ONLINE SOCIAL NETWORK ANALYSIS

PROJECT -1 REPORT

Aditya Shivakumar- A20513527

Sesha Shai Datta Kolli- A20516330

**INTRODUCTION:**

The main objective of this project is to learn the process of data crawling and perform some analysis on the extracted data. The technique of continuously gathering data from the internet using software tools is known as data crawling, often referred to as web web scraping. Data crawling has evolved into a powerful instrument for researchers and businesses to extract valuable observations and patterns from multiple web sources because of the internet's fast expansion and the massive amount of information available online. Data crawling allows us to gather and analyze enormous amounts of information from a plethora of sources, including news sources, e-commerce websites, social media platforms, and government databases quickly and efficiently.

The project outlines: Data Collection, Data Visualization and Network measure calculations are explained in detail in the below sections.

**DATA COLLECTION:**

We were given the objective of choosing a social media platform such a Twitter, Facebook, Instagram, dblp, Reddit, arXiv, ResearchGate, Stackoverflow, Stackexchange and etc for data crawling. For crawling the data we have chosen Twitter.

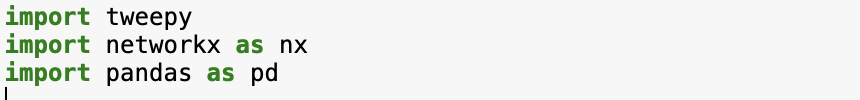
Twitter is an online social networking and news platform wherein members exchange concise messages referred to as tweets. Tweeting is the practice of broadcasting brief messages to your Twitter followers in the anticipation that they would find these useful and engaging. Twitter and tweeting could also be referred to as microblogging. Twitter is well-known for a wide range of reasons. One of the key reasons is its capacity to deliver breaking news and real-time updates on a variety of subjects. Journalists, entertainers, and politicians now frequently use Twitter to interact with their followers and the general public.

Using the Twitter API is simplified with the help and support of the Python library Tweepy .The Twitter API enables programmatic access to Twitter information including tweets, followers, and user profiles as well as a variety of operations like writing tweets and following users. By offering a simple-to-use interface for authentication, querying, and altering Twitter data, Tweepy makes it easier to engage with the Twitter API. It offers helpful features like rate restriction and pagination support and handles different Twitter API endpoints. Developers and data scientists that wish to gather and examine Twitter data now frequently use Tweepy.

The Twitter API endpoints and models are implemented by a set of class and functions in Tweepy, which effectively controls various execution specifics like:

* Data encoding and decoding
* HTTP requests
* Results pagination
* OAuth authentication
* Rate limits
* Streams

The first step of the code is to import tweepy. We have used the latest version of tweepy 4.12.1. Other libraries as such as network and pandas are imported. NetworkX is a set of Python-based tools for building, modifying, and researching the composition of large, complicated networks that are represented as graphs with nodes and edges are studied using this technique.

