

CS 579 Online Social Network Analysis

Project Report

On

Social Media Data Analysis on Twitter Data

By

Name of Student

1. Henilkumar Hareshbhai Patel
2. Harsh Gordhan Dungani

CWID

- A20513297
A20514062

**Under The Supervision
of
Prof. Kai Shu**



College Of Computing

**Illinois Institute of Technology
Chicago, Illinois.**

February 2022

CONTENT:

Section I: Project Details

- 1.1 Project Topic
- 1.2 Application Subject Area
- 1.3 Data Set Source

Section II: Proposed Approach

- 2.1 Data Collection
- 2.2 Data Processing
- 2.3 Data Visualization
- 2.4 Network Measures

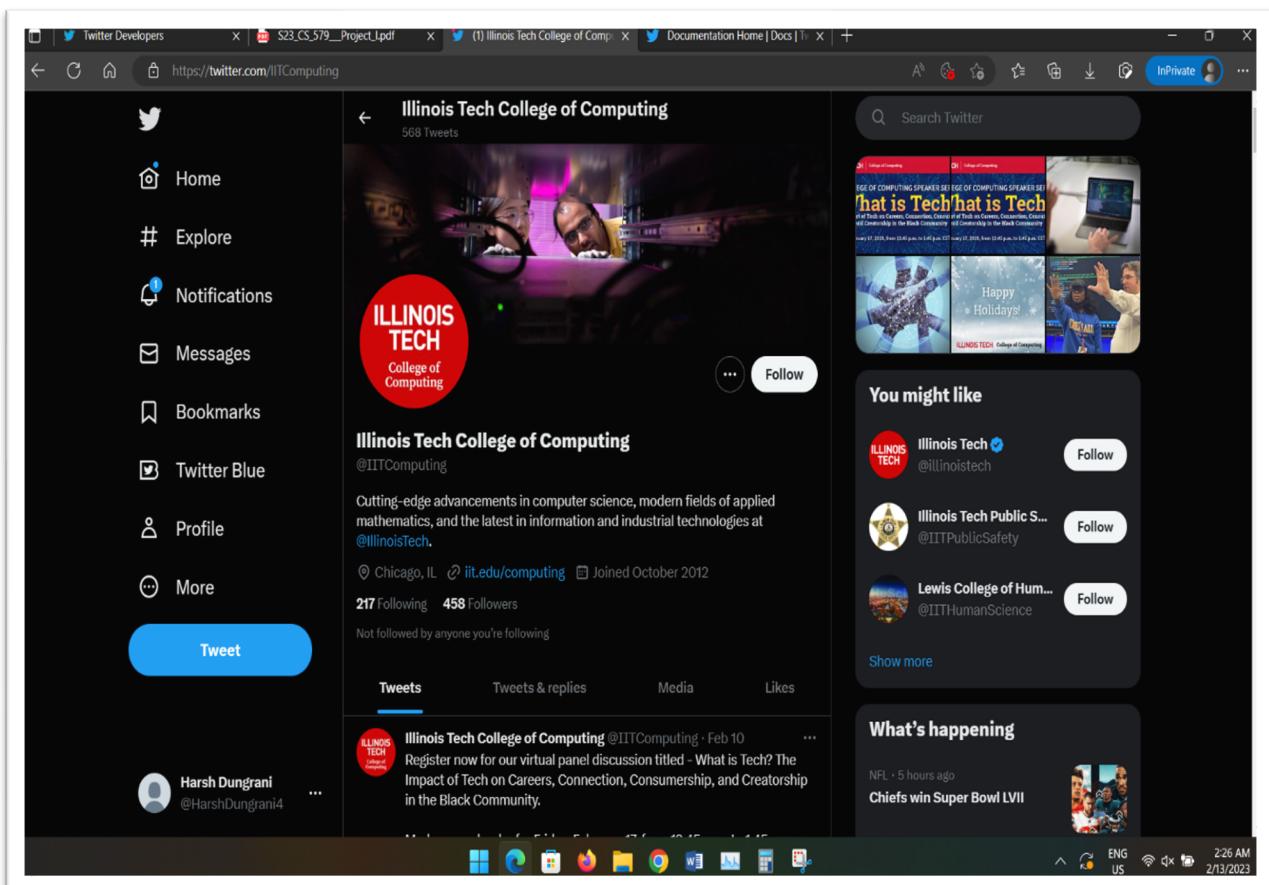
Section III: Team Efforts

Section IV: References

Section I: Project Details

1.1 Project Topic: Crawl data from Twitter website using the necessary credentials and create a social network. Also, we need to calculate the different measurements for the charts, such as degree distribution, clustering coefficient, PageRank, Diameter, Closeness, and Betweenness.

1.2 Application Subject Area: We have created a user's friendship network which is represented as a graph that has the nodes as users and edges as the relation between them. For the friendship network we have used Twitter as the social media platform and the user is the official twitter handler of Illinois Tech's College of Computing "@IITComputing". We have found the followers and following list and represented them in a friendship network.



1.3 Data Set Source: We have taken the data from the twitter using the twitter API. The following are the screenshots of the created developer account and the assigned credentials.[1]

