



## **CS-579 Online Social Network Analysis**

### **Project-1 Report**

On

### **Social Media Data Analysis**

By

#### **Name of Student**

1. Pujan Sheth
2. Akash Didigi Kashinath

#### **CWID**

- A20513790  
A20524076

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

**College Of Computing  
Illinois Institute of Technology  
Chicago, Illinois.**

**February 2023**

**CONTENTS:**

**1. Section – I: PROJECT DETAILS**

- 1.1 Project Statement

**2. Section – II: DATA COLLECTION**

- 2.1 Data Source and Authentication
- 2.2 Data Storing
- 2.3 Data Processing

**3. Section – III: DATA VISUALIZATION**

- 3.1 Creation of Nodes
- 3.2 Creation of Edges
- 3.3 Visualization Using Different Graphs

**4. Section – IV: NETWORK MEASURES**

- 4.1 Degree Distribution
- 4.2 Betweenness Centrality
- 4.3 Closeness Centrality Distribution

**5. Section – V: TEAM CONTRIBUTION**

**6. Section – VI: REFERENCES**

## Section I: Project Details

### **1.1- Project Statement:**

The aim of this project is to crawl social media data as well as process and report some analysis on the extracted data. For this project we have chosen Twitter as our social media platform. We have collected the data from the Twitter with the help of its developer account to create a social network. For creating this social network, we have chosen friendship network with 100 – 500 nodes and visualized the network as a graph using NetworkX and plotted network measures such as degree distribution, betweenness centrality and closeness centrality distribution.

## Section II: Data Collection

### **2.1- Data Source and Authentication:**

We have created a Twitter developer account based on the instructions provided in the problem statement. After our request was approved, we had access to the API key, API key secret, access token and access token secret with which we were able to call invoke Twitter API calls in our project.

```
# Twitter authentication
auth = tw.OAuthHandler(
    os.environ["api_key"],
    os.environ["api_key_secret"],
)
auth.set_access_token(
    os.environ["access_token"],
    os.environ["access_token_secret"],
)

# Creating an API object, the api will wait if the rate limit exceeds
# NOTE:- API rate limit is 180 calls every 15 mins
# wait_on_rate_limit=True use this to apply sleep of 15 min
api = tw.API(auth)
```

### **2.2 - Data Storing:**

Since the Twitter API was being moved to paid version after February 9<sup>th</sup>, we have decided to store the data in a JSON file before the deadline, so that there would be no issues going forward with project. This file acts as a data source for our project.

```

def startupCheck():
    PATH = "./data.json"
    if os.path.isfile(PATH) and os.access(PATH, os.R_OK):
        # checks if file exists
        print("File exists and is readable")
    else:
        print("Either file is missing or is not readable, creating file....")
        with io.open("data.json", "w") as db_file:
            # profile_link will contain all users and thier profile picture
            # connections is adjacency list to form a network
            # screen_names will act as queue in order to expand the network
            db_file.write(
                json.dumps(
                    {
                        "profile_link": {},
                        "connections": {},
                        "starting_name": "ps_pujansheth",
                        "screen_names": ["ps_pujansheth"],
                    }
                )
            )

startupCheck()

File exists and is readable

```

Below is the screenshot of the JSON file for few users that is stored in the local file system.

```

OSNA PJ1.ipynb  {} data.json x

Users > akash > Desktop > OSNA > project1 > {} data.json > ...

1  {"profile_link": {"ps_pujansheth": "https://pbs.twimg.com/profile_images/602014165453045760/sXKYCQod_normal.jpg",
"DPatel205Patel": "https://pbs.twimg.com/profile_images/1621599795000352773/ciFZit9m_normal.png", "irvingjoanne17": "https://pbs.twimg.com/profile_images/1595444276611239941/iZXnSyot_normal.jpg", "Pari1602": "https://pbs.twimg.com/profile_images/1280529845139587072/45amTj1x_normal.jpg", "mbarot69": "https://pbs.twimg.com/profile_images/1611569015142645760/cuqcGINJ_normal.jpg", "kush0099": "https://pbs.twimg.com/profile_images/1501568268263313415/TNbB1fM7_normal.jpg", "payal_pabari": "https://pbs.twimg.com/profile_images/902251597752885249/IVdQqfn6_normal.jpg", "MrDavid1901": "https://pbs.twimg.com/profile_images/1619293271922188289/kFTwk6WQ_normal.jpg", "JubinYadav1": "https://pbs.twimg.com/profile_images/1412783181095333892/puU7sJlw_normal.jpg", "JPL_India": "https://pbs.twimg.com/profile_images/1555260451046707200/m9MuZEwU_normal.jpg", "RashiJoshi19": "https://pbs.twimg.com/profile_images/1507276753928351746/xetSZBjB_normal.jpg", "Wright_habits": "https://pbs.twimg.com/profile_images/1559896090899611654/Xdvp51FI_normal.jpg", "HookedSounds": "https://pbs.twimg.com/profile_images/1623118805286244354/jYZ0fNGv_normal.jpg", "JeD87260622": "https://pbs.twimg.com/profile_images/1623911748683333632/6ANCInpk_normal.jpg", "LhordRicchi1": "https://pbs.twimg.com/profile_images/1620562049414619136/5G0GZW0J_normal.jpg", "Hasa007788": "https://pbs.twimg.com/profile_images/1480740316785221633/xtXD6vPD_normal.jpg", "FnSbyruqayyah": "https://pbs.twimg.com/profile_images/1480740316785221633/xtXD6vPD_normal.jpg"} ...

```

### 2.3- Data Processing:

For processing the data that was stored in the previous step, we have loaded the JSON file that has all the information saved and extracted all the user details along with their profile pictures. Below is the output for few users.

```

['ps_pujansheth': 'https://pbs.twimg.com/profile_images/602014165453045760/sXKYCQod_normal.jpg', 'DPatel205Patel':
'https://pbs.twimg.com/profile_images/1621599795000352773/ciFZit9m_normal.png', 'irvingjoanne17': 'https://pbs.twimg.com/profile_images/1595444276611239941/iZXnSyot_normal.jpg',
'Pari1602': 'https://pbs.twimg.com/profile_images/1280529845139587072/45amTj1x_normal.jpg', 'mbarot69':
'https://pbs.twimg.com/profile_images/1611569015142645760/cuqcGINJ_normal.jpg', 'kush0099': 'https://pbs.twimg.com/profile_images/1501568268263313415/TNbB1fM7_normal.jpg',
'payal_pabari': 'https://pbs.twimg.com/profile_images/902251597752885249/IVdQqfn6_normal.jpg', 'MrDavid1901':
'https://pbs.twimg.com/profile_images/1619293271922188289/kFTwk6WQ_normal.jpg', 'JubinYadav1': 'https://pbs.twimg.com/profile_images/1412783181095333892/puU7sJlw_normal.jpg',
'JPL_India': 'https://pbs.twimg.com/profile_images/1555260451046707200/m9MuZEwU_normal.jpg', 'RashiJoshi19':

```

For creating a network of friends of friends, we started the network with our Twitter ID and collected data for all our followers and collected at most 10 followers for each of our followers along with their profile pictures. For this data, we have used tweepy's api.get\_followers method. We then saved all the data regarding friends of friend. Based on this data we have written code such that the number of nodes is collected up to 500. Once the data is collected up to 500 nodes, we will stop appending data.

