Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766180

EMOTIONAL ANALYSIS ON DYNAMIC DATASET USING TWITTER API WITH MACHINE LEARNING ALGORITHMS

B.V Pranay Kumar[1] & Prof. Manchala Sadanandam[2]
[1]Department of Computer Science and Engineering, Kakatiya University, Warangal India.
[2]Department of Computer Science and Engineering, Kakatiya University, Warangal India
E-mail address: pranaybv4u@gmail.com[1], sadanb4u@gmail.com[2]

Abstract: With the advent of modern technology and innovations, the entire world is undergoing a radical transformation. The Internet has become essential for everyone, and the world wide web is used in almost every field. Affordable access to web content and mobile and other devices allows most people to participate actively in and review various activities on the internet. Human life is full of emotions and opinions. People communicate with their fellow beings using opinions and emotions over various issues. Sentiment analysis or opinion mining identifies the emotions expressed in texts on a varied range of subjects. The field of sentiment analysis has paved the way for easy preprocessing of reviews for analysis and using the proper classification algorithms to validate the usage of the classifier and predict the overall nature of sentiment expressed on topics like politics, products, movies, and other daily social problems. Twitter has grown in popularity as a popular micro-blog application where customers can express themselves. Twitter data investigation and analysis is a field in which the research community and industry have given more attention during the last decade. Tweets are analyzed and the polarity of the content of the expression is studied to know the intricacies of the emotions of people over various subjects. As a result, this research paper investigates sentiment classification algorithms: The decision tree and Naïve Bayesian classifiers on Twitter data and their outcomes. We have used a dynamic data set of tweets using Twitter API calls with 200 tweets per call. In this Textblob is used for calculating the polarity. The experiments are carried out using different classifiers and achieved good performance. We have been fortunate to achieve 90.79% accuracy and the F1-Score was 0.88 using Naïve Bayes classifier on this Twitter data stream.

Keywords: API, Twitter analysis, Sentiment analysis, Descriptive Statistical Analysis.

1. Introduction

Today, with easy accessibility to the Internet, mobile phones, and other electronic gadgets, every household has access to the Internet. These opinions have value for organizations and companies to have insights about their products and also help them customize people's needs. It also helps governments understand the heart, and needs of people with genuine feedback on the policies of the government, and requirements of the people to frame new policies or schemes. This holds the same for the industry to know authentic information about the products [1]. Researchers can extract data across various platforms where users' subjective sentiments are embodied in the expressions. It improves the accuracy with which the users' expression is understood. More or less every user's sentiment varies depending on their values, contexts, or interests. This is an important indicator for examining user tendencies. Using sentiment analysis, we can provide the best experience.

1.1. Motivation for sentiment Analysis

The following are bountiful reasons for choosing Twitter data for sentiment mining. Some of them are:

Formal people, actors, politicians, entrepreneurs, industrialists, and other leaders in society pertaining to education, religions, communities, and social workers use Twitter to express their opinions. Twitter has over 500 million tweets per day, which is considered huge data for sentiment mining.

With the advancement of mobile communication social media platforms like Twitter have attracted people from all walks of life from celebrities to ordinary people where the opinions expressed are views of all classes and age groups.

Many web browsers use the Twitter application, which has over 50 million downloads.

In most countries, the psychology of a customer before purchasing any product is to know what others feel about the product or service they are going to buy. They want to know the feedback or opinions of the product. This helps the prospective buyer to finalize his purchase. In the context of this, sentiment analysis helps us communicate our emotions and opinions about varied aspects of life.

In general, sentiment analysis of the textual content will be evaluated categorically on three levels [2] based on the scope of its work and analysis in action. Firstly, analysis at the document level, phrase level, and sentence level. Document level & sentence level do not express the overall sentiment in a single word whether they like it or not, but they evaluate various features accessible in the text.

1.2. Approaches for sentiment analysis

Analysis of text today starts by collecting the text over social media platforms. We have a very easy way to access data for research and analysis tasks over social media like Twitter. Various approaches to executing the sentiment analysis tasks are shown in fig 1 below.

- Lexicon or Dictionary-based approach
- ML-Based Approach
- Combo or Hybrid approach

Lexicon-based approach or unsupervised approach: Lexicon-based approach is also known as the bag of words approach where there are predefined words or sentences classified into sentiment labels or scores, like positive sentiment, negative sentiment, or neutral sentiment. Here, the number of words is limited, as preparing an exhaustive list of words is a limitation of this approach. The words found in the dataset are mapped with words in the target lexicon or dictionary and the overall scoring or sentiment is calculated.

