

# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**JNANA SANGAMA, BELAGAVI-590 018**



## **A Project Work Phase-II Report**

**on**

***“Sentimental analysis of Twitter Data on Economic Crisis due to COVID-19”***

*Submitted in partial fulfillment of the requirements for the final year degree in  
**Bachelor of Engineering in Computer Science and Engineering**  
of Visvesvaraya Technological University, Belagavi*

Submitted by

**VINAY D                      1RN19CS181**

**VINAY H R                1RN19CS182**

**VIVEK S HEGDE        1RN19CS186**

Under the Guidance of:

**Prof. Phanikanth K V**

**Assistant Professor**

**Dept. of CSE**



**Department of Computer Science and Engineering**

**(UG Programs-CSE, ISE, ECE, EEE & EIE are Accredited by NBA upto 30-06-2025)**

**RNS Institute of Technology**

**Channasandra, Dr. Vishnuvardhan Road, Bengaluru-560 098**

**2022-23**

# **RNS Institute of Technology**

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-98

## **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

(UG Programs-CSE, ISE, ECE, EEE EIE are Accredited by NBA upto 30-06-2025)



### **CERTIFICATE**

Certified that the project work phase-II entitled *“Sentiment Analysis of Twitter Data on Economic Crisis due to COVID-19”* has been successfully carried out at **RNS Institute of Technology** by **VINAY D**, bearing USN **1RN19CS181**, **VINAY H R**, bearing USN **1RN19CS182**, **VIVEK S HEGDE**, bearing USN **1RN19CS186**, bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project work phase-II report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

**Prof. Phanikanth K V**  
Asst. Prof., Guide  
Dept. of CSE

**Dr. Kiran P**  
Professor & HoD

**Dr. Ramesh Babu H S**  
Principal

#### **External Viva:**

**Name of the Examiners**

**Signature with date:**

- 1.
- 2.

# ACKNOWLEDGEMENT

At the very onset, I would like to place on record our gratefulness to all those people who have helped us in making this project work a reality. Our Institution has played a paramount role in guiding us in the right direction.

I would like to profoundly thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of this project work.

I would also like to thank our beloved Director, **Dr. M K Venkatesha**, for providing the necessary facilities to carry out this work.

I would also like to thank our beloved Principal, **Dr. Ramesh Babu H S**, for providing the necessary facilities to carry out this work.

I am extremely grateful to our beloved HoD-CSE, **Dr. Kiran P**, for having accepted to patronize me in the right direction with all his wisdom.

I place my heartfelt thanks to all the Coordinators of project work. I would like to thank the internal guide **Prof. Phanikanth K V**, Asst. Professor for his continuous guidance and constructive suggestions for this work.

Last but not the least, I am thankful to all the staff members of Computer Science and Engineering Department for their encouragement and support throughout this work.

**1RN19CS181**

**1RN19CS182**

**1RN19CS186**

# Abstract

Sentiment analysis is a process which deals with identifying and also classifying the opinions or sentiments expressed in the text by people. Many social networking sites are accumulating a vast amounts of data with every passing day in the form of tweets, blog posts, status updates, comments, etc.

Social networking sites like Twitter, Instagram, Facebook, etc. are rapidly gaining popularity as they allow people to share and express their views about topics, and also facilitate to have discussion with different communities or post messages across the world. There has been lot of work in the field of sentiment analysis of twitter data. This project focuses mainly on sentiment analysis of twitter data which is helpful to analyze the sentiments in the tweets.

In this project, we analyze the twitter posts or tweets that are tweeted about COVID-19 Economic impact on different countries by different twitter users.

Twitter has been specifically used in our project as it has been one of the most used social media platforms as people from various backgrounds use this platform similar to another social media platform like LinkedIn.

# Contents

<b>Acknowledgement</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vi</b>
<b>1 INTRODUCTION</b>	<b>1</b>
<b>2 LITERATURE SURVEY</b>	<b>4</b>
<b>3 PROBLEM STATEMENT</b>	<b>7</b>
3.1 Existing system and its limitation . . . . .	7
<b>4 PROPOSED METHODOLOGY</b>	<b>8</b>
4.1 System design . . . . .	8
4.2 Real-Time Tweets Fetching techniques . . . . .	10
4.2.1 Tweepy . . . . .	10
4.2.2 SNScrape . . . . .	10
4.3 Natural Language Processing (NLP) . . . . .	11
4.3.1 NLP Procedure . . . . .	11
4.3.2 VADER Library . . . . .	11
4.4 Machine Learning Algorithms . . . . .	12
4.4.1 Support Vector Classifier . . . . .	12
4.4.2 Naive Bayes Classifier . . . . .	12

4.4.3	Logistic Regression .....	13
4.5	User Interface Design .....	14
4.5.1	Hypertext Markup Language .....	14
4.5.2	Cascading Style Sheets .....	15
4.5.3	JavaScript .....	15
4.5.4	Bootstrap .....	16
4.6	Server Side Description .....	16
4.6.1	Model .....	16
4.6.2	View .....	17
4.6.3	Template.....	17
4.6.4	URLs .....	17
<b>5</b>	<b>REQUIREMENTS ANALYSIS</b>	<b>18</b>
5.1	Hardware Requirements.....	18
5.2	Software Requirements .....	18
5.3	Functional Requirements .....	19
5.4	Non-functional Requirements.....	19
<b>6</b>	<b>IMPLEMENTATION</b>	<b>20</b>
6.1	Code.....	20
6.1.1	Extracting tweets .....	20
6.1.2	Preprocessing and VADER Code .....	21
6.1.3	Model Creation and Testing.....	24
<b>7</b>	<b>RESULTS</b>	<b>30</b>
<b>8</b>	<b>CONCLUSION AND FUTURE WORK</b>	<b>34</b>
8.1	CONCLUSION.....	34
	<b>References</b>	<b>35</b>

# List of Figures

- 4.1 System Design . . . . . 9
- 4.2 Typical Visualization of Support Vector Machine ..... 12
- 4.3 Naive Bayesian equation..... 13
- 4.4 Typical Visualization of Logestic Regression..... 14
  
- 7.1 Tweets Search screen based on hashtags ..... 30
- 7.2 Tweets Search result screen based on input hashtags ..... 31
- 7.3 Accuracy comparison based on trained tweets ..... 31
- 7.4 SVM prediction on new tweets ..... 32
- 7.5 VADER prediction on new tweets ..... 32
- 7.6 Most frequently occuring words in the tweets ..... 33
- 7.7 Word Cloud showing frequently used words in the tweets ..... 33

# List of Tables

5.1	Hardware Requirements.....	18
5.2	Software Requirements .....	18



# Chapter 1

## INTRODUCTION

Social networking sites have had a significant impact on various aspects of our lives. Social media platforms like Facebook, Twitter, Instagram, and LinkedIn have become an integral part of how we communicate with each other. It has connected like-minded individuals together and has helped to build communities based on their interests, beliefs, and values. News and information can spread quickly through social media platforms, and people can have a wide range of opinions or perspectives on various events.

The COVID-19 pandemic has had a significant impact on the global economy, with many businesses and industries experiencing unprecedented challenges.

A positive impact of COVID-19 on third world economies has been the increase in digitalization. With the increasing threat of infection transmission through physical contact, the virtual space of transactions has gained popularity. The chance of its spread through social contact has accelerated online working platforms and digitally organized logistics. With online transactions and digital platforms for work, there is an opportunity to develop a centralized database that can serve as an economic asset. It has become essential to be a part of the global digital drive for improved socio-economic fortunes and mitigate the impact of the Covid-19 pandemic through digitalization.

