REPORT

On

DEPRESSION AND DIABATES PREDICTION

Submitted by

Name of Student: DOLLY

Roll No: 171500105

Name of Student:ADITI BHATIA

Roll No: 171500

Department of Computer Engineering &

Applications

Institute of Engineering & Applications

GLA University

MATHURA-281406, INDIA ,2019

Department of computer Engineering and Applications
GLA University, Mathura
17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha, Mathura – 281406

Declaration

We hereby declare that the work which is being presented in the mini project **"**Depression and Diabates Analysis**",** in partial fulfillment of the requirements for mini project viva voce, is an authentic record of our own work carried under the supervision of " Mr.Piyush Vashistha".

Signature of Candidates:

Course: B.Tech(CSE)

Year: 3rd

Semester: V

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my Mentor, Mr. Piyush Vashistha, for providing their invaluable guidance and suggestions throughout the training session. I would also thank my college faculties and instructors who guided me in making this project.

Therefore, I am grateful to my peers at GLA University for supporting and helping me with my queries whenever I was in doubt.

Last but not least, we would like to express our deep sense of gratitude and earnest thanks giving to our dear parents for their moral support and heartfelt cooperation in doing the main project.

ABSTRACT

The objective of this project is to show how Depression Analysis can help to know the people about the dilemma they are going through . The learning algorithm will learn what our symptoms are from statistical data then determine the status of depressiveness and diabates.

After that it will change their point of view to think about things and control their mind to not involve in unimportant thoughts. Suppose you are feeling tired and sad and have some unsual thoughts like suicide , or gonna want to be dead ,then , instead of directly going to doctor about diabetes or psychiatrist about depression you can check on our project whether you are diabatic or depressed . or things are happen due to other problems like climatic changes.

The project will automatically provide you with the symptoms which leads depression and diabetes.The project aims to make people aware about their anxiety, sadness or diabetes too(in addition).

Contents

CHAPTER 1

Introduction

1.1 What is Depression Analysis?

It’s the process of using a computer to identify and categorise whether the person is in depression or not using the symptoms.

1.2 Objective

The objective of this project is to show how Depression Analysis can help to know the people about the dilemma they are going through . The learning algorithm will learn what our symptoms are from statistical data then determine the status of depressiveness and diabates.

After that it will change their point of view to think about things and control their mind to not involve in unimportant thoughts. Suppose you are feeling tired and sad and have some unsual thoughts like suicide , or gonna want to be dead ,then , instead of directly going to doctor about diabetes or psychiatrist about depression you can check on our project whether you are diabatic or depressed . or things are happen due to other problems like climatic changes.

The project will automatically provide you with the symptoms which leads depression and diabetes.The project aims to make people aware about their anxiety, sadness or diabetes too(in addition).

1.3 Basic Terms Used

i) Features

A feature is an input variable—the x variable in simple linear regression. A simple machine learning project might use a single feature, while a more sophisticated machine learning project could use millions of features .

ii) Label

A label is the thing we're predicting—the y variable in simple linear regression. The label could be the future price of wheat, the kind of animal shown in a picture, the meaning of an audio clip, or just about anything.

iii) Dataset

A data set is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question.

iv) Series

Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called index. Pandas Series is nothing but a column in an excel sheet.

v) Data Frame

DataFrame is a 2-dimensional labeled data structure with columns of potentially different types. You can think of it like a spreadsheet or SQL table, or a dict of Series objects. It is generally the most commonly used pandas object.

vi) Data Wrangling

Data Wrangling is the process of converting data from the initial format to a format that may be better for analysis.

vii) Data Pre-processing

Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends.

viii) Data Normalization

Database normalization is the process of structuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity.

ix) Data Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

x) Linear Relationship

A linear relationship (or linear association) is a statistical term used to describe a straight-line relationship between a variable and a constant. Linear relationships can be expressed either in a graphical format where the variable and the constant are connected via a straight line or in a mathematical format where the independent variable is multiplied by the slope coefficient, added by a constant, which determines the dependent variable.

xi) Correlation

Correlation is a statistical measure that indicates the extent to which two or more variables fluctuate together. A positive correlation indicates the extent to which those variables increase or decrease in parallel; a negative correlation indicates the extent to which one variable increases as the other decreases.

CHAPTER 2
SRS

2.1 Software Used

i. Anaconda Distribution

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing it is most commonly used in data science, machine learning, deep learning-related applications.

ii. Spyder

Spyder (software) Spyder is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python or R language.

iii. Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected.

