**Journal of Applied Computer Science & Artificial Intelligence**

**Research Paper Vol.-2, Issue- 1, June 2022 ISSN: 2582-8673**

TWACH: THE BIAS PLATFORM  
Twitter Sentiment Analysis

**Anjan Kumar Dey, Ashish Abhishek Minz, Ayan Chatterjee, Debajyoti Ghosh, Debraj Karmakar**

1 Complete Affiliation & Email ID [font size 10]

2 Complete Affiliation [font size 10]

3 Complete Affiliation [font size 10]

**Available online at: http://jacsai.org/**

Received: ………./2021, Revised: ………..2021, Accepted: ………2021, Published: 30/June/2021

*Abstract*— This project aims to solve the problem of sentiment analysis on Twitter. Classify tweets according to the sentiment they express: positive, negative, or neutral. We may collect data and analyze tweets by providing a username or any hashtag. The user enters the number of tweets to extract and the extraction date with the hashtag or username. If the input is not empty, using the Twitter API we pull tweets from Twitter in real-time. The NLP technique is then used to clean them. The tweets are analyzed and classified according to their polarity and subjectivity scores in positive, negative, or neutral lists and finally displayed. Analyzing public sentiment or opinion about something is important for many applications such as companies trying to find out the public sentiment of their products in the market, predicting political elections and socio-economic phenomena such as the stock market. The goal of this project is to develop a functional classifier and automatic sentiment classification of recent tweets.

Keywords— Sentiment, Tweets, NLP, Hashtag, Username, Polarity, Subjectivity

# **Introduction**

Data analysis is the process of applying organized and systematic statistical techniques to data to describe, recap, check and condense it. It is applying these techniques to modify data to suit our requirements. Data mining is a multistep process that involves collecting, organizing,

cleaning of the data. It is needed because different sources like social media, enterprise data, public data, transactions etc. generate data of increasing volume every day, and it is important to handle, analyze and organize such big data. Social media is something that has become a game changer in the 21st century, something that we live by. It has changed how we and enterprises look at advertising, marketing, globalization or politics. The data we generate has been estimated to be increasing faster than ever before and in the year 2020, about 1.7 megabytes of additional data has been generated by each person. 2.5 quintillion bytes of data are produced by humans every day. More data has been generated in the past two years than ever before in the history of mankind. Internet users have also increased from millions to billions with the introduction of affordable internet services.

# **MOTIVATION**

The Database which we have chosen for the proposed study is Twitter. It is a very popular microblogging service. Compared to conventional internet articles and web blobs we feel twitter is a better approximation of public sentiments. The reason being, the amount of relevant data is much larger for twitter and the response on twitter is more prompt and more general (as the number of people who tweet daily is substantially more than those who write web articles or blogs). Sentiment analysis of the public opinion is very important in macro-scale socioeconomic phenomena like predicting the stock market rates of a particular firm. This can be done by analyzing the overall public sentiment towards that firm with respect to time and using various economic tools for finding the correlation between the firm’s stock market value and public sentiment. Firms can also review how well a product is doing in the market, which areas of the market is it having a favorable or negative response in. Since twitter allows us to download a stream of geo-tagged tweets for particular locations. If firms can get this information and analyze it, they can get the reasons behind geographically differentiated response, and can thus market their product in a more optimized manner by looking for solutions like creating suitable market segments. Predicting the results of popular political elections and polls is also an emerging application to sentiment analysis. A study was conducted by Tumasjan et al. in Germany for predicting the outcome of federal elections, which concluded that twitter is a good reflection of offline sentiment.

# **SCOPE**

This project hopes to be useful to companies, political people and the common person. It will be helpful to review the public opinion on their programs. Companies can get reviews about their new product or latest hardware or softwares. Also, the movie makers and goers can see the reviews on the currently running movie. By analyzing the tweets, the analyzer can get results on how positive, negative or neutral people are about it.

# **OVERVIEW OF THE P****ROJECT**

This project is a web application which is used to analyze the tweets about a certain topic using a hashtag or a certain person using their username. We will be building a classifier to perform sentiment analysis of these tweets to determine whether they are positive, negative or neutral. This application can be used by any organization office to review their works or by any other company to review their products or brands.

The main feature of our application is that it helps to determine the public opinion about the products, government work, politics or any other by analyzing the tweets.

# **DOMAIN** **INTRODUCTION**

This project of analyzing sentiments of tweets comes under the domain of “Data Mining” and “Pattern Classification”. These terms are very closely related and can be formally defined as the process of discovering “useful” patterns in large sets of data. This is done either automatically (unsupervised) or semi-automatically (supervised). Techniques of Natural Language Processing (NLP) are heavily relied on in extracting significant patterns and features from the large data set of tweets. Machine Learning techniques are also heavily relied on for accurately classifying individual unlabeled data (tweets) according to the pattern model that best describes them.

The features used for modeling patterns and classification can be divided into two main groups: formal language based and informal blogging based. Language based features deal with formal linguistics and include prior sentiment polarity of individual words and phrases, and parts of speech tagging of sentences(POS). Prior sentiment polarity means that words and phrases have a natural tendency to express specific sentiments. For example- the word “great” has a strong positive connotation while the word “bad” has a strong negative connotation. Whenever a word with positive connotation is used in a sentence, the sentence has a higher chance to also be expressing a positive sentiment. Parts of Speech tagging is a syntactic approach, it automatically identifies which part of speech each individual word of a sentence belongs to : noun, pronoun, adjective, verb, adverb, etc. Patterns can be extracted from analyzing the frequency distribution of these parts of speech either individually or collectively with some other part of speech, in a class of labelled tweets. Twitter based features relate with how people express themselves online on social platforms and are more informal. They compress their sentiments to 140 characters, which is the limited space offered by twitter for each tweet. They include twitter hashtags, word lengthening, word capitalization, presence of url in tweets, question marks, exclamation marks, internet emoticons and slang/shorthand. The data sets used in this field are mainly from product reviews. These reviews are important to business holders to take business decisions in accordance with the analysis of users’ opinions about their products. Sentimental Analysis finds its applications in various fields such as predicting stock markets, on political debates or news articles. In political debate, we could gain insight into people’s opinions on certain election candidates or political parties. The election results might also be predicted from political posts. Social networking sites or micro-blogging sites are very good sources of information as people freely share and discuss their opinions about various topics and can be used as data sources in the Sentiment Analysis process.