Sentiment analysis is a process of analyzing and categorizing the subjective opinions expressed in a text. Twitter, being one of the largest social media platforms with millions of users posting their

opinions every day, is a valuable source of data for sentiment analysis. Sentiment analysis of Twitter data involves using natural language processing (NLP) techniques and machine learning algorithms along with APIs to fetch tweets from twitter and to extract sentiments from tweets. This analysis can be used to understand public opinion on a particular topic, brand, or product, and can help businesses and individuals make data-driven decisions based on the insights gained from analyzing Twitter data. Twitter sentiment analysis can be performed at different levels, including document-level, sentence-level, and aspect-level analysis. Document-level analysis involves classifying the sentiment of an entire tweet, while sentence-level distinguishes the objective sentences expressing factual information and subjective sentences expressing opinions. Aspect-level analysis involves determining sentiment of each aspect word in a sentence with a sentence and some predefined aspect words as input data.

Overall, sentiment analysis of Twitter data can provide valuable insights into consumer behavior, market trends, and public opinion, making it a powerful tool for businesses, researchers, and individuals alike. Sentiment analysis of Twitter data using NLP involves a combination of data collection, cleaning and preprocessing, text representation, sentiment classification, evaluation and optimization, and real-time analysis to accurately classify the sentiment of tweets and provide insights on users opinions or sentiments on various events.

There are several methodologies and techniques that can be used for sentiment analysis of Twitter data, including:

*Lexicon-based approach:* A lexicon-based approach involves using a pre-defined dictionary or list of words with associated sentiment scores to identify the sentiment of each tweet. This approach can be effective for analyzing tweets related to specific domains or topics.

*Supervised machine learning:* Supervised machine learning involves training a machine learning model on a labeled dataset of positive and negative tweets. The model can then be used to classify the sentiment of new tweets as either positive or negative.

*Unsupervised machine learning:* Unsupervised machine learning involves using clustering or topic modeling techniques to group similar tweets based on their sentiment. This can be effective for identifying sentiment in large datasets without the need for manual labeling.

*Hybrid approach:* A hybrid approach combines multiple methodologies, such as rule-based and

machine learning, to improve the accuracy of sentiment analysis.

*Emotion analysis:* Emotion analysis involves identifying and classifying emotions expressed in tweets, such as joy, anger, or sadness. This can provide deeper insights into the sentiment expressed in tweets and can be useful in understanding public opinion and behavior.

*Time-series analysis:* Time-series analysis involves analyzing the sentiment of tweets over a particular time interval to identify trends and patterns in public opinion. This can be useful for monitoring sentiment related to specific events or topics and predicting future sentiment.

Overall, the choice of methodology for sentiment analysis of Twitter data will depend on the specific use case and the characteristics of the Twitter data being analyzed.

Sentiment analysis of data can provide valuable insights into how people are feeling about the economy in real-time. By analyzing tweets related to the economy, policymakers can identify emerging trends, public sentiment, and concerns related to various economic issues. This information can help countries assess their economic situation and develop appropriate policies to cope with it. Moreover, sentiment analysis can help countries monitor public confidence in the economy. If the sentiment is predominantly negative, it may indicate that people are losing faith in the economy, and policymakers may need to take action to boost public confidence. On the other hand, if the sentiment is positive, policymakers may be able to use that positivity to drive economic growth. Finally, sentiment analysis can also help countries track the success of their economic policies in real-time. Overall, sentiment analysis of Twitter data can provide valuable insights into public sentiment and help countries assess their economic situation and develop appropriate policies to cope with it.

# Chapter 2

## LITERATURE SURVEY

One of the newest academic disciplines is sentiment analysis using social media generated data. It is more important since it plays a significant role in the health emergencies such as the COVID-19 pandemic and its economic impact on different countries and other corporations due to COVID-19. Though much research on sentiment categorization and NLP from diverse perspectives is still ongoing, the following are some of the finished works.

**[1] “Covid-19 public sentiment observations and tweet classification using machine learning Algorithms.”**

Samuel used strong textual analytics, vital textual data visualizations to provide some understanding about the growth of dread sentiments over time as COVID-19 approached its pinnacle in the United States. They investigated public perceptions of the virus and COVID19, leading to the discovery of an increase in dread and negative emotion. They also discussed how to obtain first insights using exploratory and expanded textual analytics, as well as textual data visualization. Finally, they conducted a comparative analysis of textual classification algorithms utilized in Artificial Intelligence applications, demonstrating their significance for tweets of various lengths.

**[2] “Top tweeters’ concerns during the COVID-19 disease outbreak: An infoveillance research.”**

For this investigation, Abd Alrazaq used roughly 2.80 million tweets. 167074 tweets from 160830 different individuals met the requirements for inclusion. They looked at tweets on twelve different topics and organized them into 4 key themes: the virus’s sources, its genesis, its impact on people,

countries, its economy, and techniques for reducing the risk of infection. They discovered a favorable mean sentiment for all except two topics: COVID-19-related causalities and greater racism. For increasing racism, they found a minimum of 2722 tweets and a maximum of 13413 for economic loss. They concluded that economic loss had the largest mean of 15.40 while travel bans and the warnings had the smallest mean of around 3.9.

### **[3] “Sentiment analysis is used to predict the outcome of the 2019 Indian election.”**

The sentiment analysis score approach was used by Naiknaware to estimate the popularity of several government programmers in India. To get the results, they followed the seven-step process outlined below:

- 1) Using the Twitter API to extract relevant tweets
- 2) Preprocessing tweets
- 3) Saving the processed tweets present in CSV format
- 4) Analyze each new tweet’s sentiment
- 5) Visualization of results.

They divided the polarity of the statement into positive, negative based on sentence scores established in the 5th phase, and so presented their forecast.

### **[4] “A decision-support technique for sentiment analysis in online stock forums”.**

Wu et al used the sentiment analysis, vector support machines, and they generalized autoregressive (GARCH) conditional hetero modelling to create a unique decision system. This model Is being frequently used to forecast the time series with features like auto correlation and also heteroscedasticity characteristics. Then To accommodate intricate nonlinear and also asymmetric linkages engrafted in finicky forecasting of stocks, they use GARCH for modelling and then incorporate those results into the SVM model. They begin by assigning labels manually polarity for data set postings. Then, using sentiment analysis and a manually annotated data set, they extract features from the text put on the stock forum and use that information to forecast the polarity of other postings automatically. After that, they incorporate each stock’s daily posts. They then used the GARCH-SVM model that resulted to forecast future stock price unpredictability. They then compare the accuracy of this model was eighty one percent, to the accuracy of the lexical based method, which

was seventy five percent.

**[5] “A machine learning strategy for COVID-19 case confirmation: Positive negative death and release.”**

Dutta et al. [26] we conducted a study to see if machine learning could be used to compare or calculate how close the confirmed, negative, discharge, and death predictions were to actual values. To train the dataset, they used Deep learning methods, LSTMs, and gated regression units (GRUs). Then use the actual data to validate the forecast. They concluded that the hybrid LSTMGRU model was better at predicting positive, negative, discharge, and total deaths than the single LSTMGRU model.

# Chapter 3

## PROBLEM STATEMENT

COVID-19 created some severe issues in every nation like downfall in the economy and thus in GDP and other impacts. The social media platforms like twitter, facebook, etc give people to just express their opinions but do not feature any type analysis to bring out sentiments on these opinions expressed by twitter users.

