# SENTIMENT ANALYSIS FRAMEWORK FOR SOCIAL MEDIA

A PROJECT REPORT

*Submitted by*

**PREMKUMAR.A**

**MUKESHWAR.S**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH,**
**COIMBATORE**

**ANNA UNIVERSITY: CHENNAI 600 025**

**APRIL 2020**

# ANNA UNIVERSITY: CHENNAI 600 025

## BONAFIDE CERITIFICATE

Certified that this project report **“SENTIMENT ANALYSIS FRAMEWORK FOR SOCIAL MEDIA”** is the bonafide work of “ **PREMKUMAR.A AND MUKESHWAR.S** ” who carried out the project work under my supervision.

**SIGNATURE**

Dr. R. MANIMEGALAI

**HEAD OF THE DEPARTMENT**

Department of Computer Science
and Engineering
PSG Institute of Technology and
Applied Research
Neelambur, Coimbatore-641 062

**SIGNATURE**

Dr. P. ILANGO

**SUPERVISOR**

PROFESSOR
Department of Computer Science
and Engineering
PSG Institute of Technology and
Applied Research
Neelambur, Coimbatore-641 062

**Submitted for the project viva-voce Examination held on ________________**

---------------------------------

**INTERNAL EXAMINER**

---------------------------------

**EXTERNAL EXAMINER**

# TABLE OF CONTENTS

# ABSTRACT

In today's environment where we're justifiably suffering from data overload, companies might have mountains of customer feedback collected but for mere humans, it's still impossible to analyze it manually without any sort of error or bias. Companies might also need to summarize the feedback into few actionable insights, so that it is meaningful for the company to make use of. In this project, we exploited the fast and in memory computation framework 'Sentiment Analysis Tool' to extract user's social media data and perform sentiment analysis. The primary aim is to provide a method for analyzing sentiment score in noisy data streams. This paper reports on the design of a sentiment analysis, extracting vast number of user data. Results classify user's perception via tweets or posts into positive and negative. Secondly, we discuss various techniques to carryout sentiment analysis on social data in detail.

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| ABBREVIATION | EXPANSION |
|---|---|
| API | Application Programming Interface |
| GUI | Graphical User Interface |
| HTML | Hyper Text Markup Language |
| JSON | Javascript Object Notation |
| VADER | Valence Aware Dictionary and sEntiment Reasoner |

# CHAPTER 1

# INTRODUCTION

Sentiment analysis refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document .The attitude may be his or her judgment or evaluation affective state, or the intended emotional communication. Humans have the innate ability to determine sentiment; however, this process is time consuming, inconsistent, and costly in a business context. It's just not realistic to have people individually read tens of thousands of user customer reviews and scores them for sentiment.

For example if we consider Semantria's cloud based sentiment analysis software, it extracts the sentiment of a document and its components through the following steps:

- A document is broken in its basic parts of speech, called POS tags, which identify the structural elements of a document, paragraph, or sentence (i.e. Nouns, adjectives, verbs, and adverbs).
- Sentiment-bearing phrases, such as "terrible service", are identified through the use of specifically designed algorithms.
- Each sentiment-bearing phrase in a document is given a score based on a logarithmic scale that ranges between -10 and 10.
- Finally, the scores are combined to determine the overall sentiment of the document or sentence Document scores range between -2 and 2.

Semantria's cloud-based sentiment analysis software is based on Natural Language Processing and delivers you more consistent results than two humans. Using automated sentiment analysis, Semantria analyzes each document and its components based on sophisticated algorithms developed to extract sentiment from your content in a similar manner as a human – only 60,000 times faster.

Different approaches to sentiment analysis can be grouped into three main categories:

- Keyword spotting
- Lexical affinity
- Statistical methods

Keyword spotting is the most naive approach and probably also the most popular because of its accessibility and economy .Text is classified into affect categories based on the presence of fairly unambiguous affect words like ‘happy’, ‘sad’, ‘afraid’, and ‘bored’ .The weaknesses of this approach lie in two areas: poor recognition of affect when negation is involved and reliance on surface features .About its first weakness, while the approach can correctly classify the sentence “today was a happy day” as being happy, it is likely to fail on a sentence like “today wasn’t a happy day at all” About its second weakness, the approach relies on the presence of obvious affect words that are only surface features of the prose.

In practice, a lot of sentences convey affect through underlying meaning rather than affect adjectives. For example, the text “My husband just filed for divorce and he wants to take custody of my children away from me” certainly evokes strong emotions, but uses no affect keywords, and therefore, cannot be classified using a keyword spotting approach.

Lexical affinity is slightly more sophisticated than keyword spotting as, rather than simply detecting obvious affect words, it assigns arbitrary words a probabilistic ‘affinity’ for a particular emotion. For example, ‘accident’ might be assigned a 75% probability of being indicating a negative effect, as in ‘car accident’ or ‘hurt by accident’ these probabilities are usually trained from linguistic corpora. Though often outperforming pure keyword spotting, there are two main problems with the approach First, lexical affinity, operating solely on the word-level, can easily be tricked by sentences like “I avoided an accident” (negation) and “I met my girlfriend by accident” (other word senses). Second, lexical affinity probabilities are often biased toward text of a particular genre, dictated by the source of the linguistic corpora. This makes it difficult to develop a reusable, domain-independent model.

Statistical methods, such as Bayesian inference and support vector machines, have been popular for affect classification of texts .By feeding a machine learning algorithm, a large training corpus of affectively annotated texts, it is possible for the system to not only learn the affective valence of affect keywords (as in the keyword spotting approach), but also to take into account the valence of other arbitrary keywords (like lexical affinity), punctuation, and word co-occurrence frequencies. However, traditional statistical methods are generally semantically weak, meaning that, with the exception of obvious affect keywords, other lexical or co-occurrence elements in a statistical model have little predictive value individually .As a result, statistical text classifiers only work with acceptable accuracy when given a sufficiently large text input .So, while these methods may be able to affectively classify user's text on the page- or paragraph- level, they do not work well on smaller text units such as sentences or clauses.

## 1.1 Objective

Sentiment classification is a way to analyze the subjective information in the text and then mine the opinion. Sentiment analysis is the procedure by which information is extracted from the opinions, appraisals and emotions of people in regards to entities, events and their attributes. In decision making, the opinions of others have a significant effect on customers ease, making choices with regards to online shopping, choosing events, products, entities.

## 1.2 Proposed Methods

Sentiment Analysis or Opinion Mining is a study that attempts to identify and analyze emotions and subjective information from text. Since early 2001, the advancement of internet technology and machine learning techniques in information retrieval make Sentiment Analysis becomes popular among researchers. Besides, the emergent of social networking and blogs as a communication medium also contributes to the development of research in this area Sentiment analysis or mining refers to the application of Natural Language Processing, Computational Linguistics, and Text Analytics to identify and extract subjective information in source materials. In recent years, sentiment analysis becomes a hotspot in numerous research fields, including natural language processing (NLP), data mining (DM) and information retrieval (IR). This is due to the increasing of subjective texts appearing on the internet.

