

# **ENHANCED SENTIMENT ANALYSIS ON STUDENT'S COVID-19 RELATED TWEETS**

Submitted in partial fulfillment of the requirements for the award of Bachelor of  
Engineering Degree in

Computer Science and Engineering

By

SANGISETTY JITENDRA KALYAN

(Reg. No: 37110679)

SANDEEP CHUNDU

(Reg. No.: 37110675)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL OF  
COMPUTING**

# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by  
AICTE**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119.**

**March-2021**



# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(Deemed To Be University)  
Accredited with Grade "A" by NAAC | 12B Status by  
UGC | Approved by AICTE  
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119.  
[www.sathyabama.ac.in](http://www.sathyabama.ac.in)**



---

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### **BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the Bonafide work **SANGISETTY JITENDRA KALYAN (37110679)** who carried out the project entitled **"ENHANCED SENTIMENT ANALYSIS ON STUDENT'S COVID-19 RELATED TWEETS"** under our supervision from November 2020 to March 2021.

#### **Internal Guide**

Ms. T. ANANDHI, M.E.,

#### **Head of the Department**

Dr. S. VIGNESHWARI, M.E., Ph.D.,

---

Submitted for Viva voce Examination held on \_\_\_\_\_

**Internal Examiner**

**External Examiner**

## **DECLARATION**

I **SANGISETTY JITENDRA KALYAN (37110679)** hereby declare that the Project Report entitled as “**ENHANCED SENTIMENT ANALYSIS ON STUDENT’S COVID-19 RELATED TWEETS**” done by me under the guidance of **Ms. T. ANANDHI, M.E.**, at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

**DATE: 08-04-2021**

**S.Jitendra Kalyan**

**PLACE:**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASAIKALA, Ph.D., Dean School of Computing** and **Dr. S. VIGNESHWARI. M.E., Ph.D., Head of the Department, Dept. of COMPUTER SCIENCE ENGINEERING**, school of computing for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude of my Project Guide **Ms. T. Anandhi, M.E.**, for her valuable guidance, suggestions & constant encourage to successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **COMPUETR SCIENCE AND ENGINEERING** who were helpful in many ways for the completion of the project.

## **ABSTRACT**

Coronavirus (COVID-19) brought a mix of similar emotions from the nations towards the decisions taken by their respective governments. Social media was bombarded with posts containing both positive and negative sentiments on the COVID-19, pandemic, lockdown, and hash tags past couple of months. The purpose of this study is to analyse reaction of students to the novel Corona virus. Naïve Bayes' and Support vector machine models are used for analysing the student's emotions. The use of emotions showed a unique and novel way of validating the supervised deep learning models on tweets extracted from Twitter. The objective of this project is to design and develop an application for analysis of student activity tweets. In this project we take the students data and we train the model and we are going to analyse the live tweet.

## TABLE OF CONTENTS

| CHAPTER NO. | TITLE                              | PAGE NO.    |
|-------------|------------------------------------|-------------|
|             | <b>ABSTRACT</b>                    | <b>v</b>    |
|             | <b>LIST OF FIGURES</b>             | <b>viii</b> |
|             | <b>LIST OF ABBREVIATIONS</b>       | <b>ix</b>   |
| <b>1</b>    | <b>INTRODUCTION</b>                | <b>1</b>    |
|             | 1.1 Introduction                   | 1           |
|             | 1.2 Sentiment Analysis             | 1           |
|             | 1.3 Methodology                    | 2           |
| <b>2</b>    | <b>LITERATURE SURVEY</b>           | <b>4</b>    |
| <b>3</b>    | <b>PROJECT DESCRIPTION</b>         | <b>4</b>    |
|             | 3.1 Aim of Project                 | 5           |
|             | 3.2 Existing System                | 5           |
|             | 3.3 Proposed System                | 6           |
|             | 3.4 System Requirements            | 6           |
|             | 3.4.1 Hardware Requirements        | 6           |
|             | 3.4.2 Software Requirements        | 6           |
| <b>4</b>    | <b>MODEL DESCRIPTION</b>           | <b>7</b>    |
|             | 4.1 Data Collection                | 8           |
|             | 4.2 Data Preprocessing             | 8           |
|             | 4.3 Machine Learning Algorithms    | 9           |
|             | 4.3.1 Naive Bayes' Classifier      | 9           |
|             | 4.3.2 Support Vector Machine       | 9           |
|             | 4.4 Training and Testing           | 10          |
|             | 4.4.1 Creation of Bag of Model     | 10          |
|             | 4.4.2 Creation of Predictive Model | 10          |
|             | 4.4.3 Testing Process              | 11          |
|             | 4.5 Data Visualization             | 11          |

|          |                               |           |
|----------|-------------------------------|-----------|
| <b>5</b> | <b>SYSTEM DESIGN</b>          | <b>12</b> |
|          | 5.1 UML Diagram               | 12        |
|          | 5.2 Sequence Diagram          | 13        |
|          | 5.3 Use Case Diagram          | 13        |
|          | 5.4 Activity                  | 15        |
|          | 5.5 Collaboration Diagram     | 18        |
| <b>6</b> | <b>TESTING</b>                | <b>19</b> |
|          | 6.1 System Testing            | 19        |
|          | 6.2 Test Data and Output      | 19        |
|          | 6.2.1 Unit Testing            | 19        |
|          | 6.2.2 Functional Testing      | 20        |
|          | 6.2.3 Performance Testing     | 20        |
|          | 6.2.4 Stress Testing          | 20        |
|          | 6.2.5 Structured Testing      | 20        |
|          | 6.2.6 Integration Testing     | 21        |
|          | 6.3 Testing Techniques        | 21        |
|          | 6.3.1 Testing                 | 21        |
|          | 6.3.2 White Box Testing       | 22        |
|          | 6.3.3 Black Box Testing       | 23        |
|          | 6.3.4 Software Testing        | 23        |
|          | 6.3.5 Integration Testing     | 24        |
|          | 6.3.6 Program Testing         | 24        |
|          | 6.3.7 Security Testing        | 24        |
|          | 6.3.8 Validation Testing      | 25        |
|          | 6.3.9 User Acceptance Testing | 25        |
| <b>7</b> | <b>RESULT AND DISCUSSION</b>  | <b>26</b> |
|          | 7.1 Word Cloud                | 27        |
|          | 7.2 Login Page                | 28        |
|          | 7.3 Analysis of Tweets        | 28        |
| <b>8</b> | <b>CONCLUSION</b>             | <b>31</b> |

|                   |           |
|-------------------|-----------|
| <b>REFERENCES</b> | <b>32</b> |
| <b>APPENDIX</b>   |           |
| a. Source Code    | 35        |
| b. Paper          | 56        |
| c. Plagiarism     | 62        |

## LIST OF FIGURES

| <b>FIGURE NO.</b> | <b>FIGURE NAME</b>         | <b>PAGE NO.</b> |
|-------------------|----------------------------|-----------------|
| 4.1               | Architecture diagram       | 10              |
| 5.1               | Sequence diagram           | 11              |
| 5.2               | Use case diagram           | 16              |
| 5.3               | Activity diagram           | 21              |
| 5.4               | Collaboration diagram      | 22              |
| 7.1               | Word cloud for anti-social | 24              |
| 7.2               | Word cloud for study load  | 26              |
| 7.3               | Word cloud for depression  | 27              |
| 7.4               | Login page                 | 28              |
| 7.5               | Entered tweet              | 29              |

## LIST OF ABBREVIATIONS

| ABBREVIATION | FULL FORM                   |
|--------------|-----------------------------|
| NLP          | Natural language processing |
| SVM          | Support vector machine      |
| RBF          | Radial basis function       |
| UML          | Unified modelling language  |
| I/O          | Input output                |
| ML           | Machine learning            |
| OAUTH        | Open authorization          |

# **CHAPTER-1**

## **INTRODUCTION**

### **INTRO DUCTI ON**

The world is seeing a paradigm shift the way we conduct our daily activities amidst ongoing corona virus (COVID-19) pandemic - be it online learning, the way we socialize, interact, conduct businesses or do shopping. Such global Catastrophes have a direct effect on our social life; however, not all ages react and respond in the same way given a crisis. Even under normal circumstances, research suggests that people across different age reason differently. Due to this, people are experience a wide range of emotions.

Emotion is a mental disorder which may lead to suicide if not cured at early curable stages. So it is very important to predict depression as soon as possible. Many people in the world are suffering from depression in their day to day life. Depression is of different types and can be predicted by different ways. In this paper, study about all the techniques which are used to predict depression and their relative study about techniques, methods, algorithms used to predict depression is done.

### **SENTIMENT ANALYSIS**

Sentiment analysis is a hot topic that's been on research for decades, which intends to find the nature of text and classifies into positive, negative and neutral. Today's advanced technologies provide lot of tools for efficient retrieval, analysis and classification of text data into different classes. In today's digital world lot of data is available for sentiment analysis of both image and text through social media networks like Face book, twitter etc., but using data from some social media networks raises concern of privacy, thus, twitter is the most suitable social media network for getting enough data and also to protect us from privacy laws. This proposition aims to apply natural language processing on Twitter feeds for conducting emotion analysis focusing on depression. Individual tweets are classified as positive or negative, based on a curated word-list to detect depression tendencies. Numerous machine learning algorithms are available for the classification of the tweets; some of them are Naïve Bayes, Support vector machine algorithms, Random forest algorithms etc.

## **METHODOLOGY**

The documented experiments show better result for Naïve Bayes and support vector machine algorithms, the study is limited to these algorithms. The performance of the algorithms can be measured over different metrics like accuracy, precision, recall and confusion matrix. The Determination of the sentiment of an entire document is called as coarse level and fine level deals with attribute level sentiment analysis. In this, study about all the techniques which are used to predict depression and their relative study about techniques, methods, algorithms used to predict depression is done.

## **CHAPTER-2**

## LITERATURE SURVEY

The author [1] in this paper mainly discusses about two-part process ensures we can differentiate between external input and observed data patterns. The conceptual framework aims to support a variety of perspectives and can be used to design and build applications that specifically focus on the dynamics of collective human reaction. His influence of the press on the reactions to an event, making clear those social media reactions cannot be considered as purely a function of a given event, but rather discourse conducted through a multitude of media around an event. An individual reaction is a single reaction from one actor to one event that cannot be further differentiated in a meaningful way.

