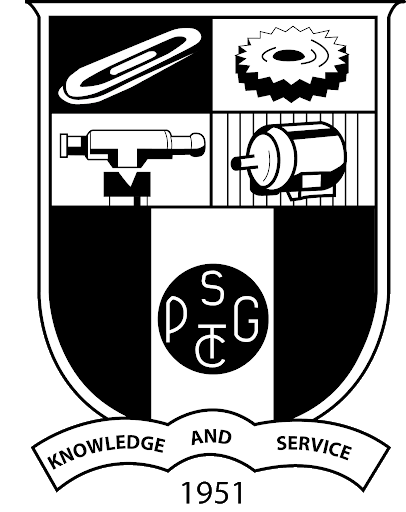
**PSG COLLEGE OF TECHNOLOGY**

**Department of Computer Science and Engineering**

**15ZH02-Social and Economic Network Analysis**



Analysing and visualizing the tweets made by Prime Minister of India

**Team Members:**

Elakkiya V(18Z217)

Nisha Arul Selvi R(18Z235)

Preneekaa R A(18Z238)

Sindhu G(18Z250)

Varsha E(18Z261)

**Problem Statement:**

Analysing and visualizing the tweets made by Prime Minister of India. One way of communicating the important announcements from the government to people is through tweets.

* Analysing the sentiments of people based on the most frequent words used by the Prime Minister in his tweets
* Analysing the tweets based on likes count
* Analysing the tweets based on retweets count
* Analysing the tweets based on replies count

Plotting the results in Gephi for better visualisation and analysis

**Data Set Description:**

The data set (narendramodi.csv) used for this analysis contains 23 columns. Out of which, we have used columns like date, username, tweet, likes\_count for our analysis.

* date – date in which tweet was posted in the format DD-MM-YY
* username – username of the account our Prime minister uses to post his tweets
* tweet – tweets posted by our Prime minister
* likes\_count – number of likes obtained for each of the tweets posted.

Dataset link: <https://www.kaggle.com/reach2ashish/all-narendra-modi-tweets>

**Tools Used:**

1. **numpy** – It is an open-source numerical Python library. NumPy contains a multi-dimensional array and matrix data structures. It can be utilised to perform a number of mathematical operations on arrays such as trigonometric, statistical, and algebraic routines. Therefore, the library contains a large number of mathematical, algebraic, and transformation functions.
2. **pandas** - Pandas can be used to perform data manipulation and analysis. It provides powerful and easy-to-use data structures, as well as the means to quickly perform operations on these structures.
3. **nltk** - NLTK (Natural Language Toolkit) is a suite that contains libraries and programs for statistical language processing. It is one of the most powerful NLP libraries, which contains packages to make machines understand human language and reply to it with an appropriate response.
4. **nltk.sentiment.vader** - using a pre-trained lexicon like the VADER which stands from Valence Aware Dictionary and sentiment Reasoner and is specifically attuned to calculate scores for sentiments
5. **tokenize** - Token is every “entity” that is a part of whatever was split up based on rules. For examples, each word is a token when a sentence is “tokenized” into words. So basically, tokenizing involves splitting sentences and words from the body of the text.
6. **matplotlib** - Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. pyplot() is the most important function in matplotlib library, which is used to plot 2D data.
7. **wordcloud** - Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance.
8. **subprocess** - The subprocess module present in Python is used to run new applications or programs through Python code by creating new processes. It also helps to obtain the input/output/error pipes as well as the exit codes of various commands.

**Challenges faced:**

1. The dataset contains tweets for 10 years where the number of tweets crosses 3000 (approx) each year. Visualisation of such huge amount of data in Gephi was difficult. So, we kept a constraint on the number of years and took the tweets of the most recent years.
2. We got problems in finding the exact packages for sentiment analysis.
3. Due to the use of nltk for sentiment analysis, we had the problem of classifying sarcastic words, abbreviations etc. So, we are not able to obtain 100 percent accuracy.