### 1.2.1 Different approaches

There are two approaches for broadly categorizing sentiment analysis: (a) Machine Learning approach, (b) Lexicon based approach as shown in Fig. 1.

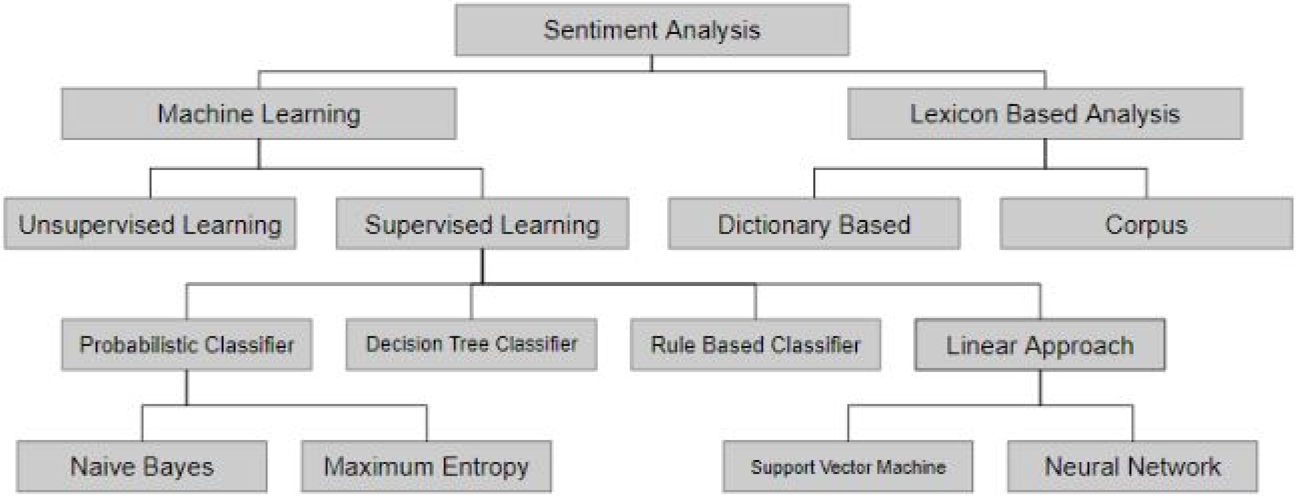


Fig 1.1 Sentiment Analysis Approaches

### A. Machine Learning Approach

Approach Machine Learning based algorithms train the classifier from manually labeled data. However, the quality and coverage of training data have a high influence to performance of the classifier. (i.e.) it requires a large database to be effective which is its only let down. This approach has better accuracy then lexicon-based.

### B. Lexicon Based Approach

This approach utilizes a sentiment lexicon to describe the polarity (positive, negative and neutral) of a textual content. This approach is more understandable and can be easily implemented in contrast to machine learning based algorithms. But the drawback is that it requires the involvement of human beings in the process of text analysis.

The more prominent the information volume, the more noteworthy the test will be for shifting through the noise, identifying the sentiment and distinguishing helpful data from various content sources. Lexicon based approach can further be divided into two categories: Dictionary based approach (based on dictionary words) and Corpus based approach (based on words in corpus).

Different algorithms have been applied so far but still the bottleneck lies in achieving remarkable accuracy. The analysis and applied processes are successful in identifying the polarity (depending on words) of a sentence but not the context i.e. a sentence can include positive words but it does not necessarily means that the sentence is positive and that will confuse the classifier.

# CHAPTER 2
# LITERATURE REVIEW

Sentiment analysis or opinion mining is the computational study of people's opinions, sentiments, attitudes, and emotions expressed in written language. It is one of the most active research areas in natural language processing and text mining in recent years.

## 2.1 Models

### 2.1.1 Naïve bayes

A naïve bayes classifier is a simple probability based algorithm. It uses the bayes theorem but assumes that the instances are independent of each other which are an unrealistic assumption in practical world naïve bayes classifier works well in complex real world situations.

The naïve bayes classifier algorithm can be trained very efficiently in supervised learning for example an insurance company which intends to promote a new policy to reduce the promotion costs the company wants to target the most likely prospects the company can collect the historical data for its customers ,including income range ,number of current insurance policies ,number of vehicles owned ,money invested ,and information on whether a customer has recently switched insurance companies .Using naïve bayes classifier the company can predict how likely a customer is to respond positively to a policy offering. With this information, the company can reduce its promotion costs by restricting the promotion to the most likely customers.

The naïve bayes algorithm offers fast model building and scoring both binary and multiclass situations for relatively low volumes of data this algorithm makes prediction using bayes theorem which incorporates evidence or prior knowledge in its prediction bayes theorem relates the conditional and marginal probabilities of stochastic events H and X which is mathematically stated as

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)\ P(H)}{P(\mathbf{X})}$$

P stands for the probability of the variables within parenthesis.

P(H) is the prior probability of marginal probability of H it's prior in the sense that it has not yet accounted for the information available in X .

P(H/X) is the conditional probability of H, given X it is also called the posterior probability because it has already incorporated the outcome of event X .

P(X/H) is the conditional probability of X given H.

P(X) is the prior or marginal probability of X, which is normally the evidence.

### 2.1.2 Vader Lexicon

Valence Aware Dictionary and Sentiment Reasoner is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. It uses a combination of sentiment lexicon which is a list of lexical features (e.g. words) which are generally labeled according to their semantic orientation as either positive or negative. It not only tells about the positivity and negativity of score but also tells us about how positive or negative a sentiment is. The model works best when applied to social media text, but it has also proven itself to be a great tool when analyzing the sentiment of movie reviews and opinion articles. The compound values generated by the sentiment intensity analyzer is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1(most extreme negative) and +1 (most extreme positive). The compounded values are further classified into three classes as shown below:

- Positive Class : score > = 0.05
- Neutral Class : score > -0.05 and score < 0.05
- Negative Class : score < = 0.05

A simple example classification using vader lexicon is shown below:

- Positive sentence example

  VADER is smart, handsome, and funny.

  ```
  {'neg': 0.0, 'neu': 0.254, 'pos': 0.746, 'compound': 0.8316}
  ```

- Punctuation emphasis sentence

  VADER is smart, handsome, and funny!

  ```
  {'neg': 0.0, 'neu': 0.248, 'pos': 0.752, 'compound': 0.8439}
  ```

- Sentence emphasized with all caps

  VADER is VERY SMART, handsome, and FUNNY.

  ```
  {'neg': 0.0, 'neu': 0.246, 'pos': 0.754, 'compound': 0.9227}
  ```

- Mixed negation sentence

  The plot was good, but the characters are uncompelling and the dialog is not great.

  ```
  {'neg': 0.327, 'neu': 0.579, 'pos': 0.094, 'compound': -0.7042}
  ```

- Sentence with emoticons

  :) and :D

  ```
  {'neg': 0.0, 'neu': 0.124, 'pos': 0.876, 'compound': 0.7925}
  ```

- Negative sentence

  Today sux

  ```
  {'neg': 0.714, 'neu': 0.286, 'pos': 0.0, 'compound': -0.3612}
  ```