The author [2] explains that the data and models to predict students' progress and performance which will be helpful in understanding learning behaviors and design suitable intervention strategies. This also motivates the students to use internet to write feedbacks about the course which they study in every semester and discuss with their friends through chat rooms, mails and social media sites. Analyzing such information improves monitoring of students performance and their learning outcome. A methodological framework for extracting the insights from students' opinion that is posted on social media websites.

The author [3] speaks about the frequency of tweets was associated with the number of infected individuals for the early stages of the COVID-19 pandemic. While tweets with misinformation and societal prejudice were present, tweets were also significantly used to disseminate valuable public health information.

The author [4] discuss about what students like and don't like most about a course, professor, or teaching methodology can be of great importance for the respective institutions. While this task represents a unique NLP challenge, many studies have proposed different approaches to tackle the problem.

The author [5] explains one topic specific to U.S. cases would start to uptick immediately after live White House Coronavirus Task Force briefings, implying that many Twitter users are paying attention to government announcements. He contribute machine learning methods not previously reported in the Covid19 Twitter literature.

The author [6] speaks about the broadcast model of one-to-many dissemination dominated. We discussed the role of different user types in the diffusion process. A few influential and hidden influential users played the key role in successful diffusion of Ebola-related messages.

The author [7] explains about strong correlation exists between rise/fall in stock prices of a company to the public opinions or emotions about that company expressed on twitter through tweets. The main contribution of our work is the development of a sentiment analyzer that can judge the type of sentiment present in the tweet

The author [8] discuss that the company should make a profit when the prediction of stocks future price could be successfully done. The Twitter is one of the most useful social media that can help to predict public mood. The predicted mood and previous day value are useful for predicting stock market price

The author [9] speaks about a comprehensive analysis of the spread of Ebola-related information. We have analyzed how spatial, temporal and social properties were related to propagation of those tweets.

The author [10] explains about the first studies to report successful creation of an automated content classification tool to analyze Zika-related tweets specifically in the area of epidemiology. Such a system will help advance the field's technological and methodological capabilities to harness social media sources for disease surveillance research.

## **CHAPTER-3**

### **PROJECT DESCRIPTION**

#### **AIM AND SCOPE**

The objective of this project is to design and develop an application for analysis of student activity tweets. In project we take the student's data and we train the model and we are going to analyze the live tweet.

## **EXISTING SYSTEM**

The existing system uses sentiment analysis technique to identify the sentiment of the tweet. However this is not sufficient to identify the anti-social messages. The sentiment analysis might give the indication of the emotion related to positive or negative. But it is important to classify whether the tweets considered the student activities.

## **PROBLEM STATEMENT:**

- Large volumes of tweets to process.
- Sentiment analysis will not cover the communal student activities.

## **PROPOSED SYSTEM**

We propose a machine learning based classification algorithm to analyze the live tweet. In this system we are using a classifier to identify the words used in the tweet. The algorithm takes specific keywords list as input and the tweet to be identified as another input. The results were plotted in a graph.

## **SYSTEM REQUIREMENTS**

### ***Hardware Requirements***

- Hard Disk : 500GB and Above
- RAM : 4GB and Above
- Processor : I3 and Above

### ***Software Requirements***

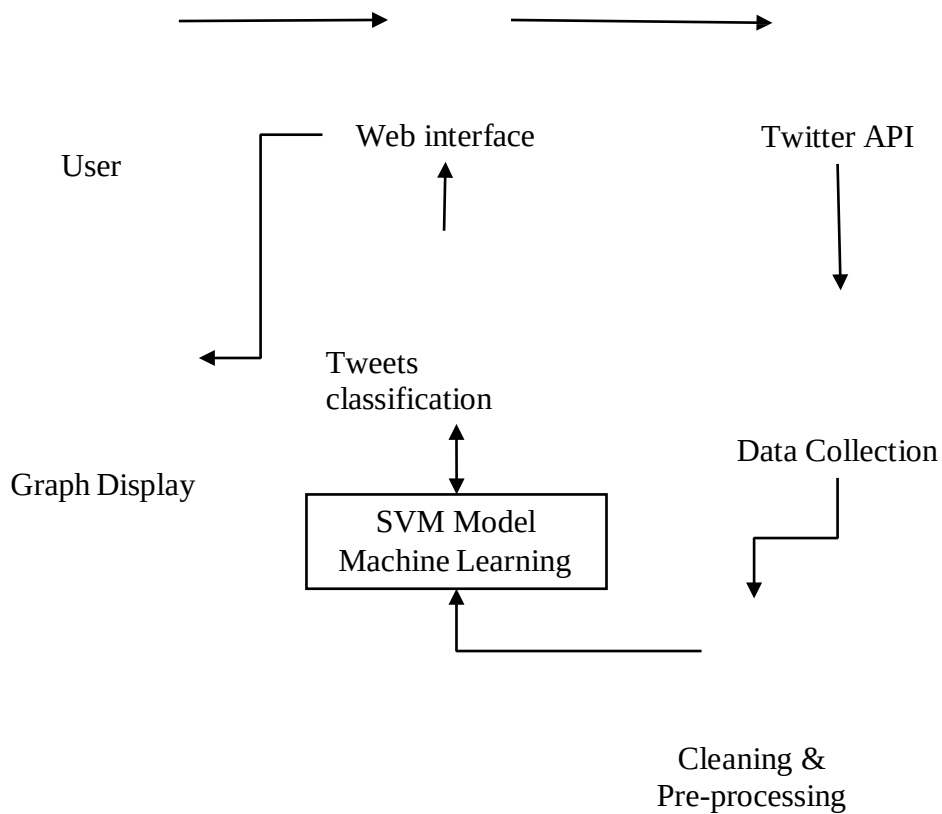
- Operating System : Windows 7 , 8, 10 (64 bit)
- Software : Python 3.7
- Tools : Anaconda (Jupyter Notebook)
- Language : Python

## **CHAPTER- 4**

### **MODULE DESCRIPTION**

- Architecture Diagram

- Data Collection
- Data Pre-processing
- Machine Learning Algorithms
- Training & Testing Process
- Data Visualization



**Fig:4.1 Architecture diagram**

#### **DATA COLLECTION:**

A Python API is used to access Twitter server for searching tweets. Generating Access Key and consumer key from Twitter website to access tweets. Initiating OAuth (Open Authorization) with the access key to login to twitter server for tweet search.

OAuth allows the python to access the twitter server without exposing the username and password.

## **DATA PREPROCESSING:**

The majority of the tweets usually can be divided into 3 parts, not specifically in the same order.

- The first part contains the people whom the tweet is intended to or who is it referencing. This is usually denoted by “@”.

Example: @Hickman.

- The second part contains the actual message. This is what the user actually wants to convey.

- The third part contains the hash tag denoted by “#”. This is usually to categorize the tweets. This can also be used by others to find tweets related to the particular content.

Example: #SupportWHO.

The tweets are pre-processed to filter the first part and third part since they do not hold very less to zero significance in sentiment analysis, later, the remaining message part of the tweet is pre-processed further for obtaining useful keywords which accounts much significance in identifying the emotions.

## **MACHINE LEARNING ALGORITHMS**

### ***NAIVE BAYES CLASSIFIER***

Naive Bayes Classifier implements Bayes theorem with a solid independence assumption, particularly, independent feature model. Bayes Theorem works on conditional probability which finds out the probability of an event given that some other event has already occurred. It predicts the conditional probability of a class given the set of evidences and finds the most likely class based on the highest one. A naive Bayes classifier is a famous and popular technique because it is very fast approach and gives a high accuracy.

### ***SUPPORT VECTOR MACHINE.***

Support Vector Machine (SVM) is a supervised learning algorithm that analyses the data and recognizes patterns. Given an input set, SVM classifies them as one or the other of two categories. SVM can deal with both linear and non-linear classification. With kernel trick, it can efficiently perform non-linear classification. It does so by mapping the input set to high dimensional feature space.

The types of kernel include polynomial, Gaussian radial basis function (RBF), Laplace RBF kernel, Hyperbolic tangent kernel and Sigmoidal kernel. Construction of hyper plane is employed by SVM for the classification. The goal of a text classification system in this paper is to determine whether a given tweet belongs to a set of predefined categories. An optimal SVM algorithm for text classification does this via multiple optimal strategies

## **TRAINING & TESTING PROCESS**

## ***CREATION OF BAG OF WORDS MODEL***

The bag-of-words method is a simplifying representation used in natural language processing and information retrieval. In this model, a text like tweets is represented as the bag of its words, disregarding grammar and even word order but keeping multiplicity.

The bag-of-words model is commonly used in methods of document classification where the (frequency of) occurrence of each word is used as a feature for training a classifier.

Ex. Joey likes food Joey likes games.

The bag of words will be ["Joey":2," likes":2," food":1," games":1].

The Bag-of-words model is mainly used as a tool of feature generation. After transforming the text into a "bag of words", we can calculate various measures to characterize the text. The most common type of characteristics, or features calculated from the Bag-of-words model is term frequency, namely, the number of times a term appears in the text.

## ***CREATION OF PREDICTIVE MODEL***

Two Predictive models are created using naïve Bayes and support vector machine algorithms which were discussed afore. The data file will have two columns with one column representing either the tweet categories and other will contain the actual tweet.

The column with tweets is pre-processed. The pre-processed data is used to create bag words model which will be used as training data to build the predictive models.

## ***TESTING PROCESS***

In testing phase, the 30% of data which was split randomly from the dataset is tested on the predictive model. The test data is pre-processed and classified either.

### **DATA VISUALIZATION:**

The trained model can be stored as a Python object so that one need not train each and every time we need to predict. The accuracy of the trained model is calculated before prediction. We use data visualization to present the result in a visualized manner.

