RV College of Engineering®
Autonomous institution affiliated to Visvesvaraya Technological University, Belagavi | Approved by AICTE, New Delhi
Go, change the world

Twitter Sentiment Analysis

A Technical Seminar Report

Submitted by,

D Chetan Karthikeya Reddy **1RV17EC034**
Pradyumna C **1RV17EC103**

Under the guidance of

Mrs.Rohini S. Hallikar
Assistant Professor
Dept. of ECE
RV College of Engineering

In partial fulfillment of the requirements for the degree of
Bachelor of Engineering in
Electronics and Communication Engineering
2020-2021

RV College of Engineering®, Bengaluru

(Autonomous institution affiliated to VTU, Belagavi)

Department of Electronics and Communication Engineering

.

CERTIFICATE

Certified that the Technical Seminar titled ***Twitter Sentiment Analysis*** is carried out by **D Chetan Karthikeya Reddy** (**1RV17EC034**) and **Pradyumna C** (**1RV17EC103**) who are bonafide students of RV College of Engineering, Bengaluru, in partial fulfillment of the requirements for the degree of **Bachelor of Engineering** in **Electronics and Communication Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2020-2021. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the Technical Seminar report deposited in the departmental library. The Technical Seminar report has been approved as it satisfies the academic requirements in respect of Technical Seminar work prescribed by the institution for the said degree.

Signature of Guide	Signature of Head of the Department	Signature of Principal
Mrs.Rohini S. Hallikar	Dr. K S Geetha	Dr. K. N. Subramanya

External Viva

Name of Examiners	Signature with Date
1.	
2.	

DECLARATION

We, **D Chetan Karthikeya Reddy** and **Pradyumna C** students of eighth semester B.E., Department of Electronics and Communication Engineering, RV College of Engineering, Bengaluru, hereby declare that the Technical Seminar titled '**Twitter Sentiment Analysis**' has been carried out by us and submitted in partial fulfilment for the award of degree of **Bachelor of Engineering** in **Electronics and Communication Engineering** during the year 2020-2021.

Further we declare that the content of the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

We also declare that any Intellectual Property Rights generated out of this project carried out at RVCE will be the property of RV College of Engineering, Bengaluru and we will be one of the authors of the same.

Place: Bengaluru

Date:

Name	Signature
1. D Chetan Karthikeya Reddy(1RV17EC034)	
2. Pradyumna C(1RV17EC103)	

ACKNOWLEDGEMENT

We are indebted to our guide, **Mrs.Rohini S. Hallikar**, Assistant Professor, RV College of Engineering . for the wholehearted support, suggestions and invaluable advice throughout our Technical Seminar and also helped in the preparation of this thesis.

We also express our gratitude to our examiner **Dr.KS Shushrutha**, Associate Professor , Department of Electronics and Communication Engineering for their valuable comments and suggestions.

Our sincere thanks to **Dr. K S Geetha**, Professor and Head, Department of Electronics and Communication Engineering, RVCE for the support and encouragement.

We express sincere gratitude to our beloved Principal, **Dr. K. N. Subramanya** for the appreciation towards this technical seminar work.

We thank all the teaching staff and technical staff of Electronics and Communication Engineering department, RVCE for their help.

Lastly, we take this opportunity to thank our family members and friends who provided all the backup support throughout the project work.

ABSTRACT

Social media have received more attention nowadays.Public and private opinion about a wide variety of subjects are expressed and spread continually via numerous social media. Twitter is one of the social media that is gaining popularity.People usually use twitter to express their opinions or views or an emotion on a particular subject. Twitter offers organizations a fast and effective way to analyze customers' perspectives toward the critical to success in the market place.

Sentiment Analysis is the process of computationally determining whether a piece of writing such as a tweet is a Positive,Negative or a Neutral News . Sentiment analysis whilst combined with twitter offers beneficial insights into what's expressed on Twitter. The big availability of online evaluations and postings in social media gives invaluable feedback for groups to make better knowledgeable choices in guidance their marketing techniques towards user's pastimes and alternatives. Sentiment evaluation is, therefore, vital for determining the general public's opinion toward selected services or products Developing a program for sentiment analysis is an approach to be used to computationally measure customers' perceptions.

This report discusses on the design of a sentiment analysis, extracting a vast amount of tweets using the Twitter Application Programming Interface (API) and then passing them through the Machine Learning Algorithms to know the accuracy and efficiency with which it's working .

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

API Application Programming Interface

CSV Comma Seperated Values

ML Machine Learning

NLP Natural Language Processing

NLTK Natural Language Toolkit

Rashtreeya Sikshana Samithi Trust
RV
INSTITUTIONS

Chapter 1

Introduction to Twitter Sentiment Analysis

CHAPTER 1

INTRODUCTION TO TWITTER SENTIMENT ANALYSIS

1.1 Introduction

Now-a-days social networking sites are at the boom, so large amount of data is generated. Millions of people are sharing their views daily on micro blogging sites, since it contains short and simple expressions. Twitter, a micro-blogging website, is a massive repository of public opinions expressed in the direction of numerous humans, offerings, companies, merchandise, etc.Twitter is a popular and one of the most used social media website. People usually use twitter to express their opinion or view or an emotion on particular subject.

Sentiment Analysis is the process of computationally determining whether a piece of writing is Positive , Negative or Neutral . Sentiment evaluation is the system of analyzing one's public evaluations. Sentiment analysis whilst combined with twitter offers beneficial insights into what's expressed on Twitter. The big availability of online evaluations and postings in social media gives invaluable feedback for groups to make better knowledgeable choices in guidance their marketing techniques towards user's pastimes and alternatives. Sentiment evaluation is, therefore, vital for determining the general public's opinion toward selected services or products.Sentiment analysis is also referred as Opinion Mining.

Sentiment Analysis is mainly used for 3 main purposes namely :-

- **Bussiness**

  In marketing field companies use it to develop their strategies, to understand customers' feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don't buy some products.

- **Politics**

  In political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well.

- **Public Actions**

  Sentiment analysis also is used to monitor and analyse social phenomena, for the spotting of potentially dangerous situations and determining the general mood of the blogosphere.

The Figure 1.1 shows the 3 sentiments namely positive , negative and neutral

Figure 1.1: Sentiments

1.2 Motivation

1. Sentimental Analysis is a hot topic of research.
2. Use of electronic media is increasing day by day.
3. Time is money or even more valuable than money therefore instead of wasting time by reading and figuring out the positivity and negativity of text we can use automated techniques for sentimental analysis.
4. Sentimental analysis is used in opinion mining.

1.3 Problem statement

To analyze and predict whether the tweet is of positive , neutral or negative sentiment using Machine Learning and NLP techniques.

1.4 Objectives

The objectives of the project are

1. To parse through the Twitter data and create our own dataset .
2. To perform detailed Text pre-processing on the twitter data.
3. To perform Sentiment Analysis of tweets using various Machine Learning (ML) algorithms and also to compare the same .

1.5 Literature Review

- **S. E. Saad and J. Yang, "Twitter Sentiment Analysis Based on Ordinal Regression," in IEEE Systems Journal, vol. 7, pp. 163677-163685, 2019, doi: 10.1109/ACCESS.2019.2952127.**

  This paper proposes a way to analyze the sentiment of twitter threads using Multinomial Logistic Regression algorithm., SVM , Random Forest Tree and Decision Tree . The accuracy obtained was around 82

- **Z. Jianqiang, G. Xiaolin and Z. Xuejun, "Deep Convolution Neural Networks for Twitter Sentiment Analysis," in IEEE Access, vol. 6, pp. 23253-23260, 2018, doi: 10.1109/ACCESS.2017.2776930.**

  A model called GloVe-DCNN is presented which implements the binary task of classifying the tweet into negative or positive sentiment categories

- **H. Rehioui and A. Idrissi, "New Clustering Algorithms for Twitter Sentiment Analysis," in IEEE Systems Journal, vol. 14, no. 1, pp. 530-537, March 2020, doi: 10.1109/JSYST.2019.2912759.**

  Sentiment analysis done using ca combination of k-means clustering algorithm(kNN) and DENCLUE .

