

**WestminsterResearch**

<http://www.westminster.ac.uk/westminsterresearch>

**Twitter Sentiment Analysis and Emotion Detection Using NLTK  
and TextBlob**

**Nehal, Divyank Jeet, Vandana Sharma, Sushruta Mishra,  
Celestine Iwendi and Jude Osamor**

This is a copy of the author's accepted version of a paper subsequently published in the proceedings of the 2023 4th International Conference on Computation, Automation and Knowledge Management (ICCAKM), Dubai, United Arab Emirates, 12 - 13 Dec 2023 IEEE.

The final published version will be available online at:

<https://doi.org/10.1109/iccakm58659.2023.10449540>

© 2023 IEEE . Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners.

# Twitter Sentiment Analysis and Emotion Detection using NLTK and TextBlob

Nehal  
Kalinga Institute of Industrial  
Technology, India  
nehalpandey463@gmail.com

Sushruta Mishra  
Kalinga Institute of Industrial  
Technology, India  
sushruta.mishrafcs@kiit.ac.in

Divyank Jeet  
Kalinga Institute of Industrial  
Technology, India  
[thisisdvynk@gmail.com](mailto:thisisdvynk@gmail.com)

Celestine Iwendi, SMIEEE  
School of Creative Technologies  
University of Bolton, United Kingdom  
[c.iwendi@bolton.ac.uk](mailto:c.iwendi@bolton.ac.uk)

Vandana Sharma, SMIEEE  
Department of Computational Sciences,  
CHRIST (Deemed to be University),  
Delhi NCR, India  
vandana.juyal@gmail.com

Jude Osamor, FHEA  
Department of Cybersecurity and  
Networks  
Glasgow Caledonian University  
Glasgow, G4 0BA Scotland, UK  
jude.osamor@gcu.ac.uk

**Abstract**—On a worldwide level, every second around 6000 tweets are sent, which counts to around 200 billion tweets in a year. People share their ideas and views publicly on Twitter, thus it serves as a good platform for analyzing public trends and behaviour towards any person, product or news. Customers frequently utilize social media platforms to share their thoughts and experiences regarding goods and services. Businesses can find areas for improvement and better understand the attitudes of their customers towards their goods and services by using sentiment analysis. In order to perform sentiment analysis on twitter, text classification using Natural Language Processing(NLP) has been proved to be very helpful. Using NLP word tokenizer, we can divide the sentences into different sets of words, thereafter we remove the stop words. Manually tokenizing long tweets and categorizing them into separate groups is challenging. The primary goal of this model is to analyze the tweets related to a given keyword entered by the user, classify the tweets as positive, negative or neutral using (VADER sentiment analysis or alternately). TextBlob library, which will help consumers as well as manufacturers to understand people's overall opinion regarding the product. This study makes an attempt to suggest a text sentiment analysis on twitter data using the NLTK and TextBlob libraries.

**Keywords**—Python, Natural Language Processing, Sentiment Analysis, NLTK, Tokenization, TextBlob, Polarity, Subjectivity

## I. INTRODUCTION

We are currently living in an era dominated by digital technology, where advancements are constantly pushing the boundaries. Data creation is occurring at an unprecedented rate, with every passing second. From our social media interactions to the intricacies of the stock market, banking transactions, IT operations, government processes, and various industries, data plays a pivotal role in driving these fields forward. The computational technique of sentiment analysis, commonly referred to as opinion mining, is used to identify and categorize the sentiment represented in text data. It entails examining the text's subjective data to determine whether the sentiment is favourable, negative, or neutral. To extract and quantify the overall sentiment or emotional tone of a text, such as a tweet, review, or news story, is the goal of sentiment analysis. There are multiple social media platforms that can be a good source of data for understanding a customer's thought process. Due to its real-time nature and the abundance of user-generated content expressing ideas and attitudes, Twitter is a useful medium for sentiment analysis.

The enormous stream of public tweets that Twitter offers can be gathered via its API called Tweepy or other data scraping techniques. This data can be the basis for sentiment analysis, capturing distinct viewpoints and sentiments on a range of subjects. To establish a connection between python and twitter, we can use the Tweepy API which requires some authentication posts which we can enter some keywords and Tweepy API gives us various live tweets which it fetches from twitter related to those keywords. We can also get a sentiment report which contains the polarity as well as the subjectivity. Due to the casual language used in tweets, sentiment analysis can be difficult because of the frequent use of acronyms, misspellings, slang, GIF and grammatical errors. Sarcasm and irony are commonly used by Twitter users, which might cause sentiment to be misclassified. Understanding the context and subtle linguistic indicators is necessary for spotting such cases. Tweets are frequently brief and incomplete in their contextual information. Disambiguating words and phrases based on the context in which they are employed is necessary to comprehend the sentiment. For sentiment analysis, labeled data collection can be expensive and time-consuming. Building precise sentiment analysis models is significantly complicated by the dearth of labeled data [1].

