**TWITTER SENTIMENT ANALYSIS ON COVID-19**



**ENGINEERING FINAL PROJECT PROPOSAL**

**Submitted To:**

**The Department of IT and Computer Engineering**

**In Partial fulfillment of requirement for the degree of**

**Bachelor in Computer Engineering**

**Submitted By:**

**Prastuti Pokhrel (160321)**

**Sneha Dahal (160339)**

**Kareena Khadka (160313)**

**Nashib Bista (160318)**

**Acknowledgement**

This project proposal is a leading way to the better understanding and implementing the acquired knowledge in the practical world. We express our gratitude to the department of it and computer engineering for providing a base for developing a project of our own that professionalize our knowledge and provides us a remarkable experience for our career.

This project is the outcome of knowledge and technical support shared by respective teachers. So, we would like to express our sincere gratitude to our Vice Principal Mr. Yuvraj Siwakoti and our HOD Mr. Bivek Ropakheti for giving us opportunity to conduct this study. We would also like to thank our supervisor Mr. Ranjan Raj Aryal for his guidance, co-operation and contribution of his valuable time in order to complete this study. Lastly, we would like to extend sincere thanks to our college and family for their continuous encouragement in our study.

**ABSTRACT**

The analysis of sentiment on social networks have become a powerful means of learning about people’s opinions and sentiments towards different issues as they spend hours daily on social medias and share their opinions. In this study, twitter being one of the most popular social media platforms, is used for the public’s opinions and information on a number of issues related to the COVID-19. Here, we have gathered tweets from 9th May 2021 to 15th May 2021 that are related to COVID-19 through python programming language using Tweepy library and then by using VADER, a lexicon based sentiment analysis tool, sentiment analysis operation has been done. From the dataset of 40541 collected tweets, it has been observed that the maximum number of the tweets portrayed positive and neutral sentiments. These labelled tweets then have been pre-processed, split into training and test sets and passed through Naïve Bayes Classifier Model, which gave us accuracy of over 72%.

**TABLE OF CONTENTS**

[1. INTRODUCTION 1](#_Toc78729713)

[**1.1 BACKGROUND** 1](#_Toc78729714)

[**1.2 PROBLEM STATEMENT** 2](#_Toc78729715)

[**1.3 OBJECTIVE** 2](#_Toc78729716)

[2. LITERATURE REVIEW 3](#_Toc78729717)

[**2.1 RELATED WORKS** 3](#_Toc78729718)

[**2.2 MOTIVATION** 6](#_Toc78729719)

[3. METHODOLOGY 6](#_Toc78729720)

[**3.1. TECHNOLOGY USED:** 6](#_Toc78729721)

[**3.1.1 Python Programming:** 6](#_Toc78729722)

[**3.1.2. Anaconda and Jupyter:** 7](#_Toc78729723)

[**3.1.3. Tweepy:** 7](#_Toc78729724)

[**3.1.4. Scikit-learn:** 7](#_Toc78729725)

[**3.1.5. Pandas:** 7](#_Toc78729726)

[**3.1.6. NLTK:** 8](#_Toc78729727)

[**3.2** **METHODS AND TECHNIQUES** 8](#_Toc78729728)

[**3.2.1 Data Extraction** 8](#_Toc78729729)

[**3.2.2 Labeling Sentiments** 8](#_Toc78729730)

[**3.2.3 Text Processing** 9](#_Toc78729731)

[**3.2.4 Feature Extraction** 10](#_Toc78729732)

[**3.2.5 Training Classifier** 12](#_Toc78729733)

[4. DESIGN 14](#_Toc78729734)

[5. IMPLEMENTATION 15](#_Toc78729735)

[**5.1.** **DATA EXTRACTION** 15](#_Toc78729736)

[**5.2.** **DATA CLEANING FOR VADER** 16](#_Toc78729737)

[**5.3.** **CREATING SENTIMENT LABELS USING VADER** 17](#_Toc78729738)

[**5.4.** **PREPROCESSING THE DATA** 18](#_Toc78729739)

[**5.4.1 Removing Emoticons** 19](#_Toc78729740)

[**5.4.2 Removing Emojis** 19](#_Toc78729741)

[**5.4.3 Acronyms** 20](#_Toc78729742)

[**5.4.4. Lowering cases and Removing Contractions** 21](#_Toc78729743)

[**5.4.5 Removing Punctuations, numbers and special characters** 21](#_Toc78729744)

[**5.4.6 Tokenization** 22](#_Toc78729745)

[**5.5. FEATURE EXTRACTION** 22](#_Toc78729746)

[**5.6.** **SPLITTING DATASET** 23](#_Toc78729747)

[**5.7.** **APPLICATION OF NAÏVE BAYES ALGORITHM** 23](#_Toc78729748)

[**5.8. ADVANCED PRE-PROCESSING:** 24](#_Toc78729749)

[**5.8.1. Lemmatization:** 24](#_Toc78729750)

[**5.8.2. Removing Stop words** 24](#_Toc78729751)

[**5.9. ASSOCIATION RULES:** 25](#_Toc78729752)

[6. RESULT AND ANALYSIS: 26](#_Toc78729753)

[**6.1. PERFORMANCE METRICS OF SENTIMENT CLASSIFICATION:** 26](#_Toc78729754)

[**6.2. CONFUSION MATRIX** 28](#_Toc78729755)

[**6.3. COMPARISON OF ACCURACY FROM DIFFERENT VECTORIZER** 29](#_Toc78729756)

[**6.4. ACCURACY IN N-GRAMS METHODS** 30](#_Toc78729757)

[7. CONCLUSION AND FUTURE WORK 30](#_Toc78729758)

[**7.1. LIMITATION AND FUTURE SCOPE:** 31](#_Toc78729759)

[8. WORKING SCHEDULE 32](#_Toc78729760)

[6. REFERENCES: 33](#_Toc78729761)

**TABLE OF FIGURES**

[Figure 1:General Architecture for Twitter Sentiment Analysis 14](#_Toc78317672)

[Figure 2: Displaying first 10 rows of our dataset 16](#_Toc78317673)

[Figure 3: Displaying first 10 rows of our dataset after removing @’s, mentions and urls 16](#_Toc78317674)

[Figure 4: Total numbers of rows and columns in our dataset after removing duplicate data 17](#_Toc78317675)

[Figure 5: 10 rows of our dataset with sentiment labels 18](#_Toc78317676)

[Figure 6: Visualization of negative,positive and neutral using seaborn plot 18](#_Toc78317677)

[Figure 7: list of tweets before and after applying emoticon\_text() 19](#_Toc78317678)

[Figure 8: Tweets before and after applying emoji\_cleaner() 19](#_Toc78317679)

[Figure 9: Top 20 of acronyms in the data set of tweets 20](#_Toc78317680)

[Figure 10: Top 20 of acronyms in the data set of tweets with their translation and count 20](#_Toc78317681)

[Figure 11: Tweets before and after lowering cases and removing contraction 21](#_Toc78317682)

[Figure 12: Tweets before and after removing punctuations, numbers, special characters. 21](#_Toc78317683)

[Figure 13: Tweets before and after performing tokenization 22](#_Toc78317684)

[Figure 14: Using CountVectorizer as our feature extractor 22](#_Toc78317685)

[Figure 15: Using TfidfVectorizer as our feature extractor 22](#_Toc78317686)

[Figure 16: Splitting our dataset into train and test set 23](#_Toc78317687)

[Figure 17: Training the model by passing X\_train and y\_train 24](#_Toc78317688)

[Figure 18: Tweets before and after applying lemmatize\_text() 24](#_Toc78317689)

[Figure 19: list of customized stop words under stop\_words 25](#_Toc78317690)

[Figure 20: Tweets before and after removing stopwords 25](#_Toc78317691)

[Figure 21: Using ngram\_range along with the vectorizer 26](#_Toc78317692)

**TABLE OF TABLES**

[Table 1: Twitter Data about Covid-19 during one week 15](#_Toc78317667)

[Table 2: Classification Report of performance metrics 28](#_Toc78317668)

[Table 3: Format of Confusion matrix 28](#_Toc78317669)

[Table 4: Comparing accuracy obtained by using different vectorizers and preprocessing steps 29](#_Toc78317670)

[Table 5: Comparing accuracy obtained by using different n-gram methods 30](#_Toc78317671)

# INTRODUCTION

## **1.1 BACKGROUND**

The emergence in the last decade of microblogging platforms such as Twitter, has enabled people to engage in social activities to express their opinions, interests and emotions on a various topics and events. The kinds of analysis as well as information that can be extracted from the social networking sites like twitter are varied and increasingly dragging attention of both world of marketing as well as social and political platform. Twitter has always been a most preferred platform where people freely discuss and share their opinions about different events and trends. So, it's not a surprise that twitter continues to play a crucial role during the world-wide outbreak of corona virus. Coronavirus known as COVID-19 which began to appear at the end of last year in Wuhan, China has been one of the most discussed and one of the most spreading diseases worldwide.

