

# PSG COLLEGE OF TECHNOLOGY

19OH01 - SOCIAL AND ECONOMIC NETWORK ANALYSIS



## TWITTER SENTIMENT ANALYSIS

- BATCH -12

NITHISA (18Z336)

POORNIMASRI (18Z341)

SRIMATHI (18Z355)

SUVALAKSHMI (18Z359)

VAISHNAVI (19z463)

## TABLE OF CONTENTS

1. Problem Statement	- 3
2. Dataset Description	- 3
3. Tools Used	- 3
4. Challenges Faced	- 3
5. Contribution Of Team Members	- 3
6. Annexure I: Code	- 4
7. Annexure II: Snapshots Of The Output	- 7
8. References	- 9



## 1. PROBLEM STATEMENT:

Twitter sentiment analysis is a technique used to analyze the sentiment of the tweets such as positive, negative, neutral, strongly positive, weekly positive, weekly negative, strongly positive. Twitter sentiment analysis is widely performed using python. Our task is to analyze the tweets of an individual twitter account in terms of polarity and to calculate the percentage of positive, negative and neutral tweets and general report of the tweets. The analysis report of the tweets are displayed in the form of the pie-chart.

## 2. DATASET DESCRIPTION:

Twitter sentimental analysis can be performed by collecting Twitter data consisting of using Twitter public API available for general public for free. It is the part of Data Collection.

## 3. TOOLS USED:

### 3.1. tweepy :

Tweepy is a Python package that makes us a convenient way to access the Twitter API with Python. Tweepy can be installed with the help of pip command.tweepy is an open source package

### 3.2. textblob :

The python library TextBlob is used for Natural Language Processing (NLP).TextBlob is used widely with Natural Language ToolKit (NLTK) to perform it's tasks. NLTK has various functions like word tokenizier.Text-Blob is a simple library which supports complex analysis and operations on textual data.

## 4.CHALLENGES FACED:

- Creating app in developer twitter account .
- Finding the polarity of some tweets
- Difficulty in visualizing the analysis report.

## 5.CONTRIBUTION OF TEAM MEMBERS:

ROLL NO	NAME	CONTRIBUTION
18Z336	Nithisa	Access key generated, Coding part(finding polarity)
18Z341	Poornimasri	Coding part ( visualization of results as pie-chart)
18Z355	Srimathi	Coding part (append tweets into csv file)
18Z359	Suvalakshmi	Created twitter API in development account, Coding part
19Z463	Vaishnavi	Documentation, Coding part

## 6.ANNEXURE I: CODE

```
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 = "
        consumerSecret = "
        accessToken = "
        accessTokenSecret = "
        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
        positive = 0
        wpositive = 0
        spositive = 0
        negative = 0
        wnegative = 0
        snegative = 0
        neutral = 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
                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
# Write to csv and close csv file
csvWriter.writerow(self.tweetText)
csvFile.close()
# 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)
# 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("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")
print()
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")
self.plotPieChart(positive, wpositive, spositive, negative, wnegative, snegative, neutral,
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, positive, wpositive, spositive, negative, wnegative, snegative, neutral
, searchTerm, noOfSearchTerms):
    labels = ['Positive [' + str(positive) + '%]', 'Weakly Positive [' + str(wpositive) + '%]', 'Str
ongly Positive [' + str(spositive) + '%]', 'Neutral [' + str(neutral) + '%]',
    'Negative [' + str(negative) + '%]', 'Weakly Negative [' + str(wnegative) + '%]', 'Str
ongly Negative [' + str(snegative) + '%]']
    sizes = [positive, wpositive, spositive, neutral, negative, wnegative, snegative]
    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(noOfSearchT
erms) + ' Tweets.')
    plt.axis('equal')
    plt.tight_layout()
    plt.show()
if __name__ == "__main__":
    sa = SentimentAnalysis()
    sa.DownloadData()

```

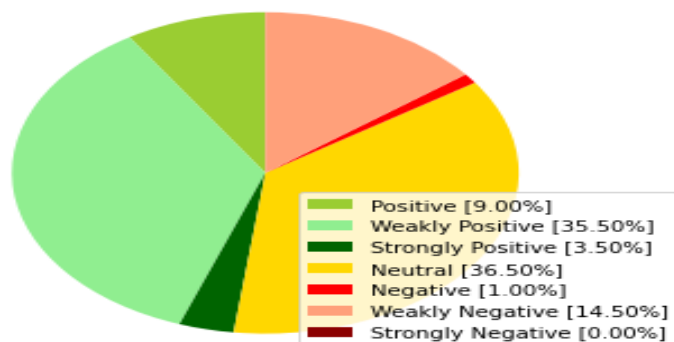
## 7.ANNEXURE II: SNAPSHOTS OF THE OUTPUT :

### 7.1.BITCOIN:

Enter Keyword/Tag to search about: bitcoin  
Enter how many tweets to search: 200  
How people are reacting on bitcoin by analyzing 200 tweets.