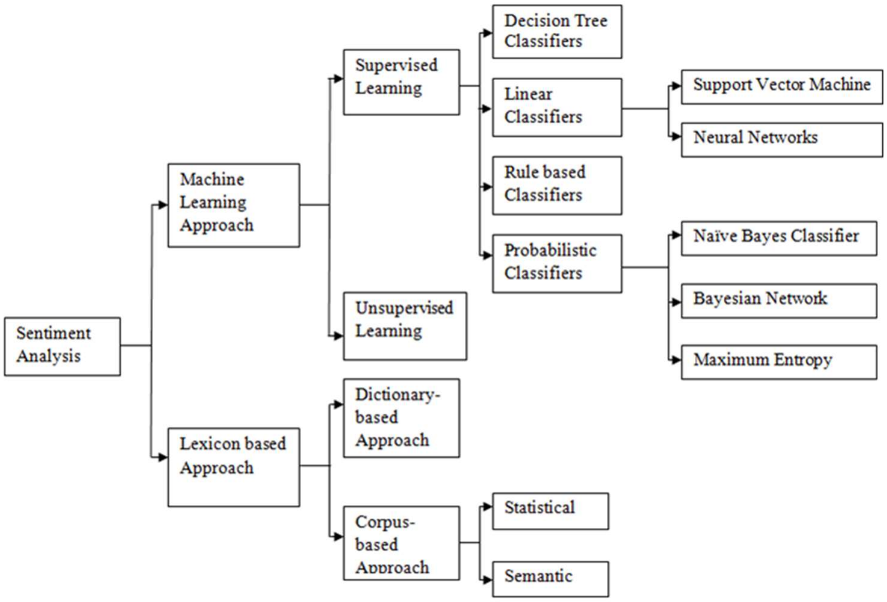


Fig.1. Sentiment Analysis Techniques

Machine learning approach: This is a branch of artificial intelligence where a mathematical model is trained using the training data and tries to evaluate the model using test data. Various popular approaches are broadly classified into supervised and unsupervised approaches based on the presence of class labels.

Hybrid approach: It is the combination of approaches that we perform lexicon and machine learning approaches. [3] And [4] are various hybrid approaches.

1.3. Threats and difficulties in sentiment analysis

Some of the major confrontations in the analysis of text are because of some linguistic issues to analyze the polarity of the text opinion [5]. The challenges include:

Reordering the words of sentences can give large variations in the analysis results.

The ambiguity of English language words that have different meaning based on the context in which it is used. As an example, the word “miss” is used as a salutation for women and miss indicates the lack of something.

Use of internet slag. Messages on the internet are very short with unpopular acronyms and emoticons.

Use of emoticons and unpopular metaphors is challenging for analysis.

2. Literature Survey

Opinion Mining or Sentiment analysis is the recent technology of NLP, an application of AI and ML. Sentiment analysis is used for extracting the emotion and opinion analysis of views of people over social media platforms. Every analysis begins with the extraction or acquisition of data. We are fortunate enough some social platforms like Twitter and Amazon provide an open and easy way to acquire the tweets and reviews shared by customers.

Tweets from the social media platform Twitter can be collected for analysis using the Twitter developer account and python library Tweepy. Amazon allows reviews to be extracted using the library beautiful soap (bs4) available in python.

Khaled Ahmed et al., [6] proposed the need for mining social media to get valuable opinions and emotion mining. The authors address several open challenges in the field of sentiment analysis, such as developing multilingual classifiers, creating common user profiles across multiple social media platforms, and employing unsupervised learning in classification to improve prediction accuracy.

Rohan Srivastava et al. [7] proposed capital market forecasting using sentiment analysis to predict the market sentiment at any particular moment and make strategic decisions to invest in the stock market and make profits.

Povado et al. [8] discuss sentiment analysis using a support vector machine on big data. Here the classifier is applied to a huge amount of data with less preprocessing data and achieves reasonably good accuracy.

Aluvalu Rajnikanth et al. [9] in the paper Twitter-based market analysis using cloud statistics evaluate 100,000 tweets and found 2,524 missing values for analysis and use statistical analysis for preprocessing of Twitter data and predict the customer investment analysis for better profits in the stock market.

RK Bakshi et al. [10] proposed the importance of Twitter [11] micro-blogging platform for gauging the public mood. In this paper, the authors undertake a stepwise methodology to understand a person's tweets on the stock prices of Samsung Pvt Ltd. An algorithm was designed to predict the fluctuations in stock prices and make an effective decision for better profits during investments. Sentiment analysis classification research scaled from document-level classification to sentence level and phrase level [12-14].

A. Go, R Bhayani et al., [15] Proposed Twitter sentiment classification using distant classification supervision. Messages in this paper are classified using query terms on data with emoticons. Emoticons in the tweets are used in this paper and they are labeled as noisy data. Accuracy of 79% is achieved using Naïve Bayes and SVM.

Efthymios Kouloumpis et al. [16] Proposed sentiment analysis of Twitter messages using linguistic features. For training and test results hashtagged dataset was used. For structuring Twitter data supervised approach was applied with hashtags present in the tweets. The semantics of tweets was the additional feature used for sentiment analysis.

A. Goel, J. Gautam, et al. [17] proposed sentiment analysis on sentiment140 training data using naive Bayes. SentiWordNet and naive Bayes were used by the authors. In this, the authors used NLTK and Twitter APIs of Python.

Minara P Antony et al. [18] in the research paper recognize stop words and separate the sentiment terms. To calculate the overall sentiment unigram method was used. The sentiment analysis analyzed the data on mobile phones with 80% accuracy.

In [19], the authors investigated reviews of products available on Amazon. The authors proposed the unigram and bigram models. The accuracy of the review of the products in this paper varies greatly depending on the items: 93.5% for mobile phones and their accessories, 93% for physical science, and 94.02% for musical instruments. Umme Aymun Siddique et al. [20], proposed feature sets and ML techniques for the analysis over Twitter data streams.

