RESEARCH ARTICLE OPEN ACCESS

Twitter Sentiment Analysis Using Machine Learning Based on Naïve Bayes Classification

Dr. Laxmi Choudhary
Department of Computer Science and Engineering, Engineering College Ajmer, Ajmer (Rajasthan)

ABSTRACT
Sentiment analysis, also known as opinion mining, is a computational technique used to identify and classify opinions expressed in textual data as positive, negative, or neutral. With the rapid growth of social media platforms such as Twitter, Facebook, and Instagram, vast amounts of user-generated content have become available, providing valuable insights into public opinions, customer preferences, and social trends. Organizations, businesses, and policymakers increasingly rely on sentiment analysis to understand consumer behavior, evaluate public perception of products and services, monitor brand reputation, and support decision-making processes. This study focuses on the classification of Twitter sentiments using a Machine Learning approach based on the Naïve Bayes Classifier. The Twitter dataset is preprocessed through various text-cleaning techniques, including tokenization, stop-word removal, and feature extraction. Unigram and bigram features are utilized to enhance the representation of textual information and improve classification performance. The proposed framework applies the Naïve Bayes algorithm to categorize tweets into sentiment classes and evaluates the model using standard performance metrics such as accuracy, precision, and recall. Experimental results demonstrate that the proposed approach effectively classifies Twitter sentiments and achieves improved prediction performance after appropriate preprocessing and feature engineering. The findings highlight the usefulness of machine learning-based sentiment analysis in extracting meaningful information from social media data and supporting applications in marketing, public opinion monitoring, customer relationship management, and social research.
Keywords — Sentiment Analysis, Opinion Mining, Twitter, Social Media Analytics, Machine Learning, Naïve Bayes Classifier, Text Mining, Natural Language Processing (NLP), Classification, Artificial Intelligence (AI).

1. INTRODUCTION

The rapid growth of social media platforms has transformed the way people communicate, share opinions, and express their views on various topics. Platforms such as Twitter, Facebook, Instagram, and other online social networks generate enormous amounts of user-generated content every day [1], [2]. This content contains valuable information regarding public opinions, customer preferences, political views, social trends, and market perceptions. Extracting meaningful insights from such large volumes of textual data has become an important research area in the fields of Artificial Intelligence (AI), Machine Learning (ML), and Natural Language Processing (NLP) [3], [4]. Sentiment Analysis, also known as Opinion Mining, is a computational technique used to identify, extract, and classify emotions, opinions, and attitudes expressed in textual data. The primary objective of sentiment analysis is to determine whether a given text expresses a positive, negative, or neutral sentiment [5], [6]. Sentiment analysis has gained significant attention due to its wide range of applications in business intelligence, customer relationship management, product review analysis, political forecasting, social media monitoring, healthcare analytics, and public opinion assessment [7], [8].

Among various social media platforms, Twitter has emerged as one of the most influential sources of real-time information and public opinion. Twitter allows users to share short messages known as tweets, making it an effective medium for expressing thoughts, experiences, and reactions toward products, services, events, and social issues [9], [10]. Millions of tweets are generated daily, creating a rich repository of data that can be analyzed to understand public sentiment and emerging trends. Organizations and businesses utilize Twitter sentiment analysis to evaluate customer satisfaction, monitor brand reputation, improve products and services, and develop effective marketing strategies [11]. The analysis of Twitter data presents several challenges due to the informal nature of tweets. Social media text often contains abbreviations, slang words, misspellings, hashtags, mentions, emojis, and special characters, making sentiment classification a complex task [12]. Therefore, effective preprocessing and feature extraction techniques are essential for improving the performance of sentiment classification models [13]. Machine Learning techniques have become increasingly popular for sentiment analysis because of their ability to learn patterns from large datasets and accurately classify textual information. Among these techniques, the Naïve Bayes Classifier is widely used due to its simplicity, computational efficiency, and effectiveness in text classification tasks [14]. The Naïve Bayes algorithm is a probabilistic learning method based on Bayes' theorem, which assumes independence among features and provides robust performance for sentiment classification problems [15].