Considering the time started working on this study, “there are total of 67,555,001 corona virus cases and 1,544,416 deaths worldwide.”(WHO | World Health Organization, 2020/12/7). From this data, we can see that this is one of the most biological virus outbreaks in the last two decades of the century. World Health Organization has suggested that isolation and self-quarantine is one of the major ways to stop this pandemic to spread with such an alarming rate. After the benefits of lockdowns witnessed by china, other major countries facing the pandemic have also implemented the similar approach. This can easily be the biggest lockdown the world has ever seen. As majority of the people are under lockdown, the importance of Twitter has increased more than ever. People have been sharing their feelings, opinions and different theories related to Covid-19. Even the renowned organization like WHO, has been using Twitter as their platform to share latest updates and information related to this issue using hashtags like #coronavirus, #Covid19.

In this study, we have extracted huge amount of twitter data to analyze the sentiments of people that they have been sharing through their tweets during Covid-19 pandemic. Before understanding research conducted for Twitter sentiment analysis, we need to understand about the Natural Language Processing (NLP) and its techniques. The Natural Language Processing allows a machine to process a natural human language and translates it to a format that the machine understands. NLP dates back to the 1960s but became very popular with the advent of the World Wide Web and search engines. Sentiment analysis is one of the areas of natural language processing that deals with the computational study of opinions, sentiments and emotions expressed in text effectively.

We have used machine learning techniques for the sentiment classification on our dataset. Firstly, dataset with each tweet labeled as positive, negative and neutral are compiled. Then, a feature extractor was used to generate a feature vector for each labeled tweet. Once feature vectors are extracted for each tweet in the labeled dataset, they were fed to classification algorithm that attempts to find relations between each value (called feature) in the vector and the labeled sentiment. Generally, three types of machine learning techniques have been used for classification i.e., supervised learning techniques, unsupervised learning techniques and reinforcement techniques. In this study, we have focused on supervised learning techniques. Supervised Learning techniques are considered standard approach for sentiment classifications of texts. These techniques use different classifying algorithms like naïve bayes, support vector machine, KNN, Decision Tree for training the dataset. The aim in supervised learning is to predict the final outcome variable given the predictor variable.

## **1.2 PROBLEM STATEMENT**

Data on the internet are growing with very high rate and people are pretty much affected with the opinion of other people. These huge amounts of data generated from the social networking sites are generally unstructured. Unstructured data do not convey any meaning until and unless they are not analyzed. In case of COVID-19 worldwide pandemic, vast opinions and sentiments are out there on different social media platforms in unstructured form. Therefore, to utilize these unstructured data, there is a need of sentiment analysis of these data i.e., take the valuable feature from these data and classify them into classes. From buying a product from a certain platform to voting the right candidate in an election, people share their views and perspective which would eventually influence others. In this scenario, sentiment analysis has a significant role on analyzing people’s sentiment which could greatly help in businesses to know about the product evaluation, politics to know about public’s opinion and on research fields where sentiments of people matter.

## **1.3 OBJECTIVE**

* To create a dataset from the extracted tweets.
* To obtain sentiment values (polarity of text) for the dataset.

# LITERATURE REVIEW

## **2.1 RELATED WORKS**

There are many researchers who has done great contribution in this area by providing support for finding user behaviors and situations in the different cases while happening around the world. Here we have discussed some researches that helped us to know about the sentiment analysis in depth.

**Koyel Chakraborty et al. [1**] have done a study on sentiment analysis of COVID-19 tweets by Deep Learning Classifiers to show how popularity is affecting accuracy in social media. They have used state-of-art classifier based proposed model to analyze the sentiments of people post lockdown imposed by the government.

**Kamaran H. et al.** [2] have done their study on Twitter Sentiment Analysis on Worldwide COVID-19 Outbreaks the twitter data has been pulled out from Twitter social media, through python programming language, using Tweepy library, then by using TextBlob library in python the sentiment analysis operation has been done. After the measuring sentiment analysis, the graphical representation has been provided on the data. The data they had collected on twitter are based on two specified hashtag keywords, which are COVID-19 and coronavirus.

**A. Bastola et al.** [3] have discussed and explained the detail case about the first 2019 novel coronavirus case in Nepal. They have been deeply studied with the symptoms and all the consequences with the initial radiograph of the patient.

**R. Muthausami et al.** [4] have done research on the topic COVID-19 Outbreak: Tweet based Analysis and Visualization towards the Influence of Coronavirus in the World. They have utilized machine learning methodology as part of a request to give a careful examination of the tweets which have been extracted from Twitter. Tweets have been classified into positive, negative and neutral classes with the help of the different machine learning classifier as Naïve Bayes, SVM, Decision Tree, MaxEntropy, LogitBoost and Random Forests. In the experiment with three classes, the LogitBoost ensemble classifier gets the most noteworthy accuracy.

**M. Lwin et al. [**5] have done research on the topic Global Sentiments Surrounding the COVID-19 Pandemic on Twitter: Analysis of Twitter Trends. They have collected over 20 million social media twitter posts made during the early phases of the COVID-19 outbreak from January 28 to April 9, 2020 using “wuhan”, “corona”, “nCov” and “covid” as search keyword and analyzed the emotions of tweets using the algorithm CrystalFeel. Pearson r correlations were conducted between emotions across time statistically. Word clouds were generated for each of the four emotions based on the top frequent unigrams and bigrams.

**J. Samuel, G. Ali, M. Rahaman et al. [**6] have identifed public sentiment associated with the pandemic using Coronavirus specific Tweets and R statistical software, along with its sentiment analysis packages. They demonstrate insights into the progress of fear-sentiment over time as COVID-19 approaches peak levels in the United States, using descriptive textual analytics supported by necessary textual data visualizations.

**Dhiraj gurkhe, Niraj pal et al.** [7] discussed how twitter data is processed firstly they collected data from various sources and eliminate those features which does not contribute to find any polarity and then this data send into the sentiment classification engine i.e., naïve bayes classification algorithm which will calculate the probabilities i.e., how much data is corrected and predict the sentiment for the given query.

**Pulkit et al. [**8] built and proposed a model which extract tweet from twitter based on the post terror activities. they made their study on terrorist attack which was occurred in uri on 18 September 2016. They considered 59,988 tweet which had taken after the attack. They consider only those tweets which has #UriAttack, #uriattack. #uriattacks. They have used the naïve bayes and SVM to extract the last re-tweet time and number of re-tweets

**S. Zivanovic et al.** [9] has explores the use of social media in Quality-of-Life (QoL) research by capturing and mapping people’s perceptions about their life based on geo-located Twitter data. The methodology is based on a mixed-method approach, combining manual coding of the messages, automated classification and spatial analysis.

**A. Dubey** [10] has done the country wise sentiment analysis of the tweets which has taken into account the tweets from twelve countries. These tweets have been gathered from 11th March 2020 to 31st March 2020, and are related to COVID19 in some or the other way. This analysis has been done to analyze how the citizens of different countries are dealing with the situation. The tweets have been collected, pre-processed, and then used for text mining and sentiment analysis.

**U. Yaqub, V. Atluri et al.** [11] have done study on Sentiment based Analysis of Tweets during the US Presidential Elections during November 2016.The main objective of their study was to identify the nature and sentiment of discussions along with understanding the behavior of users with respect to their Twitter profile and associated attributes of their tweets.

**R. Medford, S. Saleh, A. Sumarsono et al.** [12] extracted tweets matching hashtags related to COVID-19 and measured frequency of keywords related to infection prevention practices, vaccination, and racial prejudice. They have performed a sentiment analysis to identify emotional valence and predominant emotions by using topic modeling to identify and explore discussion topics over time.