In [21], the authors investigated the success or opinion of people on the movies as discussed by people on the microblogging site Twitter. Unigram and bigram models are used with an accuracy of 86.2%.

Wolny, Wieslaw, et al., in [22] predicts the sentiments of people on the Twitter dataset. The unigram and bigram models were investigated in this paper. The trigram model was not used, but to improve accuracy a hybrid of all models called N-Gram was used. However, during feature extraction emoticons with their contextual meaning and synonyms are used to improve accuracy. In this paper, the accuracy achieved was 81.3% and no acronyms were used.

N Veereanjaneyulu et al [23] investigated sentiment analysis on the Twitter data set where the bigram model was used during feature extraction time. In this paper, emoticon features are used however synonyms and acronyms are not used to improve accuracy; however, considering all such features during feature extraction will result in high accuracy.

3. Classification Methods

In literature, there are popular methods to deal with classification problems. Sentiment analysis is also a classification problem like a pattern recognition problem. Classification algorithms in machine learning use input data to predict the classification that data will fall into one of the predefined categories of sentiment like positive, negative, and neutral class labels.

In the process of classification of sentiment on the dataset, some portion of the dataset is used for training, and the rest of the dataset is used for testing. Various supervised algorithms in the literature are SVM, Decision tree algorithm, KNN, and Naïve Bayes.

3.1. Naïve Bayes Classification Algorithm

Naïve Bayes is a mathematical model popularly used for constructing classification algorithms. In this, the problem instances were assigned class labels based on the vector of feature values. The class labels come from finite classes. Naïve Bayes classifiers assume that the features in the class are identical and the value of one feature in a class is independent of other features in another class given a class variable. For instance, a fruit under question may be orange if its color is orange in color, the shape is round with a diameter of around 8 cm. Here each of these features' color, size, and shape of the fruit under question contribute independently to confirm that the fruit is orange irrespective of any correlation between various features of different fruits under consideration. Naïve Bayes classifier uses a mathematical probabilistic classifier that prognosticates the probability of contingency of a feature independent of the phenomenon of other features present in it.

$P (A/B) = P(B/A). P(A) / (B)$

3.2. Decision Tree Classifier (DT)

Decision trees are a supervised learning model used for classification and regression problems DT is a non-parametric problem. A tree is a diagram of a piecewise constant approximation. A Decision tree uses a divide and conquers strategy using a greedy search for finding those optimal split points within a tree.

In the DT classifier [25], features are marked using the interior nodes, and all the edges leaving these nodes were named as a trial on the data set weight. The leaf nodes of the tree project the possible outcome within a dataset. The complete document is classified from the tree's root node and moves down successfully to its branch nodes as you reach the leaf node. In a decision tree, learning is achieved as the model for anticipation in which it maps the information of an item to the conclusions of that item's expected value. This kind of classifier requires small data preparation.

The decision tree used in the Twitter data is shown in Figure [2]

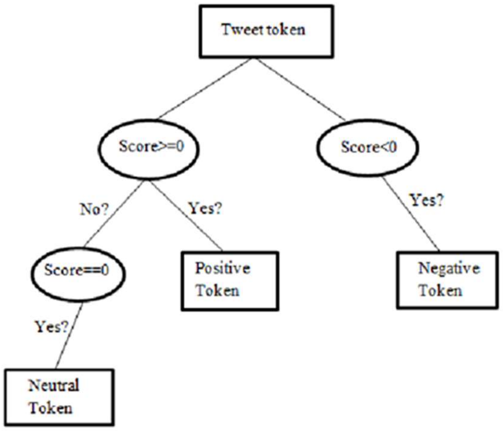


Fig.2. Decision Tree used in Twitter data

Pros of a DT Classifier: A decision tree classifier is very simple to understand and lay hands on. A decision tree can be visualized. It necessitates minimum data preparation. The number of data points used in training a tree decides the cost of the tree and is linearly proportional. DT With multi-output points, it can handle both categorical and numerical data. However, it produces extremely complicated trees that do not generalize the problem well. Learning an ideal decision tree is recognized to be an NP-complete issue.

3.3. Classification using the Classifier Support Vector Machine (SVM)

SVM is a decent choice for the speech classifier issue since it produces accurate results. For the most trained examples, a hyperplane is constructed with the maximal Euclidean distance. The SVM hyperplane [26] is resolved using a small portion of the learned datasets called support vectors. The qualified classifier is not available for the residual of the training datasets. The classifier SVMs successfully have been applied to scattered text categorization and have also been used in various sequence processing applications. Because SVMs do not require a labeled training dataset, they are commonly employed in hypertext and text classification. The support vectors and decision boundaries are as shown in the figure [3]

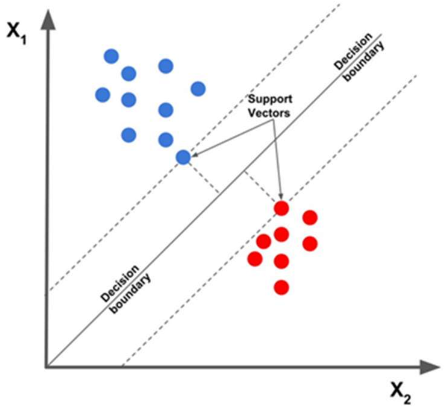


Fig.3. Support Vector Machine

3.4. Classification using K- Nearest Neighbor Classifier

KNN [28] is an unsupervised learning technique used for classifying textual data. The data items are categorized using several training data sets and their proximity among data items is used in this technique. This algorithm's benefit in text categorization is its simplicity as shown in fig 4. It moreover works well with text classification with several classes. The fundamental disadvantage of K-Nearest Neighbor is that it takes a long time to categorize entities when large data collection is involved.