It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

It is a programming language that lets you work quickly and integrate systems more efficiently.

There are two major Python versions- Python 2 and Python 3. Both are quite different.

2.2 Libraries Used

i . Data Analysis

- Pandas

  Pandas is a software library written for the Python programm- ming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series .

ii. Data Visualization

- Matplotlib

  Matplotlib is a plotting library for the Pythonprogramming language and its numerical mathematics extensionNumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits. Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.Using matplotlib you can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc.

iii. Other libraries

- tkinter

  Tkinter is a Python binding to the Tk GUI toolkit . It is the standard Python interface to the Tk GUI toolkit. Tkinter is not the only GUI Programming toolkit for Python . It is however the mostCommonly used one.

CHAPTER 3

Machine Learning

3.1 What is Machine Learning?

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer program**s** that can access data and use it to learn for themselves.

i. Types of Machine Learning Methods

- Supervised machine learning

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

- **Classification**
  It is a Supervised Learning task where output is having defined labels (discrete value). The goal here is to predict discrete values belonging to a particular class and evaluate on the basis of accuracy. It can be either binary or multi class classification. In binary classification, model predicts either 0 or 1 ; yes or no but in

multi class classification, model predicts more than one class. Example: Gmail classifies mails in more than one classes like social, promotions, updates, forum.

- **Regression**
  It is a Supervised Learning task where output is having continuous value. The goal here is to predict a value as much closer to actual output value as our model can and then evaluation is done by calculating error value. The smaller the error the greater the accuracy of our regression model.

- Unsupervised machine learning

In contrast, unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.
Unsupervised learning classified into two categories of algorithms:

- **Clustering**
  A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behaviour.

- **Association**
  An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

- Reinforcement machine learning

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

CHAPTER 4

Analyzing data with python

4.1 What is Data Analysis?

It is a process of inspecting, cleansing, transforming and modeling data with the goal of discovering useful information, informing conclusions and supporting decision-making. Data analysis has multiple approaches, applying diverse techniques under a variety of names, and is used in different business, science, and social science domains. There are a several data analysis methods including data mining, text analytics, business intelligence and data visualization.

4.2 Why Data Analysis is important?

i. Analysis of business value Chain**:** There are companies that'll help you in finding the insights of the value chains that are already there in your organization and this is going to be done through data analytics. So, the analytics will tell how the existing information is going to aid the business in finding out the gold mine that is the way to success for a company.

ii. Industry knowledge: It is another thing that you'll be able to comprehend once you get into data analytics, it is going to show how you can go about your business in the near future and what is that the economy already has its hands on. That's how you are going to avail the benefit before anyone else.

iii. Seeing the opportunities**:** As the economy keeps on changing and keeping pace with the dynamic trends is very important but at the same time profit making is one thing that an organization would most of the time aim for, Data Analytics gives us analyzed data that helps us in seeing opportunities before the time that's another way of unlocking more options.

4.3 The process of Data Analysis

i. Data requirements - The data are necessary as inputs to the analysis, which is specified based upon the requirements of those directing the analysis or customers (who will use the finished product of the analysis). Data may be numerical or categorical.

ii. Data collection - Data are collected from a variety of sources like sensors in the environment, such as traffic cameras, satellites, recording devices, etc. The requirements may be communicated by analysts to custodians of the data, such as information technology personnel within an organization.

iii. Data Processing - Data initially obtained must be processed or organized for analysis. For instance, these may involve placing data into rows and columns in a table format (i.e., structured data) for further analysis.

iv. Data Cleaning - Once processed and organized, the data may be incomplete, contain duplicates, or contain errors. The need for data cleaning will arise from problems in the way that data are entered and stored. Data cleaning is the process of preventing and correcting these errors.

v. Exploratory Data analysis - Once the data are cleaned, it can be analyzed. Analysts may apply a variety of techniques referred to as exploratory data analysis to begin understanding the messages contained in the data.

vi. Modeling and algorithms - Mathematical formulas or models called algorithms may be applied to the data to identify relationships among the variables, such as correlation or causation. In general terms, models may be developed to evaluate a particular variable in the data based on other variable(s) in the data, with some residual error depending on model accuracy.

vii. Data product - A data product is a computer application that takes data inputs and generates outputs, feeding them back into the environment. It may be based on a model or algorithm.

- Communication - Once the data are analyzed, it may be reported in many formats to the users of the analysis to support their requirements. The users may have feedback, which results in additional analysis.