## **CHAPTER-5 SYSTEM DESIGN**

## **UML DIAGRAM**

Unified Modeling language (UML) is a standardized modeling language enabling developers to specify, visualize, construct and document artifacts of a software system. Thus, UML makes these artifacts scalable, secure and robust in execution. UML is an important aspect involved in object-oriented software development

### **Characteristics of UML**

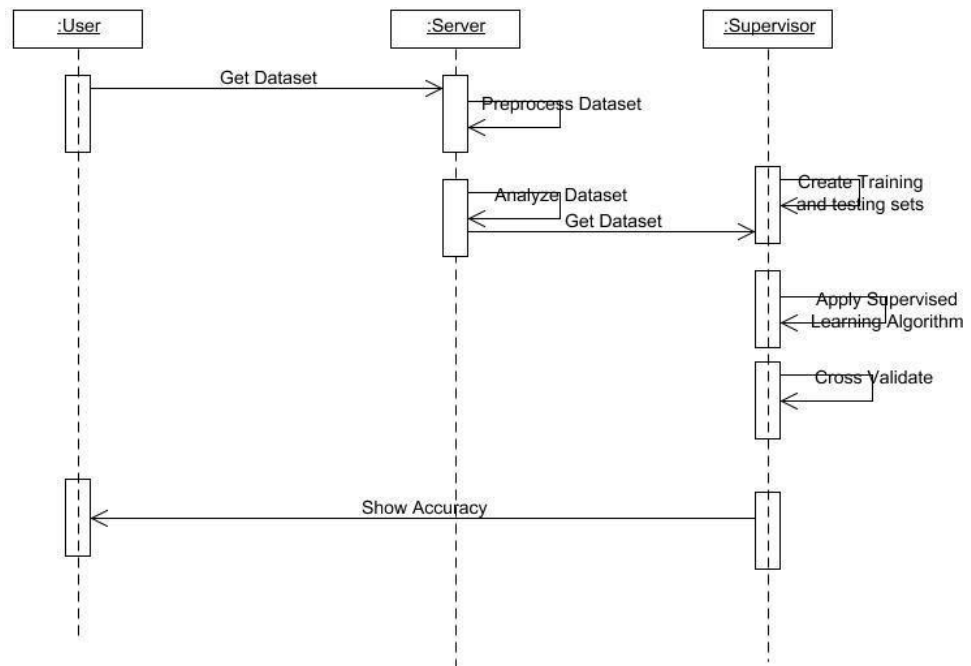
1. It is a generalized modeling language.
2. It is different from software programming languages such as Python, C, C++, etc.
3. It is a pictorial language which can be used to generate powerful modeling elements.
4. It is related to object-oriented designs and analysis.
5. It has unlimited applications even outside the software industry. It can be used to visualize the workflow of a factory.

### **Five Rules for Better UML Diagrams**

- To avoid large diagrams with too many items.
- Avoid any two lines in your diagram crossing each other.
- Lines in a diagram should go only horizontal or vertical with only right angles.
- Parent elements are higher than the child elements in generalization or realization hierarchies.
- Diagrams should be nice and clean

### **Sequence Diagram:**

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



**Fig 5.1 Sequence diagram**

### Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems.

This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

### USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

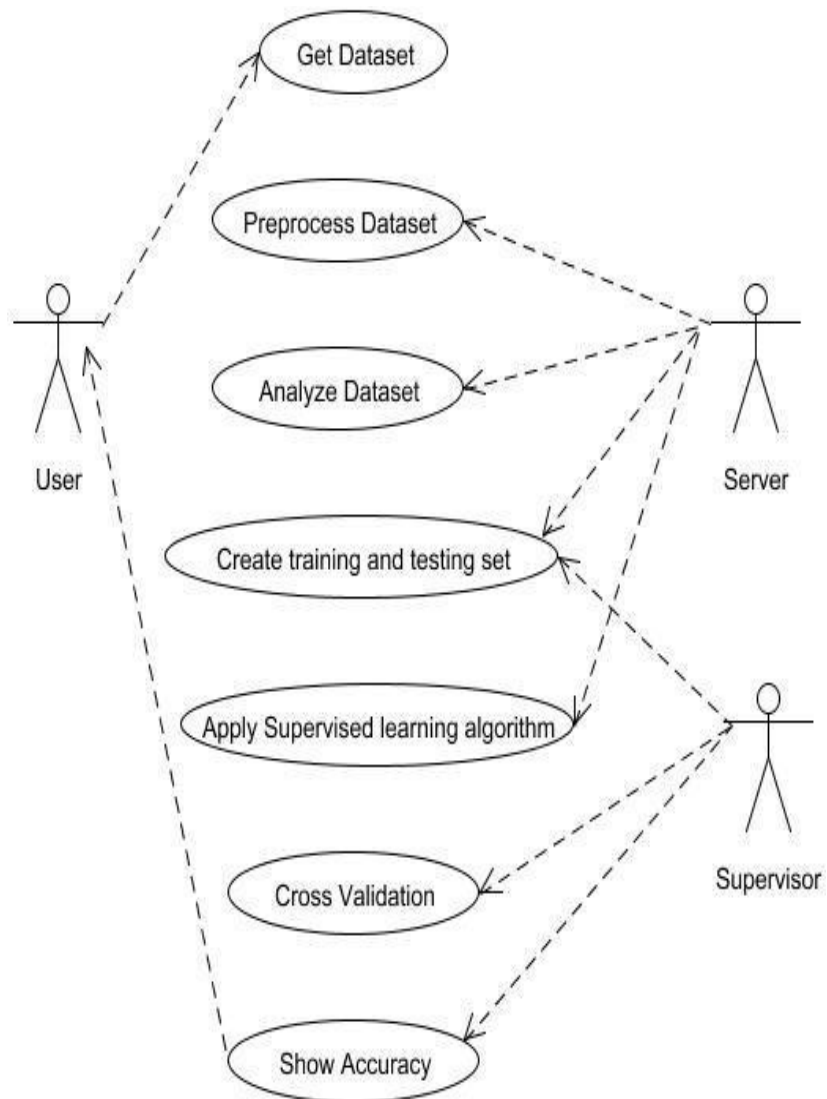
Use case diagram consists of two parts:

**Use case:**

A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

**Actor:**

An actor is a person, organization or external system that plays a role in one or more interaction with the system.



**Fig: 5.2 USECASE DIAGRAM**

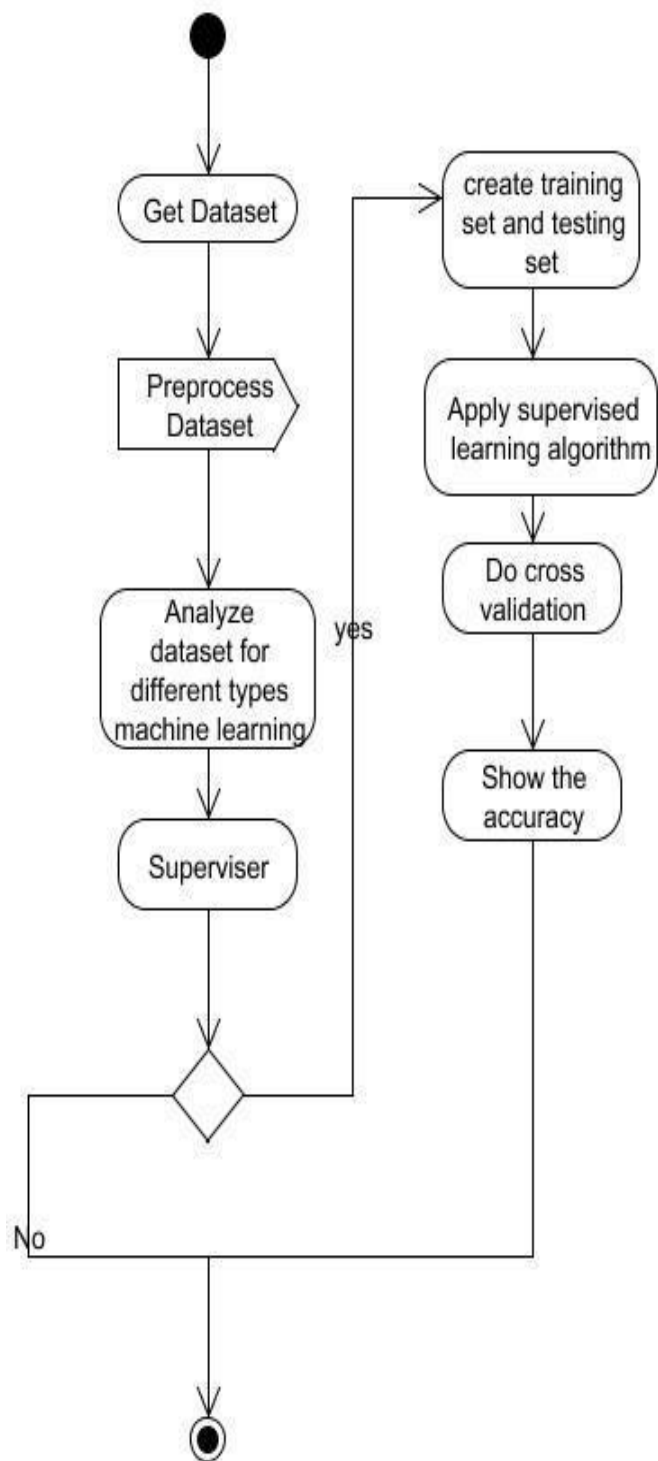
#### **ACTIVITY DIAGRAM:**

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.

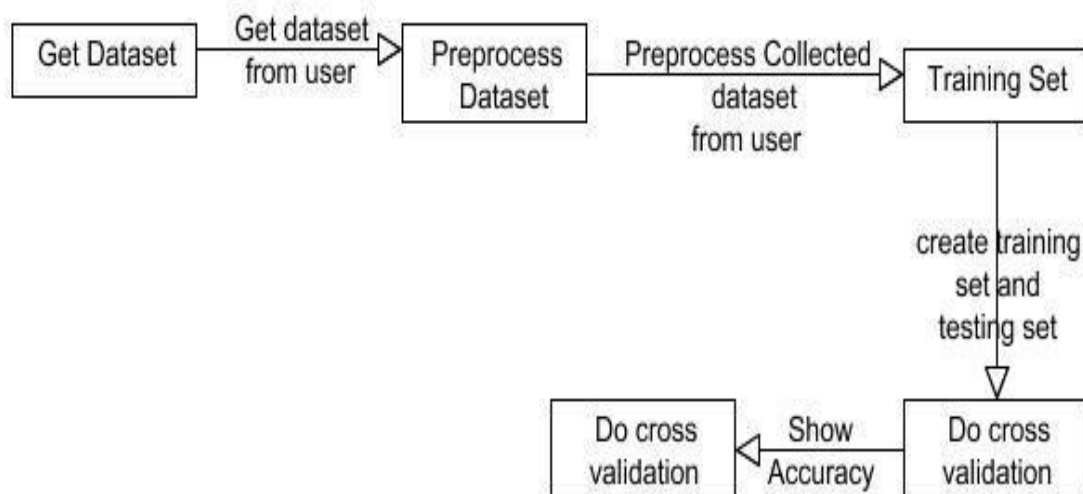
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.



**Fig:5.3 ACTIVITY DIAGRAM**

### Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.