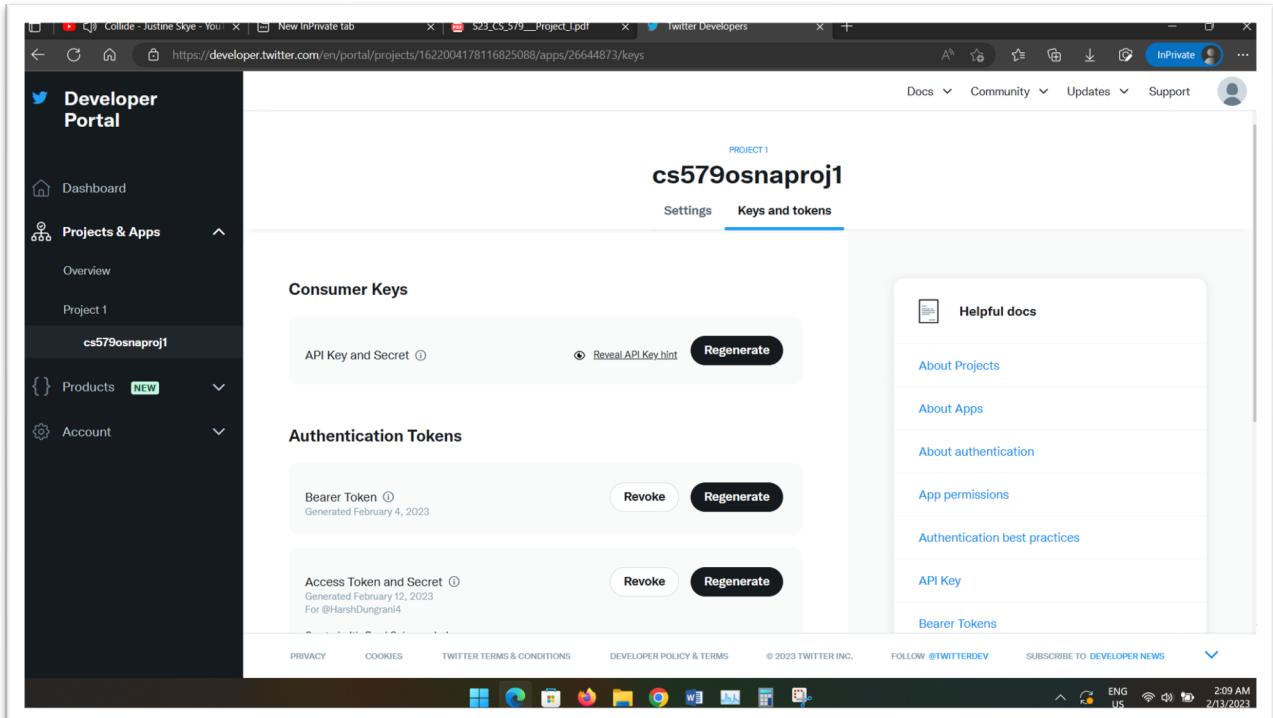
Created Twitter Developer Account with Elevated Access:

The screenshot shows the Twitter Developer Portal interface. On the left, there's a sidebar with 'Developer Portal' and 'Projects & Apps'. Under 'Project 1', it lists 'cs579osnaproj1' and 'Twitter API v2'. The main area is titled 'Project 1' and shows 'Overview' and 'Settings' tabs. Under 'Access', it says 'Elevated' with 'View detailed features' link. It lists three environments: 'Apps' (3 environments), 'Tweets' (Retrieve up to 2M Tweets per month), and 'Cost' (free). Below this is a 'Usage' section with 'MONTHLY TWEET CAP USAGE' and a bar chart. To the right, there's a 'Helpful docs' sidebar with links like 'About Projects', 'About Apps', 'About authentication', 'About Tweet caps', and 'Authentication best practices'. At the bottom, there are links for 'PRIVACY', 'COOKIES', 'TWITTER TERMS & CONDITIONS', 'DEVELOPER POLICY & TERMS', and social media icons. A footer bar includes 'FOLLOW @TWITTERDEV', 'SUBSCRIBE TO DEVELOPER NEWS', and system status icons.

Created a project with name “cs579osnaproj1”:

The screenshot shows the 'Settings' tab for the 'cs579osnaproj1' project. It displays 'App details' with 'NAME: cs579osnaproj1' and 'APP ID: 26644873'. There's an 'Edit' button next to the name. Below this is a 'DESCRIPTION' section with the note: 'This information will be visible to people who've authorized your App.' and 'This app was created to use the Twitter API.'. A 'ENVIRONMENT' section follows. To the right, there's a 'Authentication docs' sidebar with 'Authentication methods' and 'v2 endpoints available with OAuth 2.0'. A 'Quick info on App environments' button is also present. The bottom of the screen has standard footer links and system status icons.

Generated Keys and Tokens:



Section II: Proposed Approach:

2.1 Data Collection: The above given credentials for the created Twitter developer account were used to collect the required data from Twitter with the use of further libraries.

- 1) **Tweepy:** Tweepy is a python library that is useful for accessing the Twitter API.
 - Before starting with the setup of the Client API, we need to import the tweepy library, then we need to do an authentication, afterwards we can use the functions, which are available in the library. [3]
- 2) **Pandas:** Pandas is a Python package used for data analysis.
 - The panda's library has useful functions for analyzing, cleaning, exploring, and manipulating datasets.

- Here I attached the screenshots of the code, For running the code, we need to import the necessary libraries, afterwards, we have taken the twitter credentials like Consumer_key, Consumer_secret, access_token and access_token_secret, now before crawling the data on twitter, we need to do authentication using the AuthHandler.[4]

```
[1] #Import required libraries
import tweepy
import numpy as np
import pandas as pd
import networkx as nx

[ ] #Twitter Developer Account: Keys and Tokens
consumer_key = '6Uzc8TaoYUyRxWPratQYThuh3'
consumer_secret = 'lyXlEnXgZ9SCGZ1pYTeMpd8qbgp5XFNew1pmZlmEQLReKKLM0y'
access_token = '1621995398897479680-QwVJslDD6D97Y66HojXYzjQ7lsfs8b'
access_token_secret = 'qq5F18xmUlQouPCzi4SKyzClBQVg68NgFwhKQeZgpzeo'

[ ] #Set up Twitter API using tweepy for crawling data on Twitter website
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True, compression=True)
```

- We have decided to take the friendship network for Illinois Institute of Technology, and specific college of computing department, the Twitter account is start with the '@IITcomputing', Now our task is to find the user_id with the screen time '@IITcomputing', after running the code successfully, we found the user_id = 917528090.

```
#We have decided to create a friendship network for Illinois Tech's College of Computing '@IITComputing' Twitter account
#Get the user id for user with screen name as '@IITComputing'
user = api.get_user(screen_name = '@IITComputing')
uid = user.id
print(uid)

917528090
```

- Now the below code is useful for getting the people, who have been followed by the @IITcomputing account.

```
#Crawl data for the accounts that @IITComputing follows
account = ["917528090"]
following = {}
for user in account:
    follow = []
    for page in tweepy.Cursor(api.friends_ids, user_id=user).pages():
        follow.extend(page)
    for i in follow:
        temp = api.get_user(i)
        following[i] = temp.screen_name
```

