VIGNAN'S FOUNDATION FOR SCIENCE TECHNOLOGY AND RESEARCH Deemed to be UNIVERSITY

VADLAMUDI, GUNTUR DIST, ANDHRA PRADESH, INDIA, PIN-522 213

CERTIFICATE

This is to certify that the Internship Report entitled **"Twitter Sentiment Analysis based on Ordinal Regression "**that is being submitted by Jayesh Mandava **(161FA07081)**in partial fulfilment for the award of B.Tech degree in Computer Science and Engineering to the Vignan's Foundation for Science, Technology and Research, Deemed to be University, is a record of bonafide work carried out by them at **Mathlog It Solutions PVT LTD** under the supervision of Name **"Mr.Ruthvick"** and under the co-guidance of the following faculty member of IT Department.

Project Guide   Head of the Department

DECLARATION

I hereby declare that the project entitled "**TWITTER SENTIMENT ANALYSIS BASED ON ORDINAL REGRESSION**" submitted for the **DEPARTMENT OF INFORMATION TECHNOLOGY**. This dissertation is our original work and the project has not formed the basis for the award of any degree, associate-ship and fellowship or any other similar titles and no part of it has been published or sent for publication at the time of submission.

By

N.Naveen Gopi Chand(161FA07058)

Jayesh Mandava(161FA07081)

Date:

ACKNOWLEDGEMENT

Internship program is a golden opportunity for learning and self-development. Consider myself very lucky and honored to have so many wonderful people lead me through in the completion of this project.

I express my gratitude towards the Managing Director of **MATHLOG IT SOLUTIONS PVT LTD,** for permitting me to undertake the **Internship Program** in their extreme company and for the help and cooperation throughout the course of my Internship Program.

We feel it our responsibility to thank **Mr.RUTHIVCK** under whose valuable guidance that the project came out successfully after each stage, and also it is our responsibility to extend our thanks to **Mr.ANANDH KUMAR, Department Internship Coordinator,** for extending his support towards the Internship Program in **MATHLOG IT SOLUTIONS PVT LTD**, in **Hyderabad**.

It is a great pleasure for me to express my sincere thanks to **Prof.K.V.Krishna Kishore, HOD, IT** of VFSTR Deemed to be University**,** for providing me an opportunity to do my Internship Program.

I We extend our whole hearted gratitude to all our faculty members of Department of Computer Science and Engineering who helped us in our academics throughout course.

Finally we wish to express thanks to our family members for the love and affection overseas and forbearance and cheerful depositions, which are vital for sustaining effort, required for completing this work.

With Sincere regards,

N.Naveen(161FA07058)

Jayesh Mandava(161FA07081)

TABLE OF CONTENTS

LIST OF FIGURES	6
ABBREIVATIONS	7
INTERNSHIP SUMMARY	8
PROFILE OF THE COMPANY	9
1. INTRODUCTION	11
1.1 Introduction	11
1.2 Literature Survey	11
1.2 Project Background	12
1.3 Objective	12
1.4 Project Description	12
2. SOFTWARE REQUIREMENTS SPECIFICATION	15
2.1 Requirement Analysis	15
2.2 Problem Statement	15
2.3 Functional Requirements	15
2.4 Software Requirement Specification	15
2.5 Software Requirements	18
2.6 Hardware Requirements	18
2.9 External Interface Requirements	19
2.10 Feasibility study	20
3. ANALYSIS & DESIGN	22
3.1 Introduction	22
3.2 System Overview	23
3.3 System Architecture	24
End Procedure	26
End Procedure	27
4. MODELING	31

4.1 Design 31

5. IMPLEMENTATION 37

5.1 Sample Code 37

5.2 Screen Captures 44

6. TESTING 52

6.1 Software Testing 52

Software testing is the process of validating and verifying that a software application meets the technical requirements which are involved in its design and development. It is also used to uncover any defects/bugs that exist in the application. It assures the quality of the software. There are many types of testing software viz., manual testing, unit testing, black box testing, performance testing, stress testing, regression testing, white box testing etc. 52

6.2 Black box Testing 52

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing. 52

6.3 White box Testing 52

White box testing is when the tester has access to the interior data structures and algorithms including the code that implement these. 52

6.4 Performance Testing 52

Performance testing is executed to work out how briskly a system or sub-system performs under a specific workload. It also can serve to validate and verify other quality attributes of the system like scalability, reliability and resource usage. 52

7. RESULTS AND CHALLENGES 56

7.1 Results 56

7.2 Challenges 56

7. CONCLUSION 58

7.1 Conclusions 58

7.2 Scope for future work 58

BIBLIOGRAPHY 59

LIST OF FIGURES

Figure 3-1 Architecture of Django……………........………………………………..24

Figure 3-2 System Architecture ………..……………………………...……………25

Figure 4-1 Usecase Diagram for System ...……………………………………...32

Figure 4-2 Sequence Diagram for System……………………………….........33

Figure 4-3 Activity Diagram for system......………………………………..……34

Figure 4-4 Class Diagram for Online module…...……………….……………...35

Figure 5-1 Welcome page……………………..…………………………..…...45

Figure 5-2 Loading Dataset……………………………..…..………………..45

Figure 5-3 Read NLTK Tweets Data……………………...……………………46

Figure 5-4 Running SVR Algorithms………………….……………………..…...46

Figure 5-5 Running Random Forest Algorithm..……………………..……...47

Figure5-6 Running Decision Trees…………………………………..………47

Figure 5-7 Giving file for detecting Sentiment Type..………………..……….48

Figure 5-8 Sentiment Types…..……………………………………………….48

Figure 5-9 Accuracy Graph……………………………………..……………49

Figure 6-1 Fold Cross Validation Scores…..……………………….…………...51

Figure 6-2 Accuracy of all classifiers……..…………………..……………...52

Figure 6-3 Mean Squared error and mean absolute error results……….…….53

ABBREIVATIONS

● SVR	: SUPPORT VECTOR REGRESSION
● SDK	: SOFTWARE DEVELOPMENT KIT
● RF	: RANDOM FOREST
● DT	: DECISION TREE
● SDD	: SOFTWARE DESIGN DESCRIPTION
● MSE	: MEAN ABSOLUTE ERROR
● NLTK	: NATURAL LANGUAGE TEXT PROCESSING
● TF-IDF	: TERM FREQUENCY DOCUMENT INDEX FREQUENCY
● UML	:UNIFIED MODEL LANGUAGE
● WIFI	: WIRELESS FIDELITY
● SVM	:SUPPORT VECTOR MACHINE
● NLP	:NATURAL LANGUAGE PROCESSING
● API	:APPLICATION PROGRAM INTERFACE

INTERNSHIP SUMMARY

Location: Hyderabad

Center: MATHLOG IT SOLUTIONS PVT LTD.

Duration: 5 Months 18 days

Date of start: 16th December,2019

Date of submission:

Title of project: Twitter Sentiment Analysis Based on Ordinal Regression

Team Members:N.Naveen Gopi Chand(161FA07058)
M.Jayesh Rao(161FA07081)

Name of the guide: Mr.Ruthivick

Name of Faculty guide: D.Anandh Kumar, Assistant Professor, VFSTR University.

Project Area: Twitter Sentiment Analysis Based on Ordinal Regression

Abstract: 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.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.

Signature of Student	**Signature of Faculty Guide**
Date:	**Date:**

PROFILE OF THE COMPANY

About MathLog IT Solutions Pvt Ltd

Mathlog It Solutions Private Limited is a Private incorporated on 21 January 2019. It is classified as Non- govt Company and is registered at Registrar of Companies, Hyderabad. Its authorized share capital is Rs. 600,000 and its paid up capital is Rs. 300,000. It is inolved in Business activities n.e.c.

Mathlog It Solutions Private Limited's Annual General Meeting (AGM) was last held on N/A and as per records from Ministry of Corporate Affairs (MCA), its balance sheet was last filed on N/A.

Directors of Mathlog It Solutions Private Limited are Mallampalli Venkata Suguna Kumara Ruthvik and Bojja Reddy Bharath Simha.

Mathlog It Solutions Private Limited's Corporate Identification Number is (CIN) U74990TG2019PTC129891 and its registration number is 129891.Its Email address is ruthvik@mathlogit.com.

Mathlog It Solutions Private Limited's last Annual General Meeting(AGM) was held on **30 September 2019**, and date of latest balance sheet available from Ministry of Corporate Affairs(MCA) is **31 March 2019**.

Company address:

Dilsukhnagar**,** MathLog IT Solutions Pvt Ltd, Hyderabad

CHAPTER - 1

INTRODUCTION

1. INTRODUCTION

1.1 Introduction

social networks and microblogging websites being developed rapidly, Microblogging websites have become one of the largest web destinations for expressing people's thoughts, opinions, and attitudes about different topics.a vast amount of information is being generated by twitter as it is a microblogging platform and social networking service. In recent times, researchers made the use of social data for sentiment analysis of people's opinions on a product, topic, or event. Sentiment analysis, which is also known as opinion mining, is an important natural language processing task. This process gives info about the sentiment orientation of text as positive, negative, or neutral.

Twitter sentiment analysis is currently the most popular topic for research. Hence, these analysis are useful because it gathers and classifies public opinion by analyzing big social data. However, data from Twitter have certain characteristics that cause difficulty in conducting sentiment analysis.these Tweets are restricted to 140 characters written in informal English, contain irregular expressions, and also several abbreviations and slang words. To address these problems, researchers have already conducted studies focusing on sentiment analysis of tweets Twitter sentiment analysis approaches can generally be categorized into two main approaches, machine learning approach, and lexicon-based approach. In this, we use machine learning techniques to tackle twitter sentiment analysis.

1.2 Literature Survey

Literature survey is that the most vital step in software development process. Before developing the tool it's necessary to work out the time factor, economy and company strength. Once this stuff are satisfied, then next steps are to work out which OS and language are often used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support are often obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken under consideration for developing the proposed system.

• Survey on building applications on Sentiment Analysis.

• Implementation of a framework scripted in Python.

• Different online examples of Twitter Sentiment Analysis .

• Research on Machine Learning algorithms.

• Approached different Solutions for the requirement.

1.2 Project Background

The current study mainly focuses on the sentiment analysis of Twitter data (tweets) using different machine learning algorithms to deal with ordinal regression problems.Most classification algorithms are focused on predicting nominal class data labels. However, a rule for predicting categories or labels on an ordinal scale involves many pattern recognition issues. This type of problem, known as ordinal classification or ordinal regression.

1.3 Objective

Twitter Sentiment Analysis Based on Ordinal Regression, It will apply various Machine learning algoriths on tweets collected from dataset and will plot a graph regarding accuracy.In addition it will detect Sentiment type like Positive, Negative, Moderate Positive, Moderate Negative, Highly Positive,Highly Negative.