NLTK library can be useful for handling such complicated texts. The process of tokenizing raw text into smaller units, such as words or sentences, is provided by NLP and NLTK. This method is essential to many NLP applications, including sentiment analysis, text classification, and information retrieval. The word\_tokenize function can be used to tokenize the tweets or sentences into bag of words. NLTK can also be used to find part of the speech for each of the words. NLTK offers tools and functions for part-of-speech (POS) tagging [2], which assigns grammatical labels to words in a sentence. POS tagging is helpful for studying the structure of a sentence and retrieving relevant information for tasks like named entity recognition, text summarizing, and machine translation. Once we have our data ready for analysis, multiple libraries are present and prebuilt models available to perform text categorization. Rule-based methods assess sentiment using established dictionaries of positive and negative words. Machine learning algorithms that can categorize sentiment learnt from labeled training data include Support Vector Machines (SVM), Naive Bayes, and Random Forest. Complex linguistic patterns and contextual information can be captured by deep learning models like Recurrent Neural

Networks (RNNs) or Transformers for sentiment classification. In some cases, sentiment analysis involves aggregating sentiment at the document level, such as determining the overall sentiment of a collection of tweets or reviews. Aggregation methods can include averaging individual sentiment scores or using more advanced techniques like aspect-based sentiment analysis to evaluate sentiment towards specific aspects or entities within the text.

We can utilize the TextBlob library for sentiment classification of tweets as positive, neutral or negative. TextBlob works on the basis of polarity metrics and subjectivity metrics. Polarity measures the text on a scale of -1 to +1 and tells us the degree of positivity, neutrality or negativity of the input text and, on the other hand, subjectivity helps us to see how objective or subjective the given input is on a scale of 0 to 1. A rule-based sentiment analysis programme called VADER (Valence Aware Dictionary and sentiment Reasoner) [9] was created specifically to examine tweets and other social media content. It is accessible as a component of the Python NLTK library. VADER provides four sentiment metrics: 'neg' (negative), 'neu' (neutral), 'pos' (positive), and 'compound' (an aggregated score combining all three). It uses a threshold to categorize the sentiment as positive, negative, or neutral based on the compound score. There is another method, using Transformers or BERT [6] (Bidirectional Encoder Representations from Transformers) model. We can accurately analyse the sentiment of tweets by using transformer models like BERT because of their capacity to record contextual information and complex interactions between words. By delivering cutting-edge performance on a variety of tasks, transformers have considerably advanced the area of NLP, particularly sentiment analysis, and excel at comprehending the complicated nature of language.

## II. LITERATURE REVIEW

Recent studies show that various methods and preexisting algorithms can be used to perform sentiment analysis on textual data. The author of the paper in [2] aims at exploring the recent techniques of sentiment analysis. The methodology adopted in this paper is to study sentiment analysis through four different techniques, namely : Supervised learning, Unsupervised learning, Semi-supervised learning and Lexicon-based [2] approach. The outcome of this study established that there is a lack of SA research in languages other than English; and existing techniques are still unable to deal with complex sentences that require more than sentiment words and simple parsing; it also categorized some recent articles presented in SA field according to their techniques. The study in paper [3] aims at classifying sentiments using three main steps, namely, subjectivity classification, semantic association, and polarity classification [3]. The Methodology adopted for this study uses natural language processing (NLP) to extract topics from tweets and categorize tweet polarity by looking at sentiment lexicons related to topics. As an outcome of the study, SA accuracy score was analyzed for various algorithms like Alchemy API, Naive Bayes classifier, Decision tree and SVM (support vector machine) and it was found that SVM performs better if tweets were preprocessed before training and testing. In [4], the main objectives of the paper include measuring customer satisfaction on Amazon and comparison and evaluation of "TextBlob" and "Bert" in both Arabic and English. The methodology used in this study includes data collection through web scraping and then performing SA using