**Fig:5.4 collaboration diagram**

## **CHAPTER-6**

### **TESTING IMPLEMENTATION**

#### **SYSTEM TESTING**

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

#### **TEST DATA AND OUTPUT**

##### ***UNIT TESTING***

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

##### ***FUNCTIONAL TESTS***

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

### ***PERFORMANCE TEST***

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

### ***STRESS TEST***

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

### ***STRUCTURED TEST***

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition, I/O errors, buffer problems and textual errors in output information

## ***INTEGRATION TESTING***

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

## **TESTING TECHNIQUES / TESTING STRATERGIES**

### ***TESTING***

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding a yet undiscovered error. A successful test is one that uncovers a yet undiscovered error.

### ***WHITE BOX TESTING***

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

### ***BLACK BOX TESTING***

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

### ***SOFTWARE TESTING STRATEGIES:***

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

### ***INTEGRATION TESTING:***

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

### ***PROGRAM TESTING:***

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

### ***SECURITY TESTING:***

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

### ***VALIDATION TESTING***

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- The function or performance characteristics confirm to specifications and are accepted.
- A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic.

### ***USER ACCEPTANCE TESTING***

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design.
- Output screen design.

## **CHAPTER-7**

### **RESULT AND DISCUSSION**

## WORD CLOUD

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.



**FIG:7.1 WORD CLOUD FOR ANTI-SOCIAL**



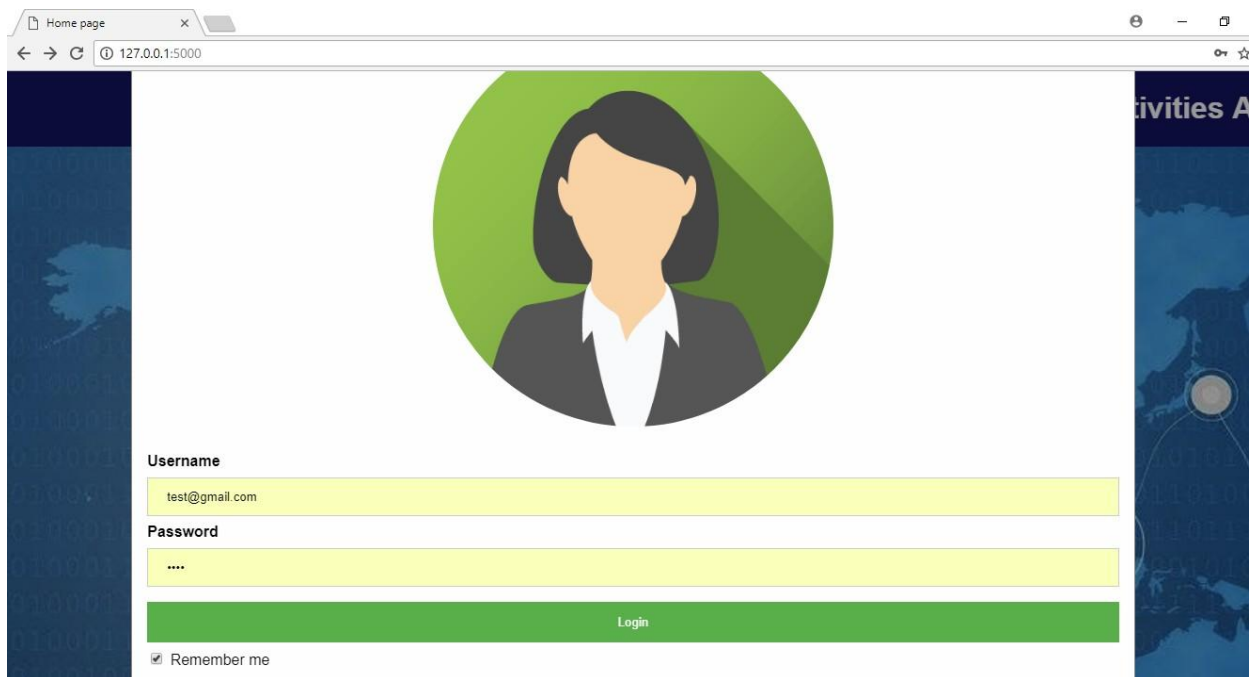
**FIG:7.2 WORD CLOUD FOR STUDY-LOAD**



**FIG:7.3 WORD CLOUD FOR DREPRESSION**

## LOGIN PAGE:

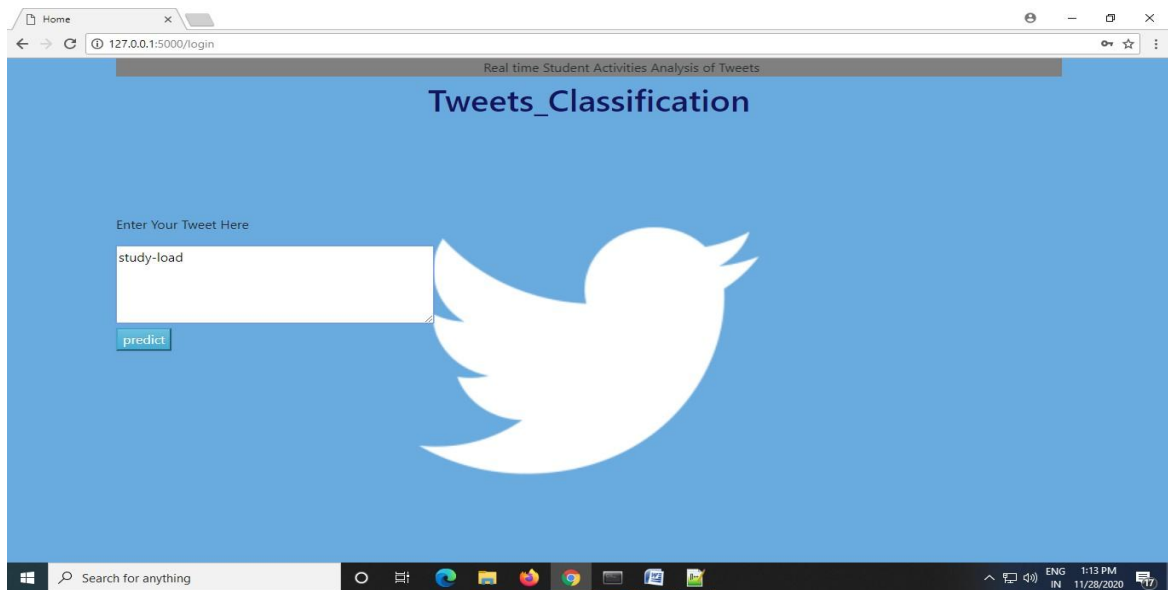
Login page consists of username and password. First we need to register and then we can login into the page. Then we can connect to twitter server and analyze the live tweet.



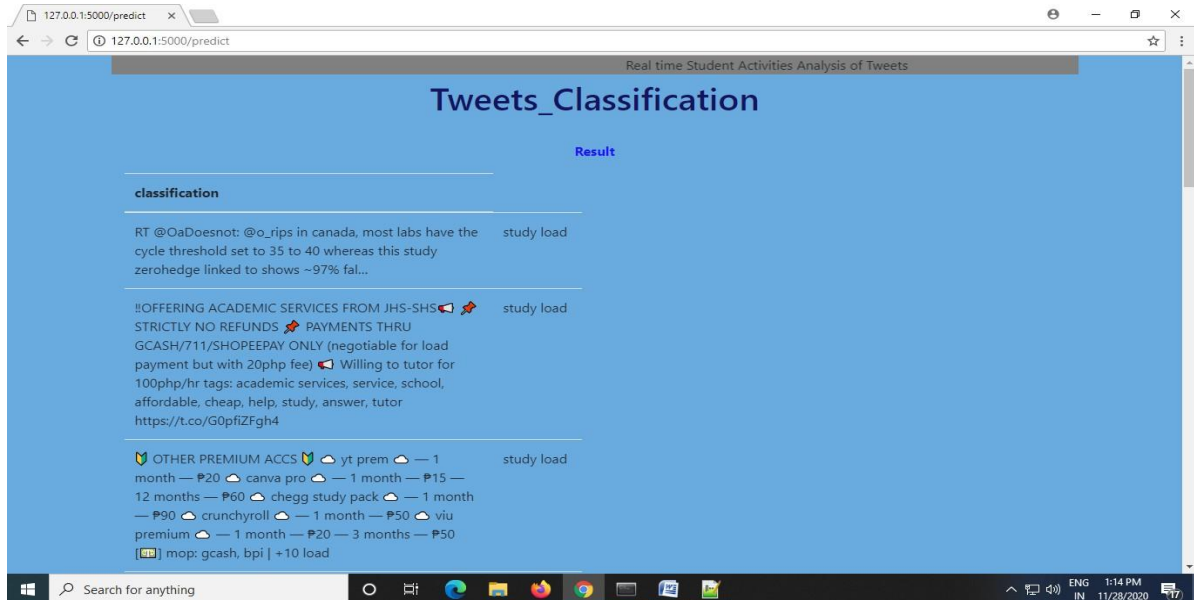
**FIG:7.4 LOGIN PAGE**

## ANALYSIS OF TWEETS

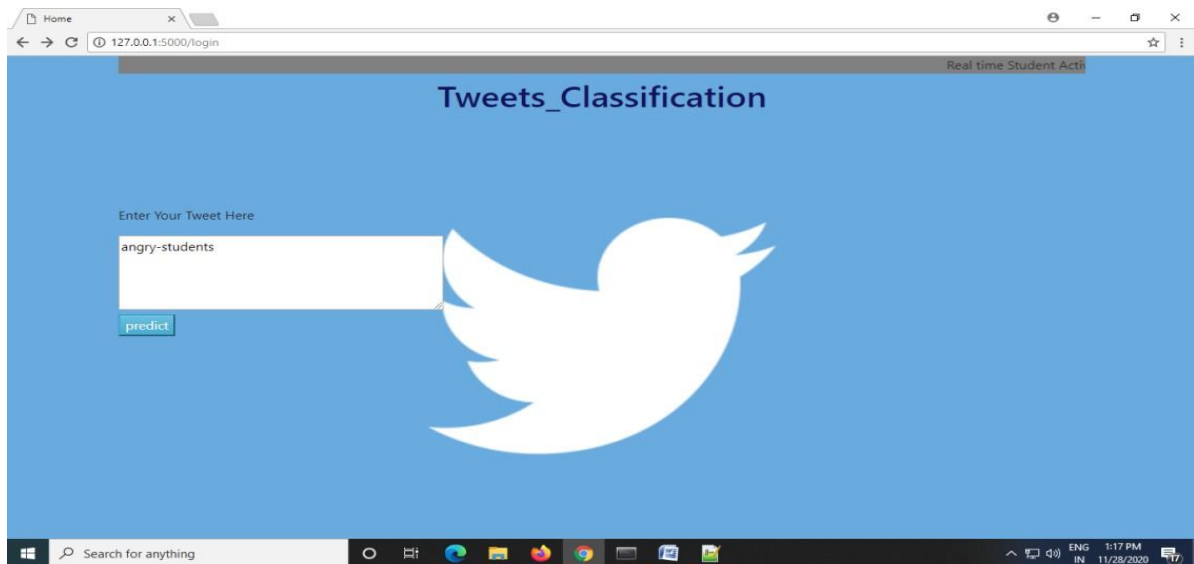
Twitter is going to search for the keyword we entered and we are going to classify the live tweets based on our data model.