1.4 Project Description

In this section all features in application are explained in brief.

1.4.1 Load NLTK Tweets

Using this module we will load twitter sentiment corpora dataset from NLTK library.

1.4.2 Read NLTK Tweets

Using this module we will read tweets from NLTK and then clean tweets by removing special symbols, stop words and then perform stemming (stemming means removing ing or tion from words for example ORGANIZATION word will become

ORGANIZE after applying stem) on each words. Then we will calculate TFIDF vector.

1.4.3 Run SVR Algorithm

In this module we will give TFIDF vector as input to train SVR algorithm. This algorithm will take 80% vector for train and 20% vector as test. Then algorithm applied 80% trained model on 20% test data to calculate prediction accuracy.Support vector machine a set of supervised learning methods supports detection of classification, regression, and outliers that are helpful for statistical theory of learning. It is possible to extend support vector classification to solve regression problems. This technique is called Support Vector Regression (SVR)

1.4.4 Decision Tree:

Decision Trees (DTs) are a non-parametrically supervised learning algorithm that is commonly used for task classification and task regression.

1.4.5 Random Forest:

Random Forest is a technique of classification and Regression which is based on bagging of Bootstraps.Boosting and Bagging are the two commonly known techniques for classification of trees.

1.4.6 Detect Sentiment Type:

Using this module we will upload test tweets and then application will apply train model on those test tweets to predict sentiment of that tweet.

1.4.7 Accuracy Graph:

Using this module we will display accuracy graph between all algorithms.

CHAPTER - 2

SOFTWARE REQUIREMENT SPECIFICATION

2. SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Requirement Analysis

For the purpose of Twitter Sentiment Analysis we proposed a desktop application withDjango as a platform because it is one of the high-level python web framework that encourages rapid development and clean, pragmatic design.As a part of this system we are going to use Machine Learning Algorithms. The required documents for these processes are as follows.

1. Problem statement
2. Data flow diagrams
3. Use case diagram
4. Other UML diagrams.

The above mentioned documents gives us diagrammatical view of the system what we are going to develop.

2.2 Problem Statement

The problem statement concentrates on Sentiment Analysis with the help of Machine Learning Algorithms for better Accuracy.

2.3 Functional Requirements

- In software engineering, a functional requirement defines a system or its component. It describes the functions that a software must perform.
- The Application need to determine Sentiment Analysis.
- The Application must have User friendly interface to interact.
- The Application must have all the functionalties specified by the client.
- We need to use Machine learning techniques to solve regression problems to improve the sentiment analysis classification of Twitter data performance and predict new results.
- The Application must have view of Sentiments of data.
- The main task of this system is the achievement of improved results.
- It must display the Accuracy graph of different Algorithms.

2.4 Software Requirement Specification

The project is developed in Python Programming Language by using the Django.

2.4.1 Purpose

The purpose of this document is to present a detailed description of **"Twitter Sentiment Analysis based on Ordinal Regression"** application.It will explain the purpose and features of the system that it will provide, constraints under which it must operate and how the system will react. The document also describes the non functional requirements of the system.

2.4.2 Scope of the project

The Application must have four main modules.

- The first module is data acquisition, which is a process of gathering labeled tweets to perform sentiment analysis.
- The second module, this dataset undergoes various steps of preprocessing to transform and refine tweets into a data set that can be easily used for subsequent analysis.

- The third module concerns the extraction of relevant features for building a classification model. Then, the balancing and scoring tweets technique isillustrated.
- Thelastmoduleis applying different machine learning classifiers that classify the tweets into high positive, moderate positive, neutral, moderate negative, and high negative.

2.4.3 Technologies Used

- <u>PYTHON</u>

  Python is an interpreted, high-level, general-purpose programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms. Including structural (particularly procedural), object-oriented, and functional programming. Python is usually described as a "batteries included" language thanks to its comprehensive standard library.

- <u>DJANGO</u>

  Django may be a high-level Python web framework that encourages rapid development and clean, pragmatic design. Django makes it easier to create better web apps quickly and with less code. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of Don’t Repeat Yourself. Django also provides an optional administrative create, read, update and delete interface that's generated dynamically via admin models.

- <u>MACHINE LEARNING</u>

  Machine Learning (ML) is essentially that field of computing with the assistance of which pc systems can provide sense to data in much an equivalent way as citizenry do. In simple words, ML may be a sort of AI that extract patterns out of data by using an algorithm or method. The key focus of ML is to permit computer systems to find out from experience without being explicitly programmed or human intervention.

2.4.4 Overview

The application is based on Djangoplatform an application which is used for the Sentiment Analysis of Twitter data(Tweets) using various Machine Learning

Algorihms.

2.4.5 Overall Description

The objective of this document therefore is to formally describe the system’s high level requirements including functional requirements, non-functional requirements. The detail structure of this document is organized as follows:

Section 1: of this document provides an overview of the SDBA application. This includes a general description of the product, user characteristics, general constraints, and any assumptions for this system. This model demonstrates the development team understands of the application with limited fields as a proof of concept and serves to maximize the team’s ability to build a system that truly does support the business.

Section 2: presents the detail requirements, which comprise the domain.

Section 3: consists of the modeling, low level, high level designing and screen captures that demonstrates the functional requirements.

Section 4: shows testing the application using different testing techniques like load testing, performance testing, manual testing etc.

2.4.6 Specific Requirements

This section will describe the functions of actors, their roles in the system and the constraints faced by the system.

2.4.7 Product Perspective

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.

2.5 Software Requirements

The software interface is the operating system, and application programming interface used for the development of the software.

IDE: Geany/Jupyter/Pycharm

Technologies used: Python,Django,Machine Learning.

2.6Hardware Requirements

Operating System: Windows XP or higher / Mac OS X 10.5.8 or later / Linux

Processor: Pentium IV or higher

RAM :256 MB

Space on Hard Disk : Minimum 512 MB

2.7 Functional Requirements(Modules)

The Project has following modules

2.7.1 Load NLTK Tweets

Using this module we will load twitter sentiment corpora dataset from NLTK library.

2.7.2 Read NLTK Tweets

Using this module we will read tweets from NLTK and then clean tweets by removing special symbols, stop words and then perform stemming (stemming means removing ing or tion from words for example ORGANIZATION word will become ORGANIZE after applying stem) on each words. Then we will calculate TFIDF vector.

2.7.3 Run SVR Algorithm

Support vector machine is a set of supervised learning methods that supports detection of classification, regression, and outliers that are helpful for statistical theory of learning. It is possible to extend support vector classification to solve regression problems. This technique is called Support Vector Regression (SVR)

2.7.4 Decision Tree:

Decision Trees (DTs) are a non-parametrically supervised learning algorithm that is commonly used for task classification and task regression.

2.7.5 **Random Forest:**

Random Forest is a technique of classification and Regression which is based on bagging of Bootstraps.Boosting and Bagging are the two commonly known techniques for classification of trees.

2.7.6 **Detect Sentiment Type:**

Using this module we will upload test tweets and then application will apply train model on those test tweets to predict sentiment of that tweet.

2.7.7 **Accuracy Graph:**

Using this module we will display accuracy graph between all algorithms.

2.8 Non –Functional Requirements

2.8.1 Reliability:

Since the application is being developed through Python,the most famous, efficient and reliable language, so it is reliable in every aspect until and unless there is an error in the programming side. Thus the application can be a compatible and reliable one.

2.8.2 Portability:

This System must be intuitive enough such that user with average background in using computer can quickly experiment with the system and learn how to use the project. The system has user friendly interface.

2.8.3 Performance:

There are two primary factors by which Sentiment Analysis performance is measured. The first factor is the efficiency with which the application performs. The second factor is the speed.

2.8.4 Robust

It is quite robust because it's based on django precisely and it gives a lot of freedom to end user. The end user can choose what application to use either the core application or downloaded application for any activity.

2.8.5 Flexibility & Scalability

Google itself has given a set of applications with sentiment analysis but the whole developer community can develop their own applications and they have access to same resources and public API which are accessible to core applications.

2.9 External Interface Requirements

2.9.1 User Interface

A critical aspect of this project was examining how the app would look and its usability. The layout of our applications is defined in a py file.Our project interface includes Displaying a welcome ScreenIn the next screen we have different options of our project,Loading NLTK dataset,Reading data(Tweets) from the dataset,Applying SVR algorithm, Applying Random Forest Algorithm, Aplying Decision Trees, Detecting Sentiment type of Tweets, Shows Accuracy graph of all Algorithms.

2.10 Feasibility study

A key a part of the preliminary investigation that reviews anticipated costs and benefits and recommends a course of action supported operational, technical, economic, and time factors. The purpose of the study is to work out if the systems request should proceed further.

2.10.1 Organisational Feasibility

The application would contribute to the overall objectives of the organization. It would provide a quick, error free and cost effective solution to the current process marketing. It would provide an answer to several issues within the current system. As the new system is flexible and scalable it also can be upgraded and extended to satisfy other complex requirements which can be raised within the future. However it's up to the organization to upgrade or extend it.

2.10.2 Technical Feasibility

To develop this application, a high speed internet connection, django framework, knowledge of python language , Machine Learning are required. The current project is technically feasible.

2.10.3 Behavioural Feasibility

The application is behaviourally feasible since it requires no technical guidance, all the modules are user friendly and execute in a manner in the way they are designed to.

CHAPTER - 3

ANALYSIS & DESIGN

3. ANALYSIS & DESIGN

3.1 Introduction

3.1.1 Purpose

In this section the purpose of the document and the project is described.

3.1.1.1 Document Purpose

An SDD is a representation of a software system that is used as a medium for communicating software design information.

3.1.1.2 Project Purpose

Our project aims to perform sentiment analysis on Twitter data(Tweets) based on ordinal Regression.For Sentiment analysis we have used various Machine Learning Techniques. In the proposed approach, Firstly we will pre-process tweets and using a feature extraction method we will extract efficient feature.we have used Support Vector Regression (SVR) ,Decision Trees (DTs) and Random Forest(RF) for classification in Sentiment Analysis.

3.1.2 Scope

In this section the scope of the document and the project is explained in brief.

3.1.2.1 Document Scope

Our document contains a entire description of the high level architecture that will be used for developing the system. Communicating at a purposefully high level, it'll only form the idea for the Software Detailed Design and implementation. However, the SDD itself will not be in sufficient detail to implement the code. It will convey the general system design of the system, the interface design and better level module design.

Design details that will not be included in the SDD are:

• Low level classes that will be used for the implementation. The full description of the implementation of every module isn't needed, but the general public modules which will be interfaced are going to be described.

• Exact detailed description of interactions within each module.