Bidirectional Encoder Representations from Transformers Model [BERT Model] and TextBlob model [5]. The outcome established by the authors was that the two models work similarly, but the complexity of the BERT model is a little higher than that of the TextBlob model. The purpose of the paper in [6] is to evaluate popular people's handles (like @realdonaldtrump) or hashtags (like #IPL2018) [7] in order to comprehend the attitudes of people in each circumstance when the person has tweeted about or responded to certain situations. The methodology adopted for this study makes use of TextBlob library to examine the opinions of several well-known people on a particular occasion or event. The outcome of this study emphasizes the importance of data pre-processing on the accuracy of the model and calculating the polarity of tweets on various topics. The proposed study involved the classification of over 4,000,000 reviews into positive and negative sentiments [8] through the application of Sentiment Analysis using various classification models like Naive Bayes, Support Vector Machine and Decision trees. The classification technique can be used to find the polarity of sentiment ( $\text{Polarity} = \text{PositiveScore} - \text{NegativeScore}$ ). Cross validation technique is used for model evaluation and the three models have been compared against each other. As a conclusion, it was found that the Support Vector Machine model gives the highest accuracy. The study in [9] compares the effectiveness of VADER and logistic regression as methods for sentiment analysis. A dataset of tweets concerning airlines was used to test the effectiveness of different methods, and the accuracy of the predictions was compared. With an accuracy of 79%, the results demonstrated that Logistic Regression fared better on this dataset than VADER. These findings imply that when applied to airline tweets, logistic regression may be a more successful method for sentiment analysis. Recent studies have been held out to discover other methods that can be used to build SA models. In the study proposed in [10], a hybrid sentiment analysis model is proposed that uses a Long Short-Term Memory network, a lexicon based on rules for sentiment analysis, and the Term Frequency Inverse Document Frequency weighting technique. A binary classification model is created by combining these three (input) models. The methodology used for this study involves the analysis through above mentioned algorithms on the IMDB dataset. The results obtained as the outcome of this study shows that the accuracy obtained through logistic regression was 5.91% better than that obtained by the three basic models. Another paper gives a survey on sentiment analysis that covers the many ideas in this field, difficulties and their fixes, tools that are currently in use, available APIs, and open challenges in this field. The methodology used in this paper talks about the levels of SA. Numerous enhancement methods have been discussed in this paper like Dimensionality reduction, data integration, crowdsourcing [11], user profiling, spam detection and text summarization. The outcome of this study highlights the various applications of SA and suggests the tools and APIs that can be used for performing SA. The paper in [12] gives an explanation of the various techniques for sentiment analysis and sentiment classification. This description serves as the foundation for the paper's classification of approaches according to their qualities, benefits, and drawbacks. The methodology adopted for this research involves studying three approaches using machine learning, Lexicon based and Hybrid models, each of which has their own benefits and limitations. The paper concludes on the note that different

approaches and tools can be applied in different techniques based on the requirement.

### III. PROBLEM STATEMENT

A company's or product's performance is directly correlated with its customer base. It is a success if the customer likes your product. If they don't like it, we will undoubtedly need to improvise by changing it. Sentiment analysis can be used to assess consumer data to determine whether or not a product is successful. With the rapid expansion of the internet and social media, it is crucial to keep an eye on and rein in the poisonous content that is being shared on sites like Twitter, Facebook, Amazon, and Instagram. Cyber-bullying is a serious social issue. Sentiment analysis assists in removing harmful language to create a safer and more peaceful online environment. At the same time, we may use the information found on these platforms to comprehend the wants and needs of the customer. Sentiment analysis can be used to determine the general consensus on any subject. For numerous reasons, sentiment analysis in social media is crucial for: understanding client feedback, keeping an eye on a brand's reputation, seeing new trends, and doing market research.

### IV. PROPOSED MODEL

With a data set received through the Tweepy API, the proposed model of text tokenization, discretion, and ultimately sentiment analysis has been established for textual data only. Instead of depending entirely on pre-built datasets, using the Tweepy API for sentiment research on Twitter data has various advantages. We use the data from the API as it will always produce fresh data. This will help in ensuring that the studies that we are carrying out will always stay relevant and up-to-date. With the help of Tweepy, we may examine the most recent tweets on a particular subject, occasion, or term to get the most recent insights and sentiment patterns. This model derives its uniqueness from the fact that it utilizes Tweepy to get control over the specific data of our interest. It makes it possible to set filters that are not just restricted to keywords but also allow hashtags, geolocations and user profiles to collect tweets that are most relevant to our study. As we know that the accuracy of the data used in an analysis directly influences the accuracy of the analysis itself, Tweepy can help us reduce the load of exploratory data analysis to a great extent. Accurate data ensures that the sample used for analysis is representative of the target population.

In order to maintain data consistency and remove noisy data, it is extremely important to use techniques like tokenization [13], tagging and removing stop words to make data suitable for performing analysis algorithms. After pre-processing the input data, each word is given a score of +1, 0 or -1 for being positive, neutral or negative respectively. Our model then calculates the overall polarity score and displays the result as a pie-chart. The proposed model is shown in figure 1.