Through our project we will analyze sentiments on economic crisis based on tweets by various twitter users. This analytics can help various organisations and governments of different countries for better governance of people to improve upon the current economic situation in both organization as well as in different countries.

### 3.1 Existing system and its limitation

The social media platforms like Twitter, Facebook, etc gives people to express their opinions but do not feature any type analysis on these opinions to bring sentiments out of such tweets which is valuable for various other purposes.

# Chapter 4

## PROPOSED METHODOLOGY

### 4.1 System design

The process of implementing numerous requirements and enabling their physical embodiment is referred to as “system design.” The system is developed using a variety of design principles; the design specification outlines the features of the system, its competitors or constituent parts, and how they will appear to end users. Economic impact or crisis related new tweets are extracted from twitter API (i.e, tweepy) and Machine Learning (ML) models along with a popular social media sentiment analyzer tool VADER both predict the sentiments on newly fetched tweets.

ML models used are Supervised ML models which require training the model before the actual prediction on new data.

Supervised ML algorithms used to demonstrate in this project are Naive Bayes, Logistic Regression, Support Vector Machines or Classifiers (SVM or SVC) which were the three best out of five different algorithms chosen to check out the results. This included Decision Tree Classifier and Random Forest Classifier. Due to more training and testing time involved for these two algorithms when compared to above mentioned three different algorithms they have been not included as part of this project.

The below section shows basic procedure followed in this project:

**Data collection:** The first step is to collect a sample of Twitter data related to a topic or brand of interest. This can be done using the Twitter API or by using some advanced web scraping tools



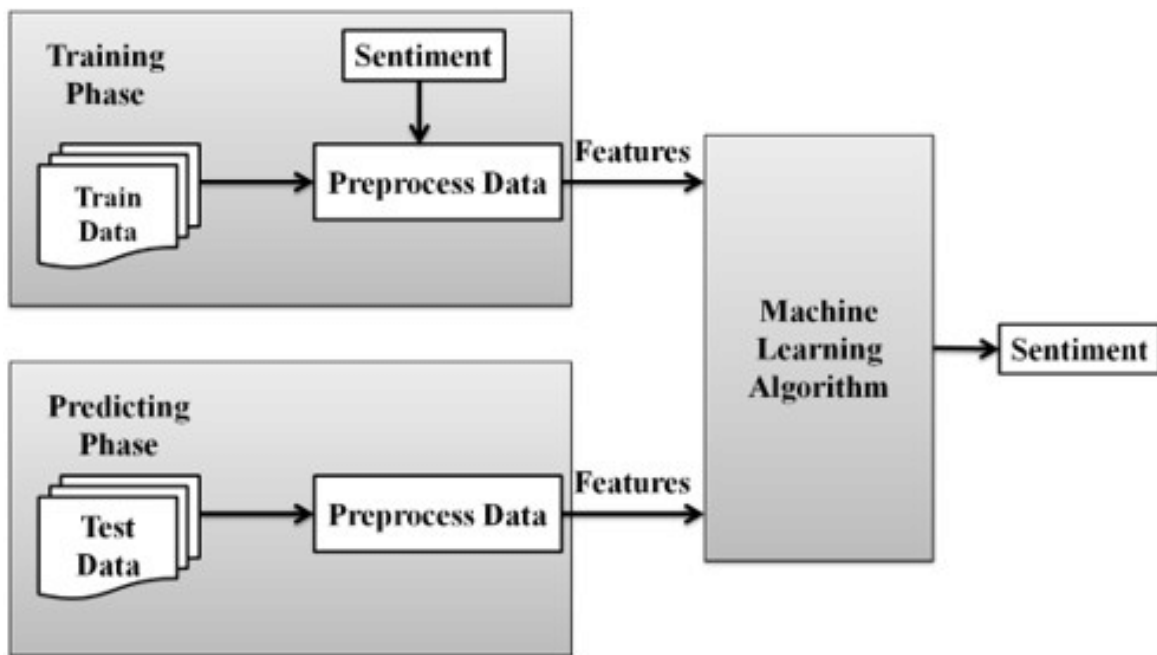


Figure 4.1: System Design

as some platforms do not allow scraping without authorization, to extract tweets containing specific hashtags.

**Data cleaning and preprocessing:** Once the data has been collected, it needs to be cleaned and preprocessed to remove any irrelevant information and prepare it for analysis. This may involve removing stop words, punctuation, and special characters, as well as stemming or lemmatizing the text to reduce the dimensionality of the data.

**Text representation:** The next step is to represent the preprocessed text in a numerical format that can be used for analysis. This can be done using methods such as bag-of-words, TF-IDF (term frequency-inverse document frequency), or word embedding's. ML models are trained on preprocessed tweets to predict sentiment of new tweets. The new tweets from twitter are fetched by providing hash tag of a subject either by using tweepy library or by the SNScrape scraping tool. The imported tweets are analyzed for their sentiments and classified either as positive or negative by trained Machine Learning models like Naïve Bayes, Support Vector Machine (SVM), Logistic Regression models based on accuracy of these models. These imported tweets are also analyzed using VADER sentiment analyzer tool in python ultimately comparing predictions of ML model producing result accurately to that of VADER sentiment analyzer tool.

## 4.2 Real-Time Tweets Fetching techniques

### 4.2.1 Tweepy

With Tweepy, a Python library that offers a straight-forward interface for accessing the Twitter API, developers can programmatically retrieve various Twitter data such as user profiles, tweets, follower lists, and more. Tweepy simplifies the process of interacting with the Twitter API by providing an easier way to authenticate, build requests, and parse responses using python as programming language. It also provides an easy-to-use interface for streaming real-time Twitter data, such as tweets matching a specific keyword or hashtag.

Some of the key features of Tweepy include:

- Tweepy provides a simple interface for authenticating with the Twitter API, using OAuth 1.0a authentication.
- Tweepy provides an intuitive and easy-to-use interface for accessing the Twitter API, making it easy to get started with minimal coding knowledge.
- Tweepy has extensive documentation, including examples and tutorials, making it easy to learn and use.
- Tweepy provides a simple interface for handling real-time streaming of Twitter data, such as tweets matching specific keywords or hashtags.

### 4.2.2 SNScrape

SNScrape is a python library utilized for social media data scraping from multiple platforms including Twitter, Instagram, YouTube, etc. It provides an alternative way to collect data from social media platforms that do not offer a public API or limit access to their API. SNScrape works by scraping the HTML code of a social media page and extracting relevant information such as tweets, Instagram posts, and YouTube videos. Moreover, this library also enables the scraping of historical data, including posts and comments that were made in the past. SNScrape is particularly useful for researchers, journalists, and data analysts who need to collect large amounts of social media data for analysis. It can be used to collect data for sentiment analysis, trend analysis, or any other type of analysis that requires social media data. Some of the key features of SNScrape include:

- SNScrape supports multi-threaded scraping, which can significantly improve scraping speed.
- SNScrape has extensive documentation that provides detailed instructions on how to use the library.
- SNScrape offers a user-friendly and straightforward interface, facilitating a smooth and uncomplicated initiation to extract social media information through web scraping techniques.

## **4.3 Natural Language Processing (NLP)**

### **4.3.1 NLP Procedure**

Many of the above-mentioned NLP activities, as well as subtasks like sentence parsing, word segmentation, stemming and lemmatization (word-trimming methods), and tokenization, are covered by the NLTK or also called natural language toolkit aid the computer in comprehending the text, the text is segmented into tokens, which involves breaking phrases, sentences, paragraphs, and passages into smaller units. It also includes libraries for developing skills like semantic reasoning that allows users to make logical inferences from text data.

