SENTIMENT ANALYSIS

OPERATING SYSTEM REPORT

NAINA KHAN	2K20/CO/284
MANSI VERMA	2K20/CO/260
MAYANK SEHRAWAT	2K20/CO/263

TABLE OF CONTENTS

CANDIDATES DECLARATION

We [Naina Khan(2K20/CO/284), Mayank Sehrawat (2K20/CO.263) and Mansi Verma (2K20/CO/260)] hereby certify that the work, which is presented in the project entitled "SENTIMENT ANALYSIS" in fulfilment of the requirement for Mid Term Evaluation submitted to the Department of Computer Engineering, Delhi Technological University, Delhi is an authentic record of our own, carried out under supervision of MR. AMAN KUMAR PANDEY. The work presented in this report has not been submitted and not under consideration for the award for any other Course/Degree of this or any other Institute/University.

ACKNOWLEDGEMENT

In performing our major project, we had to take the help and guideline of some respected persons, who deserve our most tremendous gratitude. The completion of this project gives us much pleasure. We would like to show our gratitude to MR. AMAN KUMAR PANDEY, Mentor, for most of the project, giving us a good guideline for report throughout numerous consultations. We would also like to extend our deepest gratitude to all those who have directly and indirectly guided us in writing this assignment. Many people, especially our classmates, have made valuable comment suggestions on this proposal which inspired us to improve my project. We thank all the people for their help directly and indirectly to complete our project. In addition, We would like to thank the Department of Computer Engineering, Delhi Technological University, for giving us the opportunity to work on this topic.

ABSTRACT

The *INTERNET* today is full of social networking websites such as Facebook, LinkedIn, Pinterest, etc. which contain a lot of Data. For example, Twitter is a microblogging site in which users can post updates (Tweets) to Friends (Followers). It has become an immense Dataset of the so-called *SENTIMENTS*.

Twitter is a prominent Social Networking Website where the USERS can SEND and RECIEVE "*Tweets*," / *Short Messages*. Individuals can use this to Express Their Opinions or Feelings regarding a variety of Topics.

Sentiment Analysis has been performed on such Tweets by a variety of parties, including consumers and advertisers, to gain insights about products or conduct market research. Furthermore, thanks to recent advances in Machine Learning Algorithms, the Accuracy of our Sentiment Analysis Forecasts is improving.

In this project, we analyzed various papers focusing on the topic of Sentiment Analysis using Machine Learning. For instance, in one of the papers, they introduced an approach to select a new feature set based on Information Gain, Bigram and Object-Oriented extraction methods in sentiment analysis on social networking sites like Twitter.

We will use a variety of Machine Learning Techniques to conduct Sentiment Analysis on "Tweets." We attempt to categorize the Tweet's Polarity as either Positive or Negative. If a Tweet contains both Positive and Negative parts, the prevailing Sentiment should be chosen as the Final Classification.

We utilize a Dataset from Kaggle which was crawled marked Positive/Negative. The Data given comes with Emoticons, Usernames and Hashtags which are needed to be Processed and Translated into a Standard Form. It also requires extracting important aspects from the Text like Unigrams and Bigrams which is a sort of representation of the “Tweet”. We apply multiple Machine Learning methods to do Sentiment Analysis utilizing the retrieved characteristics. However, solely depending on individual models did not produce High Accuracy. Therefore, we chose the Best Few Models to build a Model.

INTRODUCTION

With the great quantity of rise in the Online Technologies, the number of people expressing their ideas and the opinion through Web are expanding. This knowledge is very helpful for everyone including Corporations, Governments and People.

With 500+ Million Tweets every day, Twitter is becoming a significant source of Information. Twitter is a Microblogging Platform, which is generally recognized for its brief communications known as Tweets. It has a restriction of 140 Characters. Twitter has a User Base of 240+ Million active users and so it is a great source of Information. The Users regularly debate their own opinions on many themes and also on current events through Tweets.

Out of the prominent Social Medias like Facebook, Twitter, Google+, and Myspace we select Twitter because of the Factors like

- Twitter comprises a large quantity of Text Postings and it increases day by day. The gathered Corpus may be arbitrarily huge.
- Twitter's viewership spans from normal people to Celebrities, Politicians, Corporate Leaders, and even country's president. Therefore, it is Feasible to gather Text Postings of Users from various social and interest groups
- Tweets are modest in length and hence less confusing and are neutral in nature.

Using social media, Models are constructed for categorizing "Tweets" [Good, Negative, and Neutral Groups]. The Models are constructed For 2 Classification Tasks:

- A 3-Way Classification of previously separated terms in a Tweets into Positive, Negative, and Neutral Classes
- Another 3 Way Classifications of the complete message into Positive, Negative and Neutral Classes.

WHAT IS SENTIMENT ANALYSIS?

Sentiment Analysis is a Technique of Extracting Sentiments of a certain Remark or Sentence. It's a Classification Approach which draws opinion from the Tweets and formulates a Sentiment and based on which, Sentiment Classification is conducted.

Sentiments are subjective to the issue of Interest. It is essential to formulate that what type of qualities will decide for the feeling it conveys.

In the programming model, Sentiment we refer to, is class of things that the person doing Sentiment Analysis wishes to locate in the Tweets. The dimension of the Sentiment Class is key component in choosing the Effectiveness of the Model.

For Example, we may have Two-Class Tweet Sentiment Classification (Positive & Negative) / Three Class Tweet Sentiment Classification (Positive, Negative & Neutral).

Sentiment Analysis Techniques may be Roughly classed in Two Types ->

Lexicon Based & Machine Learning Based.

- Lexicon Based Strategy is unsupervised since it promises to undertake analysis using Lexicons & a Score Mechanism to assess views.
- Whereas Machine Learning Strategy includes use of Feature Extraction and Training the Model using Feature Set and some Dataset.

The Fundamental procedures for doing Sentiment Analysis comprises

- Data Collection,
- Pre-Processing of Data,
- Feature Extraction,
- Choosing Baseline Features,
- Sentiment Detection &
- Performing Classification

either using Simple Computing else Machine Learning Methodologies.

TWITTER SENTIMENT ANALYSIS

The purpose when doing Sentiment Analysis on Tweets is simply to categorize the Tweets in various Sentiment Classes appropriately. In this Area of Study, several techniques have emerged, which provide strategies to Train A Model & then test it to verify its Effectiveness.