3.1.2.2 Project Scope

Our project mainly consists of 4 modules.First module includes loading and reading the dataset.The next module includes appliying various Machine Learning algorithms like Support Vector Regression(SVR),Random Forest,Descision Trees.The next module will be used for detecting Sentiment type.The last module is used for displaying graph for various algorithms.

3.2 System Overview

3.2.1 Development Tools

Our project uses Python for development.Python is a general purpose High level Programming Language.It is a Multi-paradigm highly Interpreted Language.It provides an interactive mode in which the user can submit Commands at the python prompt.Python has a broad library support and works on various platforms such as Windows, linux, MAC etc. There are a large number of python packages available for various applications such as Machine Learning,Image Processing,Network Programming, Cryptography etc.

We have used django Framework for developing our desktop application. Django is a web development framework that helps in building and maintaining quality web applications. Django helps to eliminate repetitive tasks makes the development process easy and time saving experience.Django is based on Model Template view Architecture.It provides bridge between the data model and the database engine, and it supports a large set of database systems including MySQL, Oracle,Postgres,etc.

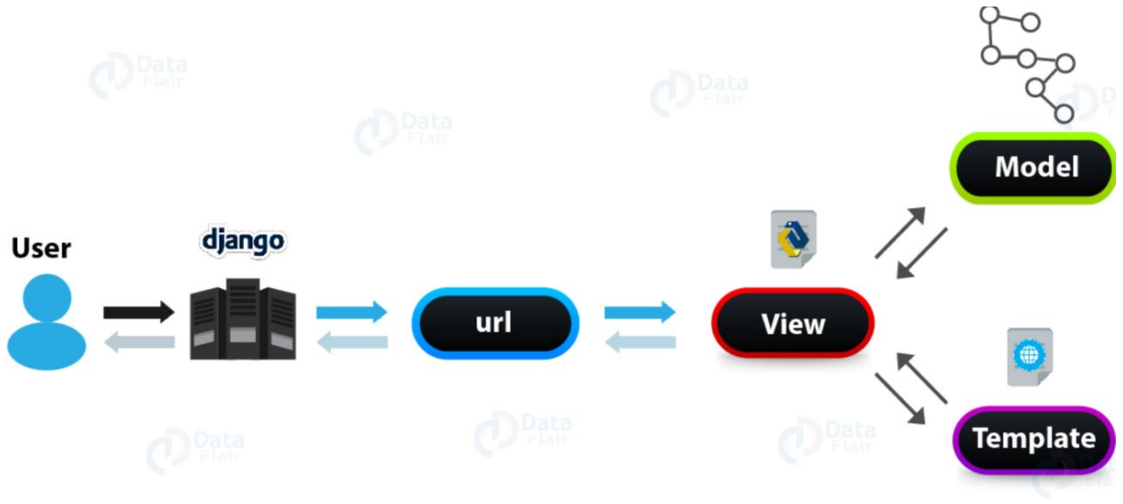


Figure 3-1 Architecture of Django

3.3 System Architecture

3.3.1 Architectural Design

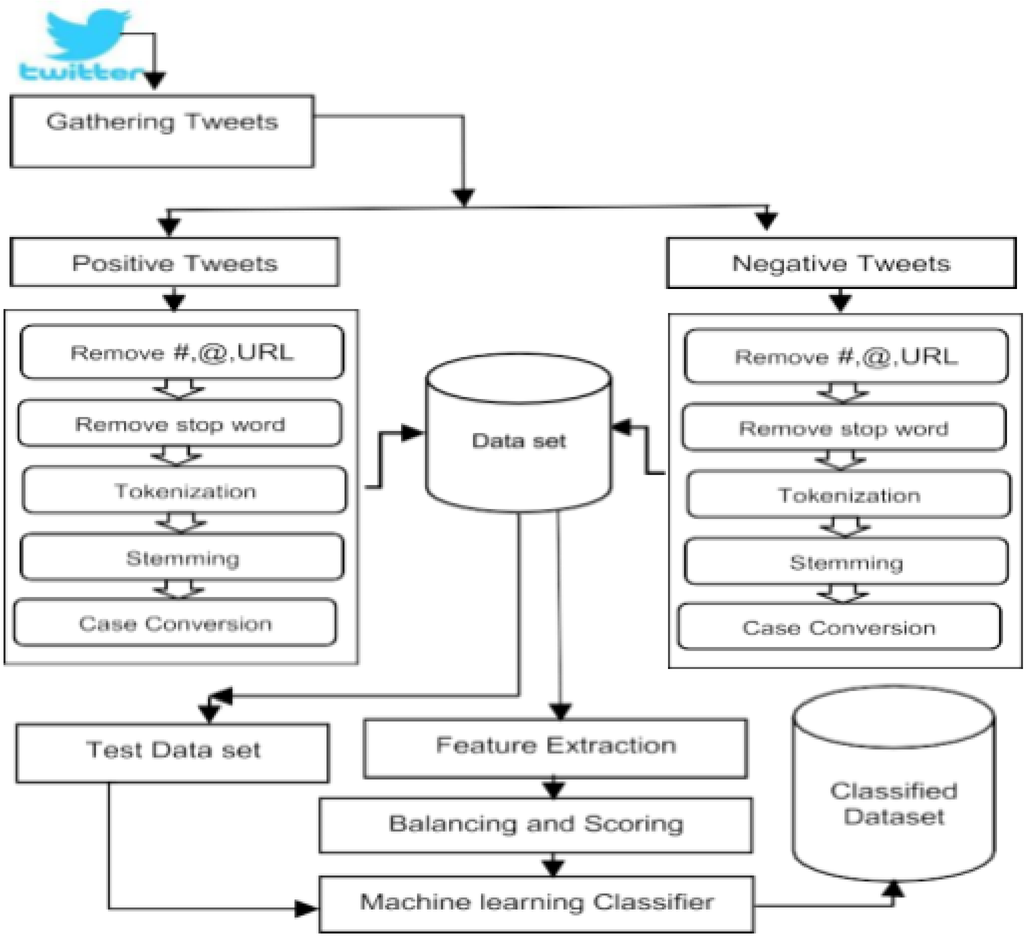


Figure 3-2 System Architecture

The proposed system is essentially composed of 4 main modules. the primary module is data acquisition, which may be a process of gathering labeled tweets to perform sentiment analysis; the second module, this dataset undergoes various steps of preprocessing to rework and refine tweets into a knowledge set which will be easily used for subsequent analysis. The third module concerns the extraction of relevant features for building a classification model. Then, the balancing and scoring tweets technique is illustrated. The last module is applying different machine learning classifiers that classify the tweets into high positive, moderate positive, neutral, moderate negative, and high negative. Figure 3-2 shows the varied steps performed for sentiment analysis using machine learning algorithms.

3.3.1.1 Dataset Collection

A dataset is collected from Twitter by using Twitter API, and therefore the data are annotated as positive or negative tweets. The dataset is publicly available within the tongue Toolkit (NLTK) corpora resource, which is well-known and extensively analyzed in many studies. the entire cor- pus size is 10000 twitter posts, consisting of 5000 positive tweets and 5000 negative tweets. For the sake of this task, we gathered and ready data sets as follows: set con- tains 8000 tweets, this set is employed for the training model. There are 2000 tweets within the other set, this set will function a test set.

3.3.1.2 Preprocessing Tweets

Raw tweets are filled with noise, misspellings, and contain numerous abbreviations and slang words. Such noisy char- acteristics often involve the performance of sentiment anal- ysis approaches. Thus, some preprocessing approaches are applied before feature extraction. The preprocessing of tweets include the subsequent steps:

- removing all hyperlinks (‘‘http://url’’), hashtags (‘‘# hashtag’’), retweet (RT), and username links (‘‘@username’’) that appear within the tweets;
- removing repeated letters (e.g., ‘‘loovee’’ becomes ‘‘love’’);
- avoiding misspellings and slang words;
- converting words into lowercase (case conversion);
- removing commonly used words that don't have spe- cial meaning, like pronouns, prepositions, and con- junctions (stop word removal);
- reducing words to their stem or common root by remov- ing plurals, genders, and conjugation (stemming);
- segmenting sentences into words or phrases called token by discarding some characters (tokenization);
- removing duplicated data tweets; and
- replacing all emoticons with their corresponding senti- ment.

3.3.1.3 Feature Extraction

In the feature extraction phase, we extract aspects or features for building a classification model. The extracted features are during a format suitable to sustain on to machine learning algorithms from datasets containing data of various formats, like text, a sequence of symbols, and pictures . Therefore, we use certain techniques to extract features from the tweets, like count vectorizer and term frequency-inverse document frequency (TF–IDF). during this study, we use TF-IDF to extract the feature matrix that reflects the importance of terms to the corpus during a text.

3.3.1.3.1 TF-IDF

TF–IDF may be a statistical measure and a term-weighting scheme that gives the bag-of-words model with information on word importance. Different aspects are extracted from the processed dataset, like verbs, adjectives, and nouns. Subsequently, these aspects are wont to calculate the sentiment polarity during a sentence to work out the opinion of people by using models, like unigrams, bigrams, or n-grams. TF–IDF is employed to guage the importance of a word to a document during a dataset. Each word is assigned a weight within the document. We use for that TF–IDF vectorizer Python module of Scikit-learn. A TF–IDF vectorizer extracts features supported word count, providing less weight to frequent words

```
Algorithm Preprocessing Tweets
 Begin
      Input Query String
      Until the data is retrieved from Twitter Streaming API, Do:
      Filter English Language Tweets Remove duplicate tweets
      Case Conversion For each tweet, Do:
      Procedure Pre-processing(tweet):
          Remove Twitter symbols(#topic, @user name, retweet (RT))
            Remove URL (‘‘http://url’’)
            Remove all symbols, numbers, Emoticons, and punctuations
            Avoiding misspellings and slang words Remove repeated letters
            Remove Stop Words Tokenization Stemming
            Return tweet
```

```
End Procedure
    Procedure Feature Extraction (clean_tweet):
        Extract Features using (TF-IDF) in a format suitable to machine learning
    algorithms
End Procedure
    Procedure Balancing and scoring(Features):
        Calculating polarity degree of the tweets,‘ balancing and labeling the
    tweets

End Procedure
    Procedure Sentiment Classification (Feature):
        Classify tweet using machine learning techniques (SoftMax, SVR,
    RF, and DT)
End Procedure
End Until End
```

The TF–IDF weight of a term within a document is Refer to ‘‘(1)’’, where *t* is a term appearing in document *d*:

TF-IDF(t,d) = TF(t,d).IDF(t)

IDF(t) = log N/IDF(t)

where TF (t, d) is that the number of times the term t appears during a particular document d , and IDF is that the total size of document N divided by the amount of documents within the entire dataset D, which contains the term t.