### **4.3.2 VADER Library**

The VADER library is a python library that is used for sentiment analysis of textual data. It is an acronym for the “Valence Aware Dictionary and Sentiment Reasoner” (VADER), created by a team of scholars from the Georgia Institute of Technology. VADER uses a lexicon of words and their associated sentiment scores to analyze the sentiment of a piece of text. The lexicon contains words that are rated on a scale from -4 to +4, with negative scores indicating negative sentiment and positive scores indicating positive sentiment. The lexicon further encompasses modifiers that modify the sentiment score of the term they are associated with. In addition to the lexicon, VADER also takes into account the context of the text and the punctuation used, as these can significantly impact the sentiment of the text. The output of VADER is a sentiment score between -1 and +1, where negative scores indicate negative sentiment, positive scores indicate positive sentiment, and scores close to zero indicate neutral sentiment.

## 4.4 Machine Learning Algorithms

### 4.4.1 Support Vector Classifier

The Support Vector Machine (SVM) is a supervised machine learning technique that can be utilized for both classification and regression tasks, and is sometimes referred to as the Support Vector Classifier. The aim of the Support Vector Machine (SVM) algorithm is to identify a hyperplane in an N-dimensional space that can effectively separate the data points into distinct classes. The hyperplane's dimensionality is defined by the number of features, where a line is sufficient for two input features to form a hyperplane. It becomes challenging to visualize the hyperplane when the number of features considered exceeds three.

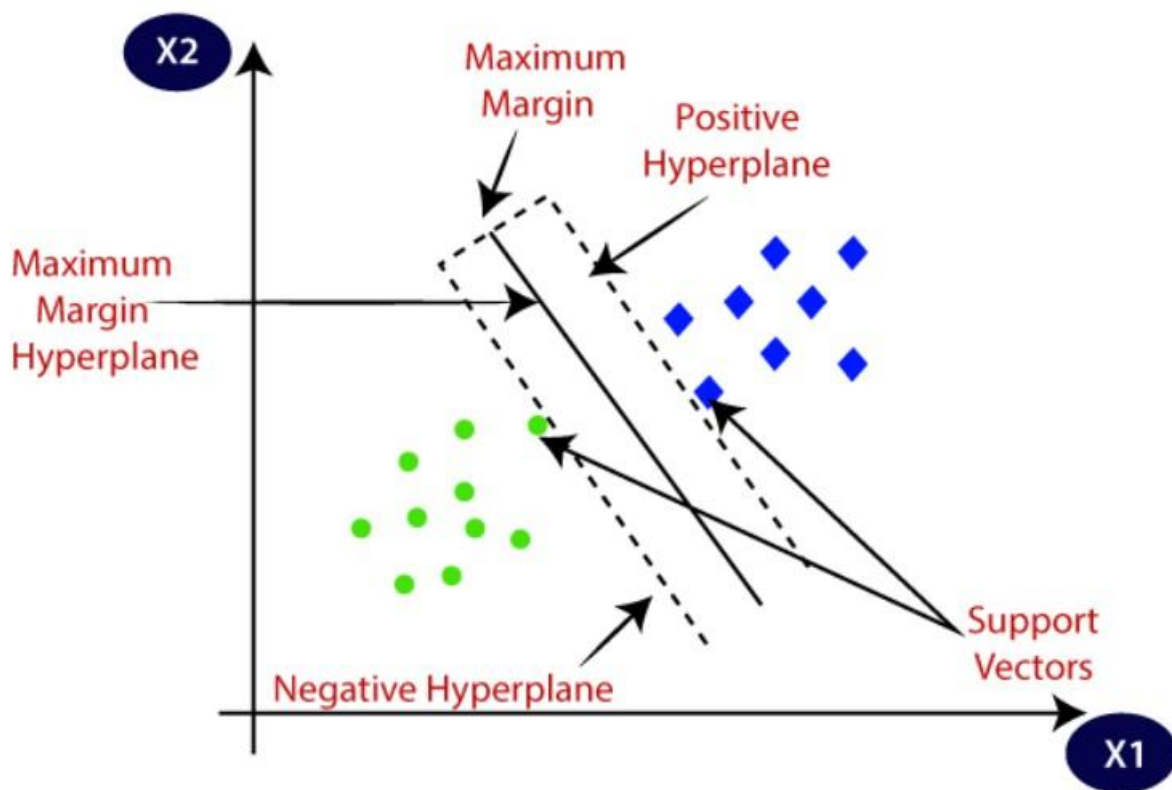


Figure 4.2: Typical Visualization of Support Vector Machine

### 4.4.2 Naive Bayes Classifier

The Naive Bayes classifier employs Bayes' theorem, also referred to as Bayes' Rule or Bayes' law, to calculate the probability of a hypothesis based on prior knowledge. It depends on the conditional probability. Bayes' theorem is given as:  $P(A|B) = (P(B|A)P(A)) / P(B)$  where,

- $P(A|B)$  is Posterior probability: Probability of hypothesis A on the observed event B.
- $P(B|A)$  is the Likelihood probability: The probability of the evidence given that a hypothesis is true.
- $P(A)$  is the Prior Probability: The probability of the hypothesis before observing the evidence.
- $P(B)$  is Marginal Probability: Probability of Evidence.

Conditional classification problem with Bayes Theorem is as follows:

$$P(y|x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i|y)}{P(x_1)P(x_2)\dots P(x_n)}$$

Figure 4.3: Naive Bayesian equation

From above figure, estimating the prior probability  $P(y)$  is a straightforward task using a dataset. However, computing the conditional probability of an observation based on a class,  $P(x_1, x_2, \dots, x_n | y)$ , is not feasible without an exceptionally large number of examples. In such cases, it is necessary to effectively estimate the probability distribution for all possible combinations of values.

### 4.4.3 Logistic Regression

Logistic Regression is a predictive modeling algorithm or a technique used to determine the categorically dependent variable based on a given set of independent variables. The outcome must be a categorical or discrete value, such as Yes or No, 0 or 1, True or False, Positive or Negative, etc. Instead of providing an exact value of 0 or 1, Logistic Regression provides probabilistic values ranging between 0 and 1 later on rounding it to nearest value among a 0 and 1. The approach of Logistic Regression is similar to Linear Regression. While Linear Regression is applied to solve regression problems, Logistic Regression is applied to address classification problems. The below figure demonstrates typical visualization of logistic regression algorithm, which can help in understanding its working phenomena.