- Mixed sentiment sentence

  Today kinda sux! But I'll get by, lol

  ```
  {'neg': 0.195, 'neu': 0.531, 'pos': 0.274, 'compound': 0.2228}
  ```

# CHAPTER 3
# SYSTEM ANALYSIS AND DESIGN

## 3.1 Software and Hardware Requirements

### Software Requirements

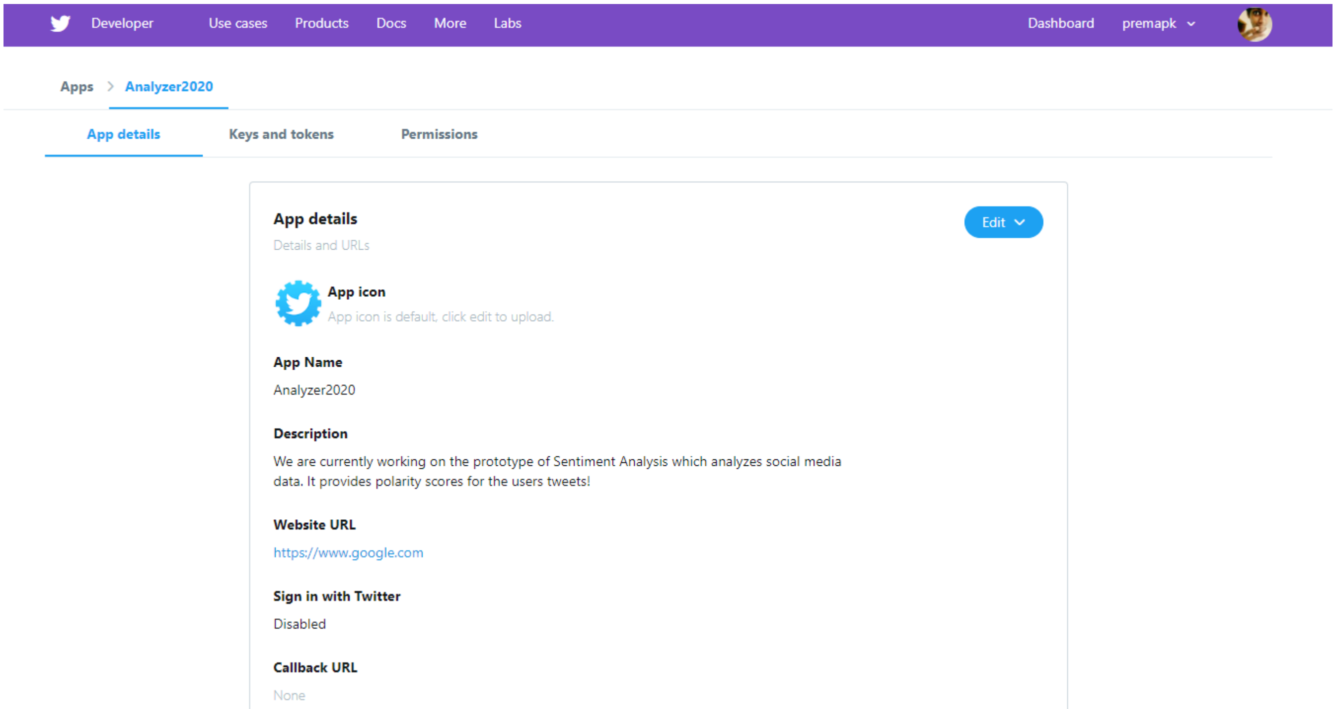
| | | |
|---|---|---|
| Operating System | : | Windows 7 and higher versions, Any Linux distributions. |
| Language | : | Python 3.0 |
| Packages Used | : | tweepy, pandas, tkinter, matplotlib, xlswriter, nltk |

### Hardware Requirements

| | | |
|---|---|---|
| RAM | : | 8GB (recommended) and more |
| Processor | : | Intel Core i5 and more |
| Disk capacity | : | 100 GB (Minimum) and more |
| Speed | : | 1GHZ and more |

## 3.2 Twitter API

Twitter's developer platform provides many API products, tools, and resources that enable you to harness the power of Twitter's open, global, and real-time communication network. There are many developer tools like Standard API, Premium API, Enterprise API, Ads API, Twitter for websites and Twitter Developer Labs. To use the API's first apply for developer account which provides access to all the developing needs. Once, you have assured developer account, you can proceed with the below procedures. Create an application for your respective project through which you can give requests and get responses from twitter. It asks all the required information like callback URL, App name, app description and usage limit. After creation of an app as shown in Fig 3.1



App details

Details and URLs

App icon

App icon is default, click edit to upload.

App Name

Analyzer2020

Description

We are currently working on the prototype of Sentiment Analysis which analyzes social media data. It provides polarity scores for the users tweets!

Website URL

https://www.google.com

Sign in with Twitter

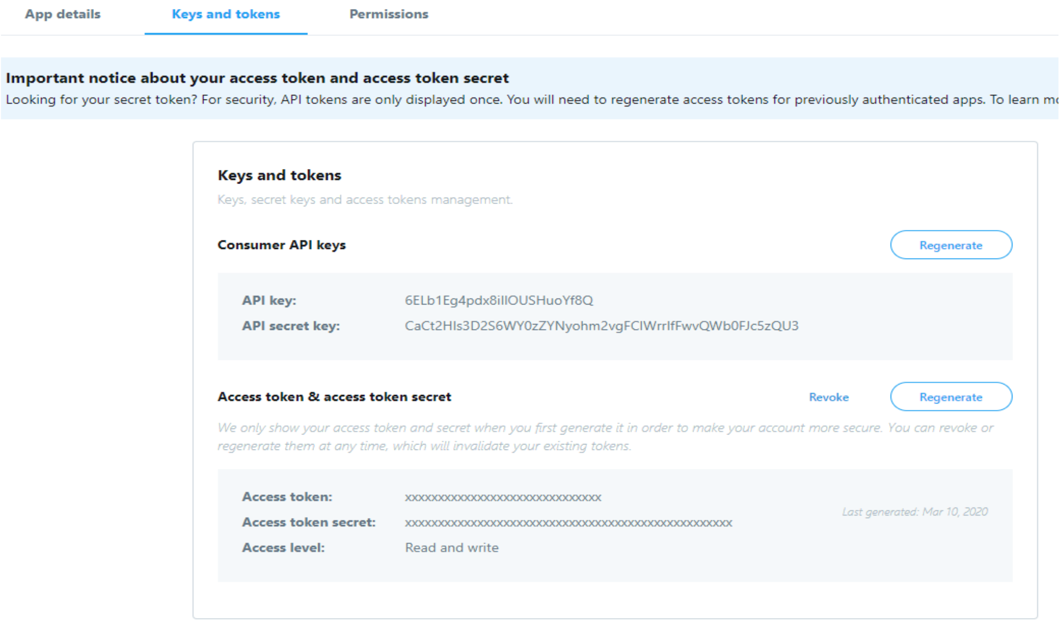
Disabled

Callback URL

None

Fig 3.1 Twitter API with Demo Application

Go to the App details of your respective application and choose Keys and tokens. To use the Twitter API, You have to include these keys in your program file so that you can request anything from the twitter account. Accessing keys and tokens page was displayed in Fig 3.2.



