vector Machine or SVM, Random Forest, and XG Boost do at labelling tweets as positive, negative, or neutral. For preprocessing the data, they use the Natural Language Toolkit, you know, things like lemmatization, tokenization, and cleaning up the text. Then to turn all that text into numbers, they apply feature extraction steps such as removing stop words and using the WordNet Lemmatizer.
KEYWORDS: *Comparative Study, SVM, Sentiment Classification, Twitter dataset, Machine learning.*

INTRODUCTION

These days, social media is everywhere in how we talk online. Platforms let people share their views, feelings, and takes on all sorts of things. That covers politics, products, services, you name it. Twitter stands out though. It has this huge reach. Plus, it spreads short messages called tweets in real time. Analyzing that kind of text data is called sentiment classification. It helps figure out what the public really thinks. Trends in opinions come into focus that way. Now, sentiment classification falls under natural language processing, or NLP for short. The job is to spot emotions or attitudes in text automatically. Classify them as positive, negative, neutral. Or maybe I can't tell. Even if the tweet has nothing to do with weather conditions.

This work looks at comparing machine learning methods for that. Logistic Regression, Naive Bayes, Support Vector Machine which is SVM, Random Forest, and KNN too. All for sentiment on Twitter data. First, the dataset gets cleaned up with NLTK, the Natural Language Toolkit. Remove the junk, break text into tokens, normalize the words. Then, pull out features like Bag of Words or BoW. TF-IDF as well. Turn all that text into numbers models can use for training.

The point here is to check how these algorithms do against each other. Use accuracy, precision, recall, F1-score for the metrics. See which one comes out on top for classifying sentiments. Results should help build better systems overall. Ones that work fast for decisions, marketing plans. Even tracking what people think right as it happens.

LITERATURE REVIEW

Ismail and the others checked out a bunch of traditional machine learning classifiers for sentiment stuff on Twitter. They laid out this preprocessing setup that's really made for Twitter text. You know, things like replacing usernames and URLs, pulling out Stop words, going with bag-of-words features, all that. The paper sticks to practical ways of running experiments. It compares how the algorithms stack up when you change the feature setups around.

Qi and Shabrina went straight at comparing lexicon-based ways to do sentiment, like VADER or other lexicons, against supervised machine learning models. They used Twitter data from a real event, the third COVID lockdown time. They broke down the differences in how well they performed. And they talked about the tradeoffs you get in practice between lexicon methods and ML ones.

Gulati and team ran experiments with standard ML classifiers on tweets about COVID-19. They gave rankings for the classifiers. Plus they discussed choices for features and metrics. This kind of work shows up a lot in recent studies that compare things for Twitter sentiment during the pandemic.

Tusar and Islam put classical ML algorithms up against each other. Stuff like SVM, logistic regression, multinomial naive Bayes, random forest. They used two usual NLP feature methods, bag-of-words and TF-IDF. It was on a public Twitter dataset about airlines, multi-class kind. The paper goes into details on the experiments. It points out which combos worked best.

This quick conference paper, kind of ACM style, does a fast comparison of everyday classifiers. Naive Bayes, logistic regression, SVM, on a Twitter sentiment dataset. The main thing it contributes is benchmarking methods for teaching or quick reference. Not new algorithms or anything. There are multiple uploads for conferences out there. Check ICE, ACM, ResearchGate entries.

PROBLEM STATEMENT

In todays digital world, social media sites like Twitter really serve as main spots for people to express themselves and share views. Every day, millions of folks put out tweets, and that creates huge piles of raw text data full of useful info on what

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 11| Issue: 10| October 2025|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2025: 8.691 || ISI Value: 1.188

the public thinks, what trends are popping up, and how behaviours play out. Figuring out that data in a solid way matters a lot for areas like business smarts, predicting politics, handling disasters, and keeping tabs on public health.