3.3.1.3.2 Balancing and Scoring Method

In order to create a sentiment analysis classification model, that has the most aim to research the emotions of tweets and classify them as high positive, moderate positive, neu- tral, moderate negative, and high negative consistent with the polarity of the tweet. At the primary stage of this work, after tokens are transformed into aspects or features using TF–IDF vectorizer. Then a replacement method for scoring and labeling tweets is then proposed. the tactic is consists of the subsequent steps:

Step 1: We assume that every tweet are often classified as positive and/or negative tweets counting on the amount of positive or negative words within the tweet and supported the emoticons within the tweet. this manner the polarity score of every tweet are often calculated. During the gathering time, we ignored neutral tweets because it's difficult and will affect the results.

Step 2: during this step, we determine the tweet polarity by summation the worth assigned to every feature stated within the tweet, for every tweet(t) the general polarity score value, is calculated as:

Polariy Score Tweet = Summation of n feature value(tweet)

where: *n* is the length of a tweet (*t*) (i.e., number of features in a single tweet(*t*)).

Tweet sentiment is

Moderate Positive if polarity(t)>0.0 and polarity (t) < 2.0

High Positive if polarity(t) > 0 and polarity (t) <= 4.0

Moderate Negative if polarity(t)>-2 and polarity(t) < 0.0

High Negative if polarity(t) >= −4 and polarity (t) <= −2

Neutral if polarity(t) = 0.0

3.3.1.4 Machine Learning Techniques

Two basic approaches for detecting sentiment analysis from the text (tweet), namely, Lexicon-based and machine learning approaches, are available. during this study, we use a machine learning approach. differing types of machine learning techniques are used for text classification and twitter sentiment analysis. Machine learning techniques train the algorithm with some specific training data with known outputs, thereby allowing working with new test data. Several machine learning algorithms include Support Vector Regression (SVR), Decision Trees (DTs), and Random Forest (RF) wont to build the study machine learning classifier.

3.3.1.4.1 Support Vector Regression

Support vector machine a group of supervised learning methods supports detection of classification, regression, and outliers that are helpful for statistical theory of learning. it's possible to increase support vector classification to unravel regression problems. this system is named Support Vector Regression (SVR). The SVR maintains all the most features of SVM and uses for classification an equivalent principles because the SVM, with only a couple of minor changes. Support vector regression has three different implementations: SVR, Nu-SVR and Linear SVR.

3.3.1.4.2 Decision Tree

Decision Trees (DTs) are a non-parametrically supervised learning algorithm that's commonly used for task classifi- cation and task regression. It works for categorical also as continuous variables dependent. The aim is to make a model that predicts the worth of a target variable by learning from the info inferred simple rules of decision.

3.3.1.4.3 Random Forest

Random Forest (RF) may be a technique of classification and regression supported the ensemble method, which is predicated on bagging of bootstraps.Boosting and Bagging are the 2 commonly known and used techniques for the classification of trees.The Random Forest consists of a mixture of trees which will be wont to predict the category label supported categorical variable for a specified datum .Using a Random subset of original features,each decision tree is trained.To determine the sample class, a replacement sample is employed to classify the mode of the outputs of every tree within the forest.

CHAPTER - 4

MODELING

4. MODELING

4.1 Design

Requirements gathering followed by careful analysis results in a scientific Object Oriented Design (OOAD). Various activities are identified and are represented using Unified Modeling Language (UML) diagrams. UML is employed to specify, visualize, modify, construct and document the artifacts of an object-oriented software-intensive system under development.

4.1.1 Use Case Diagram

In the Unified Modeling Language (UML), the use case diagram could also be a kind of behavioral diagram defined by and created from a use-case analysis. It represents a graphical over view of the functionality of the system in terms of actors, which are persons, organizations or external system that plays a task in one or more interaction with the system.

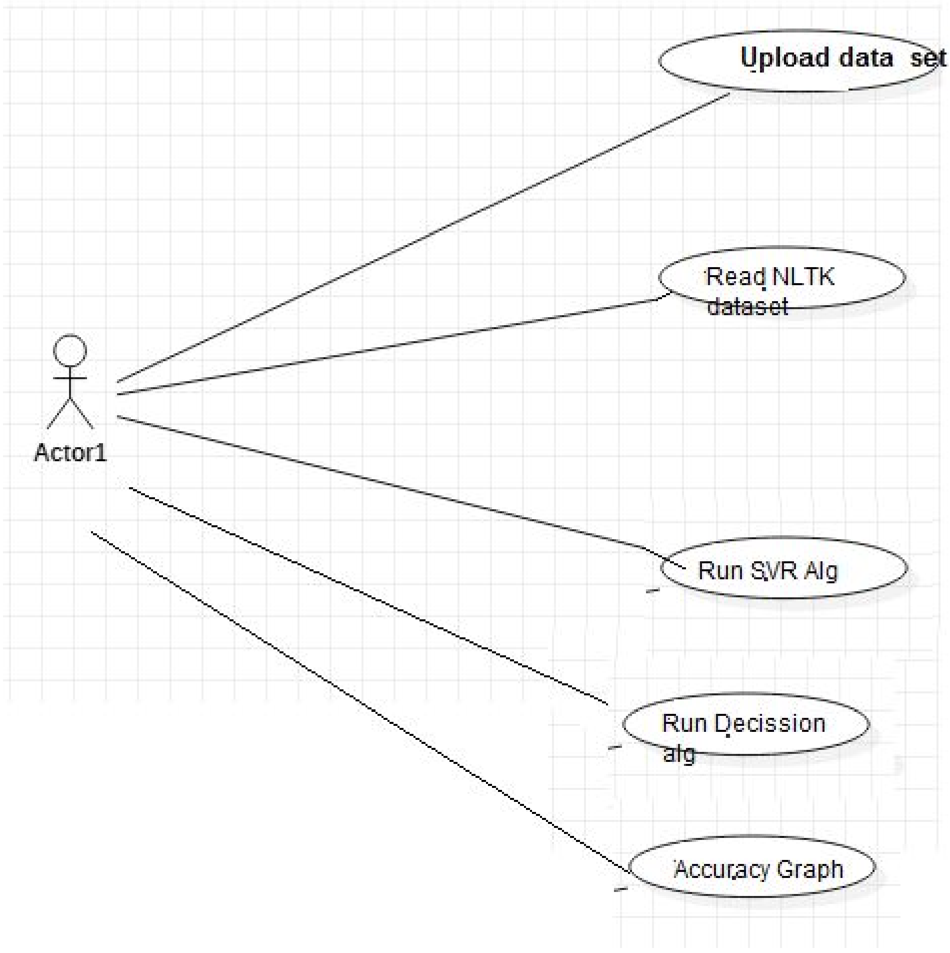


Figure 4-1 Usecase diagram for System

4.1.2 Sequence Diagram

UML sequence diagrams are wont to show how objects interact during a given situation. a crucial characteristic of a sequence diagram is that point passes from top to bottom: the interaction starts near the highest of the diagram and ends at rock bottom (i.e. Lower equals later).

A popular use for them is to document the dynamics in an object-oriented system. for every key, collaboration diagrams are created that show how objects interact in various representative scenarios for that collaboration.

Sequence diagram is that the commonest quite interaction diagram, which focuses on the message interchange between a numbers of lifelines.

The following nodes and edges are typically drawn during a UML sequence diagram: lifeline, execution specification, message, combined fragment, interaction use, state invariant, continuation, destruction occurrence.

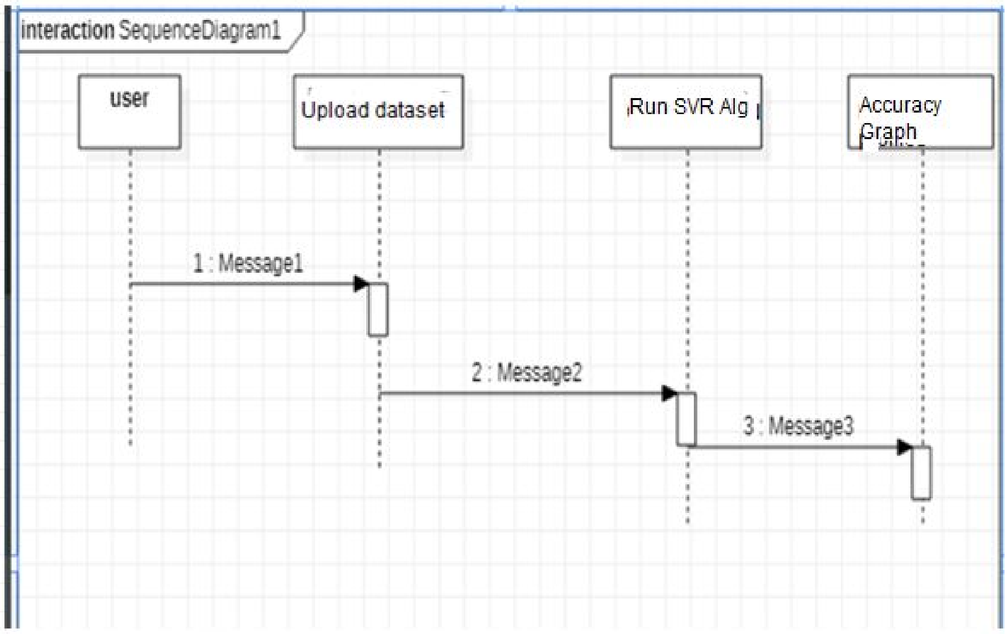


Figure 4-2 Sequence diagram for system

4.1.2 Activity Diagram

Activity diagram is another important diagram in UML to explain dynamic aspects of the system. Activity diagram is essentially a flow chart to represent the flow form one activity to a different activity. The activity are often described as an operation of the system.So the control flow is drawn from one operation to a different . This flow can be sequential, branched or concurrent. Activity diagrams deals with all sort of flow control by using different elements like fork, join etc.Activity may be a particular operation of the system.

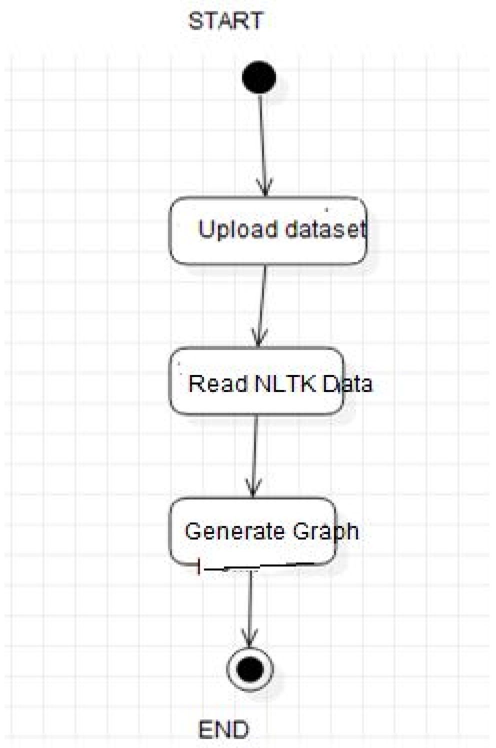


Figure 4-3 Activity diagram for System

4.1.3 Class Diagram

In software engineering, a category diagram within the Unified Modeling Language (UML) may be a sort of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and therefore the relationships among the classes.