Important notice about your access token and access token secret

Looking for your secret token? For security, API tokens are only displayed once. You will need to regenerate access tokens for previously authenticated apps. To learn mo

Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

API key: 6ELb1Eg4pdx8illOUSHuoYf8Q

API secret key: CaCt2HIs3D2S6WY0zZYNyohm2vgFCIWrrlfFwvQWb0FJc5zQU3

Access token & access token secret

*We only show your access token and secret when you first generate it in order to make your account more secure. You can revoke or regenerate them at any time, which will invalidate your existing tokens.*

Access token: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Access token secret: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

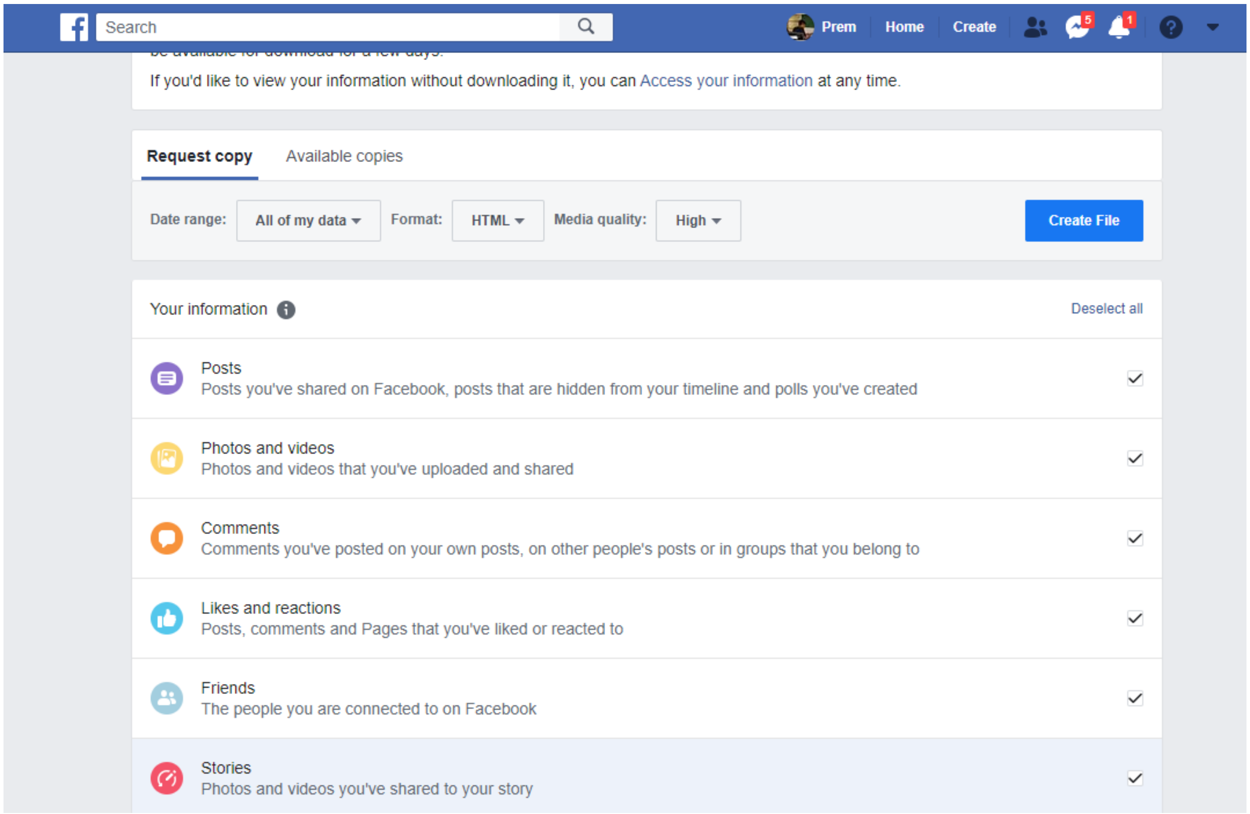
Access level: Read and write

*Last generated: Mar 10, 2020*

Fig 3.2 Accessing keys and tokens

## 3.3 Facebook data retrieval

Facebook's data can be retrieved easily without any authentication by keys. It provides an API which in turn helps the users to store their user information from a given input range. You can download several features like posts, photos and videos, comments, likes and reactions, friends list, stories, following and followers list, messages and events. You can download the data in either JSON or HTML format. The required soft copy will be available within a short span of time. You can download your data only when you have account in facebook. After signing in, go to the Settings and then to download your information tab in your left scroll view. The download page is shown in Fig 3.3



be available for download for a few days.

If you'd like to view your information without downloading it, you can Access your information at any time.

Request copy | Available copies

Date range: All of my data | Format: HTML | Media quality: High | Create File

Your information | Deselect all

Posts
Posts you've shared on Facebook, posts that are hidden from your timeline and polls you've created ✓

Photos and videos
Photos and videos that you've uploaded and shared ✓

Comments
Comments you've posted on your own posts, on other people's posts or in groups that you belong to ✓

Likes and reactions
Posts, comments and Pages that you've liked or reacted to ✓

Friends
The people you are connected to on Facebook ✓

Stories
Photos and videos you've shared to your story ✓

Fig 3.3 Facebook data download page

## 3.4 Activity Diagram

An activity diagram is a behavioral diagram. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram.