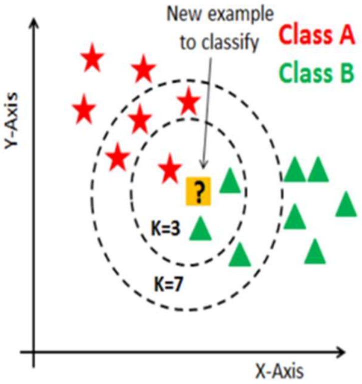


Fig.4. K Nearest Neighbor

4. Proposed Method

The proposed methodology as shown in Fig [5] is to identify the target API of Twitter and make the collection of data using Twitter calls for container data. Once data is collected, it is a good idea to remove useless URLs to obtain the resultant corpus for analysis. The corpus identified contains noise, which is not recommended for effective analysis, and hence needs to be removed. Tokenization and sanitization operations are performed on the data, followed by stop word filtering and negation filtering. Stemming is an additional task. The following figure shows the architectural framework for the proposed sentiment analysis. It shows the flow of various steps and processes we used for analysis.

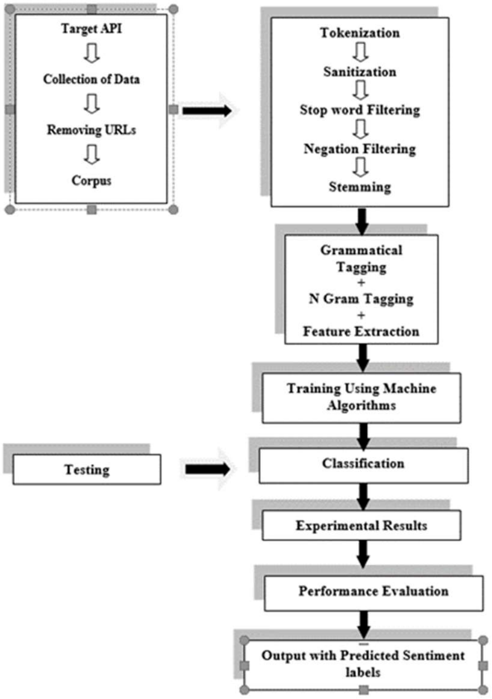


Fig.5. Architecture Framework for the proposed Sentiment Analysis

In the proposed architectural framework for sentiment analysis of Twitter data as shown in Fig 5, the first phase is to fetch data in the form of tweets by registering on the official Twitter website. Once the stream of data is collected pre-processing of tweets is performed by removing noise that is not useful for computation. Pre-processing involves the removal of URLs and special symbols, Hashtags, and white spaces to make the stream clean for processing.

In the second phase, tokenization and other transformations are performed to make a stream of data more meaningful for processing. Different tasks in this phase include tokenization, sanitization, stop word filtering, negation filtering, and stemming.

In the next phase grammatical tagging, Ingram tagging, and feature extraction operations are performed using the TextBlob functions. Once the data is ready, training of the data is performed using the machine learning classifiers like a DT and NB. Once the model has been trained, the data is tested with various classification algorithms like the decision tree classification and Nave Bayes classification techniques. The experimental findings illustrate the polarity of tweets using the decision tree and Nave Bayes classification techniques. The performance of the classification problems is assessed using recall, precision, accuracy, and F-score.

4.1. Technique Applied

The primary goal of this study is to analyze tweets based on a keyword in the popular social media handle on Twitter. On every topic, there will be many tweets on Twitter. There isn't any way to exactly predict how many of them are positive or negative. Whenever a random keyword is given as an input to the function, by using the naive Bayesian classifier, the output is obtained. The output is a value by which many insights about the tweet such as sentiment, polarity, etc. can be identified. By understanding the nature of the tweets, we can get an understanding of what type of content is in what category. In this, we considered the live tweets and the dataset which is obtained by searching for a keyword used. The tweets are passed into a function that removes special characters and tags with the help of regular expressions. The output of the paper has a consistent accuracy similar to that of the library function TextBlob. TextBlob can be used in sentiment analysis, which is how we will be using it in this paper. TextBlob calculates the polarity of the tweets and gives them a value. If the value is greater than 0, the tweet is considered positive else negative. The dataset will be auto-generated because the user gets to enter random keywords to retrieve tweets. This will be done using a Twitter developer account, to ensure there wouldn't be any security issues. A live dataset is even considered for analysis in this paper. Tags and special symbols from the tweet are removed to simplify the process. This will be done by a function using regular expressions. After the data is ridden with special symbols, tags are passed on to the TextBlob to generate polarity which lies between the bandwidth of [-1, 1] by using this we used to ascertain the sentiment of tweets that are on Twitter.

4.2. Different Phases in the Proposed Method

Twitter, the micro-blogging site achieved immense popularity and interest with people from all walks of life. It’s an effective tool for various business intelligence tasks and to get knowledge about people’s emotions towards various issues and products around the world.