4.3 Aspects

i. Importing Datasets

- Importing and exporting the data
- Understanding the data
- Importing the required datasets

ii. Data Wrangling/Data cleaning

- Identify and handle missing values
- Data Formatting
- Data Normalization
- Turning categorical values to numeric values

iii. Exploratory Data Analysis

- Descriptive statistics
- Groupby in python
- Correlation

CHAPTER 5

Project

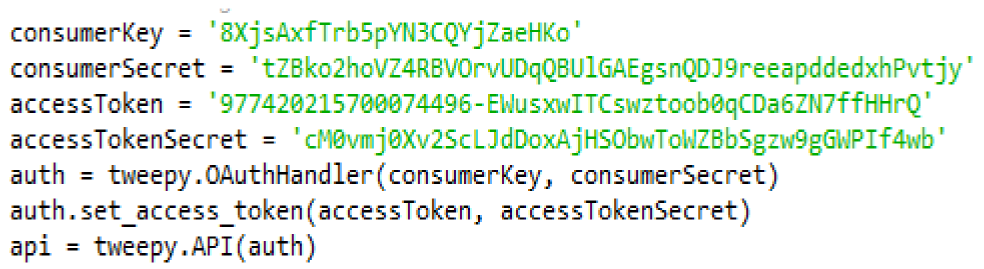
Depression is a mood disorder that may lead to severe outcomes including mental breakdown, self-injury, and suicide. Potential causes of depression include genetic, sociocultural, and individual-level factors. However, public understandings of depression guided by a complex interplay of media and other societal discourses might not be congruent with the scientific knowledge. Misunderstandings of depression can lead to under-treatment and stigmatization of depression. Against this backdrop, this study aims to achieve a holistic understanding of the patterns and dynamics in discourses about depression from various information sources in China. Our project will give provided with two datasets one is depression 2.csv and other is diabetes.csv.These datasets consists of attributes after training on it using statistical methods and Machine learning algorithm we can predict whether the person is depressed or diabatic or not.

5.3 Step 1: What? And Why?

Let us try to understand through the code session that how , why and what we done this analysis possible:

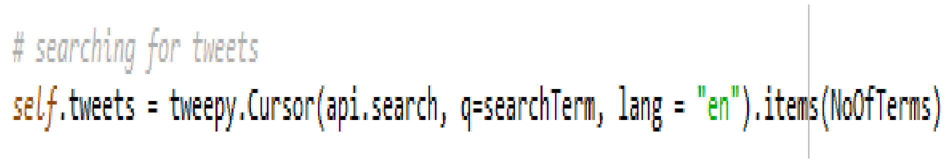
i) First of all we will import some packages named . This class contains all the methods to interact with Twitter API and parsing tweets. We use downloaddata function

- to handle the authentication of API client



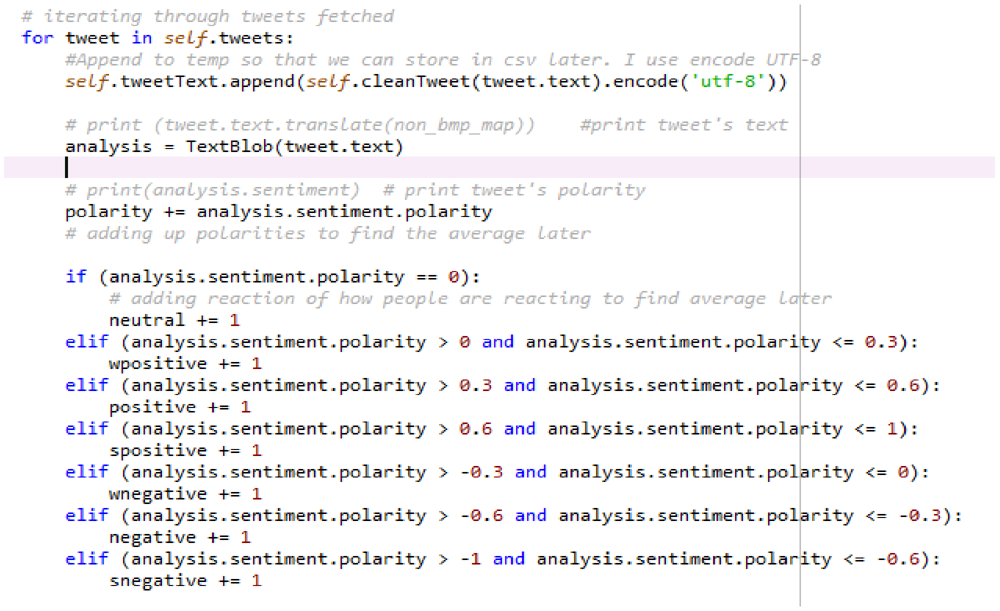
```
consumerKey = '8XjsAxfTrb5pYN3CQYjZaeHKo'
consumerSecret = 'tZBko2hoVZ4RBVOrvUDqQBUlGAEgsnQDJ9reeapddedxhPvtjy'
accessToken = '977420215700074496-EWusxwITCswztoob0qCDa6ZN7ffHHrQ'
accessTokenSecret = 'cM0vmj0Xv2ScLJdDoxAjHSObwToWZBbSgzw9gGWPIf4wb'
auth = tweepy.OAuthHandler(consumerKey, consumerSecret)
auth.set_access_token(accessToken, accessTokenSecret)
api = tweepy.API(auth)
```

