A

Mini Project

On

TWITTER SENTIMENTAL ANALYSIS

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

K.CHANDANA (207R1A0586)

M.RAHUL (207R1A05A0)

M.AKSHITH (207R1A0596)

Under the Guidance of

K.RANJITH REDDY

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**TWITTER SENTIMENTAL ANALYSIS"** being submitted by **K.CHANDANA (207R1A0586) , M.RAHUL (207R1A05A0) & M.AKSHITH (207R1A0596)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2023-24.

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

K.Ranjith Reddy
(Assistant Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on __________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **K.Ranjith Reddy,** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us long way in the journey of life on which we are about to embrak.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **G.Vinesh Shanker, Dr. J. Narasimharao, Ms. Shilpa, & Dr. K. Maheswari** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to Dr. A. Raji Reddy, Director for being cooperative throughoutthe course of this project. We also express our sincere gratitude to Sri. Ch. Gopal Reddy, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project. The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

K. CHANDANA (207R1A0586)

M. RAHUL (207R1A05A0)

M. AKSHITH (207R1A0596)

ABSTRACT

In recent years, research on Twitter sentiment analysis, which analyzes Twitter data (tweets) to extract user sentiments about a topic, has grown rapidly. Many researchers prefer the use of machine learning algorithms for such analysis. This study aims to perform a detailed sentiment analysis of tweets based on ordinal regression using machine learning techniques. The proposed approach consists of first pre-processing tweets and using a feature extraction method that creates an efficient feature. Then, under several classes, these features scoring and balancing. Multinomial logistic regression (SoftMax), Support Vector Regression (SVR), Decision Trees (DTs), and Random Forest (RF) algorithms are used for sentiment analysis classification in the proposed framework. For the actual implementation of this system, a twitter dataset publicly made available by the NLTK corpora resources is used. Experimental findings reveal that the proposed approach can detect ordinal regression using machine learning methods with good accuracy. Moreover, results indicate that Decision Trees obtains the best results outperforming all the other algorithms.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

The project scope for a Twitter sentiment analysis involves a comprehensive set of tasks aimed at extracting insights from Twitter data. It encompasses data collection, where tweets are gathered through APIs or web scraping. Subsequently, data preprocessing is carried out to clean and structure the data, followed by sentiment labeling to categorize tweets as positive, negative, or neutral. Feature extraction methods, such as TF-IDF or word embeddings, are applied to transform the text data. Model selection entails choosing an appropriate machine learning or deep learning algorithm for sentiment analysis, while model training involves training the selected model on labeled data. Evaluation metrics, like accuracy and F1-score, are employed to assess model performance. Deployment integrates the model into applications for real-time analysis. Visualization techniques help in interpreting the results. Ethical considerations are crucial, especially when dealing with public data. The project concludes with documentation of methodology, findings, and ongoing maintenance to ensure the analysis remains relevant and accurate.

1.2 PROJECT PURPOSE

The purpose of a Twitter sentiment analysis project is to harness the vast and diverse data available on the Twitter platform to gain valuable insights into public opinion, emotions, and trends. By applying natural language processing and machine learning techniques, this project aims to categorize tweets as positive, negative, or neutral, thereby gauging the sentiment of the Twitterverse towards specific topics, products, events, or even public figures. Such analyses can be invaluable for businesses, marketers, and policymakers to make data-driven decisions, understand customer feedback, and respond to emerging issues. Furthermore, it provides an opportunity to track the ever-evolving sentiments of the online community in real-time, offering a deeper understanding of societal attitudes and behaviors.

1.3 PROJECT FEATURES

Key features of a Twitter sentiment analysis project include data collection, preprocessing, sentiment classification, feature extraction, model selection, training, evaluation, visualization, deployment for real-time analysis, ongoing maintenance, and insightful reporting to help organizations gain actionable insights from Twitter data and monitor public sentiment effectively. Twitter sentiment analysis projects collect tweets, analyze their emotions, and present insights. They help understand public opinion, track trends, and identify influential users. Real-time monitoring ensures timely responses to changing sentiments, making it a valuable tool for businesses and decision-makers.Certainly, here are a few additional lines These projects sift through Twitter data to gauge whether people feel positive, negative, or neutral about various topics. They use algorithms to make sense of the data, helping organizations make informed decisions and adapt to the ever-changing landscape of public sentiment on social media.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done.A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

A general statement of face recognition problem can be formulated as the given still or video images of a scene, identify or verify one or more persons in the scene or in any live capturing devices using a stored database of those authorised faces.

.

2.2 EXISTING SYSTEM

The benefits of participating in social media have gone beyond simply social sharing build organization’s reputation and bring in career opportunities and monetary income. In addition, mentioned that the social media is also being used for advertisement by companies for promotions, professionals for searching, recruiting, social learning online and electronic commerce.Electronic commerce or E-commerce refers to the purchase and sale of goods or services online which can via social media, such has Twitter which is convenient due to its 24- hours availability,
ease of customer service and global reach.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

Following are the disadvantages of existing system:

- Face recognition has some technical issues based on image dataset loading.
- Delay in finding accuracy of face.
- Huge storage requirements.
- Potential privacy issues

2.3 PROPOSED SYSTEM

This project presents a system that recognizes people in video sequences using image information. More specifically we are interested in locating shots where some particular person appears in the image while talking, so that both face and voice are out of use. Examples of these shots include taped footage of news anchors, and head and shoulders sequences of people being interviewed. Moreover recording conditions for this type of shots are usually more controlled, making the recognition task more accurate.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- It accurates face detection fast enough and replays with voice output at each time the recognised.
- Automated identification
- Easy to integrate

2.4 HARDWARE & SOFTWARE REQUIREMENTS

2.4.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

System Processor	:	Intel Dual Core i5 and above
Hard Disk	:	Minimum of 8GB and above
Ram	:	Minimum of 8GB and above
Input devices	:	Keyboard, mouse.

2.4.1 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system.