The proposed method for the classification of dynamic twitter data has various phases like any other pattern recognition problem. These phases are

Requirements

The basic requirement is to propose our work on live Twitter data, we imported the Tweepy (It's a python client for using the official API of Twitter). Initial installation of Tweepy and TextBlob is performed followed by NLTK corpora tools that contain a large volume of data.

4.2.1. Fetching Twitter Data using Twitter (Target) API

The authors propose to calculate positive and negative tweets in percentage. Procedure for accessing Live data and processing.

Register the app on the official Twitter website to access the tweets.
Generate the private keys.
Copy the keys to be used further in the algorithm.
Collect the stream of data
Preprocessing of Tweets
Removal of URLs
Removal of special symbols
Removal of Hashtags and additional white spaces.

Now the corpus for sentiment analysis is ready

4.3. Tokenization of Twitter Text

Our paper proposes a method that bolsters the user to access real-world dynamic tweets on Twitter and analyze the sentiments on any given day if the tweets are positive, negative, or neutral. And also, we can see the various issues which people are most pursued daily over time. During this work, the authors have used various tools that enhance our work efficiently. The most important toolkits that we have made use of are the Natural Language Tool Kit (NLTK), Tweepy, and TextBlob.

4.3.1. Sanitization is the process of cleaning text from noisy data by identifying words or tokens that will not help in computation and just add complexity to the analysis.

4.3.2. Stop word Filtering: Stop words are commonly occurring words in corpora that provide no or little information with unnecessary noise and therefore need to be removed but with little care and without changing the interpretation of text under analysis.

4.3.3. Stemming: It is the process of reducing words to their root form. Rain, rain, rained all have a very similar meaning. Stemming also reduces noise and helps in dimensionality reduction. Lemmatization is the process of aggregating together the derived forms of a word into a single element or item. This functionality is made possible by the "lemmatize" property. Lemmatization is the most popular technique for textual data in NLP and ML during the pre-processing phase. Stemming is a Natural Language Processing concept that is equal to this. In both stemming and lemmatization, we strive to reduce a given term to its root word. The root word is regarded as a stem in the steaming process and a lemma in the lemmatization process. The advantage of lemmatization is that it is more precise.

4.4. Grammatical Tagging

This grammatical tagging also called POS tagging is the process of identifying Parts of speech on words based on definition and context.

4.4.1. N Gram tagging: N-grams (N>1) are a combination of multiple words that are more informative than mere words. TextBlob consists of a function namely grams that returns a tuple of n consecutive words.

4.4.2. Feature Extraction: After completion of all the above preprocess steps the corpora is ready for analysis with useful features extracted for computation.

4.5. Training Using TextBlob Algorithms

Python library used for text processing is TextBlob. It offers a straightforward Application Program Interface for delving into typical NLP activities like tagging various parts of speech recognition, extraction of noun phrases in the sentences, analysis of sentiment in text, classifying the polarity, sentence translation, etc. TextBlob is a major solution available in NLTK and pattern recognition problems, and it functions extremely well with both.

To develop custom classifiers, we employ a text blob classifier. First, we automatically generate some training and test data.

4.6. Classification

The Naïve Bayes classification makes two assumptions in simplification. Firstly, the features in the document encode word identity, not position. Secondly the conditional independence assumption:

$$p(C_k \mid x_1, \dots, x_n) = \frac{p(x_1, \dots, x_n)}{p(C_k)p(x_1, \dots, x_n \mid C_k)}$$

according to the "naive" conditional independence assumptions, for the given class Ck each feature of vector xi is conditionally independent of every other feature ***Xj*** for ***I≠j***.

$$p(x_i \mid C_k, x_1, \dots, x_n) = p(x_i \mid C_k)$$

Thus, the relation can be simplified to,

$$p(C_k \mid x_1, \dots, x_n) = \frac{p(x_1, \dots, x_n)}{p(C_k)\prod_{i=1}^{n} p(x_i \mid C_k)}$$

Since ***P(x₁, …, xn)***is constant, if the values of the feature variables are known, the following classification rule can be used:

$$P(C_k \mid x_1, \dots, x_n) \propto P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k)$$

$$\Downarrow$$

$$\hat{y} = \underset{k}{\operatorname{argmax}} P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k)$$

To avoid underflow, log probabilities can be used.

$$\hat{y} = \underset{k}{\operatorname{argmax}}(\ln P(C_k) + \sum_{i=1}^{n} \ln P(x_i \mid C_k))$$

The variety of naive Bayes classifiers primarily differs from each other by the assumptions they make regarding the distribution of ***P(xi|Ck)***, while ***P(Ck)*** is usually defined as the relative frequency of class ***Ck*** in the training dataset.

A naïve Bayes classifier is created passing this training data into the constructor. In addition to the above method, data can be loaded from files such as CSV, JSON, and TSV. The classifier is then used by calling the Classify(text) function. The label probability distribution can be determined, which aids in determining the likelihood of positive, negative, or neutral sentiment. Accuracy methods can also be used to determine the accuracy of the test set.

Important steps proposed are to authenticate the Twitter account using the credentials, followed by collecting the tweets using the GET request and classifying the tweets based on their features whether positive, negative, or neutral polarity. TextBlob inherently possesses the movie dataset whose tweets are labeled with labels like positive, negative, and neutral as training data. The data contained with the TextBlob and its features are trained with classification algorithms like a decision tree and a Naïve Bayes classifier.