**Mohammed Alhajji, A. Khalifah, M. Aljubran et al. [**13] used Naïve Bayes machine learning model to run Arabic sentiment analysis of Twitter posts through the Natural Language Toolkit (NLTK) library in Python. Tweets containing hashtags pertaining to seven public health measures imposed by the government were collected and analyzed.

**K. Suppala, N. Rao** [14] has tried to perform sentiment analysis on twitter data by using a Naïve Bayesian algorithm. By using this model, they have measured the customer opinions and perceptions.

**B.P Pokharel** [15] has done the study on twitter Sentiment analysis during COVID-19 Outbreak in Nepal based on two specified hashtags keywords: #COVID-19 and #coronavirus. The data are collected from the users who shared their location as ‘Nepal’ between 21st May 2020 and 31st May 2020.

**Abhilash Mittal** [16] has performed Sentiment Analysis on Twitter data using Machine Learning Techniques.The study focuses on techniques and types of sentiment analysis to extract tweets from twitter. Further, different machine learning techniques are compared on the same dataset and also found the standard measures.

## 

## **2.2 MOTIVATION**

The emergence in the last decade of social media platforms such as Twitter, Facebook, and Instagram, enabled people to engage in social activities to express their opinions, thoughts, and emotions on a variety of topics. We have chosen to work with twitter since it is a better approximation of public sentiment and has a larger the amount of relevant data. Moreover, the response on twitter is prompt and general. The motivation behind this work has been to portray the fact that how people are behaving in the grim times of a pandemic. This particular pandemic seems to have claimed millions of lives and is leading the world to an utter recession in terms of economy. So, a structured information of people’s sentiment during this crucial time is desirable for future reference and analysis.

# METHODOLOGY

## 

## **3.1.** **TECHNOLOGY USED:**

### **3.1.1 Python Programming:**

Python is a popular object-oriented programing language having the capabilities of high-level programming language. It's easy to learn syntax and portability capability makes it popular these days. Python has huge number of modules for covering every aspect of programming and are easily available for use. As being open-source programming language, Python is supported by a very large developer community. Due to this, the bugs are easily fixed by the Python community. This characteristic makes Python very robust and adaptive. Python is considered as one of most popular language for Machine learning and data science. Python has an extensive and powerful set of packages which are ready to be used in various domains. It also has packages like NumPy, SciPy, pandas, scikit-learn etc. which are required for machine learning and data science. Another important feature of Python that makes it the choice of language for data science is the easy and fast prototyping. This feature is useful for developing new algorithm. The field of data science basically needs good collaboration and Python provides many useful tools that make this extremely. A typical data science project includes various domains like data extraction, data manipulation, data analysis, feature extraction, modelling, evaluation, deployment and updating the solution. As Python is a multi-purpose language, it allows the data scientist to address all these domains from a common platform. The most updated version of python is python3. It is the updated version of python2 which is quite popular.

### **3.1.2. Anaconda and Jupyter:**

Anaconda is a popular python packages in data science consisting of several python libraries and a packet manager called conda. Conda is a [package manager](https://en.wikipedia.org/wiki/Package_manager) that installs, runs, and updates packages and their dependencies and is more efficient that PIP Some of the popular packages are NumPy, SciPy, jupyter, nltk, scikit-learn etc.

Jupyter is an interpreter that is based on browser that helps to work on python and R. Anaconda consist jupyter libraries. Jupyter is amazing tool for the analytical work where we could show our code in “modules” adding common formatting option between modules and include of formatted output of modules and generate the graph in well suited manner in other modules code. Jupyter assure reproducibility so if someone come back after months then they can understand by seeing the code he/she will easily get what someone has tried to do. And can exactly tell which code run which conclusion and visualization.

### **3.1.3. Tweepy:**

Tweepy is an open-source Python package that gives you a very convenient way to access the Twitter API with Python via basic Authentication and the newer method, OAuth. Tweety gives access to the well documented Twitter API. Tweepy includes a set of classes and methods that represent Twitter’s models and API endpoints, and makes it possible to get an object and use any method that the official Twitter API offers. The main Model classes in the Twitter API are Tweets, Users, Entities, and Places. Access to each returns a JSON-formatted response and traversing through information is very easy in Python.

**3.1.4. Scikit-learn:**

Another useful and most important python library for Data Science and machine learning in Python is Scikit-learn.It is built on NumPy, SciPy, and Matplotlib. It is an open source and can be reused under BSD license. It is accessible to everybody and can be reused in various contexts. Wide range of machine learning algorithms covering major areas of ML like classification, clustering, regression, dimensionality reduction, model selection etc. can be implemented with the help of Scikit-learn.

**3.1.5. Pandas:**

It is a useful Python library that is used for data manipulation, wrangling and analysis. It was developed by Wes McKinney in 2008. With the help of Pandas, we can load, prepare manipulate, model and analyze the data in data processing.

### **3.1.6. NLTK:**

The Natural Language Toolkit is a suite of program modules, data sets and tutorials supporting research and teaching in computational linguistics and natural language processing. NLTK is written in Python and distributed under the GPL open-source license. Over the past year the toolkit has been rewritten, simplifying many linguistic data structures and taking advantage of recent enhancements in the Python language.

## **METHODS AND TECHNIQUES**

### **3.2.1 Data Extraction**

Twitter is a platform that provides a user to access data and allows to use it for self-purpose. With the help of tweepy, we can extract the tweets we want easily from twitter through twitter API. After the generation of API, twitter provides access keys such as customer keys, access token key, customer secret key, access secret key. These key plays important role when any user wants to get the data. The major part of this study is to collect the tweet and to analyze those tweets which are related to Covid-19. Twitter Streaming API allows two modes of accessing tweets: Sample Stream and Filter Stream. Sample Stream simply delivers a small, random sample of all the tweets streaming at a real time. Filter Stream delivers tweet which match a certain criterion. It can filter the delivered tweets according to specific keyword to track/search for in the tweets, specific twitter user according to their name and tweets originating from specific locations. In our study, we have used keywords like “coronavirus”, “covid-19”, “pandemic”, “vaccine” and “covid-test”.

### **3.2.2 Labeling Sentiments**

**VADER**

To create a sentiment analysis model, we firstly need to provide sentiment labels (positive , negative and neutral) to our data. For this we use VADER. When it comes to analyzing comments or text from social media, the sentiment of the sentence changes based on the emoticons, along with slang, capitalization etc. So, we have choosen VADER model as it adjusts its rule-based compound score according to emojis and other common social media features that other sentiment models like textblob do not take account of. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a pre-built model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. It relies on a dictionary which maps lexical features to emotion intensities called sentiment scores.

It is available in the NLTK package and can be applied directly to unlabeled text data. VADER’s SentimentIntensityAnalyzer() takes in a string and returns a dictionary of scores in each of four categories:

* negative
* neutral
* positive
* compound (computed by normalizing the scores above)

### **3.2.3 Text Processing**

The tweet text goes through preprocessing step where the following steps are taken. These steps convert plain text of the tweet into processable elements with more information added that can be utilized by feature extractor.

**Cleaning of raw text**

This phase involves the deletion of words or characters that do not add value to the meaning of the text. Some of the standard cleaning steps are below:

* Lowering case: The words, ‘Tweet’, ‘TWEET’, and ‘tweet’ all add the same value to a sentence. Lowering the case of all the words helps to reduce the dimensions by decreasing the size of the vocabulary.
* Removal of mentions: Mentions are very common in tweets. However, as they don’t add value for interpreting the sentiment of a tweet, we can remove them. Mentions always come in the form of ‘@mention’, so we can remove strings that start with ‘@’.
* Removal of punctuation, special characters and numbers: As punctuations, numbers and special characters do not help much. It is better to remove them from the text just as we removed the twitter handles. Hence, we replace everything except characters and hashtags with spaces
* Removal of stop words: Stop words are commonly occurring words in a language, such as ‘the’, ‘a’, ‘an’, ‘is’ etc. We can remove them here because they won’t provide any valuable information for our Twitter data analysis.
* Removal of hyperlinks: It’s not uncommon for tweets to contain URLs, but we won’t need to analyze them for our task so we remove them from the data.