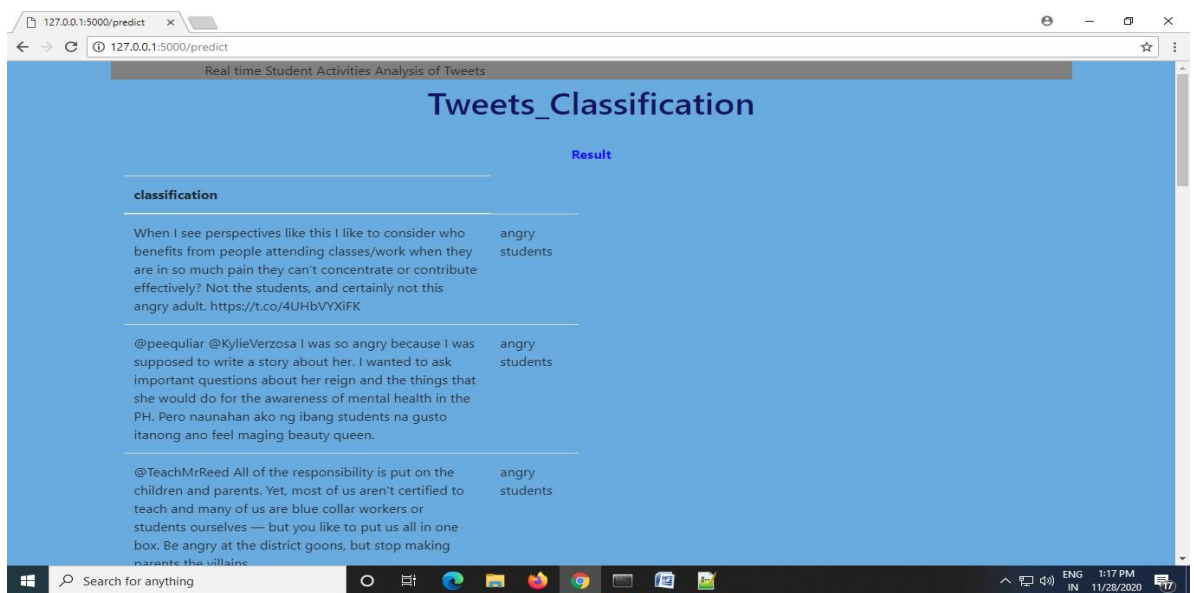
**FIG:7.5 SEARCH PAGE**



**FIG:7.6 RESULT PAGE**



**FIG:7.7 SEARCH PAGE**



**FIG:7.8 RESULT PAGE**

## **CHAPTER- 8**

### **CONCLUSION**

This article aimed to the correlation between sentiments and emotions of the students amidst coronavirus (COVID-19) outbreak from their tweets. Based on the above observations, we can conclude that for the context of depression detection Multinomial Naïve Bayes algorithm has performed well since the precision score is higher for Multinomial Naïve Bayes.

We also observed anomalies like conjoined words, separation of which will increase the efficiency of the model. Though better results have been achieved in the previous proposals, those results are over a limited number of tweets and since data is dynamic, it requires dynamic methods to tackle all the obstacles that arise in the sentiment analysis in the future.

## REFERENCES

- [1] Dunkel, Alexander ; Andrienko, Gennady "A conceptual framework for studying collective reactions to events in location-based social media", International Journal of Geographical Information Science Volume 33, 2019.
- [2] A. Jenifer Jothi Mary, S. Santiago, Dr. L. Arockiam "A Methodological Framework to Identify the Students' Opinion using Aspect based Sentiment Analysis", International Journal of Engineering Research & Technology (IJERT), Published by : Vol. 5 Issue 02, February-2016.
- [3] Richard J. Medford, Sameh N. Saleh, Andrew Sumarsono, Trish M. Perl, Christoph U. Lehmann, "An *"Infodemic"*: Leveraging High-Volume Twitter Data to Understand Early Public Sentiment for the COVID-19 Outbreak" *Open Forum Infectious Diseases*, Volume 7, Issue 7, July 2020.
- [4] Rinor Hajrizi, Krenare Pireva Nuçi, "Aspect-based sentiment analysis in education domain", October 6, 2020.
- [5] Catherine Ordun, Sanjay Purushotham, Edward Raff, "Exploratory Analysis of Covid-19 Tweets using Topic Modeling, UMAP, and DiGraph", May 2020.
- [6] Rupawari Jadhav, M. S. Wakode, "Survey : Sentiment Analysis of Twitter Data for Stock Market Prediction", International Journal of Advanced Research in Computer and Communication Engineering, March 2017.
- [7] Hai Liang, Issac Chun-Hai Fung, "How did Ebola information spread on twitter: broadcasting or viral spreading?", BMC public Health volume 19, Article number: 438(2019)
- [8] Venkata Sasank Pagolu, Kamal Nayan Reddy Challa, Ganapati Panda, Babita Majhi, "Sentiment Analysis of Twitter Data for Predicting Stock Market Movements" International conference on Signal Processing, Communication, Power and Embedded System (SCOPEs)-2016.
- [9] Thanh Tran, Kyumin Lee, "Understanding Citizen Reactions and Ebola-Related Information Propagation on Social Media", 2016.
- [10] Michele Miller MS, Dr. Tanvi Banerjee, "What Are People Tweeting About Zika? An Exploratory Study Concerning Its Symptoms, Treatment, Transmission, and Prevention", 2017.

## APPENDIX

### SOURCE CODE

```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
from matplotlib.cm import rainbow
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
sleep=pd.read_csv("dataset/sleep.csv")
load=pd.read_csv("dataset/load.csv")
depression=pd.read_csv("dataset/depression.csv")
angry=pd.read_csv("dataset/angry.csv")
antisocial=pd.read_csv("dataset/antisocial.csv")
print(load)
dataset = pd.concat([sleep, load, depression,angry,antisocial])
print(dataset)
import nltk
import random
from nltk.classify.scikitlearn import SklearnClassifier
import pickle
from nltk.classify import ClassifierI
from statistics import mode
from nltk.tokenize import word_tokenize
class VoteClassifier(ClassifierI):
    def __init__(self, *classifiers):
        self._classifiers = classifiers
```

```

def classify(self, features):
    votes = []
    for c in self._classifiers:
        v = c.classify(features)
        votes.append(v)
    return mode(votes)

def confidence(self, features):
    votes = []
    for c in self._classifiers:
        v = c.classify(features)
        votes.append(v)
    choice_votes = votes.count(mode(votes))
    conf = choice_votes / len(votes)
    return conf

sleep = open("dataset/sleep.csv", "r", encoding='utf-8').read()
load = open("dataset/load.csv", "r", encoding='utf-8').read()
depression = open("dataset/depression.csv", "r", encoding='utf-8').read()
antisocial = open("dataset/antisocial.csv", "r", encoding='utf-8').read()
angry = open("dataset/angry.csv", "r", encoding='utf-8').read()

all_words = []
documents = []
allowed_word_types = ["J"]
for p in sleep.split('\n'):
    documents.append((p, "sleep deprived"))
    words = word_tokenize(p)
    pos = nltk.pos_tag(words)

```

```

for w in pos:
    if w[1][0] in allowed_word_types:
        all_words.append(w[0].lower())
#print(all_words)
#print(documents)
classify1=pd.DataFrame(documents)
classify1.to_csv('preprocess/sleep_new.csv',header=None)
from wordcloud import WordCloud,STOPWORDS
df = pd.read_csv("preprocess/sleep_new.csv")
comment_words= ""
stopwords = set(STOPWORDS)
for val in df['tweets']:
    val = str(val)
    tokens = val.split()

    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
for words in tokens:
    comment_words = comment_words + words + ' '
wordcloud = WordCloud(width = 800, height = 800,
    background_color ='white',
    stopwords = stopwords,
    min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

```

```

plt.show()

all_words = []

documents = []

allowed_word_types = ["J"]

for p in load.split('\n'):

    documents.append((p, "study load"))

    words = word_tokenize(p)

    pos = nltk.pos_tag(words)

    for w in pos:

        if w[1][0] in allowed_word_types:

            all_words.append(w[0].lower())

classify2=pd.DataFrame(documents)

classify2.to_csv('preprocess/load_new.csv',header=None)

df = pd.read_csv("preprocess/load_new.csv")

comment_words= ""

stopwords = set(STOPWORDS)

for val in df['tweets']:

    val = str(val)

    tokens = val.split()

    for i in range(len(tokens)):

        tokens[i] = tokens[i].lower()


    for words in tokens:

        comment_words = comment_words + words + ' '

wordcloud = WordCloud(width = 800, height = 800,

                        background_color ='white',

                        stopwords = stopwords,

```

```

        min_font_size = 10).generate(comment_words)

plt.figure(figsize = (8, 8), facecolor = None)

plt.imshow(wordcloud)

plt.axis("off")

plt.tight_layout(pad = 0)

plt.show()

all_words = []

documents = []

allowed_word_types = ["J"]

for p in depression.split('\n'):

    documents.append((p, "depression"))

    words = word_tokenize(p)

    pos = nltk.pos_tag(words)

    for w in pos:

        if w[1][0] in allowed_word_types:

            all_words.append(w[0].lower())

classify3=pd.DataFrame(documents)

classify3.to_csv('preprocess/depression_new.csv',header=None)

df = pd.read_csv("preprocess/depression_new.csv")

comment_words= ""

stopwords = set(STOPWORDS)

for val in df['tweets']:

    val = str(val)

    tokens = val.split()

    for i in range(len(tokens)):

        tokens[i] = tokens[i].lower()