Sentiment Classification techniques can be divided into machine learning approach, lexicon-based approach and hybrid approach.

The Machine Learning Approach (ML) applies the famous ML algorithms and uses linguistic features.

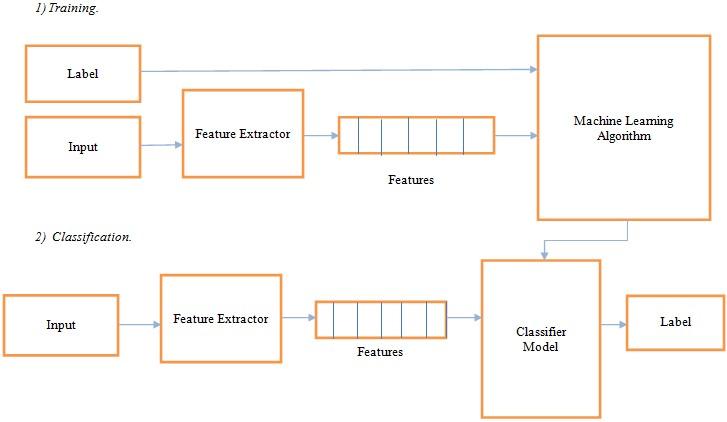


Figure1: Machine Learning Approach

The Lexicon-based Approach relies on a sentiment lexicon, a collection of known and precompiled sentiment terms. It is divided into a dictionary-based approach and corpus-based approach which use statistical or semantic methods to find sentiment polarity.

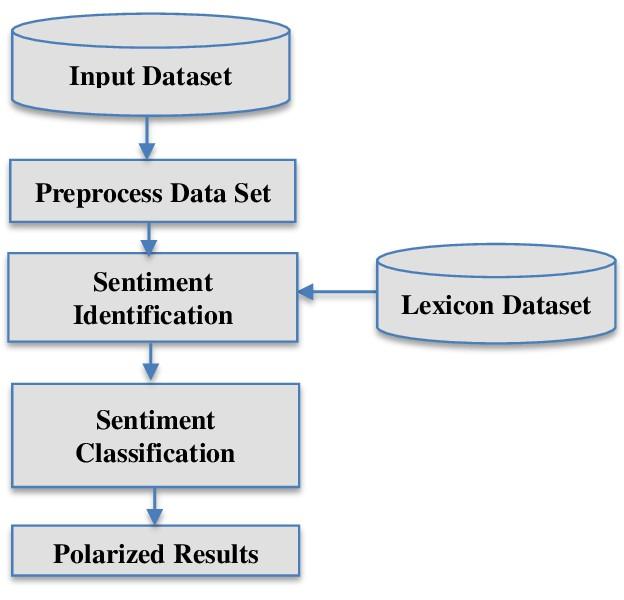


Figure 2: Lexicon-based Approach

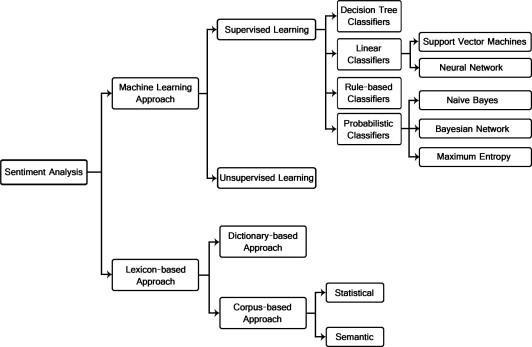
The Hybrid Approach combines both approaches and is very common with sentiment lexicons playing a key role in the majority of methods. The various approaches and the most popular algorithms:

Figure 3: Types of Sentiment Analysis

The text classification methods using ML approach can be roughly divided into supervised and unsupervised learning methods. The supervised methods make use of a large number of labeled training documents. The unsupervised methods are used when it is difficult to find these labeled training documents.

The lexicon-based approach depends on finding the opinion lexicon which is used to analyze the text. There are two methods in this approach. The dictionary-based approach which depends on finding opinion seed words, and then searches the dictionary of their synonyms and antonyms. The corpus-based approach begins with a seed list of opinion words, and then finds other opinion words in a large corpus to help in finding opinion words with context specific orientations. This could be done by using statistical or semantic methods.

ML algorithms are usually used to solve the Sentiment Classification problems for its simplicity and the ability to use training data which gives it the privilege of domain adaptability. Therefore, we are going to be using a Machine Learning algorithm for our project.

# **APPLICATION**

The application of sentiment analysis ranges from business and marketing, health, politics to public action. Sentiment analysis provides a vast application in different areas to assist in decision making.

1. **Applications that use Reviews from Websites:** The Internet now has a large collection of feedback and reviews on almost everything. This includes feedback on product issues, comments about services, product reviews, etc. A sentiment analysis system that can extract sentiments about certain services or products is therefore required which can help us automate the analysis of ratings or feedback for a given item, program, product, etc. This would meet the needs of both the vendors and the users. Allowing us to keep track of what is being said about a service or product on social media and can help us detect angry and dissatisfied customers or negative mentions before they turn into a major crisis or negatively impacts an organization's brand image.

1. **Applications in Business Intelligence:** People now tend to look for reviews of products online before they buy them. So, for many businesses, the online opinion becomes a deciding factor in the success or failure of their product or service. Sentiment Analysis plays an important role in businesses in this way. It can assist businesses and organizations to take the appropriate action to improve their product or services and business strategy . Sentiment analysis creates advantages for business owners to identify their popularity among customers and how customers think about their product or service and evaluate their business flow of stock price through social media and assessing the effectiveness and capability of business brand communication.

1. **Applications as a Sub-component Technology:** A sentiment predictor system can be helpful in recommender systems. These systems won’t recommend items that receive a lot of negative feedback or low ratings. Communication over the internet can be dangerous. We can come across abusive language and other negative elements. These can be detected by identifying a highly negative sentiment and correspondingly taking action against it.