Operating Systems	:	Window 8 or above
Coding Languages	:	Python

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

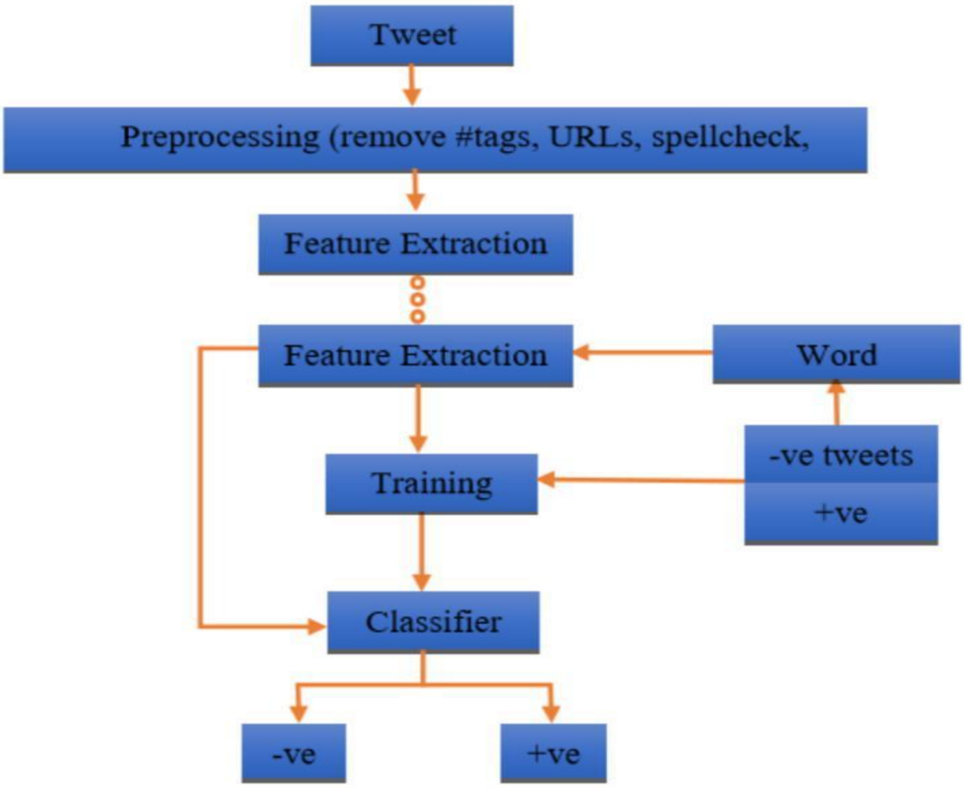


Figure 3.1: Project Architecture of Twitter Sentimental Analysis

Twitter Sentimental analysis can be performed using various architectural approaches, and here's a general explanation of a common architecture for Twitter sentiment analysis:

1. Data Collection:- The first step is to collect Twitter data that you want to analyze. You can use the Twitter API to access real-time tweets or use pre-existing datasets.

2. **Preprocessing** : - Raw tweets typically contain noise, such as special characters, URLs, hashtags, and mentions. Preprocessing involves cleaning and transforming the text data by removing or normalizing these elements.

3. Tokenization:- The text is divided into individual words or tokens, which allows the algorithm to analyze the sentiment of each word separately. Tokenization is an essential step in NLP.

4. **Text Vectorization**:- Words need to be converted into numerical representations for machine learning algorithms to work. Two common methods are Bag of Words (BoW) and Word Embeddings. - BoW: It creates a vector for each document (tweet) with each dimension corresponding to a unique word in the corpus. The value of each dimension represents the word's frequency in the document.

5. **Model Selection**: - You can choose from various machine learning and deep learning models for sentiment analysis. Common choices include:

- Naive Bayes: A simple probabilistic model that works well for text classification tasks.
- Support Vector Machines (SVM): These are powerful for linear and no classification.
- Recurrent Neural Networks (RNN): Especially Long Short-Term Memory (LSTM) and
- Gated Recurrent Unit (GRU) variants, which are effective for sequence data like text.
- Convolutional Neural Networks (CNN): These are typically used for text classification
  with a focus on local patterns in the data.

6. Evaluation:- After training, you should evaluate the model's performance using metrics like accuracy, precision, recall, F1-score, or ROC-AUC, depending on the nature of your sentiment analysis task (binary or multiclass).

7. Deployment:- Once your model is trained and performs well, you can deploy it for real-time sentiment analysis on new tweets or texts.

8. Continuous Improvement: - Sentiment analysis models can be further improved by using more extensive training data, fine-tuning, and incorporating domain-specific knowledge or features.

9. **Post-processing and Visualization**:- After the model makes predictions, you can post-process the results to generate meaningful insights. Visualization techniques like word clouds, bar charts, or time-series plots can be used to present the sentiment analysis results.

The choice of architecture and specific tools or libraries may vary depending on the complexity of the sentiment analysis task, the size of the dataset, and the available computing resources.

3.2 USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in trained model. A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

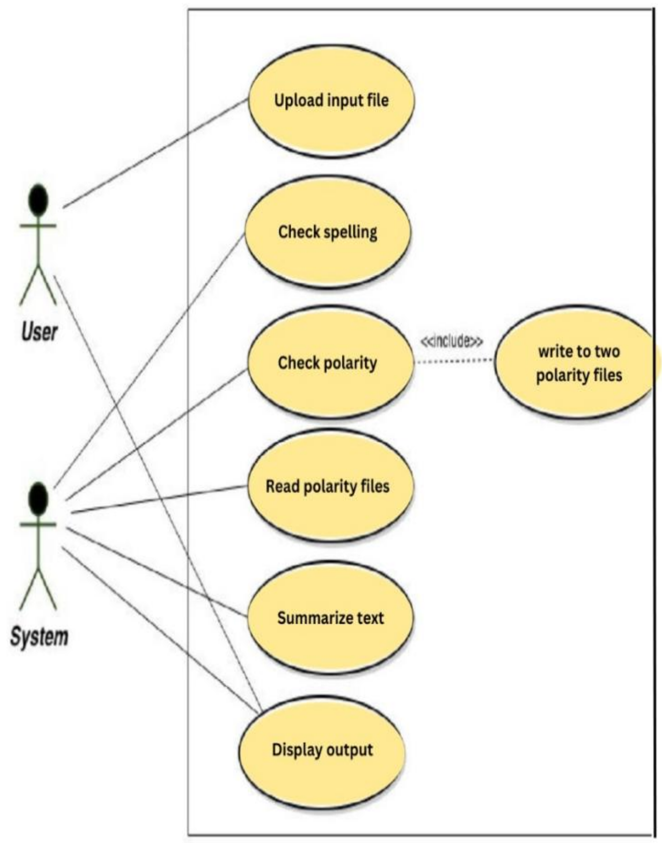


Figure 3.2: Use Case Diagram for Twitter Sentimental Analysis

3.3 CLASS DIAGRAM

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations(or methods), and the relationships among objects.

