19ZO02-Social and Economic Network Analysis

Project Report

Team Members:

19Z202	Anita Priyadarshini
19Z217	Hema Varshini
19Z218	Johanna Smriti
19Z225	Keerthna M

BACHELOR OF ENGINEERING

Branch: COMPUTER SCIENCE AND ENGINEERING

Of Anna University

PSG College of Technology

641004

1. Problem Statement

a. Objective is to analyze and detect communities present in a twitter dataset.we will collect Twitter users and relationships. We will load this data to a graph database. Finally, we will analyze the network

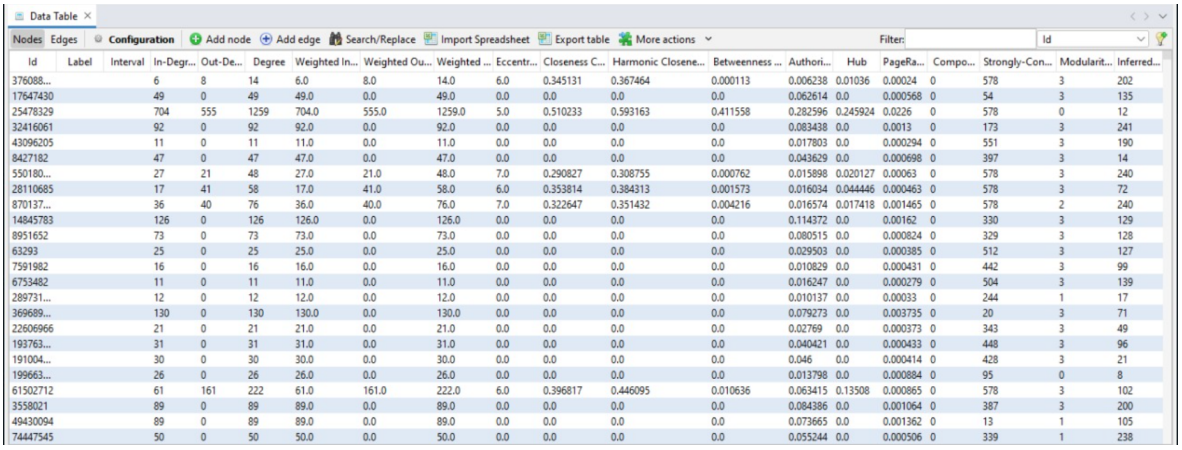
b. In this project, we will use a Python package, Tweepy, to download Twitter data from the Twitter API and another Python package, NetworkX, to build a network out of that data and run some analysis. Finally, we will use Gephi to visualize the network.

c. The resulting network has —-----users and —--- interactions with many groups of small interactions among users, and low clustering coefficient and modularity values.

d. Social network analysis (SNA) is the process of investigating social structures through the use of networks and graph theory. It characterizes networked structures in terms of nodes (individual actors, people, or things within the network) and the ties, edges, or links (relationships or interactions) that connect them.