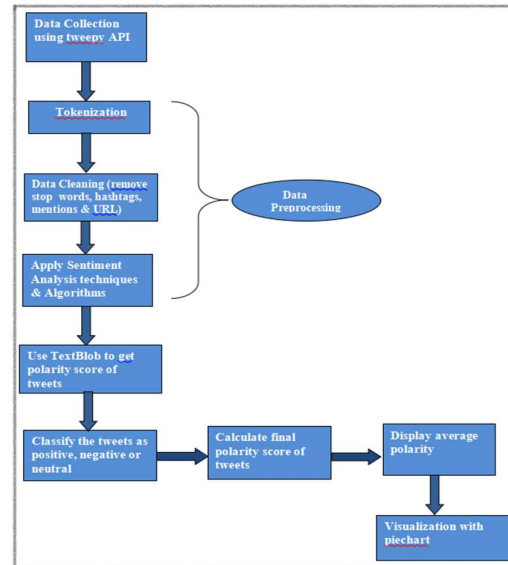


Figure 1. Proposed system architecture

In this experiment we can analyze tweets related to a given keyword and entered number of tweets. So we have considered 1000 tweets related to [14] COVID-19 for our study. Tweets related to the user-entered keyword are used as the dataset and stored in the form of a CSV(Comma Separated Value) file. The following is a glimpse of our dataset containing 1000 tweets depicting mixed emotions of different kinds of people:

```

1 Hey @Yankees @YankeesPR and @MLB - wouldn't it...
2 @diane3443 @wdunlap @realDonaldTrump Trump nev...
3 @brookbanktv The one gift #COVID19 has give me...
4 25 July : Media Bulletin on Novel #CoronaVirus...
...
995 @CassieCeleste @lobster_nanny I do understand...
996 My job isn't to enlighten you #fuck #alreadyvi...
997 #COVID19 reflection in a street art, #Kharkiv,...
998 Nice Read.\n@MoriangoJr @AdanZamuh @kungu_@mo...
999 Treat every single decision you take regarding...
Name: text, Length: 1000, dtype: object
  
```

Figure 2. Sample dataset for our implementation

If we observe some tweets of our dataset as shown in figure 2, we see that our dataset is not ready to be used for SA. It includes a lot of details like twitter mentions, URLs, Hashtags and extra spaces which make it difficult for text analysis. In order to have accurate results, we need to make sure that those elements that could hamper text analysis have been removed.

Unstructured [15] text data might present challenges when trying to solve NLP issues. Any unstructured text data must be preprocessed in order to create a useful NLP model. Pre-processing textual input is therefore a crucial step in creating any NLP [16-17] model. For the data-cleaning step, we have two methods available- nltk library and neat-text library.

#### A. Neat-text for data cleaning

For cleaning the unorganized data we are going to make use of neat-text library available in NLP package. It has several prebuilt functions which make the task of data

cleaning easier. The figure 3 is the list of some functionalities it offers:

```
'clean_text',
'defaultdict',
'digit2words',
'extract_btc_address',
'extract_currencies',
'extract_currency_symbols',
'extract_dates',
'extract_emails',
'extract_emojis',
'extract_hashtags',
'extract_html_tags',
'extract_mastercard_addr',
'extract_md5sha',
'extract_numbers',
'extract_pattern',
'extract_phone_numbers',
'extract_postoffice_box',
'extract_shortwords',
'extract_special_characters',
'extract_stopwords',
'extract_street_address',
'extract_terms_in_bracket',
'extract_urls',
'extract_userhandles',
'extract_visacard_addr',
'fix_contractions',
'generate_sentence',
'hamming_distance',
'inverse_df',
'lexical_richness',
```

Figure 3. Functions in neartext library for data cleaning

The process of cleaning twitter data becomes a breeze because of these sophisticated features. There are functions like 'extract\_hashtags', 'extract\_urls', 'extract\_stopwords' and 'extract\_userhandles' which proved to be really helpful as the code becomes a one-liner with the availability of these features.

	text	clean_tweet
0	If I smelled the scent of hand sanitizers toda...	If I smelled the scent of hand sanitizers toda...
1	Hey @Yankees @YankeesPR and @MLB - wouldn't it...	Hey and wouldnt it have made more sense to ha...
2	@diane3443 @wdunlap @realDonaldTrump Trump nev...	Trump never once claimed was a hoax We all cl...
3	@brookbanktv The one gift #COVID19 has give me...	The one gift has give me is an appreciation f...
4	25 July : Media Bulletin on Novel #CoronaVirus...	25 July : Media Bulletin on Novel
...	...	...
995	@CassieCeleste @lobster_nanny I do understand...	I do understand In 2017 we suffered a standar...
996	My job isn't to enlighten you #fuck #alreadyv...	My job isn't to enlighten you
997	#COVID19 reflection in a street art, #Kharkiv...	reflection in a street art Ukraine
998	Nice Read. In @MoriangoJr @AdanZamuh @kunga_@mo...	Nice Read Rea...
999	Treat every single decision you take regarding...	Treat every single decision you take regarding...