```

```

for words in tokens:

    comment_words = comment_words + words + ' '

wordcloud = WordCloud(width = 800, height = 800,
                        background_color ='white',
                        stopwords = stopwords,
                        min_font_size = 10).generate(comment_words)

plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.show()

all_words = []
documents = []
allowed_word_types = ["J"]
for p in antisocial.split('\n'):

    documents.append((p, "anti social behaviour"))

    words = word_tokenize(p)
    pos = nltk.pos_tag(words)

    for w in pos:

        if w[1][0] in allowed_word_types:

            all_words.append(w[0].lower())

classify4=pd.DataFrame(documents)

classify4.to_csv('preprocess/anti social behaviour_new.csv',header=None)

df = pd.read_csv("preprocess/anti social behaviour_new.csv")

comment_words= ""

stopwords = set(STOPWORDS)

```

```

for val in df['tweets']:

    val = str(val)

    tokens = val.split()

    for i in range(len(tokens)):

        tokens[i] = tokens[i].lower()

    for words in tokens:

        comment_words = comment_words + words + ' '

wordcloud = WordCloud(width = 800, height = 800,

                        background_color = 'white',

                        stopwords = stopwords,

                        min_font_size = 10).generate(comment_words)

plt.figure(figsize = (8, 8), facecolor = None)

plt.imshow(wordcloud)

plt.axis("off")

plt.tight_layout(pad = 0)

plt.show()

all_words = []

documents = []

allowed_word_types = ["J"]

for p in angry.split('\n'):

    documents.append((p, "angry students"))

    words = word_tokenize(p)

    pos = nltk.pos_tag(words)

    for w in pos:

        if w[1][0] in allowed_word_types:

            all_words.append(w[0].lower())

classify5=pd.DataFrame(documents)

```

```

classify5.to_csv('preprocess/angry_students_new.csv',header=None)

df = pd.read_csv("preprocess/angry_students_new.csv")

comment_words= ""

stopwords = set(STOPWORDS)

for val in df['tweets']:

    val = str(val)

    tokens = val.split()

    for i in range(len(tokens)):

        tokens[i] = tokens[i].lower()

        for words in tokens:

            comment_words = comment_words + words + ' '

wordcloud = WordCloud(width = 800, height = 800,

                        background_color ='white',

                        stopwords = stopwords,

                        min_font_size = 10).generate(comment_words)

plt.figure(figsize = (8, 8), facecolor = None)

plt.imshow(wordcloud)

plt.axis("off")

plt.tight_layout(pad = 0)

plt.show()

preprocess_all = pd.concat([classify1,classify2,classify3,classify4,classify5])

preprocess_all.to_csv('combined_preprocess_dataset/preprocess_all.csv')

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.naive_bayes import MultinomialNB

from sklearn.externals import joblib

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

```

```

from sklearn.svm import SVC, LinearSVC, NuSVC

dataset2=pd.read_csv('combined_preprocess_dataset/preprocess_all.csv')

dataset2['label'] = dataset2['1'].map({'sleep deprived': 0, 'depression': 1,'study
load':2,'anti social behaviour':3,'angry students':4})

y = dataset2['1']

X = dataset2['0']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state
= 0)

print(X_train)

print(X_test)

print(y_train)

print(y_test)

rcParams['figure.figsize'] = 10,6

plt.bar(dataset2['1'].unique(), dataset2['1'].value_counts(), color = ['red',
'green','blue','yellow','brown'])

plt.xticks([0,1,2,3,4])

plt.xlabel('students activities')

plt.ylabel('Count')

plt.title('Tweets_Classification')

all_words = []

documents = []

allowed_word_types = ["J"]

for p in sleep.split('\n'):

    documents.append((p, "sleep deprived"))

    words = word_tokenize(p)

    pos = nltk.pos_tag(words)

    for w in pos:

        if w[1][0] in allowed_word_types:

```

```

        all_words.append(w[0].lower())
for p in load.split('\n'):
    documents.append((p, "study load"))
    words = word_tokenize(p)
    pos = nltk.pos_tag(words)
    for w in pos:
        if w[1][0] in allowed_word_types:
            all_words.append(w[0].lower())
for p in depression.split('\n'):
    documents.append((p, "depression"))
    words = word_tokenize(p)
    pos = nltk.pos_tag(words)
    for w in pos:
        if w[1][0] in allowed_word_types:
            all_words.append(w[0].lower())
for p in antisocial.split('\n'):
    documents.append((p, "anti social behaviour"))
    words = word_tokenize(p)
    pos = nltk.pos_tag(words)
    for w in pos:
        if w[1][0] in allowed_word_types:
            all_words.append(w[0].lower())
for p in angry.split('\n'):
    documents.append((p, "angry"))
    words = word_tokenize(p)
    pos = nltk.pos_tag(words)
    for w in pos:

```

```

        if w[1][0] in allowed_word_types:
            all_words.append(w[0].lower())

save_documents = open("Pickle/documents.pickle", "wb")
pickle.dump(documents, save_documents)
save_documents.close()

all_words = nltk.FreqDist(all_words)
word_features = list(all_words.keys())[:5000]

save_word_features = open("Pickle/word_features.pickle", "wb")
pickle.dump(word_features, save_word_features)
save_word_features.close()

def find_features(document):
    words = word_tokenize(document)
    features = {}
    for w in word_features:
        features[w] = (w in words)
    return features

featuresets = [(find_features(rev), category) for (rev, category) in documents]
random.shuffle(featuresets)

print(len(featuresets))

testing_set = featuresets[100:]
training_set = featuresets[:100]

classifier = nltk.NaiveBayesClassifier.train(training_set)

print("Naive Bayes Algo accuracy percent:", (nltk.classify.accuracy(classifier,
testing_set)) * 130)

classifier.show_most_informative_features(15)

save_classifier = open("Pickle/naivebayes.pickle", "wb")
pickle.dump(classifier, save_classifier)

```

```

save_classifier.close()

MNB_classifier = SklearnClassifier(MultinomialNB())

MNB_classifier.train(training_set)

save_classifier = open("Pickle/MNB_classifier.pickle", "wb")

pickle.dump(MNB_classifier, save_classifier)

save_classifier.close()

LinearSVC_classifier = SklearnClassifier(LinearSVC())

LinearSVC_classifier.train(training_set)

print("SVM_classifier accuracy percent:",
      (nltk.classify.accuracy(LinearSVC_classifier, testing_set)) * 135)

save_classifier = open("pickle/SVM_classifier.pickle", "wb")

pickle.dump(LinearSVC_classifier, save_classifier)

save_classifier.close()

import pickle

from nltk.classify import ClassifierI

from statistics import mode

from nltk.tokenize import word_tokenize

class VoteClassifier(ClassifierI):
    def __init__(self, *classifiers):
        self._classifiers = classifiers

    def classify(self, features):
        votes = []

        for c in self._classifiers:
            v = c.classify(features)
            votes.append(v)

            print(v)

        return mode(votes)

```

```

def confidence(self, features):
    votes = []
    for c in self._classifiers:
        v = c.classify(features)
        votes.append(v)
    choice_votes = votes.count(mode(votes))
    conf = choice_votes / len(votes)
    return conf

documents_f = open("pickle/documents.pickle", "rb")
documents = pickle.load(documents_f)
documents_f.close()

word_features5k_f = open("pickle/word_features.pickle", "rb")
word_features = pickle.load(word_features5k_f)
word_features5k_f.close()

def find_features(document):
    words = word_tokenize(document)
    features = {}
    for w in word_features:
        features[w] = (w in words)

    return features

open_file = open("pickle/naivebayes.pickle", "rb")
classifier = pickle.load(open_file)
open_file.close()

open_file = open("pickle/MNB_classifier.pickle", "rb")
MNB_classifier = pickle.load(open_file)

```

```

open_file.close()

save_classifier = open("pickle/SVM_classifier.pickle", "wb")

pickle.dump(LinearSVC_classifier, save_classifier)

save_classifier.close()

voted_classifier = VoteClassifier(
    classifier )

def prediction(text):

    feats = find_features(text)

    return (voted_classifier.classify(feats))

tweet = ['angry','depression','sleep','load','angry','anti-social','depression','College
students are one of the most sleep-deprived populations','I finally get to study the
facets of Finance that I enjoy but the work-load is killing my vibe','Please Retweet:
Major Depressive Episode-A Humbling Experience https://t.co/UrhdnC5y93 Please
Retweet and help to #EndTheStigma! #depression #suicide #MentalIllness
https://t.co/N1wEcKMy1f','sleep']

b=[]

for i in tweet:

    b.append(prediction(i))

c=[]

for i in tweet:

    c.append(i)

import json

import tweepy as tw

import pandas as pd

consumer_key = 'k00YCSU3mWjKL0pemcyMnnK6Q'

consumer_secret =
'LZJdDRzWjYKVKiJzO4Vw2gciYEInnRm6RMSSIRP2ZC3kYo3TO5'

access_key= '1204284374134312960-NukMmknCYWn8tv7ndPFjt57GBh9znU'

access_secret = 'WBk7sJ3DVBwb01XU9xURAFcqiXHDWixAzMBWG0lc78PfW'

```

```

auth = tw.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_key, access_secret)
api = tw.API(auth)
search_words = "depression"
date_since = "2020-01-31"
COLS = ['id', 'timestamp', 'username', 'tweets', 'followers', 'place']
tweets = tw.Cursor(api.search, q =
search_words, lang="en", tweet_mode='extended', since=date_since).items(100)
data = [[tweet.id, tweet.created_at, tweet.user.screen_name, tweet.full_text,
tweet.user.followers_count, tweet.user.location] for tweet in tweets]
tweet_text = pd.DataFrame(data , columns= COLS)
tweet = tweet_text.filter(["tweets"])
print(tweet['tweets'].to_list())
tweet = tweet['tweets'].to_list()
print(len(tweet))
b=[]
for i in tweet:
    b.append(prediction(i))
c=[]
for i in tweet:
    c.append(i)
dict1={"input":c,"output":b}
data3=pd.DataFrame(dict1)
data3
#data3.to_csv('test.csv')
import pandas as pd

```

```

import matplotlib.pyplot as plt

from matplotlib import rcParams
from matplotlib.cm import rainbow

%matplotlib inline

import warnings

warnings.filterwarnings('ignore')

rcParams['figure.figsize'] = 10,6

plt.bar(data3['output'].unique(),data3['output'].value_counts(), color = ['red',
'green','blue','yellow','brown'])

plt.xticks([0,1,2,3,4])

plt.xlabel('student activities')

plt.ylabel('Count')

plt.title('Tweet classification')