2. Dataset Description

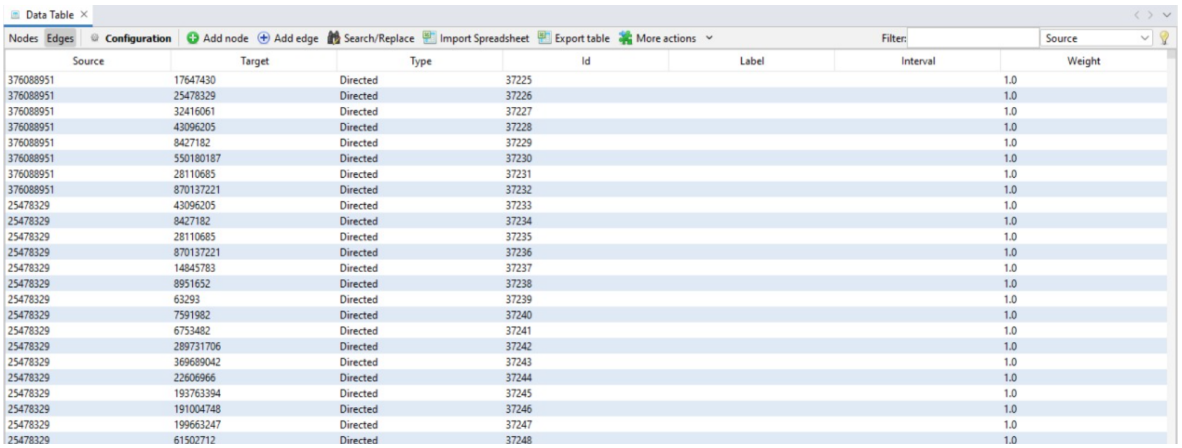
a. The steps taken to extract our dataset is as follows
    i. Use Tweepy to scrape Twitter for all of a certain users followers and (most of) their followers
    ii. Create pandas Dataframe from all these connections
    iii. On google collab we initially perform some simple network analysis using NetworkX
    iv. Visualize the network using Gephi
b. We finally extract two datasets to carry out our analysis-
    i. Edges
    ii. Nodes
c. These datasets are labeled and have the features necessary to make Observations and visualizations.



Data Table

Nodes | Edges | Configuration | Add node | Add edge | Search/Replace | Import Spreadsheet | Export table | More actions | Filter: | Id

Id	Label	Interval	In-Degr...	Out-De...	Degree	Weighted In...	Weighted Ou...	Weighted ...	Eccentr...	Closeness C...	Harmonic Closene...	Betweenness ...	Authori...	Hub	PageRa...	Compo...	Strongly-Con...	Modularit...	Inferred...
376088...			6	8	14	6.0	8.0	14.0	6.0	0.345131	0.367464	0.000113	0.006238	0.01036	0.00024	0	578	3	202
17647430			49	0	49	49.0	0.0	49.0	0.0	0.0	0.0	0.0	0.062614	0.0	0.000568	0	54	3	135
25478329			704	555	1259	704.0	555.0	1259.0	5.0	0.510233	0.593163	0.411558	0.282596	0.245924	0.0226	0	578	0	12
32416061			92	0	92	92.0	0.0	92.0	0.0	0.0	0.0	0.0	0.083438	0.0	0.0013	0	173	3	241
43096205			11	0	11	11.0	0.0	11.0	0.0	0.0	0.0	0.0	0.017803	0.0	0.000294	0	551	3	190
8427182			47	0	47	47.0	0.0	47.0	0.0	0.0	0.0	0.0	0.043629	0.0	0.000698	0	397	3	14
550180...			27	21	48	27.0	21.0	48.0	7.0	0.290827	0.308755	0.000762	0.015898	0.020127	0.00063	0	578	3	240
28110685			17	41	58	17.0	41.0	58.0	6.0	0.353814	0.384313	0.001573	0.016034	0.044446	0.000463	0	578	3	72
870137...			36	40	76	36.0	40.0	76.0	7.0	0.322647	0.351432	0.004216	0.016574	0.017418	0.001465	0	578	2	240
14845783			126	0	126	126.0	0.0	126.0	0.0	0.0	0.0	0.0	0.114372	0.0	0.00162	0	330	3	129
8951652			73	0	73	73.0	0.0	73.0	0.0	0.0	0.0	0.0	0.080515	0.0	0.000824	0	329	3	128
63293			25	0	25	25.0	0.0	25.0	0.0	0.0	0.0	0.0	0.029503	0.0	0.000385	0	512	3	127
7591982			16	0	16	16.0	0.0	16.0	0.0	0.0	0.0	0.0	0.010829	0.0	0.000431	0	442	3	99
6753482			11	0	11	11.0	0.0	11.0	0.0	0.0	0.0	0.0	0.016247	0.0	0.000279	0	504	3	139
289731...			12	0	12	12.0	0.0	12.0	0.0	0.0	0.0	0.0	0.010137	0.0	0.00033	0	244	1	17
369689...			130	0	130	130.0	0.0	130.0	0.0	0.0	0.0	0.0	0.079273	0.0	0.003735	0	20	3	71
22606966			21	0	21	21.0	0.0	21.0	0.0	0.0	0.0	0.0	0.02769	0.0	0.000373	0	343	3	49
193763...			31	0	31	31.0	0.0	31.0	0.0	0.0	0.0	0.0	0.040421	0.0	0.000433	0	448	3	96
191004...			30	0	30	30.0	0.0	30.0	0.0	0.0	0.0	0.0	0.046	0.0	0.000414	0	428	3	21
199663...			26	0	26	26.0	0.0	26.0	0.0	0.0	0.0	0.0	0.013798	0.0	0.000884	0	95	0	8
61502712			61	161	222	61.0	161.0	222.0	6.0	0.396817	0.446095	0.010636	0.063415	0.13508	0.000865	0	578	3	102
3558021			89	0	89	89.0	0.0	89.0	0.0	0.0	0.0	0.0	0.084386	0.0	0.001064	0	387	3	200
49430094			89	0	89	89.0	0.0	89.0	0.0	0.0	0.0	0.0	0.073665	0.0	0.001362	0	13	1	105
74447545			50	0	50	50.0	0.0	50.0	0.0	0.0	0.0	0.0	0.055244	0.0	0.000506	0	339	1	238