Below is the screenshot of the output of the code which includes adding a node of my followers of ps\_pujansheth (me) including me.

```

Starting at ps_pujansheth
Adding a node with screen name ps_pujansheth
Adding a node with screen name irvingjoanne17
Adding a node with screen name Pari1602
Adding a node with screen name mbbarot69
Adding a node with screen name kush0099
Adding a node with screen name payal_pabari
Adding a node with screen name MrDavid1901
Adding a node with screen name JubinYadav1
Adding a node with screen name JPL_India
Adding a node with screen name RashiJoshi19
Adding a node with screen name PenhaAllen
Adding a node with screen name michaelvbuffalo
Adding a node with screen name Wright_habits
Adding a node with screen name HookedSounds
Adding a node with screen name JeD87260622
<class 'Exception'>
Leaving the program due to exception
Saving the connections to the json file
Stopped the streaming due to an error
Stopped the streaming

```

## Section III: Data Visualization

### 3.1 - Creation of Node:

For creating nodes, we have extracted the name of the user and profile picture of all the followers i.e., the nodes which were created in the previous step and for the node of followers of followers in the data variable. The maximum number of nodes we considered are 500 nodes.

```

['ps_pujansheth', 'irvingjoanne17', 'Pari1602', 'mbbarot69', 'kush0099', 'payal_pabari', 'MrDavid1901', 'JubinYadav1', 'JPL_India', 'RashiJoshi19', 'PenhaAllen', 'michaelvbuffalo', 'Wright_habits', 'HookedSounds', 'JeD87260622', 'LhordRicchi1', 'Hasa007788', 'FnSbyruqayyah', 'maxine_dwnunder', 'SapetaJes', 'WIZZYCHRISMUST3', 'SwatiRSaha', 'BhattKesha2806', 'MarooMansi', 'siddharth_sonak', 'PawaskarKeyur', 'kimaya_mhatre_', 'aditi_metkar', 'AdityaSatpurkar', 'durgappadk9', 'pandeyvinay1297', 'Rajeevshuklabjp', 'JuhiDeshmukh1', 'Jitharshjaihind', 'Jayprak67848307', 'Rajusingh990', 'Sanjuve2024', 'shaileshbrahm', 'Supriya00646442', 'drmahendra069', 'Kokoombu', 'jaggim', '1710RAJ', 'shottrader2006', 'Takfaheem', '9cf94c8bf7c6472', 'csheth12', 'keshavimehta13', 'talati_palak', 'IngriddeBock', 'TheSecretLawofA', 'KRAFTON_BGMI', 'bhavya10373', 'msuper20506679', 'hockenberrycaro', 'DavidSSKCapital', 'barker_mildred9', 'Mulyadi58882863', 'dividendinvest', 'SarasSheikh05', 'Shiori11824613', 'rei14530718', 'moore_andrea44', 'oliviabbxo', 'fiona7669', 'Keithpp45165615', 'TheDailyHPotter', 'shomer_A', 'yashsoni07', 'khazakh', 'bhavyashah983', 'tururueyes', 'IslaDelaney25', 'RajendraThakkar', 'rahulkirtikar', 'TTandiwala', 'akshat54', 'Pimpleshank', 'RM_Rajdeep007', 'Prakash27061996', 'haidar2930', 'Harsh_Sudan', 'ishapatel999', 'yash165_', 'RaniTutun', 'VILAS_KAVAR', 'vorajanam4545', 'aditya5070', 'KM1085S', 'deeppotter', 'shahvaii1916', 'harshalshinde11', 'MamboDavid7', 'vallabh_rajesh', 'chinmaypednekar', 'Cr7Ronit', 'Nisha53415744', 'Simmisolanki23', 'ArzanAzmi', 'itsmoksbitch', 'TrolledByDewey', 'TacticalPear', 'eythnwarnes05', 'BeerlyHockey', 'JackEichBUST', 'UMBILLIEVABLE_', 'corey_lew3', 'ThomasGrundner3', 'NatoApril', 'prideauxcindy17', 'hamedabdulrah10', 'chimeManifest12', 'uoerawe', '__19_keys__', 'Pch0Pch', 'CASHAPP_MONEY1_', '__19key_s__', 'CashApp_4U_2K', 'brendah44318421', 'DJEMzDnb', 'absstephens99', 'Bellaelisedolce', 'emilytjoness', 'Unstable_Lines', 'houseartistman', 'AflekUK', 'manik_dnb', 'TurrallC98']