**Tokenization**

Tokenization is the process of splitting text words, punctuation marks, numeric digits, etc., called tokens. Token are the basic building blocks of document object. Everything that helps us to understand the meaning of text is derived from token and their relationship to one other. Each token is an input to the machine learning algorithm as a feature. Tokens can be separated by whitespace characters and/or punctuation characters. It is done so that we can look at tokens as individual components that make up a tweet. Emoticons and abbreviations (e.g., OMG, WTF, BRB) are identified as part of the tokenization process and treated as individual tokens. NLTK (Natural Language Toolkit) provides a utility function for tokenizing data.

**Lemmatization**

Lemmatization the process of removing and replacing suffixes from a token to obtain base or dictionary form of a word, also known as the lemma. It is similar to stemming, in turn, it gives the stripped word that has some dictionary meaning. For example, Lemmatization clearly identifies the base form of ‘troubled’ to ‘trouble’’ denoting some meaning whereas, Stemming will cut out ‘ed’ part and convert it into ‘troubl’ which has the wrong meaning and spelling errors.

Example:

‘troubled’ ‘trouble’

troubled’ ‘troubl’

### **3.2.4 Feature Extraction**

Analyzing text data is an extremely complex task for a machine as it’s difficult for a machine to understand the semantics behind text. So, after the data is preprocessed, it needs to be converted into features for further analysis which simply means the text data should be converted into a machine-understandable format. However, it’s important to note that performing this transformation could cause data loss. Therefore, key then is to maintain an equilibrium between conversion and retaining data. There are two commonly used terminologies when it comes to feature extraction i.e.

* Each text data point is called a **Document**
* An entire set of documents is called a **Corpus**

Depending upon the usage, text features can be constructed using assorted techniques – Bag-of-Words, TF-IDF, and Word2vec.

**Bag of Words**

Bag of Words is a set of features where the frequency of tokens is indicated in a feature vector. In simple words, A bag of words is a representation of text that describes the occurrence of words within a document disregarding the order in which they appear. It is called a “bag” of words because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document.

An entry in the feature vector is assigned to each unique token found in the labeled training set. If the respective token occurs in a tweet, it is assigned a binary value of 1 otherwise it is 0. The effectiveness of using bag of words for sentiment analysis have been reported that indicating only presence of a word yields higher performance than indicating the frequency of a word. The reason behind this behavior may be that the sentiment class is not usually varied if certain words occur more than once in the text.

**Term Frequency — Inverse Document Frequency**

Term Frequency refers to the relationship between a word and a document whereas Inverse Document Frequency refers to the relationship between a word and the corpus.TF-IDF uses the relationship between these elements to convert text data into vectors. Converting a document into a vector, using TF-IDF values is called TF-IDF vectorization. TF-IDF allows us to understand the context of words across the entire corpus of documents instead of just its relative importance in single document.

**Calculating Term Frequency**

Term frequency is the probability of the word wⱼ in the document dᵢ. It is calculated as below:

**Calculating IDF**

Inverse Document Frequency (IDF) says how frequently a word occurs in the entire corpus. This is calculated below.

Now,

***TF-IDF =Term frequency \* Inverse document frequency*** *i.e.*

***TF-IDF****= TF(word, document) \* IDF(word, corpus).*

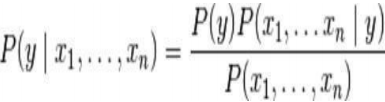
TF-IDF vectorization gives high importance to words which are:

* frequent in a document (from TF)
* rare in the corpus (from IDF)

### **3.2.5 Training Classifier**

**Naïve bayes**

The Naive Bayes Classifier is a well-known machine learning classifier with applications in Natural Language Processing (NLP). Naïve bayes is a “probabilistic classifier” that applies Bayes rules for forming classification probabilities. Despite its simplicity, it is able to achieve above average performance in different tasks like sentiment analysis. Naïve bayes is beneficial for big data sets and can be built easily. Let us consider a class variable ‘y’ and a dependent vector from x1 to xn. So according to naïve bayes:



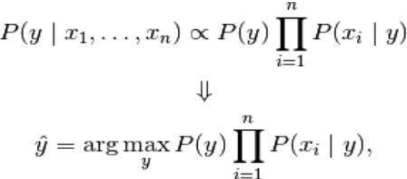
so according to mutually independent assumption:



for each value of i this function behaves:



Joint model could be expressed as:



We have considered Naïve Bayes as our Classifier Model for following reasons:

1. When dealing with text, it’s very common to treat each unique word as a feature, and since the typical person’s vocabulary is many thousands of words, this makes for a large number of features. So, the independent features assumption of Naive Bayes classifier makes it a strong performer for classifying texts.
2. It is simple and it converges quicker than other discriminative models like logistic regression, so we need less training data.
3. It is easy and fast to predict class of test data set and perform well in multi class prediction.

# DESIGN

Figure 1: General Architecture for Twitter Sentiment Analysis

# IMPLEMENTATION

## **DATA EXTRACTION**

For initializing our data extraction process, we accessed information from twitter using Twitter API. We created a twitter developer account and accessed the keys and tokens needed for twitter api by creating an app. Once we got all four authorization keys: **API key**, **API secret key, Access token**, **Access token secret, we were ready to extract our data from twitter. We have used Tweepy python library for data extraction from Twitter API (Application programming interface). W**e have passed following parameters to our api for searching tweets:

**q**: This is the keyword to be searched in the tweet. For our project, we pass six different keywords: *‘corona virus’, ‘covid-19’, ‘lock down’, ‘vaccine’, covid test* and *‘pandemic’.*

**lang**: This is the language of the tweets we want to fetch from the API. In this study, we’ll be retrieving the tweets that are in English.

**since**: This is the date from which we want to retrieve tweets. For our purposes, we have collected tweets 8th May, 2021 to 15th May, 2021.

We have created four features for our datasets: text, user, location, date and keyword and stored into a csv format with the help of pandas. The following tables show the retrieved data from Twitter API of one week after dropping duplicates tweets or spam tweets having word ‘giveaway’.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Date |  |  | No. of tweets |  |  |  | Subtotal tweets |
| corona virus | covid-19 | pandemic | lockdown | vaccine | covid test |  |
| 5-9-2021 | 410 | 779 | 1196 | 1335 | 1120 | 902 | 5742 |
| 5-10-2021 | 472 | 1222 | 1048 | 1236 | 1399 | 649 | 6026 |
| 5-11-2021 | 777 | 1175 | 1155 | 1049 | 1339 | 760 | 6255 |
| 5-12-2021 | 804 | 1270 | 1278 | 1225 | 1345 | 969 | 6891 |
| 5-13-2021 | 627 | 1116 | 1023 | 1224 | 1363 | 790 | 6143 |
| 5-14-2021 | 776 | 1221 | 1199 | 1261 | 1436 | 749 | 6642 |
| 5-15-2021 | 908 | 1049 | 840 | 1135 | 1270 | 851 | 6053 |
| Total tweets per week | 4774 | 6657 | 7703 | 8465 | 9272 | 5670 | 43752 |

Table 1: Twitter Data about Covid-19 during one week

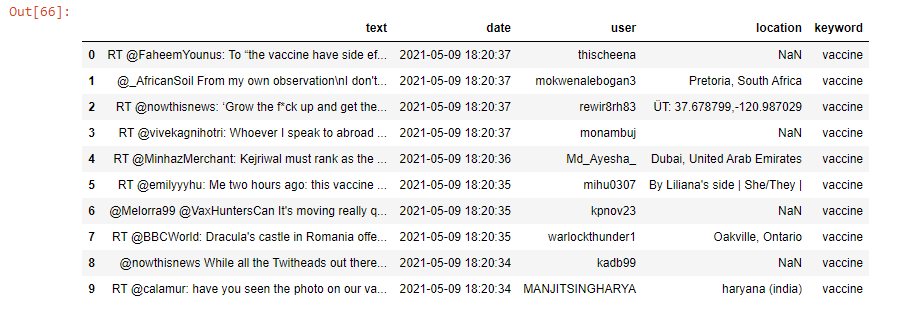
After collection of data for one week, we concatenated all those datasets into a single data frame and stored in csv format as total tweets.csv.

Figure 2: Displaying first 10 rows of our dataset

## **DATA CLEANING FOR VADER**

Before feeding our data to Sentiment Analysis library i.e., VADER, we need to remove unnecessary characters that don't add any value in sentiment analysis of our data. So, for this, we prepare a function including the following points to clean our tweets:

1. Remove mentions (@username)
2. Remove RT
3. Remove hyperlinks and URLs

We used ‘re’, built-in package of Python for data cleaning process.