Data Table

Nodes | Edges | Configuration | Add node | Add edge | Search/Replace | Import Spreadsheet | Export table | More actions | Filter: | Source

Source	Target	Type	Id	Label	Interval	Weight
376088951	17647430	Directed	37225			1.0
376088951	25478329	Directed	37226			1.0
376088951	32416061	Directed	37227			1.0
376088951	43096205	Directed	37228			1.0
376088951	8427182	Directed	37229			1.0
376088951	550180187	Directed	37230			1.0
376088951	28110685	Directed	37231			1.0
376088951	870137221	Directed	37232			1.0
25478329	43096205	Directed	37233			1.0
25478329	8427182	Directed	37234			1.0
25478329	28110685	Directed	37235			1.0
25478329	870137221	Directed	37236			1.0
25478329	14845783	Directed	37237			1.0
25478329	8951652	Directed	37238			1.0
25478329	63293	Directed	37239			1.0
25478329	7591982	Directed	37240			1.0
25478329	6753482	Directed	37241			1.0
25478329	289731706	Directed	37242			1.0
25478329	369689042	Directed	37243			1.0
25478329	22606966	Directed	37244			1.0
25478329	193763394	Directed	37245			1.0
25478329	191004748	Directed	37246			1.0
25478329	199663247	Directed	37247			1.0
25478329	61502712	Directed	37248			1.0