```

### **3.2-Creation of Edges:**

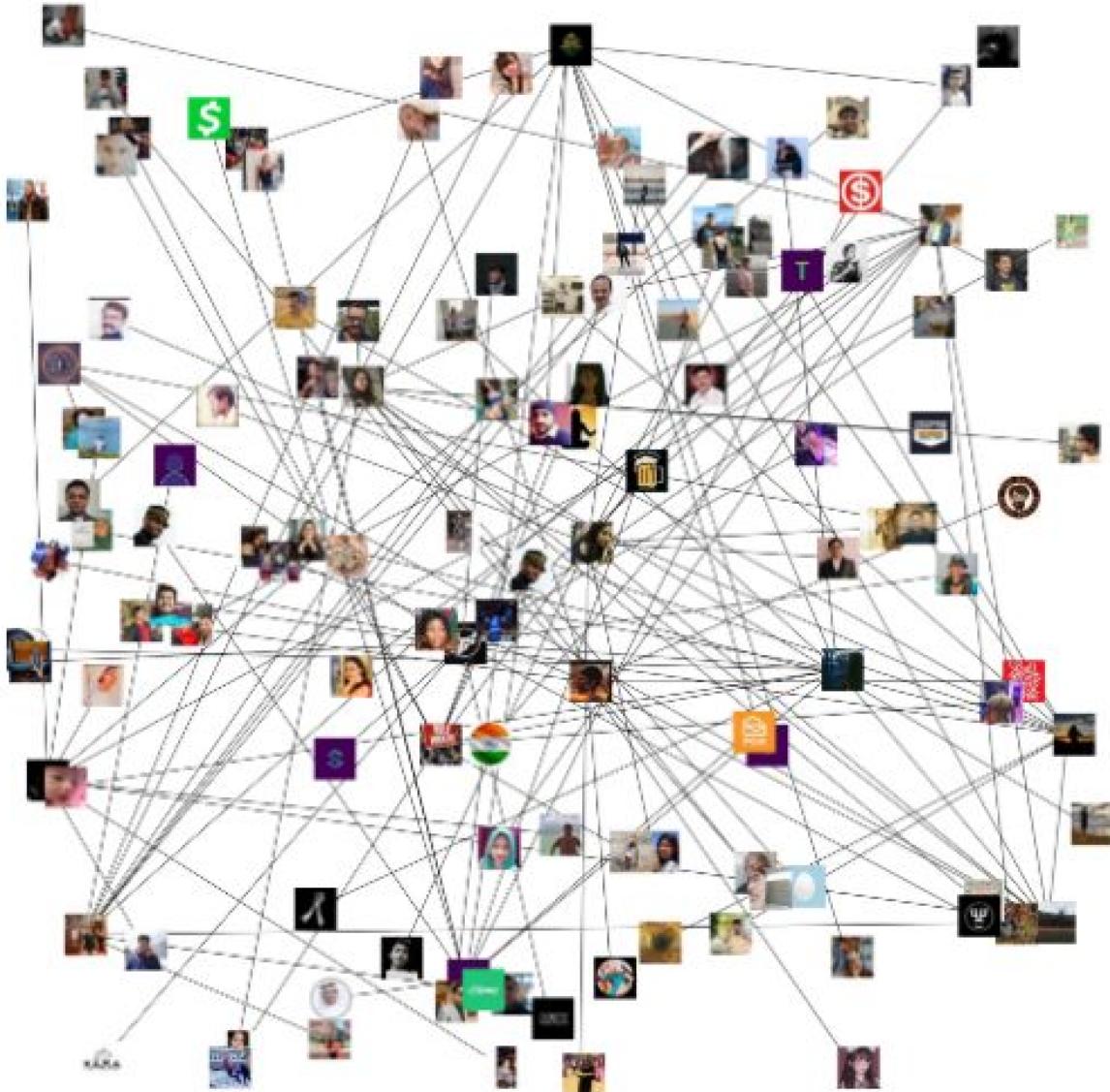
Once the nodes are created, the next step was to create an edge between all the nodes of my followers and followers of followers. We have written a code to create an edge among all the nodes which were created in the previous step. Below is the output of all the pairs of nodes which have an edge.

```
[('ps_pujansheth', 'irvingjoanne17'), ('ps_pujansheth', 'Pari1602'), ('ps_pujansheth', 'mbbarot69'), ('ps_pujansheth', 'kush0099'), ('ps_pujansheth', 'payal_pabari'), ('ps_pujansheth', 'MrDavid1901'), ('ps_pujansheth', 'JubinYadav1'), ('ps_pujansheth', 'JPL_India'), ('ps_pujansheth', 'RashiJoshi19'), ('ps_pujansheth', 'PenhaAllen'), ('irvingjoanne17', 'michaelvbuffalo'), ('irvingjoanne17', 'Wright_habits'), ('irvingjoanne17', 'HookedSounds'), ('irvingjoanne17', 'JeD87260622'), ('irvingjoanne17', 'LhordRicch1'), ('irvingjoanne17', 'Hasa007788'), ('irvingjoanne17', 'FnSbyruqayyah'), ('irvingjoanne17', 'maxine_dwnunder'), ('irvingjoanne17', 'SapetaJes'), ('irvingjoanne17', 'WIZZYCHRISMUSI3'), ('Pari1602', 'SwatiRKSAha'), ('Pari1602', 'BhattKesha2806'), ('Pari1602', 'MarooMansi'), ('Pari1602', 'siddharth_sonak'), ('Pari1602', 'PawaskarKeyur'), ('Pari1602', 'kimaya_mhatre'), ('Pari1602', 'aditi_metkar'), ('Pari1602', 'AdityaSatpurkar'), ('Pari1602', 'durgappadk9'), ('mbbarot69', 'pandeyvinay1297'), ('mbbarot69', 'RajeevShuklaBjp'), ('mbbarot69', 'JuhiDeshmukh1'), ('mbbarot69', 'Jitharshjaihind'), ('mbbarot69', 'Jayprak67848307'), ('mbbarot69', 'RajuSingh990'), ('mbbarot69', 'Sanjuve2024'), ('mbbarot69', 'shaileshbrahm'), ('mbbarot69', 'Supriya00646442'), ('mbbarot69', 'drmahendra3069'), ('kush0099', 'KokoOmbu'), ('kush0099', 'jaggirm'), ('kush0099', '1710RAJ'), ('payal_pabari', 'shottrader2006'), ('payal_pabari', 'Takfaheem'), ('payal_pabari', '9cf94c8bf7c6472'), ('payal_pabari', 'csbeth12'), ('payal_pabari', 'keshavimehta13'), ('payal_pabari', 'talati_palak'), ('payal_pabari', 'IngridBock'), ('payal_pabari', 'TheSecretLawofA'), ('payal_pabari', 'KRAFTON_BGNI'), ('payal_pabari', 'bhavya10373'), ('MrDavid1901', 'msuper20506679'), ('MrDavid1901', 'hockenberrycaro'), ('MrDavid1901', 'DavidSSKCapital'), ('MrDavid1901', 'barker_mildred9'), ('MrDavid1901', 'Mulyadi58882863'), ('MrDavid1901', 