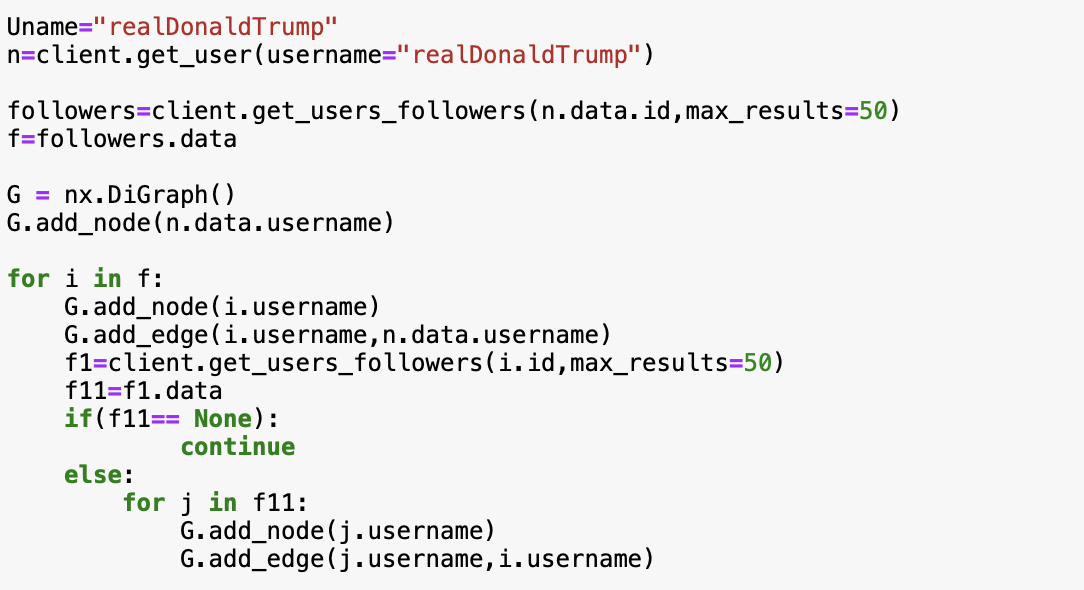


The next step is to initialize tweepy.client(). Tweepy.client provides access to the twitter API using the given credentials. The bearer token is obtained via the developer portal under the keys and tokens tab.

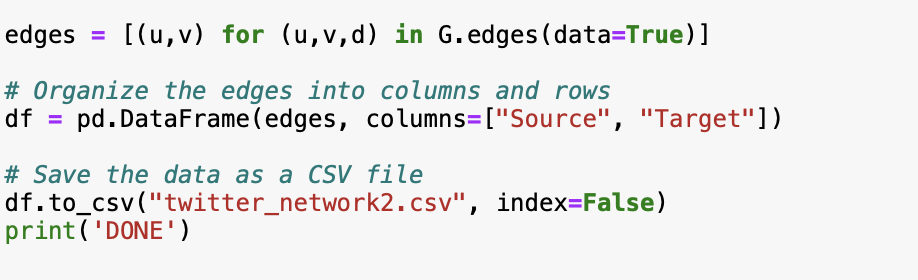


A username with many followers is to be chosen. We have chosen the Username as “Donald Trump”. Client.user() is then used to gather information about the user, such as the user id, username etc.

Next, client.get\_users\_followers(n.data.id,max\_results=50) is used to get the data about followers of the user. In this case, 50 random follower’s username and id is retrieved. The number of results to be obtained is flexible and can be set based on the objectives and preferences. The next step is to run a loop that obtains the followers of each of the 50 followers of the initially chosen username .The number of results to retrieve is again set to be 50.



In the end the data that has been scrapped is saved as a csv file using the pandas library.. The columns are set as source and target. Once the edges are organised into rows and columns it is saved as a csv file.