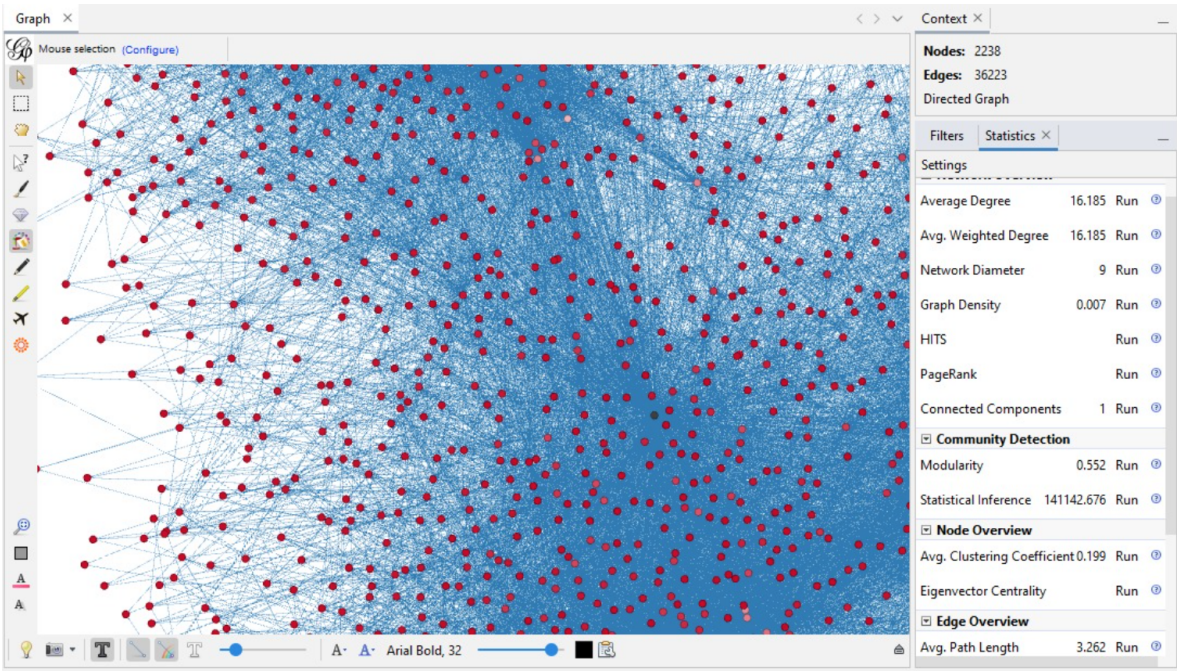
3. Tools Used

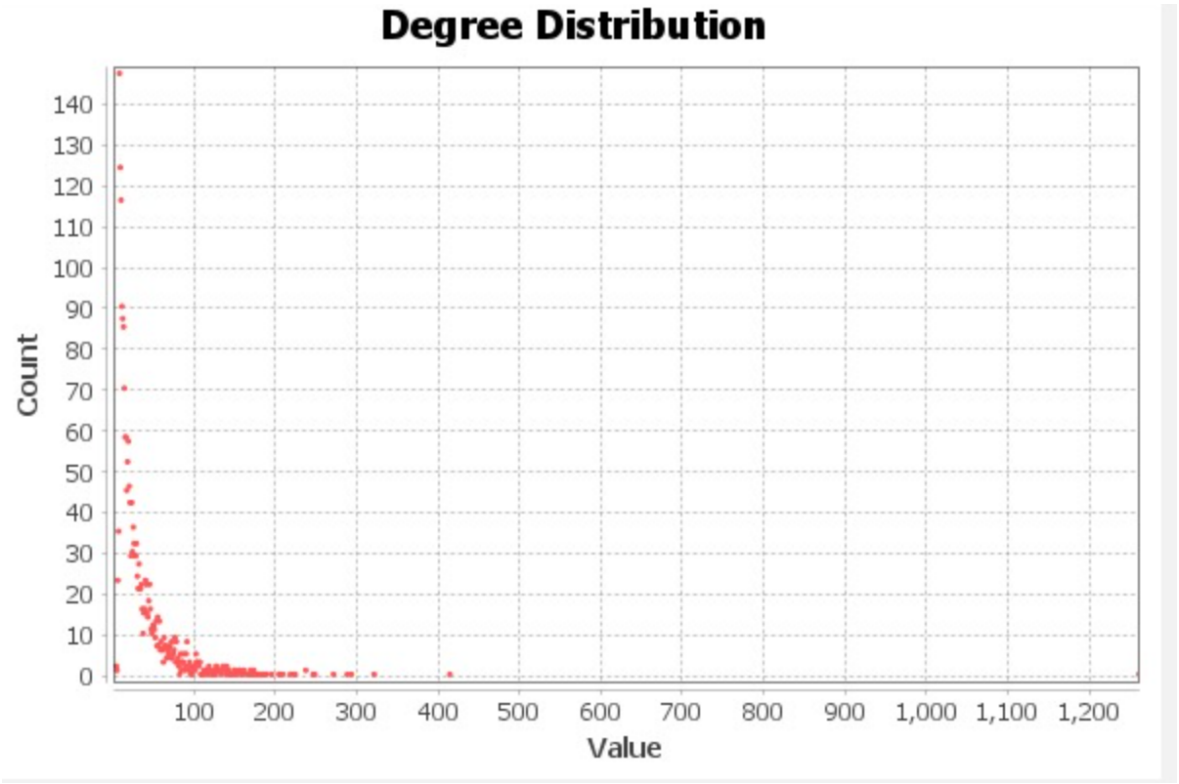
a. Google Collab
   i. Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.
b. Tweepy
   i. Tweepy is an open source Python package that gives you a very convenient way to access the Twitter API with Python.
c. NetworkX
   i. We use this package to convert the DataFrame into a graph or network.
d. Gephi
   i. Gephi is an open-source network analysis and visualization software package written in Java on the NetBeans platform.