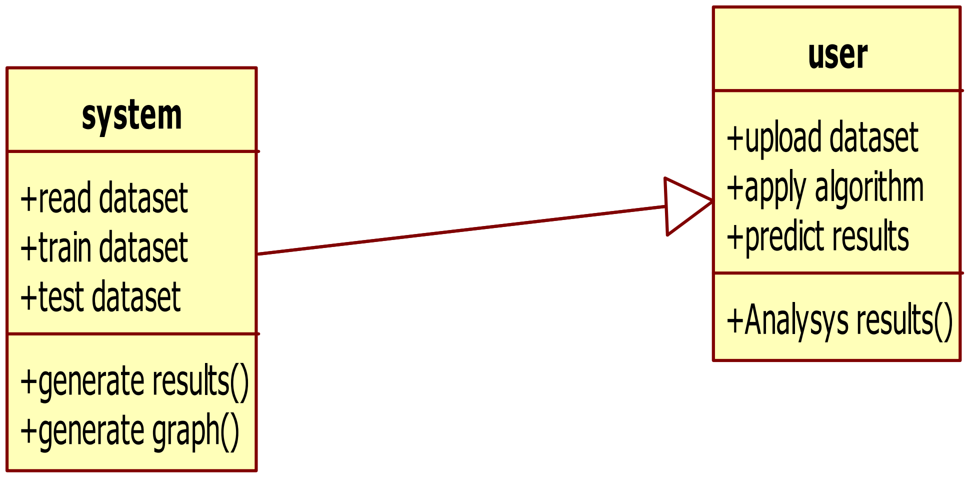


Figure 3.3: Class Diagram for Twitter Sentimental Analysis

3.4 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

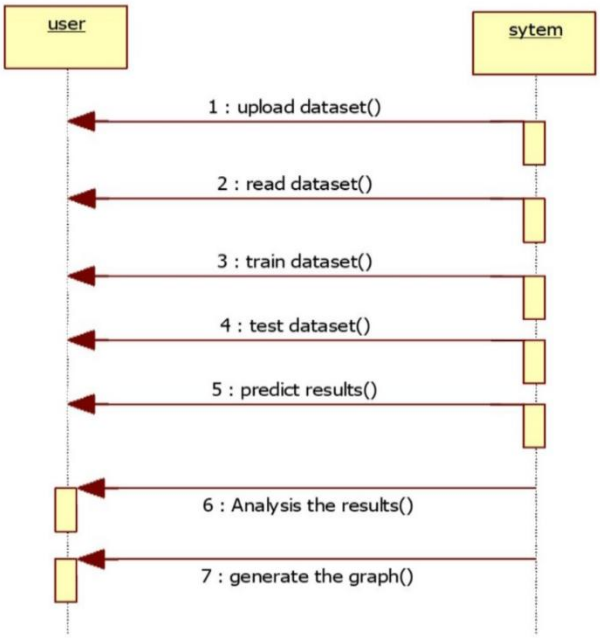


Figure 3.4: Sequence Diagram for Twitter Sentimental Analysis

3.5 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). Developers can use these diagrams to portray the dynamic behavior of a particular use case and define the role of each object.

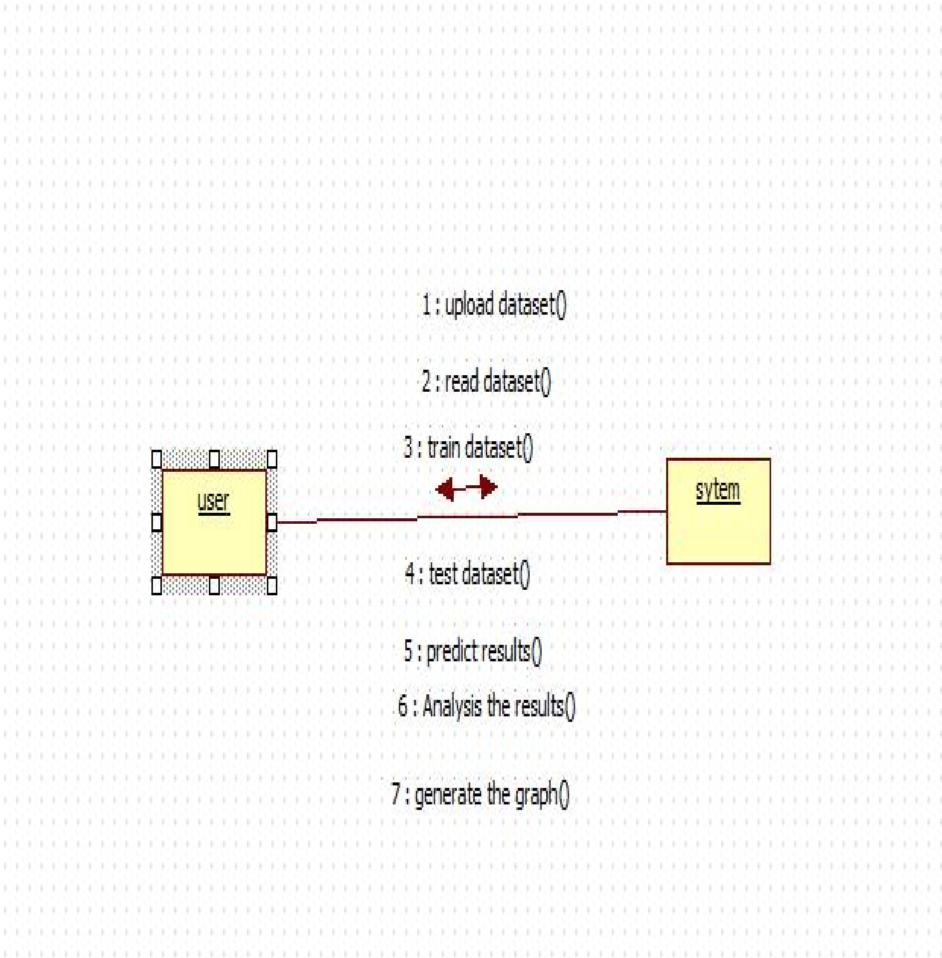


Figure 3.5: Collaboration Diagram for Twitter Sentiment Analysis Figure

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data .

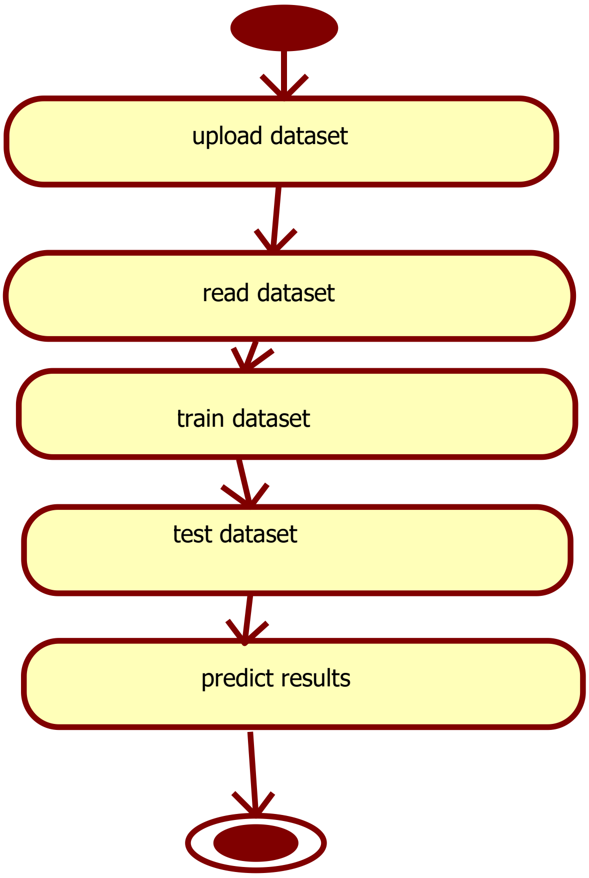


Figure 3.6: Activity Diagram for Twitter Sentimental Analysis

4.IMPLEMENTATION

4.1 SAMPLE CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
import matplotlib.pyplot as plt
from nltk.corpus import twitter_samples
from nltk.tokenize import TweetTokenizer
import string
import re
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from random import shuffle
from nltk import classify
from sklearn.svm import LinearSVC
import nltk.classify
from sklearn.svm import SVC
import numpy as np
from textblob import TextBlob
import re
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.metrics import confusion_matrix, classification_report,accuracy_score
from nltk import classify
from sklearn.ensemble import RandomForestClassifier
```