This research focuses on the classification of Twitter sentiments using a Machine Learning approach based on the Naïve Bayes Classifier. The study involves preprocessing Twitter data, extracting relevant textual features, and applying the Naïve Bayes algorithm to classify tweets into positive, negative, and neutral categories. The performance of the proposed model is evaluated using standard metrics such as accuracy, precision, recall, and F1-score. The objective of this work is to develop an efficient sentiment analysis framework capable of extracting meaningful information from Twitter data and providing accurate sentiment predictions. The outcomes of this research can support organizations, businesses, policymakers, and researchers in understanding public opinion and making informed decisions based on social media analytics.

2. Proposed Methodology

While the Bayes theorem determines the likelihood of one event occurring given the chance of another event occurring, Naive Bayes alters the procedure by "naively" assuming that each event is conditionally independent of the others. Instead, the following equation can be applied for independent events:

$$P(\text{ spam}|\text{viagra, enlarge }) = \frac{P(\text{ spam })*P(\text{ viagra} \mid \text{spam })*P(\text{ enlarged spam })}{P(\text{ viagra })*P(\text{ enlarge })} \quad (1)$$

The good news is that the equation grows linearly with each extra variable, allowing you to process massive amounts of data at once without any issues.Because of its simplicity, Naive Bayes is a fast and scalable method that outperforms more complex models as long as your data set doesn't get too large If you meet data with a variable with zero probability, Naive Bayes will cause your equation to collapse when multiplied with the other variables. This can be fixed, though, if you smooth the data beforehand, removing zero probability.

Let's imagine you have to decide whether or not to go to the beach. Our minds can examine all of the pros and disadvantages of travelling in a fraction of a second, however if a computer were asked to make the decision for us, it would need to follow a set of rules to arrive at the most beneficial answer. To make the problem easier to understand, the machine will calculate the outcome using three different variables (weather, air temperature, and where friends are going).The following principles are used to construct a relationship between the different variables so that the computer can make sense of it. The rales are read in order from top to bottom.

The variables of the row in the table above are compared to the criteria of each set of rules. When a rule's conditions are met, the final value is determined, and the remaining rules are not controlled.

Table 1: A couple of scenarios used to decide if you want to go to the beach or not.

Soenario ID	Weather	Air temperature	Friends are going
1	Sunny	20°C	Yes
2	Cloudy	23°C	Yes
3	Rainy	17°C	No
…..	...	...	...
n	Rainy	26°C	Yes

Here's an example of a rule set for the table 1. If Weather = Sunny and Friends are going = Yes then Will enjoy the beach = Yes Else If Weather = Sunny and Air temperature ≥ 20°C then Will enjoy the bench = Yes Else If Weather = Cloudy and Friends are going = Yes then Will enjoy the beach = Yes Else If Weather = Rainy then Will enjoy the beach = No Else Will enjoy the beach = No

Let's imagine you have to decide whether or not to go to the beach. Our minds can examine all of the pros and disadvantages of travelling in a fraction of a second, however if a computer were asked to make the decision for us, it would need to follow a set of rules to arrive at the most beneficial answer. To make the problem easier to understand, the machine will calculate the outcome using three different variables (weather, air temperature, and where friends are going).

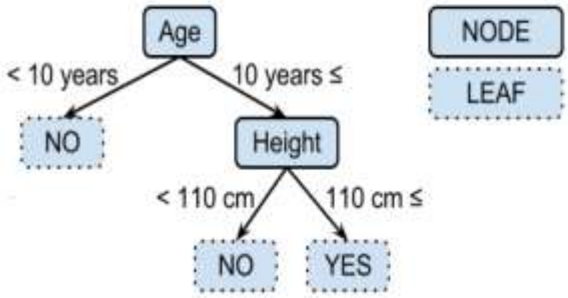

Figure 1: Decision Tree Algorithm

Classification is a technique used in data mining and machine learning to determine which categorical class a data instance belongs to. Figure depicts a categorization example, in which data is split down into individual components until a precise classification is identified. It could be anything from a single data point to a 500-page book that needs to be categorised. This strategy is frequently combined with supervised learning. By assessing one or more sets of labelled training data and utilising the data's unique properties, a classification model that appears to identify the correct class can be developed. This classification model can then predict which class they belong to using new data that is equivalent to the training data. Some of the most prevalent classification algorithms include Bayes/Naive Bayes, Decision trees, neural networks, Rule-based techniques, and Support Vector Machines.

