CS-579 ONLINE SOCIAL NETWORK ANALYSIS  
Twitter Friendship Network Project Report

# Introduction

In this project we aim to analyze a social friendship network for which the data is crawled from a popular social media platform twitter. We have visualized the crawled data into a directed graph with the users being nodes and the edges between them representing either following or being followed by another user.  
  
With ever so increasing popularity and demand in social media platforms, the analysis of these networks can provide insights into user behavior and friendship structures.

In the data collection part of this report, we describe how we have crawled the raw data of usernames of friends and followers of a user using tweepy library and twitter API. In the data visualization part of this report, we describe how we have used the crawled data to visualize the friendship network into a directed graph using the networkx package. In the network measure section of the report, we describe how we used the graph data to form a degree distribution, clustering coefficient and pagerank.

Data Collection

In this section of the report, we will go through how we have collected the data from twitter.

We have registered as a twitter developer to get access to API keys. These API keys lets us have access to the vast amount of twitter data to collect and process. To build a friendship network of a user the main challenge would be that the data could be enormous and could exceed the specification of 100 – 500 node constraint or could also be sparse. So, we have decided to crawl the data by limiting the followers and friends to a maximum of 5. By doing so we were able to be in the limit. We have in total collected 142 of a specific users friend/following information. We used tweepy python package to retrieve usernames of the mentioned nodes. We have collected all the information and stored it into a pandas data frame while crawling the data and then moved it to a CSV file.



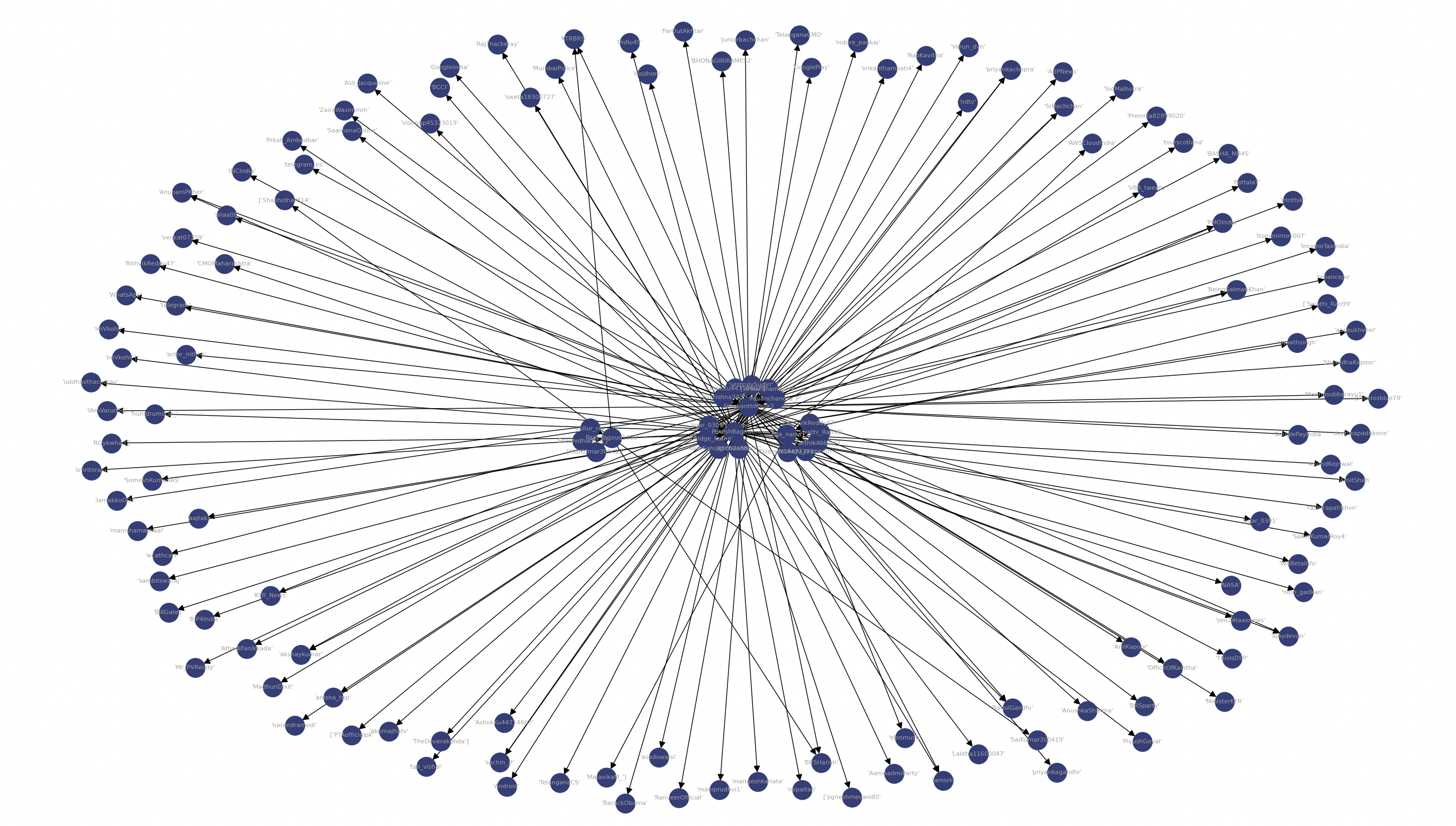
### Data Visualization

In this section of the report, we will describe how we have visualized the data into a directed graph.