**Contribution of Team Members:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Roll No** | **Contribution** |
| Elakkiya V | 18Z217 | Likes count analysis, polarity |
| Nisha Arul Selvi R | 18Z235 | Data pre-processing |
| Preneekaa R A | 18Z238 | Replies count analysis, documentation |
| Sindhu G | 18Z250 | EDA(wordcloud), polarity |
| Varsha E | 18Z261 | Retweet analysis, documentation |

**Annexure I - Code:**

***# Importing the necessary packages***

import numpy as np

import pandas as pd

from nltk.sentiment.vader import SentimentIntensityAnalyzer

from nltk.sentiment.util import \*

from nltk import tokenize

***# calling the SentimentIntensityAnalyzer() from nltk package***

sid = SentimentIntensityAnalyzer()

***#reading the dataset***

tweets = pd.read\_csv('narendramodi1.csv')

***# removes pattern in the input text***

def remove\_pattern(input\_txt, pattern):

r = re.findall(pattern, input\_txt)

for word in r:

input\_txt = re.sub(word, "", input\_txt)

return input\_txt

***# remove twitter handles (@user)***

df['clean\_tweet'] = np.vectorize(remove\_pattern)(df['Tweets'], "@[\w]\*")

***# remove special characters, numbers and punctuations***

df['clean\_tweet'] = df['clean\_tweet'].str.replace("[^a-zA-Z#]", " ")

***# remove short words***

df['clean\_tweet'] = df['clean\_tweet'].apply(lambda x: " ".join([w for w in x.split() if len(w)>3]))

***# individual words considered as tokens***

tokenized\_tweet = df['clean\_tweet'].apply(lambda x: x.split())

tokenized\_tweet.head()

***# stem the words***

from nltk.stem.porter import PorterStemmer

stemmer = PorterStemmer()

tokenized\_tweet = tokenized\_tweet.apply(lambda sentence: [stemmer.stem(word) for word in sentence])

***# combine words into single sentence***

for i in range(len(tokenized\_tweet)):

tokenized\_tweet[i] = " ".join(tokenized\_tweet[i])

df['clean\_tweet'] = tokenized\_tweet

***#finding the polarity scores for the tweets***

sid = SentimentIntensityAnalyzer()

df['sentiment\_compound\_polarity']=df.clean\_tweet.apply(lambda x:sid.polarity\_scores(x)['compound'])

df['sentiment\_neutral']=df.clean\_tweet.apply(lambda x:sid.polarity\_scores(x)['neu'])

df['sentiment\_negative']=df.clean\_tweet.apply(lambda x:sid.polarity\_scores(x)['neg'])

df['sentiment\_pos']=df.clean\_tweet.apply(lambda x:sid.polarity\_scores(x)['pos'])

df['sentiment\_pos']=df.clean\_tweet.apply(lambda x:sid.polarity\_scores(x)['pos'])

df['sentiment\_type']=''

***#classifying the tweets based on their polarity score***

df.loc[df.sentiment\_compound\_polarity>0,'sentiment\_type']='POSITIVE'

df.loc[df.sentiment\_compound\_polarity==0,'sentiment\_type']='NEUTRAL'

df.loc[df.sentiment\_compound\_polarity<0,'sentiment\_type']='NEGATIVE'

***#plotting for visualisation***

df.sentiment\_type.value\_counts().plot(kind='bar',title="sentiment analysis")

***# importing necessary packages***

import matplotlib as mpl

import matplotlib.pyplot as plt

%matplotlib inline

from subprocess import check\_output

from wordcloud import WordCloud, STOPWORDS

***# code for generating wordcloud***

all\_words = " ".join([sentence for sentence in df['clean\_tweet']])

wordcloud = WordCloud(width=800, height=500, random\_state=42, max\_font\_size=100).generate(all\_words)

# plot the graph

plt.figure(figsize=(15,8))

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis('off')

plt.show()

**------------------------------------------------------------------------------------------------**

Two edge.csv files will be generated for visualising in Gephi.

|  |  |  |  |
| --- | --- | --- | --- |
|  | edge1.csv | edge2.csv | edge3.csv |
| Source | tweet number | tweet number | tweet number |
| Target | year | year | year |
| Edge weight | likes\_count | retweets\_count | replies\_count |