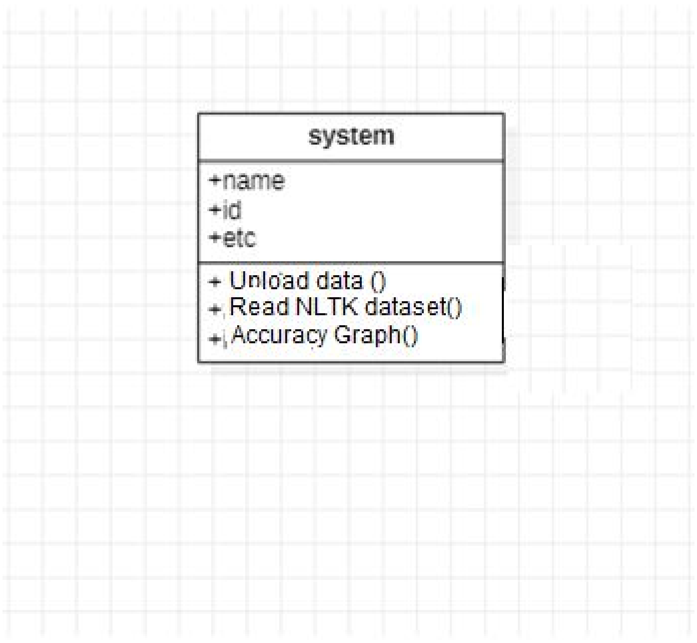


Figure 4-4 class diagram for System

CHAPTER - 5

IMPLEMENTATION

5. IMPLEMENTATION

5.1 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 sklearn import tree
from tkinter import simpledialog
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
emoticons_sad = set([
    ':L', ':-/', '>:/', ':S', '>:[', ':@', ':-(', ':[', ':-||', '=L', ':<',
    ':-[', ':-<', '=\\', '=/', '>:(', ':(', '>.<', ":'-(", ":'(", ':\\', ':-c',
    ':c', ':{', '>:\\', ';('
    ])
```

```
# all emoticons (happy + sad)
emoticons = emoticons_happy.union(emoticons_sad)
def clean_tweets(tweet):
    # remove stock market tickers like $GE
    tweet = re.sub(r'\$\w*', '', tweet)
    # remove old style retweet text "RT"
    tweet = re.sub(r'^RT[\s]+', '', tweet)
    # remove hyperlinks
    tweet = re.sub(r'https?:\/\/.*[\r\n]*', '', tweet)
    # remove hashtags
    # only removing the hash # sign from the word
    tweet = re.sub(r'#', '', tweet)
    # tokenize tweets
            tokenizer = TweetTokenizer(preserve_case=False, strip_handles=True,
reduce_len=True)
    tweet_tokens = tokenizer.tokenize(tweet)
    tweets_clean = []
    for word in tweet_tokens:
        if (word not in stopwords_english and # remove stopwords
             word not in emoticons and # remove emoticons
              word not in string.punctuation): # remove punctuation
          #tweets_clean.append(word)
          stem_word = stemmer.stem(word) # stemming word
          tweets_clean.append(stem_word)
    return tweets_clean
def bag_of_words(tweet):
    words = clean_tweets(tweet)
    words_dictionary = dict([word, True] for word in words)
    return words_dictionary
def text_processing(tweet):
    #Generating the list of words in the tweet (hastags and other punctuations removed)
    def form_sentence(tweet):
        tweet_blob = TextBlob(tweet)
        return ' '.join(tweet_blob.words)
    new_tweet = form_sentence(tweet)
    #Removing stopwords and words with unusual symbols
    def no_user_alpha(tweet):
        tweet_list = [ele for ele in tweet.split() if ele != 'user']
        clean_tokens = [t for t in tweet_list if re.match(r'[^\W\d]*$', 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'))
    for tweet in neg_tweets:
        neg_tweets_set.append((bag_of_words(tweet), 'neg'))
    text.delete('1.0', END)
```

```
        text.insert(END,"NLTK Total No Of Tweets Found : "+str(len(pos_tweets_set)+len(neg_tweets_set))+"\n")
def readNLTK():
    global msg_train, msg_test, label_train, label_test
    global test_set ,train_set
    train_tweets = pd.read_csv('dataset/train_tweets.csv')
    test_tweets = pd.read_csv('dataset/test_tweets.csv')
    train_tweets = train_tweets[['label','tweet']]
    test = test_tweets['tweet']
    train_tweets['tweet_list'] = train_tweets['tweet'].apply(text_processing)
    test_tweets['tweet_list'] = test_tweets['tweet'].apply(text_processing)
    train_tweets[train_tweets['label']==1].drop('tweet',axis=1).head()
    X = train_tweets['tweet']
    y = train_tweets['label']
    test = test_tweets['tweet']
    test_set = pos_tweets_set[:1000] + neg_tweets_set[:1000]
    train_set = pos_tweets_set[1000:] + neg_tweets_set[1000:]
     msg_train, msg_test, label_train, label_test = train_test_split(train_tweets['tweet'], train_tweets['label'], test_size=0.2)
    text.insert(END,"Training Size : "+str(len(train_set))+"\n\n")
    text.insert(END,"Test Size : "+str(len(test_set))+"\n\n")
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)
    text.insert(END,"Decision Tree Accuracy Details\n\n")
    text.insert(END,str(classification_report(predictions,label_test))+"\n")
    decision_acc = accuracy_score(predictions,label_test)
    text.insert(END,"Decision Tree Accuracy : "+str(decision_acc)+"\n\n")
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:
                msg = 'Moderate Negative'
            else:
                msg = 'Neutral'
        text.insert(END,test[i]+" == tweet classified as "+msg+"\n")
def graph():
    height = [svr_acc,random_acc,decision_acc]
    bars = ('SVR Accuracy', 'Random Forest Accuracy','Decision Tree Accuracy')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()
font = ('times', 16, 'bold')
title = Label(main, text='Twitter Sentiment Analysis Based on Ordinal Regression')
title.config(bg='LightGoldenrod1', fg='medium orchid')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)
font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=350,y=100)
text.config(font=font1)
font1 = ('times', 14, 'bold')
uploadButton = Button(main, text="Load NLTK Dataset", command=upload)
uploadButton.place(x=50,y=100)
uploadButton.config(font=font1)
readButton = Button(main, text="Read NLTK Tweets Data", command=readNLTK)
readButton.place(x=50,y=150)
readButton.config(font=font1)
svrButton = Button(main, text="Run SVR Algorithm", command=runSVR)
svrButton.place(x=50,y=200)
svrButton.config(font=font1)
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)
graphButton.config(font=font1)
main.config(bg='OliveDrab2')
main.mainloop()
```

5.2Screen Captures

5.2.1 User LoginScreen:

To run this project double click on 'Main.py' file to get below screen

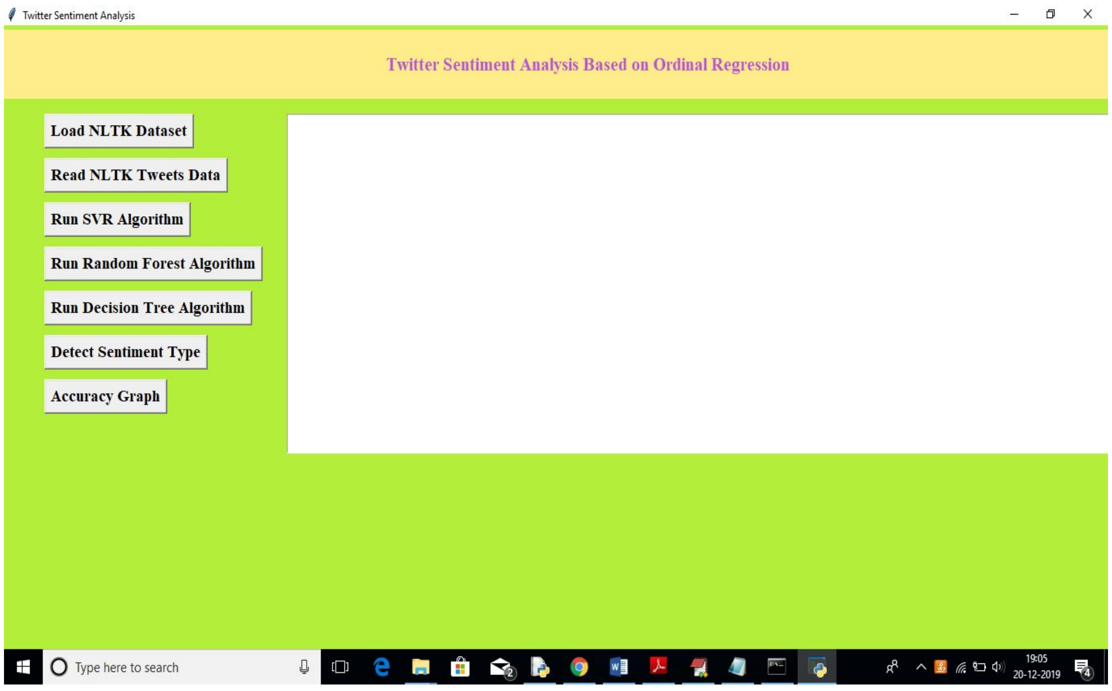
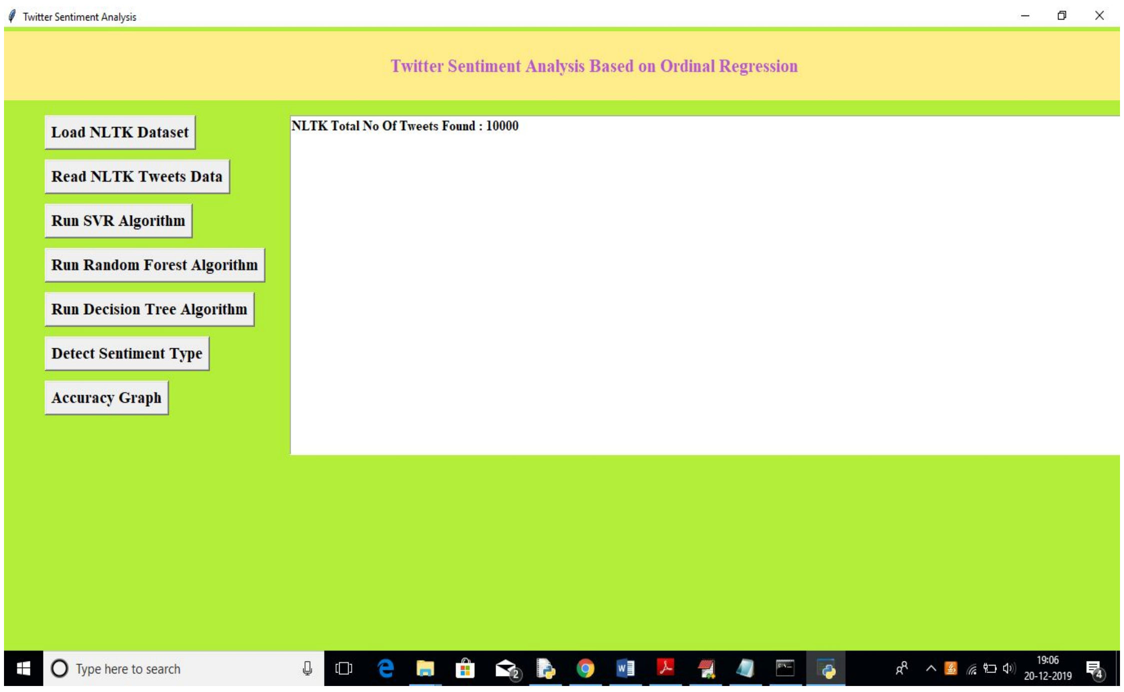

Twitter Sentiment Analysis
Twitter Sentiment Analysis Based on Ordinal Regression
Load NLTK Dataset
Read NLTK Tweets Data
Run SVR Algorithm
Run Random Forest Algorithm
Run Decision Tree Algorithm
Detect Sentiment Type
Accuracy Graph
Type here to search
19:05
20-12-2019

Figure 5-1 Welcome page

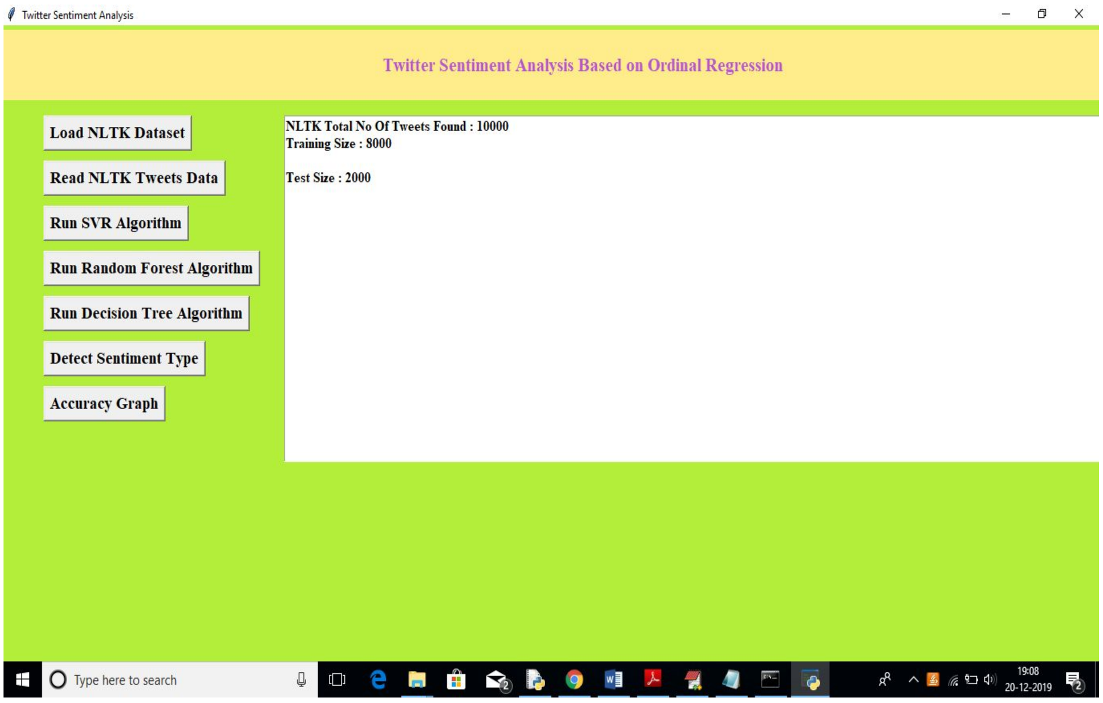
In above screen click on 'Load NLTK Dataset' to load tweets dataset from NLTK library


Twitter Sentiment Analysis
Twitter Sentiment Analysis Based on Ordinal Regression
Load NLTK Dataset
Read NLTK Tweets Data
Run SVR Algorithm
Run Random Forest Algorithm
Run Decision Tree Algorithm
Detect Sentiment Type
Accuracy Graph
Type here to search
19:06
20-12-2019