1. **Applications across Domains:** Recent researches in sociology and other fields like medical, sports have also been benefited by Sentiment Analysis that show trends in human emotions especially on social media.

1. **Applications in Politics:** Predicting the results of popular elections and polls is also an emerging application to sentiment analysis. Data analyzed from twitter is more reliable as a platform where 94% of correlation has been found to polling data and have the potential to become a platform that is able to rival sophisticated polling techniques

Applications in Healthcare: We can see the application of sentiment analysis in healthcare and where the study uses Sentiment analysis as a service framework is proposed and utilizes spatio-temporal properties to identify locations of disease outbreaks. Sentiment analysis can be used to find the level of depression of a person by observing and analyzing emotions from the text in their social media. In addition, sentiment analysis can identify sentiment needs of people during a disaster and prepare an appropriate response to rescue.

# **LITERATURE REVIEW**

Sentiment analysis has been handled as a Natural Language Processing task at many levels of granularity. From being handled as a document level classification task (Turney, 2002; Pang and Lee, 2004), handled at the sentence level (Hu and Liu, 2004; Kim and Hovy, 2004) and also at the phrase level (Wilson et al., 2005; Agarwal et al., 2009). Classifying the sentiment of tweets is most similar to sentence level sentiment analysis, Given the 140-character limit on tweets. However, the informal and specialized language of using shorthands/internet slangs used in tweets, as well as the very nature of the microblogging domain make Twitter sentiment analysis a very different task. Microblog data like Twitter, on which users post real time reactions to and opinions about “everything”, poses newer and different challenges. (Barbosa and Feng 2010) exploit existing Twitter sentiment sites for collecting training data. (Davidov, Tsur, and Rappoport 2010) also use hashtags for creating training data, but they limit their experiments to sentiment/non-sentiment classification, rather than 3-way polarity classification, as we do.

Some of the early results on sentiment analysis of Twitter data are by Go et al.

(2009), Pak and Paroubek (2010) and (Bermingham and Smeaton, 2010). Go et al. (2009) used distant learning to acquire sentiment data. They build models using Naive Bayes, MaxEnt and Support Vector Machines (SVM), and reported that SVM outperforms other classifiers. In terms of feature space, they try a Unigram, Bigram model in conjunction with parts-of-speech (POS) features. Reporting that the unigram model outperforms all other models and bigrams and POS features do not help. (Das and Chen, 2001) reported that both Naive Bayes and SVM classifiers achieve a higher accuracy without applying the negation rules. (Joachims, 1998) reported that both these two classifiers have achieved more than 0.9 accuracy using unigram features in traditional topic-based classification.

SVMs were used by Yung-Ming Li and Tsung-Ying Li(2013) as a sentiment polarity classifier. Unlike the binary classification problem, they argued that opinion subjectivity and expresser credibility should also be considered for the accuracy of the model. They proposed a framework that gives a compact numeric summarization of opinions on micro-blogs platforms.

Hanhoon Kang, Seong Joon Yoo and Dongil Han(2012) proposed an improved NB classifier to correct the tendency for the positive classification accuracy to appear up to approximately 10% higher than the negative classification accuracy. This decreases the average accuracy when the accuracies of the two classes are expressed as an average value. They reported that using this algorithm with restaurant reviews narrowed the gap between the positive accuracy and the negative accuracy compared to NB and SVM. The accuracy is improved in recall and precision compared to both NB and SVM.

After analyzing these articles, it is clear that the enhancements of Sentiment

Classification algorithms are still an open field for research. Naïve Bayes and Support Vector Machines are the most used ML algorithms for solving sentiment classification problems. They are often considered a reference model to compare many proposed algorithms to. SVM is known to be the model giving the best results but, in this project, we focus only on the probabilistic model that is Naive Bayes that has been widely used in this field.

The Naive Bayes model which is well known for its simplicity and efficiency for text classification. We report accuracy results on data that does not suffer from any known biases. We perform cross validation experiments and check for the variance in performance of the classifier across folds.

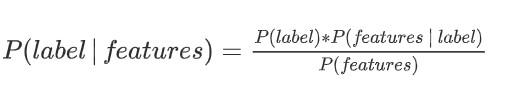
## **Naïve Bayes Classifier (NB)**

It is a Probabilistic classifier which uses mixture models for classification. The mixture model assumes that each class is a component of the mixture. Each mixture component is a generative model that provides the probability of sampling a particular term for that component. These kinds of classifiers are also called generative classifiers.

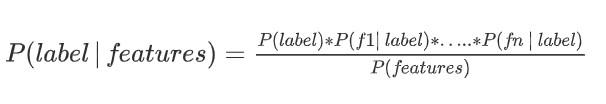
The Naïve Bayes Classifier is the simplest and most commonly used classifier. It computes the posterior probability of a class, based on the distribution of the words in the document.

The model works with the BoWs feature extraction which ignores the position of the word in the document. A bag-of-words model, or BoW for short, is a way of extracting features from text for use in modeling, such as with machine learning algorithms. A bag-of-words is a representation of text that describes the occurrence of words within a document. The approach is very simple and flexible, and can be used in a myriad of ways for extracting features from documents.

It uses Bayes Theorem to predict the probability that a given feature set belongs to a particular label.