- to call the twitter API to fetch tweets based on the no_of_terms given i.e.,searching for only given no_of_tweets about the given word.



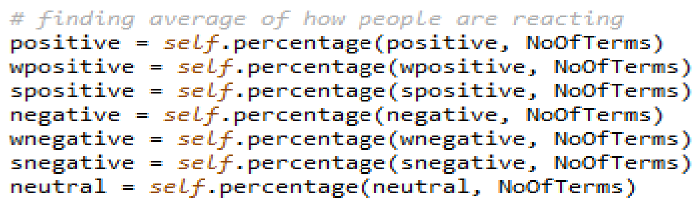
```
# searching for tweets
self.tweets = tweepy.Cursor(api.search, q=searchTerm, lang = "en").items(NoOfTerms)
```

- for parsing tweets one by one . In this we also used textblob module. TextBlob is actually a high level library built over top of NLTK library. First we call **clean_tweet** method to remove links, special characters, etc. from the tweet using some simple regex. Then, as we pass **tweet** to create a **TextBlob** object, following processing is done over text by textblob library:
    - Tokenize the tweet ,i.e split words from body of text.
    - Remove stopwords from the tokens.(stopwords are the commonly used words which are irrelevant in text analysis like I, am, you, are, etc.)
    - Do POS(part of speech) tagging of the tokens and select only significant features/tokens like adjectives, adverbs, etc.



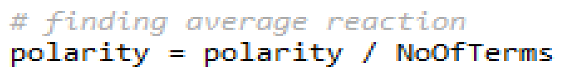
```
# iterating through tweets fetched
for tweet in self.tweets:
    #Append to temp so that we can store in csv later. I use encode UTF-8
    self.tweetText.append(self.cleanTweet(tweet.text).encode('utf-8'))

    # print (tweet.text.translate(non_bmp_map))    #print tweet's text
    analysis = TextBlob(tweet.text)

    # print(analysis.sentiment)  # print tweet's polarity
    polarity += analysis.sentiment.polarity
    # adding up polarities to find the average later

    if (analysis.sentiment.polarity == 0):
        # adding reaction of how people are reacting to find average later
        neutral += 1
    elif (analysis.sentiment.polarity > 0 and analysis.sentiment.polarity <= 0.3):
        wpositive += 1
    elif (analysis.sentiment.polarity > 0.3 and analysis.sentiment.polarity <= 0.6):
        positive += 1
    elif (analysis.sentiment.polarity > 0.6 and analysis.sentiment.polarity <= 1):
        spositive += 1
    elif (analysis.sentiment.polarity > -0.3 and analysis.sentiment.polarity <= 0):
        wnegative += 1
    elif (analysis.sentiment.polarity > -0.6 and analysis.sentiment.polarity <= -0.3):
        negative += 1
    elif (analysis.sentiment.polarity > -1 and analysis.sentiment.polarity <= -0.6):
        snegative += 1
```