```
NLTK Total No Of Tweets Found : 10000
```

Figure 5-2 Loading dataset

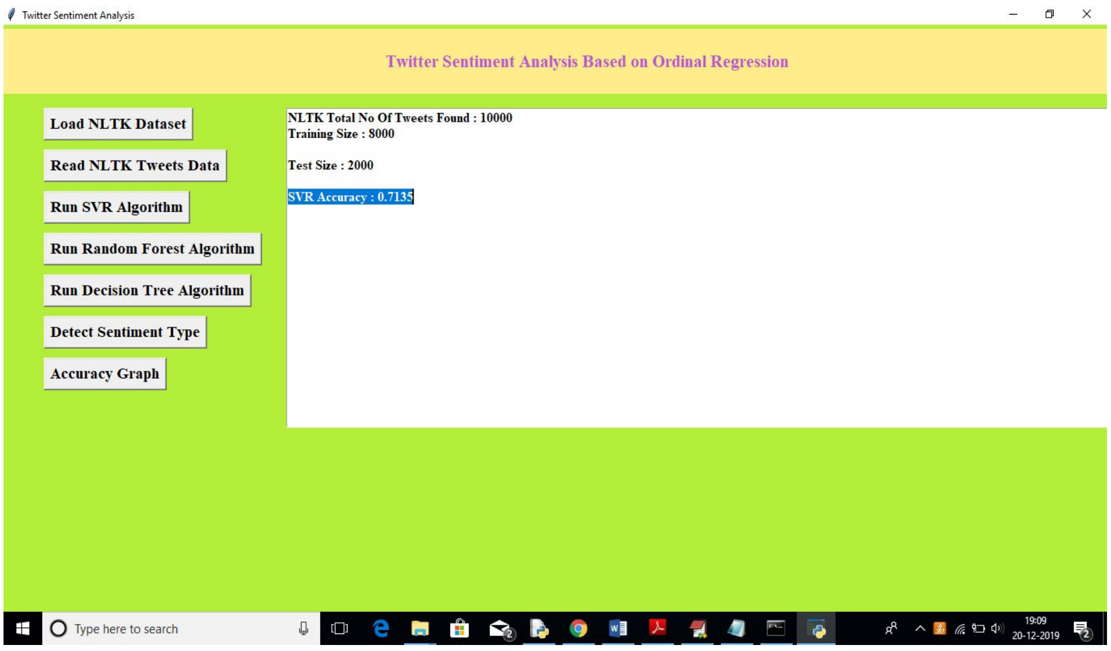
In above screen we can see total 10000 tweets are there in NLTK library, now click on ‘Read NLTK Tweets Data’ button to read all tweets and to build TFIDF vector. Upon each button click you need to wait for some seconds to get output. See below screen


Twitter Sentiment Analysis
Twitter Sentiment Analysis Based on Ordinal Regression
Load NLTK Dataset
Read NLTK Tweets Data
Run SVR Algorithm
Run Random Forest Algorithm
Run Decision Tree Algorithm
Detect Sentiment Type
Accuracy Graph
Type here to search
19:08
20-12-2019

```
NLTK Total No Of Tweets Found : 10000
Training Size : 8000

Test Size : 2000
```

Figure 5-3 Read NLTK Tweets Data

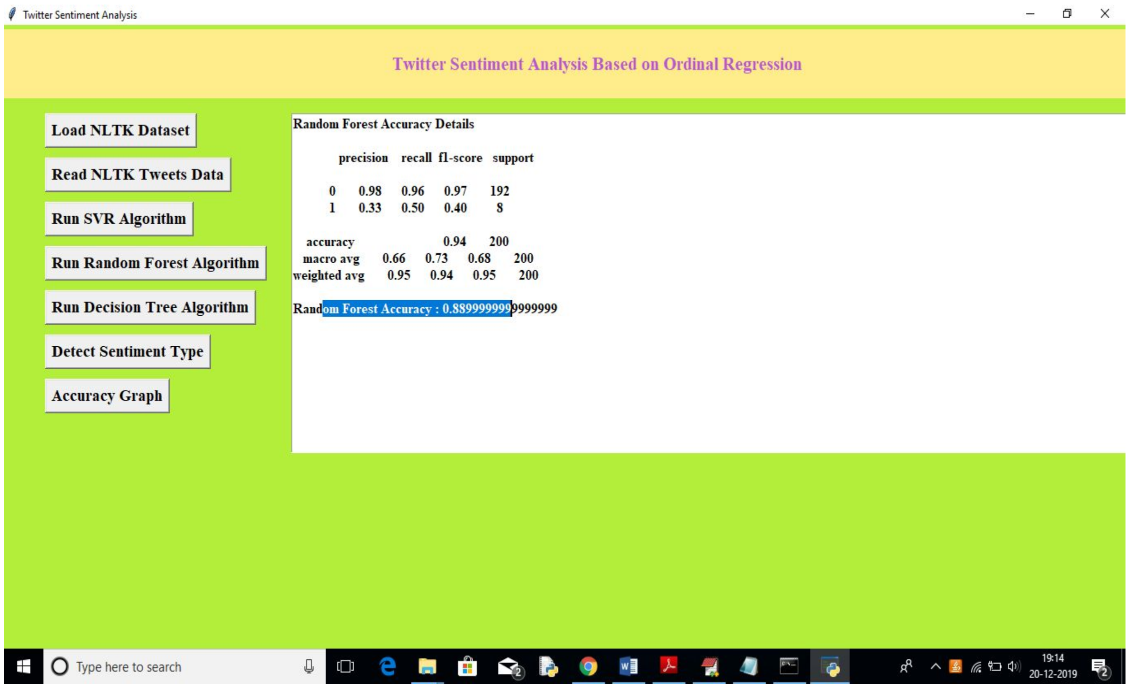
In above screen we can see total 8000 tweets vector used for training purpose and 2000 tweets used for testing purpose. Now click on 'Run SVR Algorithm' to build train model on that dataset and to calculate accuracy


Twitter Sentiment Analysis
Twitter Sentiment Analysis Based on Ordinal Regression
Load NLTK Dataset
Read NLTK Tweets Data
Run SVR Algorithm
Run Random Forest Algorithm
Run Decision Tree Algorithm
Detect Sentiment Type
Accuracy Graph
Type here to search
19:09
20-12-2019