P(label) is the prior probability of a label or the likelihood that a random feature sets the label. P(features|label) is the prior probability that a given feature set is being classified as a label. P(features) is the prior probability that a given feature set occurs. Given the Naïve assumption which states that all features are independent, the equation could be rewritten as follows:



Rather than computing P(features) explicitly, our algorithm just calculates the numerator for each label, and normalizes them so they sum to one:

P(label) \* P(f1|label) \* ... \* P(fn|label)

P(label|features)

= ━━━━━━━━━━━━━━━━━

SUM[l]( P(l) \* P(f1|l) \* ... \* P(fn|l) )

Despite the assumption we made above is not always held in the real word, the Naive Bayes model performs surprisingly well. (Domingos and Pazzani, 1997) show that even with some dependent features, Naive Bayes is optimal for certain problems. Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets. It can be used for Binary as well as Multi-class Classifications. It is the most popular choice for text classification problems

**Advantages of Naïve Bayes Classifier:**

* It is easy and simple to implement
* It performs well even without much training data
* It handles both continuous and discrete data
* It is highly scalable with the number of predictors and data points
* It is fast and can be used to make real-time predictions
* It is not sensitive to irrelevant features

## **TextBlob**

Textblob is a python library that provides text mining, text analysis and text processing modules for python developers. It reuses NLTK corpora. Textblob is a sentence level analysis tool. First, it takes a dataset as the input then it splits the review into sentences. A common way of determining polarity for an entire dataset is to count the number of positive and negative sentences/reviews and decide whether the response is positive and negative based on total number of positive and negative reviews. Polarity and subjectivity of a given review can be obtained using the sentiment module. It returns a named tuple with two parameters called polarity and subjectivity. The default sentiment analyzer algorithm uses the implementation PatternAnalyzer which is based on a library called pattern.The sentiment analysis lexicon bundled in Pattern focuses on adjectives. It contains adjectives that occur frequently in customer reviews, hand-tagged with values for polarity and subjectivity. The polarity score is ranging from -1 to 1 and subjectivity ranges are from 0 to 1 where 0 is most objective and 1 is most subjective.

# **SYSTEM REQUIREMENTS STUDY**

* USER CHARACTERISTICS

Sentiment analysis is a process that automates mining of attitudes, views, opinions and emotions from text, tweets, speech and database sources through Natural Language Processing (NLP). It involves classifying opinions in text into categories like "positive" or "negative" or "neutral". Also known as subjectivity analysis, appraisal extraction or opinion mining.

The words sentiment, view, opinion and belief are used interchangeably but they are somewhat different.

* Opinion: A conclusion open to dispute (because different experts have different opinions)
* View: subjective opinion
* Belief: deliberate acceptance and intellectual assent
* Sentiment: opinion representing one’s feelings

Sentiment Analysis includes many tasks such as sentiment extraction, subjectivity classification, sentiment classification, opinion spam detection, summarization of opinions and many more. It aims to analyze people’s sentiments, opinions, attitudes, emotions, etc. towards elements such as individuals, organizations, products, topics and services.

**The main components of this application are:**

* **UI:** The graphical interface of this web application has been developed using html, CSS, and streamlit which is an open-source Python library.

It is simple and consists of a text input field for a hashtag to be searched, two sliders on the sidebar to input other fields and two buttons: one to analyse the hashtag and the other to display the raw twitter data that has been obtained from data mining through the twitter API.

* **Backend:** The backend of this web application has been developed using Python3 and its libraries.

**In this project some libraries used:**

* **Tweepy,** a Python library for accessing the Twitter API for scraping data or data mining.
* **Natural Language Toolkit (NLTK),** a NLP library in Python, to clean the tweets.
* **TextBlob,** a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks. It is used for sentiment analysis.
* **Scikit-learn (Sklearn)** is the most useful and robust library for machine learning in Python.It provides a selection of efficient tools for machine learning and statistical modeling including various classification, regression and clustering algorithms.
* **Jupyter Notebook,** an interactive computational environment, in which you can combine code execution, rich text, mathematics, plots and rich media. It is used for faster implementation of the experiments.

The web application is hosted using heroku, a cloud platform.

**HARDWARE REQUIREMENTS:**

* x86 64-bit CPU (Intel / AMD architecture)
* 4 GB RAM
* 5 GB free disk space

**SOFTWARE REQUIREMENTS:**

* Windows 7 or higher
* A web browser preferably Chrome, Firefox
* Python-3
* Twitter developer account

# **SYSTEM ANALYSIS**

### **Feasibility study**

* **Technical Feasibility:**

Our system is technically feasible since all the required tools are easily available. PyCharm which is a hybrid-platform IDE for python and Streamlit which is an open-source library for python makes the system more user and developer friendly. Jupyter Notebook is an easy to use, interactive data science environment.

* **Operational Feasibility:**

In this project, we have made a simplified web application for analyzing the tweets. It is simple to operate and can be used in any webpages. It is free and not costly to operate.

* **Economic Feasibility:**

This is a web-based application. Hosted on a free cloud service platform; heroku. Creation of application is not costly.

* **Schedule Feasibility:**

This application development is feasible in terms of time.

### **Requirement Definition**

After the extensive analysis of the problems in the system, we are familiarized with the requirement that the current system needs. The requirement that the system needs is categorized into the functional and non-functional requirements.

These requirements are listed below:

1. **Functional Requirements:**

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. The functional requirements that this system must require are as follows:

* System should be able to set twitter authentication keys.
* System should be able to authorize tweepy and access twitter API.
* System should be able to analyse data and classify each tweet polarity.

1. **Non-Functional Requirements:**

Non-functional Requirements define system attributes such as reliability, maintainability, security, performance, scalability and usability. They serve as constraints on the design of the system across different backlogs. They ensure the usability and effectiveness of the entire system.

Based on this the non-functional requirements are as follows:

* User friendly
* System should provide better accuracy
* To perform with efficient throughput and response time
* Easy to maintain and is reliable

# **SYSTEM DESIGN**

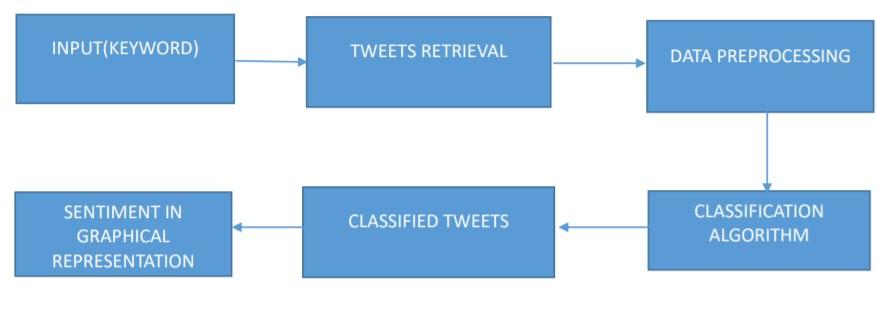


Figure 4: System Design

# **METHODOLOGY**

For the development a dataset containing labelled tweets is extracted from Kaggle. Pre-processing takes place on the data obtained. Then by using Naive Bayes algorithm a classifier will be trained using training data. Several methodologies will be used to extract features from the source text. After features are added to the feature vector, each tweet in the training data is associated with the class label and passed to the classifier. Test tweets will be given to the model and classification will be performed with the help of these trained classifiers. We get the tweets classified into the positive and negative. We then extend the classification to include the neutral class. The model can now be saved and loaded when a prediction is to be made. It is ready to be implemented in our web application.