- After taking the following data from the screen_name = ‘ IITComputing’, Here we have printed the data, which have three labels like target, label and source.

```
#Following Data for '@IITComputing'
print(df)

      target      label      source
0    217147448  iitcsdept  917528090
1  1128095824473276418  KaiShu0327  917528090
2    1225586995  IITHumanScience  917528090
3  1166134633609814016  RealTechNewsIIT  917528090
4    296451422  RayTrygstad  917528090
..     ...
212   183102912  jamiepixi  917528090
213   17093617  hootsuite  917528090
214   218288929  IITAdmission  917528090
215   87246946  IITCampusLife  917528090
216   16932547  illinoistech  917528090

[217 rows x 3 columns]
```

- On the other hand, using the same way, we found the data for the followers, here there are three labels like source, label and target, here the target is fixed because we can consider target as root node.

```
#Followers Data for '@IITComputing'
print(df1)

      source      label      target
0  1621133261505429504  AnnaSrinivasan4  917528090
1  1621853970284044288  Lorenzo87567280  917528090
2  1623050008567095316  VatsalM05797052  917528090
3  1619934482055761920          astjzl  917528090
4    4184514795        Ssanidhya2  917528090
..     ...
453   726409806  IITCommunity  917528090
454   45899585  ScottPfeiffer54  917528090
455   87246946  IITCampusLife  917528090
456   16932547  illinoistech  917528090
457   218288929  IITAdmission  917528090

[458 rows x 3 columns]
```

- The below code is useful for creating the .csv file, as we have used the google colab Platform so used the below command; we have downloaded the .csv file into folder.

```
#Create csv file from dataframe  
df.to_csv('Following_Data.csv')
```

```
#Download csv file  
from google.colab import files  
files.download('Following_Data.csv')
```

```
#Create csv file from dataframe  
df1.to_csv('Followers_Data.csv')
```

```
#Download csv file  
from google.colab import files  
files.download('Followers_Data.csv')
```

2.2 Data Processing:

- Now after getting the data from .csv file, we have imported into excel, for data cleaning as some of the data may be duplicate so we can filter it.

The screenshot shows a Microsoft Excel spreadsheet titled "Followers Data". The data consists of three columns: "source", "label", and "target". The "label" column contains names of individuals, some of which are highlighted in blue. A network graph is overlaid on the data, where each node represents a person from the "label" column and edges represent connections between them based on the "source" and "target" columns. The network is color-coded by node degree, with higher-degree nodes appearing in darker shades of blue. The Excel ribbon at the top includes tabs for FILE, HOME, INSERT, PAGE LAYOUT, FORMULAS, DATA, REVIEW, VIEW, DEVELOPER, and POWERPIVOT. The "DATA" tab is selected. The "FORMULAS" tab has "Wrap Text" checked. The "PAGE LAYOUT" tab has "Merge & Center" checked. The "DATA" tab has "Format Painter" checked. The "REVIEW" tab has "AutoSum" checked. The "VIEW" tab has "Calculation" set to "Calculation". The "DEVELOPER" tab has "Check Cell" checked. The "POWERPIVOT" tab is visible. The status bar at the bottom right shows "9:00 AM".

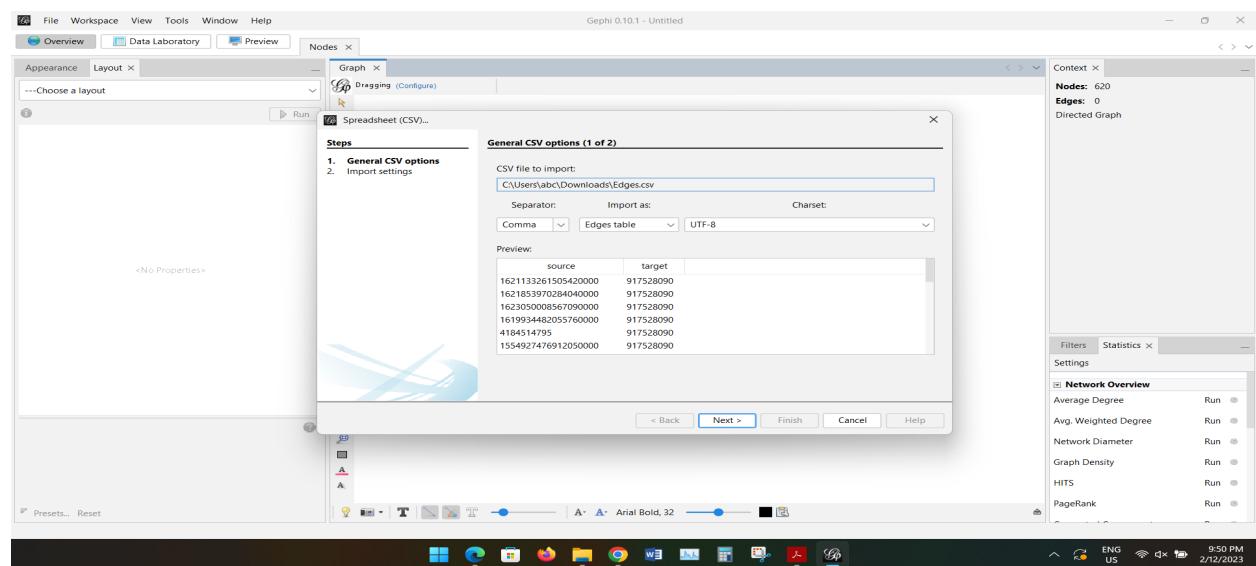
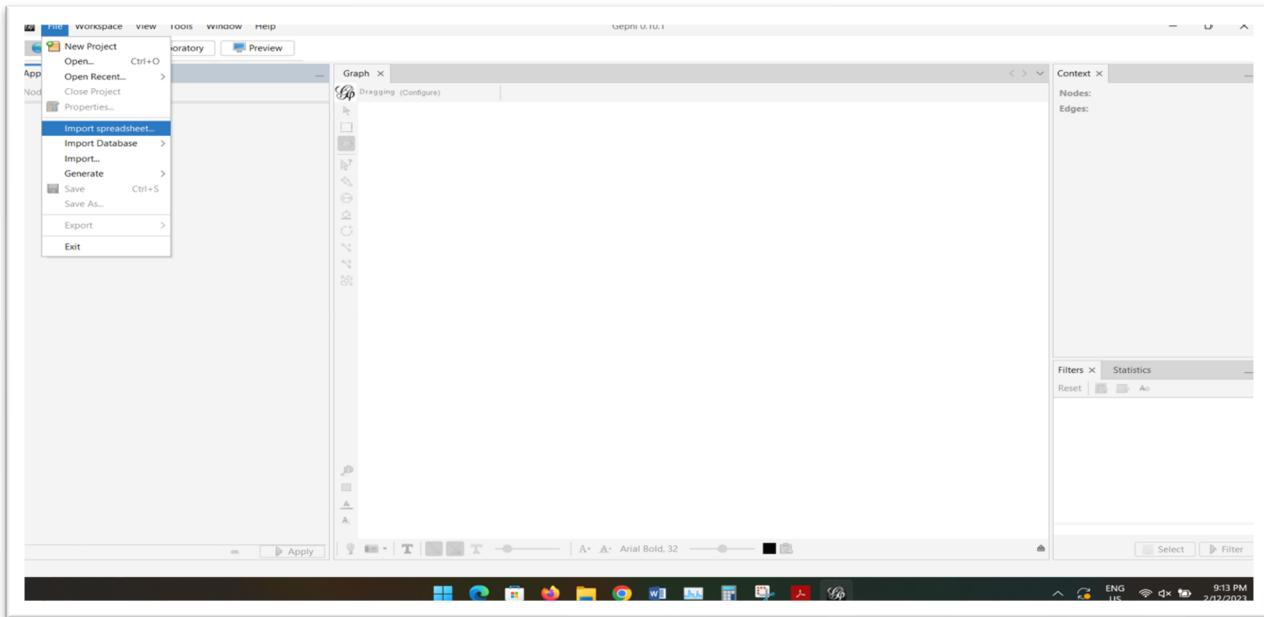
- Now for the nodes, we have found the 56 duplicate values, so have 620 unique nodes available for the id and label.