'd1videndinvest'), ('MrDavid1901', 'SaraSheikh05'), ('MrDavid1901', 'Shiori11824613'), ('MrDavid1901', 'rei14530718'), ('MrDavid1901', 'moore_andrea44'), ('JubinYadav1', 'oliviabbxo'), ('JubinYadav1', 'fiona7669'), ('JubinYadav1', 'Keithpp45165615'), ('JubinYadav1', 'TheDailyHPotter'), ('JubinYadav1', 'shomer_A'), ('JubinYadav1', 'yashsoni07'), ('JubinYadav1', 'khazazakh'), ('JubinYadav1', 'bhavyashah983'), ('JubinYadav1', 'tururueyes'), ('JPL_India', 'IslaDelaney25'), ('JPL_India', 'RajendraThakkar'), ('JPL_India', 'rahulkirtikar'), ('JPL_India', 'TTandiwala'), ('JPL_India', 'akshat54'), ('JPL_India', 'PimpleSushank'), ('JPL_India', 'RM_Rajdeep007'), ('JPL_India', 'Prakash27061996'), ('JPL_India', 'haidar2930'), ('JPL_India', 'Harsh_Sudan'), ('RashiJoshi19', 'ishapate1999'), ('RashiJoshi19', 'yash165_'), ('RashiJoshi19', 'RaniTutun'), ('RashiJoshi19', 'VILAS_KAVAR'), ('RashiJoshi19', 'vorajanam4545'), ('RashiJoshi19', 'aditya5070'), ('RashiJoshi19', 'KM108SS'), ('RashiJoshi19', 'deeppotter'), ('RashiJoshi19', 'shahvai1916'), ('RashiJoshi19', 'harshalshinde11'), ('PenhaAllen', 'MamboDavid7'), ('PenhaAllen', 'vallabh_rajesh'), ('PenhaAllen', 'chimmaypednekar'), ('PenhaAllen', 'Cr7Ronit'), ('PenhaAllen', 'Nisha53415744'), ('PenhaAllen', 'Simmisolanki23'), ('PenhaAllen', 'ArzanAzmi'), ('PenhaAllen', 'itsmokshbitch'), ('michaelvbuffalo', 'TrolledByDewey'), ('michaelvbuffalo', 'TacticalPear'), ('michaelvbuffalo', 'eythnwarnes05'), ('michaelvbuffalo', 'BeerlyHockey'), ('michaelvbuffalo', 'JackEich8UST'), ('michaelvbuffalo', 'UNBILLIEABLE_'), ('michaelvbuffalo', 'corey_lew3'), ('michaelvbuffalo', 'ThomasGrundner3'), ('michaelvbuffalo', 'NatoApril'), ('Wright_habits', 'prideauxcindy17'), ('Wright_habits', 'hamedabdulrah10'), ('Wright_habits', 'chimeManifest12'), ('Wright_habits', 'uoerawe'), ('Wright_habits', '__19_keys__'), ('Wright_habits', 'Pch0Pch'), ('Wright_habits', 'CASHAPP_MONEY1_'), ('Wright_habits', '__19key_s__'), ('Wright_habits', 'CashApp_4U_2K'), ('Wright_habits', 'brendah44318421'), ('HookedSounds', 'DJEmzDnb'), ('HookedSounds', 'absstephens99'), ('HookedSounds', 'Bellaelisedolce'), ('HookedSounds', 'emilytjoness'), ('HookedSounds', 'Unstable_Lines'), ('HookedSounds', 'houseartistman'), ('HookedSounds', 'AflekUK'), ('HookedSounds', 'manik_dnb'), ('HookedSounds', 'TurrallC98')]
```