```

```

from flask import Flask, render_template, url_for, request
from flask import url_for, redirect, render_template
from flask import Flask, render_template, url_for, request
from flask import Flask ,render_template,request,jsonify,session
from flask import url_for, redirect, render_template
import pandas as pd
import pickle
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.externals import joblib
import json
import tweepy as tw      #extract tweets from the twitter account
import pandas as pd

```

```

import matplotlib.pyplot as plt

from matplotlib import rcParams
from matplotlib.cm import rainbow

#%matplotlib inline

import warnings

warnings.filterwarnings('ignore')

from sklearn.externals import joblib

import sqlite3 as sql

import base64

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])


def home():

    return render_template('index.html')

def validate(username,password):

    con = sql.connect('static/chat.db')

    completion = False

    with con:

        cur = con.cursor()

        cur.execute('SELECT * FROM persons')

        rows = cur.fetchall()

        for row in rows:

            dbuser = row[1]

            dbpass = row[2]

            if dbuser == username:

                completion = (dbpass == password)

```

```

        return completion

@app.route('/login', methods=['GET', 'POST'])
def login():
    error = None

    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        completion = validate(username,password)
        if completion == False:
            error = 'invalid Credentials. please try again.'
        else:
            session['username'] = request.form['username']
            return render_template('home.html')

    return render_template('home.html', error=error)

@app.route('/register', methods = ['GET','POST'])
def register():
    if request.method == 'POST':
        try:
            name = request.form['name']
            username = request.form['username']
            password = request.form['password']
            with sql.connect("static/chat.db") as con:
                cur = con.cursor()

                cur.execute("INSERT INTO persons(name,username,password) VALUES
(?,?,?)",(name,username,password))

                con.commit()

                msg = "Record successfully added"

```

```

except:
    con.rollback()
    msg = "error in insert operation"
finally:
    return render_template("index.html",msg = msg)
    con.close()

return render_template('register.html')

@app.route('/list')
def list():
    con = sql.connect("static/chat.db")
    con.row_factory = sql.Row
    cur = con.cursor()
    cur.execute("select * from persons")
    rows = cur.fetchall();
    return render_template("list.html",rows = rows)

@app.route('/predict', methods=['POST'])
def predict():
    from sklearn.feature_extraction.text import CountVectorizer
    from sklearn.naive_bayes import MultinomialNB
    from sklearn.externals import joblib
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler
    from sklearn.svm import SVC, LinearSVC, NuSVC
    import pickle

    from nltk.classify import ClassifierI
    from statistics import mode
    from nltk.tokenize import word_tokenize

```

```

class VoteClassifier(ClassifierI):

    def __init__(self, *classifiers):
        self._classifiers = classifiers

    def classify(self, features):
        votes = []

        for c in self._classifiers:
            v = c.classify(features)
            votes.append(v)
            print(v)

        return mode(votes)

    def confidence(self, features):
        votes = []

        for c in self._classifiers:
            v = c.classify(features)
            votes.append(v)

        choice_votes = votes.count(mode(votes))
        conf = choice_votes / len(votes)
        return conf

```

```

documents_f = open("pickle/documents.pickle", "rb")
documents = pickle.load(documents_f)
documents_f.close()

word_features5k_f = open("pickle/word_features.pickle", "rb")
word_features = pickle.load(word_features5k_f)

```

```

word_features5k_f.close()

def find_features(document):
    words = word_tokenize(document)
    features = {}
    for w in word_features:
        features[w] = (w in words)

    return features

open_file = open("pickle/naivebayes.pickle", "rb")
classifier = pickle.load(open_file)
open_file.close()

open_file = open("pickle/MNB_classifier.pickle", "rb")
MNB_classifier = pickle.load(open_file)
open_file.close()

voted_classifier = VoteClassifier(classifier)

def prediction(text):
    feats = find_features(text)
    return (voted_classifier.classify(feats))

o=[]

if request.method == 'POST':

    search_words = request.form['message']
    print(search_words)

    consumer_key = 'k00YCSU3mWjKL0pemcyMnnK6Q'
    consumer_secret =
'LZJdDRzWjYKVKiJzO4Vw2gciYElnnRm6RMSSIRP2ZC3kYo3TO5'

```

```

access_key= '1204284374134312960-NukMmknCYWn8tv7ndPFjt57GBh9znU'
access_secret = 'WBk7sJ3DVBwb01XU9xURAFcqiXHDWlxAzMBWG0lc78PfW'
auth = tw.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_key, access_secret)
api = tw.API(auth)

date_since = "2020-01-31"

COLS = [ 'id', 'timestamp', 'username', 'tweets', 'followers', 'place']

tweets = tw.Cursor(api.search, q =
search_words, lang="en", tweet_mode='extended', since=date_since).items(100)

data = [[tweet.id, tweet.created_at, tweet.user.screen_name, tweet.full_text,
tweet.user.followers_count, tweet.user.location] for tweet in tweets]

tweet_text = pd.DataFrame(data , columns= COLS

tweet = tweet_text

tweet123 = tweet_text.filter(["username"])

tweet = tweet['tweets'].to_list()

tweet123 = tweet_text['id'].to_list()

#print(len(tweet))

l=[]

lst=[]

b=[]

for i in tweet:

    b.append(prediction(i))

print(b)

dict1={"output":b}

data3=pd.DataFrame(dict1)

```

```
for i in b:
    lst.append(i)
for j in tweet:
    l.append(j)

return render_template('result1.html',tweets=l, prediction=lst,display=True)
if __name__ == '__main__':
    app.run(debug=True)
#df['label'] = df['v1']
```

paper

*by* Paper  
Paper

---

**Submission date:** 05-Apr-2021 06:01PM (UTC-0400)

**Submission ID:** 1551315859 **File**

**name:** paper.docx (238.81K)

**Word count:** 2040

**Character count:** 11431

## ENHANCED SENTIMENT ANALYSIS ON STUDENT'S COVID-19 RELATED TWEETS

S.JITENDRA KALYAN [1], CHUNDU SANDEEP [2], ANANDHI T<sup>1</sup> [3]

[1][2] UG Student, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

[3] Associate professor, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

**ABSTRACT:** The COVID-19 pandemic has raised understanding of the mental health of those that have been affected. Epidemics also known to intensify or develop new stressors, like fear and worries regarding oneself or loved ones, restrictions on physical movement and social relationships due to quarantine, and sudden and unexpected lifestyle changes. In the last few months, social media has also been flooded with posts expressing both supportive and negative thoughts regarding COVID-19, the pandemic, the lockdown, and hash tags. Students responded to the pandemic in a number of ways. The purpose of this work is to examine students' reaction to the new coronavirus and build a prototype to interpret tweets into different activities. Bayes, Naive Models for estimating sentiment polarity and emotions collected from tweets using help vector machine algorithms.

**Keywords:** sentiment analysis, Machine learning, emotions, twitter.

### I INTRODUCTION

In the midst of the on-going coronavirus (COVID-19) pandemic, the world is witnessing a paradigm shift in the way we manage our everyday activities, whether that's online learning, social interaction, communicating, conducting business, or shopping. Global events have a strong influence on our social lives. Individuals use social media platforms to share ideas, views, and reactions with others to cope with and react to acute crises, and these interaction networks play a key role during these occasions. As a result, the purpose of this analysis will be on the emotions expressed by students on social media. To this end, thousands of students' tweets were analyzed to see how they behaved and responded to the coronavirus.

A modified version of the publicly available tweets dataset was also used. It is proposed that a new model for sentiment and emotion analysis be created.

Since modern culture has made life so busy, people are more prone to mental problems such as depression and anxiety. Many aspects of human life may be affected by depression. Determining and avoiding depression is incredibly difficult, and it has been a hot subject of study for the past decade. Stresses, work schedules, and a number of other factors have an effect on everyday life and can contribute to depression. Researchers consider depression to be a type of mental cancer. Loss of interest in hobbies, suicidal thoughts, feelings of worthlessness or hopelessness, and a worsened capacity to think and focus are all signs of depression. Sentiment analysis is a hot topic that has been the focus of decades of study, with the goal of determining the meaning of text and categorizing it as positive, negative, or neutral. Today's advanced technologies include a variety of methods for efficiently extracting, analyzing, and categorizing text data through various categories. In today's digital world, social networks such as Facebook, Twitter, and

others offer a wealth of data for sentiment analysis of image and text; However, accessing data from some online platforms raises security concerns; as a result, Twitter is the ideal social media platform for gathering sufficient data while staying consistent with privacy laws.

This plan aims to use natural language processing to perform emotion analysis on Twitter feeds, with an emphasis on depression. Based on a curated word list designed to identify depression, specific tweets are categorized as positive or negative. For the classification of tweets, a number of machine learning algorithms are available, like Naive Bayes, Support vector machine algorithms, Random forest algorithms, and others.

## II RELATED WORK

*Dunkel et al.[1]* We may explore variations and correlations in collective interpretation of meaning over space and time for a thematically well-defined class of occurrence, like storms. Other cases, such as political processes like Elections, can have hugely complicated spatio-temporal signatures that can be broken down into a series of discrete events.

*A.Jenifer Jothi Mary et al.[2]*

Opinions, it is obvious, will assist the teaching community in identifying students' views and emotions toward learning. Rather than taking feedback at the end semester or course, feedback taken during the course can help to address teaching-learning issues. For a deeper understanding of the students, the views obtained from social media sites should be evaluated using some of the sentiment analysis techniques. Analyzing this relevant data for better monitoring of students' performance and learning goals.

*Richard J. Medford et al. [3]* During virus outbreaks, Twitter has been used to monitor

patterns and spread health information. On January 21, 2020, The Centers for Disease Control and Prevention triggered its Emergency Operations Center, and the World Health Organization issued first ever Corona virus Disease Situation Report of 2019, which garnered a lot of media attention. There is no reason for how the content and mood on Twitter shifted during the initial stages of any outbreak, such as the Corona virus outbreak.

*Rinor Hajrizi et al. [4]* Institutions and companies have long benefited from large-scale data analysis. People's articulated opinions via text have recently become a very important part of this research. Aspect-Based Sentiment Analysis which is also called ABSA is a natural-language processing methodology that has evolved in response to this challenge. ABSA has been found to be effective in a wide variety of domains due to its ability to derive polarity for each component of opinion separately. ABSA can be used successfully in a number of areas, including education.