The screenshot shows a Microsoft Excel spreadsheet with data in columns A and B. Column A contains IDs and column B contains labels. A Microsoft Excel dialog box is open, stating "56 duplicate values found and removed; 620 unique values remain." The ribbon at the top includes tabs for FILE, HOME, INSERT, PAGE LAYOUT, FORMULAS, DATA, REVIEW, VIEW, DEVELOPER, and POWERPIVOT. The DATA tab is selected, showing various tools like Connections, Refresh, Sort & Filter, Data Tools, and Analysis.

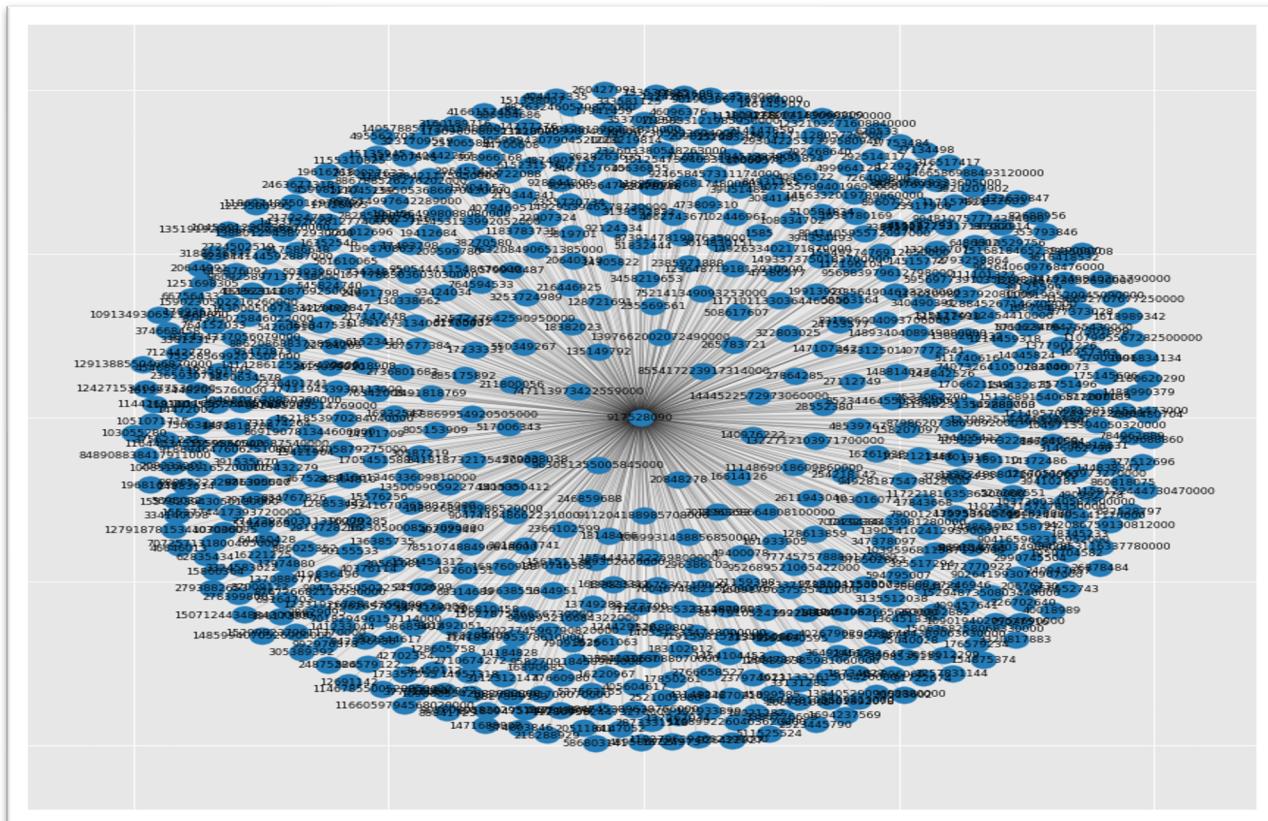
	A	B
1	Id	Label
2		917528090
3	16211338350542000	AnnaSrinivasan4
4	16218389709000000	VishnuKoz78567280
5	16245000567090000	VethaiM05797052
6	16199344820576000	astj2
7	16199344820576000	Ssandhya2
8	155492747691205000	Takville7
9	155492747691205000	tanay1
10	1590230502216250000	emekogonzalez
11	1529487350803440000	California8IC
12	1615231108769250000	nndub8910861
13	732603380548263000	AshkiHafiz
14	14926334021784739	ashkikhafiz
15	14926334021784739	Andresrococoe
16	1609179637535410000	Newell.Hancock
17	1388012543872930000	DineshCherupala3
18	1513689154068120000	alexvassar2025
19	1495076398541730000	Jahidulislam12
20	9081301815247730000	MayaGattu22
21	1489140404809860000	SciPolKate
22	1405788577441540000	TrinhGi43058640
23	102933899	EstLizhaifa
24	841818732179	caitlinmccormick
25	305118533	avangman
26	1480268410986520000	OriginaltRUFC
27	16352546	shadibeddas
28	1485994070526900000	AdianM1707
29	1141894995378610000	ManuelTrollis5
30	391535670	laudelg27