### **3.3-Visualization Using Different Graphs:**

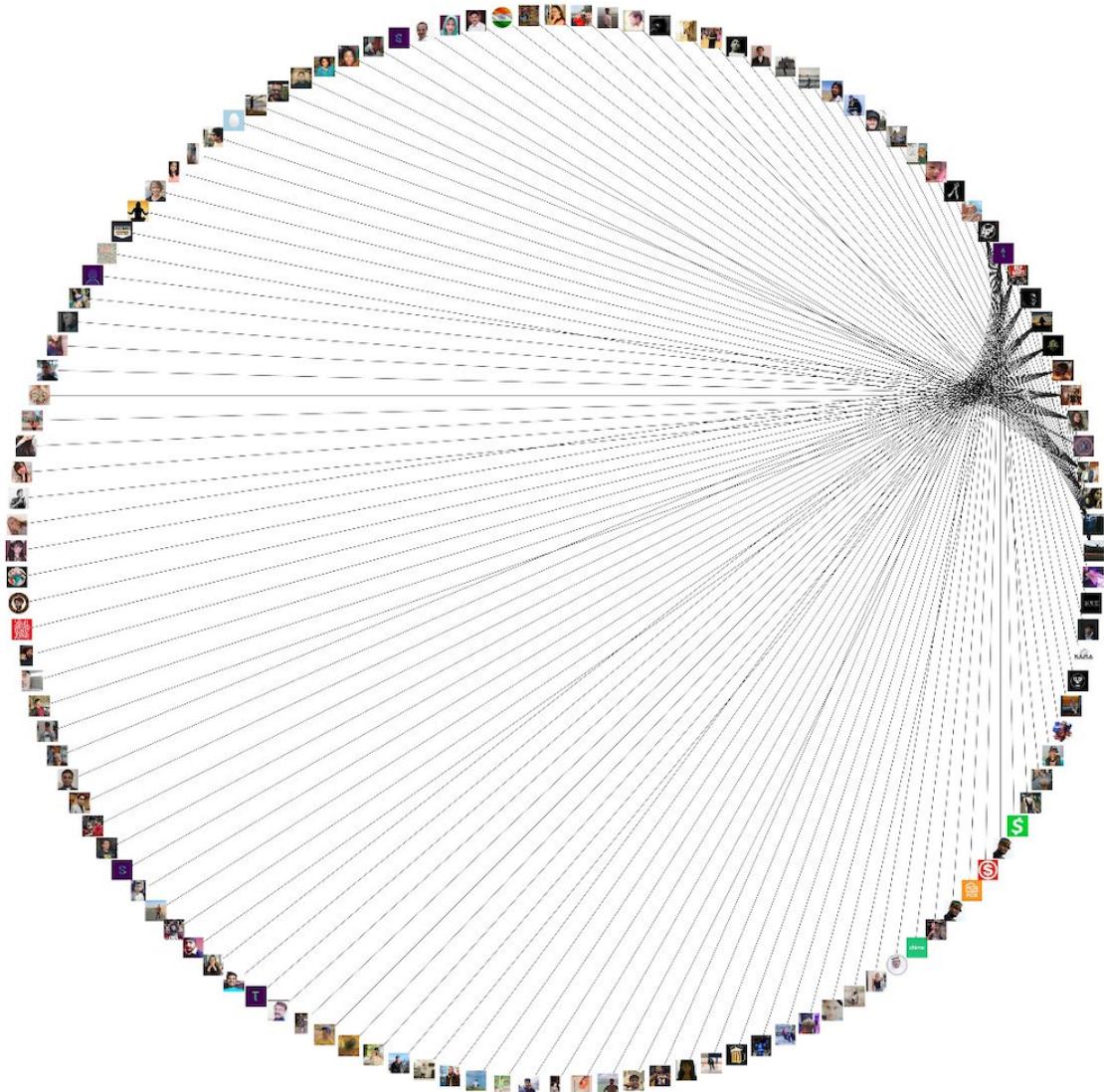
Once the above nodes and edges among all the nodes have been established, we have experimented with different types of graphs available on the NetworkX software.