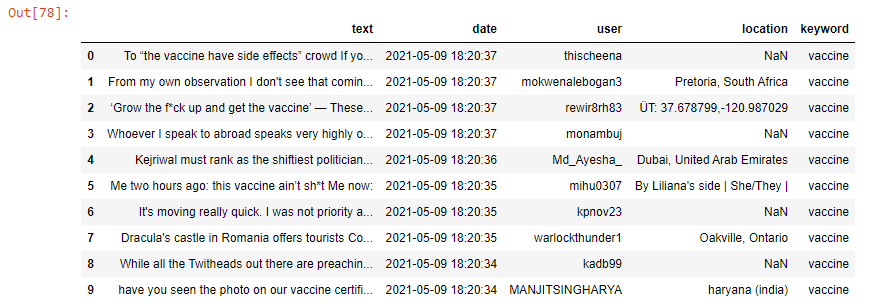


Figure 3: Displaying first 10 rows of our dataset after removing @’s, mentions and urls

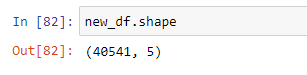
Now that we removed RT and mentions from our text data, we again check for duplicate dataset and drop them if any. There were 3211 duplicates data dropping which we got total of 40541 as final tweets.

Figure 4: Total numbers of rows and columns in our dataset after removing duplicate data

## **CREATING SENTIMENT LABELS USING VADER**

VADER’s SentimentIntensityAnalyzer() takes in a string and returns a dictionary of scores in each of four categories:

* Negative
* Neutral
* Positive
* Compound (computed by normalizing the scores above)

Vader returns positive, negative and neutral value and based on which another fourth label i.e., compound value is computed. To create the sentiment labels for the new\_df dataset, we created a function called SentimentPredict() and provide it the compound score to determine whether they are positive, negative, or neutral. In the function, if the text has compound score less than or equal to -0.05, it was labeled as negative. Similarly, if the compound score of text is more than or equal to 0.05, it was labeled positive and else we labeled our data as neutral. The results of the SentimentPredict function were saved in a new column called, label.

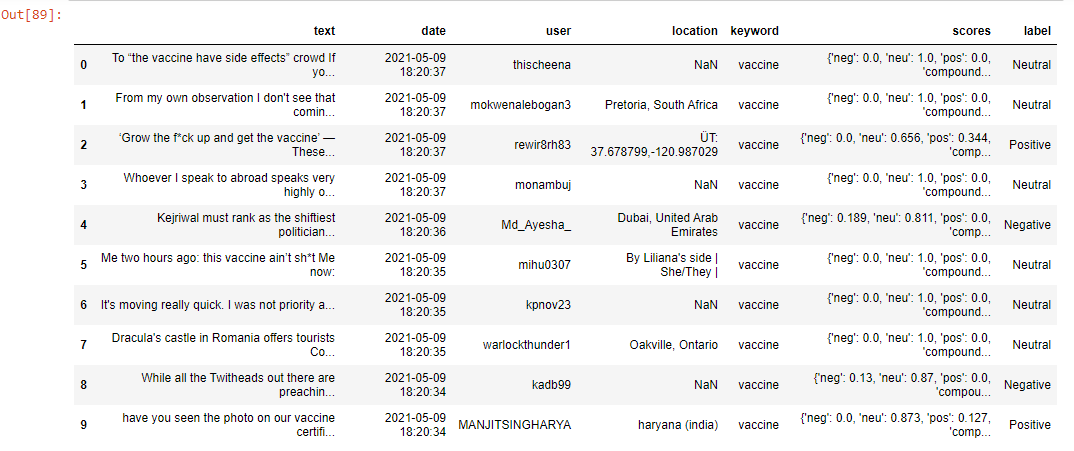


Figure 5: 10 rows of our dataset with sentiment labels

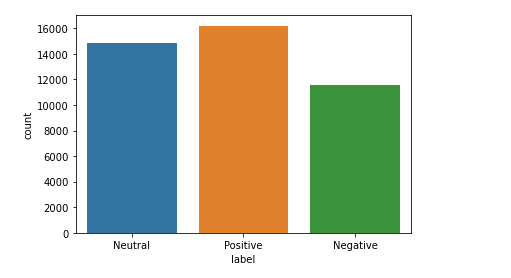
As shown below in the figure 6, there were total of 13,984 neutral labels, 11,181 negative labels, and 15,376 positive labels under the label column in the new\_df dataset. The new\_df data frame with the sentiment labels was exported as a CSV file named, tweets\_sentiment.csv.

Figure 6: Visualization of negative,positive and neutral using seaborn plot

## **PREPROCESSING THE DATA**

Now that we have corpus of tweets, preprocessing of the data was done on the text column to prepare the new\_df dataframe for the NLP analysis of the project. Modifications we do during this process directly impacts the classifier’s performance. Preprocessing includes cleaning, normalization, transformation, feature extraction and selection, etc. All of the tweets are preprocessed by passing through the following steps in the same order.

### **5.4.1 Removing Emoticons**

A function called, emoticon\_text, was created to convert emoticons into their respective texts via a custom dictionary containing commonly used emoticons with their respective textual meanings (ex. :) : happy / smile, xD : laugh, :( : frown /sad / pouting, etc.) before it was applied on the text column with the new column called, cleaned\_text, which contains the cleaned text of the Tweets.

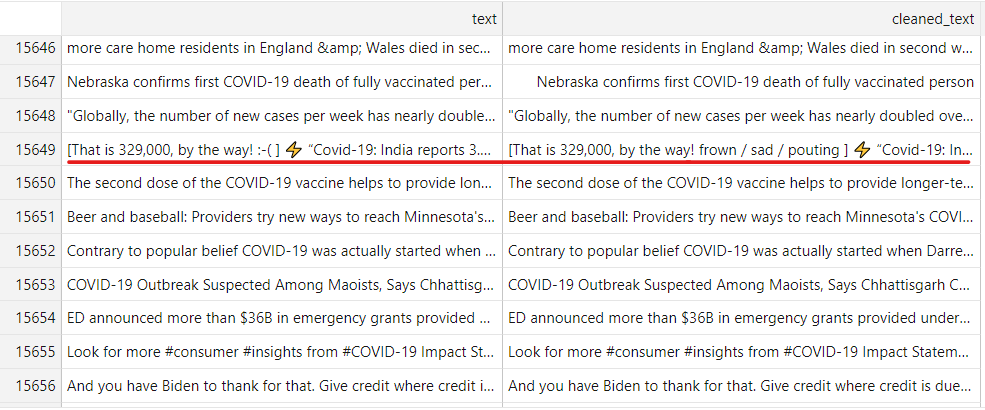


Figure 7: list of tweets before and after applying emoticon\_text()

### **5.4.2 Removing Emojis**

Then, a function called, emoji\_cleaner, was made to convert emoji (ex. 😂, ❤, 😍 etc.) into their respective text by using demojize() from python package called emoji under the cleaned\_text column.

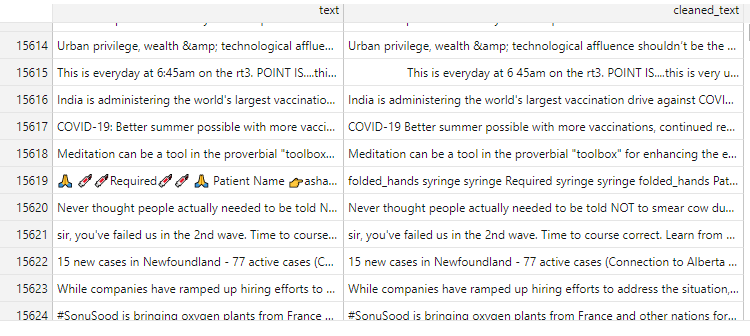


Figure 8: Tweets before and after applying emoji\_cleaner()

### **5.4.3 Acronyms**

**An acronym dictionary** of 5465 acronyms with their translation was used to replace all the acronyms present in our tweets. Most used acronyms in our data is represented below in bar chart.