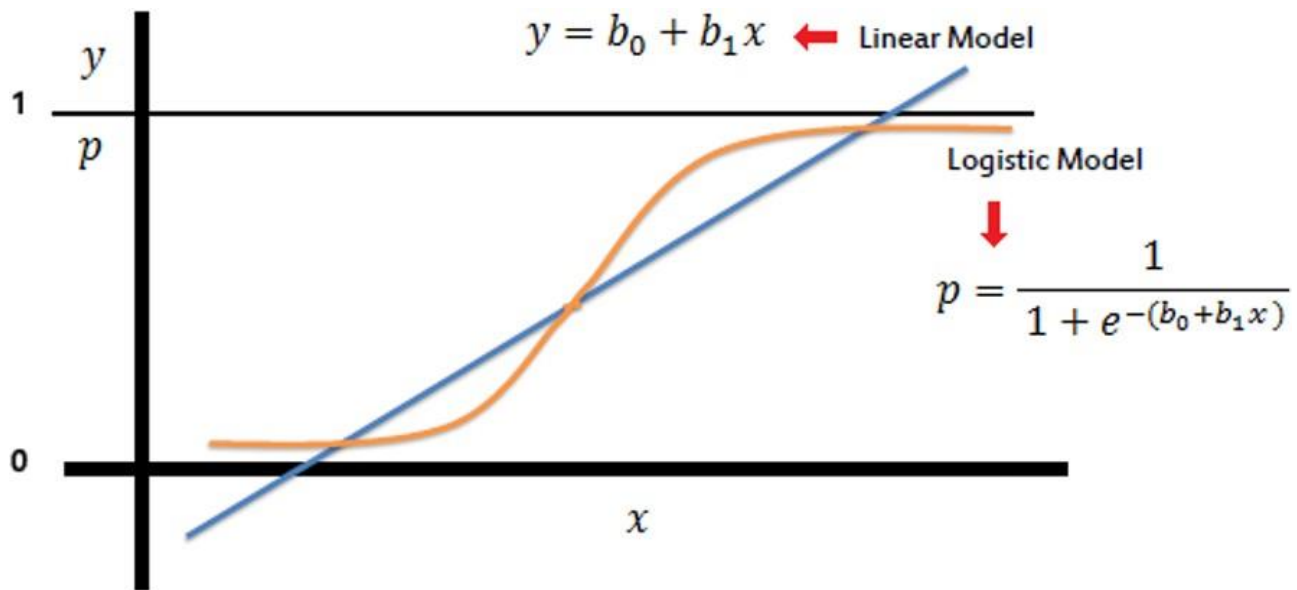


Figure 4.4: Typical Visualization of Logistic Regression

## 4.5 User Interface Design

The front-end is built using a combination of technologies such as Hypertext Markup Language (HTML), JavaScript, Bootstrap and Cascading Style Sheets (CSS) used with the help of Django templates. Front-end developers design and construct the user experience elements on the web page or app including buttons, menus, pages, links, graphics and more.

### 4.5.1 Hypertext Markup Language

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web (WWW). Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages, HTML, describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images, and other objects, such as interactive forms, may be embedded into the rendered page. It provides a direct link. In order to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `<img>` and `<input>` introduce content into the page others such

as `<p>...</p>` surround and provide information about document text and may include other tags as subelements. Browsers do not display the HTML tags but use them to interpret the content of the page. HTML can embed programs written in a JavaScript which affect the behavior and content of web pages.

### **4.5.2 Cascading Style Sheets**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML Document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML, JavaScript, CSS which is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications. CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate CSS file, and reduce complexity and repetition in the structural content.

### **4.5.3 JavaScript**

JavaScript is the Programming Language for the Web. It can update and change both HTML and CSS. JavaScript can calculate, manipulate, and validate data. JavaScript is a dynamic computer programming language. It is lightweight and most used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities. JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

### 4.5.4 Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile First front-end web development. It contains CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

- Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing a responsive and mobile friendly website.
- It is free to download and use.
- It is a front-end framework used for easier and faster web development.
- It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
- It can also use JavaScript plug-ins.
- It facilitates you to create responsive designs.

## 4.6 Server Side Description

Django is a python framework that makes it easier to create web sites using Python. Django emphasizes on reusability of components, also referred to as DRY (Don't Repeat Yourself), and comes with ready-to-use features like login system, database connection and CRUD operations (Create, Read, Update, Delete). Django follows the MVT design pattern (Model View Template).

### 4.6.1 Model

The model provides data from the database. In Django, the data is delivered as an Object Relational Mapping (ORM), which is a technique designed to make it easier to work with databases. The most common way to extract data from a database is using SQL. One problem with SQL is that an expectation of pretty good understanding of the database structure to be able to work with it. Django, with ORM, makes it easier to communicate with the database, without having to write complex SQL statements. The models are usually located in a file called models.py in Django.



### 4.6.2 View

A view is a function or method that takes HTTP requests as arguments, imports the relevant model(s), and finds out what data to send to the template, and returns the final result. The views are usually located in a file called `views.py` in Django.

### 4.6.3 Template

A template is a file where you describe how the result should be represented. Templates typically are files with `.html` extension, with HTML code describing the layout of a web page. Django uses standard HTML to describe the layout, but uses Django tags to add logic. The templates of an application is located in a folder named `templates`. The variables in these templates obtained as arguments from the View section are expressed using a pair of double curly braces as `{{}}` and for any conditional statements to be expressed as `{% ... %}` which can include variables to satisfy the condition.

### 4.6.4 URLs

Django also provides a way to navigate around the different pages in a website. When a user requests a URL, Django decides which view it will send it to that route. This is done in a file called `urls.py` in Django. When we have installed Django and created your first Django web application, and the browser requests the URL, this is basically what happens:

- Django receives the URL, checks the `urls.py` file, and calls the view that matches the URL.
- The view, located in `views.py`, checks for relevant models.
- The models are imported from the `models.py` file.
- The view then sends the data to a specified template in the template folder.
- The template contains HTML and Django tags, and with the data it returns finished HTML content back to the browser.

# Chapter 5

## REQUIREMENTS ANALYSIS

### 5.1 Hardware Requirements

Name of Components	Specification
Processor	Intel CORE i5 10th Gen
RAM	8 GB
Hard Disk	256 GB SSD

Table 5.1: Hardware Requirements

### 5.2 Software Requirements

Name of Components	Specification
Operating System (OS)	Windows 10 or 11
Language	Python, HTML, CSS, Javascript
Browser	Google Chrome
Integrated Development Environment (IDE)	Microsoft Visual Studio Code

Table 5.2: Software Requirements

## 5.3 Functional Requirements

- **Data Extraction:-** Using “Tweepy” python library which gives access to twitter API to extract tweets.
- **Data Preprocessing:-** Using Natural Language Toolkit (NLTK) library of Python.
- **VADER Lexicon Sentiment Analysis Library:-** Tweets classified as positive or negative using VADER library.
- **Machine learning classification techniques:-** Primarily algorithms like Naïve Bayes, Support Vector Machine (SVM), Logistic Regression models are trained on tweets which can be later used to analyze sentiment of new tweets.

## 5.4 Non-functional Requirements

- **Accessibility:-** The design of services or environments so as to be easily accessible and usable by people (Web Application).
- **Emotional factors:-** Users emotions or sentiments can be understood either as positive or negative.
- **Maintainability:-** Tools used are open source and has easy documentation support.