- Now for the edges, we have two labels, one is the source and another one is target.

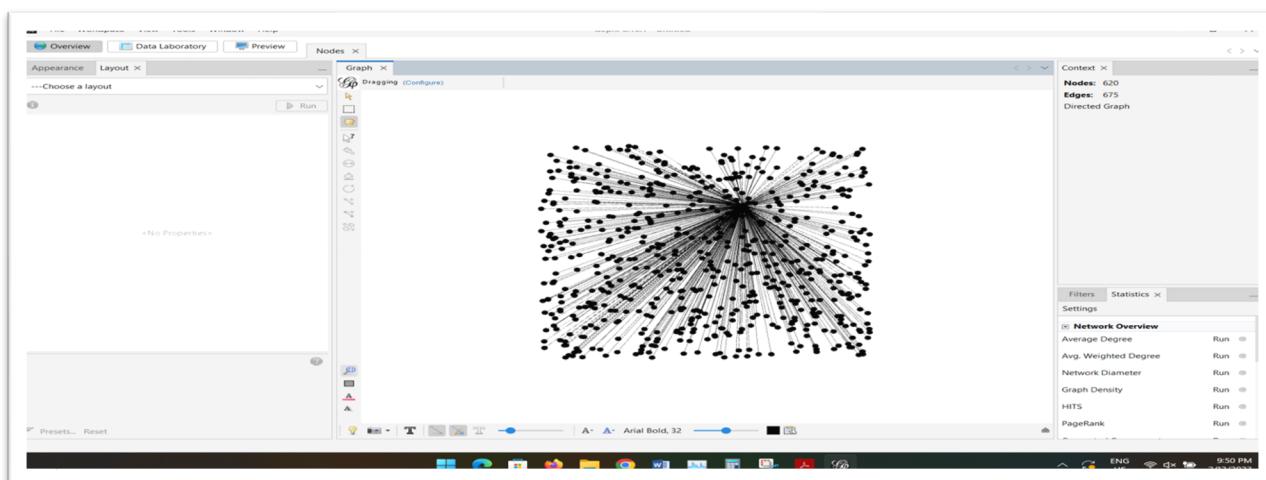
2.3 Data Visualization: Now for the data visualization part, we have used the Gephi, for the first step, we need to import the spreadsheet into .csv format.[6]



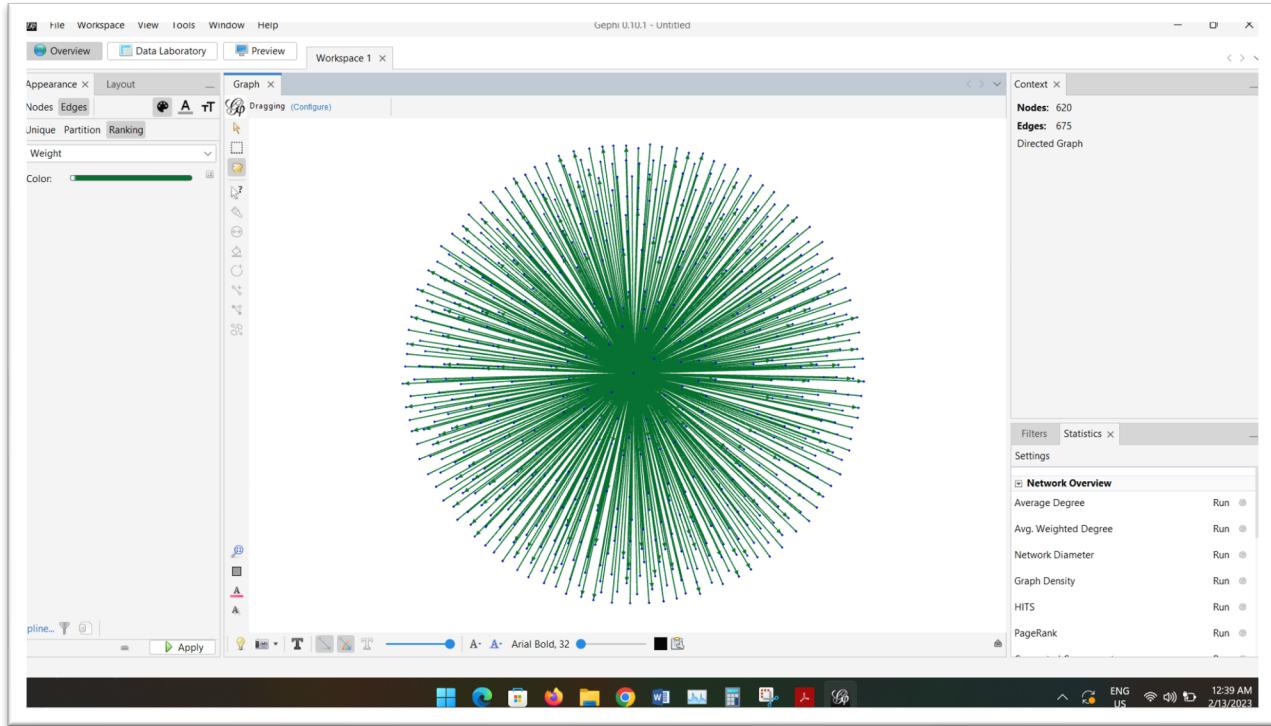
- In the below screenshot, The blue color represents the nodes, and the light black color represents the edges, here there are 620 unique nodes, here we used the user_id for representing the graph. It is showing the friendship network.