***#python code for creating edge.csv file from the original dataset***

import csv

import pandas as pd

col=['date','retweets\_count']

***#extracting columns from the dataset***

df = pd.read\_csv('E:\\SENA\\narendramodi.csv',usecols=col)

***#reading the dataset***

result = df.head(4091)

for i in range(0,len(result)):

x = result["date"][i].split("-")

year=x[-1]

print(year)

***#if the year is 2019, target node ‘a’ will be assigned else ‘b’ is assigned***

if(year=="2019"):

str1='a'

else:

str1='b'

str2='Undirected'

***#writing in newly created retweets.csv file***

file = open('E:\\SENA\\retweet.csv', 'a', newline='')

to\_append=f'{i+1}\ //writing in newly created retweets.csv file

{str1}\

{str2}\

{result["retweets\_count"][i]}'

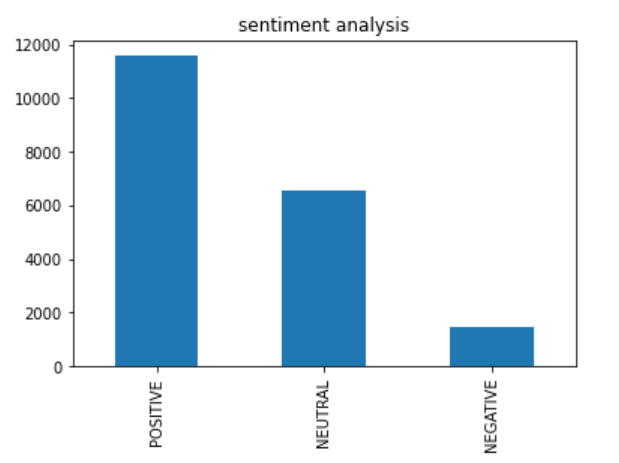
with file:

writer = csv.writer(file)

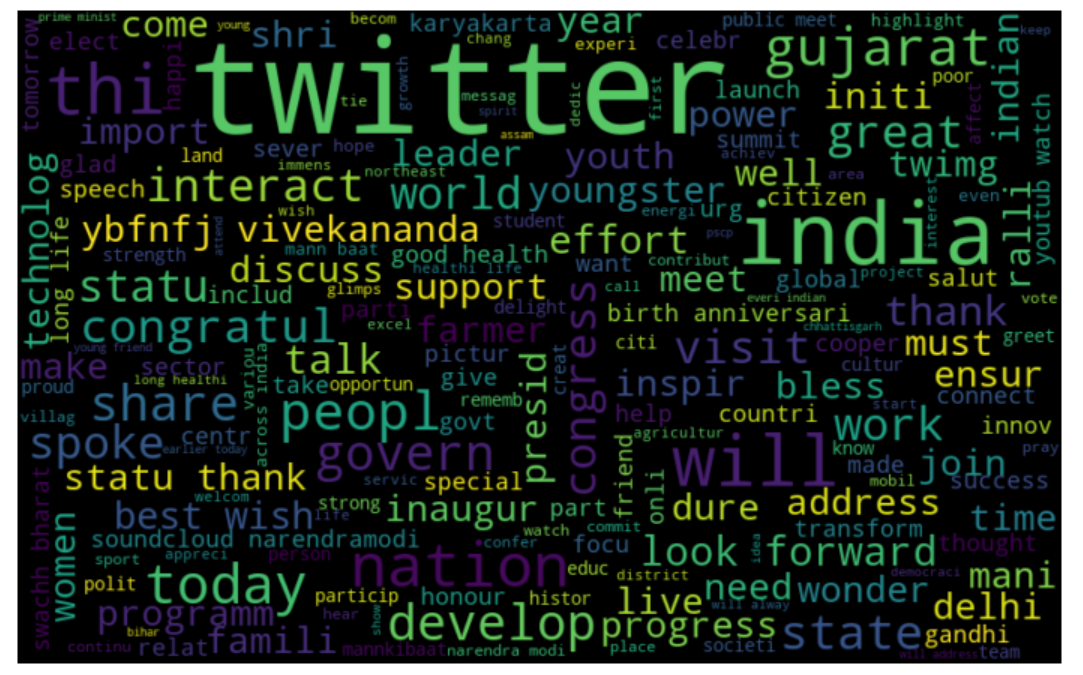
writer.writerow(to\_append.split())