- **Z. Jianqiang and G. Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis," in IEEE Access, vol. 5, pp. 2870-2879, 2017, doi: 10.1109/ACCESS.2017.2672677.**

  The paper shows that the accuracy and F1-measure of Twitter sentiment classification classifier are improved when removing URLs, removing stop words. The

NaBs and Random Forest classifiers are more sensitive than Logistic Regression and SVMs classifiers when various pre-processing methods were applied

- **L. Mandloi and R. Patel, "Twitter Sentiments Analysis Using Machine Learninig Methods," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-5, doi: 10.1109/INCET49848.2020.**
  The dataset obtained here was by webscraping twitter and using at as a dynamic database or dataset

- **Sentiment Analysis of Twitter Data: A Survey of Techniques - International Journal of Computer Applications (0975 – 8887) Volume 139 – No.11, April 2016.**
  Various machine learning algorithms like Naive Bayes, Max Entropy, and Support Vector Machine were discussed In this paper Lexicon based approach was discussed as well.

- **Sentimental Analysis of Twitter Data with respect to General Elections in India, ScieneDirect,Volume-170, 2020 .**
  . In this paper, Tweets were collected from the period of Jan 2019 to March 2019. Using that tweets, sentiment analysis was performed to gain the opinion polarity of the folks concerning general elections held in India

- **A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis ,ScienceDirect , Volume - 110 , 2018.**
  This paper experimentally compared 16 commonly used pre-processing techniques on two Twitter datasets for Sentiment Analysis, employing four popular machine learning algorithms, namely, Linear SVC, Bernoulli Naïve Bayes, Logistic Regression, and Convolutional Neural Networks. They evaluate the pre-processing techniques on their resulting classification accuracy and number of features they produce. It was found that techniques like lemmatization, removing numbers, and replacing contractions improve accuracy, while others like removing punctuation do no.

1.6 Brief Methodology of the project

The figure 1.2 is explained in detail below :

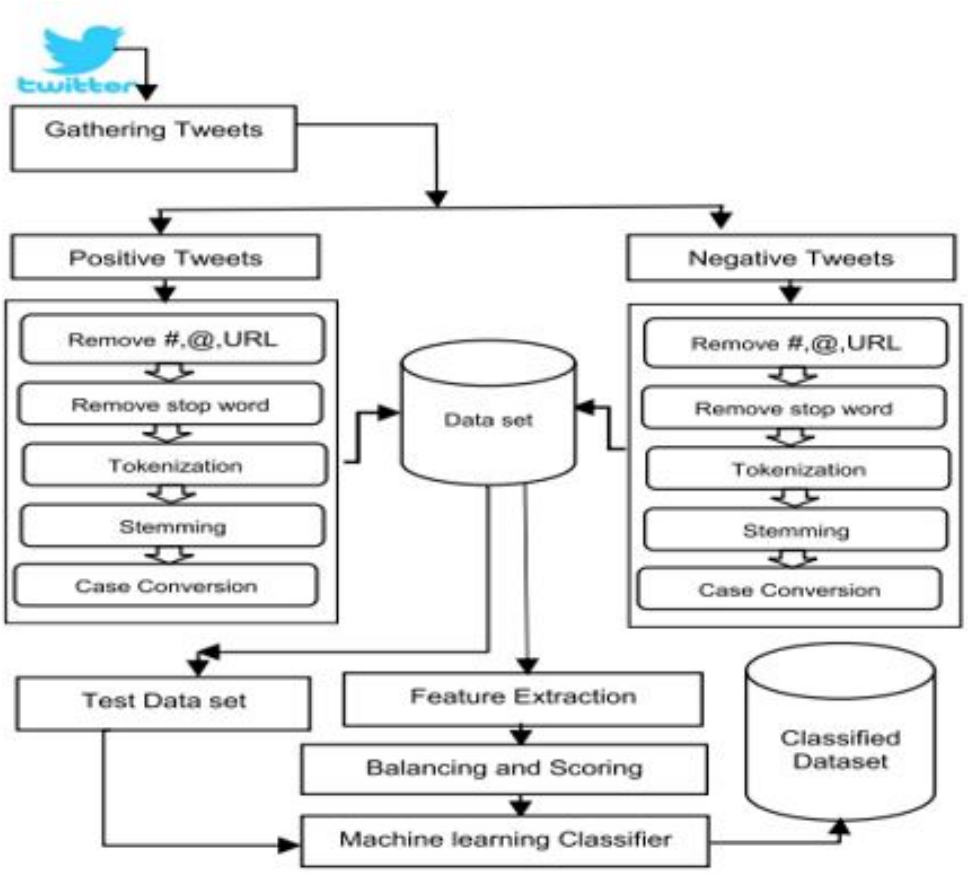

Figure 1.2: Methodology

- **Gathering Tweets :**

  The process of gathering data depends on the project and it includes collection of various datasets that are required for the project.Here , We will be creating our own dataset by parsing through the tweets in TWitter using the Twitter API functions called the TWeepy and Textblob .

- **Division into positive and negative tweets :**

  Using Tweepy and Textblob the parsed through tweets are formed as a dataset and they are classified as positive, negative or neutral tweet .Which combined together forms a dataset .

- **Data pre-processing :**

  Before entering into the project the datsets should be verified whether there are any missing values and so on.Some of the pre-processing techniques performed are checking the quality of the data and ensuring that there are no null or any missing values and next Tokenization stemming,countvectorzing and td-idf will be imple-

mented .This pre-processing has to be done to both the negative tweets and the positive tweets .

- **Training and testing :**

  Once the datasets are ready for the algorithm then the implementation of the model starts. Here datasets are divided into two training dataset,testing dataset .

- **ML algorithms :**

  The data is then passed through various ML algorithms i.e:Logistic regression,MuliNomialNB,Rand Tree Classifier etc. Then the algorithm are compared based on their accuracy and f1-scores and confusion matrix .The model is trained using trained dataset and then performed testing using test dataset.

1.7 Assumptions made / Constraints of the project

The one and only assumption that we are making is that we are assuming our parsed tweets from twitter to be self balanced that is we are assuming it to be equally positive and equally negatve . Otherwise inclination towards positive only or negative only would lead to inaccuracies such as overfitting and underfitting .

1.8 Organization of the report

This report is organized as follows :-

- Chapter 2 discusses the the requirement gathering that is the pre-requisites and the software required for the project and its analysis.
- Chapter 3 discusses the Theory and concepts involved in algorithms and methodology .
- Chapter 4 discusses Design and Implementation of the Twitter Sentiment Analysis.
- Chapter 5 discusses Results and Discussion .
- Chapter 6 discusses COnclusion and Future Scope.

.

Chapter 2

Requirement Gathering and Analysis

CHAPTER 2

REQUIREMENT GATHERING AND ANALYSIS

Gathering requirements for product development and then analyzing them is one of major steps to consider. It is important to gather the information related to each and every field that is present

2.1 Pre-Requisites

- PYTHON
- ML

.