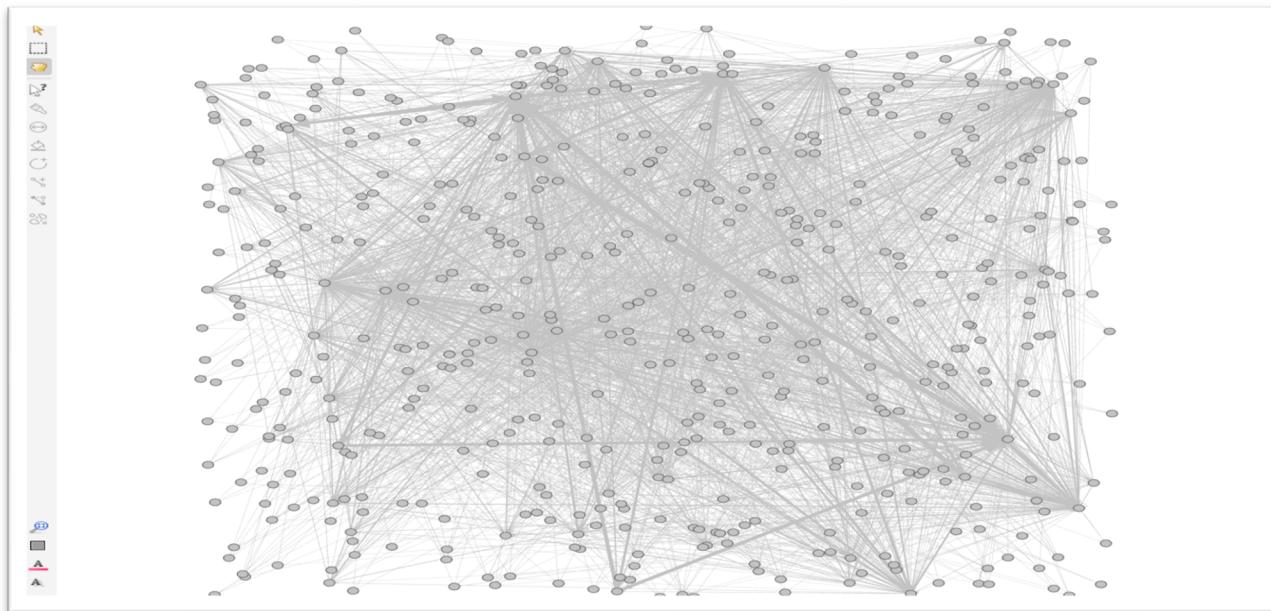
- The below graph showed the initial visualization of the nodes and edges graph, the followers that following the IITComputing.

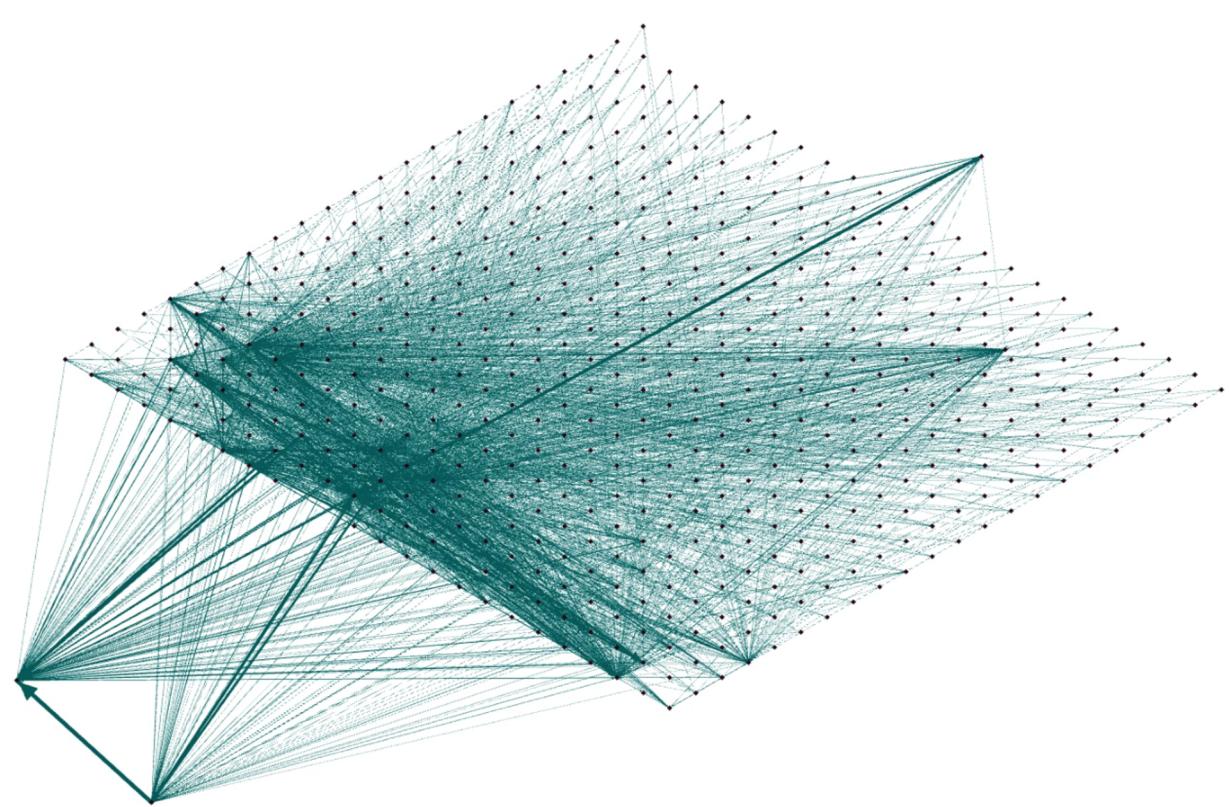
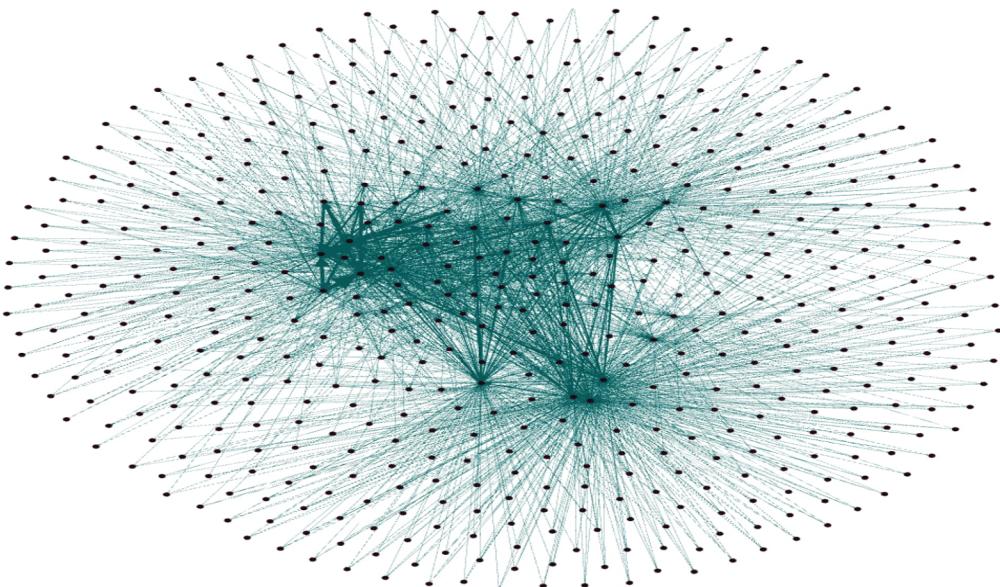