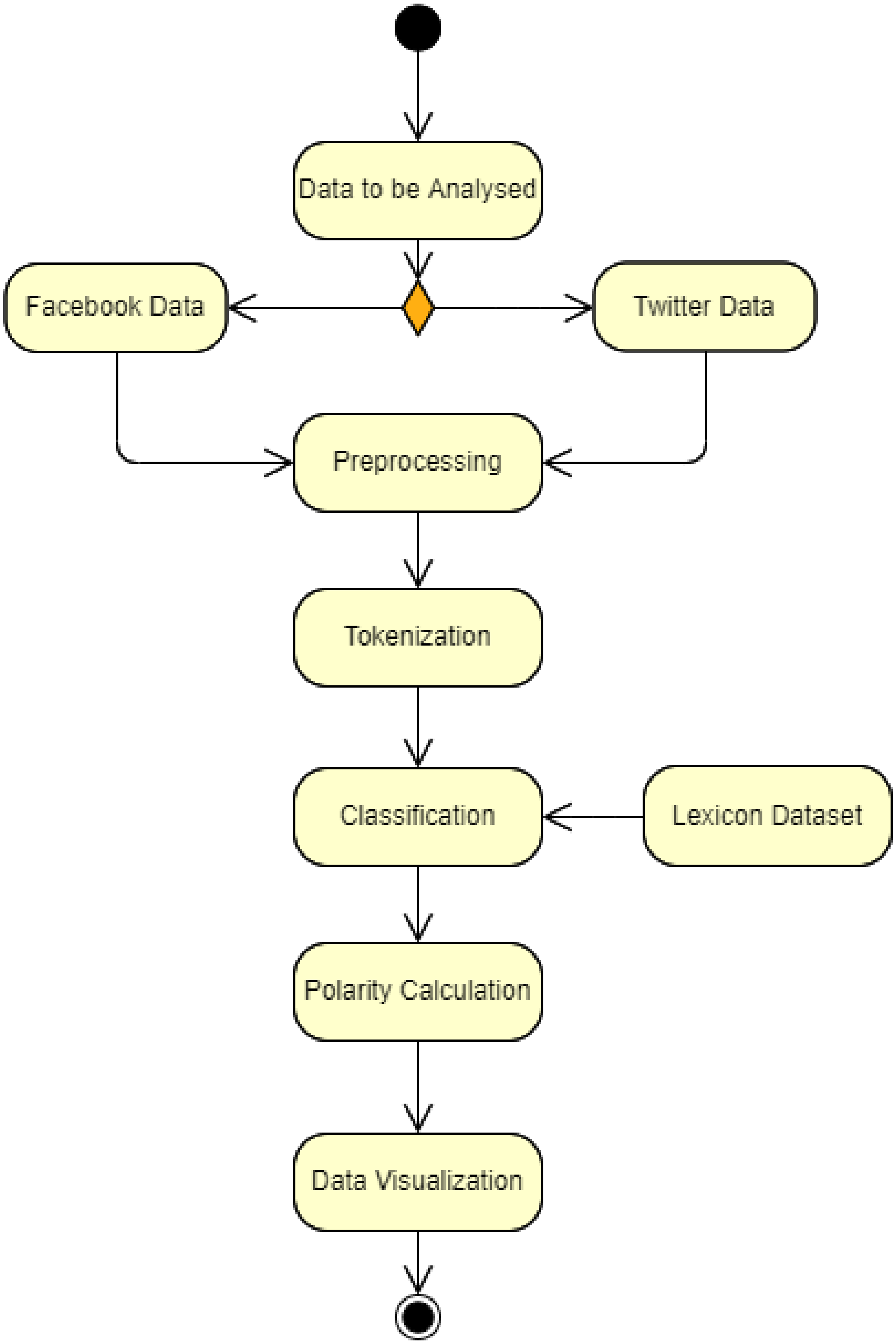


Fig 3.4 Activity diagram for proposed model

# CHAPTER 4
# IMPLEMENTATION

We used python (version 3.0) with anaconda for the implementation of “sentiment analysis of social media” using lexicon based approach. There are several stages involved during implementation of our problem. Among them, classification phase is the main phase.

## 4.1 Tokenization

Tokenization is the process of tokenizing or splitting a string, text into a list of tokens. One can think of token as parts like a word is a token in a sentence, and a sentence is a token in a paragraph. Implementation is shown below:

```
from nltk.tokenize import sent_tokenize
text = "Hello everyone. How was your day?"
word_tokenize(text)
```

The above code will tokenize the sentence into a list of words. Implicitly, word_tokenize function is a wrapper function that calls tokenize() on an instance of the TreebankWordTokenizer. It works by separating the words using punctuation and spaces.

## 4.2 Data Preprocessing

It is the stage where the raw data is transformed into useful and efficient format. It involves various methods like data cleaning - removing the noisy data, data transformation – normalization, etc…

### 4.2.1 Data Transformation

Here, the data gathered from facebook and twitter are available in HTML or JSON format. Thus, we have to extract the data from the webpages and save it in Excel file so that we can perform data cleaning in an easier manner. This extraction involves retrieving the data from the division tag of required classes in case of HTML whereas in JSON, we extract the values of respective keys. Code for this implementation is shown below:

```
Import xlswriter
while c != "b'":
        c = str(f1.read(1))
        char = c[2:3]
        if c == -1:                                  break
        if (char == "<")and(flag == 0):               flag = 1
        elif (char == "d")and(flag == 1):             flag = 2
         elif(char == "i")and(flag == 2)             flag = 3
        elif(char == "v")and(flag == 3):             flag = 4
        elif(char == ">" and flag == 4 and input_1==1)or(char ==" " and flag==4 and
input_1==0):
           if(char==">" and input_1==1):             flag = 5
           else:                                      flag=11
        elif ((char=="c" and flag==11 and input_1==0))or((char != "<" and flag == 5 and
input_1==1)and((char.isalpha() == True)or(char == " ")or(char.isdigit() == True)or(char
== ":")or(char == "."))):
           if(flag==11 and input_1==0):               flag=6
           else:        char1 = char1 + char          flag1 = 1
        elif(char=="l")and(flag==6)and input_1==0:
           flag=7
        elif (char == "a") and (flag == 7):           flag = 15
        elif (char == "s") and (flag == 15):          flag = 16
        elif (char == "s") and (flag == 16):          flag = 17
        elif (char == "=") and (flag == 17):           flag = 18
```

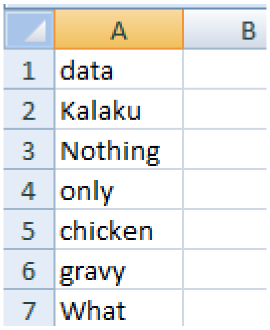
```
            elif (char == "\"") and (flag == 18):           flag = 19
            elif(char == "_")and(flag==19):                   flag=8
            elif(char == "3")and(flag==8):                  flag=13
            elif((char == "-")and(flag==13))or(flag==9)or(flag==12):
                if(char == "-")and(flag==8):                  flag=9
                elif(char==">" and flag==9):                  flag=12
                elif(flag==12):
                    if((char.isalpha() == True)or(char == " ")or(char.isdigit() == True)  or(char
== ":")or(char == "."))and(char!="<"):                char1 = char1 + char
                        flag1 = 1
                    elif(char=="<"):                              flag=14
                else:                                              flag=9
            else:                                                   flag = 0
                if flag1 == 1:          flag1 = 0             flag2 = 1

          if flag2 == 1:
                worksheet.write(j,i,char1)
                j = j+1
                char1 = ""
                flag2 = 0
        workbook.close()
        f1.close()
```

### 4.2.2 Data cleaning

The data can have many irrelevant and missing parts. To handle this part, data cleaning should be done. It involves handling of missing data, noisy data etc… Here, after creating an excel file with required data, the blank spaces and punctuations are removed with the help of pandas library. Implementation is shown below:

```
import pandas as pd
File_name = 'new.xlsx'
data = pd.ExcelFile(file_name)
parse_data = data.parse(data.sheet_names[0])
filter_data = list(parse_data['data'])
```

The above code will parse the data into excel sheet in the first column with Heading as data. Data after preprocessing and cleaning is shown in the fig 4.1.



|  | A | B |
|---|---|---|
| 1 | data |  |
| 2 | Kalaku |  |
| 3 | Nothing |  |
| 4 | only |  |
| 5 | chicken |  |
| 6 | gravy |  |
| 7 | What |  |