General Report:  
Weakly Positive

Detailed Report:  
9.00% people thought it was positive  
35.50% people thought it was weakly positive  
3.50% people thought it was strongly positive  
1.00% people thought it was negative  
14.50% people thought it was weakly negative  
0.00% people thought it was strongly negative  
36.50% people thought it was neutral  
How people are reacting on bitcoin by analyzing 200 Tweets.

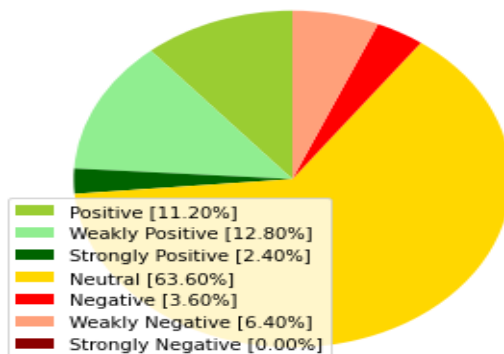


### 7.2. TAMULNADU ELECTION:

Enter Keyword/Tag to search about: tamilnadu election  
Enter how many tweets to search: 250  
How people are reacting on tamilnadu election by analyzing 250 tweets

General Report:  
Weakly Positive

Detailed Report:  
11.20% people thought it was positive  
12.80% people thought it was weakly positive  
2.40% people thought it was strongly positive  
3.60% people thought it was negative  
6.40% people thought it was weakly negative  
0.00% people thought it was strongly negative  
63.60% people thought it was neutral  
How people are reacting on tamilnadu election by analyzing 250 Tweets.



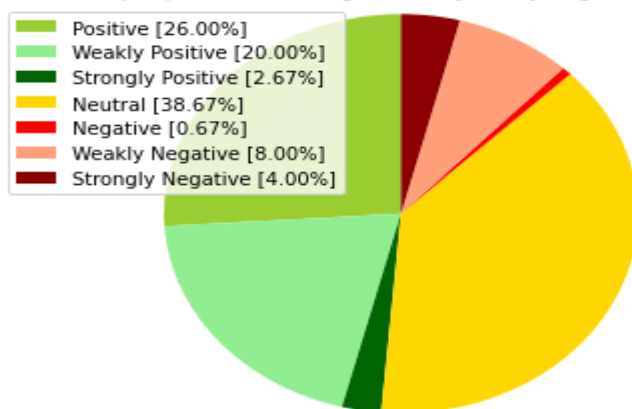
### 7.3. Internet of things(IoT):

Enter Keyword/Tag to search about: Iot  
Enter how many tweets to search: 150  
How people are reacting on Iot by analyzing 150 tweets.

General Report:  
Weakly Positive

Detailed Report:  
26.00% people thought it was positive  
20.00% people thought it was weakly positive  
2.67% people thought it was strongly positive  
0.67% people thought it was negative  
8.00% people thought it was weakly negative  
4.00% people thought it was strongly negative  
38.67% people thought it was neutral

How people are reacting on Iot by analyzing 150 Tweets.



### 8. REFERNCES:

Creating Twitter API :-

1. <https://stackoverflow.com/questions/1808855/getting-new-twitter-api-consumer-andsecret-keys>

2. <https://www.slickremix.com/docs/how-to-get-api-keys-and-tokens-for-twitter/>

Twitter Sentiment Analysis:

1. <https://monkeylearn.com/blog/sentiment-analysis-of-twitter/>

2. <https://towardsdatascience.com/twitter-sentiment-analysis-classification-using-nltkpython-fa912578614c>

3. <https://www.kaggle.com/ragnisah/text-data-cleaning-tweets-analysis>

4. <https://www.youtube.com/watch?v=ujId4ipkBio>