Performing Sentiment Analysis is tough on Twitter Data, as we said previously. Here we describe the Reasons behind this->

- Limited Tweet Size: With only 140 Characters, Compact Assertions are constructed, which yields Sparse Collection of Characteristics.
- Use of Slangs: These terms are distinct from English Words, and it might render an approach obsolete because of the evolutionary use of Slangs.
- Twitter Features: It permits the use of Hashtags, User Reference & URLs. These need distinct processing than ordinary words.
- User Diversity: The Users express their ideas in a variety of Methods, some using different language in between, while others utilizing repeated phrases or symbols to communicate an Emotion.

All these Challenges are necessary to be tackled in the Pre-Processing portion.

Apart from this, we Encounter Issues in Feature Extraction with fewer features & Minimizing the Dimensionality of Features.

RELATED WORKS

There were Several Research in Sentiment Analysis but virtually those focused on a part of Texts or Criticisms. A Tweet is only restricted to 140 Characters; thus, it is as different as a Criticism.

Bing Liu (2010), Tang and Colleagues (2009) offered an overview in Sentiment Analysis in which examined the strong points and the weak points of Sentiment Analysis and they gave various research approaches of Sentiment Analysis. Pang and Lee (2004, 2008) analyzed several Classifiers on Movie Reviews and presented a vision consisting of understanding and comprehension in Sentiment Analysis and Opinion Mining.

Authors also utilized Star Rating as a feature for Categorization. Go Et Al (2009) investigated on Bigram and POS. They deleted emoticons out from their Training Data for the classification and compared with Naive Bayes, MaxEnt, Support Vector Machine (SVM). They assessed that SVM surpasses others.

Barbosa and Feng (2010) highlighted that N-gram is sluggish. Thus, they researched on Microblogging Characteristics. Agarwal Et Al (2011) tackled Microblogging, POS and Lexicon Characteristics, additionally they created Tree Kernel to categorize Tweets and applied on POS and N-Gram. Akshi Kumar and Teeja Mary Sebastian (2012) approached a Dictionary Technique for evaluating the Emotion Polarity of Tweets.

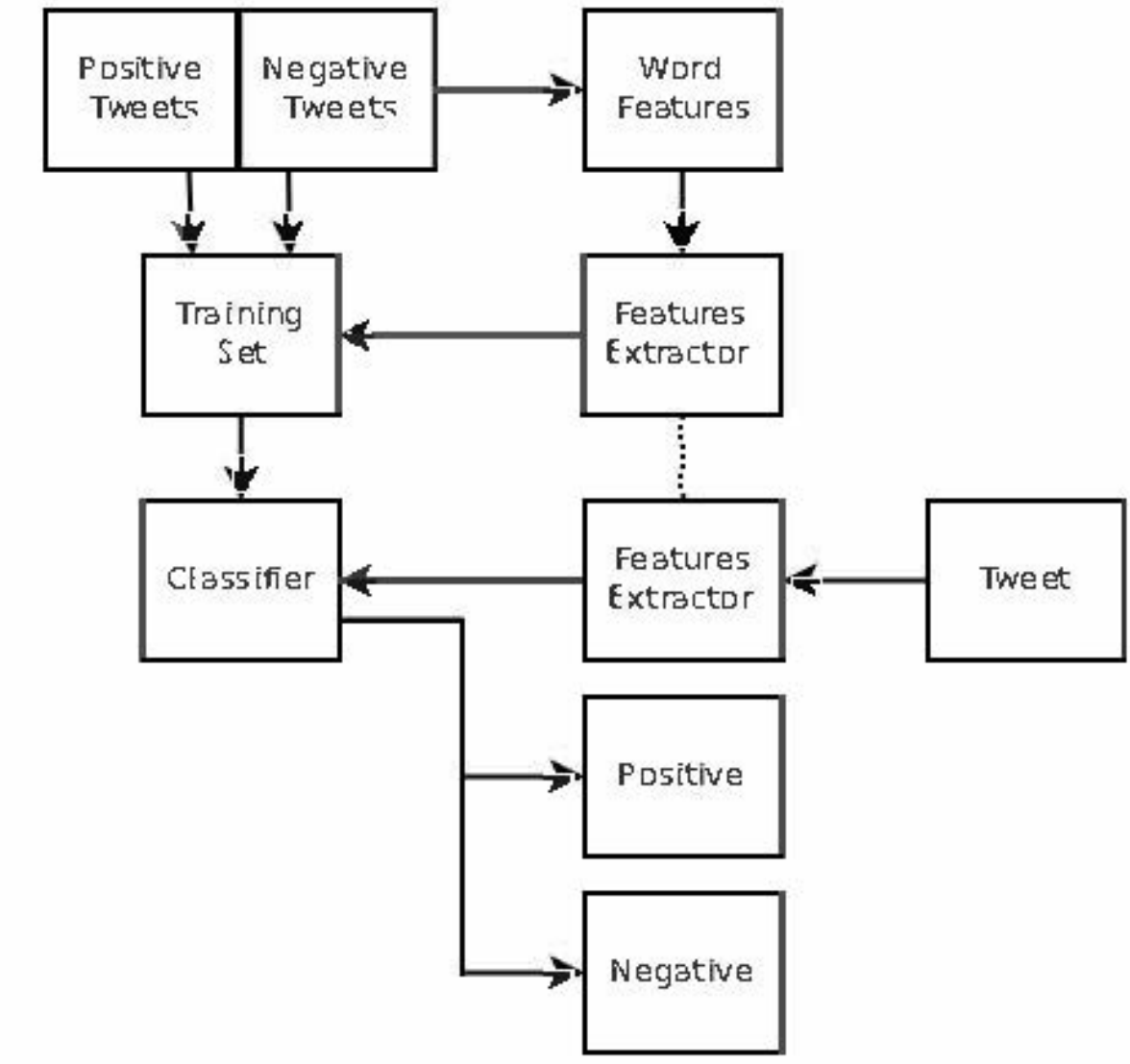
Stanford University (2013) executes a Twitter Sentiment Classifier based on Maximum Entropy and they created a Recursive Deep Model with a Sentiment Tree Bank, but they used on Movie Reviews.