- This below layout is the depiction of the random layout of all the nodes which represents my followers and the followers of my followers and the edges connecting the followers of a particular node. We have used the library of NetworkX library called nx.random\_layout on the data after the nodes and edges have been created.



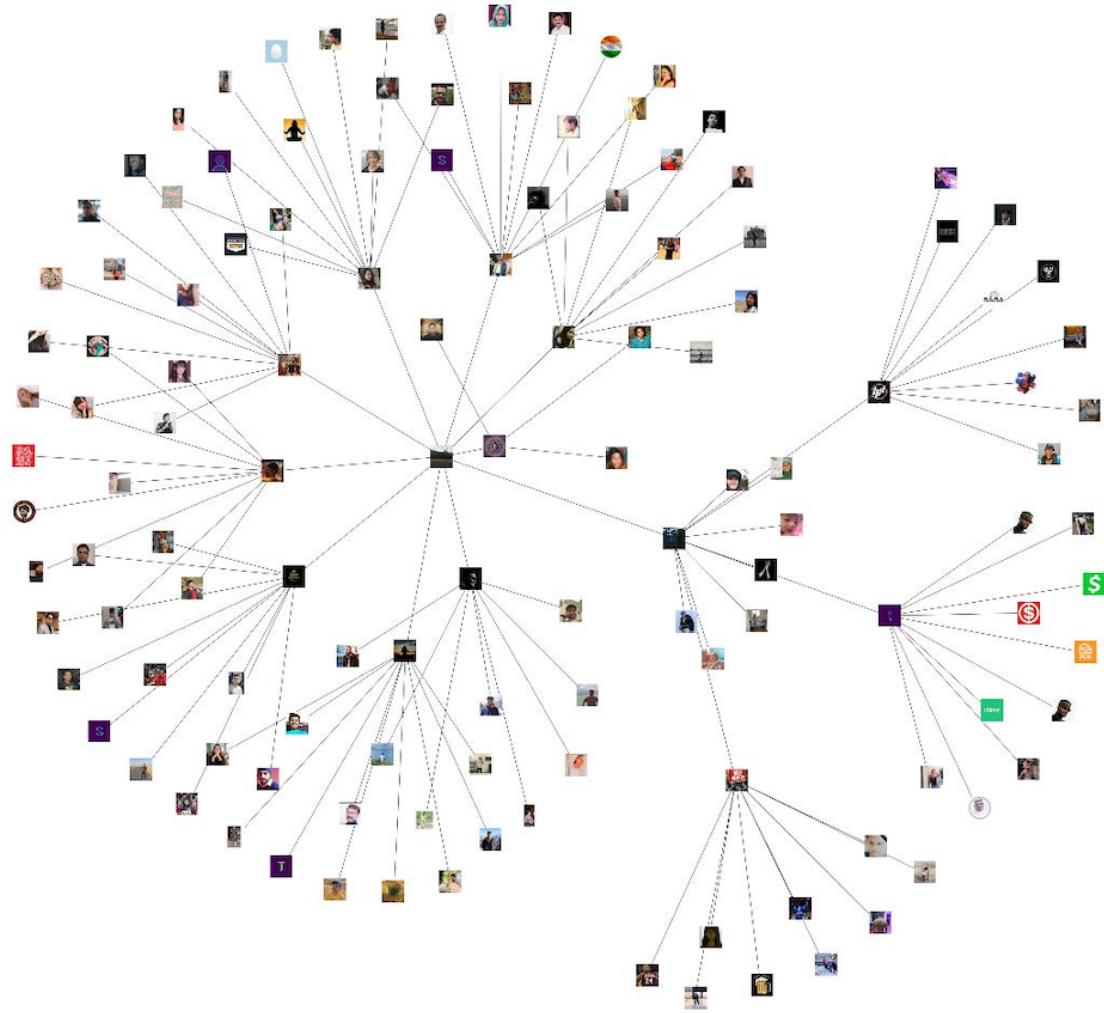
## Random Layout

- The next graph we experimented from the NetworkX was circular layout using its `nx.circular_layout`. The below graph clearly depicts the connection of all nodes with the edges.



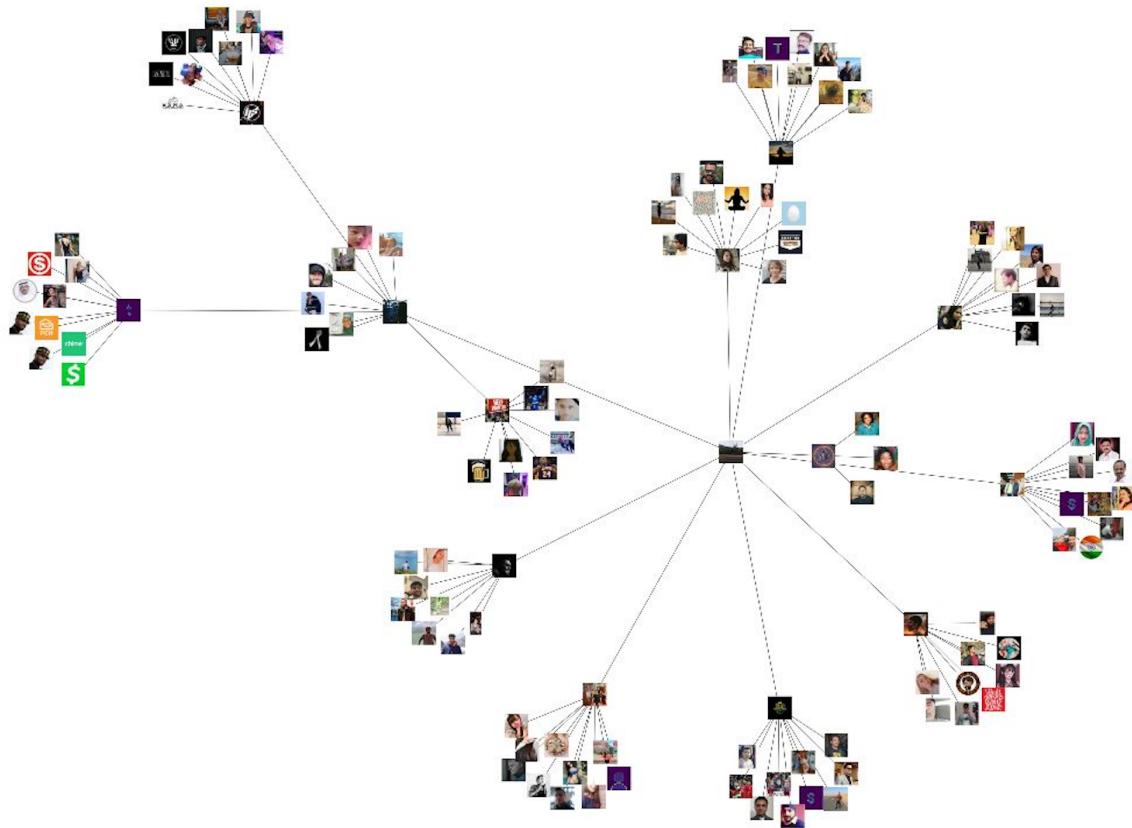
## Circular Layout

- The graph which we wanted to explore was the kamada kawai layout of the NetworkX using `nx.kamada_kawai_layout`. The main reason we choose this depiction of the all the nodes and edges to clearly represent the connection of the followers and their followers and the mutual connection that the followers and their followers have.