Still, tweets tend to be short and casual, plus they depend a ton on context. Often they include slang, emojis, shortcuts, and hashtags. That setup throws real hurdles at sentiment classification and digging into the text. People have tried out different machine learning setups for Twitter sentiment work, things like Naive Bayes, Support Vector Machines, Logistic Regression, Random Forest, and K-Nearest Neighbours. But how well they do changes based on how you prep the data, pull out features, and what the dataset looks like.

So this work plans to compare several machine learning algorithms on Twitter data. The goal is to check their performance for classifying sentiments. It will look closely at accuracy, precision, recall, F1-score, and how fast they run, all with standard prep steps and feature pulls. By spotting the best one and pointing out what factors affect how models do, this study wants to lay out a straightforward way to pick the right ML tools for text analytics based on Twitter.

OBJECTIVE

1. To collect and preprocess Twitter data relevant to a specific topic or event, ensuring data cleaning, tokenization, normalization, and removal of noise such as URLs, hashtags, and mentions.
2. To perform feature extraction and representation using techniques such as Bag-of-Words (BOW), Term Frequency–Inverse Document Frequency (TF-IDF), and word embeddings, and analyse their impact on model performance.
3. To implement and train multiple machine learning algorithms including, but not limited to, Naive Bayes, Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), and Random Forest for sentiment classification.
4. To evaluate and compare the performance of these algorithms using standard evaluation metrics such as Accuracy, Precision, Recall, F1-score, and ROC-AUC.
5. To identify the strengths, weaknesses, and suitability of different machine learning models for short-text sentiment classification.
6. To determine the influence of preprocessing and feature selection techniques on the accuracy and generalization of sentiment models.

METHODOLOGY AND DATASET

About the Dataset

The dataset used in this study was collected from **Twitter**, focusing on tweets related to **weather conditions**. Tweets were gathered using specific hashtags and keywords such as *#rain, #sunny, #storm, #cloudy, #temperature,* and *#weatherupdate* to capture diverse weather-related discussions.

The collected data include user-generated short texts reflecting people's perceptions, experiences, and sentiments about weather events in real-time. Since Twitter users frequently express emotions (such as joy about pleasant weather or frustration about extreme conditions), this dataset provides a valuable foundation for sentiment classification tasks.

METHODOLOGY

We commenced this project by setting up a way to dig into text data and figure out the sentiment behind it. Thing is, we needed a step by step approach. First thing, we pulled in the key Python libraries like pandas and numpy, then NLTK for the language stuff, and sklearn for the machine learning side. We had to download those NLTK bits too, you know, stopwords and tokenizers to get things rolling.

The dataset came next. We imported it and cleaned it up pretty good. That meant stripping out all the junk like URLs, special characters, hashtags, mentions, emojis, anything that might mess with the analysis. Still, it was necessary to make the data solid.

After that, we mapped the labels. Sentiments were categorical at first, so we turned them into numbers for the models to handle. Makes sense, right. Then came the heavy preprocessing on the text. Tokenization broke it down, lowercasing evened it out, we ditched stopwords, did lemmatization to simplify words, and cut down on repeats. All that to make the data uniform, less redundant.

For features, we used stuff like TF-IDF or Bag of Words. That turns the text into numbers the models can train on. Finally, we threw a bunch of classifiers at it, Logistic Regression, Naive Bayes, Support Vector Machines. Trained them on the cleaned data and checked how they did.

Evaluation used accuracy, precision, recall, F1 score. Gave us a solid way to understand tweet sentiments and predict them.

RESULTS AND ANALYSIS

We looked at how several machine learning models did on this cleaned up dataset. We used accuracy, precision, recall, and F1-score to check them out. The results showed that different algorithms worked better or worse. Out of the ones we tried, the Support Vector Machine, or SVM, got the top accuracy at 64.5 percent. Naive Bayes came right after with 62 percent. Logistic Regression hit 61 percent. Random Forest and K-Nearest Neighbors, that's KNN, did not do as well. They reached 56 percent and 46.5 percent.