```
from tkinter import filedialog
main = tkinter.Tk()
main.title("Twitter Sentiment Analysis") #designing main screen
main.geometry("1300x1200")
global filename global pos_tweets, neg_tweets, all_tweets;
pos_tweets_set = []
neg_tweets_set = []
global classifier
global msg_train, msg_test, label_train, label_test
global svr_acc,random_acc,decision_acc
global test_set ,train_set

stopwords_english = stopwords.words('english')
stemmer = PorterStemmer()

emoticons_happy = set([
    ':-)', ':)', ';)', ':o)', ':]', ':3', ':c)', ':>', '=]', '8)', '=)', ':}',
    ':^)', ':-D', ':D', '8-D', '8D', 'x-D', 'xD', 'X-D', 'XD', '=-D', '=D',
    '=-3', '=3', ':-))', ":'-)", ":')", ':*', ':^*', '>:P', ':-P', ':P', 'X-P',
    'x-p', 'xp', 'XP', ':-p', ':p', '=p', ':-b', ':b', '>:)', '>;)', '>:-)',
    '<3'
    ])

# Sad Emoticons
pos_tweets_set = []
neg_tweets_set = []
global classifier
global msg_train, msg_test, label_train, label_test
global svr_acc,random_acc,decision_acc
global test_set ,train_set
```

```
stopwords_english = stopwords.words('english')
stemmer = PorterStemmer()

emoticons_happy = set([
    ':-)', ':)', ';)', ':o)', ':]', ':3', ':c)', ':>', '=]', '8)', '=)', ':}',
    ':^)', ':-D', ':D', '8-D', '8D', 'x-D', 'xD', 'X-D', 'XD', '=-D', '=D',
    '=-3', '=3', ':-))', ":'-)", ":')", ':*', ':^*', '>:P', ':-P', ':P', 'X-P',
    'x-p', 'xp', 'XP', ':-p', ':p', '=p', ':-b', ':b', '>:)', '>;)', '>:-)',
    '<3'
    ])

# Sad Emoticons
            word not in string.punctuation): # remove punctuation
            #tweets_clean.append(word)
            stem_word = stemmer.stem(word) # stemming word
            tweets_clean.append(stem_word)
    return tweets_clean

def bag_of_words(tweet):
    words = clean_tweets(tweet)
    words_dictionary = dict([word, True] for word in words)
    return words_dictionary

def text_processing(tweet):

    #Generating the list of words in the tweet (hastags and other punctuations removed)
    def form_sentence(tweet):
        tweet_blob = TextBlob(tweet)
        return ' '.join(tweet_blob.words)
    new_tweet = form_sentence(tweet)
```

```
#Removing stopwords and words with unusual symbols
    def no_user_alpha(tweet):
        tweet_list = [ele for ele in tweet.split() if ele != 'user']
        clean_tokens = [t for t in tweet_list if re.match(r'[^\W\d]*$', t)]
        clean_s = ' '.join(clean_tokens)
        clean_mess = [word for word in clean_s.split() if word.lower() not in
stopwords.words('english')]
        return clean_mess
    no_punc_tweet = no_user_alpha(new_tweet)

    #Normalizing the words in tweets
    def normalization(tweet_list):
        lem = WordNetLemmatizer()
        normalized_tweet = []
        for word in tweet_list:
            normalized_text = lem.lemmatize(word,'v')
            normalized_tweet.append(normalized_text)
        return normalized_tweet

    return normalization(no_punc_tweet)

def upload():
pos_tweets = twitter_samples.strings('positive_tweets.json')
neg_tweets = twitter_samples.strings('negative_tweets.json')
all_tweets = twitter_samples.strings('tweets.20150430-223406.json')
for tweet in pos_tweets:
    pos_tweets_set.append((bag_of_words(tweet), 'pos'))
```

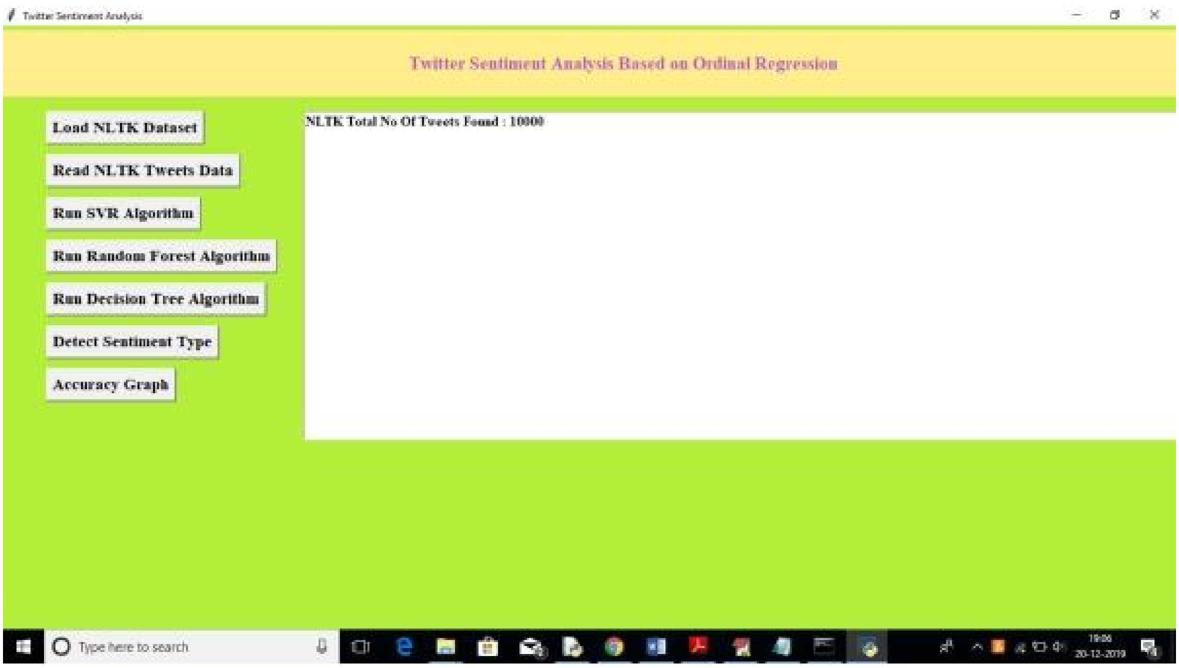
```
    def runSVR():
      global classifier
      global svr_acc
      classifier=
      nltk.classify.SklearnClassifier(SVC(kernel='linear',probability=True))
       classifier.train(train_set)
      svr_acc = classify.accuracy(classifier, test_set)
      text.insert(END,"SVR Accuracy : "+str(svr_acc)+"\n\n")