- For finding average how many people are reacting using percentage( ) function



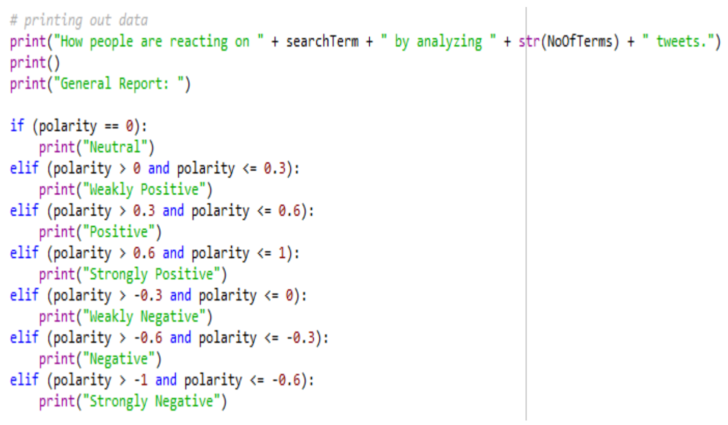
```
# finding average of how people are reacting
positive = self.percentage(positive, NoOfTerms)
wpositive = self.percentage(wpositive, NoOfTerms)
spositive = self.percentage(spositive, NoOfTerms)
negative = self.percentage(negative, NoOfTerms)
wnegative = self.percentage(wnegative, NoOfTerms)
snegative = self.percentage(snegative, NoOfTerms)
neutral = self.percentage(neutral, NoOfTerms)
```

- For finding average reactions



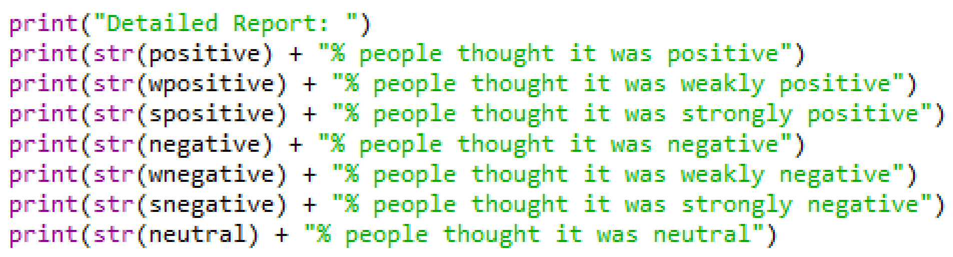
```
# finding average reaction
polarity = polarity / NoOfTerms
```

- For printing the general report



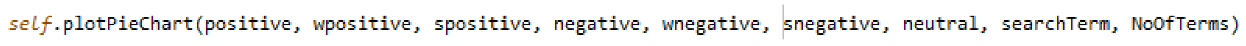
```
# printing out data
print("How people are reacting on " + searchTerm + " by analyzing " + str(NoOfTerms) + " tweets.")
print()
print("General Report: ")

if (polarity == 0):
    print("Neutral")
elif (polarity > 0 and polarity <= 0.3):
    print("Weakly Positive")
elif (polarity > 0.3 and polarity <= 0.6):
    print("Positive")
elif (polarity > 0.6 and polarity <= 1):
    print("Strongly Positive")
elif (polarity > -0.3 and polarity <= 0):
    print("Weakly Negative")
elif (polarity > -0.6 and polarity <= -0.3):
    print("Negative")
elif (polarity > -1 and polarity <= -0.6):
    print("Strongly Negative")
```

- For printing the detailed report



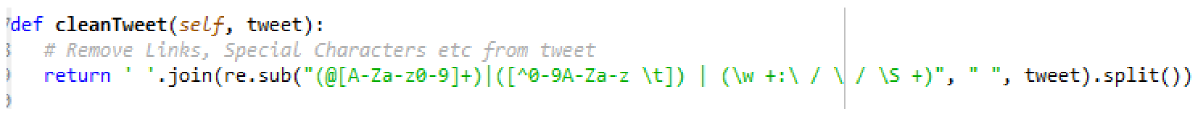
```
print("Detailed Report: ")
print(str(positive) + "% people thought it was positive")
print(str(wpositive) + "% people thought it was weakly positive")
print(str(spositive) + "% people thought it was strongly positive")
print(str(negative) + "% people thought it was negative")
print(str(wnegative) + "% people thought it was weakly negative")
print(str(snegative) + "% people thought it was strongly negative")
print(str(neutral) + "% people thought it was neutral")
```

- Then showing the pieplotchart( ) function of that detailed data



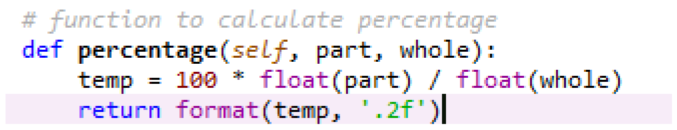
```
self.plotPieChart(positive, wpositive, spositive, negative, wnegative, snegative, neutral, searchTerm, NoOfTerms)
```

ii) Cleaning of tweets using <u>Cleantweet</u>( ) function



```
def cleanTweet(self, tweet):
    # Remove Links, Special Characters etc from tweet
    return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t]) | (\w +:\ / \ / \S +)", " ", tweet).split())
```