1. **Loading sentiment data**

The Dataset for this project is extracted from Kaggle . This data set contains more than a million tweets extracted using the twitter api. The tweets have been annotated (0 = negative, 4 = positive) and they can be used to detect sentiment .

It contains the following 6 fields:

1. target: the polarity of the tweet
2. ids: The id of the tweet
3. date: the date of the tweet
4. flag: The query. If there is no query, then this value is NO\_QUERY.
5. user: the user that tweeted
6. text: the text of the tweet

The files contain positively labelled and negatively labelled tweets. First, the dataset is loaded. We use a fraction of this to train our model for better time efficiency. 200000 tweets were loaded. We replaced the polarity value of positive as: 4→ 1. Next, we check the number of positive and negative tagged tweets.

No. of positive tagged sentences: 99956

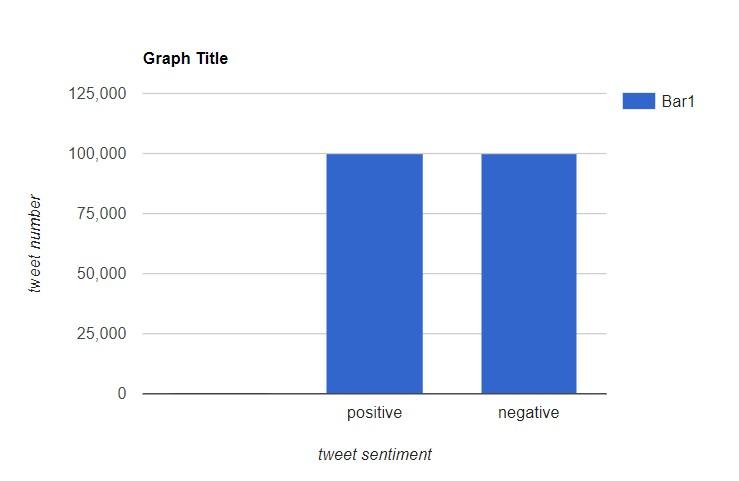
No. of negative tagged sentences: 100044

Figure 6: Dataset bias

We see that the dataset is balanced. Therefore, the chance baseline is 50%.

We drop the unnecessary columns leaving only polarity and text in the data.

1. **Pre-processing Data**

After loading data, pre-processing takes place.

The Preprocessing steps taken are:

* Lower Casing: Each text is converted to lowercase.
* Removing URLs: Links starting with "http" or "https" or "www" are replaced by "".
* Removing Usernames: Replace @Usernames with word "".
* Removing Short Words: Words with length less than 2 are removed.
* Removing Stopwords: Stopwords are the English words which do not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. (eg: "the", "she", "have")
* Lemmatizing: Lemmatization is the process of converting a word to its base form. ( eg: runs, running, ran are all forms of the word run, therefore run is the lemma of all these words ).

1. **Training Naïve Bayes Classifier**
2. **Convert text to word frequency vectors:**

TF-IDF: This is an acronym that stands for Term Frequency – Inverse Document Frequency which are the components of the resulting scores assigned to each word.

* 1. Term Frequency: This summarizes how often a given word appears within a document.
  2. Inverse Document Frequency: This downscales words that appear a lot across documents.

1. **Split train and test:**

The Preprocessed Data is divided into 2 sets of data:

* 1. Training Data: The dataset on which the model would be trained on. Contains 80% data.
  2. Test Data: The dataset on which the model would be tested against. Contains 20% data.

1. **Model Building:**

We build the Bernoulli Naive Bayes model from the sklearn library by feeding it the training data.

1. **Implementation of evaluation metric**

Finally, for testing the accuracy of the model, F1 score is evaluated. In statistical analysis of binary classification, the F1-score or F-measure is a measure of a test's accuracy. For the evaluation of data confusion matrix is used. A confusion matrix is a technique for summarizing the performance of a classification algorithm. Calculating a confusion matrix can give us a better idea of what our classification model is getting right and what types of errors it is making. The number of correct and incorrect predictions are summarized with count values and broken down by each class.

There are 4 important terms in confusion metric:

* 1. True Positives: The cases in which we predicted YES, and the actual output was also YES.
  2. True Negatives: The cases in which we predicted NO, and the actual output was NO.
  3. False Positives: The cases in which we predicted YES, and the actual output was NO.
  4. False Negatives: The cases in which we predicted NO, and the actual output was YES.

On evaluating our model we found:

Accuracy of model on training data : 86.868749999

Accuracy of model on testing data : 75.697499999

The F1 score : 0.76

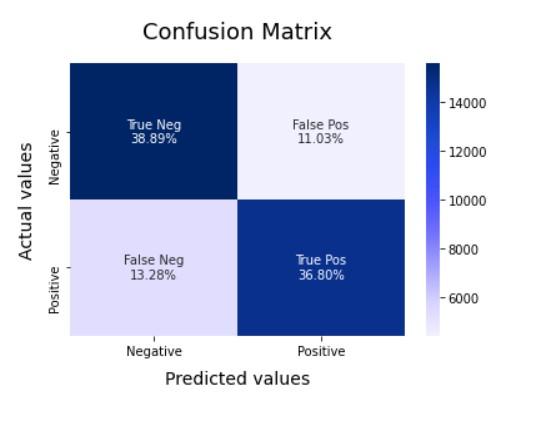


Figure 7: ***Confusion Matrix of Model***

Now we save this model and load it at the time of making a prediction.