2.2 Software Requirement

- Python and its associated libraries - Natural Language Processing (NLP), SCIKIT-LEARN
- Jupyter notebook
- Datasets from Kaggle
- Twitter API

2.3 Libraries Used

- **Pandas**

  Pandas is a Software Library in Computer Programming. It is written for the Python Programming Language to deal with data analysis and manipulation. Pandas help us to organize data and manipulate the data by putting it in a tabular form usually known as the data frame.

- **NLTK**

  The Natural Language Toolkit(NLTK) is a platform built mainly for Python programs which work with human language data for applying in statistical natural language processing (NLP). It contains many text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning etc .

- **SEABORN**

  Seaborn is a data visualization library in which is based on matplotlib. This library provides a an interface for drawing attractive and informative statistical graphs which inturn helps us understand data in a much better way.

- **SCI-KIT LEARN**

  Scikit-learn is a machine learning library for Python. It has various algorithms like random forests ,support vector machine, and k-neighbours, and it also supports libraries like NumPy and SciPy which helps in efficient predictive data analysis.

- **TWEEPY**

  Tweepy is the python client for the official Twitter API . Tweepy is an open source Python package that gives a very convenient way to access the Twitter API with Python. Tweepy includes a set of classes and methods that represent Twitter's models and API endpoints, and it transparently handles various implementation details, such as:

  - Data encoding and decoding
  - HTTP requests
  - Results pagination
  - OAuth authentication
  - Rate limits
  - Streams

  If you weren't using Tweepy, then you would have to deal with low-level details having to do with HTTP requests, data serialization, authentication, and rate limits.

- **TEXTBLOB**

  Textblob is the python library for processing textual data.It helps in diving into common NLP tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

Chapter 3

Theory and Concept of Methodology and Algorithms

CHAPTER 3

THEORY AND CONCEPT OF METHODOLOGY AND ALGORITHMS

This chapter discusses in depth about the theory and concept of the various pre-processing methods, algorithms , and methodology that are being used in the project .

3.1 Pre-Analysis of the model

The algorithms used in the model should have a certain kind of data available for application. Hence after the data collection , the data should be pre-processed in a manner that it is compatible with the algorithm that is being used.

1. **Data Collection**

   Choosing the right data-set helps this machine learning model to train and learn fast,it also gives us the benefit of predicting the correct output of the model. Dividing the train and test input data efficiently also plays an important role in getting good efficiency of the machine learning model . In this project a pre-existing dataset such as the one's from Kaggle is not being used instead we are creating a dataset by ourseleves automatically by parsing through the tweet from Twitter using Tweepy and Textblob . Here , For Data Collection using Tweepy and Textblob the Twitter API keys are needed :

   - Consumer Key
   - Consumer Secret Key
   - Access Token Key
   - Access Token Secret Key

   The data collected using the tweepy and textblob has to be looped through as the maximum amount that can be collected using these is only 200. Hence now we have a dataset of all kind of tweets from a particular tweetarati's account .

2. **Labelling the Data**

   Creating a labeled data provides an ease of processing our input data To do that

part, all the training and testing data is joined separately.This helps in clearly visualising the frequency of the words and also it's importance in the data. In this project the labelling is done by making use of the polarity and subjectivity that is obtained from the textblob.sentiment function . This Polarity helps us in determining whether the tweet is of positive emotion , negative or neutral emotion .

3. **Sentiment Analysis of data**

   After labelling of the data , the sentiment of a particular tweet has to be found out and this can be done using the textblob.sentiment.polarity() function . The polarity of all the tweets was obtained in the previous labelling of dataset step . Hence now a threshold has to be set to know the emotion of the tweet . This threshold is a predefined one . So if the textblob.sentiment.polarity() is more than 0 then the tweet is said to be of positive emotion if its equal to 0 then it is of neutral sentiment and if its less than 0 then it is of negative semtiment . Later the positive tweets and the negative tweets are displayed and visualized before going to the pre-processing steps to be able to pass through the algorithms

4. **Pre-Processing Of Data**

   Before giving the data to the model, pre-processing of the data is needed to be done so that it removes the redundant and unwanted data from datasets. There are many popular pre-processing techniques.

   - **Tokenization**

     Tokenization is a method of dividing a large text into smaller parts which are called as tokens. NLTK library provides 2 ways of tokenizing the data

     (a) Word Tokenizing

     (b) Sentence Tokenization

     Here sentence tokenizing is used for further preprocessing methods such as lemmetization.

   - **Stemming**

     Stemming is the process of producing morphological varaints of a root/base word or reducing the dreived word to its base word .The algorithm used for

stemming are called as stemmer's. stemming reduces the words like "change", "changing","changes","changed" to the root word "change".Stemming is used in information retrival systems like search engines and used to determine domain vocabularies in domain analysis.

- **Lemmatization**

  Lemmetization is a process of removing unwanted endings ,in other words it can be said that reducing the given word to it's base dictionary word. So do the lemmetization to the data that is tokenized from the above pre-processing techniques. filtration to every word in a sentence is done by using lemmetization and put back at it's previous position . This would reduce our time in training the model.

- **Stop Word Removal**

  It's very important to train the model with useful words ,hence it is good to go for removing the stopwords from the data. First,creating a set of all stopwords like in,of,the,on etc and removal of all the stop words from the data so that the data will get lighter for the model to process.

- **Removing Punctuations**

  Machines don't understand the human language ,so There is no any useful need to put any punctuation inside the data . So use REGEX (which stands for regular expression) to remove all of the punctuation s like " ","!",","," etc. To do this import the string library from the python and create a table of all of the punctuation symbols and apply the removal of these on our data

Representation of the data using graphically gives us a good understanding of the entire data based on the frequency of the words.For representing this, Word-cloud library in python is used

3.2 Machine Learning Algorithms

3.2.1 Logistic Regression

Logistic Regression is a supervised learning technique and is the most popular machine learning algorithm . If a given set of independent variables are given then it can detect categorical dependent variables which are discrete .It can be either true or false,0 or

1,Yes or No and it gives the probabilistic values which lie between 0 and 1. Logistic Regression algorithms can be used in situations where there is need to predict if the cells are cancerous or not, a mouse is obese or not based on its weight, etc.Logistic Regression has the ability to provide probabilities and classify new data using continuous and discrete datasets. It can determine the most effective variables which are used for classification given the different types of data hence it can be used to classify the observations.

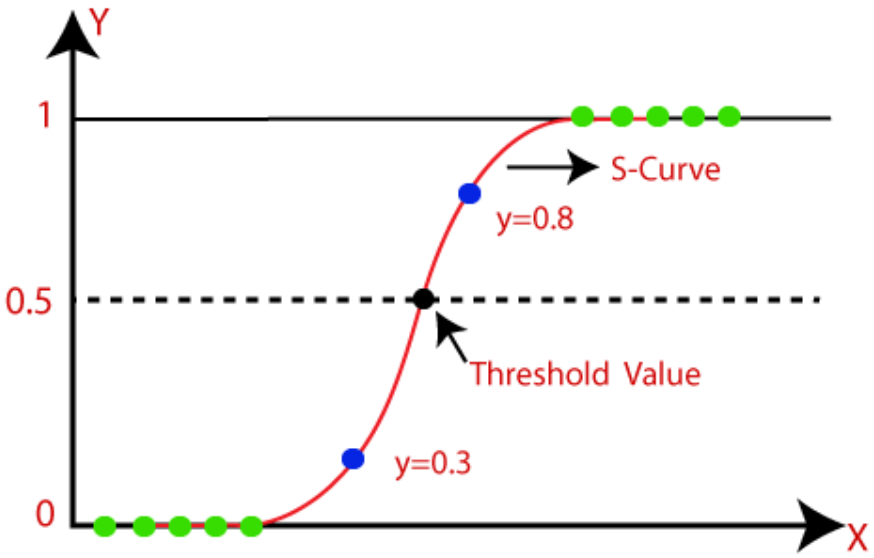

Figure 3.1: Logistic Regression curve

Logistic Regression Equation: The Logistic regression equation is obtained from the Linear Regression equation. The m steps to get the Logistic Regression equations are :

$$y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + ... + b_n \cdot x_n \tag{3.1}$$