Fig 4.1 Preprocessed data

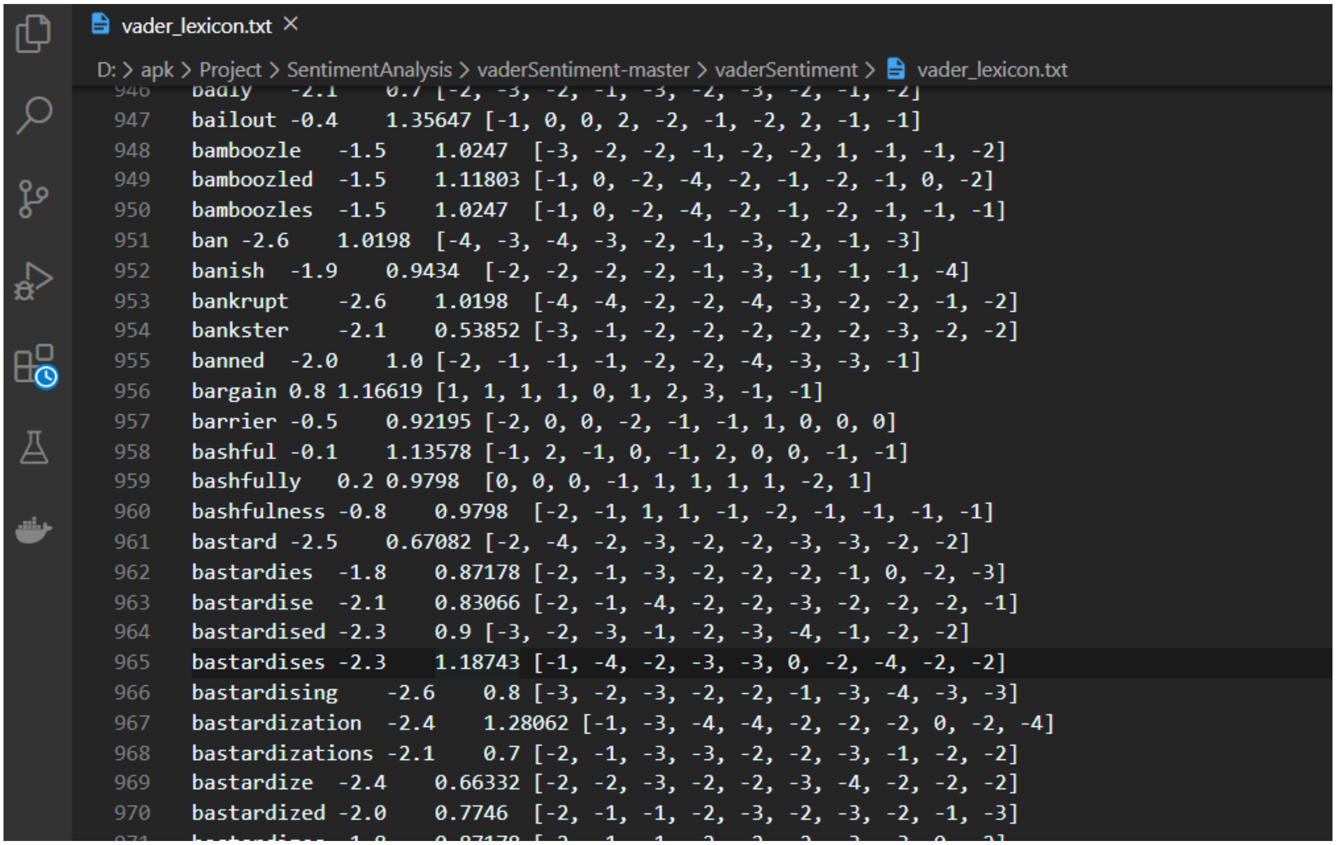
### 4.3 Classification

The next stage is that classification which is the process of finding a model that describes and distinguishes data classes and concepts. Classification is the problem of identifying to which set of categories does the respective data should belongs based on the trained data. Here, each word has been classified into three classes namely positive, negative and neutral based on the polarity scores. With the help of VADER lexicon, each word is compared and polarity scores were calculated. Implementation is shown below:

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
Sentence   = “The movie was good”
Obj = SentimentIntensityAnalyzer()
```

```
Sentiment_value = Obj.polarity_scores(Sentence)
If(Sentiment_value['compound'] >= 0.05):
        print("Positive")
else:
        print("Negative")
```

The lexicon content is shown in the Fig 4.2.


vader_lexicon.txt ×
D: > apk > Project > SentimentAnalysis > vaderSentiment-master > vaderSentiment > vader_lexicon.txt

```
badly	-2.1	0.7	[-2, -3, -2, -1, -3, -2, -3, -2, -1, -2]
bailout -0.4    1.35647 [-1, 0, 0, 2, -2, -1, -2, 2, -1, -1]
bamboozle   -1.5    1.0247  [-3, -2, -2, -1, -2, -2, 1, -1, -1, -2]
bamboozled  -1.5    1.11803 [-1, 0, -2, -4, -2, -1, -2, -1, 0, -2]
bamboozles  -1.5    1.0247  [-1, 0, -2, -4, -2, -1, -2, -1, -1, -1]
ban -2.6    1.0198  [-4, -3, -4, -3, -2, -1, -3, -2, -1, -3]
banish  -1.9    0.9434  [-2, -2, -2, -2, -1, -3, -1, -1, -1, -4]
bankrupt    -2.6    1.0198  [-4, -4, -2, -2, -4, -3, -2, -2, -1, -2]
bankster    -2.1    0.53852 [-3, -1, -2, -2, -2, -2, -2, -3, -2, -2]
banned  -2.0    1.0 [-2, -1, -1, -1, -2, -2, -4, -3, -3, -1]
bargain 0.8 1.16619 [1, 1, 1, 1, 0, 1, 2, 3, -1, -1]
barrier -0.5    0.92195 [-2, 0, 0, -2, -1, -1, 1, 0, 0, 0]
bashful -0.1    1.13578 [-1, 2, -1, 0, -1, 2, 0, 0, -1, -1]
bashfully   0.2 0.9798  [0, 0, 0, -1, 1, 1, 1, 1, -2, 1]
bashfulness -0.8    0.9798  [-2, -1, 1, 1, -1, -2, -1, -1, -1, -1]
bastard -2.5    0.67082 [-2, -4, -2, -3, -2, -2, -3, -3, -2, -2]
bastardies  -1.8    0.87178 [-2, -1, -3, -2, -2, -2, -1, 0, -2, -3]
bastardise  -2.1    0.83066 [-2, -1, -4, -2, -2, -3, -2, -2, -2, -1]
bastardised -2.3    0.9 [-3, -2, -3, -1, -2, -3, -4, -1, -2, -2]
bastardises -2.3    1.18743 [-1, -4, -2, -3, -3, 0, -2, -4, -2, -2]
bastardising    -2.6    0.8 [-3, -2, -3, -2, -2, -1, -3, -4, -3, -3]
bastardization  -2.4    1.28062 [-1, -3, -4, -4, -2, -2, -2, 0, -2, -4]
bastardizations -2.1    0.7 [-2, -1, -3, -3, -2, -2, -3, -1, -2, -2]
bastardize  -2.4    0.66332 [-2, -2, -3, -2, -2, -3, -4, -2, -2, -2]
bastardized -2.0    0.7746  [-2, -1, -1, -2, -3, -2, -3, -2, -1, -3]
```

Fig 4.2 Vader Lexicon

The above figure describes the polarity values with speech recognition word mapping array so that sentiment analysis can be done by using data source as speech recognition.

### 4.4 Data Visualization

Data visualization is used to communicate information clearly and efficiently to users by the usage of information graphics such as tables and cards. It helps users in analyzing a large amount of data in a simple way. It makes complex data more accessible, understandable, and usable. Here, after classifying the data, the total count in each class has been calculated and plotted in the form of pie-chart with the help of matplot library. Implementation is shown below:

```
import matplotlib.pyplot as plt
labels = ["Positive", "Negative", "Neutral"]
sizes= [positive, negative, neutral]
colors = ["yellowgreen", "lightcoral", "gold"]
explode = (0.1, 0, 0)
plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct="%1.1f%%", shadow=True)
plt.tight_layout()
plt.axis("equal")
plt.show()
```

From the inference, end user can have a better knowledge about their insights. Through this implementation stages, user’s role is to provide only the data or search keyword. This tool will provide detailed summary about their insights with info graphical notations.

# CHAPTER 5

# TESTING

Testing is the process of evaluating a system or its component's with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results. i.e. testing is executing a system in order to identify any errors or missing requirements in contrary to the actual desire or requirements.

## 5.1 Testing strategies

In order to make sure that system does not have any errors, the different levels of testing strategies that are applied at different phases of software development is shown in Fig 5.1.

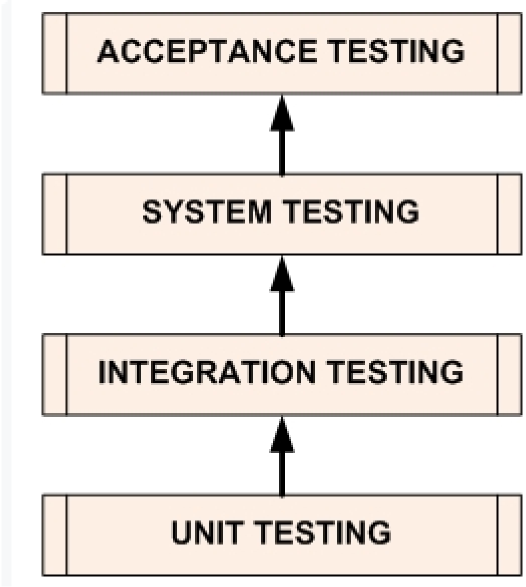


Fig 5.1 Phases of Testing in software development

### 5.1.1 Unit Testing

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality. Each module in the tool is tested separately with the help of various users social media accounts.

### 5.1.2 Integration Testing

The testing of combined parts of an application to determine if they function correctly together is Integration testing. The approach followed here is Bottom-up Integration testing where the smallest modules are tested first and then combined together.

### 5.1.3 System Testing

This is the next level in the testing and tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards. The complete system is tested with thousands of records from different users and measured accuracy.

### 5.1.4 Acceptance Testing

The main purpose of this Testing is to find whether application meets the intended specifications and satisfies the client's requirements. Model has been subjected to both alpha and beta testing and gathered their feedbacks.

## 5.2 Validation

All the levels in the testing (unit, integration, system) are implemented in our application successfully and the results obtained as expected.

### 5.3 Limitations

Though the lexicon based approach gives reasonable accuracy, machine learning approach will give much better accuracy in some use cases since it learns through the training dataset.

### 5.4 Test Results

The testing is done among the team members and by the end users. It satisfies the specified requirements and finally we obtained the results as expected.

# CHAPTER 6

# RESULT ANALYSIS

## 6.1 Graphical User Interface

### 6.1.1 Home screen

The first screen will ask the user to choose which social media account did he/she want to gather insights. The entire GUI was created with Python Tkinter. The home screen of our application is shown in Fig 6.1.

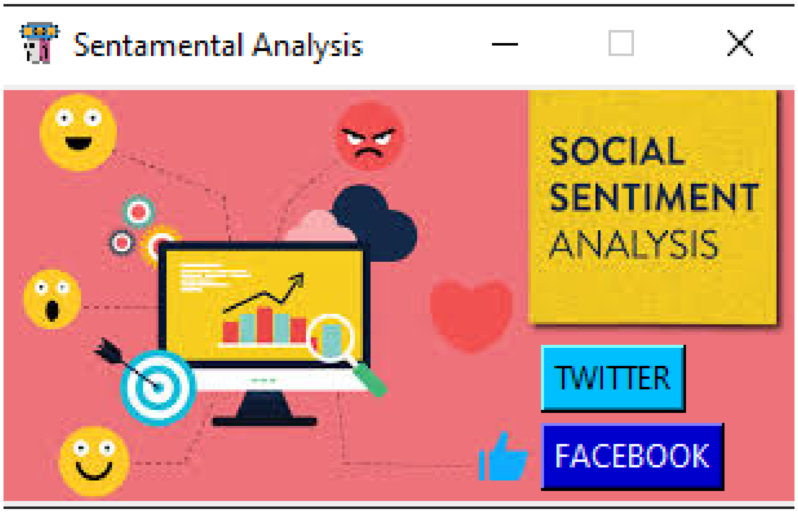
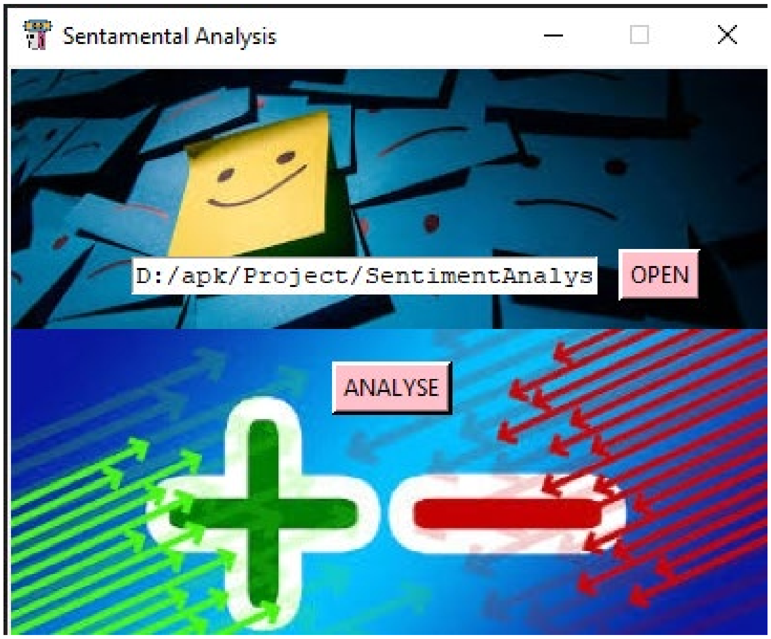


Fig 6.1 Homescreen

### 6.1.2 Facebook Analysis screen

The next screen from the home screen is based on the user’s choice in the homescreen. Facebook analysis screen will ask the user to load their file which was downloaded in before for analysis. Using the filedialog package, user can navigate their directories in their respective system. Facebook analysis screen is shown in Fig 6.2.



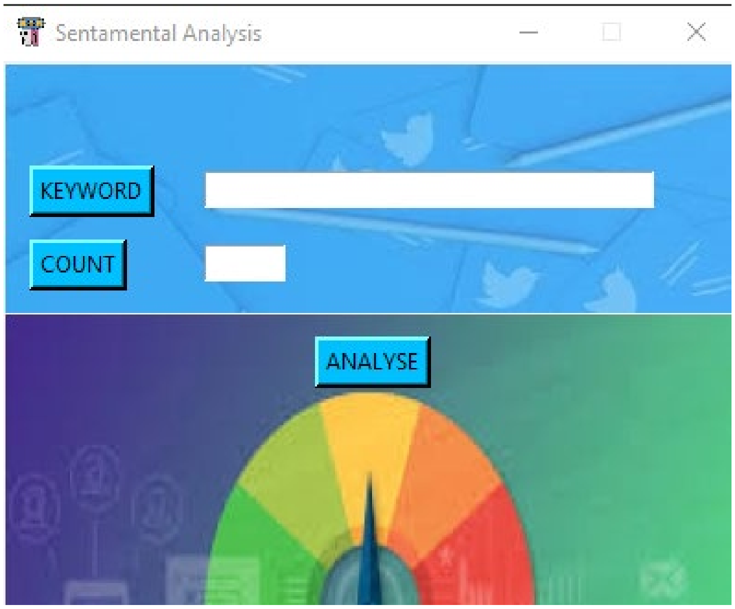
Sentamental Analysis

D:/apk/Project/SentimentAnalys

Fig 6.2 Facebook Analysis screen

### 6.1.3 Twitter Analysis screen

If the user's choice is twitter, he/she is asked to give the keyword and count so that it can scrap the data from twitter API . Twitter analysis screen is shown in Fig 6.3.



Sentamental Analysis

Fig 6.3 Twitter Analysis screen

### 6.2 Analysis screen

The final output of the analyzed data will be shown in a Pie chart. This was done with the help of matplotlib library. Here, the class along with their accuracy has been depicted clearly. The Output screen of our application is shown in Fig 6.4.