1. **Handling the Neutral Class**

In the previous sections, neutral sentiment was disregarded. The training and test data only had text with positive and negative sentiments. In this section, we explore what happens when neutral sentiment is introduced.

**Naive Bayes with Three Classes**

Our model returns its results as a named tuple of the form: Sentiment (classification, p\_pos, p\_neg)

eg:Sentiment(classification='pos', p\_pos=0.5057908299783777, p\_neg=0.49420917002162196)

We extended the Naive Bayes Classifier to handle 3 classes: positive, neutral, and negative. By taking the difference of p\_pos and p\_neg as a polarity score (polarity = p\_pos - p\_neg), we have classified the three classes as such:

Positive: if the polarity score is more than or equal to 0.2.

Negative: if the polarity score is less than or equal to - 0.2.

Neutral: if the polarity score is between 0.2 and -0.2.

Although rudimentary it works quite well for this type of classification.

Finally, after classifying the tweets, we can visualize their count using graphs. We also visualize the polarity and subjectivity obtained through the textblob library in a scatter plot.

# **IMPLEMENTATION**

In this paper, Tweepy python library has been utilized for data extraction from Twitter API (Application programming interface). Moreover, Tweepy allows appropriate data retrieval by searching via keywords, hashtags, timelines, trends, or geo-location. In spite of having numerous restrictions from Twitter API, we have applied successive attempts to access as many posts as possible. We test out our modified Naive Bayes algorithm and compare it to the inbuilt PatternAnalyzer in TextBlob. We have used usernames to test our algorithm in contrast to hashtags as the test data extracted varies when using hashtags in real time. In order to avoid redundancy, a specific timeframe has been considered to pull out the tweets during a week. The gathered data has been stored in CSV format, and fed to our Sentiment Analysis model and Textblob. The data gets analysed, classified and finally visualized.

**Implementation In Web Application:**

For example: Giving the input of “uber” in the hashtag option, we can analyse the sentiments (positive, negative or neutral), polarity and subjectivity of the tweets that are tagged with #uber as shown below.

**SNAPSHOTS:**

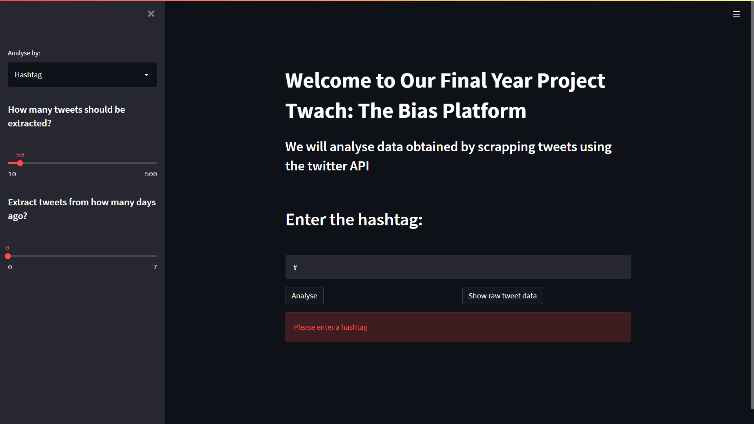


Figure 9: ***Snapshot of Web Application #1***

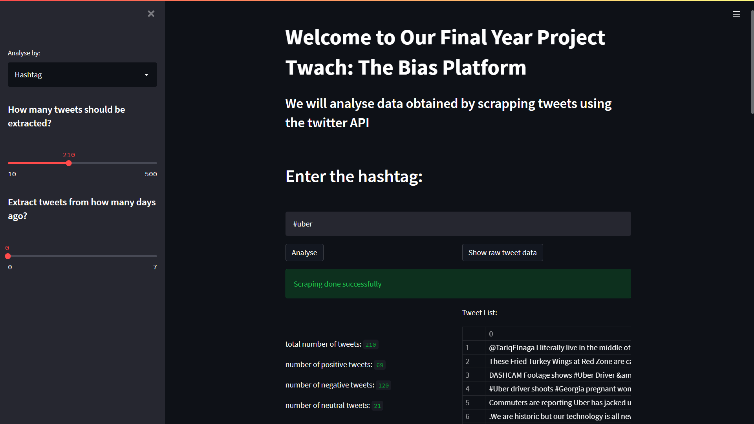


Figure 10: ***Snapshot of Web Application #2***

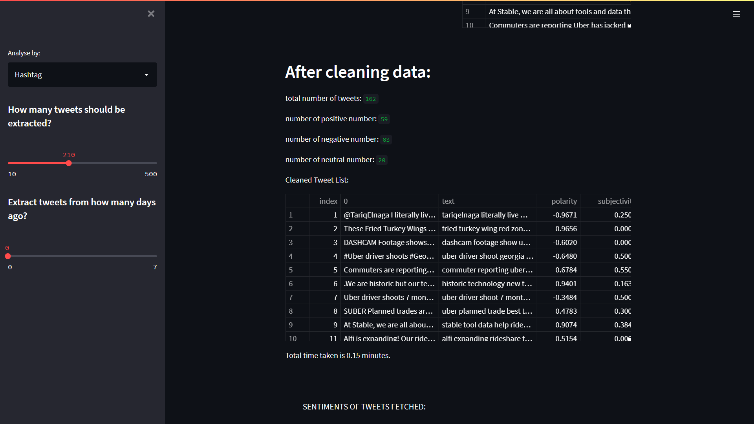
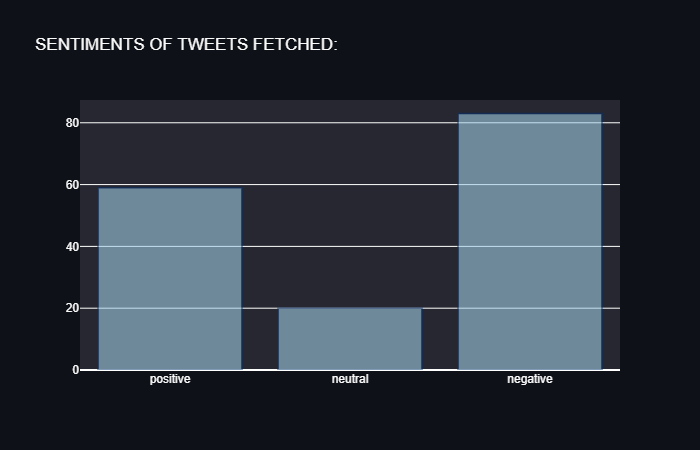