In Logistic Regression y should be between 0 and 1 only, so let's divide the above equation by (1-y):

$$log_{10}\frac{y}{y-1} = \begin{cases} 0 & \text{if y=0} \\ \infty & \text{if y=1} \end{cases} \tag{3.2}$$

But there is a need to know the range between -[infinity] to +[infinity], then take

logarithm of the equation it will become:

$$\log_{10}(\frac{y}{1-y}) = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + ... + b_n \cdot x_n \tag{3.3}$$

3.2.2 Decision Tree Classifier

Decision Tree Classifier is a supervised learning algorithm.As the name suggests it is similar to a tree structure so it is fairly simple to understand.The structure of this classifiers starts with a root node which represents the whole data set and it is then split into different branches,which represent the classified data based on an assumed parameter or randomly.The nodes at the end are called leaf nodes.This process continues until the question asked is answered.Here, the question asked is the classification problem.Every branch leads to a possible answer to the question.Then best possible answer is choose from all possibilities using a cost function.

There are two types of cost function based on which data can be spilt.

- Gini Index
- Information gain

Gini Index is a cost function used to measure the purity of the data set.The tree with low gini index must be used to get final output.One of the disadvantages is it creates only binary splits.Gini index can be calculated using the following equation:

$$GiniIndex = 1 - \sum_j P_j \tag{3.4}$$

Information gain is the measure of entropy in the data after classification.This gives an idea about how close the current classification is to desired output.Data is spilt by maximising the gain value.It can be calculated using the equation

$$Gain = Entropy(S) - [(WeightedAvg) * Entropy(eachfeature)] Entropy(S) = \sum_i -p_i * \log_2 p_i \tag{3.5}$$

Though Decision trees are simple and easy to use,they have some disadvantages which might contradict the purpose of algorithm.Overfitting is one such problem.overfitting

causes the algorithm to correctly predict the classification only for the training data.This causes it to give wrong prediction for test data.To overcome these problem many complex algorithms are built on decision trees which are discussed next.

3.2.3 Random Forest Classifier

Random forest is a multiple decision tree and hence merges them together to get a more accurate and stable prediction. It is a supervised learning algorithm. The "forest" it builds, is an collection of decision trees, which is trained using the "bagging" method. The bagging method combines all the learning models and increases the accuracy. The advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Below you can see how a random forest would look like with two trees:

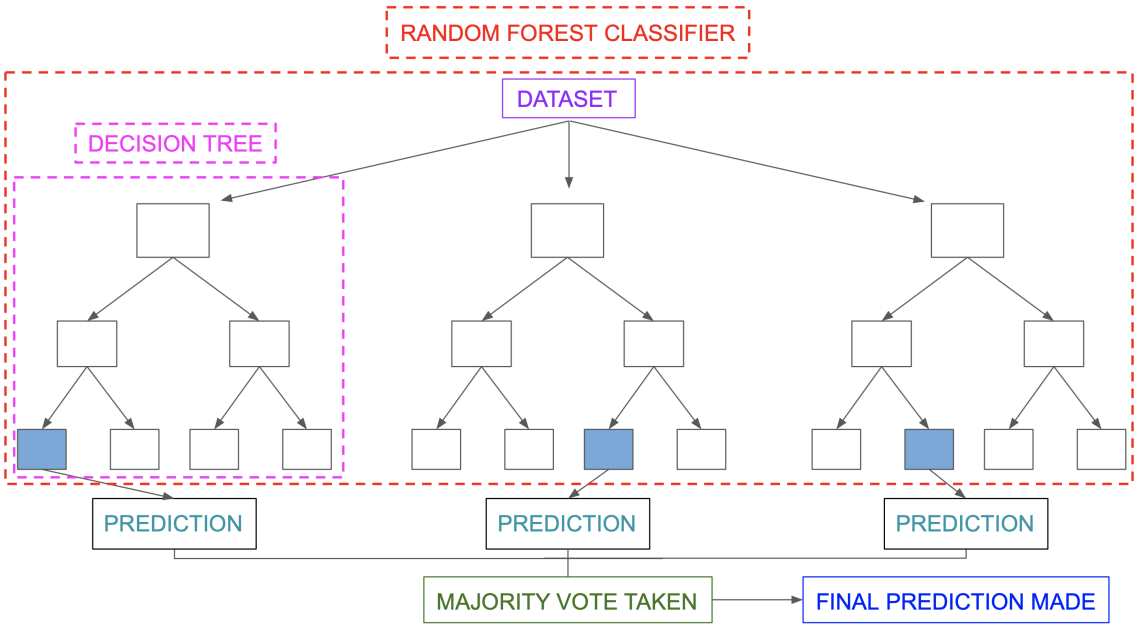

Figure 3.2: Random Tree Classifier

As the tree grows additonal randomness is added.This algorithm searches for the best feature among all the random subset of features in the data hence this results in a wide diversity which in turn results in a better model with better performance and accuracy . Therefore, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node.These tress can even be made more random by using random thresholds.

3.2.4 Xgb classifier

Xgb classifier is a boosting algorithm which is getting popular in the competitive machine learning field.The algorithm uses gradient boosted decision trees for speed and better performance.The main reasons for its better performance are due to its ability to deal with regularization,it has an inbuilt cross validation function.It can deal with missing values which is a very big issue in any dataset,algorithm finds the trends in these missing values and deal with them.It also supports objective functions,these are the functions used to measure the perforamance of the model.

The core of the algorithm is parallelizable which makes it easier to handle heavy data sets and iterations.This gives the power to harness complete performance of Graphics Processing Unit.It also allows continuous training which helps to boost already fitted model on new data.There are a lot hyper performance to consider such as the depth of trees, number of trees to be used,learning rate.This just shows though all these algorithms are automated there is a significant involvement of human mind in the performance.

3.2.5 MultiNomial Naive Bayes classifier

Naive Bayes is a supervised algorithm using bayes theorem for predictive model.It is called Naive due to its assumption that all feartures are independent of each other.It is called bayes due to its usage of bayes theorem. Bayes theorem is used to determine the probability of already know hypothesis,it depends on conditional probability

$$p(A|B) = \frac{p(B|A) * p(A)}{p(B)} \tag{3.6}$$

It works well with multi class prediction.This is go to algorithm for text prediction.There are differnt types of naive bayes model,gaussian version which assumes normal distribution in data,mutinomial for data with multinomially distributed data,bernouli for multi class distribution.similar to all other models, the process starts with data processing,then data fitting to the model,training the model and start working the results.These models are easier to build but performs better than any other model in classification problems.This classifier is used in many critical fields such as medical and sentiment analysis.

Class probabilities,probabilities of each feature class in data and conditional probabilities,probability of each input data is created after model is trained. These distributions

are generally useful than in a model.It is easier to use new data with old data to update parameters and work with changing probability distributions.one more problem arises during all this,multiplying small values generated by finding out probabilities of each input becomes numerically unstable.this can be overcome by using natural logarithm,it creates large negative which makes the values near to zero when added.

Apart from these,the main disadvantage of Naive bayes is that it assumes the data is independent.but surprisingly it works well with non independent data as well.It cannot be reasoned with since it is naive.