Step-1: Preparing The Test Set

Step A.1: Getting the authentication credentials

Step A.2: Authenticating our Python script

Step A.3: Creating the function to build the Test set

Step-2: Preparing the Training Set

Step-3: Pre-processing Tweets in the Data Sets

Step-4: Naive Bayes Classifier

Step D.1: Building the vocabulary

Step D.2: Matching tweets against our vocabulary

Step D.3: Building our feature vector

Step D.4: Training the classifier

Step-5 Testing the Model

A comma-separated values file with tweets and their corresponding sentiments is provided. As an example of the training dataset, the tweet id is a unique integer that identifies a specific tweet, and the tweet sensation is either 1 (positive) or 0 (negative). Our model will be cross-validated using just this one set of data.

Another way, a Naive Bayes classifier assumes that the presence of one particular feature in a class is independent of any other feature. For example, if a fruit is red, round, and about 3 inches in diameter, it may be referred to as an apple. Despite the fact that some of these characteristics are interdependent or predicated on the existence of others, they all work together to increase the likelihood that this fruit is an apple, hence the name "Naive."For large datasets, the Naive Bayes model is a great option. Naive Bayes is known to outperform even the most sophisticated classification methods due to its simplicity.

If P (B) is greater than zero, then the Bayes Theorem applies to events A and B.

$$P(A \mid B)=(P(B \mid A)\, P(A))/(P(B)) \qquad (2)$$

Predicting a member of class (c, target) with a posterior probability P (A|B) (x, attributes) .A class's prior probability is denoted by the notation P (A). It is the probability of the predictor given the class that is P (B|A). The prior probability of predictor, P (B), is given by this expression.Predicting the type of test data set is simple and quick. It's also good at predicting multiple classes at once. A Naive Bayes classifier performs better than other models like logistic regression when the assumption of independence holds, and less training data is required as a result of this improved performance. With categorical inputs as opposed to numerical ones, it performs admirably (s). The assumption of a normal distribution is made for numerical variables (bell curve, which is a strong assumption).This means that the model will fail to predict anything if a categorical variable has a category that was not observed in training data. "Zero Frequency" is a common term for this. We can use the smoothing technique to solve this problem. Laplace estimation is a simple smoothing method. Laplace estimation is one of the simplest methods. As an alternative, the output of predict proba should not be taken too seriously because naive Bayes is a bad estimator. The assumption of independent predictors is yet another flaw in Naive Bayes' approach. In the real world, it is almost impossible to get a set of independent predictors.

Classification algorithms that use the Gaussian distribution assume that features have a normal distribution.It is used to count discretely. Suppose, for the sake of argument, that we are faced with the challenge of text classification. Instead of "word occurring in the document," we have "count how often word occurs in the document," and we can think of it as a "number of times outcome number x i is observed over the n trials." Bernoulli trials are a step further. Bernoulli: If your feature vectors are binary, the binomial model is useful (i.e. zeros and ones). Using the "bag of words" model, the 1s and 0s represent "words are present" and "words are absent" in a document, respectively.

3. RESULT ANALYSIS

The data has been gathered in the sentiment140 dataset. It contains 1,600,000 tweets extracted from the Twitter API. The tweets have been labelled (0 = negative, 4 = positive) so that sentiment may be determined. Another essential phase in the sentiment analysis process is feature extraction, which entails processing tweets and creating a word cloud based on sentiment. Figures 2 to 4 depict the various stages of the procedure.

	polarity	text	processed_tweets
[illegible]	0	@MOCAShop Wow, normally a Ferrell fan. This ma...	[illegible]
[illegible]	0	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	0	@Rayuen yeah a couple like randall munroe of ...	[illegible]
[illegible]	[illegible]	I feel for the families of the plane crash dis...	feel family plane crash disaster one
[illegible]	0	i dont feel well today	dont feel well today
[illegible]	0	[illegible]	[illegible]
[illegible]	0	CANT READ	ant read
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	0	just got up and i have a toothache	ust got toothache