# Chapter 6

## IMPLEMENTATION

### 6.1 Code

#### 6.1.1 Extracting tweets

```
from tweepy import OAuthHandler
from tweepy import API
from tweepy import Cursor
from datetime import datetime, date, time, timedelta
import tweepy
import numpy as np
import pandas as pd
import snsrape.modules.twitter as sntwitter

class Import_tweet_sentiment:
    consumer_key="QIqgjITOfksfMW4lRLDacQ"
    consumer_secret="R8x0xN9iSKXGNxUtGKA2hgnlIhh5INZIOdgEfxzk"
    access_token="1401204486BeLUAuruh294KeJX8NXvdqjCeZOQcLl6HWm
MlgA"
    access_token_secret="pwjiLF42TbORaXtkCS5Oc24qywOU0eFN0esVci
bA"
```

```
def get_hashtag(self, hashtag):  
    auth = OAuthHandler(self.consumer  
        _key, self.consumer_secret)  
    auth.set_access_token(self.access_token, self  
        .access_token_secret)  
  
    q = "((India AND Pakistan) AND (economic crisis  
        OR financial OR income OR tax OR gdp OR #COVID19  
        OR #Corona OR pay OR loan OR inflation OR market  
        OR gst)) -filter:retweets" # for tweepy  
  
    api = tweepy.API(auth)  
    atweets = []  
    for tweet in tweepy.Cursor(api.search, q,  
        lang="en").items(700):  
        cleaned_tweet = senti.sentiment_  
            analysis_code.cleaning(self, tweet.text)  
        tweet_sentiment = senti.sentiment_  
            analysis_code.get_tweet_sentiment(self  
            , cleaned_tweet)  
        atweets.append([tweet.user.screen_name,  
            tweet.text, cleaned_tweet ,  
            tweet_sentiment])  
    return atweets
```

## 6.1.2 Preprocessing and VADER Code

```
import re  
from nltk.stem.wordnet import WordNetLemmatizer  
import itertools
```

```

import numpy as np

import nltk

from vaderSentiment.vaderSentiment

import SentimentIntensityAnalyzer

from nltk.stem import WordNetLemmatizer

from nltk.corpus import stopwords

import string


class sentiment_analysis_code:

    def cleaning(self, text):

        text = text.lower()

        text = re.sub(r'(\u[0-9A-Fa-f]+)', ' ', text)

        text = re.sub(r'[\x00-\x7f]', ' ', text)

        text = re.sub('@[\s]+', ' ', text)

        text = " ".join(re.split(r'\s|-', text))

        text = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', ' ', text)

        text = re.sub(r'#covid', r' covid ', text)

        text = re.sub(r'#corona', r' corona ', text)

        text = re.sub(r'#coronavirus', r' coronavirus ', text)

        text = re.sub(r'#([^\s]+)', r' ', text)

        text = ''.join([i for i in text if not i.isdigit()])

        text = re.sub(r"(!)\1+", ' ', text)

        text = re.sub(r"(?)\1+", ' ', text)

        text = re.sub(r"(\.)\1+", ' ', text)

        text = re.sub('(:\)|;\)||:-\)|\(-:|:-D|=D|:P|xD|X-p|\^\^|:-|\^\.\^\|^\-^\|^\_\^\|,-\)|\)-:|:\'\(|:-\(|:\S|T\.T|\.\_\.\.:<|:-\S|:-<|\\-\\*|:O|=O|=\\-O|O\o|XO|O\_O|:-\\@|=|:/|X\\-\\(|>\\.<|>=\\(|D:', ' ', text)

        text = re.sub('&', ' and', text)

```

```
#stoplist = stopwords.words('english')
stop_words = set(stopwords.words('english'))
stop_words.discard('not')
stop_words.discard('and')
stop_words.discard('but')

translator = str.maketrans('', '', string.punctuation)
text = text.translate(translator)
# Technique 7: remove punctuation
tokens = nltk.word_tokenize(text)
tokens = [tokens for tokens in tokens if not tokens
in stop_words]

lemmatizer = WordNetLemmatizer()
tokens = [lemmatizer.lemmatize(token) for token in tokens]

tagged = nltk.pos_tag(tokens)
# Technique 13: part of speech tagging
allowedWordTypes = ["J", "R", "V", "N"]
# J is Adjective, R is Adverb, V is Verb, N is Noun.
These are used for POS Tagging
final_text = []
for w in tagged:
    if (w[1][0] in allowedWordTypes):
        final_word = sentiment_analysis_code()
        .addCapTag(w[0])
        final_word = lemmatizer.lemmatize(final_word)
        final_text.append(final_word)
    text = " ".join(final_text)

return text
```

```
def addCapTag(self, word):  
    """ Finds a word with at least 3 characters capitalized and  
    adds the tag ALL_CAPS_ """  
    if(len(re.findall("[A-Z]{3,}", word))):  
        word = word.replace('\', ' ' )  
        transformed = re.sub("[A-Z]{3,}", "ALL_CAPS_"+word  
        ,word)  
        return transformed  
    else:  
        return word  
  
def get_tweet_sentiment(self, tweet):  
    #cleaning of tweet  
    sid_obj = SentimentIntensityAnalyzer()  
    sentiment_dict = sid_obj.polarity_scores(  
    sentiment_analysis_code().cleaning(tweet))  
    if sentiment_dict['compound'] > 0 :  
        return 'Positive'  
    else :  
        return 'Negative'
```

### 6.1.3 Model Creation and Testing

```
from django.shortcuts import render, redirect, HttpResponseRedirect  
from django.contrib import messages  
import csv  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.metrics import accuracy_score, precision_score
```



```
from sklearn.svm import LinearSVC #Linear SVM
from sklearn.naive_bayes import ComplementNB #Naive Bayes
from sklearn.linear_model import LogisticRegression
#Logistic Regression
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer

def sentiment_analysis_import(request):
    if request.method == 'POST':
        form = Sentiment_Imported_Tweet_analyse_form(request.POST)
        tweet_text = Import_tweet_sentiment()
        analyse = sentiment_analysis_code()
        if form.is_valid():
            handle = form.cleaned_data['sentiment_imported_tweet']
            dict={}
            a=[]
            if handle in ["#COVID19", "#CovidIsNotOver",
"#CovidVaccines", "#Coronavirus", "#CoronavirusUpdates",
"#Corona"]:
                list_of_tweets = tweet_text.get_hashtag(handle)
                list_of_tweets_and_sentiments = []
                for i in list_of_tweets:
                    list_of_tweets_and_sentiments.append(['@' +
                    i[0], i[1], i[2], i[3]])
                    a.append(['@' + i[0], i[1], i[2], i[3]])
                df = pd.DataFrame(a, columns=['Tweet Username',
'Tweet Text', 'Cleaned Tweet Text', 'Result'])
                df.to_csv('compare.csv', index=False)
                df = pd.read_csv("compare.csv")
```

```
dataset = pd.read_csv('final.csv',
encoding_errors='ignore')
dataset = dataset.dropna()
dataset = dataset.reset_index(drop=True)
X=dataset['Cleaned Tweet Text']
Y=dataset['Result']
X_train, X_test, Y_train, Y_test =
train_test_split(X,Y,test_size=0.2,
random_state=42)
data = pd.read_csv("finalD.csv",
encoding_errors='ignore', low_memory=False)
data = data.dropna()
data = data.reset_index(drop=True)
XX=data['Cleaned Tweet Text']
YY=data['Result']
XX_train, XX_test, YY_train, YY_test =
train_test_split(XX,YY,test_size=0.2,
random_state=42)
tfacc = TfidfVectorizer(ngram_range=(3,3))
xx_train = tfacc.fit_transform(XX_train)
xx_test = tfacc.transform(XX_test)
print("Xtrain =",len(XX_train))
print("Xtest =",len(XX_test))
print("ytrain =",len(YY_train))
print("ytest =",len(YY_test))
count2=len(XX_train)

BNBmodel = ComplementNB()
BNBmodel.fit(xx_train, YY_train)
y_predBNB = BNBmodel.predict(xx_test)
```

```
score =
round(accuracy_score(YY_test, y_predBNB)*100, 3)
pscore =
round(precision_score(YY_test, y_predBNB,
average="micro")*100, 3)
print("Accuracy of Naive Bayes: %0.3f" % score)
print("Precision of Naive Bayes: %0.3f" % pscore)

LGmodel = LogisticRegression(C=1, max_iter=1000)
LGmodel.fit(xx_train, YY_train)
y_predLG = LGmodel.predict(xx_test)
score2 =
round(accuracy_score(YY_test, y_predLG)*100, 3)
pscore2 =
round(precision_score(YY_test, y_predLG,
average="micro")*100, 3)
print("Accuracy of Logistic Regression:
%0.3f" % score2)
print("Precision of Logistic Regression:
%0.3f" % pscore2)

SVMaccModel = LinearSVC()
SVMaccModel.fit(xx_train, YY_train)
y_predaccSVM = SVMaccModel.predict(xx_test)
score1 =
round(accuracy_score(YY_test, y_predaccSVM)*100,
3)
pscore1 =
round(precision_score(YY_test, y_predaccSVM,
average="micro")*100, 3)
```

```
print("Accuracy of SVM: %0.3f" % score1)
print("Precision of SVM: %0.3f" % pscore1)

tfidf_vectorizer = TfidfVectorizer()
tfidf_xtrain = tfidf_vectorizer.fit_transform
(X_train)
tfidf_xtest = tfidf_vectorizer.transform(X_test)
SVCmodel = LinearSVC()
SVCmodel.fit(tfidf_xtrain, Y_train)
y_predSVC = SVCmodel.predict(tfidf_xtest)
ml_pred = pd.read_csv("compare.csv")
ml_pred1 = tfidf_vectorizer.transform
(ml_pred['Cleaned Tweet Text'])
test_pred = SVCmodel.predict(ml_pred1)

dicti = {'Prediction': test_pred}
df1 = pd.DataFrame(next(iter(dicti.values()))),
columns=['ML Result'])
out = pd.merge(ml_pred, df1,
left_index=True, right_index=True)
out.to_csv("compareres.csv", index=False)
args = {'vad_ml_sentiment': vad_ml_sentiment,
'handle': handle, 'overall_senti': se,
'most': most, 'Total': count1,
'Positive_Count': poscont1,
'Negative_Count': negcont1,
'Positive_Count1': poscont,
'Negative_Count1': negcont, 'Naive': score,
'Logistic': score2, 'SVM': score1,
'total': count2}
```

```
        return render(request,
                        'home/sentiment_import_result.html', args)
    messages.error(request,
                    "Please give an appropriate hashtag.")
    return render(request, 'home/sentiment_import.html')
else:
    form = Sentiment_Imported_Tweet_analyse_form()
    return render(request, 'home/sentiment_import.html')
```