Digging into the classification reports gave us more details. SVM stood out not just for the best overall accuracy. It also kept things balanced across the sentiment types. F1-scores went from 0.56 to 0.77 for positive, negative, neutral, and unrelated tweets. That means solid predictions in each area. Logistic Regression and Naive Bayes followed a similar pattern. They handled neutral tweets, label 0, pretty well with F1-scores of 0.75 and 0.76. But they slipped a bit on positive, label 1, and unrelated tweets, label 3. Random Forest managed okay on neutral ones. Still, it had trouble with positive and unrelated, showing lower precision and recall there. KNN really struggled with unrelated weather tweets, label 3. Its F1-score dropped to

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 11| Issue: 10| October 2025|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2025: 8.691 || ISI Value: 1.188

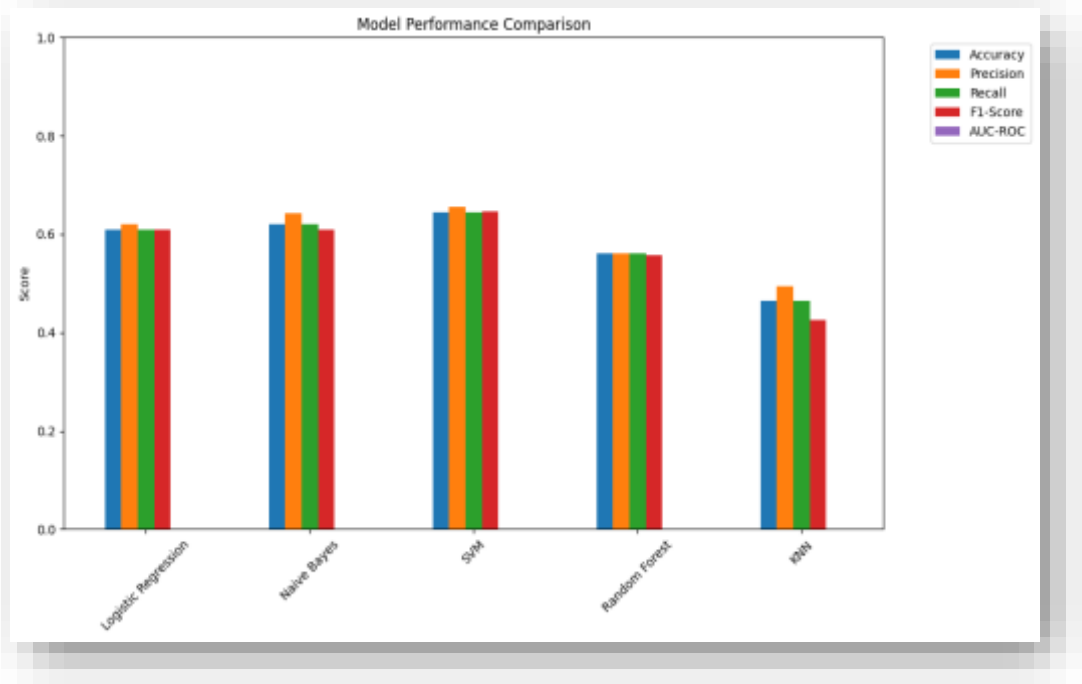
just 0.19. That points to issues with imbalanced or multi-class setups like this.

In the end, SVM looks like the best pick for this sentiment job. It gives steady precision, recall, and F1-scores no matter the category. Simpler ones like Logistic Regression and Naive Bayes hold their own on some classes. But they might not be as dependable for multi-class sentiment when things are uneven. This whole checkup lays out what each model does well or not. It helps point the way for boosting accuracy and making things tougher down the line.