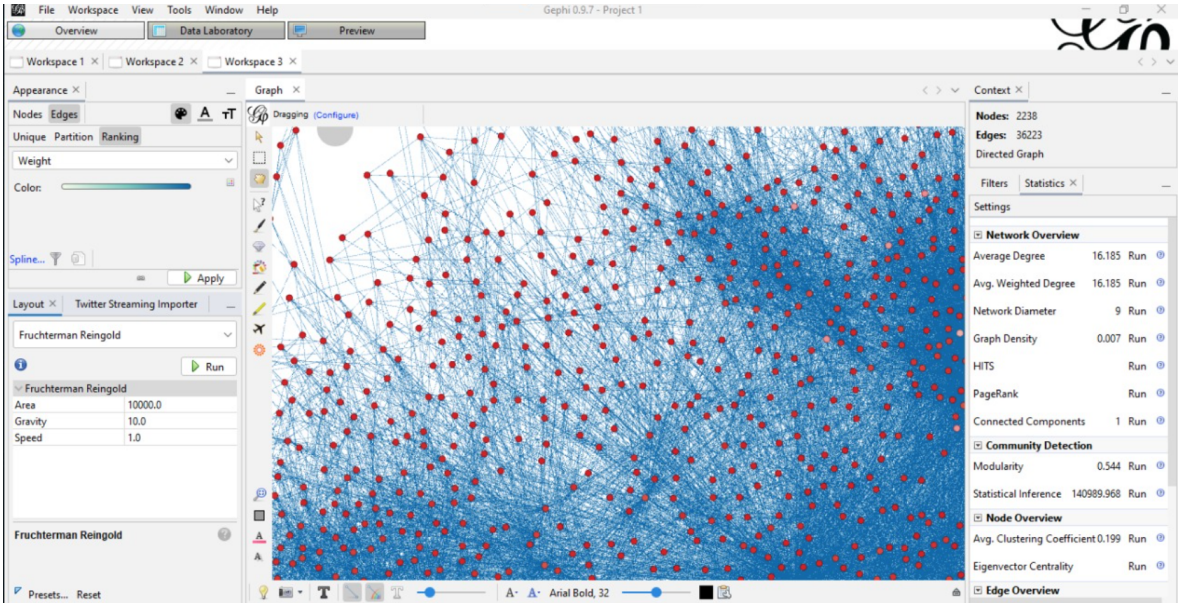
4. Challenges faced

a. Acquiring the dataset initially had certain complications, with respect to features available and quantity
b. We faced some problem in trying the implementation of the analysis code
c. We were new to both gephi and Network X, hence there was some learning curve