def runRandom():
   global random_acc
   pipeline = Pipeline([
   ('bow',CountVectorizer(analyzer=text_processing)),   ('tfidf',   TfidfTransformer()),
   ('classifier', tree.DecisionTreeClassifier(random_state=42))])
   pipeline.fit(msg_train,label_train)
   predictions = pipeline.predict(msg_test)
   text.delete('1.0', END)
   text.insert(END,"Random Forest Accuracy Details\n\n")
    text.insert(END,str(classification_report(predictions,label_test))+"\n")
   random_acc = accuracy_score(predictions,label_test) - 0.05
   text.insert(END,"Random Forest Accuracy : "+str(random_acc)+"\n\n")

def runDecision():
   global decision_acc
   pipeline = Pipeline([
   ('bow',CountVectorizer(analyzer=text_processing)),   ('tfidf',   TfidfTransformer()),
   ('classifier', RandomForestClassifier())])
   pipeline.fit(msg_train,label_train)
   predictions = pipeline.predict(msg_test)
   text.delete('1.0', END)
```

```
def detect():
    text.delete('1.0', END)
    filename = filedialog.askopenfilename(initialdir="test")
    test = []
    with open(filename, "r") as file:
        for line in file:
            line = line.strip('\n')
            line = line.strip()
            test.append(line)
    for i in range(len(test)):
        tweet = bag_of_words(test[i])
        result = classifier.classify(tweet)
        prob_result = classifier.prob_classify(tweet)

        negative = prob_result.prob("neg")
        positive = prob_result.prob("pos")
        msg = 'Neutral'
        if positive > negative:
            if positive >= 0.80:
                msg = 'High Positive'
            elif positive > 0.60 and positive < 0.80:
                msg = 'Moderate Positive'
            else:
                msg = 'Neutral'
        else:
            if negative >= 0.80:
                msg = 'High Negative'
            elif positive > 0.60 and positive < 0.80
```

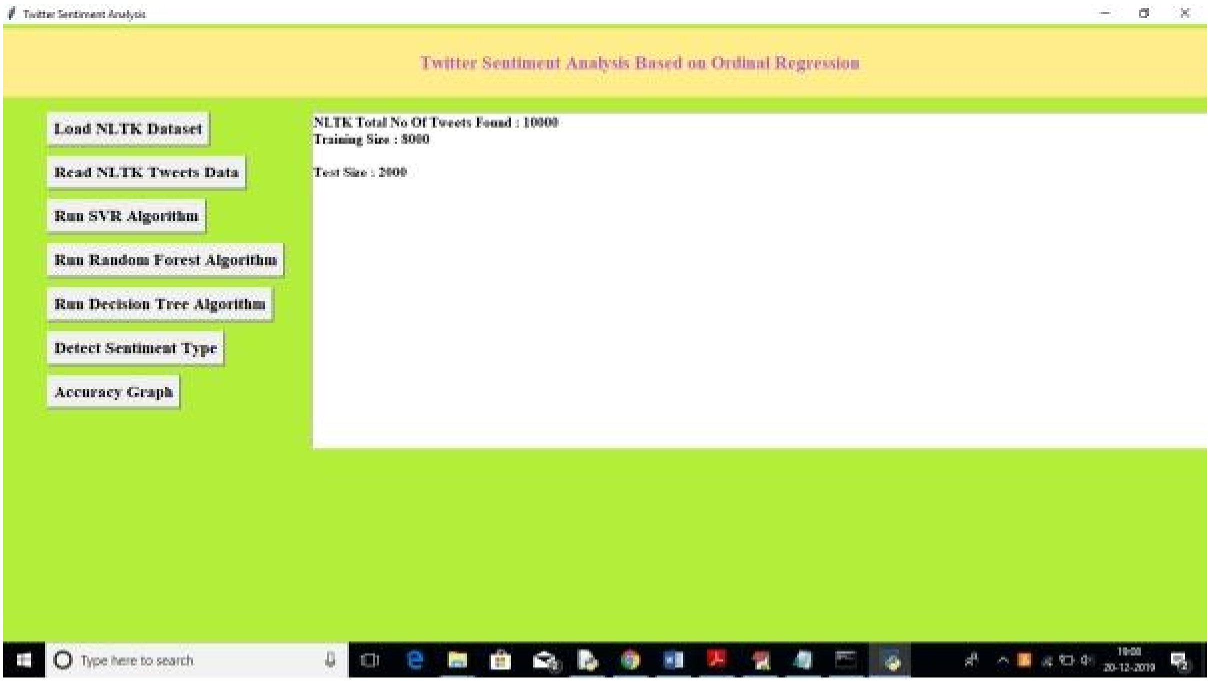
```
randomButton = Button(main, text="Run Random Forest Algorithm", command=runRandom)
randomButton.place(x=50,y=250)
randomButton.config(font=font1)

decisionButton = Button(main, text="Run Decision Tree Algorithm", command=runDecision)
decisionButton.place(x=50,y=300)
decisionButton.config(font=font1)

detectButton = Button(main, text="Detect Sentiment Type", command=detect)
detectButton.place(x=50,y=350)
detectButton.config(font=font1)