*Catherine Ordun et al. [5]* Using five different techniques, this paper shows For Corona virus tweets, how to determine the individuality of topics, key terms and attributes, information propagation speed, and network patterns. To generate twenty unique topics about case distribution, hospital staff, and personal protective equipment, we used pattern recognition and then topic modelling with Latent Dirichlet Allocation.

## III EXISTING SYSTEM

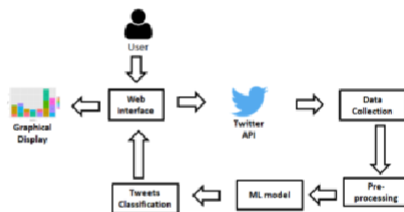
To evaluate the sentiment of a tweet, the current method employs sentiment analysis techniques. This, however, is insufficient to detect anti-social messages. The sentiment analysis may indicate whether the emotion is positive or negative. However, it's vital to classify whether the tweets were about student events.

Statement of the problem:

Huge quantity of tweets to sort.

The communal student behaviors would not be covered by sentiment research.

#### IV RECOMMENDED SYSTEM



**FIG 1 OVERVIEW OF THE RECOMMENDED SYSTEM**

To analyze the live tweet, we suggest a machine learning-based classifier model. A classification model is used in this system to identify the words used in the tweet. The algorithm is fed a range of different keywords and another input, which is the tweet to be found. The data is then plotted on a graph.

#### V MODULES DESCRIPTION

##### 5.1 DATA COLLECTION

For searching tweets, a Python API is used to connect to the Twitter server. To view tweets, create an Access Key and a Consumer Key from the Twitter website. Using the access key to connect to the Twitter database for tweet search via OAuth. Python can access the Twitter server via OAuth without sharing the login details.

##### 5.2 DATA PREPROCESSING

The majority of tweets can typically be divided into three sections, but not always in that order. The first segment includes the individuals to whom the tweet is addressed or who it refers to. "@" is typically used to indicate this. The actual message is found in the second segment. This is the message the user really wants to send. The hash tag,

denoted by the letter "#," appears in the third segment. This is usually done to classify the tweets into groups. Others might use this to identify tweets that are important to the material. Since the first and third portions of the tweets have little or no significance in sentiment analysis, they are pre-processed to filter them out. The remaining message section of the tweet is then further pre-processed to obtain valuable keywords for emotion recognition.

##### 5.3 MACHINE LEARNING ALGORITHMS

The Naive Bayes Classifier is used to implement the Bayes theorem, which relies on a strong independence hypothesis, especially an independent design model. The Bayes Theorem is based on conditional probability, which measures the chances of something happening if something else has already happened. Provided a collection of evidences, it calculates the conditional probability of a class and based on the highest one, determines the most probable class. Since it is a simple method that yields high precision, the naive Bayes classifier is a well-known also widely used methodology.

<sup>11</sup> The Support Vector Machine is a supervised learning algorithm for evaluating and identifying patterns in data. SVM <sup>3</sup> categorizes an input set into one of two groups. Both linear and non-linear classification can be treated by SVM. It can perform non-linear classification efficiently using the kernel trick. It accomplishes this by converting the inputs into <sup>8</sup> high-dimensional feature space. Polynomial, Gaussian radial basis function, Laplace RBF kernel, hyperbolic tangent kernel, and sigmoidal kernel are examples of kernel forms. SVM uses the architecture of hyper planes for classification. In this paper, the aim of a text classification system is to decide whether a given tweet falls into one of several predefined categories. Many optimal strategies are used by an efficient Support Vector Machine algorithm for text classification.

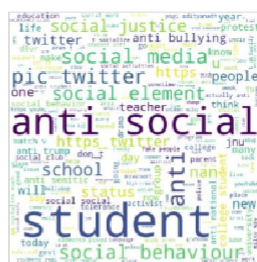
## 5.4 TRAINING AND TESTING

The training phase has two process-Construction of the Bag of Words model and the Predictive model and in testing phase using the naive Bayes and support vector machine algorithms mentioned previously, two predictive models are developed. The data file will have two columns, for each of the tweet categories and the other one for each individual tweet. The tweets column has been pre-processed. The pre-processed data can be used to construct a bag words system for training predictive models.

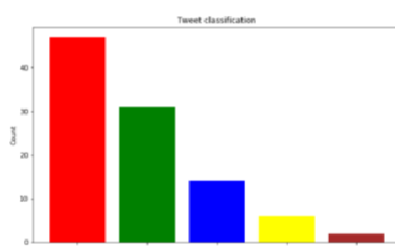
## 5.4 DATA VISUALIZATION

During the development process, 30% of the data is randomly split from the sample and used to test the predictive model. The data from the tests has been pre-processed or it is labeled.

## VI RESULT AND DISCUSSION

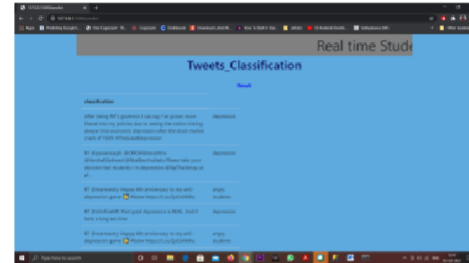


**Fig 3 Image cloud of dataset**



**Fig 4 Graphical representation of tweets**

Fig4 shows the graphical representation of tweets using mat plot library.



**Fig 5 Illusion of tweets classification**

## VI CONCLUSION

The goal of this study was to see if there was a relation between students' sentiments and emotions during the coronavirus crisis depending on their tweets. Based on the obtained findings, we can conclude that the Multinomial Nave Bayes algorithm performed well in the context of emotion detection as a result, Multinomial Naive Bayes has a higher precision value. Anomalies such as terms that have been joined together were also discovered, which could be separated to improve the model's accuracy. Previous proposals had better outcomes, but they were limited to a tiny proportion of tweets, and since data is dynamic, dynamic approaches are needed to overcome all potential challenges in sentiment analysis.

## REFERENCES

- [1] Dunkel, Alexander ; Andrienko, Gennady  
“A conceptual framework for studying  
collective reactions to events in location-based  
social media”, *International Journal of  
Geographical Information Science* Volume  
33, 2019.
- [2] A.Jenifer Jothi Mary, S. Santiago, Dr. L.  
Arockiam “A Methodological Framework to  
Identify the Students’ Opinion using Aspect

based SentimenAnalysis",International Journal of Engineering Research & Technology (IJERT), Published by : Vol. 5 Issue 02, February-2016.

[3] Richard J. Medford, Sameh N. Saleh,Andrew Sumarsono, Trish M. Perl, Christoph U. Lehmann," An "Infodemic": Leveraging High-Volume Twitter Data to Understand Early Public Sentiment for the COVID-19 Outbreak"Open Forum Infectious Diseases, Volume 7, Issue 7, July 2020.

[4] Rinor Hajrizi, Krenare Pireva Nuçi,"Aspect-based sentiment analysis in education domain",October 6, 2020.

[5] Catherine Ordun, Sanjay Purushotham, Edward Raff," Exploratory Analysis of Covid-19 Tweets using Topic Modeling, UMAP, and DiGraph",May 2020.

[6] Rupawari Jadhav , M. S. Wakode," Survey : Sentiment Analysis of Twitter Data for Stock Market Prediction",International Journal of Advanced Research in Computer and Communication Engineering,March 2017.

[7] Hai Liang,Issac Chun-Hai Fung,"How did Ebola information spread on twitter: broadcasting or viral spreading?", BMC Public Health volume 19,Article number: 438 (2019).

[4] Venkata Sasank Pagolu, Kamal Nayan Reddy Challa, Ganapati Panda, Babita Majhi,"Sentiment Analysis of Twitter Data for Predicting Stock Market Movements "International conference on Signal Processing, Communication, Power and Embedded System (SCOPE5)"-2016.

[9] Thanh Tran, Kyumin Lee," Understanding Citizen Reactions and Ebola-Related Information Propagation on Social Media",2016.

[60] Michele Miller MS, Dr. Tanvi Banerjee," What Are People Tweeting About Zika? An Exploratory Study Concerning Its Symptoms, Treatment, Transmission, and Prevention",2017.

paper

ORIGINALITY REPORT

11%

8%

9%

5%

SIMILARITY INDEX      INTERNET SOURCES      PUBLICATIONS      STUDENT PAPERS

PRIMARY SOURCES

|   |                                                                    |   |
|---|--------------------------------------------------------------------|---|
| 1 | Submitted to Humble Independent School District<br>Student Paper   | 2 |
|   | <a href="#">academic.oup.com</a><br>Internet Source                | 2 |
|   | %                                                                  | 1 |
| 3 | Mandar Deshpande, Vignesh Rao. "Depression detection using emotion |   |

artificial intelligence", 2017 International Conference on Intelligent Sustainable  
Systems (ICISS), 2017

Publication

---

[ijcseonline.org](http://ijcseonline.org)

Internet Source

%

1

---

[www.tandfonline.com](http://www.tandfonline.com)

Internet Source

%

1

---

[publichealth.jmir.org](http://publichealth.jmir.org)

Internet Source

%

1

---

[dblp.dagstuhl.de](http://dblp.dagstuhl.de)

Internet Source

%

1

---

Khaled Assi, Syed Masiur Rahman, Umer

8

Mansoor, Nedal Ratrout. "Predicting Crash

Injury Severity with Machine Learning Algorithm

%

Synergized with Clustering Technique: A Promising Protocol", International Journal of Environmental Research and Public Health, 2020  
Publication

[pdfs.semanticscholar.org](https://pdfs.semanticscholar.org)

Internet Source

1

[arxiv.org](https://arxiv.org)

Internet Source

1

[www.irjet.net](http://www.irjet.net)

Internet Source

12

"Encyclopedia of Social

Network Analysis and

Mining", Springer Science

and Business Media LLC,

2018

Publication

13

Alexander Dunkel,

Gennady Andrienko,

Natalia Andrienko, Dirk

Burghardt, Eva Hauthal,

Ross Purves. "A

conceptual framework for

studying collective

reactions to events in

location-based social

media", International

Journal of Geographical

1%

Information Science, 2018

<1%

Publication

<1%

<1%

<1%

|                |         |                 |     |
|----------------|---------|-----------------|-----|
| Exclude quotes | On      | Exclude matches | Off |
| bibliography   | Exclude |                 |     |
|                | On      |                 |     |