- The below graph represents the data visualization after applying appearance properties like the colors, size, label size and label colors. (This graph is only for the followers and following data sets).[9]



- In the below graph, the followers are nodes, and the edges show the connection between the followers and their followers, it is called the friendship network.



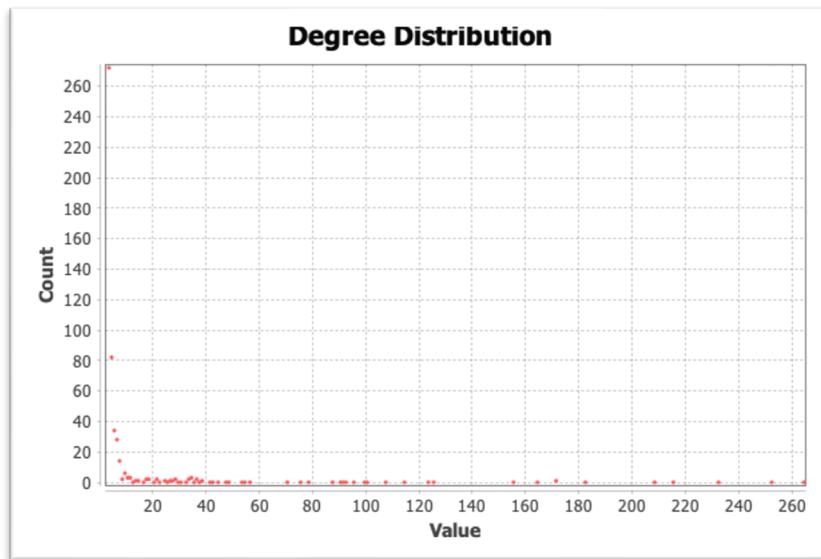


2.4 Network Measures:

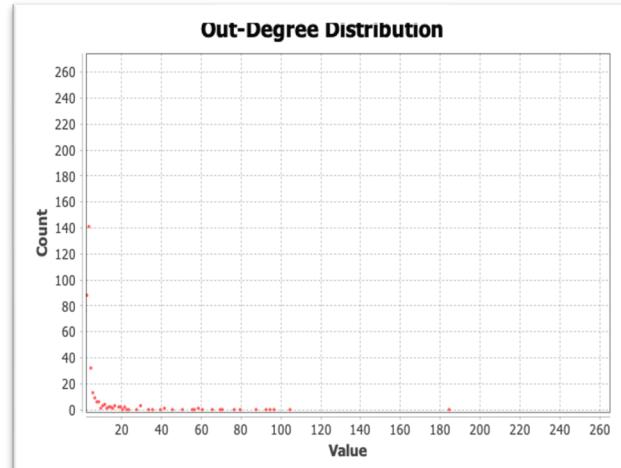
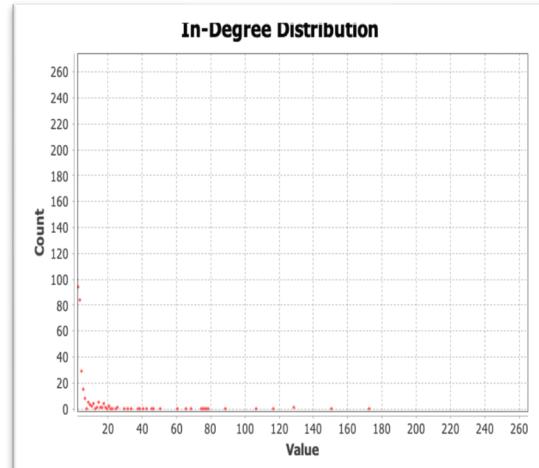
Degree Distribution: The degree of distribution of a graph is the probability distribution of the over the entire network.

Degree : The number of connections that node has with other nodes.

- In the below graph, the x axis represents the value(degree), and the y axis represent the count.
- The average distribution is 6.293, it means that one follower has connect with the average 6 peoples.[13]
- For the degree 1, there are above 260 followers connect with each other's.