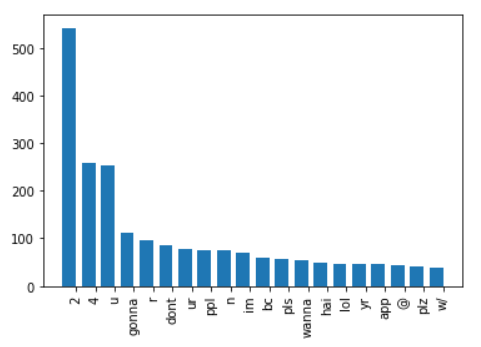


Figure 9: Top 20 of acronyms in the data set of tweets

As you can see, “2”, “4”, “u”, “gonna” are really often used by users. The table below shows the top 20 acronyms with their translation and their count.

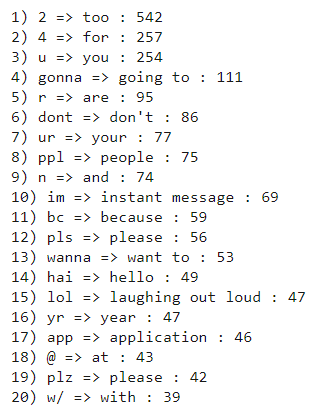
**

Figure 10: Top 20 of acronyms in the data set of tweets with their translation and count

### **5.4.4. Lowering cases and Removing Contractions**

A function called, replace\_contractions, was written to convert contractions in the text of the cleaned\_text column into their basic words (ex. don’t: do not, hadn’t: had not, hadn't've : had not have, etc.). Similarly, lower () was used to convert all the tweets into lowercase.

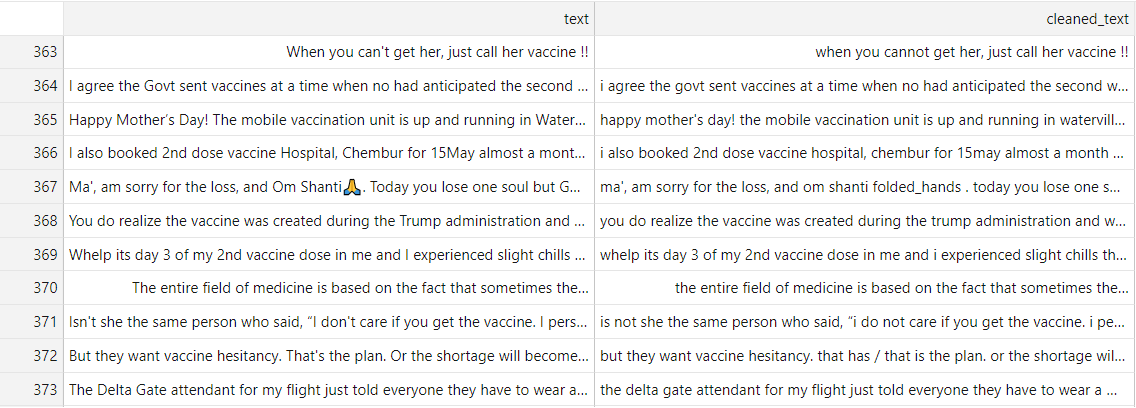


Figure 11: Tweets before and after lowering cases and removing contraction

### **5.4.5 Removing Punctuations, numbers and special characters**

A function called, tweet\_cleaner, was made to remove to remove punctuations, special characters or number and keep only alphabetical words under the cleaned\_text column.

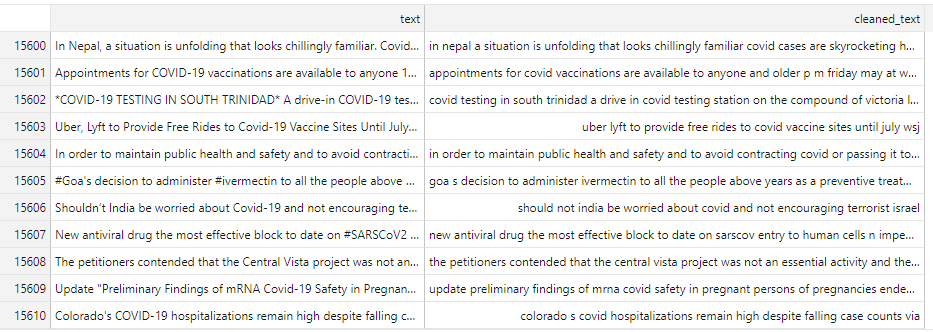


Figure 12: Tweets before and after removing punctuations, numbers, special characters.

### **5.4.6 Tokenization**

At this point, tweets were also tokenized by using string.split() instead of nltk.tokenize in order to do the process really fast.



Figure 13: Tweets before and after performing tokenization

## **5.5. FEATURE EXTRACTION**

For feature extraction, we have considered two common methods i.e. Bag of words and TF-IDF and selected the best method based on their performance.

**Bag of Words:** We used *CountVectorizer* from sklearn for this purpose. Count Vectorizer counts the number of words in the document.

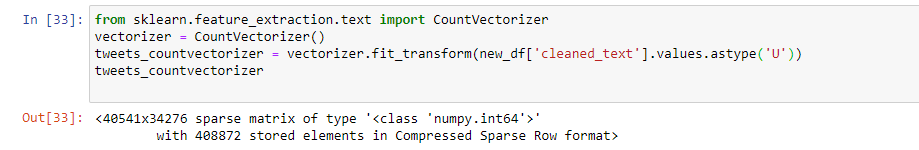


Figure 14: Using CountVectorizer as our feature extractor

**Term Frequency–Inverse Document Frequency:** For TF-IDF, we have imported TfidfVectorizer from sklearn.

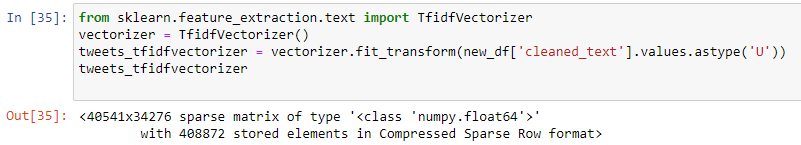


Figure 15: Using TfidfVectorizer as our feature extractor

After performing these feature extraction techniques, we observed that CountVectorizer gives better accuracy in this sentiment analysis than tf-idf.

## **SPLITTING DATASET**

After performing feature extraction on our data we divided the dataset into two parts, the training set and the test set for cross validation. We split them into 8:2 ratio of (train: test) data randomly and get 4 components: X\_train, y\_train, X\_test and y\_test, where X and y are features and labels respectively. We got total of 32432 tweets as training set and 8109 tweets as test sets.

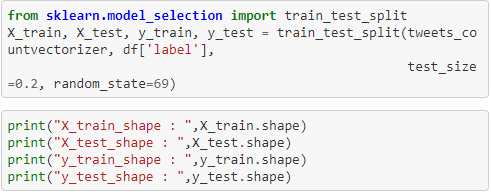


Figure 16: Splitting our dataset into train and test set



## **APPLICATION OF NAÏVE BAYES ALGORITHM**

Once data was extracted and preprocessed the next step was the implementation of the classification algorithm. As we mentioned in methodology section of our report, we have used Naïve Bayes Classifier model. There are mainly their variants of Naive Bayes classifiers: **Multi-variate Bernoulli Model, Multinomial Model, and Gaussian Model**. As Multinomial Classification model is considered as the most effective method for text classification. Hence, we decided to move forward with this classifier model.

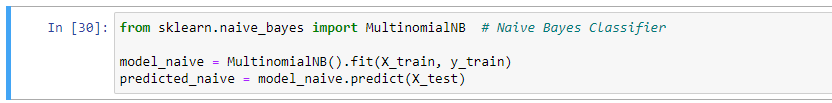


Figure 17: Training the model by passing X\_train and y\_train

After splitting the data into train set and test set, we passed our variables X\_train and y\_train inside MultinomialNB() and trained our model. After training our model we then tested it by passing on X\_test(which is corpus of tweets of testing data).

## **5.8. ADVANCED PRE-PROCESSING:**

To further improve the accuracy of the classifier, another filtering module was build to be used in the training stage of the classification. So, two approaches were considered:

1. Using lemmatization to perform word reductions.

2. Excluding tokens with low sentiment value (Stop words removal).

### **5.8.1. Lemmatization:**

A function called, lemmatize\_text was written to help to lemmatize the words of the text under the cleaned\_text column.