5. Gephi Implementation:-







6. Collab Implementation:-

a. Implementation of tweepy

```
[ ] my_consumer_key = "tW31leTZJTGyRh5eU82LJpca0"
    my_consumer_secret = "YwnWXQz2VxFL8Q8XsbKpdHhoLDoQfEhbM2v7k4CVWI1tqpo82m"

    my_access_token = "1590748969193406470-hbA0H5bnsvwXc9WinETV4SRHYBXKzg"
    my_access_secret = "xvdl3VdfrlVwRMAdD2wVI05k9WCQKlhi2R9YEgBvvW2ay"

    my_bearer_token = "AAAAAAAAAAAAAAAAAAAAAAenjAEAAAAAOW%2FFQFZnjJPzMLjLdRZ14qUBXho%3DizD0unBTCt7sWJfSJavmwAoCt29yobzDPalWVujadQ173M3HPp"
```

```
[ ] api = tweepy.Client(
        wait_on_rate_limit = False,
        consumer_key = my_consumer_key,
        consumer_secret = my_consumer_secret,
        access_token = my_access_token,
        access_token_secret = my_access_secret,
        bearer_token = my_bearer_token,
    )
```

```
[ ] api

    <tweepy.client.Client at 0x7f5a6cca5e90>
```

b. Scrapping followers from Kylie Jenner

```
[ ] me = api.get_user(username = 'kyliejenner')
    me.data.id

    236699098
```

```
[ ] williams_followers = api.get_users_followers(
        id = me.data.id ,
        user_fields = ["created_at", "description","public_metrics","verified",],
        max_results = 1000)
```

```
[ ] print("Number of followers collected from Twitter API: ",len(williams_followers.data))

    Number of followers collected from Twitter API:  1000
```

```
[ ] williams_followers_df = pd.DataFrame()

    for i in williams_followers.data:
      temp_data = pd.json_normalize( i.data ,  sep = "_")
      williams_followers_df = williams_followers_df.append(temp_data,ignore_index=True)
```

```
[ ] williams_followers_df["id"]

    0      1010654403102171136
    1      1591334448804597761
    2      1488160778393948163
    3      1591333602628976642
    4      1591333668173148160
```