**Annexure II – Snapshots of the output**

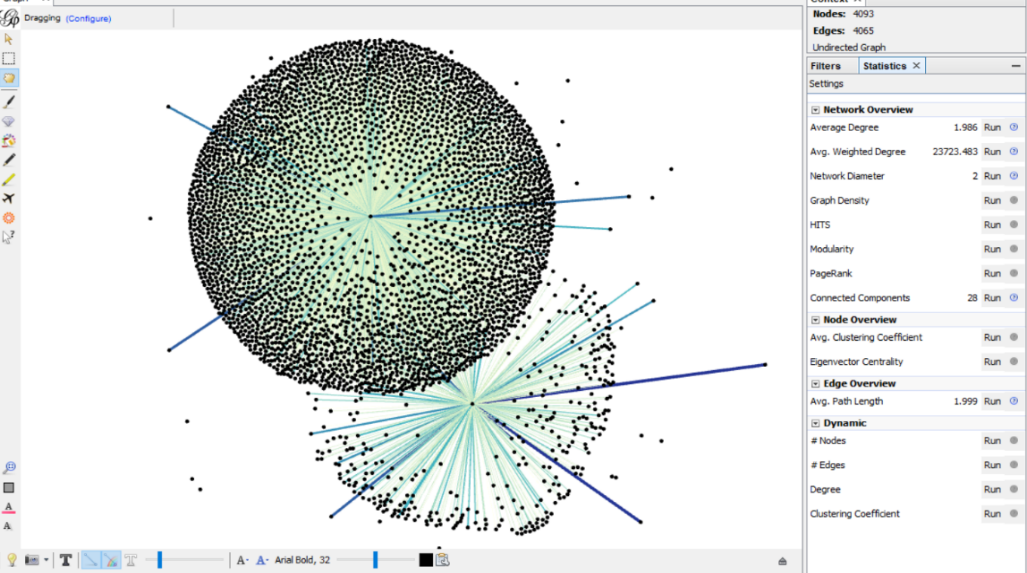
**Sentiment analysis**

****

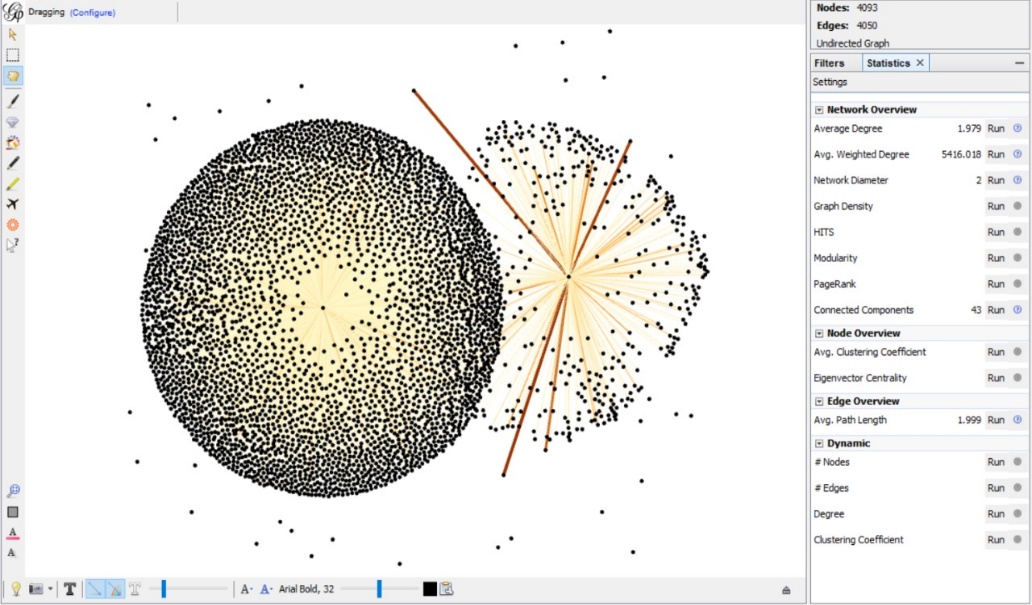
**Wordcloud for more frequent words used**