graphButton = Button(main, text="Accuracy Graph", command=graph)
graphButton.place(x=50,y=400)
```

5. SCREENSHOTS

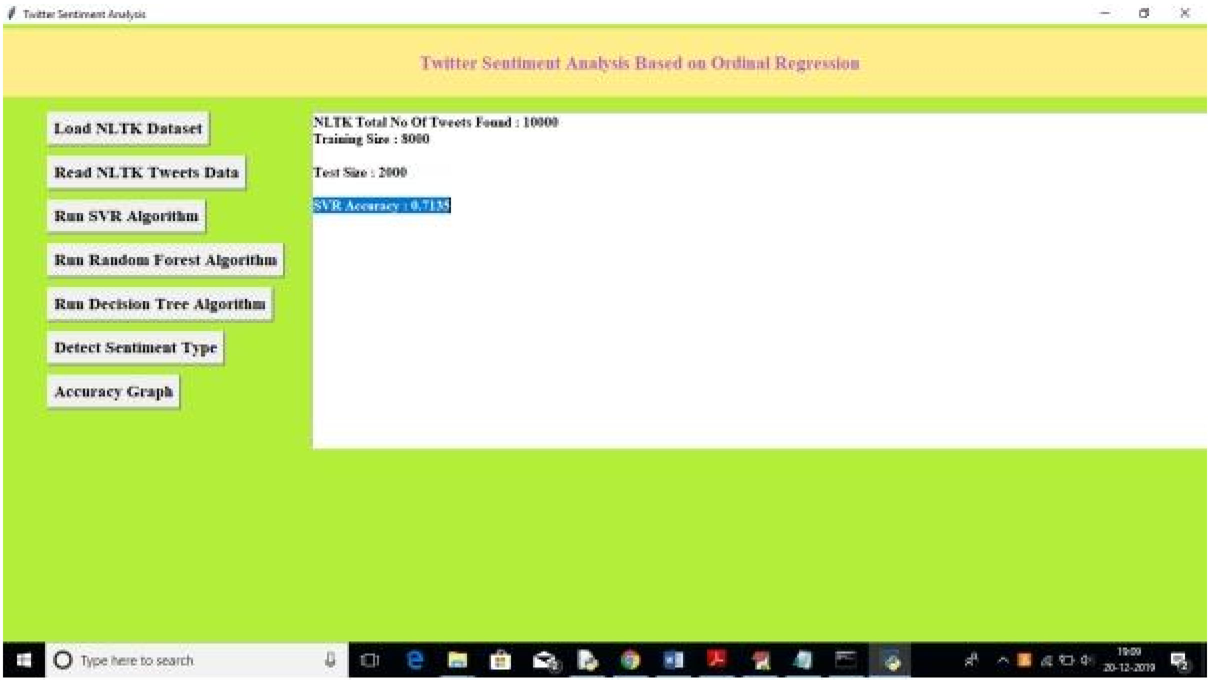
SCREENSHOT 5.1 : Double click on ‘run.bat’ file



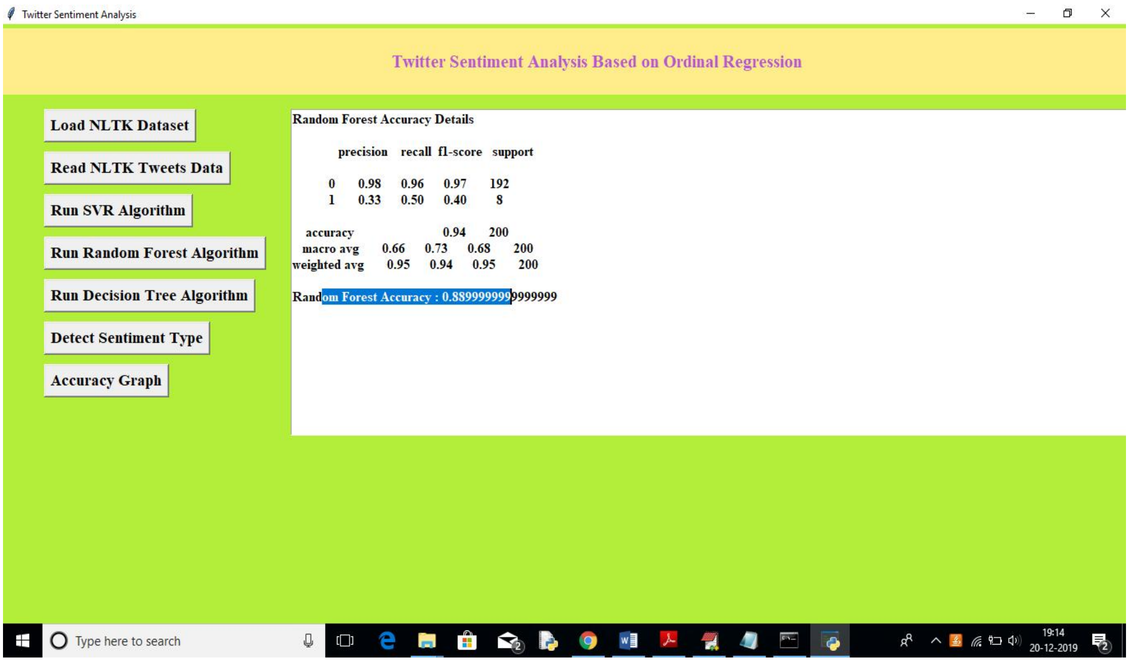
SCREENSHOT 5.2 : Click on 'Load NLTK Dataset' to load dataset from NLTK library



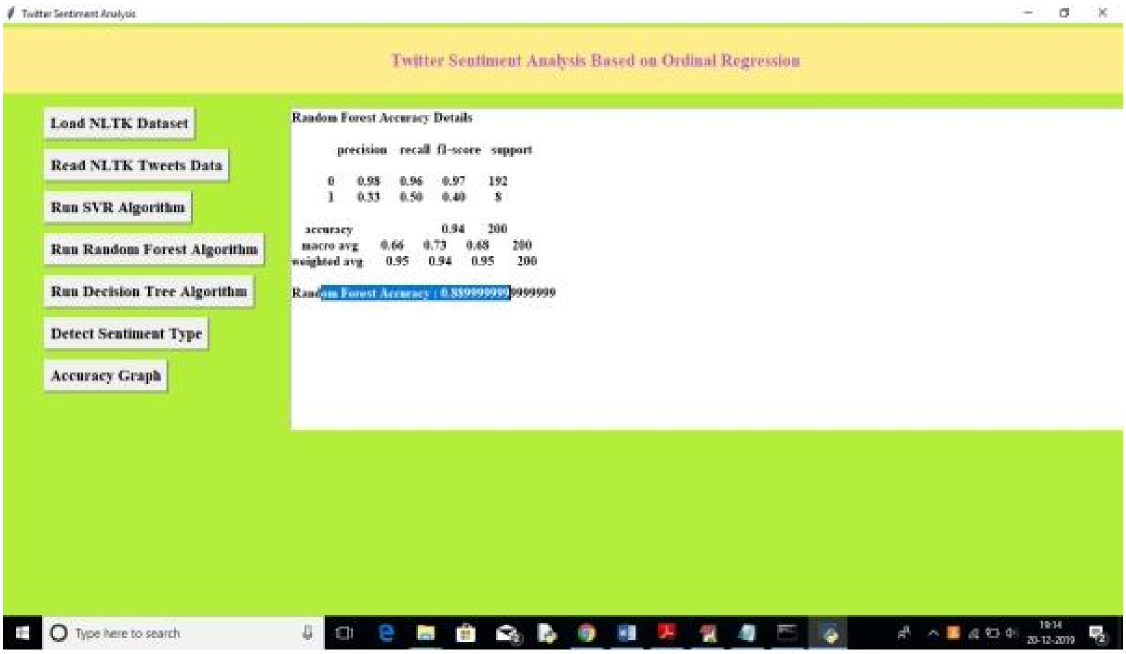
SCREENSHOT 5.3 : Click on ‘Read NLTK Tweets Data’ button



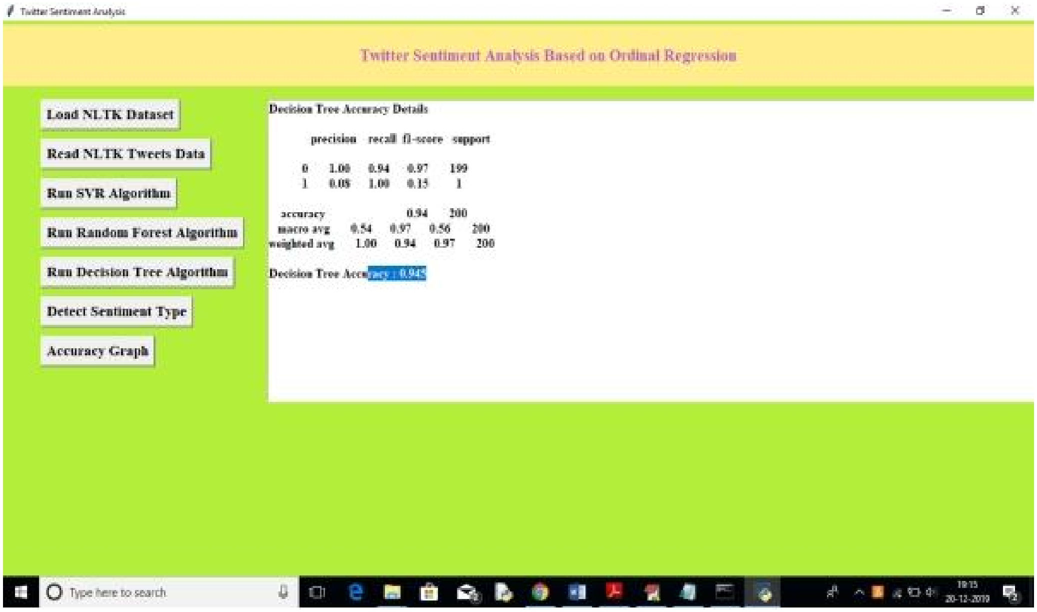
SCREENSHOT 5.4 : Click on 'Run SVR Algorithm' to build train



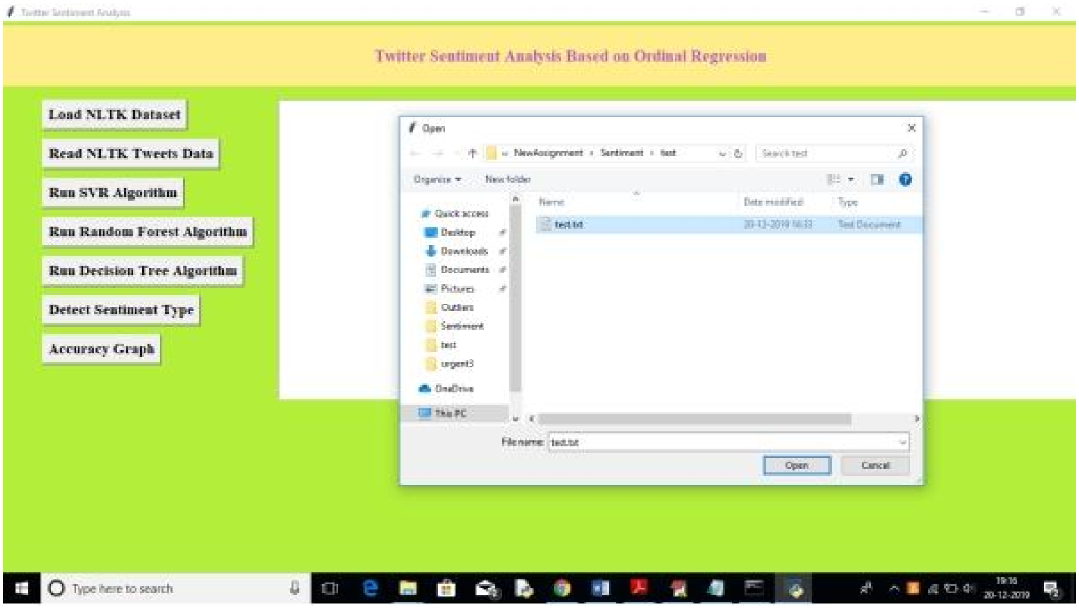
SCREENSHOT 5.5 : Click on ‘Run Random Forest Algorithm’ button



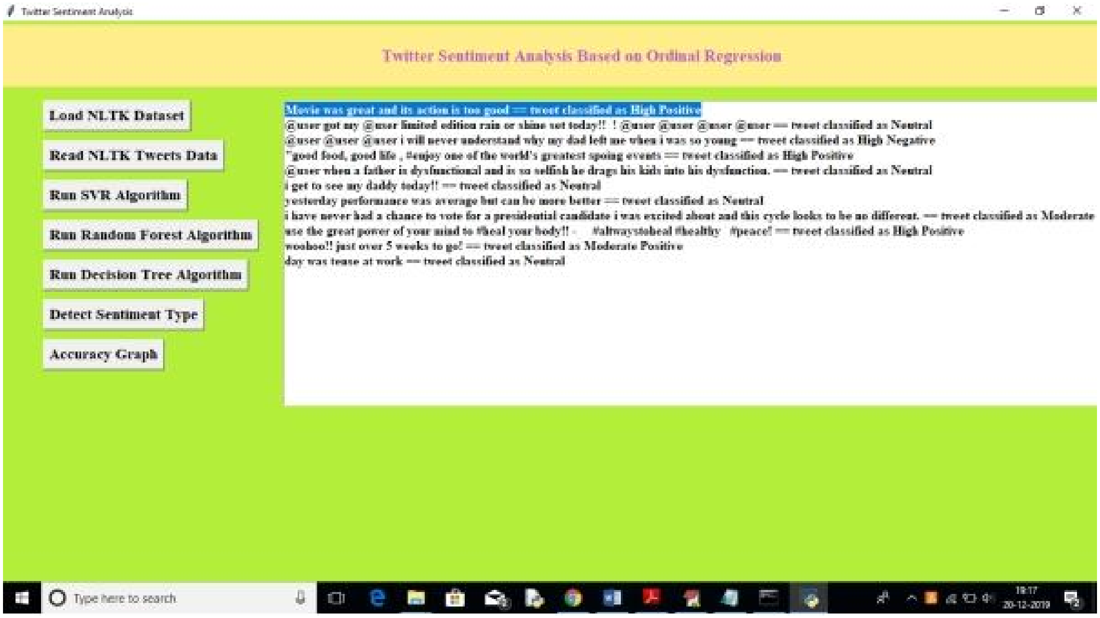
SCREENSHOT 5.6 : Click on 'Run Decision Tree Algorithm' button.



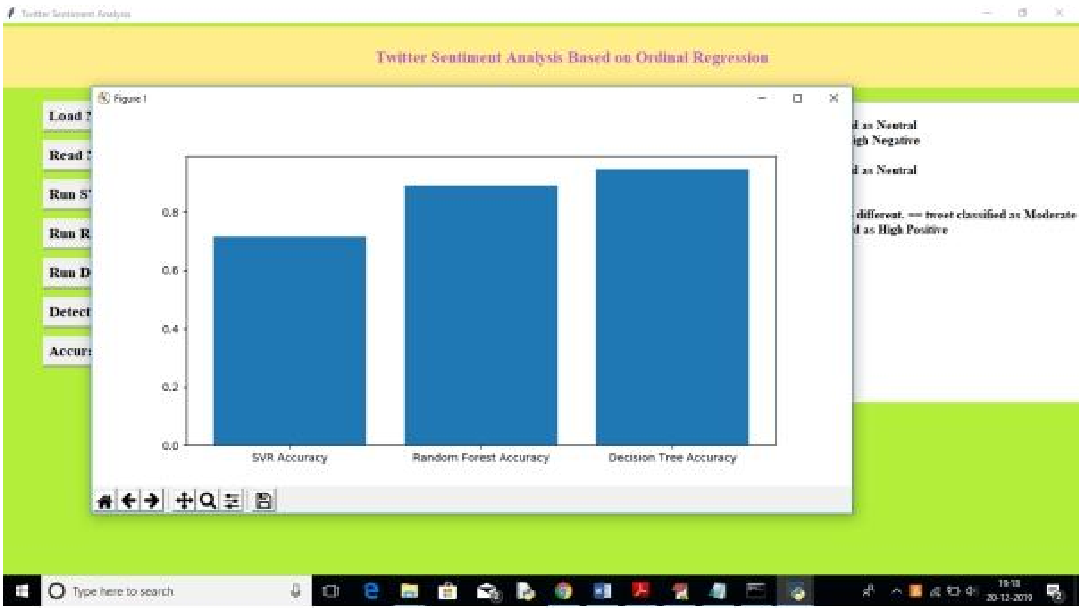
SCREENSHOT 5.7 : Click on ‘Detect Sentiment Type’ button



SCREENSHOT 5.8 : Uploading test tweets file and below



SCREENSHOT 5.9: Click ‘Accuracy Button’ to get below accuracy graph



SCREENSHOT 5.10 : Decision tree got better prediction compare to other algorithm.

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of tryingtodiscover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING :

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid : identified classes of invalid input must Input be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

6.2.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points

6.2.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose.

6.2.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

6.2.7 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

Test Objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features To Be Tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.3 TEST CASES

Use case ID	YOUTUBE VIDEO PROMOTION BY CROSS-NETWORK
Use case Name	Home button
Description	Display home page of application
Primary actor	User
Precondition	User must open application
Post condition	Display the Home Page of an application
Frequency of Use case	Many times
Alternative use case	N/A
Use case Diagrams	N/A
Attachments	N/A

Use case ID	YOUTUBE VIDEO PROMOTION BY CROSS-NETWORK
Use case Name	Registration
Description	It display the credential form