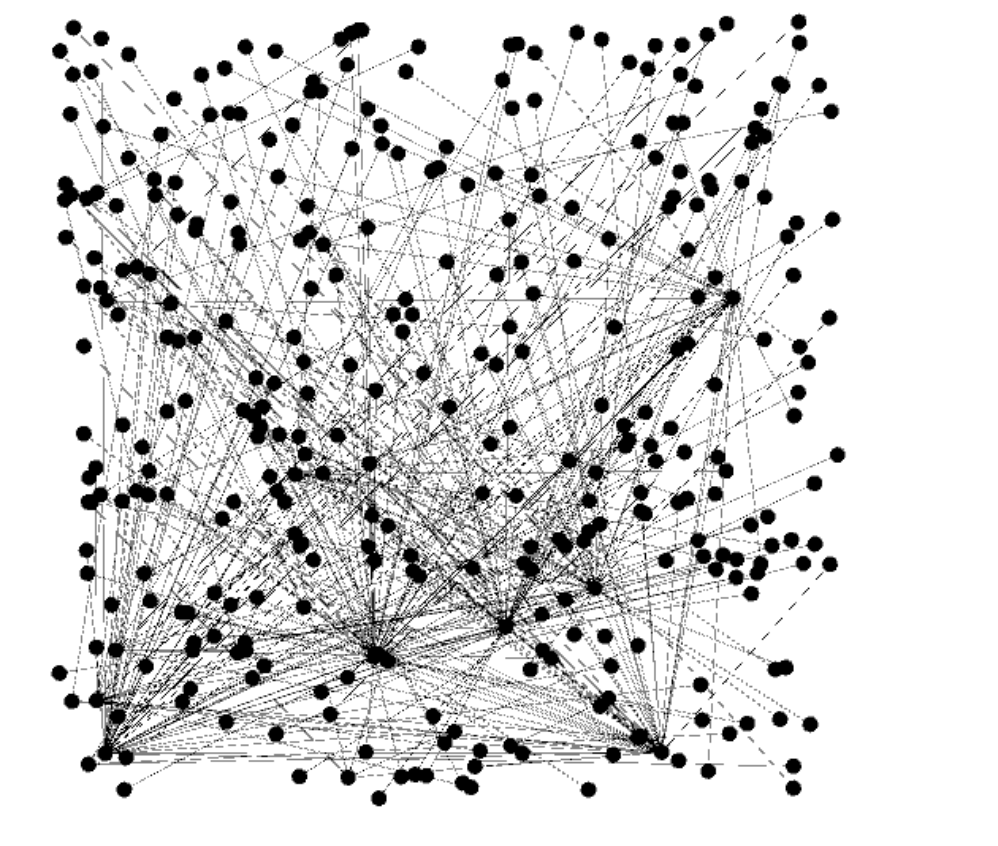


Given below is an example of how our scrapped data looks when it exported as a csv file.The final number of nodes retrieved is 344 nodes.

