REPORT

On

Using textual data for Personality Prediction:A Machine Learning Approach

Submitted by

Name of Student: Deeksha Agrawal

Roll No: 171500090

Name of Student: Sonal Agrawal

Roll No: 171500342

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

I hereby declare that the work which is being presented in the B.Tech. Project "**Using textual data for Personality Prediction**", in partial fulfilment of the requirements for the award of the Bachelor of Technology in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University. Mathura, is an authentic record of my own work carried under the supervision of **Mrs.Sonia Narang**, Technical Trainer. The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Signature of Candidate:

Name of Candidate: Deeksha agrawal & Sonal agrawal

Roll. No. : 171500090 and 171500342

Course: B.Tech(CSE)

Year: 3rd

Semester:V

ACKNOWLEDGEMENT

The success and partial outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Mrs. Sonia Narang , for providing me an opportunity to do the project work in GLA University and giving me all support and guidance which made me complete the project duty. I am extremely thankful to her for providing such a nice support and guidance, although she had busy schedule managing the corporate affairs.

I owe my deep gratitude to our project guide Mrs.Sonia mam who take interest on my project work and guided me all along, till the completion of our project work by providing all the necessary information for developing a good project.

I would like to thank the lab staff for the operation extended to us throughout the project. After doing this project I can confidently say that this experience has not only enriched me with technical knowledge but also has unparsed the maturity of thought and vision. The attributes required in being a successful professional

ABSTRACT

Personality is an important parameter as it differentiates various individuals from one another. Personality prediction is an evergreen area of research. Predicting personality with the help of data through social media is a promising approach as this method does not require any questionnaires to be filled by users thus reducing time and increasing credibility.

Thus having knowledge of personality is an interesting domain for researchers to work on. Predicting personality has many applications in real world. Use of social media is increasing day by day. Huge amount of textual data as well as images continue to explode to the web daily. Current work focuses on Linear Discriminate Analysis, Multinomial Naive Bayes and AdaBoost over Twitter standard dataset.

Keywords—Machine Learning, Big Five test, Social Media, Statistical analysis.

Contents

CHAPTER 1

Introduction:

Personality is a key aspect of human life. More specifically personality is a branch of psychological study. Personality is constituted of elements like person's thoughts, feelings, behavior which continuously keeps on changing over time. Prediction of personality is an area of study where person gets categorized in a class according to his/her personality. There are number of psychological tests that yield different type of personality classes. Popular tests include Big Seven test which yield personality classification in seven categories as openness to experience, conscientiousness, extraversion, agreeableness, nervousness, sickiness and neuroticism. DISC is another test of psychology that classifies personality in categories as Dominance, Influence, Steadiness and Compliance [4]. All these traditional methods of personality prediction use questionnaire for personality prediction. Filling a lengthy questionnaire is time consuming and tedious job. Social media is a promising approach for this task. It is easier to gather a dataset from social media like Twitter or Facebook and use it for prediction. Hence in this work we have used real-time Twitter dataset for predicting personality. Twitter provides API like Twitter streaming API [7] which is useful for building dataset.

1.1 Objective

The objective of this project is to show how personality prediction can help improve the user experience over a social network or system interface. The learning algorithm will learn what our emotions are from statistical data then determine the mood. After that it will change our social interactions accordingly on our social network sites or other interfaces like desktop or system services or web-pages. Suppose you are bored or sad ,in the case of social networks one thing the computer could do is to be more

suggestive of things that lighten your mood and change interactions like backgrounds color's ,icons services. The site could automatically try suggesting interactions with people and applications that would help improve the mood, while hiding others that might make it worse. The project aims to implement these in the social network community as well as services and interfaces of our systems, while making our lives better and our experience richer and efficient.

1.2 Basic Terms Use

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 col-

lectively 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, and is likely to contain many errors.

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.

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. Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R.

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

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 programs that can access data and use it to learn for themselves.

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 accura-

cy. 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.

vii. 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

Personality prediction, also refers as opinion mining, is a sub machine learning task where we want to determine which is the general sentiment of a given document. Using machine learning techniques and natural language processing we can extract the subjective information of a document and try to classify it according to its polarity such as openness to experience, conscientiousness, extraversion, agreeableness, nervousness, sickiness and neuroticism .

It is a really useful analysis since we could possibly determine the overall opinion about a selling objects, or predict stock markets for a given company like, if most people think positive about it, possibly its stock markets will increase, and so on. Sentiment analysis is actually far from to be solved since the language is very complex (objectivity/subjectivity, negation, vocabulary, grammar,...) but it is also why it is very interesting to working on. In this project I choose to try to classify tweets from Twitter into “positive” or “negative” sentiment by building a model based on probabilities.

Twitter is a microblogging website where people can share their feelings quickly and spontaneously by sending a tweets limited by 140 characters. You can directly address a tweet to someone by adding the target sign “@” or participate to a topic by adding an hastag “#” to your tweet. Because of the usage of Twitter, it is a perfect source of data to determine the current overall opinion about anything.