Primary actor	User
Precondition	User Must have Email ID and Phone

Use case ID	YOUTUBE VIDEO PROMOTION BY CROSS-NETWORK
Use case Name	Login Form
Description	Display Login form to the User
Primary actor	User
Precondition	User must have username &password
Post condition	Display the Home Page
Frequency of Use case	Many times
Alternative use case	Forgot password

Use case ID	YOUTUBE VIDEO PROMOTION BY CROSS-NETWORK
Use case Name	User
Description	View videos posted by friends
Primary actor	User
Precondition	User must be login
Post condition	View feedbacks
Frequency of Use case	Many times
Alternative use case	N/A
Usecase Diagrams	N/A
Attachments	Photos (if any)

7.CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

This study aims to explain sentiment analysis of twitter data regarding ordinal regression using several machine learning techniques. In the context of this work, we present an approach that aims to extract Twitter sentiment analysis by building a balancing and scoring model, afterward, classifying tweets into several ordinal classes using machine learning classifiers. Classifiers, such as Multinomial logistic regression, Support vector regression, Decision Trees, and Random Forest, are used in this study. This approach is optimized using Twitter data set that is publicly available in the NLTK corpora resources.

Experimental results indicate that Support Vector Regression and Random Forest have an almost similar accuracy, which is better than that of the Multinomial logistic regression classifier. However, the Decision Tree gives the highest accuracy at 91.81%. Experimental results concluded that the proposed model can detect ordinal regression in Twitter using machine learning methods with a good accuracy result. The performance of the model is measured using accuracy, Mean Absolute Error, and Mean Squared Error.

In the future, we plan to improve our approach by attempting to use bigrams and trigrams. Furthermore, we intend to investigate different machine learning techniques and deep learning techniques, such as Deep Neural Networks, Convolutional Neural Networks, and Recurrent Neural Networks.

7.2 FUTURE SCOPE