5. Dataset

In this, a dynamic data set is used every time we search for tweets. All tweets that contain the keyword are compiled as a dataset using Twitter-API-Calls [29] which can at most fetch a maximum of 200 tweets per one call. We also have a loaded static data set, with a collection of 3 lakh tweets that can also be used for the analysis. Text Blob calculates the polarity based on a value, if value is greater than 0, the tweet is considered positive else negative. The dataset will be auto-generated because the user gets to enter random keywords to retrieve tweets. This will be done using a Twitter developer account, to ensure there wouldn't be any security issues. A live dataset is even considered for analysis in this paper. Tags and special symbols from the tweet are removed to simplify the process. This will be done by a function using regular expressions.

A cumulative number of rows: 3,00,000
Several columns: 2
Number of rows: 200 (*at most as per a single Twitter-API-Call)
Columns: 2

6. Experimental Results and Output Summary

In statistical data, accuracy corresponds to how nearer a result is to its true value. While several methods are present to classify the tweets on social media, we proposed using Naïve Bayes and a Decision tree classifier. High accuracy can be improved on false positives using the proposed model composed of a decision tree and Naïve Bayes. In this paper, we have experimentally its evident that Naïve Bayes was 9079% accurate and the decision tree was only 77.82%.

a. Performance evaluation parameters

In the field of machine learning, the experimental investigations of the classification algorithm are better evaluated using a performance matrix called a confusion matrix. This matrix is decided by four vital varying terms such as False Negative (FN), and True Negative (TN), True Positive (TP), False positive (FP).

Where True Negative is if the outcome of the review is negative and the prediction is also negative.

True positive is if the outcome of the experimental review and its prediction as calculated is both positive. False Negative is the result of a review being positive but predicted outcome as negative. False Positive is the result is negative but predicted as Positive.

		Predicted	
		Positive review	Negative Review
Actual	Positive Review	*True Positive (TP)*	*False Negative (FN)*
	Negative Review	*False positive (FP)*	*True Negative (TN)*

Table I: Shows performance parameters.

The accuracy [30] of measurement in statistics is how nearer we approach to the true value.

$$Accuracy = \frac{(\mathrm{TP{+}T}\ \ )}{(\mathrm{TP{+}FP{+}TN{+}F}\ \ )}\ X\ 100$$

Recall or sensitivity is defined as true predicted data to all existing data that are positive.

$$Recall = \frac{\mathrm{TP}}{(\mathrm{TP{+}FN})}\ X\ 100$$

$$Precision\ = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FP})}\ X\ 100$$

F-Score is the single measure required to determine the performance of a model. It is the harmonic mean of precision and recall.

$$F - score =\ \mathrm{x}\ \frac{\mathrm{precision\ x\ recall}}{\mathrm{precision} +\ \mathrm{recall}}$$

Table II & III displays comparative results where Naïve Bayes was a better option for classification.

Prediction	**True: Yes**	**True: No**	**Class Precision**	**F-Score**	**Accuracy**
Outcome Yes	186	28	86.91%	0.64	**77.82%**
Outcome No	184	558	75.20%		
Class Recall	50.27%	95.22%			

Table II: Shows sentiment analysis with an accuracy of 77.82% using a decision tree

Prediction	**True: Yes**	**True: No**	**Class Precision**	**F-Score**	**Accuracy**
Outcome Yes	337	55	85.96%	0.88	**90.79%**
Outcome No	33	531	94.14%		

Class Recall	91.08%	90.61%

Table III: Shows sentiment analysis with an accuracy of 90.79% using Naïve Bayes classification

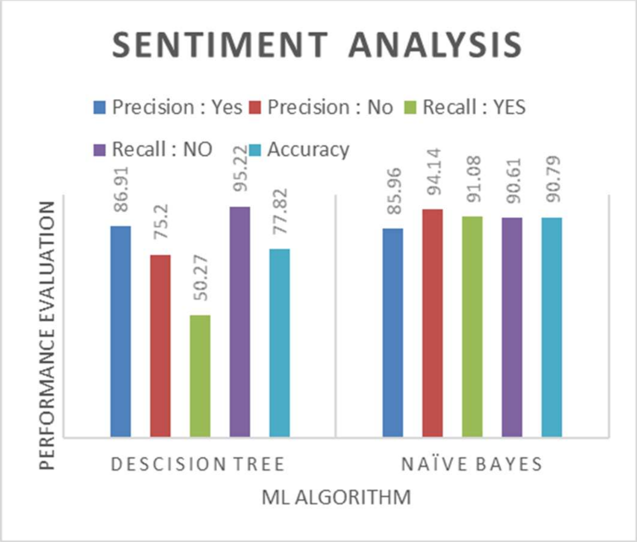


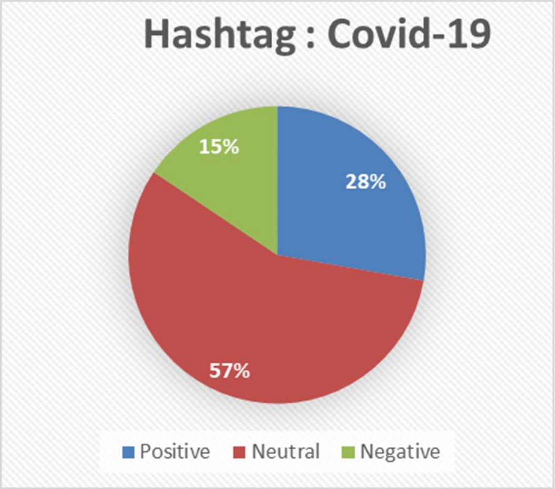
Chart I: Show the comparative analysis of the decision Tree and Naive Bayes Classification