APPROACH FOLLOWED

Our Technique employed Classifiers to Sort Sentiment into Positive / Negative & we used a Good Feature Extractor for Boosting Accuracy. The Classifiers which are incorporated into a Model to successfully categories include Naive Bayes (NB) and Support Vector Machine (SVM). The Most Efficient Algorithm will also be specified by observing the Accuracy of the Classification Algorithms.

METHODOLOGY

In this project, we studied a model which Classified Tweets collected from Twitter APIs and other websites into the Positive Class or the Negative Class. The models based on Three Steps as, Categorizing Tweets into Objective Tweets or Subjective Tweets, Organizing Subjective Tweets-> Positive / Negative & Finally, Summarizing Tweets into a Virtual Graph were studied in this Project, which were based on Supervised Learning such as Naive Bayes & Support Vector Machine for enhancing Effective Classification.



STEPS INVOLVED IN CLASSIFICATION OF TWEETS

APPROACH

A. IMPORTING LIBRARIES

The project is based on Data Extraction using API, Natural Language Processing (NLP) & Machine Learning. Therefore, we must include the suitable PYTHON Libraries for that.

SOME OF THE LIBRARIES USED-

- TWEEPY (For Data Extraction)
- TEXTBLOB (For NLP)
- WORDCLOUD (For Data Visualization)
- PANDAS (For Data Storage)
- NUMPY (For Array Operations)
- RE (For Text Cleaning)
- MATPLOTLIB (For Data Representation-GRAPHS)

B. GETTING TWITTER API

Tweet Collecting entails accumulating relevant Tweets on the specific subject of interest. The Tweets are gathered using Twitter's Streaming API, any other Mining Tool (For Example-> WEKA), during the necessary time period of Study. The Format of the obtained Text is transformed as per convenience (For Example JSON).

The Dataset obtained is crucial for the Effectiveness of the Model. The Separation of Dataset into Training and Testing sets is also a Decisive Element for the Efficiency of the Model. The Training Set is the Primary Factor upon which the outcomes rely.

For Accessing and Storing Tweets we need a Genuine API. Now, this can occur to the thoughts that "What is an API"? An Application Programming Interface [API], allows enterprises to provide their Application's Data and Functionality to External Third-Party Developers,

Commercial Partners, & Internal Departments inside their Company. This enables services and products to connect with one other and exploit each other's data and capabilities via a specified interface.

In summary, "An API enables a User to access 'PUBLIC DATA' directly".For utilising the Twitter API, one has to have a developer access Twitter Account. Request for the same thing can take 2–3 Hours to acquire a Permission.

Once, you're done with the set up build an App, in it, you will acquire Keys and Tokens, which will enable us access Data from Twitter.

->They function as Login Credentials.

C. AUTHENTICATING API CREDENTIALS

The Consumer Keys & Authentication Token retrieved from the App now are passed through Tweepy Library's Functions to Authenticate the credentials.

D. EXTRACTING TWEETS

Extraction of Tweets can be done easily using different Library Functions of Tweepy like user_timeline() & Cursor(). The Tweets are stored into a 1D Array.

E. PRE-PROCESSING OF TWEETS

Preparation of the Data is a Highly Crucial Stage as it determines the Efficiency of the Following Processes. It requires Syntactical Adjustment of the Tweets as Requested.

The processes involved should strive at making Data more Machine Readable in order to avoid ambiguity in Feature Extraction.

A FEW STEPS USED FOR PRE-PROCESSING OF TWEETS-

- Removal of Re-Tweets
- Converting Upper Case to Lower Case -> In case we are employing Case Sensitive Analysis, we could interpret two occurrence of similar words as distinct owing to their Sentence Case. It crucial for a Good Analysis not to supply such doubts to the Model.
- Stop Word Removal -> Stop Words that don't affect the meaning of the Tweet are Removed (For Example-> 'and'; 'or'; 'still'). Employing a WEKA Machine Learning Program for this purpose examines each of the Word from the Text against a Dictionary.
- Twitter Feature Removal (Tags/Hyperlinks/URLs) -> UserNames & URLs are not significant from the perspective of Future Processing, so their existence is meaningless. All Usernames and URLs are changed to Generic Tags/Erased.
- Stemming -> Replacing words with their origins, Eliminating distinct sorts of words with comparable meanings. This assists in Lowering the Dimensionality of the Feature Collection.
- Special Character & Digit Removal -> Digits & Special Characters don't convey any Sentiment. Sometimes they are intermingled with Words, so their removal might help connecting two terms that were otherwise regarded separate.
- Expansion of Slangs and Abbreviations
- Spelling Correction

F. FEATURE EXTRACTION

This includes identification of the object words of tweets which have Sentiment Polarity and Extraction of those into a Feature Set. Its name is Object-Oriented Feature. We also detected that several words are meaningless or insignificant. So, we must remove them.

A feature is a piece of Data that can be used as a Characteristic to help in Issue Solving (like Prediction). The Quantity and Quality of Characteristics are critical for the outcomes given by the chosen Model. Feature Extraction is the process of extracting Valuable Words from Tweets.

- Unigram Features -> One Word at a Time is considered to see whether it can be turned into a Feature.
- N-Gram Characteristics are when more than one Word is considered at the same time.
- External Lexicon -> A Collection of Terms with predetermined Positive / Negative connotation is used.

Frequency Analysis is a Technique used in to collect Features with the Highest Frequencies. They also deleted some of them due to the prevalence of terms with comparable Sentiments (For Example, Happy, Joy, Euphoric, & So On) & grouped them together. Affinity Analysis is carried out in conjunction with this, with a focus on higher order N-Grams in Tweet Feature Representation

.->Barnaghi Et Al. [3] employs Term Frequency Inverse Document Frequency (TF-IDF) to find the Weight of a given Feature in a Text and Hence Filter the Characteristics with the Highest Weight. The TF-IDF is a very Effective Method for Text Categorisation & Data Mining that is Frequently Utilised.

->Bouazizi Et Al. [4] suggests a Method in which they consider not only the Language Employed, but also the Idioms and Sentence Structure utilised in various situations. Sentiment-Based Features, Punctuation & Syntax-Based Features, Unigram-Based Features, & Pattern-Based Features were all divided into Four Categories.