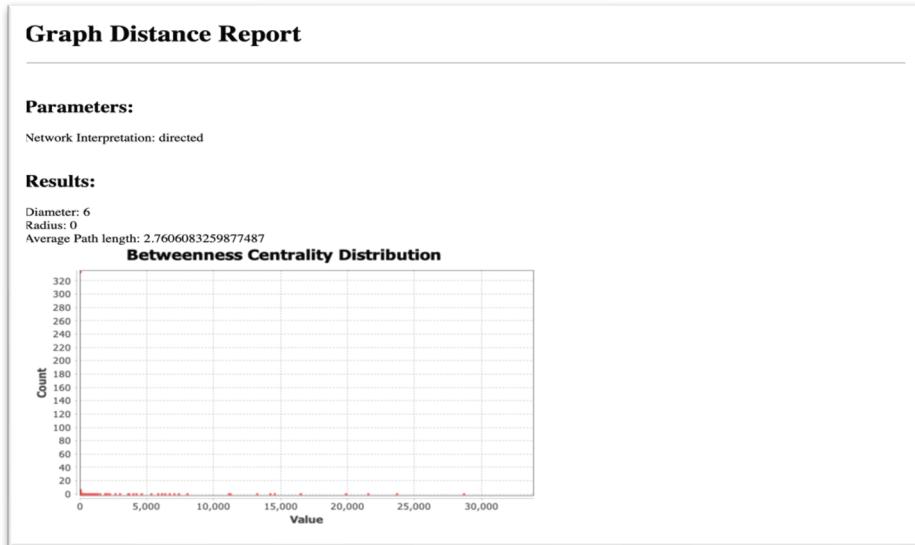


- The below graphs represent the In and Out Degree distribution.



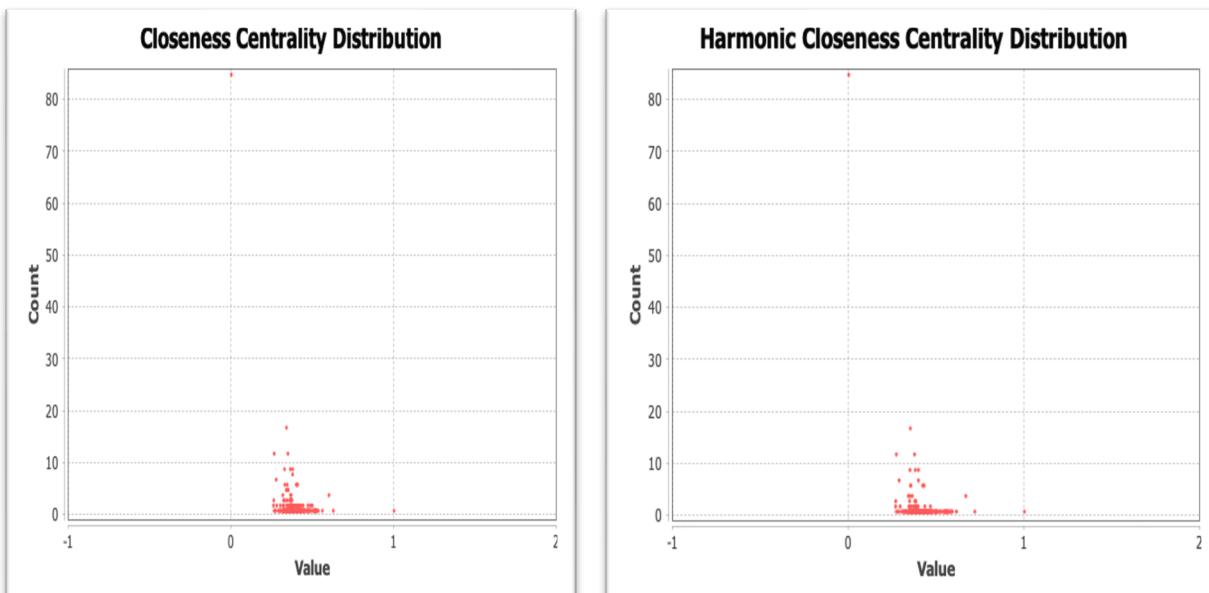
Diameter : The diameter of a graph is largest distance between the any pair of nodes.

- For the below graph, the largest distance for the any follower's node is six.[10]



Betweenness : The number of these shortest paths that cross through that vertex is the betweenness centrality for that vertex.[11]

Closeness : The closeness centrality of network is the inverse calculated sum of the length of the shortest paths between the node and all other nodes in the graph.(class notes)



Graph Density : The number of edges in directed graph divide by the maximum number of edges present into the graph. [12]

- The maximum edges are calculated using the number of nodes * (number of nodes – 1).
- Here the graph density for the directed graph is 0.012.



Section III: Team Efforts:

- 1) Social Network Selection and Background Researchers by Harsh and Henil
- 2) Coding for the followers and their followers by Harsh and Henil
- 3) Gephi Visualization by Harsh
- 4) Documentation by Henil

Section IV: References:

[1] Setting up Twitter Developer Account: <https://developer.twitter.com/en/support/twitter-api/developer-account>

[2] Twitter Developer Platform Documentation: <https://developer.twitter.com/en/support/twitter-api/developer-account>

[3] Tweepy: <https://www.tweepy.org/>

[4] Article on getting data from Twitter using API: <https://towardsdatascience.com/downloading-data-from-twitter-using-the-rest-api-24becf413875>

[5] NetworkX: <https://networkx.org/>

- [6] Gephi: <https://gephi.org/>
- [7] Gephi Tutorial: <https://gephi.org/users/quick-start/>
- [8] Tweepy Function: <https://www.jechouinard.com/tweepy-basic-functions/>
- [9] Gephi Layout : <https://www.linkedin.com/pulse/visualization-graph-techniques-different-layouts-kinjal-ami/>
- [10] Graph Diameter : <https://transportgeography.org/contents/methods/graph-theory-measures-indices/diametergraph/#:~:text=The%20diameter%20of%20a%20graph,of%20this%20graph%20is%204.>
- [11] Betweenness Centrality : <https://www.geeksforgeeks.org/betweenness-centrality-centrality-measure/>
- [12] Graph Density : <https://www.baeldung.com/cs/graph-density>
- [13] Graph Distribution : <https://youtu.be/UJgPTnBmAZM>