b. Descriptive Statistical Analysis

From the above tables, it is apparent from the descriptive statistical analysis that the accuracy of the sentiment classification is 79% and the other quality factor F1-Score of the analysis is 0.64 using the decision tree classifier and the accuracy using Naïve Bayes Classifier is 90.79 % and the F1-Score is 0.88.

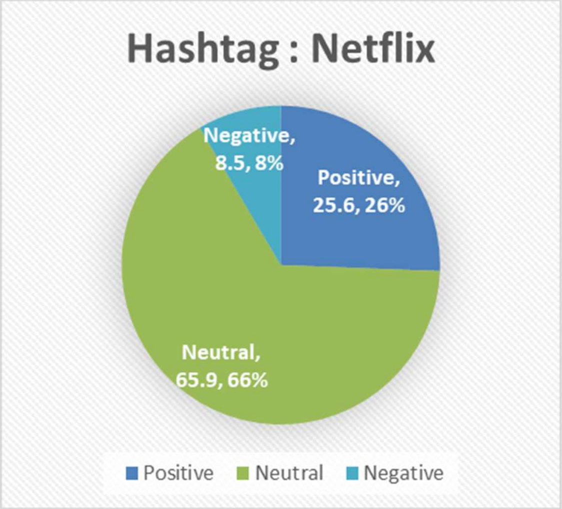
Naive Bayes classifiers have the advantage of not being prone to overfitting because they "ignore" irrelevant features. They are, however, susceptible to poisoning; a phenomenon that occurs when we are attempting to predict a class and features that are not characteristic of that class appear, resulting in misclassification. Naïve Bayes is highly scalable and can be implemented easily. Its computational complexity increases in a linear way with the increase of data.

Finally, the proper classification doesn't somehow necessitate a large number of data inputs, although more data results in more accurate calculations. Several enhancements to this algorithm can be made by incorporating additional data preprocessing and statistical techniques. Naive Bayes is strongly associated with text-based classification.



PieChart 1: Trend of Topic Covid

Pie chart I shows of the 800 total tweets taken on the subject Covid-19 its show 56.7% are neutral about the issue and 27.8% are Positive and 15.6% are Negative



Pie Chart II : Shows the trend of NETFLIX

Pie Chart II Shows the percentage of Tweets on the topic Netflix. Total Tweets 2000 of which 65.9% are Neutral, 25.6 are positive, and 8.5% are Negative.

7. Conclusion

People today want to express their opinions and emotions over conglomerated issues and personalities via social media platforms such as Twitter, which has both advantages and disadvantages. However, few Twitter users are using these platforms to spread hatred and disrespect toward specific communities, religions, celebrities, and politicians. As a result, this paper effectively assists police and other government agencies in understanding what types of emotions are being spread on Twitter and in taking defensive measures to avoid communal riots and channeled hatred. This research article investigated emotional analysis using sentiment methods such as dictionary-based approaches, machine learning, and ensemble approaches on the micro-blog Twitter. This paper investigated Naïve Bayes and decision tree classifiers using Textblob on Twitter dynamic streaming data. Research outcomes demonstrated that machine learning techniques such as the naive Bayes achieved an accuracy of approximately 90.79% and an F1-Score is 0.88 as compared to the decision tree of 79% with an f1-score of 0.64 for performing sentiment of tweets on social media. The scope may be extended to alternative hybrid models including lexicon-based approaches and other deep learning models for better accuracy of predictions.

References

Neha S. Joshi, Suhasini A. Itkat, “A Survey on Feature Level Sentiment Analysis” (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4), 2014, 5422-5425.

P. K. Soni and R. Rambola, "A Survey on Implicit Aspect Detection for Sentiment Analysis: Terminology, Issues, and Scope," in IEEE Access, vol. 10, pp. 63932-63957, 2022, doi: 10.1109/ACCESS.2022.3183205.

Bashar, M.K. A Hybrid Approach to Explore Public Sentiments on COVID-19. SN COMPUT. SCI. 3, 220 (2022). https://doi.org/10.1007/s42979-022-01112-1.

Nan Jing, Zhao Wu, Hefei Wang, A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction, Expert Systems with Applications, Volume 178, 2021, 115019, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2021.115019.

Qianwen Ariel Xu, Victor Chang, Chrisina Jayne, A systematic review of social media-based sentiment analysis: Emerging trends and challenges, Decision Analytics Journal, Volume 3, 2022, 100073, ISSN 2772-6622, https://doi.org/10.1016/j.dajour.2022.100073. (https://www.sciencedirect.com/science/article/pii/S2772662222000273)

Ahmed, Khaled, Neamat El Tazi, and Ahmad Hany Hossny. "Sentiment Analysis over Social Networks: An Overview." Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on. IEEE, 2015.