->[5]'s Approach is a little different in that they propose to find Trending Ideas in an area rather than focusing on a specific topic or event. Common Features and Tweet Specific Features are the two types of Features that were Extracted. The former includes • Network Features, User Sentiment Features, & Emoticons, whereas the other one includes • Network Features, User Sentiment Features, & Emoticons. A Feature Vector is created based on each User's Post Time.

G. CLASSIFICATION

We used Twitter APIs as a Library Tool to collect tweets from internet for Sentiment Analysis and are planning to build a system based on Naive Bayes (NB) & Support Vector Machine (SVM).

The Classification can be performed using the following Algorithms:->

- **Logistic Regression :**

  It identifies Features and Optimizes the Text Categorization Process. It generates sparse Prediction Models for Text Data using a Laplace prior to avoid over-fitting. The parametric form of the Logistic Regression estimation is :->

$$P(c|f) = \frac{1}{z(f)}\ e\text{^}((\sum i\ \lambda_{i_c} F_{i,e}(f,c)))$$

  where a Normalization Function is a Binary Function that takes as input a Feature and a Class Label and returns a Vector of Weight Parameters for the Feature Set. It is activated when a specific trait is present, and the Sentiment is Predicted in a specific way.

- **Naive Bayes Classifier :**

  It is used to categorize Subjective Tweets and Objective Tweets. The Subjective Training Set is sentences labeled Subjective / Objective and we applied Unigram, Bigram & Object-Oriented Features for Training.

  Naive Bayes is a Probabilistic Classifier that is best for identifying Classes with Highly Dependent Features because of its strong conditional Independence Requirement.

  The Bayes Theorem is used to calculate Sentiment Class Adherence.

$$P(X|Y_i) = \prod_{i=1}^{m} P(x_i|y_i)$$

  y_i is a Class Label and is a Feature Vector. Defined as X = {x_1, x_2 ..... x_m}.

  The Naive Bayes Classifier is a fairly Basic Classifier that produces Decent Results but not as well as other Classifiers.

- **SVM Classifier :**

  It classifies the Subjective Tweets into the Positive Class or the Negative Class. The Sentiment Training Set is sentences labeled Positive/Negative & we applied Unigram, Bigram, Object-Oriented Features for Training.

  The system also draws a Graph after the analysis of Feelings.

  Support Vector Machines are Supervised Models that Examine Data for Classification and Regression Analysis . The concept of Decision Planes is used to define Decision Boundaries.

$$g(X) = w^T\Phi(X) + b$$

  'X' stands for Feature Vector, 'w' for Vector Weights, and 'b' for Bias Vector. Φ(X) is the Non-Linear Transformation from Input Space to a High-Dimensional Feature Space. SVMs can be used to Recognise Patterns .

- **Artificial Neural Network :**

  The Multi-Layer Perceptron is a Feed Forward Model that Maps Data onto a collection of Relevant Outputs, and it is the ANN Model used for Supervised Learning. Training Data is fed into the Input Layer, which is then processed by Hidden Intermediate Layers before being sent to the Output Layers. The number of Hidden Layers is a crucial statistic for the Model's Success.

  MLP NN works in Two Stages:

  - Feed Forward Propagation, which involves Learning Features from a Feed Forward Propagation Algorithm, &
  - Back Propagation, which involves learning features from a Back Propagation Technique.

Zimbra Et Al. suggests using Dynamic Architecture For Artificial Neural Network (DAN2), A Machine-Learned Model with enough Sensitivity to Moderate Expression in Tweets. They want to look at Brand-Related Feelings where Mild Sentences are frequently used.DAN2 differs from ordinary Neural Networks in that the number of Hidden Layers isn't determined before the Model is used. As information is provided, knowledge and learning are accumulated at each

level and passed on to the next. The Hidden Layers are formed dynamically till the desired level of performance is reached.

- **Case Base Reasoning :**

  Issues that have previously been successfully addressed are accessed, and their answers are obtained and applied again. It doesn't need an explicit domain Model; Thus, Elicitation is only a matter of Accumulating Case Histories, and the CBR System can learn new things as cases. This makes it easy to maintain big columns of Data.

- **Maximum Entropy Classifier :**

  This Classifier makes no assumptions about Feature Relationships; It always aims to Maximize a System's Entropy by calculating its conditional distribution of Class Labels [9].

$$P_\lambda(y|X) = 1/Z(X)\ e^{\sum i \lambda_i f_i(x,y)} \quad .$$

  The Feature Vector is 'X', and the Class Label is 'y'.

  The Normalisation Factor = Z(X),

  λ_i = Weight Coefficient.

- **Ensemble Classifier :**

  To create the Best Classification, this Classifier tries to combine Characteristics from all of the Basis Classifiers. [9] Employed Nave Bayes, SVM, and Maximum Entropy as its Base Classifiers. The Classifier divides the Data into Categories depending on the output of the majority of Classifiers (Voting Rule).

- **Testing and Experimentation :**

  This is the last step. The Model that is trained using the suitable classifier can now be used to 'PREDICT' the SENTIMENT of any Text / Data provided by the User. We are planning to make a Web Application which will guess the SENTIMENT using simple HTML, CSS & JavaScript.

PROCEDURE

TWITTER SENTIMENT ANALYSIS WITH PYTHON

TECHNOLOGIES REQUIRED->

- PYTHON
- Anaconda
- Google Collab
- Natural Language Processing (Python->NLTK Library)
- SCIKIT-Learn
- NumPy
- Word Cloud
- TweePy
- Pandas
- Matplotlib
- Regular Expression
- Pickle
- Keras

A) PYTHON

- Python is a High-Level Interpreted Programming Language. The Language is well-known for its Code Readability and Short Code Lines.
- To Delimit Blocks, it employs White Space Inundation.
- Python comes with a wide Standard Library that may be used for a variety of tasks, with Natural Language Processing, Machine Learning, & Data Analysis.
- Because of its Simplicity, wide variety of Features, and Dynamic Nature, it is preferred for Complicated Projects.