This Chapter discussed in depth about the various step involved in the making of twitter sentiment analysis that is this chapter discussed about the ways involved for dataset collection , for pre-processing data and the algorithms used to determine the accuracy of the model . The next Chapter would discuss about the implementation of the same .

Chapter 4

Design and Implementation of Twitter sentiment analysis

CHAPTER 4

DESIGN AND IMPLEMENTATION OF TWITTER SENTIMENT ANALYSIS

This chapter discusses the design and Implementation of the twitter sentiment analysis that is the code used to find the sentiments the implementation of various algorithms and their comparison using the accuracy .

4.1 Implementation

This part discusses the implementation of the twitter sentiment analysis in detail step by step.

4.1.1 Importing Libraries

```
# Import the libraries
import tweepy
from textblob import TextBlob
from wordcloud import WordCloud
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
```

Figure 4.1: Importing Libraries

The above figure 4.1 shows the code required to import the necessary libraries to implement the project.

4.1.2 Twitter API keys

```
consumerKey = 'eIm3uZZk3u0iqxEdKj9Lr3Jf6'
consumerSecret = 'guSP4VrYgwVzD0QiaWAMtJoESVD2cnC3GEsTP56ZXWpuXYN5I7'
accessToken = '1391239076708188160-GJ2uRRUuMeWh4rR8JhoQrhoIPl3MOW'
accessTokenSecret = 'Y1fRqLoCm8Pjx12x4Uz3ojFmNtl8ADi93KHqOaS5jPGRO'
```

```
# Create the authentication object
authenticate = tweepy.OAuthHandler(consumerKey, consumerSecret)

# Set the access token and access token secret
authenticate.set_access_token(accessToken, accessTokenSecret)

# Creating the API object while passing in auth information
api = tweepy.API(authenticate, wait_on_rate_limit = True)
```

Figure 4.2: Twitter API keys

The figure 4.2 shows the twitter API keys are used in order to extract the tweets using the tweepy and the textblob functions and also shows the authentication that is done using the APIs.

4.1.3 Extraction of tweets

```
# Extract 100 tweets from the twitter user\
all_tweets=[]
while True :
  posts = api.user_timeline(screen_name="BillGates",count = 200,lang ="en", tweet_mode="extended")
  if(len(all_tweets)==1000):
    break
  all_tweets.extend(posts)
  print('N of tweets downloaded till now {}'.format(len(all_tweets)))
```

Figure 4.3: Extraction of tweets

The figure 4.3 shows the code that is used to extract the tweets from a twitter user . The twitter user that we used in our project is Bill Gates.

```
userID = "BillGates"
from pandas import DataFrame
outtweets = [[tweet.id_str,
              tweet.created_at,
              tweet.favorite_count,
              tweet.retweet_count,
              tweet.full_text.encode("utf-8").decode("utf-8")]
             for idx,tweet in enumerate(all_tweets)]
df = DataFrame(outtweets,columns=["id","created_at","favorite_count","retweet_count", "text"])
df.to_csv('tweets.csv' ,index=False)
df
```

Figure 4.4: Logging data to csv

The figure 4.4 shows how to send and save the obtained tweets or data from the twitter user to a Comma Seperated Values (CSV) file . Also called as logging the data .

	id	created_at	favorite_count	retweet_count	text
0	1400126638067576833	2021-06-02 16:26:18	1137	176	This partnership between @Breakthrough Energy ...
1	1399863361383276545	2021-06-01 23:00:08	1736	215	The pandemic has exacerbated existing racial h...
2	1398800221132312579	2021-05-30 00:35:35	6019	1006	Avoiding a climate disaster is possible if gov...
3	1398673403003822081	2021-05-29 16:11:40	0	154	RT @GlobalFund: Stronger collaboration can hel...
4	1390452813545697282	2021-05-06 23:45:58	0	544	RT @gatesfoundation: As our CEO @MSuzman says,...
...	...	...	...	...	...
995	1288606082248486913	2020-07-29 22:43:25	4535	631	It's important that we continue to follow scie...
996	1288579704203616256	2020-07-29 20:58:36	0	897	RT @eji_org: "We all have an obligation, a mis...
997	1288575466199117831	2020-07-29 20:41:46	3246	390	"I truly believe that if there is faith and ho...
998	1288485036396666881	2020-07-29 14:42:26	5119	737	It's hard to overstate how important finding a...
999	1288188016444366848	2020-07-28 19:02:11	4218	502	Investments in digital technology that strengt...

1000 rows × 5 columns

Figure 4.5: Dataset obtained from parsing

The figure 4.5 shows the dataset that is 1000 tweets obtained by parsing through BillGates twitter account. Only 1000 tweets have been taken as the model may be overfitted if more number of tweets is extracted .

4.1.4 Cleaning the tweets

```
# Create a function to clean the tweets
def cleanTxt(text):
 text = re.sub('@[A-Za-z0-9]+', '', text) #Removing @mentions
 text = re.sub('#', '', text) # Removing '#' hash tag
 text = re.sub('RT[\s]+', '', text) # Removing RT
 text = re.sub('https?:\/\/\S+', '', text) # Removing hyperlink

 return text


# Clean the tweets
df['text'] = df['text'].apply(cleanTxt)

# Show the cleaned tweets
df
```

Figure 4.6: Code to clean the tweets

The figure 4.6 shows the code to clean the tweets i.e, removing mentions, hyperlinks, hashtags from the text part.

	id	created_at	favorite_count	retweet_count	text
0	1400126638067576833	2021-06-02 16:26:18	1137	176	This partnership between Energy and the _Comm...
1	1399863361383276545	2021-06-01 23:00:08	1736	215	The pandemic has exacerbated existing racial h...
2	1398800221132312579	2021-05-30 00:35:35	6019	1006	Avoiding a climate disaster is possible if gov...
3	1398673403003822081	2021-05-29 16:11:40	0	154	: Stronger collaboration can help countries re...
4	1390452813545697282	2021-05-06 23:45:58	0	544	: As our CEO says, no barriers should stand i...
...	...	...	...	...	...
995	1288606082248486913	2020-07-29 22:43:25	4535	631	It's important that we continue to follow scie...
996	1288579704203616256	2020-07-29 20:58:36	0	897	_org: "We all have an obligation, a mission, a...
997	1288575466199117831	2020-07-29 20:41:46	3246	390	"I truly believe that if there is faith and ho...
998	1288485036396666881	2020-07-29 14:42:26	5119	737	It's hard to overstate how important finding a...
999	1288188016444366848	2020-07-28 19:02:11	4218	502	Investments in digital technology that strengt...