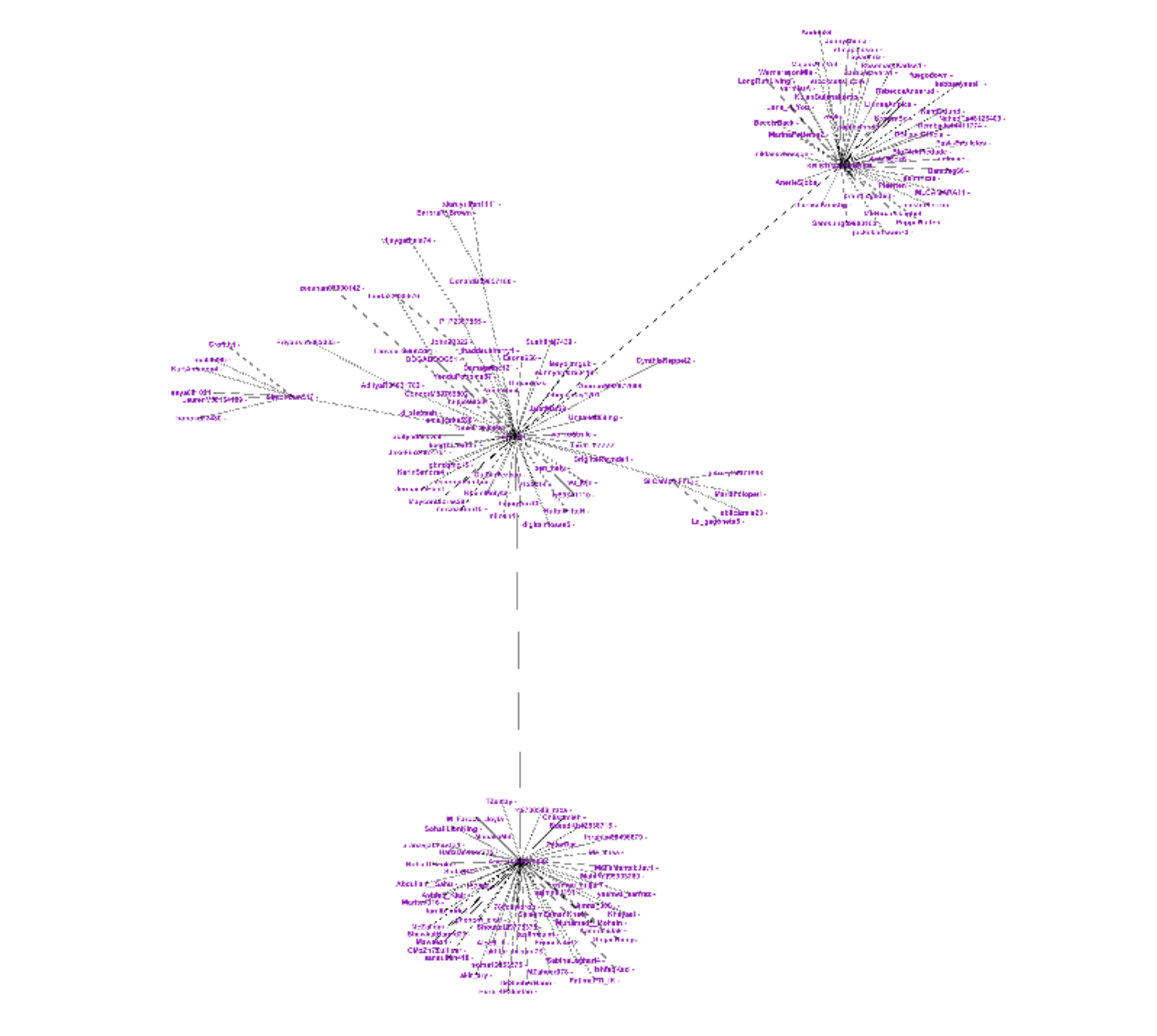
**Scraped Data in CSV format:**



**Visualization of the network:**

****

Visualization of the friendship network using gephi

****

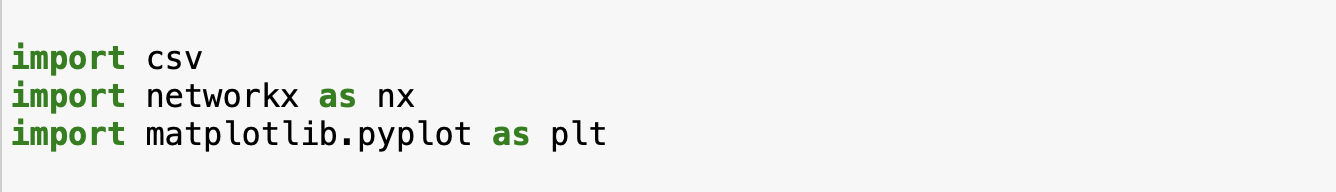
Visualization of the friendship network using gephi ( Force Atlas)

**Network Measure Calculation:**

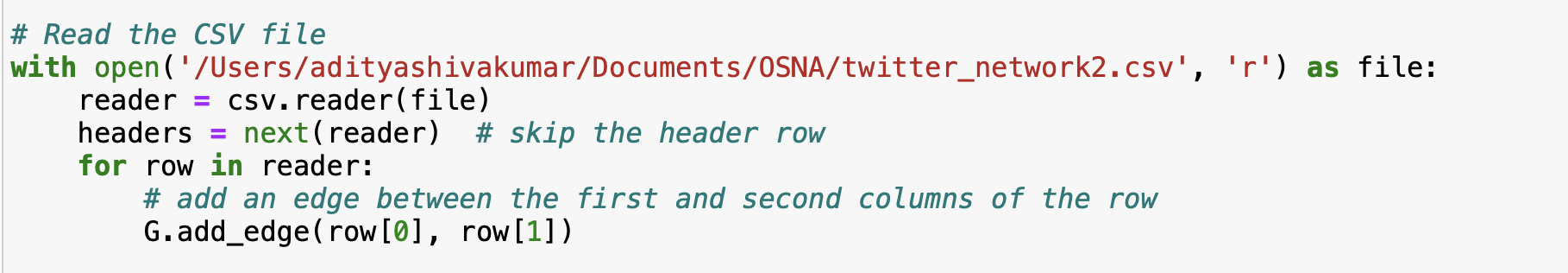
For calculating the network measures, we were asked to plot a histogram of degree distribution and any other two measures.