B). Natural Language Processing (NLTK)

- The Natural Language Toolkit (NLTK) is a Python Framework that serves as the Foundation for Text Processing and Categorization. NLTK may be used to do Tasks such as Tokenization, Tagging, Filtering, & Text Manipulation.
- Various Trainable Classifiers (For Example, The Nave Bayes Classifier) are also included in the NLTK Library.
- A Bag-of-Words Model, which is a Form of Unigram Model for Text, is created using the NLTK Package. The number of times each word appears in this Model is tallied. The Collected Data may be utilized to Train Classifier Models.
- The Sentiment of the Full Tweets is calculated by utilizing a Sentiment Lexicon to give a Subjective Score to each Word.

C) SCIKIT-LEARN

Scikit-Learn Project as Scikits.Learn-> A Google Summer Code Project. It's a Robust Library that includes a variety of Machine Learning Classification Methods as well as effective Data Mining & Analysis Tools.

The Following are some of the Functions that this Library can perform->

- CLASSIFICATION is the Process of determining to which category an item belongs to.
- REGRESSION is the Process of Predicting the Value of a continuous-valued property linked with an Object.
- CLUSTERING is the process of automatically Grouping Comparable Items into Sets.

- DIMENSION REDUCTION is the process of Reducing Amount of Random Variables that must be considered.
- MODEL SELECTION-> Comparing, Validating, & Selecting Parameters & Models.
- PREPROCESSING is the process of Extracting Features and Normalizing Data to convert it for usage with a Machine Learning Algorithm.

NumPy must be installed on the system to Function with Scikit-Learn.

D) NumPy

NumPy is the most important Python Module for Scientific Computing.

It includes a High-Performance Multidimensional Array Object as well as utilities for manipulating them. It includes, among other things, the Following->

- An N-Dimensional Array Object with a lot of Power.
- Advanced (Broadcasting) Capabilities
- Useful Linear Algebra, Fourier Transform, & Random Number Capabilities.
- Tools for Integrating C/C++ and Fortran Programs.

E) Setting Up Environment->IMPORTING LIBRARIES

Downloading & Installation Successfully, you'll need the following Components

- Python 2.6 must be Downloaded & Installed at the Specified Place.
- NumPy may be Downloaded & Installed.
- Installing the NLTK Library on your Computer.
- Installing the Scikit-learn Library on your Computer.

F) Data Collection

For Sentiment Analysis, we have Two Approaches for Gathering Data.

The First is to utilize Tweepy, A Twitter Application Programming Interface (API) Client (API).

To get Tweets from the Twitter API, Users must First Create An App using their Twitter Account.

The Following Steps are then carried out:

- Go to https://apps.twitter.com/ & Choose the 'Create New App' Option.
- Fill up the Blanks with the Information Requested.
- The Page will be automatically loaded when the App is Built.
- Go to the 'Keys & Access Tokens' Tab and Click it.
- 'Consumer Key', 'Consumer Secret', 'Access Token', 'Access Token Secret' should all be Copied.

The copied keys are then entered into the Code, resulting in Dynamic Collection of Tweets each time it is Executed.

The Alternative Option is to Collect Data in a Non-Dynamic Manner utilizing Existing Data from Websites (Such as Kaggle.com) & Store it in whichever Format we need (For Example JSON, CSV Etc.).

The First Technique is Sluggish since it collects Tweets every time the Software is started. The Latter Method may not provide the High-Quality Tweets we seek.

To remedy this, we may place the Tweet collecting Code in a distinct Module so that it doesn't execute every time the Project is started.

G) Pre-Processing Using Python

Because of the Methods supplied by the Standard Library, Pre-Processing in Python is Simple.

The Following are some of the Steps->

- All Upper-Case Letters are being converted to Lower Case.
- Removing URLs: Regular Expression (http|https|ftp)://[a-zA-Z0-9./]+ may be used to Filter URLs.

- Removing Handles (User Reference): Using the Regular Expression @(w+), Handles may be Deleted.
- Removing Hashtags: #(w+) is a Regular Expression that may be used to Remove Hashtags.
- Getting Rid of Emoticons: We may use an Emoticon Dictionary to filter out Emoticons or preserve their occurrences in a Separate File.
- Removing Characters that have been used before.

H) Feature Extraction

In Today's World, there are variety of Approaches for Extracting Characteristics. Inverse Term Frequency Frequent Documenting is a Good Strategy. The TF-IDF is a Numerical Statistic that represents the overall worth of a word in the TWEET.

Scikit-Learn includes Vectorizers that convert Texts into Feature Vectors.
We may use the Library Function TF-IDFVectorizer() to Specify Parameters for the kind of features we wish to maintain, as well as the Minimum Frequency of Features that are acceptable.

I) TRAINING THE MODEL->

The Scikit-Library contains several Machine Learning Models that are simple to Implement in Code.
For Example,

classifier_poly = svm.SVC()

May be used to Quickly generate a Support Vector Machine Instance.
To utilize Machine Learning Models, ensure NumPy is installed correctly and that the appropriate Model is imported from Scikit-Learn.
We use the same instance to Test the Model once it has been Trained, and we store the Results.
Features that are acceptable.

EXPERIMENTAL STIMULATION

PYTHON CODE

```
#DESCRIPTION: SENTIMENTAL ANALYSIS OF TWITTER DATA
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
plt.style.use('fivethirtyeight')

#TWITTER API CREDENTIALS
consumerKey = "FLj6bcrpPsic3rZ6kLQNa7mbH"
consumerSecret = "psOlDGnwnqBd6xo20mVHlGkAGvvKt7C6OcsysJphXQnajOrcD4"
accessToken = "1442491980554981384-v9ThsGb9jnogNgzCM0kgPJjxvKQPGb"
accessTokenSecret = "DwXNMU82ievaZRw9mh4wq0AnzIYZRkHOwmGlhmmxvLjGK"

#CREATING AUTHENTICATION OBJECT
authenticate = tweepy.OAuthHandler(consumerKey, consumerSecret)

#SET ACCESS TOKEN AND ACCESS TOKEN SECRET
authenticate.set_access_token(accessToken, accessTokenSecret)

#CREATING API OBJECT WHILE PASSING AUTH INFORMATION
T_API = tweepy.API(authenticate, wait_on_rate_limit=True)

#EXTRACTING 100 TWEETS FROM TWITTER USER
posts = T_API.user_timeline(screen_name="BillGates", count=100, lang="en",
tweet_mode="extended")

#PRINTING THE LAST 5 TWEETS FROM THE ACCOUNT
print("SHOWING 5 TWEETS: \n")
i=1
for tweet in posts[0:5]:
  print(str(i)+")", tweet.full_text, "\n")
  i+=1


#CREATING DATAFRAME WITH COLUMN CALLED TWEETS
DF = pd.DataFrame([tweet.full_text for tweet in posts], columns=['Tweets'])

#SHOW THE FIRST 5 ROWS OF DATA
DF.head()
```

```
SHOWING 5 TWEETS:

1) RT @TEDTalks: To prevent future pandemics, @BillGates says we must invest in 3 things *now*:

1. Disease monitoring
2. Research and develop…

2) Seven years ago, I gave a TED Talk about how the world wasn't ready for the next epidemic. A lot has changed since then: https://t.co/3oT8MJIYrO

3) We can't reach zero carbon emissions without innovation. In this new series, you'll meet some people who are on the cutting edge of clean energy: #EarthDay https://t.co/o6CISr0GD2

4) Interesting article on how Liberia's health workers use data to provide crucial insights and stop diseases from spreading: https://t.co/vKT5si3kHR

5) I hope Namzi is feeling better. Her comics are great! https://t.co/P8nKWRy4bY
```