Srivastava, Rohan, et al. "Capital market forecasting by using sentimental analysis." Next Generation Computing Technologies (NGCT), 2016 2nd International Conference on. IEEE, 2016.

Povoda, Lukas, Radim Burget, and Malay Kishore Dutta. "Sentiment analysis based on Support Vector Machine and Big Data." Telecommunications and Signal Processing (TSP), 2016 39th International Conference on. IEEE, 2016.

Gupta, S., & Aluvalu, R. (2019). Twitter-Based Capital Market Analysis Using Cloud Statistics. International Journal of Sociotechnology and Knowledge Development, 11(2), 54–60. doi:10.4018/ijskd.2019040104

Rushlene Kaur Bakshi, Navneet Kaur, Ravneet Kaur, Gurpreet Kaur, "Opinion Mining and Sentiment Analysis”, 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2016, pp.452-455.

Haque, Md. "Sentiment analysis by using fuzzy logic." arXiv preprint arXiv:1403.3185 (2014).

B. Pang, L. Lee, A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, in Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL’04) (2004), pp. 271–278

Faisal Alshuwaier, Ali Areshey, Josiah Poon, Applications and Enhancement of Document-Based Sentiment Analysis in Deep learning Methods: Systematic Literature Review, Intelligent Systems with Applications, Volume 15, 2022, 200090, ISSN 2667-3053, https://doi.org/10.1016/j.iswa.2022.200090.

A. Esuli, F. Sebastiani, Determining term subjectivity and term orientation for opinion mining 2006, in 11th Conference of the European chapter of the association for computational linguistics (2006)

A. Go, R. Bhayani, L. Huang, Stanford University, Twitter sentiment classification using distant supervision, in The Third International Conference on Data Analytics (2009)

E. Koulompis, T. Wilson, J. Moore, Twitter sentiment analysis: the good the bad and the OMG! in The Fifth International AAAI Conference on Weblogs and social media (2011)

A. Goel, J. Gautam, and S. Kumar, "Real-time sentiment analysis of tweets using Naive Bayes," 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), 2016, pp. 257-261, DOI: 10.1109/NGCT.2016.7877424.

Anto, Minara P., et al. "Product rating using sentiment analysis." Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on. IEEE, 2016.

Neha S. Joshi, Suhasini A. Itkat, “A Survey on Feature Level Sentiment Analysis” (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4), 2014, 5422-5425.

U. A. Siddiqua, T. Ahsan and A. N. Chy, "Combining a rule-based classifier with the ensemble of feature sets and machine learning techniques for sentiment analysis on the microblog," 2016 19th International Conference on Computer and Information Technology (ICCIT), 2016, pp. 304-309, DOI: 10.1109/ICCITECHN.2016.7860214

Kharde, Vishal & Sonawane, Sheetal. (2016). Sentiment Analysis of Twitter Data: A Survey of Techniques. International Journal of Computer Applications. 139. 5-15. 10.5120/ijca2016908625.

Wolny, Wieslaw. (2016). Twitter Sentiment Analysis Using Emoticons And Emoji Ideograms.

N. Veeranjaneyulu, Akkineni Raghunath, B, Jyostna Devi, Venkata Naresh Mandhala, “Scene Classification Using Support Vector Machines with LDA “journal of theoretical and applied information technology 31 May 2014. Vol. 63 No.3

Pablo Gamallo, Marcos Garcia, “Citius: A Naive-Bayes Strategy for Sentiment Analysis on English Tweets” Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 171–175, Dublin, Ireland, August 23-24 2014.

R. Bibi, U. Qamar, M. Ansar, and A. Shaheen, "Sentiment Analysis for Urdu News Tweets Using Decision Tree," 2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA), 2019, pp. 66-70, DOI: 10.1109/SERA.2019.8886788.

F. Luo, C. Li, and Z. Cao, "Affective-feature-based sentiment analysis using SVM classifier," 2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2016, pp. 276-281, DOI: 10.1109/CSCWD.2016.7566001.

X. Fei, H. Wang, and J. Zhu, "Sentiment word identification using the maximum entropy model," Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering (NLPKE-2010), 2010, pp. 1-4, DOI: 10.1109/NLPKE.2010.5587811.

A. Bayhaqy, S. Sfenrianto, K. Nainggolan and E. R. Kaburuan, "Sentiment Analysis about E-Commerce from Tweets Using Decision Tree, K-Nearest Neighbor, and Naïve Bayes," 2018 International Conference on Orange Technologies (ICOT), 2018, pp. 1-6, DOI: 10.1109/ICOT.2018.8705796.

Chouhan, Kuldeep & Yadav, Mukesh & Rout, Ranjeet & Sahoo, Kshira & Zaman, Noor & Masud, Mehedi & Aljahdali, Sultan. (2022). Sentiment Analysis with Tweets Behaviour in Twitter Streaming API. Computer Systems Science and Engineering. 45. 10.32604/csse.2023.030842.

Contreras, D., Wilkinson, S., Alterman, E. et al. Accuracy of a pre-trained sentiment analysis (SA) classification model on tweets related to emergency response and early recovery assessment: the case of 2019 Albanian earthquake. Nat Hazards 113, 403–421 (2022).