Figure 4. Contrast between Original text and Cleaned text

Post-processing and after data-cleaning our data becomes suitable for applying nltk tools. As it can be seen in figure 4 that all the twitter mentions like @Donald, URLs and Hashtags have been removed along with some special characters which did not any value to our text. The distinction between the original text and the cleaned text can be clearly seen in the given figure.

### B. NLTK for Tokenization and POS Tagging

We import the necessary libraries, like nltk to divide our tweets into tokens and remove the stop words and perform part of speech tagging [18]. We use a nltk function called word\_tokenize to tokenize the CSV file containing tweets. Next, we use nltk to apply averaged perception tagger to tag all the tokens. Nltk can find part of the speech for each of the words. After tokenization, we clean the data. Cleaning data involves removing all the special characters or any other word which do not add any value to the analytics part.

The figure 5 is the depiction of how a tweet looks after being tokenized:

```
tokens = nltk.word_tokenize(example)
tokens[:10]

['This', 'oatmeal', 'is', 'not', 'good', '.', 'Its', 'mushy', ',', 'soft']
```

Figure 5. Division of a tweet into tokens

A POS [19] tag represents the grammatical category or syntactic role of a word in the sentence. The English language includes 8 parts of speech, namely noun, pronoun, adjective, verb, adverb, preposition, conjunction and interjection. POS tagging helps us classify the words into parts of speech, which proves to be really helpful as later on we can remove pronouns, prepositions and utilize adjectives and verbs. Words like "was", "the", "is", "he", "she" will be removed as we do not need those words which do not add any value to the analytics parts. The output of the tagging process is a sequence of word-tag pairs, where each word is associated with its assigned POS tag. For example, "The/DT cat/NN is/VB sleeping/VBG" represents the tagging output for the sentence "The cat is sleeping," with DT (determiner), NN (noun), VB (verb), and VBG (verb, gerund or present participle) as the POS tags.

The outcome of tagging sample is represented in figure 6.

```
tagged= nltk.pos_tag(tokens)
tagged[:10]

[('This', 'DT'),
 ('oatmeal', 'NN'),
 ('is', 'VBZ'),
 ('not', 'RB'),
 ('good', 'JJ'),
 (',', ','),
 ('Its', 'PRP$'),
 ('mushy', 'NN'),
 ('.', '.'),
 ('soft', 'JJ')]
```

Figure 6. POS Tagging of tweet using nltk

### C. Calculation of Polarity

We use the TextBlob library to convert the processed , ready-to-use data to metadata and find its polarity scores .It works as an excellent sentiment analyzer and gives us polarity scores ranging from -1 to +1.

Let’s consider a tweet regarding a concert which says “The concert was great!”. This tweet is divided into 5 tokens which are “The”, “concert”, “was”, “great”, “!”. After performing the above filtration steps, we remove words and expressions like “The”, “was” and “!” which are not going to help in our study and we are left with only two words, “concert” and “great”.Our task is to classify them as either positive, negative or neutral words. For a positive word, we give our sentiment score of +1, for negative, we give our sentiment score as -1 and for neutral we give zero.

For our calculation, we have only two words- "concert" (score=0) as it is a neutral word and "great" (score=+1). So the final score is +1. As the polarity is greater than zero, the statement is positive. So, this is how our calculation works. Using the above technique, we calculate the polarity of all the tweets. We later create a result file that stores the polarity of each of the tweets fetched through Tweepy API. The polarity values analysis is shown in figure 7.

```
[ ] res
#now we have the polarity score of each review

[1: {'neg': 0.0, 'neu': 0.695, 'pos': 0.305, 'compound': 0.9441},
2: {'neg': 0.138, 'neu': 0.862, 'pos': 0.0, 'compound': -0.5664},
3: {'neg': 0.091, 'neu': 0.754, 'pos': 0.155, 'compound': 0.8265},
4: {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0},
5: {'neg': 0.0, 'neu': 0.552, 'pos': 0.448, 'compound': 0.9468},
6: {'neg': 0.029, 'neu': 0.809, 'pos': 0.163, 'compound': 0.883},
7: {'neg': 0.034, 'neu': 0.693, 'pos': 0.273, 'compound': 0.9346},
8: {'neg': 0.0, 'neu': 0.52, 'pos': 0.48, 'compound': 0.9487},
9: {'neg': 0.0, 'neu': 0.851, 'pos': 0.149, 'compound': 0.6369},
10: {'neg': 0.0, 'neu': 0.705, 'pos': 0.295, 'compound': 0.8313},
11: {'neg': 0.017, 'neu': 0.846, 'pos': 0.137, 'compound': 0.9746},
12: {'neg': 0.113, 'neu': 0.887, 'pos': 0.0, 'compound': -0.7579},
13: {'neg': 0.031, 'neu': 0.923, 'pos': 0.046, 'compound': 0.296},
14: {'neg': 0.0, 'neu': 0.355, 'pos': 0.645, 'compound': 0.9466},
15: {'neg': 0.104, 'neu': 0.632, 'pos': 0.264, 'compound': 0.6486},
```