	Tweets
0	RT @TEDTalks: To prevent future pandemics, @Bi...
1	Seven years ago, I gave a TED Talk about how t...
2	We can't reach zero carbon emissions without i...
3	Interesting article on how Liberia's health wo...
4	I hope Namzi is feeling better. Her comics are...

```python
#CLEANING THE TEXT
#CREATING A FUNCTION TO CLEAN THE TWEETS
def cleanText(text):
  text = re.sub(r'@[A-Za-z0-9]+', '', text) #REMOVES @ TAGS
  text = re.sub(r'#', '', text) #REMOVES # SYMBOL
  text = re.sub(r'RT[\s]+', '', text) #REMOVES RT(RETWEETS)
  text = re.sub(r'https?:\/\/\S+', '', text) #REMOVES HYPERLINK

  return text

#CLEANING THE TEXT
DF['Tweets'] = DF['Tweets'].apply(cleanText)

#SHOWING CLEANED TEXT
DF
```

	Tweets
0	: To prevent future pandemics, says we must i...
1	Seven years ago, I gave a TED Talk about how t...
2	We can't reach zero carbon emissions without i...
3	Interesting article on how Liberia's health wo...
4	I hope Namzi is feeling better. Her comics are...
...	...
95	I'm excited to talk with today about her work...
96	Through my foundation work I'm very lucky to h...
97	Understanding how the connections in our brain...
98	. is one of my favorite authors. I was eager t...
99	Heroes like are spreading important informati...

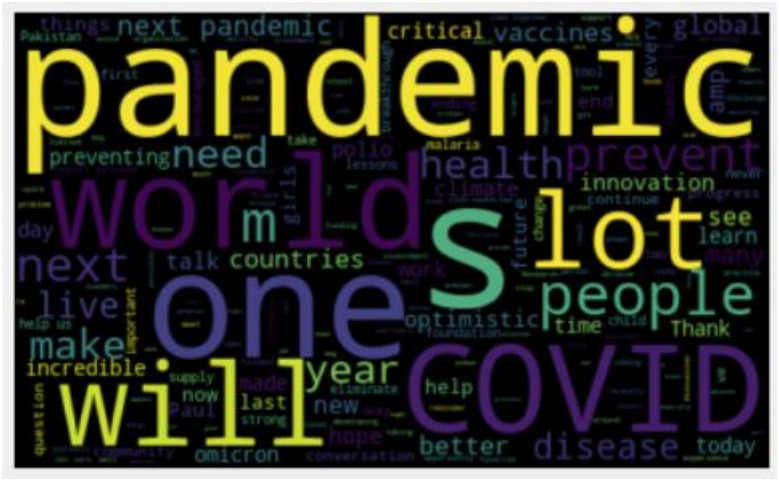
100 rows × 1 columns

```
#CREATING A FUNCTION TO GET SUBJECTIVITY
def getSubjectivity(text):
  return TextBlob(text).sentiment.subjectivity

#CREATING A FUNCTION TO GET POLARITY
def getPolarity(text):
  return TextBlob(text).sentiment.polarity

#CREATE NEW COLUMNS
DF['Subjectivity'] = DF['Tweets'].apply(getSubjectivity)
DF['Polarity'] = DF['Tweets'].apply(getPolarity)

#SHOWING DATAFRAME WITH NEW COLUMNS
DF
```

	Tweets	Subjectivity	Polarity
0	: To prevent future pandemics, says we must i...	0.125000	0.000000
1	Seven years ago, I gave a TED Talk about how t...	0.250000	0.100000
2	We can't reach zero carbon emissions without i...	0.684848	-0.032323
3	Interesting article on how Liberia's health wo...	0.750000	0.250000
4	I hope Namzi is feeling better. Her comics are...	0.625000	0.750000
...	...	...	...
95	I'm excited to talk with today about her work...	0.333333	0.125000
96	Through my foundation work I'm very lucky to h...	0.475000	0.238889
97	Understanding how the connections in our brain...	0.700000	0.666667
98	. is one of my favorite authors. I was eager t...	0.950000	0.500000
99	Heroes like are spreading important informati...	1.000000	0.366667