# Chapter 7

## RESULTS

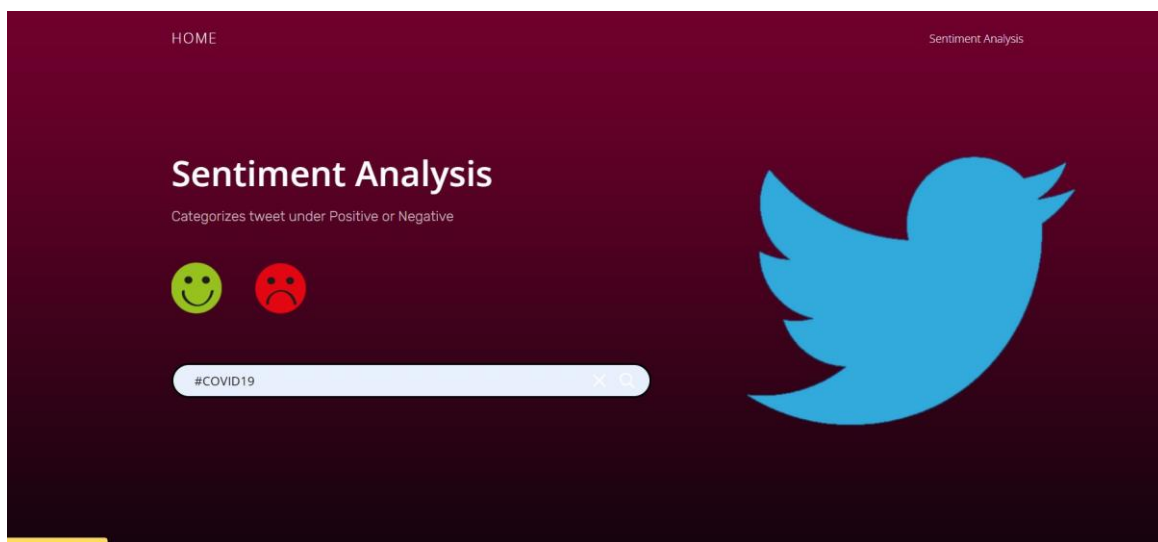


Figure 7.1: Tweets Search screen based on hashtags

In the above figure, the browser screen depicts a search box where users can enter COVID-19 related hashtags to fetch economic crisis related tweets.

HOME Sentiment

HANDLE	TWEET	CLEANED TWEET	VADER SENTIMENT	ML(SVM) SENTIMENT	BASED ON ML EMOTAG
@TanveerAmann	A positive step towards diplomacy.India is the forth largest economy and a mega market. What Pakistan's	positive step towards diplomacyindia forth largest economy mega market pakistan desperately	Positive	Positive	😊




Figure 7.2: Tweets Search result screen based on input hashtags

The below picture depicts information on what hashtag is used to fetch economic crisis related tweets along with total number of tweets used to train ML models and total numbers of tweets fetched for the given hashtag.

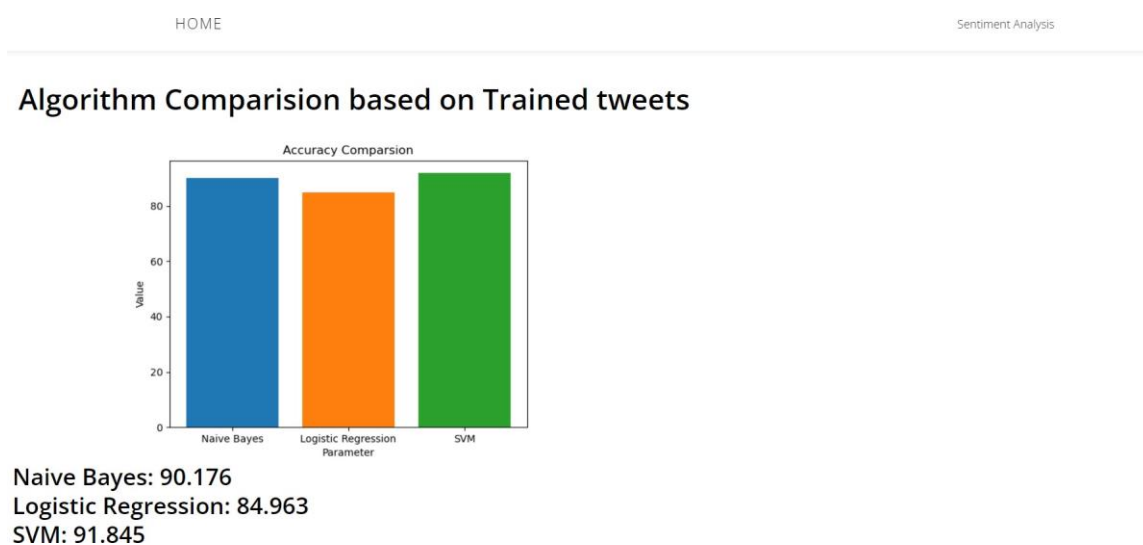
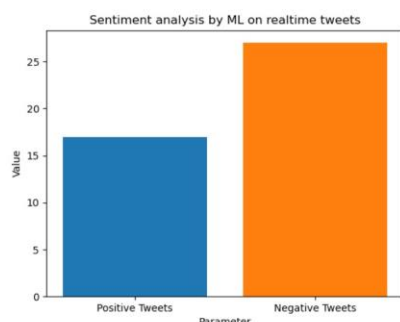


Figure 7.3: Accuracy comparison based on trained tweets

The above image depicts accuracies of three different algorithms namely Naive Bayes, Logistic Regression and SVM. It clearly shows that SVM has highest accuracy out of all three algorithms.