Figure 7. Polarity score of tweets after applying TextBlob

## V. RESULT AND ANALYSIS

### A. EXPERIMENTAL SETUP

A core i5 8th generation processor, 1.10 GB of RAM, and a 30 GB hard drive are needed for this experiment. In order to meet the software requirements, I utilized the most recent version of Windows software, a Google Colab notebook, the Python programming language, Pycharm editor. The libraries used for this study are NLTK , pandas, numpy, matplotlib, seaborn, neat-text, TextBlob and VADER.

With the help of TextBlob, we can determine whether the input, which comes in the form of a CSV file, is positive, negative, or neutral.

### B. RESULTS

For the given data set, we analyzed the sources of the tweets and the following trend could be seen:

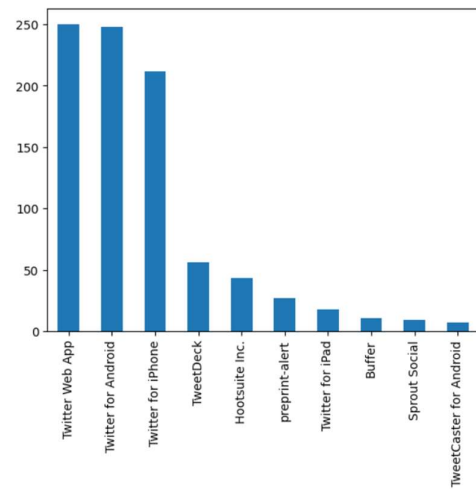


Figure 8. Bar graph showing the sources of tweets

In the given figure 8 we have obtained a bar plot of the 10 highest value counts of the sources of tweets. The plot shows that maximum tweets have been posted by Twitter Web App followed by Twitter for android and Twitter for iPhone. Therefore, our dataset includes a collection of tweets published by various people via various sources [20].

After applying SA on our dataset we classified the tweets as positive, negative or neutral. This is a crucial step as in this step we get the statistics and overall perception of the tweets related to the topic of user interest. The figure 9 shows the number of positive tweets, negative tweets and neutral tweets.

```
Neutral 419
Positive 410
Negative 171
Name: sentiment, dtype: int64
```

Figure 9. Count of different categories of tweets

In this example, where we are analyzing 1000 tweets, the plot shows that 410 tweets were positive, 419 tweets were negative and about 171 tweets were neutral related to COVID-19. This result as shown in figure 10 denotes that the majority of people had expressed either positive or neutral emotions during the pandemic.

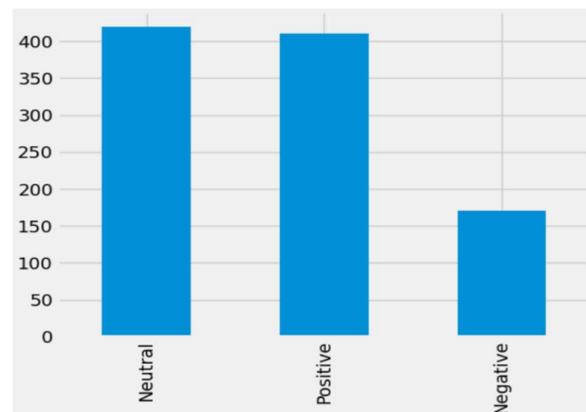


Figure 10. Count of sentiments in the tweets

The final result can be seen through a bar plot, pie chart and a word-cloud. Pie charts provide a clear and intuitive representation of data distribution and enable easy comparison between different categories or data points. We Therefore use a pie chart for better visual representation [21]. It shows us the percentage of tweets that are positive, negative and neutral and provides us more clarity about the result. So, whenever one wants to find the overall review of any product, one can refer to the pie chart for quick analysis. This study shows that out of the 1000 tweets considered for analysis, 41.0 % of the tweets were positive, 17.1 % of the tweets were negative and 41.9 % of the tweets were neutral. Figure 11 shows the overall outcome.