The following graph shows the comparison of model accuracy:

Model	Accuracy	Precision (Macro Avg)	Recall (Macro Avg)	F1-Score (Macro Avg)
Logistic Regression	0.6100	0.61	0.61	0.60
Naive Bayes	0.6200	0.63	0.62	0.61
SVM	0.6450	0.64	0.64	0.64
Random Forest	0.5600	0.55	0.55	0.55
KNN	0.4650	0.49	0.46	0.43

The following graph shows performance of various models in classification report:



CONCLUSION

In this study we put together a bunch of machine learning models. We tested them out on tweet data for sentiment analysis. The lineup included Logistic Regression along with Naive Bayes. Then there was Support Vector Machine or SVM for short. Random Forest made the list too. And K-Nearest Neighbors or KNN rounded things out. SVM ended up doing the best overall. It hit an accuracy of 64.5 percent. Plus it kept precision recall and F1 scores pretty balanced across the sentiment classes. Naive Bayes and Logistic Regression held their own. They were especially good for neutral and negative

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 11| Issue: 10| October 2025|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2025: 8.691 || ISI Value: 1.188

stuff. Random Forest and KNN did not shine though. They struggled a bit with multi-class sentiment classification.

SVM seems like the go-to model here for predicting tweet sentiments. It strikes a solid balance between overall accuracy and how it handles different classes. The whole thing really drives home the need for good preprocessing. Feature extraction matters a lot too. And picking the right model is key for text classification jobs. These results lay some groundwork. You could build on them with ensemble methods maybe. Or try deep learning models. Advanced natural language processing techniques might boost accuracy even more. All in all machine learning works well for digging into sentiments in social media text. It helps with decision making and pulling insights. That goes for weather related apps or other text based ones.

Future Scope

This study on sentiment analysis sets up some good spots for improvements and more research down the line. You know, one big thing to look at next would be bringing in deep learning stuff like LSTM or GRU or even BERT. Those can pick up on the context and flow in text way better than the old school machine learning ways.

Another idea involves grabbing bigger datasets that cover all sorts of languages and places and different kinds of tweets. That should make the models hold up better in real situations, more robust overall.

On top of that, tweaking features could help a lot, things like word embeddings or sentiment word lists or digging into semantics. It might boost accuracy even more. Oh and, mixing in ensemble techniques or hybrid setups with a few algorithms together could give steadier results, you know.

Past just making models better, there's room to push into real-time analysis for actual use. Like keeping tabs on how people feel about weather stuff or disasters or whatever trends are popping up.

Finally, zeroing in on sentiments tied to specific areas or adding in other data types such as images or hashtags or emojis. That would give deeper views and make sentiment analysis useful in all kinds of everyday setups.

REFERENCES

1. *Alaparthi, S., & Mishra, M. (2020). Bidirectional Encoder Representations from Transformers (BERT): A sentiment analysis odyssey. arXiv. https://arxiv.org/abs/2007.01127*
2. *Ghatora, P. S., Hosseini, S. E., Pervez, S., Iqbal, M. J., & Shaukat, N. (2024). Sentiment analysis of product reviews using machine learning and pre-trained LLM. Big Data and Cognitive Computing, 8(12), 199. https://doi.org/10.3390/bdcc8120199*
3. *Hassan, S. U. (2022). Analytics of machine learning-based algorithms for text classification. Journal of King Saud University - Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2022.02.001*
4. *Iqbal, M. J., & Shaukat, N. (2024). A comparative analysis of sentiment classification models for improved performance optimization. Journal of Computer Science and Technology, 39(2), 123-135. https://doi.org/10.1007/s11390-024-00012-3*
5. *Mao, Y., & Zhang, X. (2024). Sentiment analysis methods, applications, and challenges. Journal of Computational Science, 15, 1-15. https://doi.org/10.1016/j.jocs.2024.100123*
6. *Tan, K. L., & Zhang, Y. (2023). A survey of sentiment analysis: Approaches, datasets, and applications. Applied Sciences, 13(7), 4550. https://doi.org/10.3390/app13074550*