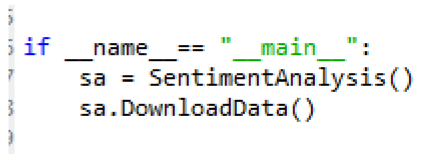
i) For finding % of the data using percentage ( ) function



```
# function to calculate percentage
def percentage(self, part, whole):
    temp = 100 * float(part) / float(whole)
    return format(temp, '.2f')
```

ii) Creating object of Sentimental Analysis class and then call Downloaddata( ) function using this object and returned the parsed tweets .Trained on Naïve Bayes Classifier.

- <u>Naïve Bayes Classifier</u>
  Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.



```
if __name__== "__main__":
    sa = SentimentAnalysis()
    sa.DownloadData()
```

CHAPTER 6

CONCLUSION

CHAPTER 7

REFERENCES

7.1 http://www.ijcaonline.org/research/volume125/number3/dandrea-2015-ijca-905866.pdf

7.2 https://textblob.readthedocs.io/en/dev/quickstart.html#sentiment-analysis

7.3 textblob.readthedocs.io/en/dev/_modules/textblob/en/sentiments.html

7.4 https://www.geeksforgeeks.org/naive-bayes-classifiers/

7.5 https://realpython.com

7.6 https://matplotlib.org/

7.7 https://pandas.pydata.org/

CHAPTER 8

APPENDIX

8.1 Python source code

```
import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset=pd.read_excel("depression 2.xls")

X = dataset.iloc[:, 0:6].values

y= dataset.iloc[:, 6].values

from sklearn.model_selection import
train_test_split


X_train,X_test,y_train,y_test=train_test_split(X
,y)


from sklearn.tree import
DecisionTreeClassifier

clf = DecisionTreeClassifier(max_depth=6)

clf.fit(X_train, y_train)


predicted = clf.predict(X_test)

expected = y_test

print(predicted)

print(expected)

from sklearn import metrics

y_pred = clf.predict(X)

print(metrics.confusion_matrix(y_pred, y))

from sklearn.metrics import precision_score,
recall_score, f1_score

a=precision_score(y,
y_pred,average='macro')

b=recall_score(y, y_pred,average='macro')

c=f1_score(y, y_pred,average='macro')

clf.score(X, y)

Hospt=input("Hospt")

#5 hospitals (1, 2, 3, 5, or 6)

Treat=input("Treat")

#The treatment received by the patient
(Lithium, Imipramine, or Placebo)

Time=input("Time")

'''Either the time (days) till recurrence

, or if no recurrence, the length (days) of the
patient's participation in the study.'''

AcuteT=input("AcuteT")

#The time (days) that the patient was
depressed prior to the study.
```

```
Gender=input("Gender")

#gender (1 = Female, 2 = Male)

T=clf.predict([[Hospt,Treat,Time,AcuteT,Age,Gender]])

# diabates

dataset=pd.read_csv("diabetes.csv")

X = dataset.iloc[:, 0:8].values

y = dataset.iloc[:, 8].values

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X,y)

from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier(max_depth=6)

clf.fit(X_train, y_train)

predicted = clf.predict(X_test)

expected = y_test

print(predicted)

print(expected)

from sklearn import metrics

y_pred = clf.predict(X)

print(metrics.confusion_matrix(y_pred, y))

from sklearn.metrics import precision_score, recall_score, f1_score

d=precision_score(y, y_pred,average='macro')

e=recall_score(y, y_pred,average='macro')

f=f1_score(y, y_pred,average='macro')

clf.score(X, y)

No_of_times_pregnant=input("No_of_times_pregnant")

glucose_concentration=input("glucose_concentration")

blood_pressure=input("blood_pressure")

skin_fold_thickness=input("skin_fold_thickness")

serum_insulin=input("serum_insulin")

BMI=input("BMI")

Diabetes_pedigree=input("Diabetes_pedigree")

Age=input("Age")

j=clf.predict([[No_of_times_pregnant,glucose_concentration,blood_pressure,skin_fold_thickness,serum_insulin,BMI,Diabetes_pedigree,Age]])

if(T==0 or j==0):

        print("person is NOT depressive and non diabatic")

elif(T==1 and j==0):

        print("DEPRESSIVE")

elif(T==0 and j==1):

         print("Diabatic")

else:

         print("both DEPRESSIVE and diabatic")
```