Figure 11: ***Snapshot of Web Application #3***



Figure 12 : ***Snapshot of Web Application #4***

 Figure 13 : **Graphical Representation of Tweet Sentiments by Bar Graph**

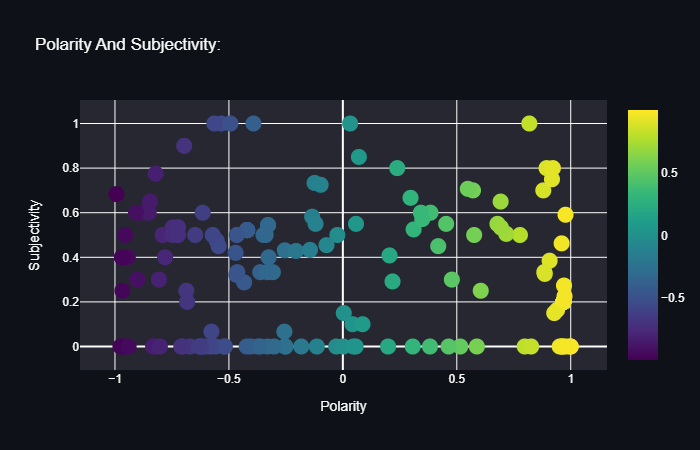


Figure 14: ***Graphical Representation of Tweet Sentiments by Scatter Plot***

# **LIMITATIONS**

Sentiment Analysis is a very challenging task. Following are some of the limitations of Sentiment Analysis of Twitter:

* + - 1. **Identifying subjective parts of text:** Subjective parts represent sentiment bearing content. The same word can be treated as subjective in one case, or an objective in some other. This makes it difficult to identify the subjective portions of text.

Example:

1. The language of Mr Danny was very crude.
2. Crude oil is obtained by extraction from the sea beds. The word “crude” is used as an opinion in the first sentence, while it is completely objective in the second sentence.
3. **Domain dependence:** The same sentence or phrase can have different meanings in different domains. Example:

The word “unpredictable” is positive in the domain of movies, dramas, etc, but if the same word is used in the context of a vehicle steering, then it has a negative opinion.

1. **Sarcasm Detection:** Sarcastic sentences express negative opinions about a target using positive words in a unique way.

Example: “Wow that’s amazing!; Well done!”

The sentence contains only positive words but actually it expresses a negative sentiment.

1. **Thwarted expressions:** In some sentences only some part of the text determines the overall polarity of the document.

Example: “This Movie should have been amazing. It sounds like a great plot, the popular actors, and the supporting cast is talented as well”.

In this case, a simple bag-of-words approach will term it as positive sentiment, but the ultimate sentiment is negative.

1. **Explicit Negation of sentiment:** Sentiment can be negated in many ways as opposed to using simple no, not, never, etc. It is very challenging to identify such negations. Example: “It avoids all clichés and predictability found in Hollywood movies.” Here the words clichés and predictable bear a negative sentiment, but the usage of the word “avoids” negates their respective sentiments.
2. **Order dependence:** Discourse Structure analysis is essential for Sentiment Analysis/Opinion Mining.

Example: Dan is older than John, conveys the exact opposite opinion from, John is older than Dan.

1. **Building a classifier for subjective vs. objective tweets:** Our current research work focuses mostly on classifying positive, negative and neutral tweets correctly. There is a need to look at classifying tweets with sentiment vs. no sentiment closely.
2. **Handling comparisons:** Bag of words model doesn't handle comparisons very well.

Example: "IIT’s are better than most of the private colleges", the tweet would be considered positive for both IIT’s and private colleges using the bag of words model because it doesn't take into account the relation towards "better".

1. **Applying sentiment analysis to Facebook messages:** There has not been much work on sentiment analysis on Facebook data due to mainly the various restrictions by Facebook graph API and security policies in accessing data.
2. **Internationalization**: Current research work focuses on mainly English content, but Twitter has many varied users from across the globe with different languages.
3. **Entity Recognition:** It is necessary to separate out the text about a specific entity and then analyze the sentiment towards it.

Example: “I hate Microsoft, but I like Linux”. A simple bag-of-words approach will label it as neutral, however, it carries a specific sentiment for both the entities present in the statement.

# **CONCLUSION AND FUTURE SCOPE**

Therefore, by entering a keyword, we get the visual representation of the tweets analyzed (positive, negative, neutral) as well as the polarity and subjectivity through the web app.

The conducted systematic literature review provides information on studies on sentiment analysis in social media. There are various methods introduced by researchers. Choosing the most efficient method of sentiment analysis depends on the data itself. Both methods demonstrated a similar accuracy. The things that we need to consider are the structure of the text, time and amount of data. If the data structure is messy, there is limited time available to analyse and a small amount of data, lexicon-based methods are more suitable. Whereas, machine learning based methods are more suitable for bigger data as it requires more data and time to train. In order to improve the accuracy and quality of the result, combining both lexicon and machine learning methods would be recommended.

We identified the most common type of social media site to extract information from for sentiment analysis, which is Twitter. Most of the reviewed papers use twitter as their social media context. This is due to the accessibility, availability and richness of Twitter content. There are millions of tweets every day on almost any topic. This indicates that social media is becoming a precious source of information.

We demonstrate the application of sentiment analysis in social media through making a web app. Sentiment analysis has a broad application and can be utilized in different areas such as political forecasting an election result, improving strategy and quality of products in business, monitoring disease outbreak, perception towards a particular topic or person and improving response to a disaster. This shows that sentiment analysis plays a huge role in understanding people's perception and helps in decision making.

Sentiment Analysis, especially in the domain of micro-blogging is still in the developing stage and far from being complete. We propose some ideas which we feel are worth exploring in the future and may result in improved performance.