Step 1: Installation

i) Tweepy: tweepy is the python client for the official Twitter API.

Install it using following pip command:

```
pip install tweepy
```

ii) TextBlob: textblob is the python library for processing textual data.

Install it using following pip command:

```
pip install textblob
```

Step 2: Authentication

In order to fetch tweets through Twitter API, one needs to register an App through their twitter account. Follow these steps for the same:

i) Open this https://developer.twitter.com and click the button: 'Create New App'

ii) Fill the application details. You can leave the callback url field empty.

iii) Once the app is created, you will be redirected to the app page.

iv) Open the 'Keys and Access Tokens' tab.

v) Copy 'Consumer Key', 'Consumer Secret', 'Access token' and 'Access Token Secret'.

Step 3: Classification of tweets

Let us try to understand through the code session that how we done this classification of Tweets possible:

i) First of all we create a class named SentimentAnalysis. This class contains all the methods to interact with Twitter API and parsing tweets. We use download data function

- to handle the authentication of API client

```
consumerKey = '8XjsAxITrb5pYN3COYjZaeHKu'
consumerSecret = 'tZBko2hoVZ4RBVOrvUDqQBUlGAEgsrQDJ9rccopddcdxh7vtjy'
accessToken = '[illegible]'
accessTokenSecret = 'cN0vnj0Xv2ScLJdDuxAjHSOLwTuNZBUSg2w8gGWPII4wb'
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.

Fig 1: Iterating through tweets fetched

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

```
# finding average of how people are reacting
neuroticsm  = self.percentage(neuroticsm , NoOfTerms)
extraversion = self.percentage(extraversion, NoOfTerms)
openness = self.percentage( openness, NoOfTerms)
conscientiousness = self.percentage(conscientiousness, NoOfTerms)
agreeableness = self.percentage(agreeableness, NoOfTerms)
sickiness = self.percentage(sickiness, NoOfTerms)
nervousness = self.percentage(nervousness, 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("nervousness")
elif (polarity > 0 and polarity <= 0.3):
    print("extraversion")
elif (polarity > 0.3 and polarity <= 0.6):
    print("neuroticsm ")
elif (polarity > 0.6 and polarity <= 1):
    print(" openness")
elif (polarity > -0.3 and polarity <= 0):
    print("agreeableness")
elif (polarity > -0.6 and polarity <= -0.3):
    print("conscientiousness")
elif (polarity > -1 and polarity <= -0.6):
    print("sickiness")
```

- For printing the detailed report

```
print("Detailed Report: ")
print(str(neuroticsn ) + "% people thought it was neuroticsn ")
print(str(extraversion) + "% people thought it was extraversion")
print(str( openness) + "% people thought it was  openness")
print(str(conscientiousness) + "% people thought it was conscientiousness")
print(str(agreeableness) + "% people thought it was agreeableness")
print(str(sickiness) + "% people thought it was sickiness")
print(str(nervousness) + "% people thought it was nervousness")
```

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

```
self.plotPieChart(neuroticsn , extraversion,  openness, conscientiousness, agreeableness, sickiness, nervousness, searchTerm, NoOfTerms)
```

ii) Cleaning of tweets using Cleantweet( ) 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())
```

iii) 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')
```

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

- Naïve Bayes Classifier:

  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

Psychology is a broad domain of study and personality prediction is its integral part. Social media is a most promising platform to determine personality of a person. In the world full of competition, everyone is not able to present his/her views to the world. Hence social media like Twitter, Facebook or Instagram etc. proves most promising solution in this scenario. People express their opinion completely on such platforms without thinking whether they are right or wrong.

Hence personality prediction helps in such scenarios where it is easy to compute sentiment of a tweet or post posted by user by applying algorithms.

As discussed in our work we have applied Multinomial Naïve Bayes, AdaBoost and LDA to compare which algorithm has higher relevance. Thus, according to our results it is found that Multinomial Naïve Bayes has highest accuracy of 73.43, precision of 0.7, and recall of 0.71 and F1-score of 0.72. Future scope aims in improving accuracy of algorithm.

CHAPTER 7

REFERENCES

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

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

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

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

5. https://realpython.com

6. https://matplotlib.org/

7. https://pandas.pydata.org/

CHAPTER 8

Appendix

```
import sys,tweepy,csv,re
from textblob import TextBlob
import matplotlib.pyplot as plt


class SentimentAnalysis:

    def __init__(self):
        self.tweets = []
        self.tweetText = []

    def DownloadData(self):
        # authenticating
        consumerKey = '8XjsAxfTrb5pYN3CQYjZaeHKo'
        consumerSecret = 'tZBko2hoVZ4RBVOrvUDqQBUlGAEgsnQDJ9reeapddedxhPvtjy'
        accessToken = '977420215700074496-EWusxwITCswztoob0qCDa6ZN7ffHHrQ'
        accessTokenSecret = 'cM0vmj0Xv2ScLJdDoxAjHSObwToWZBbSgzw9gGWPIf4wb'
        auth = tweepy.OAuthHandler(consumerKey, consumerSecret)
        auth.set_access_token(accessToken, accessTokenSecret)
        api = tweepy.API(auth)