**SVM has more accuracy. SVM is used to Analyse the Fetched tweets**

**Sentiments on Fetched tweets by ML(SVM) algorithm**

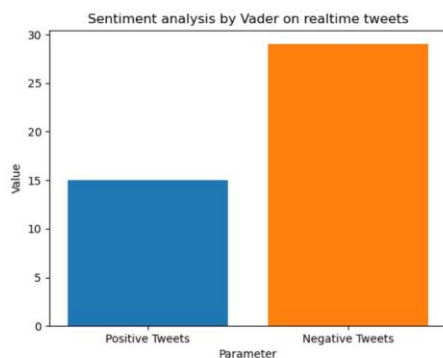


**Positive Tweets: 17**  
**Negative Tweets: 27**

Figure 7.4: SVM prediction on new tweets

This image depicts the classification done by SVM algorithm because of its highest accuracy. It also includes total number of positive and negative tweets count predicted by SVM.

**Sentiments on Fetched tweets by VADER**



**Positive Tweets: 15**  
**Negative Tweets: 29**

Figure 7.5: VADER prediction on new tweets

This image depicts the classification done by VADER Lexicon sentiment analysis library which is a popular library for social media sentiment analysis. It also includes total number of positive and negative tweets count predicted by VADER.

From previous two images showing the count of positive and negative sentiments of both the prediction of ML model, SVM model and VADER library can be compared.



WORD	OCCURENCES
pakistan	21
india	19
economic	18
crisis	10

Figure 7.6: Most frequently occurring words in the tweets

This is an image of a table in the application depicting most frequently occurring words and its frequency.

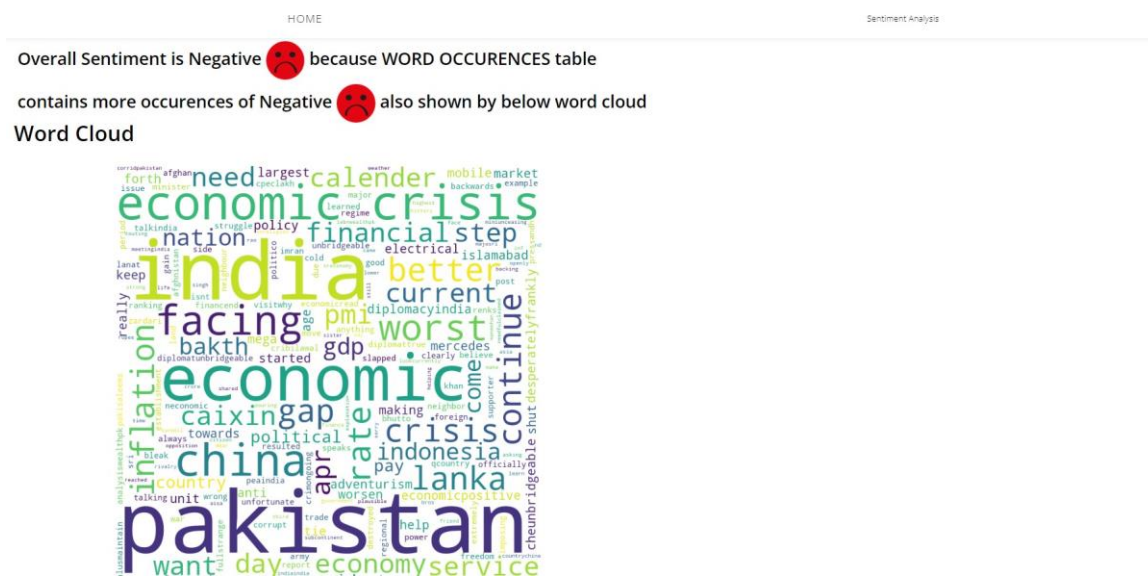


Figure 7.7: Word Cloud showing frequently used words in the tweets

These two figures help in determining overall sentiment of the fetched tweets. The process of determining overall sentiment for the given tweets is done by finding sentiment on most occurred words.

# Chapter 8

## CONCLUSION AND FUTURE WORK

### 8.1 CONCLUSION

Social media is witnessing a massive increase in the number of users per day.

People prefer to share their honest opinions on social media instead of sharing with someone in person due to its impact. Using the posts from Twitter, we can examine the common public's aggregate reaction on economic impact due to COVID-19.

Then the web application which takes the tweets as input and predict the sentiments using Machine Learning model SVM is used to obtain the sentiment of new tweets due to its high accuracy, along with VADER library results and compared to see that both results are comparable.

# REFERENCES

- [1] Wu, J. T., Leung, K., Leung, G. M. (2020). Now casting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: a modelling study. *The Lancet*, 395(10225), 689-697.
- [2] Medford, R. J., Saleh, S. N., Sumarsono, A., Perl, T. M., Lehmann, C. U. (2020). An “Infodemic”: leveraging high-volume Twitter data to understand public sentiment for the COVID-19 outbreak. *medRxiv*. Preprint posted online April 7.
- [3] Li, S., Wang, Y., Xue, J., Zhao, N., Zhu, T. (2020). The impact of COVID-19 epidemic declaration on psychological consequences: a study on active Weibo users. *International journal of environmental research and public health*, 17(6), 2032.
- [4] Kayes, A. S. M., Islam, M. S., Watters, P. A., Ng, A., Kayesh, H. (2020). Automated measurement of attitudes towards social and economic impact using social media: a COVID-19 case study.
- [5] Pastor, C. K. L. (2020). Sentiment analysis on synchronous online delivery of instruction due to extreme community quarantine in the Philippines caused by COVID-19 pandemic. *Asian Journal of Multidisciplinary Studies*, 3(1), 1-6.
- [6] Dubey, A. D. (2020). Decoding the Twitter Sentiments towards the Leadership in the times of COVID-19: A Case of USA and India. Available at SSRN 3588623.
- [7] Chen, L., Lyu, H., Yang, T., Wang, Y., Luo, J. (2020). In the eyes of the beholder: analyzing social media use of neutral and controversial terms for COVID-19. *arXiv preprint arXiv:2004.10225*.
- [8] Alhajji, M., Al Khalifah, A., Aljubran, M., Alkhalifah, M. (2020). Sentiment analysis of tweets in Saudi Arabia regarding governmental preventive measures to contain COVID19.
- [9] Samuel, J., Ali, G. G., Rahman, M., Esawi, E., Samuel, Y. (2020). Covid-19 public sentiment

insights and machine learning for tweets classification. *Information*, 11(6), 314.

[10] Liu, R., Shi, Y., Ji, C., Jia, M. (2019). A survey of sentiment analysis based on transfer learning. *IEEE Access*, 7, 85401-85412.

[11] A. Abd-Alrazaq, D. Alhuwail, M. Househ, M. Hamdi, and Z. Shah, "Top concerns of tweeters during the COVID-19 pandemic: Infoveillance study," *J. Med. Internet Res.*, vol. 22, no. 4, Apr. 2020, Art. no. e19016.