Figure 2: Feature Extraction

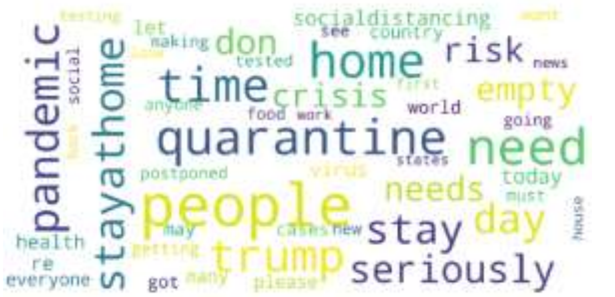

Figure 3: Word Cloud Formation for Negative Tweets

Figure 4: Word Cloud Formation for Positive Tweets

Table 2: Analysis of Performance Parameters for the Sentiment Analysis

Parameters	Value
Accuracy of model on training data	84.44%
Accuracy of model on testing data	78.77 %
Precision	0.80 (Negative) and 0.71 (Positive)
Recall	0.97 (Negative) and 0.23 (Positive)
Macro average Accuracy	75 %
Weighted Average Accuracy	77 %

Table 3: Comparative Analysis of Accuracy

Parameter	Random Forest Method	Decision Tree Method	Proposed Method (Naïve Bayes Classifier)
Accuracy of model on testing data	76 %	75.89 %	84.44 % on Training 78.86 % on Testing of Unknown Data

For classification analysis and sentiment prediction from dataset analysis, a Naïve Bayes based classifier is used. Figures of merit, such as the confusion matrix illustrated in figure 5, have been used to express the process. The procedure was carried out in Python using the Jupyter notebook software. Classifying a large number of English tweets about specific items into positive and negative emotions is now possible thanks to the research presented in this paper. Excellent accuracy is achieved when sentiment features are used instead of traditional text categorization. A business group can use this method to rank sentiment classifiers that are acceptable and to help them plan for the product's future business development.

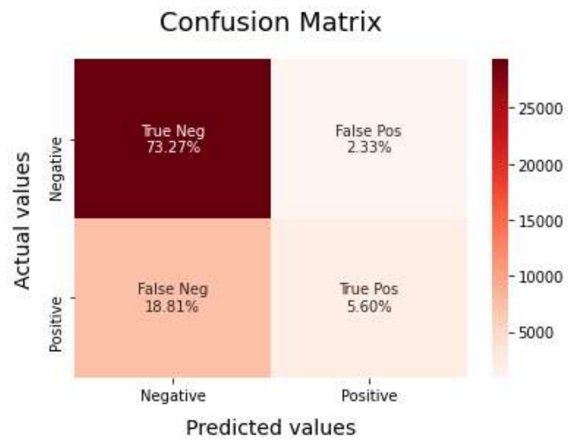

Figure 5: Analysis of Confusion Matrix by Naïve Bayes Classifier

5. CONCLUSION

Sentiment analysis can determine whether a piece of text has a good, negative, or neutral sentiment. Sentiment analysis is a subtype of natural language processing that includes information extraction. The choice of algorithms is an important component of a data scientist's job. The most effective technique is frequently to test a wide range of algorithms. Machine learning-based sentiment analysis algorithms are projected to be the most effective since they can be tuned to a specific type of data, such as tweets or reviews. Machine learning methods, on the other hand, require much larger datasets than the emotion lexicon algorithm or off-the-shelf algorithms. There should also be a set of tweets for practise.It should be observed that the three types of tweets in the training set were split unevenly: good, negative, and neutral. There were very few negative or indifferent comments. If the data had been dispersed more evenly, the machine learning systems would have found that the vast majority of tweets were positive and would have come to rely on that assumption for every tweet they received. There are various methods for classifying product criticism (which can take the form of tweets) based on the criticisms expressed in Twitter to determine whether the massive behaviour is positive, negative, or neutral, and then using that information to evaluate the product market. Using data from Twitter, we were able to rate the "satisfied" sentiment classifier for online product evaluations. It's possible to compare different classifiers for classifying a large number of English tweets about specific products into positive and negative thoughts thanks to this research. When sentiment features are used instead of standard text categorization, excellent accuracy is achieved. Using this method, companies can rank the most acceptable sentiment classifiers and use the results to help them develop long-term product strategies.