The degree of a node in a network is decided by the connections it has with other nodes, and the degree distribution is the probability distribution of these degrees across the entire network. For plotting degree distribution we have written the following lines of code:

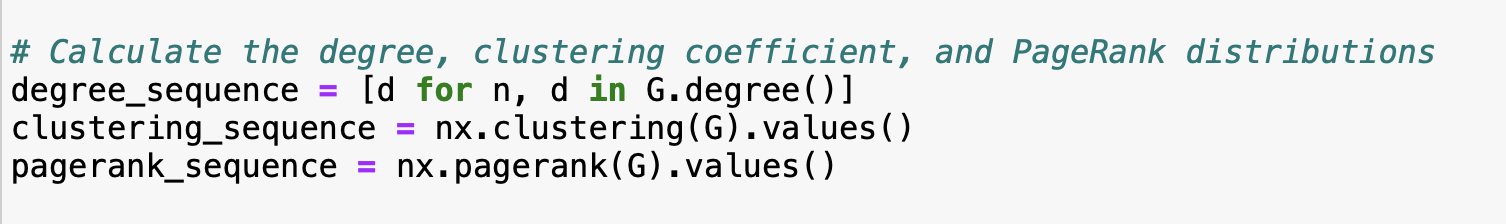
Libraries like csv, netowrkx and matplotlib are imported to calculate ands visualize the network measures.



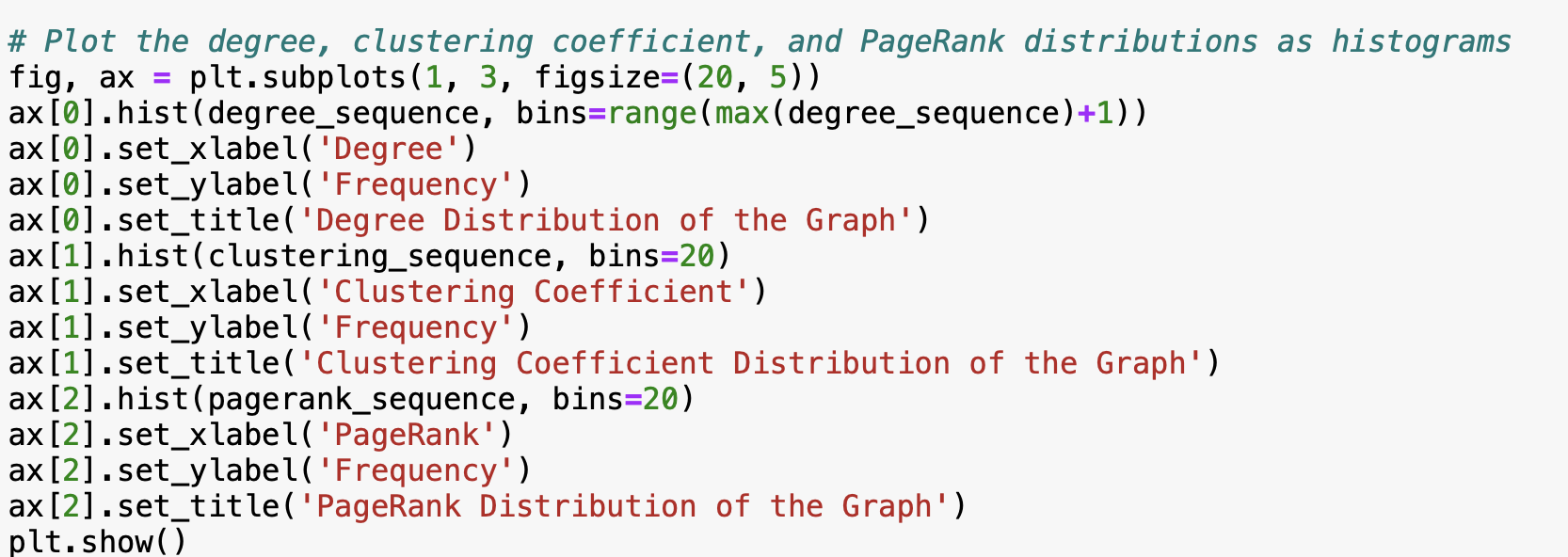
The first step is to read the csv file(scrapped data) which is stored in the computer. To open the file, with open() command is used with the path of the file specified. After the file is opened, csv.reader is used to read the csv file. The header row is skipped and an edge is being created between the columns (first and second).