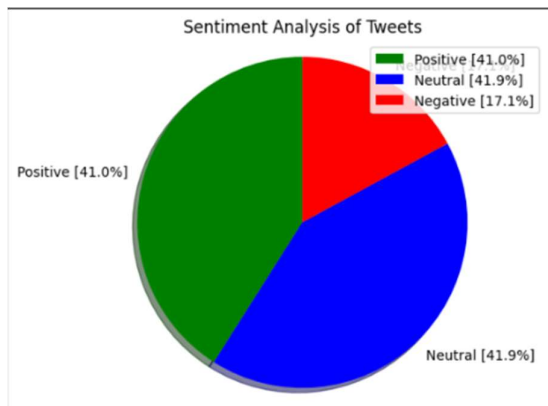


Figure 11. Visualization of result

Word-clouds highlight the frequency or importance of different words within a given text or a collection of texts. The size or prominence of a word in the cloud corresponds to its frequency, allowing viewers to quickly identify the most commonly used terms. As the result shown in figure 11, the majority of the tweets were neutral, the following word cloud shows the neutral words in our dataset.

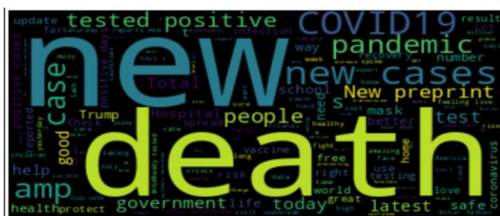


Figure 12. Word cloud showing frequency of neutral words

Figure 12 represents that words like “death”, “new”, “pandemic”, “COVID19” have been used most frequently in the list containing neutral words. For supporting the above derivation and for clearer observation, we can consider the following word count of neutral words:

3	Coronavirus	17
4	Covid19	16
5	people	16
6	today	13
7	testing	13
8	pandemic	13
9	amp	12
10	Hospital	11
11	Trump	10
12	Government	10
13	•	10
14	started	9
15	Plasma	9
16	Man	9
17	Singh	9
18	fight	9

Figure 13. Word count of neutral words

The above figure 13 shows that the most frequently used words in the tweets related to COVID-19 were “Corona virus”, “Covid-19”, “people” ,”testing”, “pandemic” and “hospital”. These word counts help us in summarizing the overall sentiment of the tweets.

## VI. CONCLUSION

As the volume of data expands, there is a growing emphasis on data-driven decision making. Organizations are leveraging advanced analytics and machine learning techniques to extract valuable insights from large datasets. Data-driven decision making can help optimize processes, identify market trends, and uncover new business opportunities, allowing data scientists to dig deeper into the data available online and apply necessary Machine Learning Algorithms to get insightful information. The vast amount of data available in this era makes it difficult to perform exploratory analysis, which paves a way for APIs like Tweepy that provide us with useful and relevant piece of information related to the domain of our interest. In order to break into the realm of data and apply sentiment analysis, NLP and TextBlob are wonderful tools to address this issue. They are undemanding yet efficient techniques that make comprehending complex sentences easier. The objective of this study was to find a direct and accessible way to inquire about people’s opinions or sentiments related to any article, campaign, product, persona, or industry. This paper proposes the steps to achieve the above mentioned objective. The data collected in the CSV file was gathered by a unique method. This data was later cleaned, transformed and then given to the TextBlob tool for measuring the polarities.

The TextBlob tool examined the tweet's scores in order to determine the tweet's real sentiment. Sentiment refers to a tweet's overall tone, either favourable or not favourable. The

model is reliable on account of its high accuracy which is above 80 percent. The future scope of this model is to be able to analyse not only textual data but also the data present in images, GIF, stickers, emojis or video form.