****

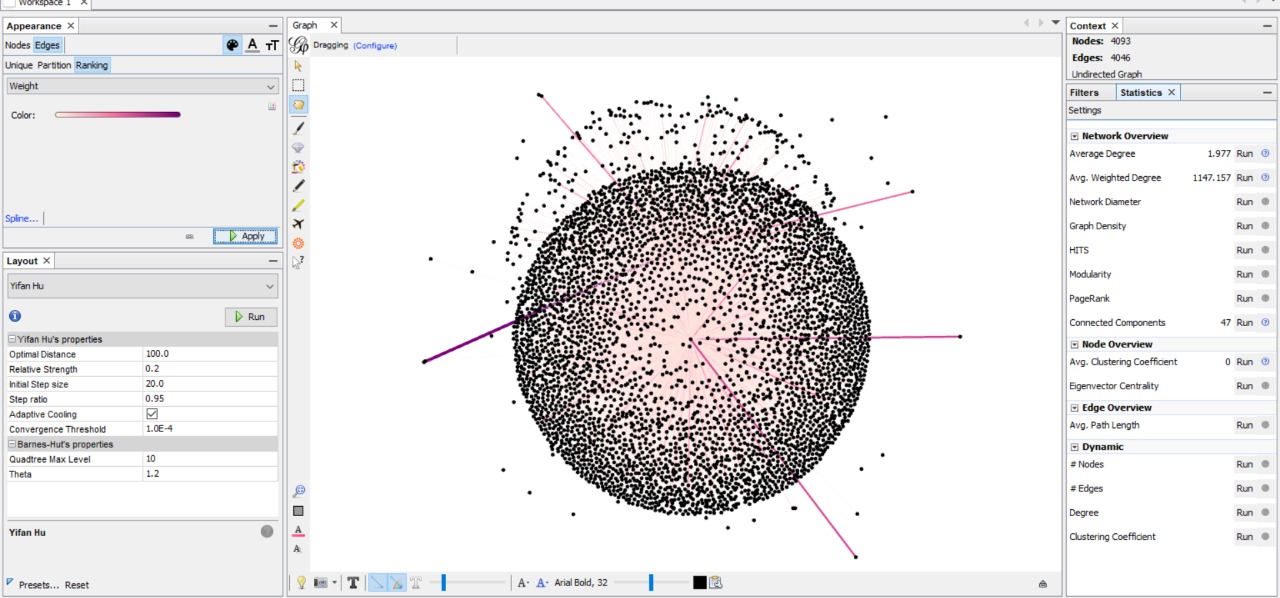
**Gephi Visualisation for Tweets based on likes count as edge weight**



**Gephi visualisation for tweets based on retweets count as edge weight**



**Gephi visualisation for tweets based on replies count as edge weight**



**References:**

1. <https://www.hindustantimes.com/india-news/prime-minister-narendra-modi-s-twitter-followers-cross-60-million-mark/story-fEyzOaoFpRQlKu9LA3Q3rJ.html>
2. <https://www.trackalytics.com/twitter/profile/snarendramodi/>
3. <http://https//www.socialtracker.io/twitter/narendramodi/>
4. <https://towardsdatascience.com/analyzing-the-twitter-profile-of-indias-newly-elected-pm-b61ae0edf6a5>
5. <https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/>
6. <https://realpython.com/python-nltk-sentiment-analysis/>
7. <https://towardsdatascience.com/step-by-step-twitter-sentiment-analysis-in-python-d6f650ade58d>
8. <https://towardsdatascience.com/a-beginners-guide-to-sentiment-analysis-in-python-95e354ea84f6>
9. <https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk>
10. <https://stackabuse.com/python-for-nlp-sentiment-analysis-with-scikit-learn/>