100 rows × 3 columns

```
#PLOTTING WORD CLOUD TO GET AN IDEA
allWords = ' '.join([twts for twts in DF['Tweets']])
wordCloud = WordCloud(width=500, height=300, random_state=21, max_font_size
=120).generate(allWords)

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



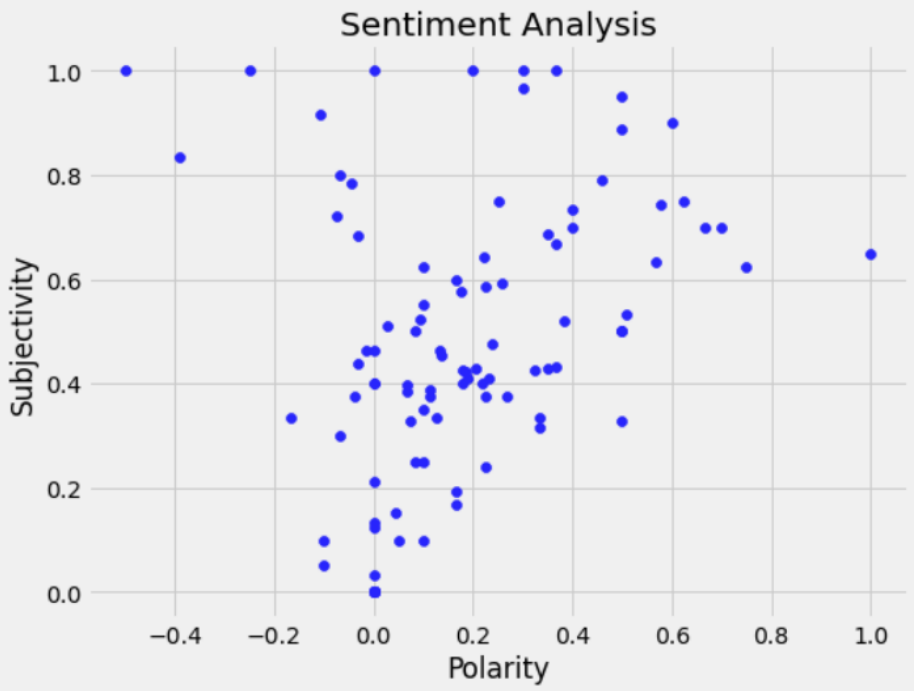
```
#FUNCTION TO COMPUTE THE NEGETIVE NEUTRAL AND POSITIVE ANALYSIS
def getAnalysis(score):
  if score>0:
    return "POSITIVE"
  elif score==0:
    return "NEUTRAL"
  else:
    return "NEGATIVE"
DF['Analysis'] = DF['Polarity'].apply(getAnalysis)

#SHOW DATAFRAME
DF
```

	Tweets	Subjectivity	Polarity	Analysis
0	: To prevent future pandemics, says we must i...	0.125000	0.000000	NEUTRAL
1	Seven years ago, I gave a TED Talk about how t...	0.250000	0.100000	POSITIVE
2	We can't reach zero carbon emissions without i...	0.684848	-0.032323	NEGATIVE
3	Interesting article on how Liberia's health wo...	0.750000	0.250000	POSITIVE
4	I hope Namzi is feeling better. Her comics are...	0.625000	0.750000	POSITIVE
...	...	...	...	...
95	I'm excited to talk with today about her work...	0.333333	0.125000	POSITIVE
96	Through my foundation work I'm very lucky to h...	0.475000	0.238889	POSITIVE
97	Understanding how the connections in our brain...	0.700000	0.666667	POSITIVE
98	. is one of my favorite authors. I was eager t...	0.950000	0.500000	POSITIVE
99	Heroes like are spreading important informati...	1.000000	0.366667	POSITIVE

100 rows × 4 columns

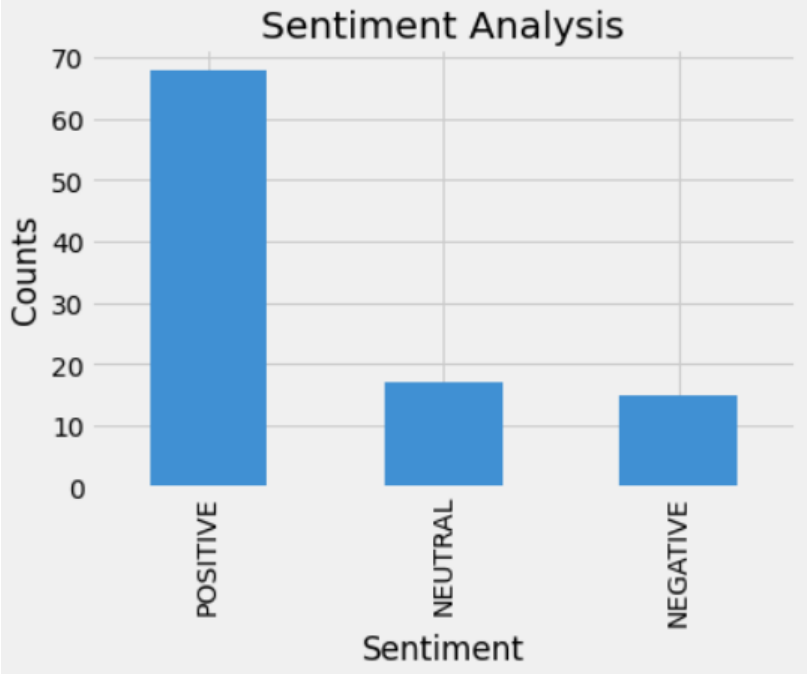
```
#PRINTING ALL POSITIVE TWEETS
j=1
sortedDF = DF.sort_values(by=['Polarity'])
for i in range(0, sortedDF.shape[0]):
  if (sortedDF['Analysis'][i] == "POSITIVE"):
    print(str(j)+')', sortedDF['Tweets'][i], "\n")
    j=j+1;
#PRINTING ALL NEGATIVE TWEETS
j=1
sortedDF = DF.sort_values(by=['Polarity'])
for i in range(0, sortedDF.shape[0]):
  if (sortedDF['Analysis'][i] == "NEGATIVE"):
    print(str(j)+')', sortedDF['Tweets'][i], sortedDF['Polarity'][i], "\n")
    j=j+1;

#PRINTING ALL NEUTRAL TWEETS
j=1
sortedDF = DF.sort_values(by=['Polarity'])
for i in range(0, sortedDF.shape[0]):
```

```
  if (sortedDF['Analysis'][i] == "NEUTRAL"):
    print(str(j)+')', sortedDF['Tweets'][i], "\n")
    j=j+1;

#PLOTTING THE POLARITY AND SUBJECTIVITY
plt.figure(figsize=(8,6))
for i in range(0, DF.shape[0]):
  plt.scatter(DF['Polarity'][i], DF['Subjectivity'][i], color='Blue')

plt.title('Sentiment Analysis')
plt.xlabel('Polarity')
plt.ylabel('Subjectivity')
plt.show()
```




```
#GET THE PERCENTAGE OF POSITIVE TWEETS
posTweets = DF[DF.Analysis == 'POSITIVE']
posTweets = posTweets['Tweets']

round((posTweets.shape[0] / DF.shape[0])*100, 1)