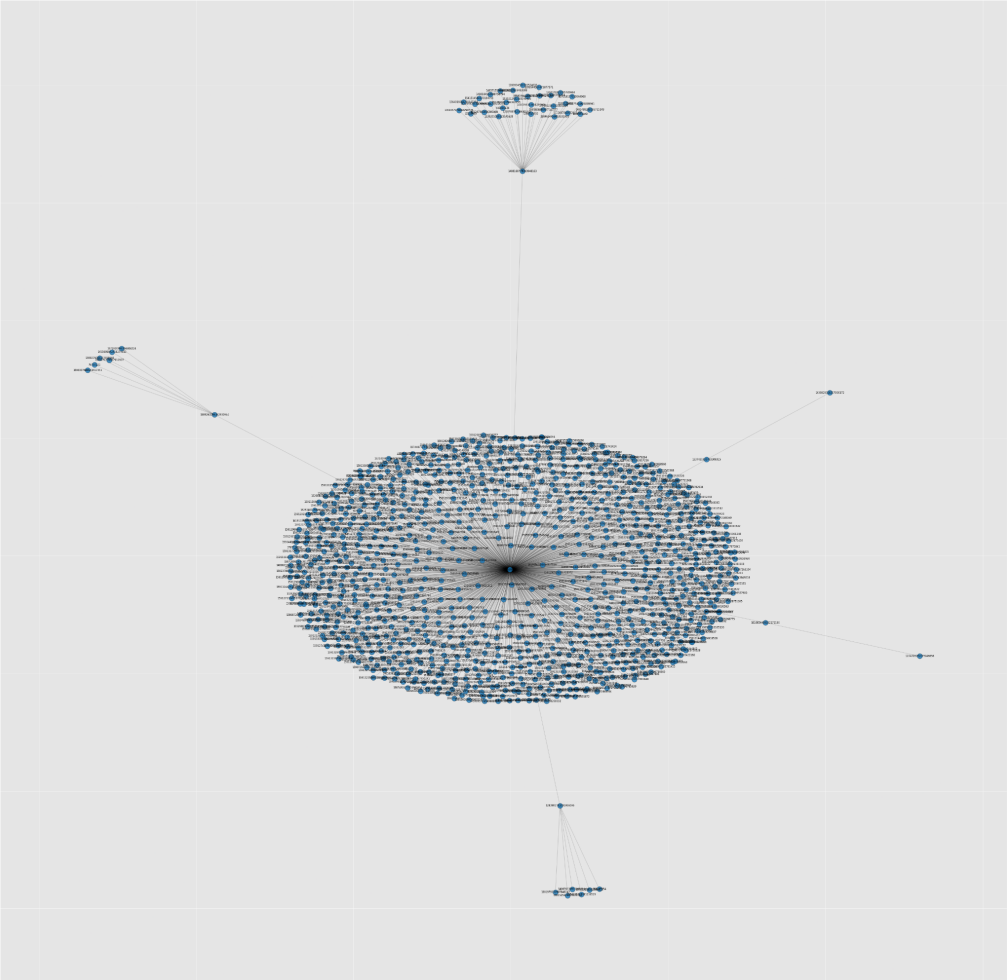
c. Dataset

df

	source	target
0	236699098	1010654403102171136
1	236699098	1591334448804597761
2	236699098	1488160778393948163
3	236699098	1591333602628976642
4	236699098	1591333668173148160
...	...	...
995	236699098	1591272174496669697
996	236699098	1591271063337504771
997	236699098	1591272286862057472
998	236699098	1590968791881887744
999	236699098	1590774739370270720