REFERENCES

[1]. H. Sharma and R. Ajmera, “Comprehensive review and analysis on machine learning based Twitter opinion mining framework,” Tuijin Jishu/Journal of Propulsion Technology, vol. 44, no. 5, 2023.

[2]. M. K. Jha, G. K. Soni, G. Jain, S. Tiwari, K. Gupta and B. Singhal, "Comparative Analysis of Classical Machine Learning Models for Twitter Sentiment Classification," 2026 5th International Conference on Communication, Computing and Electronics Systems (ICCCES), pp. 1949-1954, 2026.

[3]. D. Saxena, J. Sharma, G. K. Soni, Y. Rao, S. Sharma and S. Lavania, "Sentimental Analysis and Forecasting using Machine Learning Algorithms," 2025 4th International Conference on Automation, Computing and Renewable Systems (ICACRS), pp. 917-921, 2025.

[4]. R. Ajmera, A. Johari, A. Goyal, A. Purohit, A. Kumar and J. A. Ashok, "Multilingual Sentiment Analysis Based on Fine-Tuned Transformer Architectures," 2026 5th International Conference on Communication, Computing and Electronics Systems (ICCCES), p. 1589-1592, 2026.

[5]. S. Kumari, S. Sharma, A. Johari, G. K. Soni, H. Kaushik and S. Jain, "Twitter Sentiment Analysis using Machine Learning Techniques," 2026 6th International Conference on Expert Clouds and Applications (ICOECA), pp. 1711-1717, 2026.

[6]. M. R. Ibánez, A. C. Ventura, F. C. Mateos, P. M. C. Jiménez, “A review on sentiment analysis from social media platforms ”, Expert Systems with Applications, Vol. 223, 2023.

[7]. S. Thapar, G. K. Soni, H. Kaushik, R. Singh, S. Bisht and S. K. Bansal, “A Comparative Machine Learning Framework for Detecting Fake Accounts on Facebook,” 2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), pp. 1567–1571, 2025.

[8]. M. k. Jha, R. Ranjan, G. K. Dixit and K. Kumar, "An Efficient Machine Learning Classification with Feature Selection Techniques for Depression Detection from Social Media," 2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI), pp. 481-486, 2023.

[9]. Y. Sharma, A. Dadhich, M. Sharma, D. S. Rao, R. Ajmera and S. Soni, "Enhanced Smart Retail Decision Through Sentiment Analysis Using Machine Learning," 2026 International

Conference on Signal, Systems, and Computing for Next-Gen Automation (ICSSCNA), pp. 1731-1736, 2026.

[10]. K. A, S. K. A, N. K. V and K. V, "Analyzing Sentiments: A Comprehensive Study of Roberta-Based Sentiment Analysis on Twitters," 2024 4th International Conference on Advancement in Electronics & Communication Engineering (AECE), pp. 626-630, 2024.

[11]. Yili Wang, Jiaxuan Guo, Chengsheng Yuan, Baozhu Li, "Sentiment Analysis of Twitter Data", Applied Sciences, Vol. 12(22), 2022.

[12]. S. A. Phand and J. A. Phand, "Twitter sentiment classification using stanford NLP," 2017 1st International Conference on Intelligent Systems and Information Management (ICISIM), pp. 1-5, 2017.

[13]. J. Ferdoshi, S. D. Salsabil, E. R. Rhythm, M. H. K. Mehedi and A. A. Rasel, "Unveiling Twitter Sentiments: Analyzing Emotions and Opinions through Sentiment Analysis on Twitter Dataset," 2023 Computer Applications & Technological Solutions (CATS), pp. 1-7, 2023.

[14]. H. Parveen and S. Pandey, "Sentiment analysis on Twitter Data-set using Naive Bayes algorithm," 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), pp. 416-419, 2016.

[15]. T. Mantoro, M. Merdianti and M. A. Ayu, "Sentiment Analysis of the Papuan Movement on Twitter Using Naïve Bayes Algorithm," 2021 IEEE 7th International Conference on Computing, Engineering and Design (ICCED), pp. 1-5, 2021.