#GET THE PERCENTAGE OF NEGATIVE TWEETS
negTweets = DF[DF.Analysis == 'NEGATIVE']
negTweets = negTweets['Tweets']

round((negTweets.shape[0] / DF.shape[0])*100, 1)

#SHOWING PERCENTAGES ON GRAPH
DF['Analysis'].value_counts()
```

```
plt.title('Sentiment Analysis')
plt.xlabel('Sentiment')
plt.ylabel('Counts')
DF['Analysis'].value_counts().plot(kind='bar')
plt.show()
```



EXPERIMENTAL RESULTS

We Focus on Training and Verifying the Model's Performance after Completing the Pre-Processing and Feature Extraction Phases. The Dataset is Split into Two Parts-> A Training Set & A Testing Set.

The Training Set used to Train the Classifier (Machine Learning Model), whereas The Testing Set is used to Test the Classifier. Depending on the Application, the Ratio of Training & Testing Datasets may vary.

- [1] Separates the Dataset into 70% Training & 30% remainder Testing, while [3] Splits it into 10 Pieces and applies cross validation on it. This approach picks 90% of the Training Set and 10% of the Testing Set.
- [4] Separated the collection into a Training Set with 21000 Tweets and a Testing Set with 1400 Tweets (About 93% & 7%, respectively), whereas [5] used 75% Data for Training and [9] used approx. 83% for Training.
- Because [6]'s Categorization Work Topic-Based and Adaptable, Superfluous Manual Labeling is avoided, Reducing the size of the Training Set.

The Training Set of Data is used to Train the Model that will be used in the Experiment. The Trained Model is then used to categorize additional Data, allowing us to verify its Correctness.

The Suggested Work of [3] is a little different in that they use timestamp to link the Event and Emotion. It is Feasible to break a certain event into sub-events & enhance the investigation of User Feelings using this Strategy. When we choose a Huge Event and want to examine How User Emotions Change over time, we Employ this method, which is difficult but yields highly detailed findings.

The number of Classes to be Classified is entirely up to the User. Depending on the kind of Application, Binary, Ternary, or Multi-Class Classification might be used. However, it has been discovered that as the number of Classes grows, Classifier performance falls [1], [3].

Average Accuracies for Different Models

S. No.	Classifier	Accuracy
1	DAN2	86.06%
2	SVM	85.0%
3	Bayesian Logistic Regression	74.84%
4	Naïve Bayes	66.24%
5	Random Forest Classifier	87.5%
6	Neural Network	89.93%
7	Maximum Entropy	90.0%
8	Ensemble Classifier	90.0%

APPLICATIONS

- **COMMERCE**

  Businesses may utilize this Research to collect Public Opinion on their Brand & Goods. In the Company's Opinion, a Survey of the Target Population is essential for Determining Product Evaluations. As a result, Twitter may be a useful tool for Gathering & Analyzing Data to Gauge Client Happiness.

- **POLITICS**

  Politics is the subject of the Majority of Tweets on Twitter. Because Twitter is so widely used, many politicians are attempting to interact with their constituents via it. People express their Approval / Disapproval of Government Policies, Acts, Elections, Debates, & So On. As a result, Studying Data from it may aid in determining Public Opinion.

- **SPORTS EVENTS**

  Sports Events include variety of Competitions, Championships, Meetings, & Disputes. Many individuals are Die-Hard Sports Fans who use Twitter to keep up with their favorite players. These individuals routinely tweet about various Sporting Events. We may utilize the information to get a Public Perspective on a Player's Activity, A Team's Performance, & Official Decisions, among other things.

CONCLUSION

The Text & Opinion Mining Category includes Twitter Sentiment Analysis.

It focuses on Assessing the Sentiments of Tweets and Feeding the Data to A Machine Learning Model to Train & then Evaluate its Correctness, so that we may utilize the Model in the Future Based on the Findings.

Data Gathering, Text Pre-Processing, Model Training, Sentiment Categorization, Sentiment Detection, & Testing are all phases in the Process. This study issue has progressed over the previous decade, with Models attaining Efficiencies of almost 85% - 90%. However, it still lacks the Dimension of Data Variety.

It also has a lot of Application concerns due to the Slang & Abbreviated Versions of Words utilized. When the number of Classes is raised, many analyzers perform poorly. Also, the Model's Accuracy for subjects other than the one under discussion has yet to be proven.
As a result, Sentiment Analysis has a lot of room for growth in the Future.

REFERENCES

[1] David Zimbra, M. Ghiassi and Sean Lee, "Brand-Related Twitter Sentiment Analysis using Feature Engineering and the Dynamic Architecture for Artificial Neural Networks", IEEE 1530-1605, 2016.

[2] Varsha Sahayak, Vijaya Shete and Apashabi Pathan, "Sentiment Analysis on Twitter Data", (IJIRAE) ISSN: 2349-2163, January 2015.

[3] Peiman Barnaghi, John G. Breslin and Parsa Ghaffari, "Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment", 2016 IEEE Second International Conference on Big Data Computing Service and Applications.

[4] Mondher Bouazizi and Tomoaki Ohtsuki, "Sentiment Analysis: from Binary to Multi-Class Classification", IEEE ICC 2016 SAC Social Networking, ISBN 978-1-4799-6664-6.

[5] Nehal Mamgain, Ekta Mehta, Ankush Mittal and Gaurav Bhatt, "Sentiment Analysis of Top Colleges in India Using Twitter Data", (IEEE) ISBN -978-1-5090-0082-1, 2016.

[6] Halima Banu S and S Chitrakala, "Trending Topic Analysis Using Novel Sub Topic Detection Model", (IEEE) ISBN- 978-1-4673-9745-2, 2016.

[7] Shi Yuan, Junjie Wu, Lihong Wang and Qing Wang, "A Hybrid Method for Multi-class Sentiment Analysis of Micro-blogs", ISBN- 978-1-5090-2842-9, 2016.

[8] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow and Rebecca Passonneau, "Sentiment Analysis of Twitter Data" Proceedings of the Workshop on Language in Social Media (LSM 2011), 2011.

[9] Neethu M S and Rajasree R, "Sentiment Analysis in Twitter using Machine Learning Techniques", IEEE – 31661, 4th ICCCNT 2013.