## Kamada Kawai Layout

- The final graph we plotted is the Spring Layout of the NetworkX using its `nx.spring_layout`. Which clearly graphs me the primary node at the center and all my followers at one level and their followers in the second layer.



**Spring Layout**

## Section IV: Network Measures

### **4.1- Degree Distribution:**

For plotting the degree measures we calculated the degree sequence which contains sorted degree and took with maximum degree. We haven then plotted the largest connected component as shown below to find the Degree distribution of the nodes [8].

```
# largest connected component
components = nx.connected_components(G)
largest_component = max(components, key=len)
H = G.subgraph(largest_component)
```

```

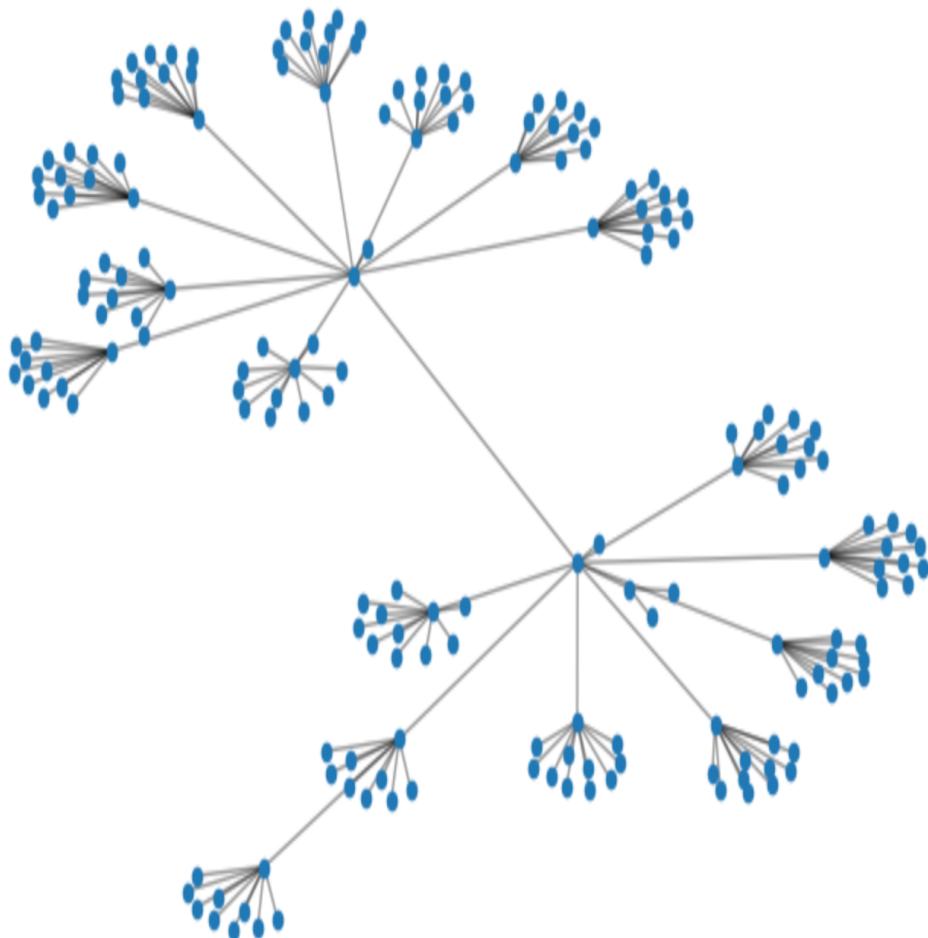
6.degree

✓ 0.0s

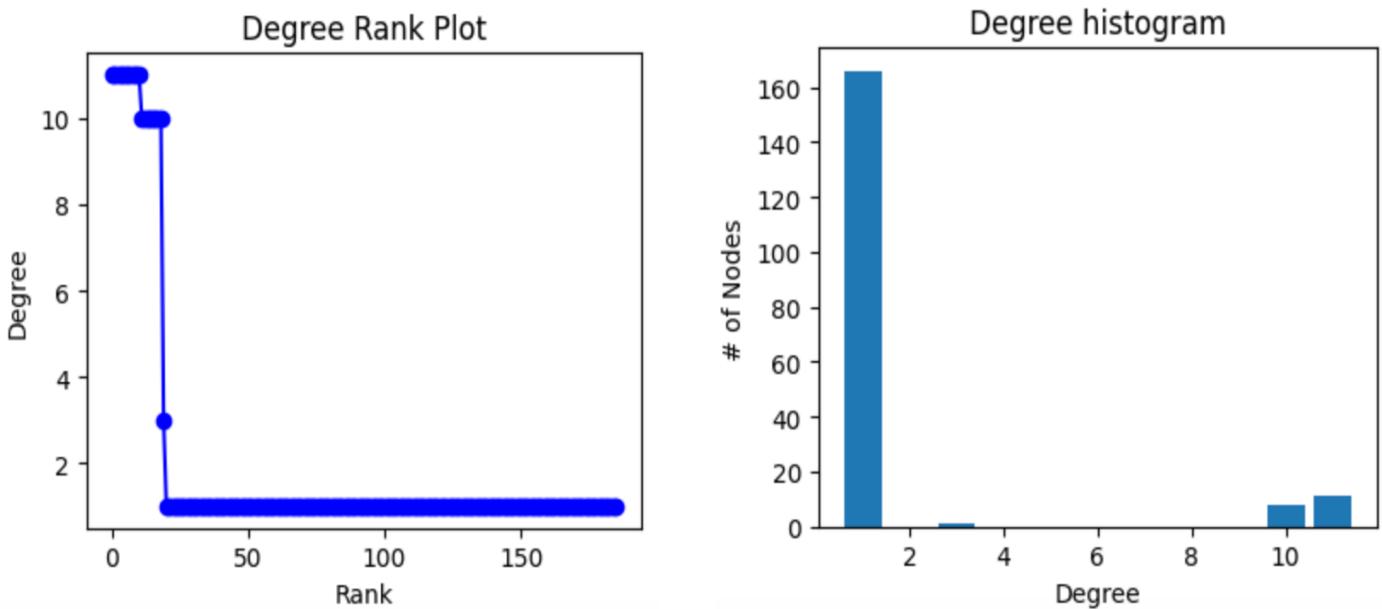
DegreeView({'ps_pujansheth': 10, 'irvingjoanne17': 11, 'Pari1602': 10, 'mbbarot69': 11, 'kush0099': 4, 'payal_pabari': 11, 'MrDavid1901': 11, 'JubinYadav1': 10, 'JPL_India': 11, 'RashiJoshi19': 11, 'PenhaAllen': 9, 'michaelvbuffalo': 10, 'Wright_habits': 11, 'HookedSounds': 10, 'JeD87260622': 1, 'LhordRicchi': 1, 'Hasa007788': 1, 'FnSbyruqayyah': 1, 'maxine_dnunder': 1, 'SapetaJes': 1, 'WIZZYCHRISMUSI3': 1, 'SwatiRKSaha': 1, 'BhattKesha2806': 1, 'MarooMansi': 1, 'siddharth_sonak': 1, 'PawaskarKeyur': 1, 'kimaya_mhatre_': 1, 'aditi_metkar': 1, 'AdityaSatpurkar': 1, 'durgappadk9': 1, 'pandeyvinay1297': 1, 'RajeevShuklaBjp': 1, 'JuhiDeshmukhi': 1, 'Jitharshjaihind': 1, 'Jayprak67848307': 1, 'RajusSingh990': 1, 'Sanjuve2024': 1, 'shaileshbrahm': 1, 'Supriya00646442': 1, 'drmahendra3069': 1, 'KokoOmbu': 1, 'jaggirm': 1, '1710RAJ': 1, 'shottrader2006': 1, 'Takfaheem': 1, '9cf94c8bf7c6472': 1, 'csheh12': 1, 'keshawimehata13': 1, 'talatipalak': 1, 'IngriddleBock': 1, 'TheSecretLawofA': 1, 'KRAFTON_BGMI': 1, 'bhavya10373': 1, 'msuper20506679': 1, 'hockenberrycaro': 1, 'DavidSSKCapital': 1, 'barker_mildred9': 1, 'Mulyadi58882863': 1, 'dividendinvest': 1, 'SarasSheikh05': 1, 'Shiori11824613': 1, 'rei14530718': 1, 'moore_andrea44': 1, 'oliviabbbx': 1, 'fiona7669': 1, 'Keithpp45165615': 1, 'TheDailyHPotter': 1, 'shomer_A': 1, 'yashsoni07': 1, 'khazakah': 1, 'bhavyashah983': 1, 'tururueyes': 1, 'IslaDelaney25': 1, 'RajendraThakkar': 1, 'rahulkirtikar': 1, 'TTandiwala': 1, 'akshat54': 1, 'PimpleSushank': 1, 'RM_Rajdeep007': 1, 'Prakash27061996': 1, 'haidar2930': 1, 'Harsh_Sudan': 1, 'ishapatel999': 1, 'yash165_': 1, 'RaniTutun': 1, 'VILAS_KAVAR': 1, 'vorajanam4545': 1, 'aditya5070': 1, 'KM108SS': 1, 'shahvai1916': 1, 'harshalshinde11': 1, 'MamboDavid7': 1, 'vallabh_rajesh': 1, 'chinmaypednekar': 1, 'Cr7Ronit': 1, 'Nisha53415744': 1, 'Simmisolanki23': 1, 'ArzanAzmi': 1, 'itsmokshbitch': 1, 'TrolledByDewey': 1, 'TacticalPear': 1, 'eythnwarnes05': 1, 'BeerlyHockey': 1, 'JackEichBUST': 1, 'UNBILLIEVABLE': 1, 'corey_lew3': 1, 'ThomasGrundner3': 1, 'NatoApril': 1, 'prideauxcindy17': 1, 'hamedabdulrah10': 1, 'chimeManifest12': 1, 'uuerawe': 1, '_19_keys_': 1, 'Pch0Pch': 1, 'CASHAPP_MONEY1_': 1, '_19key_s_': 1, 'CashApp_4U_2K': 1, 'brendah44318421': 1, 'DJEmzDnb': 1, 'absstephens99': 1, 'Bellaelisedolce': 1, 'emilytjoness': 1, 'Unstable_Lines': 1, 'houseartistman': 1, 'AflekUK': 1, 'manik_dnb': 1, 'TurrallC98': 1})