 # input for term to be searched and how many tweets to search
```

```
searchTerm = input("Enter Keyword/Tag to search about: ")
NoOfTerms = int(input("Enter how many tweets to search: "))

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

# Open/create a file to append data to
csvFile = open('result.csv', 'a')

# Use csv writer
csvWriter = csv.writer(csvFile)



# creating some variables to store info
polarity = 0
neuroticsm = 0
extraversion = 0
openness = 0
conscientiousness = 0
agreeableness = 0
sickiness = 0
nervousness = 0



# 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
            nervousness += 1
        elif (analysis.sentiment.polarity > 0 and analysis.sentiment.polarity <= 0.3):
            extraversion += 1
        elif (analysis.sentiment.polarity > 0.3 and analysis.sentiment.polarity <= 0.6):
            neuroticsm  += 1
        elif (analysis.sentiment.polarity > 0.6 and analysis.sentiment.polarity <= 1):
            openness += 1
        elif (analysis.sentiment.polarity > -0.3 and analysis.sentiment.polarity <= 0):
            agreeableness += 1
        elif (analysis.sentiment.polarity > -0.6 and analysis.sentiment.polarity <= -0.3):
            conscientiousness += 1
        elif (analysis.sentiment.polarity > -1 and analysis.sentiment.polarity <= -0.6):
            sickiness += 1


    # Write to csv and close csv file
    csvWriter.writerow(self.tweetText)
    csvFile.close()



    # finding average of how people are reacting
    neuroticsm  = self.percentage(neuroticsm , NoOfTerms)
```

```
        extraversion = self.percentage(extraversion, NoOfTerms)
        openness = self.percentage( openness, NoOfTerms)
        conscientiousness = self.percentage(conscientiousness, NoOfTerms)
        agreeableness = self.percentage(agreeableness, NoOfTerms)
        sickiness = self.percentage(sickiness, NoOfTerms)
        nervousness = self.percentage(nervousness, NoOfTerms)

        # finding average reaction
        polarity = polarity / NoOfTerms

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

        if (polarity == 0):
            print("nervousness")
        elif (polarity > 0 and polarity <= 0.3):
            print("extraversion")
        elif (polarity > 0.3 and polarity <= 0.6):
            print("neuroticsm ")
        elif (polarity > 0.6 and polarity <= 1):
            print(" openness")
        elif (polarity > -0.3 and polarity <= 0):
            print("agreeableness")
        elif (polarity > -0.6 and polarity <= -0.3):
            print("conscientiousness")
        elif (polarity > -1 and polarity <= -0.6):
```

```
            print("sickiness")

        print()

        print("Detailed Report: ")
        print(str(neuroticsm ) + "% people thought it was neuroticsm ")
        print(str(extraversion) + "% people thought it was extraversion")
        print(str( openness) + "% people thought it was  openness")
        print(str(conscientiousness) + "% people thought it was conscientiousness")
        print(str(agreeableness) + "% people thought it was agreeableness")
        print(str(sickiness) + "% people thought it was sickiness")
        print(str(nervousness) + "% people thought it was nervousness")

            self.plotPieChart(neuroticsm , extraversion,   openness, conscientiousness, agreeableness,
sickiness, nervousness, searchTerm, NoOfTerms)



    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())

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

     def plotPieChart(self, neuroticsm , extraversion,  openness, conscientiousness, agreeableness,
sickiness, nervousness, searchTerm, noOfSearchTerms):
```

```
        labels = ['neuroticsm  [' + str(neuroticsm ) + '%]', 'extraversion [' + str(extraversion) + '%]','
openness [' + str(neuroticsm ) + '%]', 'nervousness [' + str(nervousness) + '%]',
                  'conscientiousness [' + str(conscientiousness) + '%]', 'agreeableness [' + str(agreeab-
leness) + '%]', 'sickiness [' + str(sickiness) + '%]']
          sizes = [neuroticsm , extraversion,  openness, nervousness, conscientiousness, agreeable-
ness, sickiness]
       colors = ['yellowgreen','lightgreen','darkgreen', 'gold', 'red','lightsalmon','darkred']
       patches, texts = plt.pie(sizes, colors=colors, startangle=90)
       plt.legend(patches, labels, loc="best")
          plt.title('How people are reacting on ' + searchTerm + ' by analyzing ' + str(noOfSearch-
Terms) + ' Tweets.')
       plt.axis('equal')
       plt.tight_layout()
       plt.show()



       #plt.plot([3,1,4,1,5], 'ks-', mec='w', mew=5, ms=20)
       #mpld3.enable_notebook()
       #mpld3.show()



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