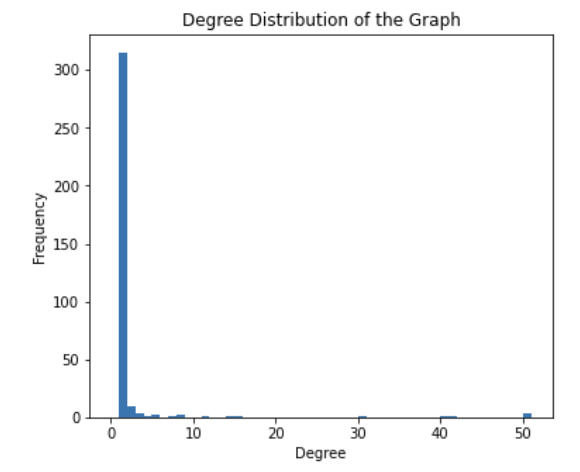
The degree distribution is calculated. In addition to it, clustering coefficient and Pagerank are also calculated. The degree to which nodes in a graph are likely to cluster together is defined by the clustering coefficient. Pagerank also calculates the importance of each node in the network.



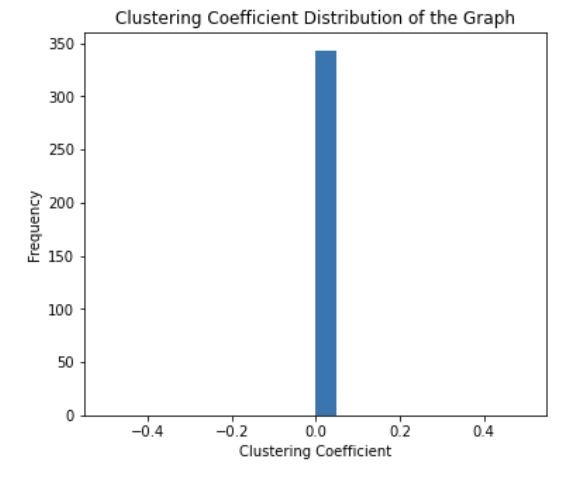
After calculating the above mentioned network measures, the next step is to plot the histogram plt.subplot() is used to define the number of plots to be visualized. The xlabels, y- labels and the titles are specified for each of the network measure. Finally plt.show() is used to display the graphs.



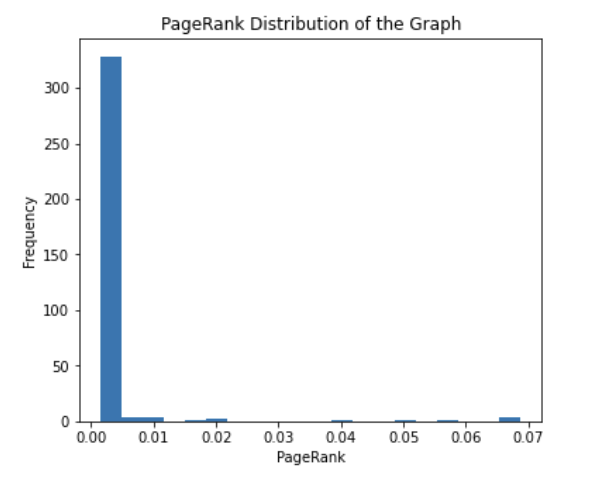
**Degree Distribution:**



**Clustering Coefficient:**



**PageRank Distribution:**



**Conclusion:**

The data from twitter was collected using the tweepy library. A total of 344 nodes were obtained and the data was downloaded as a csv file and the friendship network was visualized using Gephi. The network measures such as Degree Distribution, Pagerank Distribution and Clustering coefficients were calculated and the respective histograms were plotted.​​ The three main objectives of this project was completed and satisfied. The desired results were also obtained.

**References:**

1. https://docs.tweepy.org/en/stable/api.html
2. https://pandas.pydata.org/docs/
3. https://networkx.org/
4. https://matplotlib.org/
5. https://www.freecodecamp.org/news/python-web-scraping-tutorial/
6. https://www.geeksforgeeks.org/extraction-of-tweets-using-tweepy/