1000 rows × 5 columns

Figure 4.7: Cleaned tweets

The fig 4.7 shows the cleaned tweets after executing the code showed in fig 4.6

4.1.5 Getting Subjectivity and Polarity

```
def getSubjectivity(text):
   return TextBlob(text).sentiment.subjectivity

# Create a function to get the polarity
def getPolarity(text):
   return  TextBlob(text).sentiment.polarity


# Create two new columns 'Subjectivity' & 'Polarity'
df['Subjectivity'] = df['text'].apply(getSubjectivity)
df['Polarity'] = df['text'].apply(getPolarity)

# Show the new dataframe with columns 'Subjectivity' & 'Polarity'
df
```

Figure 4.8: Code to get Subjectivity and Polarity

The figure 4.8 shows the code to get subjectivity and polarity which plays significant role in evaluating sentiment of the tweet whether positive or negative.

	id	created_at	favorite_count	retweet_count	text	Subjectivity	Polarity
0	1400126638067576833	2021-06-02 16:26:18	1137	176	This partnership between Energy and the _Comm...	0.000000	0.000000
1	1399863361383276545	2021-06-01 23:00:08	1736	215	The pandemic has exacerbated existing racial h...	0.454545	0.136364
2	1398800221132312579	2021-05-30 00:35:35	6019	1006	Avoiding a climate disaster is possible if gov...	0.600000	0.000000
3	1398673403003822081	2021-05-29 16:11:40	0	154	: Stronger collaboration can help countries re...	0.000000	0.000000
4	1390452813545697282	2021-05-06 23:45:58	0	544	: As our CEO says, no barriers should stand i...	1.000000	0.500000
...	...	...	...	...	...	...	...
995	1288606082248486913	2020-07-29 22:43:25	4535	631	It's important that we continue to follow scie...	0.727273	0.268182
996	1288579704203616256	2020-07-29 20:58:36	0	897	_org: "We all have an obligation, a mission, a...	0.000000	0.000000
997	1288575466199117831	2020-07-29 20:41:46	3246	390	"I truly believe that if there is faith and ho...	0.000000	0.000000
998	1288485036396666881	2020-07-29 14:42:26	5119	737	It's hard to overstate how important finding a...	0.547222	0.036111
999	1288188016444366848	2020-07-28 19:02:11	4218	502	Investments in digital technology that strengt...	0.400000	0.000000

1000 rows × 7 columns

Figure 4.9: Subjectivity and Polarity scores

The figure 4.8 shows the evaluated subjectivity and polarity scores of each tweet using getsubjectivity and getpolarity functions showed in figure 4.9.

After getting the Polarity and Subjectivity , the Polarity is used to get the positive tweets and the negative tweets after putting in some conditional statements for the sentiment determination The Section 4.1.7 would explain about the same

4.1.6 Word-Cloud

```
# word cloud visualization
allWords = ' '.join([twts for twts in df['text']])
wordCloud = WordCloud(width=500, height=300, random_state=21, max_font_size=110).generate(allWords)


plt.imshow(wordCloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```

Figure 4.10: Word Cloud

The fig 4.10 shows the word cloud which basically stresses on more frequent words in the tweets. This is used by training models to predict accuracy.

4.1.7 Sentiment Determination

```
# Create a function to compute negative (-1), and positive (+1) analysis
def getAnalysis(score):
  if score < 0:
    return 'Negative'
  elif score == 0:
    return 'Neutral'
  else:
    return 'Positive'
df['Analysis'] = df['Polarity'].apply(getAnalysis)
# Show the dataframe
df
```

Figure 4.11: Sentiment Determination

The figure 4.11 shows the code to determine the sentiment of the tweet by using conditional statements such as if and else if with a given pre-define threshold . Thus creating a Analysis named column which tells us the sentiment of the tweet. The result of the following would be shown in Chapter 5 .

4.1.8 Data Cleaning

The following fig 4.12 shows the code for data cleaning that is removal of words which are of less than 3 letters of length which may be of no good or meetle for finding the

sentiment . Since its redundant hence it has to be removed .

```
df['clean_tweet'] = df['text'].apply(lambda x: " ".join([w for w in x.split() if len(w)>3]))
df.head()
```

	id	created_at	favorite_count	retweet_count	text	Subjectivity	Polarity	Analysis	clean_tweet
0	1400126638067576833	2021-06-02 16:26:18	1137	176	This partnership between Energy and the _Comm...	0.000000	0.000000	Neutral	This partnership between Energy _Commission wi...
1	1399863361383276545	2021-06-01 23:00:08	1736	215	The pandemic has exacerbated existing racial h...	0.454545	0.136364	Positive	pandemic exacerbated existing racial health in...
2	1398800221132312579	2021-05-30 00:35:35	6019	1006	Avoiding a climate disaster is possible if gov...	0.600000	0.000000	Neutral	Avoiding climate disaster possible governments...
3	1398673403003822081	2021-05-29 16:11:40	0	154	: Stronger collaboration can help countries re...	0.000000	0.000000	Neutral	Stronger collaboration help countries recover ...
4	1390452813545697282	2021-05-06 23:45:58	0	544	: As our CEO says, no barriers should stand i...	1.000000	0.500000	Positive	says, barriers should stand equitable access v...

Figure 4.12: Data Cleaning

4.1.9 Tokenization

The following fig 4.13 shows the tokenization part where the sentences is broken down to words for pre-processing to be compatible with the ML algorithms

```
tokenized_tweet = df['clean_tweet'].apply(lambda x: x.split())
tokenized_tweet.head()
```

```
0    [This, partnership, between, Energy, _Commissi...
1    [pandemic, exacerbated, existing, racial, heal...
2    [Avoiding, climate, disaster, possible, govern...
3    [Stronger, collaboration, help, countries, rec...
4    [says,, barriers, should, stand, equitable, ac...
Name: clean_tweet, dtype: object
```

Figure 4.13: Tokenization

4.1.10 Stemming

```
# stem the words
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()

tokenized_tweet = tokenized_tweet.apply(lambda sentence: [stemmer.stem(word) for word in sentence])
tokenized_tweet.head()
```

```
0    [thi, partnership, between, energi, _commiss, ...
1    [pandem, exacerb, exist, racial, health, inequ...
2    [avoid, climat, disast, possibl, govern, today...
3    [stronger, collabor, help, countri, recov, fro...
4    [says,, barrier, should, stand, equit, access,...
Name: clean_tweet, dtype: object
```

Figure 4.14: Code snippet for Stemming

Stemming helps in reducing a word to its word to its word stem to remove redundancy and is important in NLP . The above figure 4.14 shows the process of stemming.

4.1.11 Pre-processed tweets after tokenization annd stemming

```
for i in range(len(tokenized_tweet)):
    tokenized_tweet[i] = " ".join(tokenized_tweet[i])

df['clean_tweet'] = tokenized_tweet
df.head()
```

	id	created_at	favorite_count	retweet_count	text	Subjectivity	Polarity	Analysis	clean_tweet
0	1400126638067576833	2021-06-02 16:26:18	1137	176	This partnership between Energy and the _Comm...	0.000000	0.000000	Neutral	thi partnership between energi _commiss will c...
1	1399863361383276545	2021-06-01 23:00:08	1736	215	The pandemic has exacerbated existing racial h...	0.454545	0.136364	Positive	pandem exacerb exist racial health inequ unit ...
2	1398800221132312579	2021-05-30 00:35:35	6019	1006	Avoiding a climate disaster is possible if gov...	0.600000	0.000000	Neutral	avoid climat disast possibl govern today. zero...
3	1398673403003822081	2021-05-29 16:11:40	0	154	: Stronger collaboration can help countries re...	0.000000	0.000000	Neutral	stronger collabor help countri recov from impa...
4	1390452813545697282	2021-05-06 23:45:58	0	544	: As our CEO says, no barriers should stand i...	1.000000	0.500000	Positive	says, barrier should stand equit access vaccin...

Figure 4.15: Dataset after pre-processing

The fig 4.15 shows the cleaned-tweet column which is the data after applying pre-processing techniques like Tokenization and Stemming .

4.2 Training and Building various ML algorithms

This section explains about the Feature extraction , splitting of data to training and testing . Later training the model using various ML algorithms and then testing the model and then comparing their accuracies .

4.2.1 Feature Extraction

```
# feature extraction
from sklearn.feature_extraction.text import CountVectorizer
bow_vectorizer = CountVectorizer()
bow = bow_vectorizer.fit_transform(df['clean_tweet'])
```

Figure 4.16: Feature Extraction

4.2.2 Train and Test Data Split

The Train and Test data is split in the ratio of 80:20

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(bow, df['Analysis'], random_state=0)
print(x_train.shape,y_train.shape)
```

```
(750, 1190) (750,)
```

Figure 4.17: Train and Test data Split

.