```
NLTK Total No Of Tweets Found : 10000
Training Size : 8000

Test Size : 2000

SVR Accuracy : 0.7135
```

Figure 5-4 Running SVR Algorithm

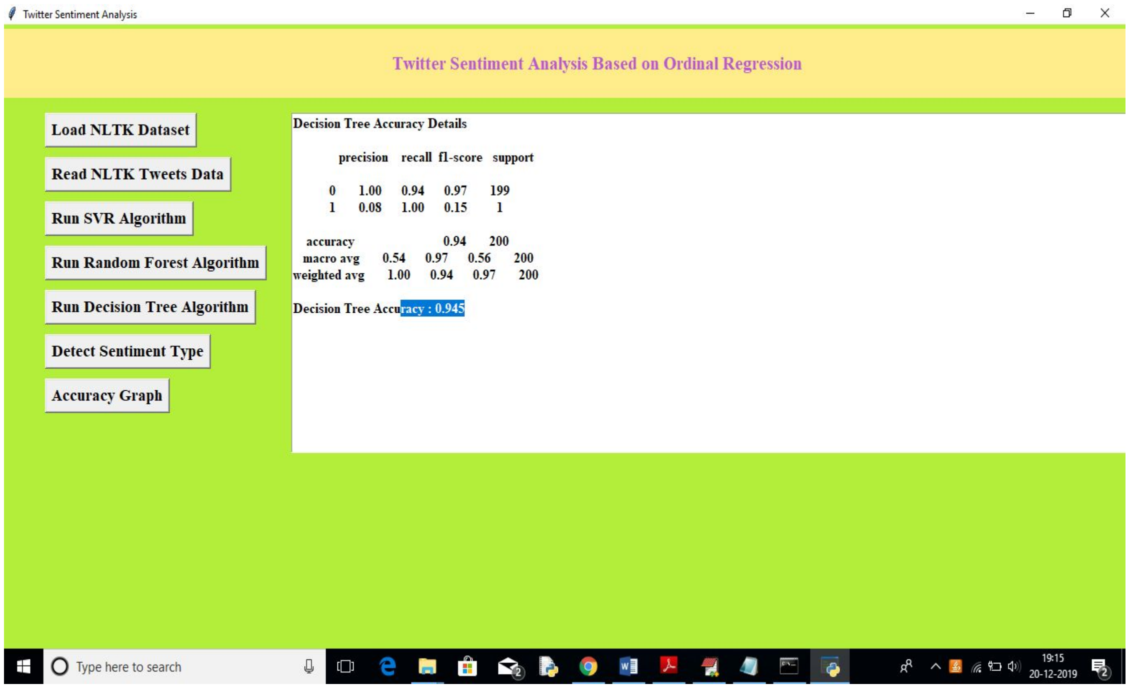
In above screen we can see SVR generate 0.71% prediction accuracy, now click on 'Run Random Forest Algorithm' button to calculate its accuracy


Twitter Sentiment Analysis
Twitter Sentiment Analysis Based on Ordinal Regression
Load NLTK Dataset
Read NLTK Tweets Data
Run SVR Algorithm
Run Random Forest Algorithm
Run Decision Tree Algorithm
Detect Sentiment Type
Accuracy Graph
Type here to search
19:14
20-12-2019

```
Random Forest Accuracy Details

        precision   recall  f1-score   support

   0      0.98      0.96      0.97      192
   1      0.33      0.50      0.40        8

  accuracy                    0.94      200
 macro avg       0.66      0.73      0.68      200
weighted avg     0.95      0.94      0.95      200

Random Forest Accuracy : 0.8899999999999999
```

Figure 5-5 Running Random Forest Algorithm

In above screen Random Forest got 0.88% accuracy, now click on 'Run Decision Tree Algorithm' button to calculate its accuracy


Twitter Sentiment Analysis
Twitter Sentiment Analysis Based on Ordinal Regression
Load NLTK Dataset
Read NLTK Tweets Data
Run SVR Algorithm
Run Random Forest Algorithm
Run Decision Tree Algorithm
Detect Sentiment Type
Accuracy Graph
Type here to search
19:15
20-12-2019

```
Decision Tree Accuracy Details

        precision   recall  f1-score   support

   0      1.00      0.94      0.97      199
   1      0.08      1.00      0.15        1

  accuracy                    0.94      200
 macro avg       0.54      0.97      0.56      200
weighted avg     1.00      0.94      0.97      200

Decision Tree Accuracy : 0.945
```

Figure 5-6 Running Decision Trees

In above screen Decision Tree got 0.94% accuracy, Now click on 'Detect Sentiment Type' button and upload test tweets to predict it sentiment. In test folder inside test.txt you can see there is no sentiment label and application will detect it.

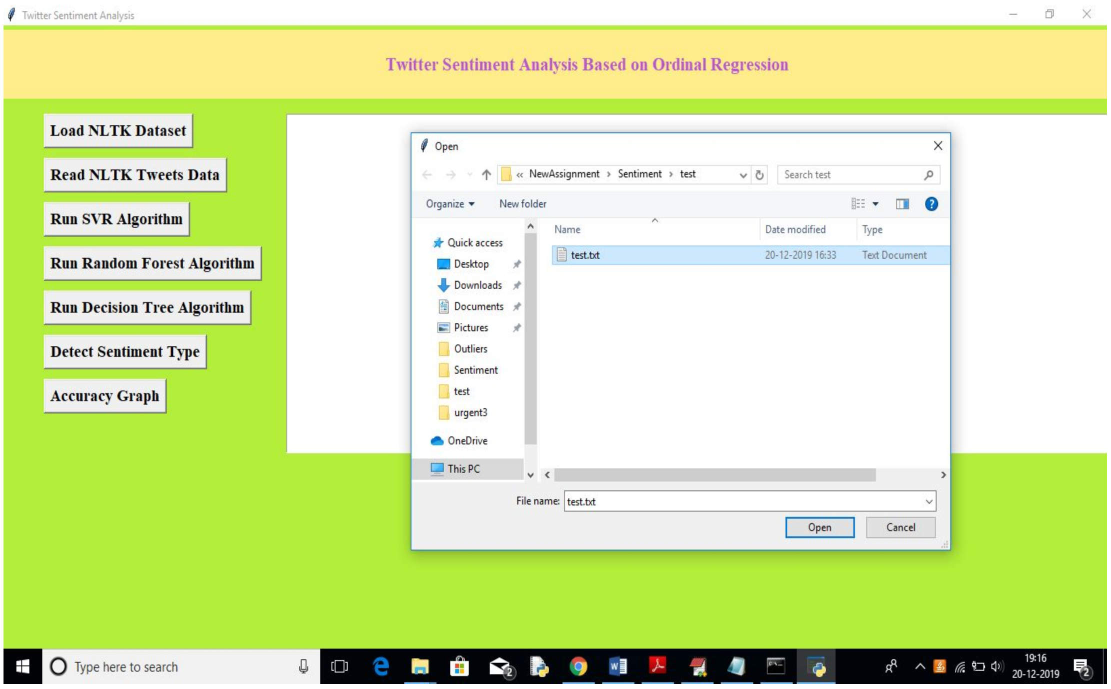
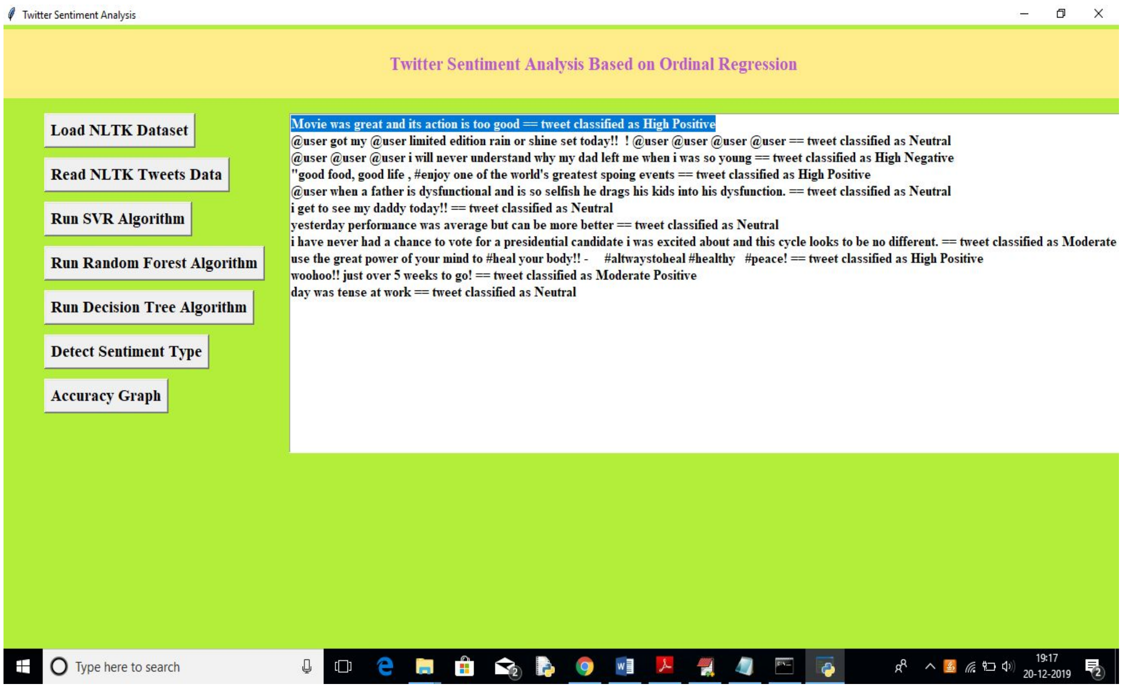


Figure 5-7 Giving file for detecing Sentiment Type

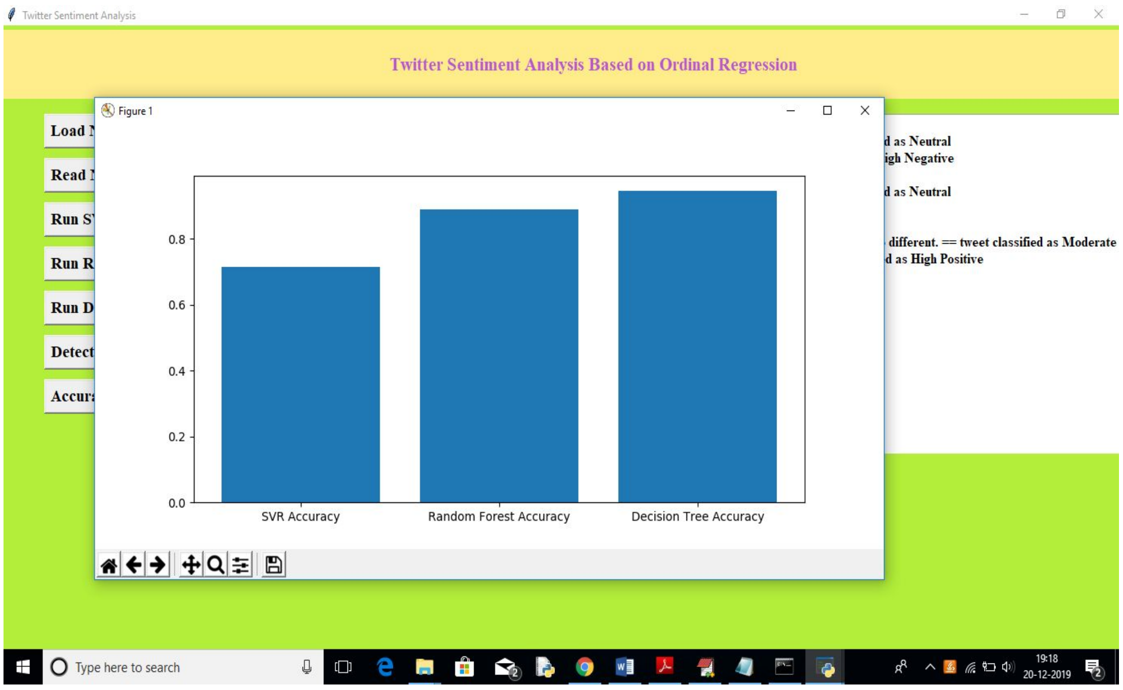
In above screen uploading test tweets file and below are the prediction results


Twitter Sentiment Analysis
Twitter Sentiment Analysis Based on Ordinal Regression
Load NLTK Dataset
Read NLTK Tweets Data
Run SVR Algorithm
Run Random Forest Algorithm
Run Decision Tree Algorithm
Detect Sentiment Type
Accuracy Graph
Type here to search
19:17
20-12-2019