1000 rows × 2 columns

d. Visualisations:-

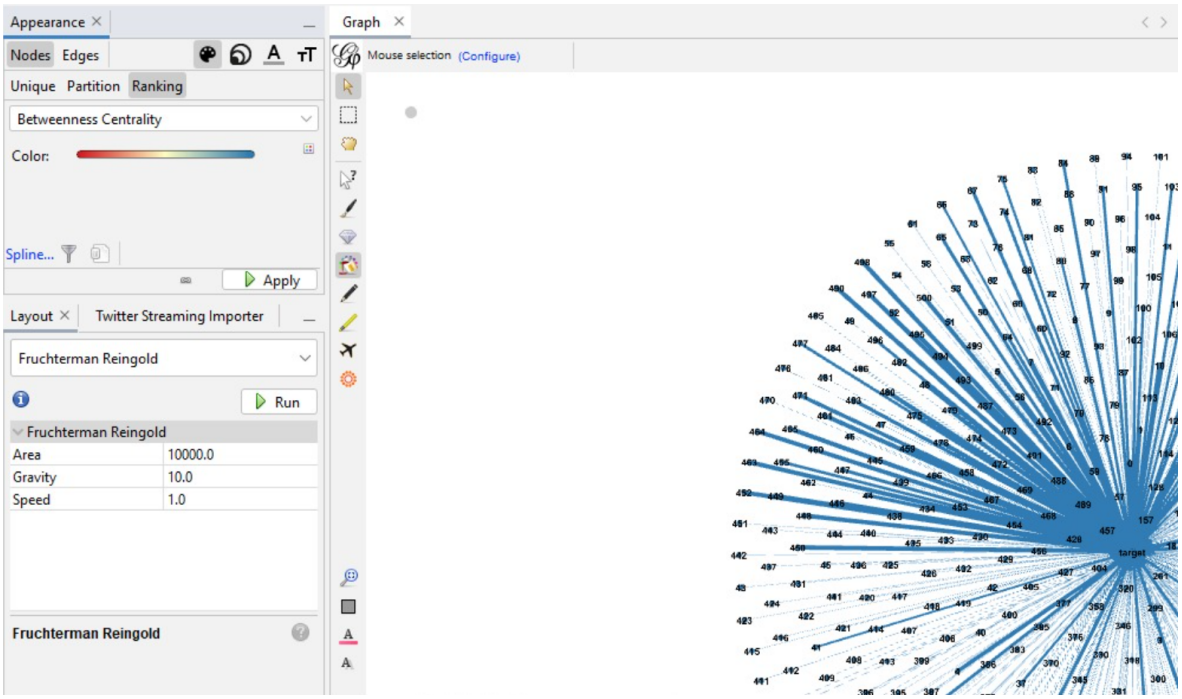


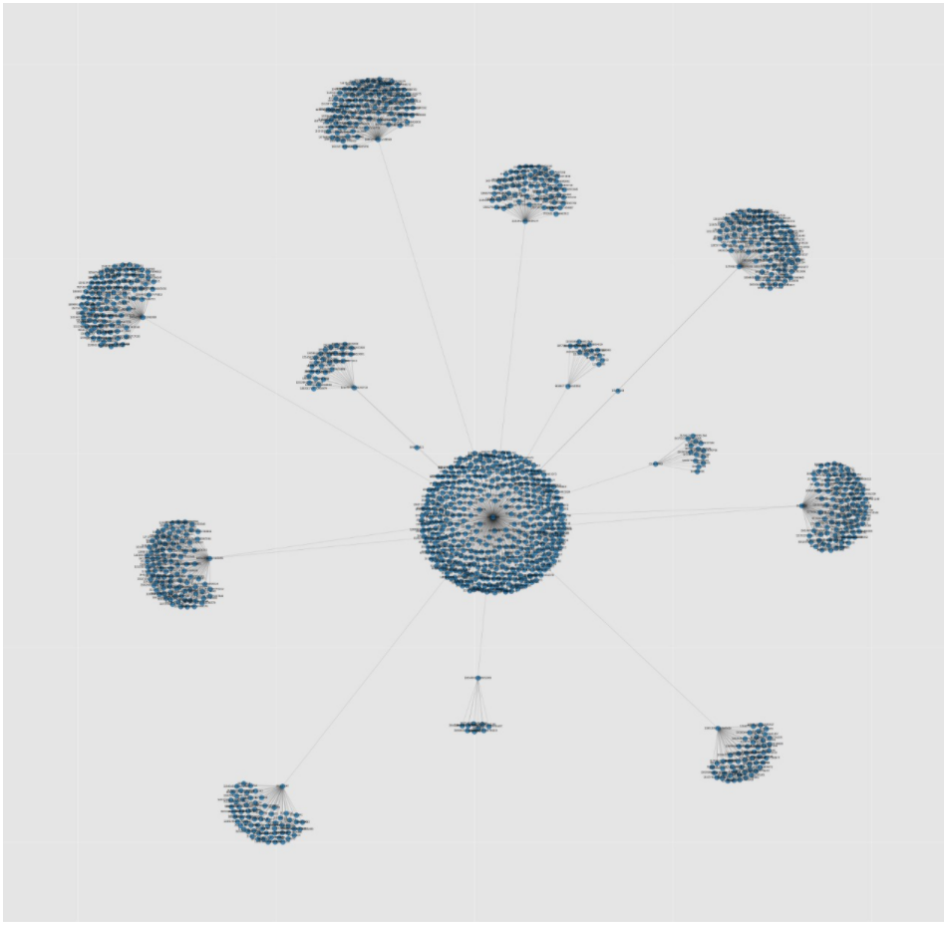
7. Contributions

Name (Roll number)	Contribution
Anita Priyadarshini (19Z202)	Documentation and Dataset scrapping
Johanna Smriti (19Z218)	Graph Visualization on Gephi
Hema Varshini (19Z217)	Documentation and Dataset scrapping
Keerthna M (19Z225)	Coding and analysis on Google Collab

8. Annexure

Communities detection:-





9. References

a. https://towardsdatascience.com/how-to-download-and-visualize-your-twitter-network-f009dbbf107b

b. https://towardsdatascience.com/how-to-download-and-visualize-your-twitter-network-f009dbbf107b
c. https://towardsdatascience.com/analyzing-twitter-user-network-1cfcef1dd89d
d. https://www.uu.nl/sites/default/files/analyzing_and_visualizing_your_twitter_networks_in_gephi.pdf
e. https://medium.com/@Luca/guide-analyzing-twitter-networks-with-gephi-0-9-1-2e0220d9097d
f. https://www.toptal.com/r/social-network-analysis-in-r-gephi-tutorial