4.2.3 Algorithms

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
logreg = LogisticRegression()
logreg.fit(x_train,y_train)
pred = logreg.predict(x_test)
#print('The accuracy of Logistic classifier training set is {:.2f}'.format(logreg.score(x_train,y_train)))
#print('The accuracy of Logistic classifier test set is {:.2f}'.format(logreg.score(x_test,y_test)))
print("accuracy: {}%".format(round(accuracy_score(y_test, pred)*100,2)))
f1_score(y_test, pred,average = 'micro')
```

Figure 4.18: Logistic Regression

```
from sklearn.naive_bayes import MultinomialNB

mb = MultinomialNB()
mb.fit(x_train,y_train)
pred = mb.predict(x_test)
#print('The accuracy of mb classifier training set is {:.2f}'.format(mb.score(x_train,y_train)))
print('The accuracy of mb classifier test set is {:.2f}'.format(mb.score(x_test,y_test)))
cm = confusion_matrix(y_test,pred)
cm
```

Figure 4.19: MultiNomial Naive Bayes

```
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
from sklearn.metrics import plot_confusion_matrix
#pipe = Pipeline([('vect', CountVectorizer()),
  #               ('tfidf', TfidfTransformer()),
   #              ('model', RandomForestClassifier(n_estimators=50, criterion="entropy"))])
random=RandomForestClassifier()
model = random.fit(x_train, y_train)
prediction = model.predict(x_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
cm = confusion_matrix(y_test, prediction)
cm
```

Figure 4.20: Random Forest Classifier

```
from sklearn.tree import DecisionTreeClassifier
# Vectorizing and applying TF-IDF
#pipe = Pipeline([('vect', CountVectorizer()),
 #                ('tfidf', TfidfTransformer()),
  #               ('model', DecisionTreeClassifier(criterion= 'entropy',
   #                                               max_depth = 20,
    #                                              splitter='best',
     #                                             random_state=42))])
decision=DecisionTreeClassifier()
# Fitting the model
model = decision.fit(x_train, y_train)
# Accuracy
prediction = model.predict(x_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
```

Figure 4.21: Decision Tree Classifier

In this Chapter we discussed about the implementation and the design involved . Next Chapter would contain the results obtained and its discussion .

Chapter 5

Results & Discussions

CHAPTER 5

RESULTS & DISCUSSIONS

This Chapter would discuss about the results obtained after training and testing the data .

5.1 Experimental Results

```
# Printing positive tweets
print('Printing positive tweets:\n')
j=1
sortedDF = df.sort_values(by=['Polarity']) #Sort the tweets
for i in range(0, sortedDF.shape[0] ):
  if( sortedDF['Analysis'][i] == 'Positive'):
    print(str(j) + ') '+ sortedDF['text'][i])
    print()
    j= j+1
```

Figure 5.1: Printing Positive tweets

```
Printing positive tweets:

1) : Promising news from the  COVAX AMC Summit as they have exceeded today's fundraising target. This means more people in lo…

2) The pandemic has exacerbated existing racial health inequities in the United States. The Health Equity Tracker is a new tool bringing visibility to disparities in U.S. medicine

3) : As our CEO  says, no barriers should stand in the way of equitable access to vaccines. We are supportive of a…

4) : The , an unprecedented global alliance to develop &amp;amp; deliver the tests, treatments &amp;amp; vaccines the 🌍 needs to fight …

5) Ambitious short-term goals like this are critical to moving closer to a net-zero future. As we rapidly scale the solutions we have, we must also invest in innovation to reach o

6) It's encouraging to see  Biden and  Kerry re-establish America's leading role on climate change. I look forward to joining leaders from around the world to talk about some of t

7) Yesterday's verdict was a step in the right direction. But one court ruling alone will not bring to an end the injustice and inequity that Black people experience daily.  I hop

8) "Time" is a poetic portrait of a family who love and support each other despite their difficult circumstances. I can't recommend it highly enough.

9) If "Time" wins the Oscar this year, it will be the first documentary directed by a Black woman to do so. Garrett Bradley's talent makes her worthy of that milestone. This is on

10) : In February, Ghana became the first African country to receive vaccines through COVAX. Meet one of the nurses spearheading the v…

11) To get to net-zero emissions globally by 2050, leaders from around the world must work together. It's encouraging to see governments, business leaders, and financial instituti

12) The amount of cement China has consumed is a staggering statistic and reminder of how much emissions have grown in low- and middle-income countries. (Minecraft concrete doesn'

13) Who installed the most concrete?

Check your answer:

14) The best way to prevent new variants from emerging is by stopping transmission of the virus altogether:

15) It's encouraging to see innovation and clean energy investments at the forefront of 's AmericanJobsPlan. Building markets for new energy technologies is good for jobs today an

16) The technological transformation we need to address climate change can create good, safe jobs and build a more equitable, prosperous economy. To make that happen, we need to t
```

Figure 5.2: Positive tweets

```
# Printing negative tweets
print('Printing negative tweets:\n')
j=1
sortedDF = df.sort_values(by=['Polarity'],ascending=False) #Sort the tweets
for i in range(0, sortedDF.shape[0] ):
  if( sortedDF['Analysis'][i] == 'Negative'):
    print(str(j) + ') '+sortedDF['text'][i])
    print()
    j=j+1
```

Figure 5.3: Printing Negative tweets

.

```
Printing negative tweets:
1) Communities of color have been hit hard by COVID-19. One of the reasons why parts of the medical system often fail Black and brown people is because it's not designed with them
2) Dr. Stephaun Wallace () has spent the last year helping make COVID-19 vaccines work for everybody. Stephaun and his colleagues at  are working to reach the people who are usual
3) It's deeply unfair that the people who contribute the least to climate change will suffer the worst from its effects:
4) : Over the past few weeks health workers in Ethiopia ET, Nigeria NG, Sudan SD and the Philippines PH were vaccinated against COVI…
5) For decades, Australian researcher Ruth Bishop led global efforts to identify and combat rotavirus. Her life is a reminder of the importance of scientific research to uncover u
6) : Black folks have questions about the COVID-19 vaccine. I sat down w/ Black healthcare workers &amp;amp; they answered my questions.…
7) Recent extreme weather events are a stark reminder that we're already seeing the effects of climate change here at home and around the world. This type of observation system wi
8) There are several ways individuals can help move us closer to a zero-carbon future. Here are a few:
9) : The Weekly Planet: Lately, Bill Gates has been thinking about what he calls the "hard stuff" of climate change. These hard…
10) : "People who think a plan is easy are wrong. People who think a plan is impossible are wrong. It's super hard and very broad,…
11) : Only 3% of Black students learn computer science in high school or beyond. Please watch and share this video. Inspire a studen…
12) COVID-19 has cost lives, sickened millions, and thrust the global economy into a devastating recession. But hope is on the horizon:
13) Here are four other ways that America can advance its leadership on climate change this year and put the world on a path to zero emissions by 2050:
14) The President's commitment to reengage with the world gives me hope that the recovery will reach everyone, including communities of color in the U.S. and people in poor countr
15) We need to revolutionize the world's physical economy—and that will take, among other things, a dramatic infusion of ingenuity, funding, and focus from the federal government.
16) Nearly forty million people have Alzheimer's or dementia today. We have no way to stop or even slow the disease at this point, but I'm hopeful the new Alzheimer's Disease Data
17) 30: Even with his busy schedule, Dr. Fauci took the time to sit down with me (AGAIN) and talk about what we've gotten right…
```

Figure 5.4: Negative tweets

The figure 5.2 and 5.10 shows positive tweets and negative tweets after parsing through the tweets of a twitter user and also by using the polarity using the textblob function .

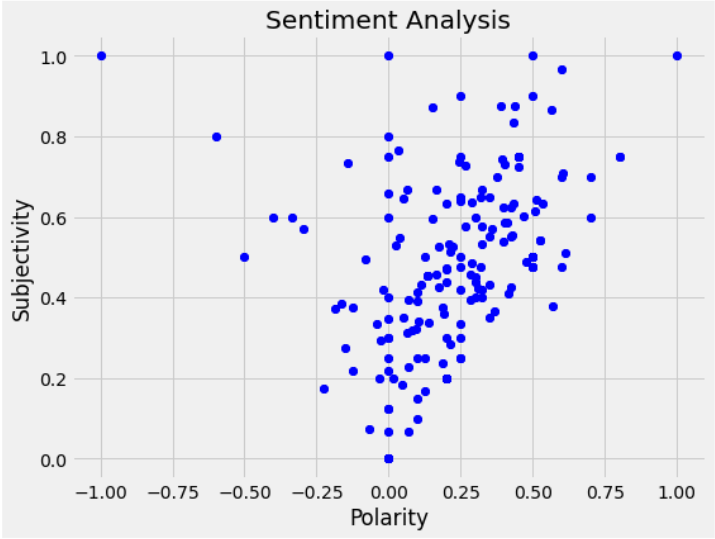

Figure 5.5: Analysis Graph

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
logreg = LogisticRegression()
logreg.fit(x_train,y_train)
pred = logreg.predict(x_test)
#print('The accuracy of Logistic classifier training set is {:.2f}'.format(logreg.score(x_train,y_train)))
#print('The accuracy of Logistic classifier test set is {:.2f}'.format(logreg.score(x_test,y_test)))
print("accuracy: {}%".format(round(accuracy_score(y_test, pred)*100,2)))
f1_score(y_test, pred,average = 'micro')
```

```
accuracy: 97.5%
```