```
Movie was great and its action is too good == tweet classified as High Positive
@user got my @user limited edition rain or shine set today!!  ! @user @user @user @user == tweet classified as Neutral
@user @user @user i will never understand why my dad left me when i was so young == tweet classified as High Negative
"good food, good life , #enjoy one of the world's greatest spoing events == tweet classified as High Positive
@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. == tweet classified as Neutral
i get to see my daddy today!! == tweet classified as Neutral
yesterday performance was average but can be more better == tweet classified as Neutral
i have never had a chance to vote for a presidential candidate i was excited about and this cycle looks to be no different. == tweet classified as Moderate
use the great power of your mind to #heal your body!! -     #altwaystoheal #healthy   #peace! == tweet classified as High Positive
woohoo!! just over 5 weeks to go! == tweet classified as Moderate Positive
day was tense at work == tweet classified as Neutral
```

Figure 5-8 Sentiment Types

In above screen for each tweet we can see the classified/predicted sentiments. Now click on 'Accuracy Button' to get below accuracy graph



d as Neutral
igh Negative

d as Neutral

different. == tweet classified as Moderate
d as High Positive

Figure 5-9 Accuracy Graph

In above graph x-axis represents algorithm name and y-axis represents accuracy, from above graph we can see decision tree got better prediction compare to other algorithm.

CHAPTER - 6

TESTING

6. TESTING

6.1 Software Testing

Software testing is the process of validating and verifying that a software applicationmeets the technical requirements which are involved in its design and development. It is alsoused to uncover any defects/bugs that exist in the application. It assures the quality of thesoftware. There are many types of testing software viz., manual testing, unit testing, black box testing, performance testing, stress testing, regression testing, white box testing etc.

6.2 Black box Testing

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

6.3 White box Testing

White box testing is when the tester has access to the interior data structures and algorithms including the code that implement these.

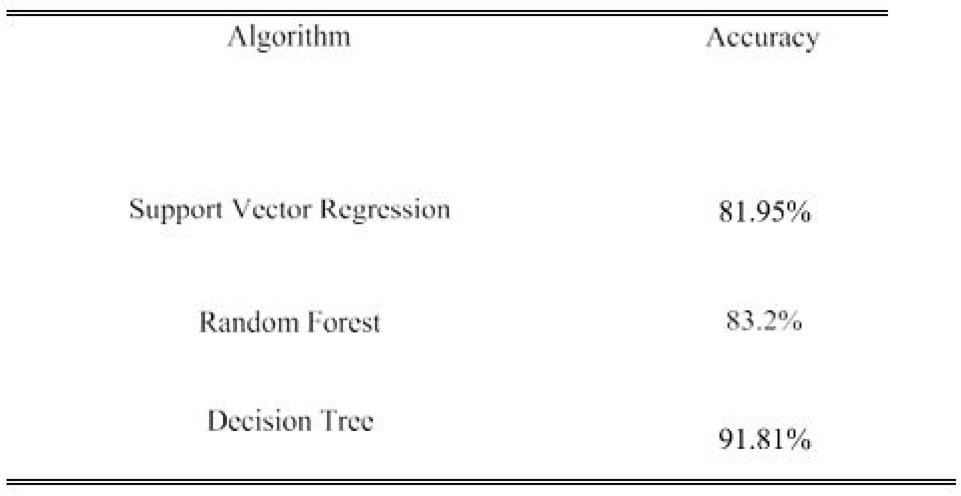
6.4 Performance Testing

Performance testing is executed to work out how briskly a system or sub-system performsunder a specific workload. It also can serve to validate and verify other quality attributes of thesystem like scalability, reliability and resource usage.

Algorithm	Score1	Score2	Score3	Score4	Score5	Score6	Score7	Score8	Score9	Score10
SVR	0.828	0.819	0.812	0.819	0.812	0.820	0.812	0.791	0.828	0.827
RF	0.699	0.728	0.730	0.744	0.720	0.745	0.729	0.734	0.755	0.751
DT	0.863	0.868	0.838	0.858	0.853	0.871	0.869	0.860	0.864	0.882

Figure 6-1 10-Fold Cross Validation Scores

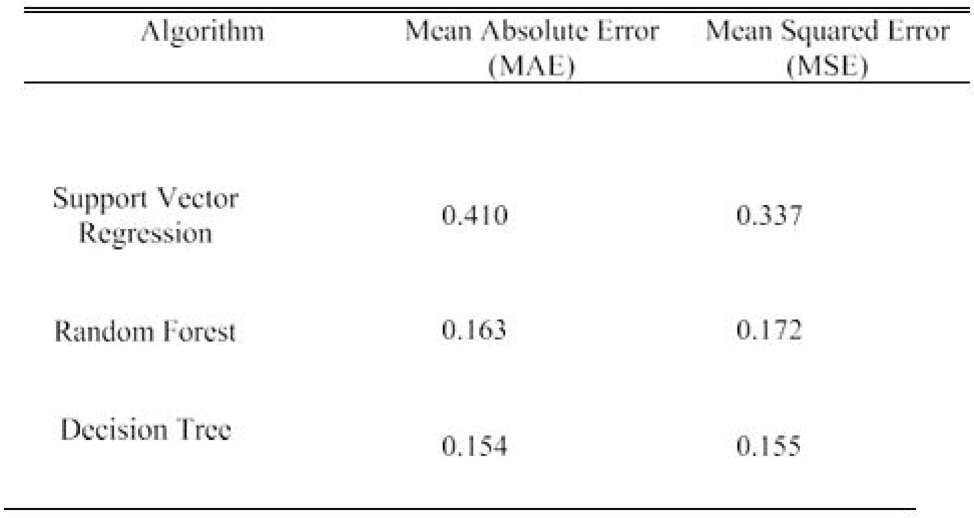
Regarding the evaluation of the methods, the overall accuracy obtained by the 10-fold cross-validation in each of the classifiers is used as one of the main classification performance metrics.Figure 6-1 shows the classification performance of all machine learning algorithms using the 10-fold cross-validation scores, which are used to partition data randomly into 10 subsets in which the class is represented in approximately the same proportions as in the fulldataset.



Algorithm	Accuracy
Support Vector Regression	81.95%
Random Forest	83.2%
Decision Tree	91.81%

Figure 6-2 Accuracy of all classifiers

In order to evaluate the efficiency of the algorithms used for ordinal regression, several evaluation metrics are used. However, Mean Square Error (MSE) and Mean Absolute Error (MAE) are the most common ones. In contrast to the accuracy measure, lower values in MAE and MSE are bet- ter because MAE and MSE are an error measurement. The resultsofapplyingtheMeanSquaredError(MSE)andMean Absolute Error (MAE) are shown in Figure 6-3.



Algorithm	Mean Absolute Error (MAE)	Mean Squared Error (MSE)
Support Vector Regression	0.410	0.337
Random Forest	0.163	0.172
Decision Tree	0.154	0.155

Table 6-3 Mean squared error and mean absolute error results

CHAPTER - 7

RESULTS &CHALLENGES

7. RESULTS AND CHALLENGES

7.1 Results

The current application is developed using python, django framework. In the current study,weusethreemachinelearningtechniques to perform sentiment analysis of Twitter data basedonordinal regression. SVR, RF, and DT are the algorithms used for categorization. Experiments are conducted using Scikit-learn , an open-source of machine learning software packages in Python. The Scikit-library provides various machine learning models to implement codes easily.

At the time of submission of my application was capable of doing the following:

- Displaying a welcome Screen
- In the next screen we have different options of our project
- Loading NLTK dataset
- Reading data(Tweets) from the dataset
- Applying SVR algorithm
- Applying Random Forest Algorithm
- Aplying Decision Trees
- Detecting Sentiment type of Tweets
- Shows Accuracy graph of all Algorithms

7.2 Challenges

- Understanding the client requirements was one of the crucial tasks of the whole project
- Detecting Sentiment type of Tweets , Understanding ML Algoithms.
- Learning different technologies and frameworks with little guidance.

CHAPTER - 7

CONCLUSIONS & FUTURE WORK

7. CONCLUSION

7.1 Conclusions

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 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, However, the Decision Tree gives the highest accuracy at 91.81%.

7.2 Scope for future work

In 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.

BIBLIOGRAPHY

Code snippets for project development

http://stackoverflow.com/

Study on Sentiment Analysis

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.

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.

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

Software Testing

http://en.wikipedia.org/wiki/Software_testing

Manual Testing

http://en.wikipedia.org/wiki/Manual_testing

Performance Testing

http://en.wikipedia.org/wiki/Software_performance_testing