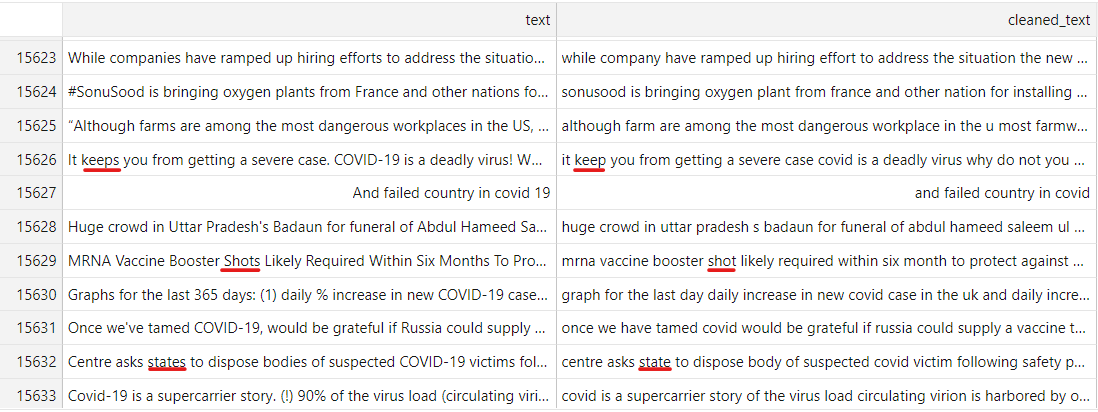


Figure 18: Tweets before and after applying lemmatize\_text()

### **5.8.2. Removing Stop words**

As list of stop words from nltk contains words like "no", "nor" etc. which plays significant roles in sentiment analysis, so instead of using them we created our custom list of stopwords.



Figure 19: list of customized stop words under stop\_words

Then we create a function called remove\_stopwords to remove these stop words from our tweets which is under cleaned\_text column.

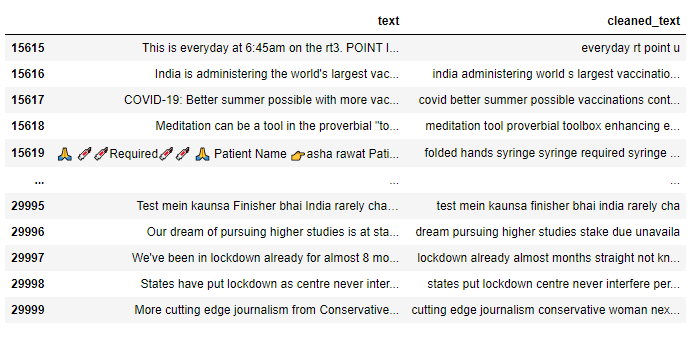


Figure 20: Tweets before and after removing stopwords

## **5.9. ASSOCIATION RULES:**

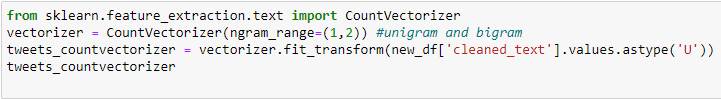
Earlier we used only single word features in our model, which we call 1-grams or unigrams. For the further improvement of predictive power to our model we can use n-grams in the form of bigrams and trigrams. So, we used the ngram\_range argument with any of the ‘Vectorizer’ classes provided by scikit-learn library and compared the accuracy we get from trying different combinations.

Figure 21: Using ngram\_range along with the vectorizer

# 6. RESULT AND ANALYSIS:

## **6.1. PERFORMANCE METRICS OF SENTIMENT CLASSIFICATION:**

After our machine learning process is complete, we have to use performance metrics to evaluate how our model did. Generally, to measure the performance we use some predefined standards such as accuracy, precision, recall and f1 score.

**Accuracy** in classification problems is the number of correct predictions made by model divided by the total number of predictions.



Where,

tp =true positive, case was positive and it predicted positive

tn =true negative, case was negative and it predicted negative

fn =false negative, case was positive and it predicted negative

fp = false positive, case was negative and it predicted positive

It is suitable for balanced classes. In case of unbalanced classes, we need to introduce recall and precision.

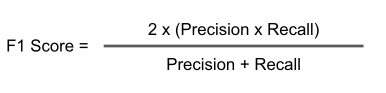
**Recall**: Recall is the ability of the machine learning model to find all the relevant cases within a data set.



**Precision**: Precision is the ability of a classification model to identify only the relevant data points.



**F1 score**: The F1 score is a harmonic mean of Precision and Recall measured when one of the values goes very high or very low and it becomes difficult to take a decision based on only Precision or Recall. Then we can decide upon the validity of the result by considering the F1 score.



The classification report after calculating all the above metrics for our model is given below:

Table 2: Classification Report of performance metrics

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | F1-score |
| Negative | 0.68 | 0.69 | 0.69 |
| Neutral | 0.78 | 0.66 | 0.72 |
| Positive | 0.72 | 0.81 | 0.76 |
| Accuracy |  |  | 0.73 |

## **6.2. CONFUSION MATRIX**

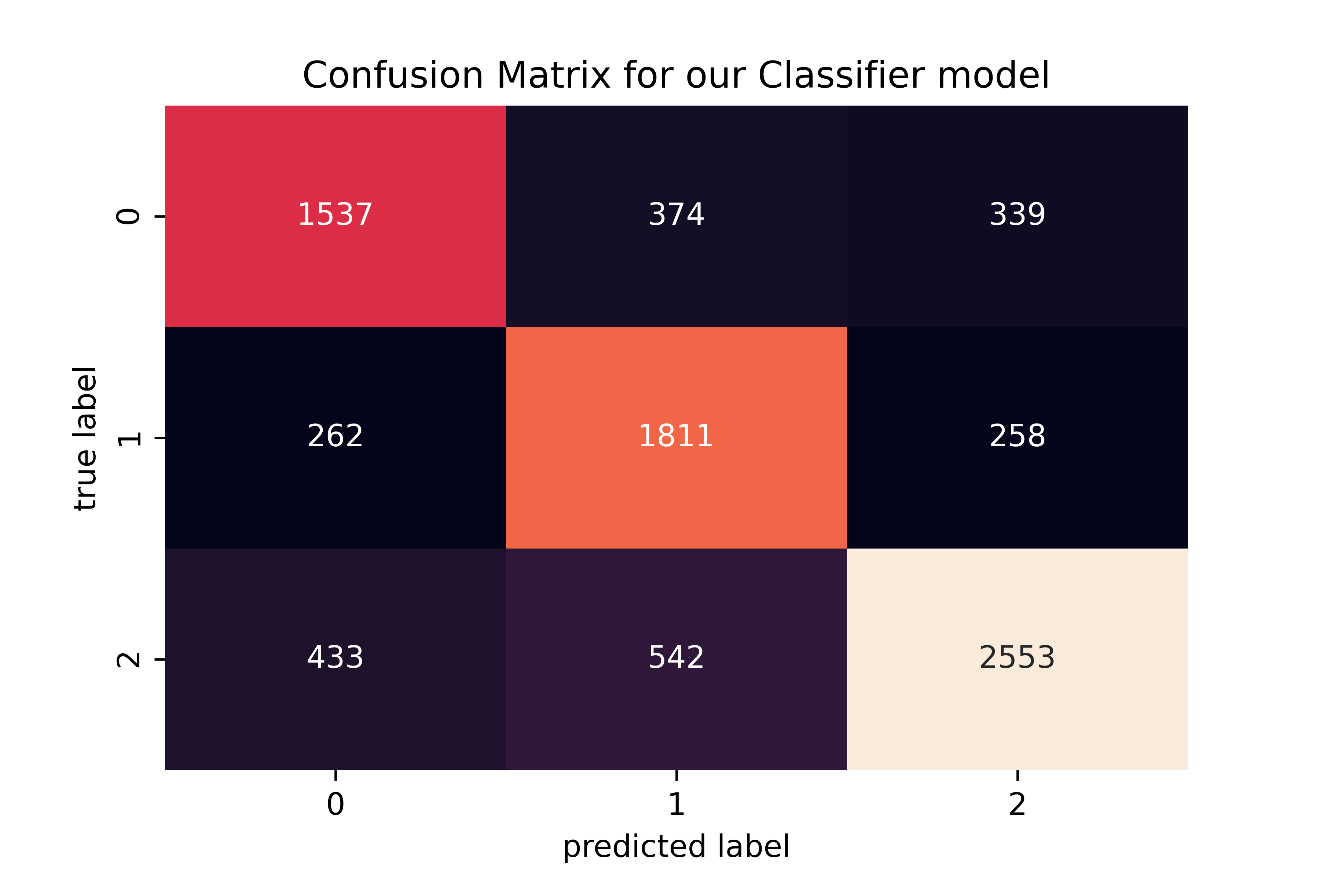
During the testing phase of our classification process we have two categories:

* True condition
* Predicted condition

And we can represent the comparison between these predicted values and true values using confusion matrix. Table 3 below shows the basic format of a confusion matrix:

Table 3: Format of Confusion matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted | Condition |
|  | Total Population | Predicted Positive | Predicted Negative |
| True Condition | Condition positive | True Positive (TP) | False Negative (FN) |
|  | Condition negative | False positive (FP) | True Negative (TN) |



## **6.3. COMPARISON OF ACCURACY FROM DIFFERENT VECTORIZER**

For achieving the better accuracy, we tried to build the pipeline using different filtering modules and feature extractors and compared their results.