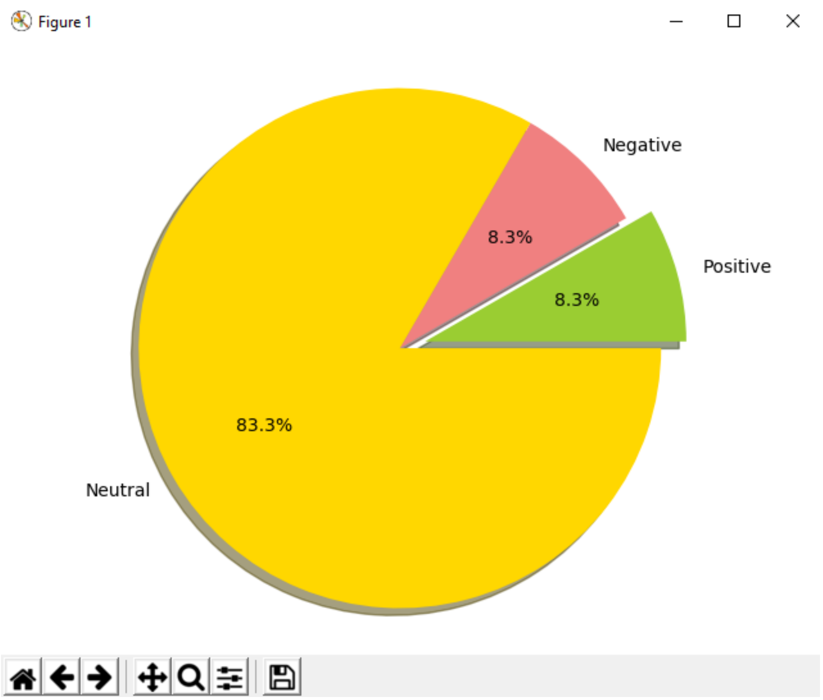


Fig 6.4 Output screen

# CHAPTER 7
# CONCLUSION

It is a very important fact to analyze how people think in different context about different things. This becomes more important when it comes to the business world because business is dependent on their customers and they always try to make products or services in order to fulfill customer requirements. So knowing what they want, what they think and talk about existing products, services and brands is more useful for businesses to make decisions such as identifying competitors and analyzing trends. From the view at the top level of the project, we get data from social media sites to gather insights from them and keep record of those sentiments with the information of the users who stated those sentiments for later use. Finally it does data mining with the extracted sentiments so that it can be used in product profiling, trend analysis and forecasting. After developing the crawler, the main challenge that was to be addressed was, how to decide whether a given sentence was positive or negative or neutral. The first thing that was found to address this challenge deals with lexical data source which is called VADER, in that it has positive and negative score for each word. Though there are positive and negative scores for almost all words in English language, when it comes to sentences, it differs the overall polarity of a sentence with other words and according to the context. Other than that, it cannot analyze words with short terms which in returns reduce the accuracy and sometimes it makes the result incorrect. In some cases, it did not give correct polarity values for sentences which includes terms like ‘not good’, ‘not bad’. During the implementation of the sentiment module we had to consider several issues such as, the comment by the user of a product or a brand can be not only in English but also mix with other language (Sinhala/Tamil), with emotional symbols etc., the comment may not completely match with what exactly user need to express about the product or brand, identifying the entity, identifying the relation of a particular comment with previous comments, ambiguity of words of the comment, human language is noisy and chaotic and the users may use different jargon or slang communications. But with the implementation of machine learning techniques, it could achieve more accurate results after building classifiers training on large labeled data sets but still there are some issues of processing natural language. Finally, using the sentiment scores for sentiments regarding particular product or service with the user's information, it could successfully profile the products, analyze trends and forecasting. So, as overall, the system is capable of saying that how a set of people of a particular age range, a particular area with a particular profession think about a particular product or service and how it will change it the future which are most useful information when it comes to business world.

## REFERENCES

1. Cristianini, J Shawe-Taylor (2000), “An Introduction to Support Vector Machines and Other Kernel-based Learning Methods” , Cambridge University Press.
2. Dan Jurafsky (2014),”Text Classification and Naïve Bayes” ,The Task of Text Classification, https://web stanford edu/class/cs124/lec/naivebayes ,web.
3. Hiroshi Shimodaira (2014),”Text Classification using Naïve Bayes” ,Document model, https://www.infedacuk/teaching/courses/inf2b/learnnotes/inf2b-learn-note07-2up,web.
4. H Kim, P Howland, and H Park (2005),”Dimention Reduction in Text Classification with Support Vector Machines”, Journal of Machine Learning Research.
5. Lindsay I Smith (2014),”Principle Component Analysis”, http://www cs otago acnz/cosc453/student_tutorials/principal_components, web.
6. Laura Auria, Rouslan A Moro (2014),”Support Vector Machines”, web.
7. Pang- Ning Tan , Michael Steinbach, Vipin Kumar ,”Introduction to data Mining”, pearson Publication.