1. We may find more reliable sources of information on other social media platforms like Facebook, Instagram, LinkedIn, Youtube, etc. It is necessary to explore other social media platforms to perform and analyze sentiment analysis results.
2. In our contribution, we chose to use the Naïve Bayes model. But there are other models that may provide interesting results such as lexicon-based algorithms. In this project we tried to show the basic way of classifying tweets into positive, negative or neutral categories using our modified Naive Bayes. We could further improve our classifier by trying to extract more features from the tweets, tuning the parameters of the Naïve Bayes classifier, trying different kinds of features, or trying another classifier all together.
3. We are focusing on general sentiment analysis. There is potential for research in the field of sentiment analysis with partially known context. For example, users generally use our website to search with specific types of keywords which can be divided into distinct classes: celebrities, politics/politicians, sports/sportsmen, brands/products, movies/media and music. So we can attempt to perform separate sentiment analysis on tweets that only belong to one of these classes (i.e. the training data would not be general but specific to one of these categories) and compare the results with those that we would get if we apply general sentiment analysis on it instead.
4. One more feature that we feel is worth exploring is whether the information about relative position of words in a tweet has any effect on the performance of the classifier. Although Pang et al. explored a similar feature and reported negative results, they worked on an extremely simple model and their results were based on reviews which are very different from tweets.

# **ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to Dr. Dharampal Singh of the department of Computer Science and Engineering, whose role as a project guide was invaluable for the project. We are extremely thankful for the keen interest he took in advising us, for the books and reference materials provided and for the moral support extended to us.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skill that will always remain as an asset throughout our lives and to all non-teaching staff for the gracious hospitality they offered us.

Place: JIS College of Engineering, Kalyani

Date: 23rd February, 2022

Anjan Kumar Dey

Ashish Abhishek Minz

Ayan Chatterjee

Debajyoti Ghosh

Debraj Karmakar

department of Computer Science and Engineering,

JIS College of Engineering, Kalyani,

West Bengal, India.

REFERENCES:

* “tweepy documentation” -<https://docs.tweepy.org/en/latest/api.html>
* “textblob documentation” -<https://textblob.readthedocs.io/en/dev/>
* “streamlit documentation”-<https://docs.streamlit.io/en/stable/>
* https://tealfeed.com/twitter-sentimental-analysis-using-naive-bayes-9eb73
* https://stackabuse.com/sentiment-analysis-in-python-with-textblob/
* https://investigate.ai/investigating-sentiment-analysis/comparing-sentimentanalysis-tools/
* https://www.jeremyjordan.me/evaluating-a-machine-learning-model/
* https://towardsdatascience.com/nlp-in-python-data-cleaning-6313a404a470
* https://python.plainenglish.io/scraping-tweets-with-tweepy-python59413046e788
* Alexander Pak and Patrick Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In Proceedings of international conference on Language Resources and Evaluation (LREC), 2010.
* B. Pang and L. Lee. "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts" in Proceedings of ACL, 2004.
* Peter D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL), 2002.
* Hu, M., &amp; Liu, B. (2004)."Mining and Summarizing Customer Reviews." Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '04 (2004).
* Kang, Hanhoon, Seong Joon Yoo, and Dongil Han. “Senti-Lexicon and Improved Naïve Bayes Algorithms for Sentiment Analysis of Restaurant Reviews.” Expert Systems with Applications (2012)
* Kim, Soo-Min, and Eduard Hovy. "Determining the Sentiment of Opinions." Proceedings of the 20th International Conference on Computational Linguistics - COLING '04 (2004).
* Agarwal, Apoorv, Owen Rambow, and Nandini Bhardwaj. "Predicting Interests of People on Online Social Networks." 2009 International Conference on ● Computational Science and Engineering (2009).
* Alec Go, Richa Bhayani and Lei Huang. Twitter Sentiment Classification using Distant Supervision. Project Technical Report, Stanford University, 2009.
* R Kusumawati et al 2019 J. Phys.: Conf. Ser. 1320 012016 Comparison Performance of Naive Bayes Classifier and Support Vector Machine Algorithm for Twitter's Classification of Tokopedia Services
* Walaa Medhat, Ahmed Hassan, Hoda Korashy,Sentiment analysis algorithms and applications: A survey, Ain Shams Engineering Journal,Volume 5, Issue 4,2014.
* pdf.sciencedirectassets.com/280203
* Rahman, S. A. El, F. A. AlOtaibi, and W. A. AlShehri. (2019, 3-4 April 2019). “Sentiment Analysis of Twitter Data”, in the 2019 International Conference on Computer and Information Sciences (ICCIS).
* Joyce, Brandon, and Jing Deng. (2017) “Sentiment Analysis of Tweets for the 2016 US Presidential Election”, in IEEE MIT Undergraduate Research Technology Conference (URTC), Cambridge, MA, USA: IEEE.
* Poecze, Flora, Claus Ebster, and Christine Strauss. (2018) “Social Media Metrics and Sentiment Analysis to Evaluate the Effectiveness of Social Media Posts.” Procedia Computer Science 130: 660-666.
* Suman, N., P. K. Gupta, and P. Sharma. (2017, 11-12 Dec. 2017). “Analysis of Stock Price Flow Based on Social Media Sentiments”, in the 2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS).
* Ali, Kashif, Hai Dong, Athman Bouguettaya, Abdelkarim Erradi, and Rachid Hadjidj. (2017) “Sentiment Analysis as a Service: A Social Media Based Sentiment Analysis Framework”, in IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA: IEEE.
* Hassan, Anees Ul, Jamil Hussain, Musarrat Hussain, Muhammad Sadiq, and Sungyoung Lee. (2017) “Sentiment Analysis of Social Networking Sites (SNS) Data Using Machine Learning Approach for the Measurement of Depression”, in International Conference on Information and Communication Technology Convergence (ICTC), Jeju, South Korea: IEEE.
* Ragini, J. Rexiline, P. M. Rubesh Anand, and Vidhyacharan Bhaskar. (2018) “Big Data Analytics for Disaster Response and Recovery Through Sentiment Analysis.” International Journal of Information Management 42: 13-24.
* Twitter REST API <[https://dev.twitter.com/docs/api>](https://dev.twitter.com/docs/api)