Table 4: Comparing accuracy obtained by using different vectorizers and preprocessing steps

|  |  |  |  |
| --- | --- | --- | --- |
| **Normalization** | **Stop words Removal** | **Vectorizer** | **Accuracy** |
| None | no | Count | 0.711925021580959 |
| Lemmatizing | no | Count | 0.711185102972006 |
| None | yes | Count | 0.727709951905290 |
| Lemmatizing | yes | Count | 0.723147120483413 |
| None | Yes | Tf-Idf | 0.701566161055617 |
| Lemmatizing | Yes | Tf-Idf | 0.701442841287458 |
| None | No | Tf-Idf | 0.684548033049697 |
| Lemmatizing | No | Tf-Idf | 0.683191515599950 |

From the above table it is clear that best combination for sentiment analysis with our dataset is using

* Without normalization
* Without stop words
* Count vectorizer

## **6.4. ACCURACY IN N-GRAMS METHODS**

Different N-gram methods were passed in our vectorizer to check which combination gives us the best accuracy which is shown in the table below:

Table 5: Comparing accuracy obtained by using different n-gram methods

|  |  |
| --- | --- |
| **N-gram Methods** | **Accuracy** |
| Unigram and Bigram | 0.721174 |
| Bigram and Trigram | 0.58268 |
| Unigram and Trigram | 0.70760 |

So, we concluded that with unigram and bigram, without any normalization, stop words or tf-idf we get the accuracy of 0.72117.

7. CONCLUSION AND FUTURE WORK

One of the major goals of this research was to extract tweets from twitter and create dataset. We were able to collect 40541 tweets of which 30984 were labeled neutral, 11181 were negative and 15376 were positive by using VADER. From this first stage of our study we concluded that despite of the hardships this outbreak has brought in the lives of people, most of them are trying to cope with the disease positively followed by some people with neutral opinions. Now moving on to the second stage of our study, the labelled tweets that we had collected were a mixture of words, emoticons, hashtags, and symbols. So, before training the dataset we pre-processed the tweets to make it suitable for feeding into models. We implemented machine learning algorithm, Naive Bayes Classifier to classify the polarity of the tweet. We further tried to improve the performance of our classifier by using additional prepressing steps which yielded the best accuracy of 0.7277.

## **7.1. LIMITATION AND FUTURE SCOPE:**

Due to lack of time, and computational process, study was limited to only certain extend and hence many aspects has been left for the future works. It would be interesting to take the following research area into consideration:

1. Accuracy that we have obtained for our classifier model is not ideal. So, we can further boost classification metrics of our classifier model by adding hyper parameters tuning to the process and using techniques such as syntactic and semantic features.
2. This model was limited to work only on English Tweets but the better improvement would be if the model could analyze tweets in other languages like Nepali, Hindi so that we could have broader range of emotions and sentiments.
3. Similarly, instead of simply using the conventional positive and negative sentiment, we can also train this model to extract opinions from tweets and to evaluate their sentiments in terms of emotions such as joy, surprise, anger, or fear and so on.
4. The analysis still faces challenges like sarcasm in tweets, taking care of spelling mistakes and spams in tweets where we can conduct further researches.
5. Along with twitter, exploring other social media platforms such as Facebook, Instagram can help us to get opinions from larger range of people which is very essential regard to sentiment analysis.

# WORKING SCHEDULE

The study is done as major project of the eighth semester. Considering the output expected to receive after the completion of this study, we worked on the project according to the working schedule presented below:

# REFERENCES:

*[1] Koyel Chakraborty, Surbhi Bhatia, Siddhartha Bhattacharyya, Jan Platos, Rajib Bag, Aboul Ella Hassanein “Sentiment Analysis of COVID-19 tweets by Deep Learning Classifiers” Applied Soft Computing Journal, September 2020.*

*[2] Kamaran H. Manguri, Rebaz N. Ramadhan ,Pshko R. Mohammed Amin “Twitter Sentiment Analysis on Worldwide COVID-19 Outbreaks” Kurdistan Journal of Applied Research, May 2020*

*[3] A. Bastola et al., “The first 2019 novel coronavirus case in Nepal,” Lancet Infect. Dis., vol. 20, no. 3, pp. 279–280, 2020.*

*[4] R. Muthusami, A. Bharathi, and K. Saritha, “Covid-19 outbreak: Tweet based analysis and visualization towards the influence of coronavirus in the world,” Gedrag en Organ., vol. 33, no. 2, pp. 534–549, 2020.*

*[5] M. O. Lwin et al., “Global sentiments surrounding the COVID-19 pandemic on*

*Twitter (Preprint),” JMIR Public Heal. Surveill., vol. 6, pp. 1–4, 2020.*

*[6] J. Samuel, G. Ali, M. Rahman, E. Esawi, and Y. Samuel, “COVID-19 Public Sentiment Insights and Machine Learning for Tweets Classification,” SSRN Electron. J., no. May, pp. 1–21, 2020.*

*[7]Gurkhe D., Pal N. and Rishit B. "Effective Sentiment Analysis of Social Media Datasets using Naïve Bayesian Classification." (2014).*

*[8] Pulkit Garg, Himanshu Garg, Virender Ranga “Sentiment Analysis of the Uri Terror Attack Using Twitter” International Conference on Computing, Communication and Automation (ICCCA2017).*

*[9] S. Zivanovic, J. Martinez, J. Verplanke “Capturing and mapping quality of life using Twitter data” Geo Journal, December 2018.*

*[10] Dr. Akash D Dubey, “Twitter Sentiment Analysis during COVID19 Outbreak”, Jaipuria Institute of Management, 2020*

*[11] U. Yaqub, S. Ae Chun, V.Atluri, J. Vaidya, “Sentiment based Analysis of Tweets during the US Presidential Elections”, In Proceedings of ACM Digital Government Research,June 2017 (dg.o'17), 10 pages.*

*[12] R. J. Medford, S. N. Saleh, A. Sumarsono, T. M. Perl, and C. U. Lehmann, “An ‘ Infodemic ’: Leveraging High -Volume Twitter Data to Understand Public Sentiment for the COVID-19 Outbreak,” 2020.*

*[13] Mohammed Alhajji, A. Al Khalifah, M. J. Aljubran, and M. Alkhalifah, “Sentiment analysis of tweets in Saudi Arabia regarding governmental preventive measures to contain COVID-19,” no. April, 2020.*

*[14] K. Suppala and N. Rao,“Sentiment analysis using naïve bayes classifier,” Int. J. Innov. Technol. Explor. Eng., vol. 8, no. 8, pp. 264–269, 2019.*

*[15] Bishwo Prakash Pokharel “Twitter Sentiment analysis during COVID-19 Outbreak in Nepal”, Nepal Open University,May 2020*

*[16] Abhilash Mittal, “Sentiment Analysis on Twitter data Using Machine Learning Techniques”. Master’s Projects, Delhi Technological University, June 2019. [3] Apoorv Agarwal Boyi Xie Ilia Vovsha Owen Rambow Rebecca Passonneau “Sentiment Analysis of Twitter Data", Columbia University, New York, pp 30–38, June 2011*

*[17] Viju Raghupathi, Jie Ren, Wullianallur Raghupathi “Studying Public Perception about Vaccination: A Sentiment Analysis of Tweets”.Article, New York, NY 10023, USA. 15 May 2020.*