Figure 5.6: Accuracy of Logistic Regression

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import MultinomialNB

model2 = MultinomialNB()
model2.fit(x_train , y_train)
pred = model2.predict(x_test)
f1_score(y_test,        pred , average = 'micro')
```

```
0.9500000000000001
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,pred)
cm
```

```
array([[ 22,   4,   0],
       [  0,  36,   1],
       [  5,   0, 132]])
```

Figure 5.7: Accuracy of Naive Bayes Classifier

```
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
from sklearn.metrics import plot_confusion_matrix
#pipe = Pipeline([('vect', CountVectorizer()),
  #               ('tfidf', TfidfTransformer()),
   #              ('model', RandomForestClassifier(n_estimators=50, criterion="entropy"))])
random=RandomForestClassifier()
model = random.fit(x_train, y_train)
prediction = model.predict(x_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
cm = confusion_matrix(y_test, prediction)
cm
```

```
accuracy: 98.0%
array([[ 22,   0,   4],
       [  0,  37,   0],
       [  0,   0, 137]])
```

Figure 5.8: Accuracy of Random Forest Classifier

```
from sklearn.tree import DecisionTreeClassifier
# Vectorizing and applying TF-IDF
#pipe = Pipeline([('vect', CountVectorizer()),
 #                ('tfidf', TfidfTransformer()),
  #               ('model', DecisionTreeClassifier(criterion= 'entropy',
   #                                               max_depth = 20,
    #                                              splitter='best',
     #                                             random_state=42))])
decision=DecisionTreeClassifier()
# Fitting the model
model = decision.fit(x_train, y_train)
# Accuracy
prediction = model.predict(x_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
```

```
accuracy: 98.0%
```

Figure 5.9: Accuracy of Decision Tree Classifier

```
from xgboost import XGBClassifier
decision=XGBClassifier()
# Fitting the model
model = decision.fit(x_train, y_train)
# Accuracy
prediction = model.predict(x_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
```

```
accuracy: 91.5%
```

Figure 5.10: Accuracy of XGB classifier

5.2 Performance Comparison of Various Models

From the simulation of our code while training and building of the model the accuracy of each model is observed for different models on the same dataset. The Table 5.1 shows the same.

Table 5.1: Accuracy comparison of 8 different models

Name of the model	Accuracy in Percentage
Logistic Regression	97.5
Decision Tree Classifier	98
Random Forest Classifier	98
XGB Classifier	91.5
Multinomial Naive Bayes Classifier	95.01

5.3 Inference

From the table 5.1 it is concluded that Random Forest classifier and Decision tree Classifier have higher accuracy than any of the remaining ML(machine learning) models.

Chapter 6

Conclusion and Future Scope

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

Discussed the creation of dataset by collecting the tweets by parsing them using the Twitter APIs. Then found the sentiment of the tweet as either Positive or Negative . Discussed the methodology of pre-processing the text that should be fed as input for the various algorithms . to a model. Further idea is to feed these lighter and efficient data to models like Naive based,random forest,logistic .The classification accuracy of algorithms were compared and the sentiment was found out . Hence the objectives of our project was met .

6.2 Future Scope

- This model can be pipelined to a flask app which can help user enter there text and find the sentiment of the sentence entered.
- This model can be used to put in future application where the application takes in the tweets directly from the twitter and analyzes the sentiment of the tweet which can be helpful for marketting and stuff.
- Can be used to find the sentiment of the real time data when someone is entering a data to help them paraphrase there data to a better sentiment.

.

6.3 Learning Outcomes of the Project

- Web Scraping or Web parsing .
- Working of Twitter APIs.
- Data analysis using python.
- Pre-processing methods and techniques for data cleaning.
- Working of different Machine Learning Models.
- Features extraction from the texts using different methods in Machine learning.

BIBLIOGRAPHY

[1] S. E. Saad and J. Yang *Twitter Sentiment Analysis Based on Ordinal Regression,* IEEE Systems Journal, vol. 7, pp. 163677-163685, 2019, doi: 10.1109/ACCESS.2019.2952127

[2] Z. Jianqiang, G. Xiaolin and Z. Xuejun, *Deep Convolution Neural Networks for Twitter Sentiment Analysis,* in IEEE Access, vol. 6, pp. 23253-23260, 2018, doi: 10.1109/ACCESS.2017.2776930.

[3] H. Rehioui and A. Idrissi, *New Clustering Algorithms for Twitter Sentiment Analysis,* in IEEE Systems Journal, vol. 14, no. 1, pp. 530-537, March 2020, doi: 10.1109/JSYST.2019.2912759.

[4] Z. Jianqiang and G. Xiaolin, *Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis,* in IEEE Access, vol. 5, pp. 2870-2879, 2017, doi: 10.1109/ACCESS.2017.2672677

[5] L. Mandloi and R. Patel, *Twitter Sentiments Analysis Using Machine Learninig Methods,* 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-5, doi: 10.1109/INCET49848.2020.

[6] Sentiment Analysis of Twitter Data: *A Survey of Techniques - International Journal of Computer Applications* (0975 – 8887) Volume 139 – No.11, April 2016

[7] *Twitter sentiment analysis: A case study in the automotive industry* IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT),2016

[8] Ankita Sharmaa, Udayan Ghosh *Sentimental Analysis of Twitter Data with respect to General Elections in India* International Conference on Smart Sustainable Intelligent Computing and Applications under ICITETM2020 Volume 170, April 2020

[9] SymeonSymeonidis DimitriosEffrosynidis AviArampatzis

A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis Science Direct, Expert Systems Applications Volume 110 – No.11, Nov-2018