```

## Connected components of G



- We have plotted the degree distribution across the whole network as a histogram and below is the output of the histogram along with the degree rank plot



#### **4.2- Betweenness Centrality:**

The nodes which are at the center and connect to the other group of nodes have much higher betweenness centrality. We have shown in the output the betweenness centrality of all the nodes[9].

```
# compute centrality
centrality = nx.betweenness_centrality(H, k=10, endpoints=True)

# compute community structure
lpc = nx.community.label_propagation_communities(H)
community_index = {n: i for i, com in enumerate(lpc) for n in com}

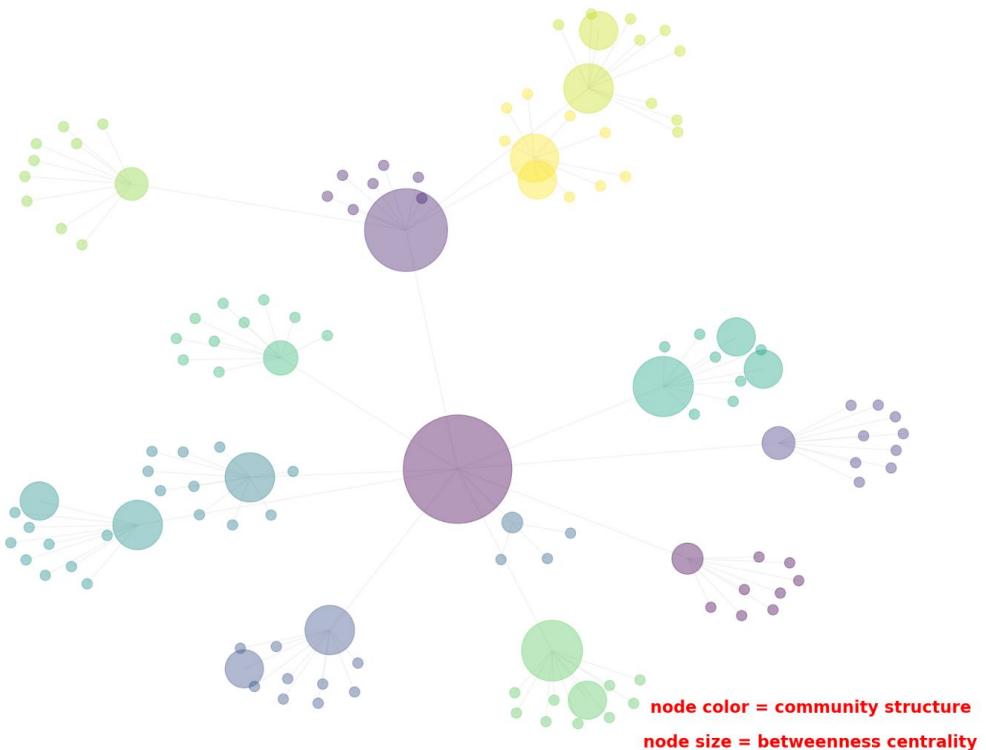
##### draw graph #####
fig, ax = plt.subplots(figsize=(20, 15))
pos = nx.spring_layout(H, k=0.15, seed=4572321)
node_color = [community_index[n] for n in H]
node_size = [v * 20000 for v in centrality.values()]
nx.draw_networkx(
    H,
    pos=pos,
    with_labels=False,
    node_color=node_color,
    node_size=node_size,
    edge_color="gainsboro",
    alpha=0.4,
)
```

```

centrality
✓ 0.0s
Output exceeds the size limit. Open the full output data in a text editor
{'ps_pujansheth': 0.8629921259842519,
'irvingjoanne17': 0.49921259842519683,
'Pari1602': 0.07874015748031496,
'mbbarot69': 0.1779527559055118,
'kush0099': 0.1283464566929134,
'payal_pabari': 0.08661417322834646,
'MrDavid1901': 0.08661417322834646,
'JubinYadav1': 0.1708661417322834646,
'JPL_India': 0.269291385826772,
'RashiJoshi19': 0.1779527559055118,
'PenhaAllen': 0.16377952755905512,
'michaelvbuffalo': 0.07874015748031496,
'Wright_habits': 0.269291385826772,
'HookedSounds': 0.07874015748031496,
'JeD87260622': 0.007874015748031496,
'LhordeRich1': 0.007874015748031496,
'Hasa007788': 0.10708661417322834,
'FnSbyruqayyah': 0.007874015748031496,
'maxine_dnwunder': 0.007874015748031496,
'SapetaJes': 0.007874015748031496,
'WIZYCHRISMUS13': 0.007874015748031496,
'SwatirKSaha': 0.007874015748031496,
'BhattKesha2806': 0.007874015748031496,
'MarooMansi': 0.007874015748031496,
'siddharth_sonak': 0.007874015748031496,
}

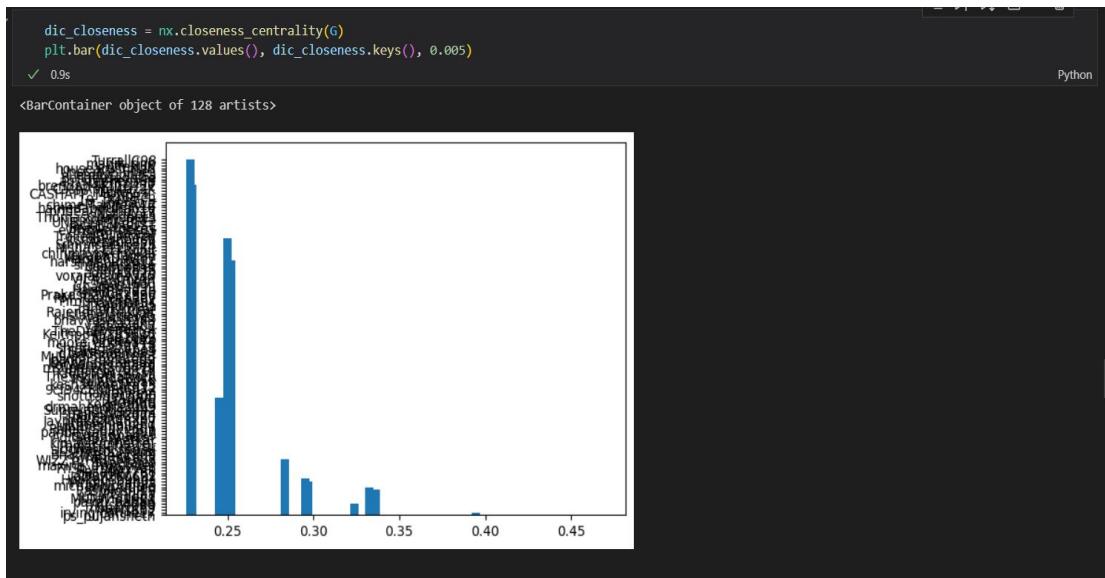
```

Gene functional association network (*C. elegans*)



#### **4.3- Closeness Centrality Distribution:**

The closeness centrality is the measure of centrality in the graph which provides the average distance from each node to all other nodes. We have plotted the closeness centrality distribution in the histogram below.



#### **Section V: Teams' Contribution:**

Tasks	Name
Data Sourcing and Authentication	Pujan
Data Storing and Processing	Akash
Network Measures Calculation	Pujan
Graph Visualization	Akash, Pujan
Observations and Research	Akash
Project Report Writing	Akash, Pujan

## **Section VI: REFERENCE:**

- 1) <https://docs.python.org/3/library/>
- 2) <https://networkx.org/>
- 3) <https://stackoverflow.com/questions/73994570/how-to-get-user-profile-image-url-with-tweepy-twitter-api-v2>
- 4) <https://matplotlib.org/stable-devel/contributing.html>
- 5) [https://www.researchgate.net/figure/Degree-distribution-of-the-Twitter-friendship-graph-Main-plot-Degree-distribution\\_fig1\\_268154614](https://www.researchgate.net/figure/Degree-distribution-of-the-Twitter-friendship-graph-Main-plot-Degree-distribution_fig1_268154614)
- 6) [https://www.tutorialspoint.com/python/python\\_nodes.htm](https://www.tutorialspoint.com/python/python_nodes.htm)
- 7) <https://stackoverflow.com/questions/49722640/how-to-create-edges-for-nodes>
- 8) <https://stackoverflow.com/questions/53958700/plotting-the-degree-distribution-of-a-graph-using-nx-degree-histogram>
- 9) <https://www.geeksforgeeks.org/betweenness-centrality-centrality-measure/>
- 10) [https://www.youtube.com/watch?v=0EekpQBEP\\_8](https://www.youtube.com/watch?v=0EekpQBEP_8)
- 11) <https://gephi.org/plugins/#/>