## REFERENCES

- [1] Z. Hu, J. Hu, W. Ding and X. Zheng, "Review Sentiment Analysis Based on Deep Learning," 2015 IEEE 12th International Conference on e-Business Engineering, Beijing, China, 2015, pp. 87-94, doi: 10.1109/ICEBE.2015.24.
- [2] Z. Madhoushi, A. R. Hamdan and S. Zainudin, "Sentiment analysis techniques in recent works," 2015 Science and Information Conference (SAI), London, UK, 2015, pp. 288-291, doi: 10.1109/SAI.2015.7237157.
- [3] W. Y. Chong, B. Selvaretnam and L. -K. Soon, "Natural Language Processing for Sentiment Analysis: An Exploratory Analysis on Tweets," 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology, Kota Kinabalu, Malaysia, 2014, pp. 212-217, doi: 10.1109/ICAIET.2014.43.
- [4] H. P. Patil and M. Atique, "Sentiment Analysis for Social Media: A Survey," 2015 2nd International Conference on Information Science and Security (ICISS), Seoul, Korea (South), 2015, pp. 1-4, doi: 10.1109/ICISSEC.2015.7371033.
- [5] A. J. Nair, V. G and A. Vinayak, "Comparative study of Twitter Sentiment On COVID - 19 Tweets," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 1773-1778, doi: 10.1109/ICCMC51019.2021.9418320.
- [6] Jesus 6.A. Mahgoub et al., "Sentiment Analysis: Amazon Electronics Reviews Using BERT and Textblob," 2022 20th International Conference on Language Engineering (ESOLEC), Cairo, Egypt, 2022, pp. 6-10, doi: 10.1109/ESOLEC54569.2022.10009176.
- [7] S. Zahoor and R. Rohilla, "Twitter Sentiment Analysis Using Lexical or Rule Based Approach: A Case Study," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2020, pp. 537-542, doi: 10.1109/ICRITO48877.2020.9197910.
- [8] Zeenia Singla, Sukhchandan Randhawa and Sushma Jain, "Sentiment analysis of customer product reviews using machine learning" Conference: 2017 International Conference on Intelligent Computing and Control (I2C2), doi 10.1109/I2C2.2017.8321910
- [9] P. Dhanalakshmi, G. A. Kumar, B. S. Satwik, K. Sreeranga, A. T. Sai and G. Jashwanth, "Sentiment Analysis Using VADER and Logistic Regression Techniques," 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCOIS), Coimbatore, India, 2023, pp. 139-144, doi: 10.1109/ICISCOIS56541.2023.10100565.
- [10] Swarna 10.K. Ahmed, N. E. Tazi and A. H. Hossny, "Sentiment Analysis over Social Networks: An Overview," 2015 IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong, China, 2015, pp. 2174-2179, doi: 10.1109/SMC.2015.380.
- [11] N. Das, S. Gupta, S. Das, S. Yadav, T. Subramanian and N. Sarkar, "A Comparative Study of Sentiment Analysis Tools," 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICES), Chennai, India, 2021, pp. 1-7, doi: 10.1109/ICES52305.2021.9633905.
- [12] Mohamed Chiny, Marouane Chihab, Omar Bencheref, Younes Chihab, "LSTM, VADER and TF-IDF based Hybrid Sentiment Analysis Model" ,International Journal of Advanced Computer Science and Applications. 12. 2021. 10.14569/IJACSA.2021.0120730.
- [13] S.Subhra, S. Mishra, A. Alkhayyat, V. Sharma and V. Kukreja, "Climatic Temperature Forecasting with Regression Approach," 2023 4th International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2023, pp. 1-5, doi: 10.1109/ICIEM59379.2023.10166883.
- [14] M.Sen, K. Sharma, S. Mishra, A. Alkhayyat and V. Sharma, "Designing a Smart and Intelligent Ecosystem for Autistic Children," 2023 4th International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2023, pp. 1-5, doi: 10.1109/ICIEM59379.2023.10166057.
- [15] Sharma, V., Mishra, N., Kukreja, V., Alkhayyat, A., & Elngar, A. A. (2023, March). Framework for Evaluating Ethics in AI. In 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA) (pp. 307-312). IEEE.
- [16] S.Mohanty, A. Behera, S. Mishra, A. Alkhayyat, D. Gupta and V. Sharma, "Resumate: A Prototype to Enhance Recruitment Process with NLP based Resume Parsing," 2023 4th International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2023, pp. 1-6, doi: 10.1109/ICIEM59379.2023.10166169.
- [17] A.Srivastava, S. Samanta, S. Mishra, A. Alkhayyat, D. Gupta and V. Sharma, "Medi-Assist: A Decision Tree based Chronic Diseases Detection Model," 2023 4th International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2023, pp. 1-7, doi: 10.1109/ICIEM59379.2023.10167400.
- [18] D. Mehta, P. P. Das, S. Ghosh, S. Mishra, A. Alkhayyat and V. Sharma, "A Normalized ANN Model for Earthquake Estimation," 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2023, pp. 151-155, doi: 10.1109/ICAAIC56838.2023.10140242.
- [19] A. Kumar and R. Jain, "Sentiment analysis and Feedback Evaluation," 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE), Amritsar, India, 2015, pp. 433-436, doi: 10.1109/MITE.2015.7375359.
- [20] K. Ahmed, N. E. Tazi and A. H. Hossny, "Sentiment Analysis over Social Networks: An Overview," 2015 IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong, China, 2015, pp. 2174-2179, doi: 10.1109/SMC.2015.380.
- [21] V. K. Vijayan, K. R. Bindu and L. Parameswaran, "A comprehensive study of text classification algorithms," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 2017, pp. 1109-1113, doi: 10.1109/ICACCI.2017.8125990.