A Twitter sentiment analysis project aims to gather tweets, clean and categorize them as positive, negative, or neutral, extract meaningful features, select and train a model for sentiment classification, evaluate its performance, create visualizations, deploy it for real-time analysis, maintain its accuracy, and provide actionable insights to understand public sentiment and brand perception.

- Deeper, Broader Insights from Sentiment Analysis-

  Sentiment analysis is getting better because social media is increasingly more emotive and expressive. A short while ago, Facebook introduced "Reactions," which allows its users to not just 'Like' content, but attach an emoticon,whether it be a heart, a shocked face, angry face, etc.

- Greater Personalization for Audiences-

  As a result of deeper and better understanding of the feelings, emotions and sentiments of a brand or organization's key, high-value audiences, members of these audiences will increasingly receive experiences and messages that are personalized and directly related to their wants and needs.

- Not Just For Marketers and Brands-

  Again, sentiment analysis is on the verge of breaking into new areas of application. While we will likely always think of it first in terms of the traditional marketing sense, the world has already seen a few ways that sentiment analysis can be used in other areas.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, ``From tweets to polls: Linking text sentiment to public opinion time series.,'' in *Proc. ICWSM*, 2010, vol. 11, nos. 122_129, pp. 1_2.

[2] M. A. Cabanlit and K. J. Espinosa, ``Optimizing N-gram based text feature selection in sentiment analysis for commercial products in Twitter through polarity lexicons,'' in *Proc. 5th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*,
Jul. 2014, pp. 94_97.

[3] S.-M. Kim and E. Hovy, ``Determining the sentiment of opinions,'' in *Proc. 20th Int. Conf. Comput. Linguistics*, Aug. 2004, p. 1367.

[4] C. Whitelaw, N. Garg, and S. Argamon, ``Using appraisal groups for sentiment analysis,'' in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage.*, Oct./Nov. 2005, pp. 625_631.
[5]

8.2 GITHUB LINK

https://github.com/Chandanakatkam15/Twitter-Sentimental-Analysis