We first initialize screennames as source nodes and then followers and friends as leaf nodes.

Now we construct the graph with usernames as source node and friends/followers as leaf nodes. We add edges according to the friendship network between them. We use networkx package for generating the directed graph to add nodes and connect them with edges.

In this graph we have 142 number of nodes and 155 number of edges.



### Network Measures

In this, we calculate the network measures based on the final graph.

1. Degree distribution

Degree distribution is a count of number of edges from each node and gives the result in the form of histogram.

Chart, histogram

Description automatically generated

1. Clustering coefficient

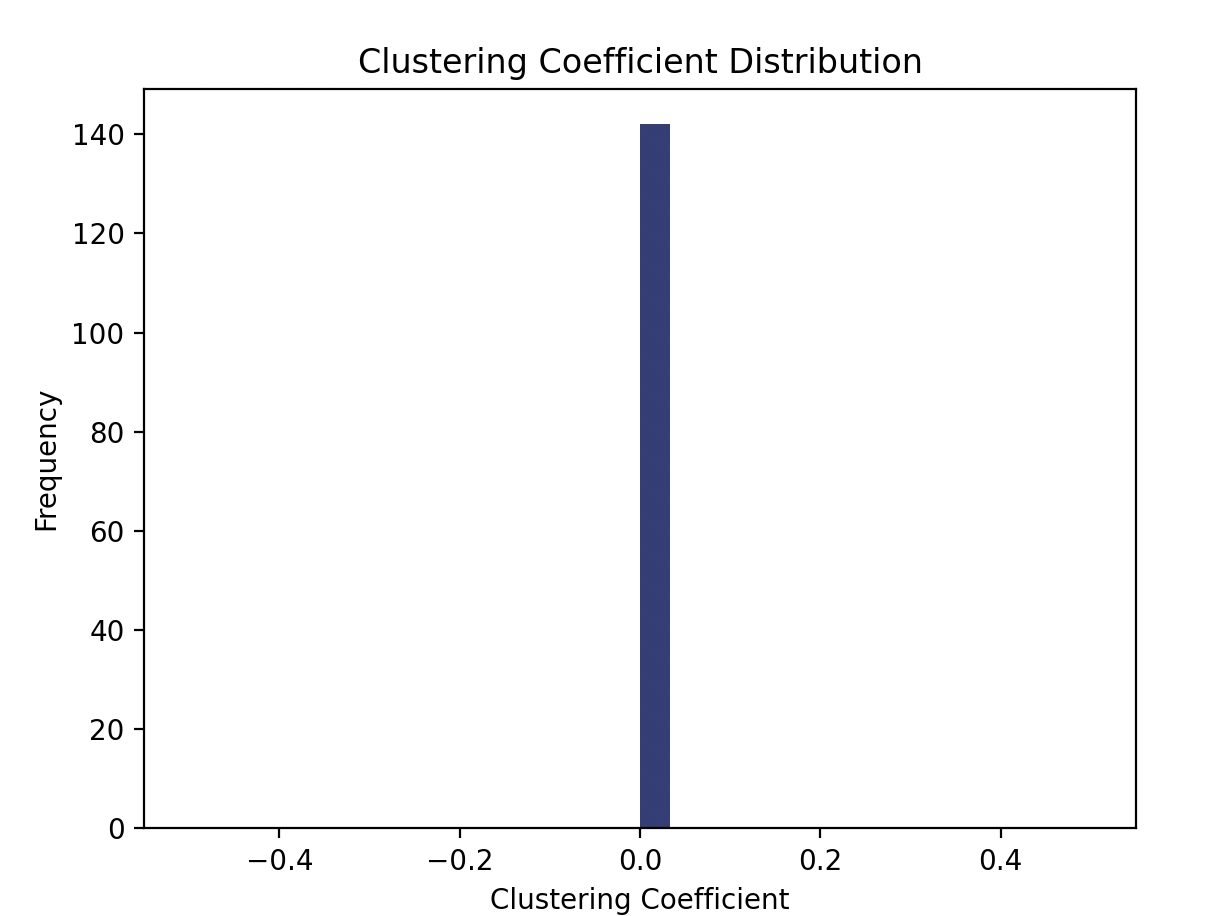
Clustering coefficient is a likelihood of how the neighbors of its node are being connected to each other. If the clustering coefficient is higher, it indicates that the nodes are tightly bonded together whereas a low value indicates the nodes are sparsely connected to each other. It is also used to calculate the overall clustering level in a network.

The clustering coefficient is calculated as:

Ci = (2 \* T..i)/(k..i \* (k..i -1)). Where 0 < Ci < 1

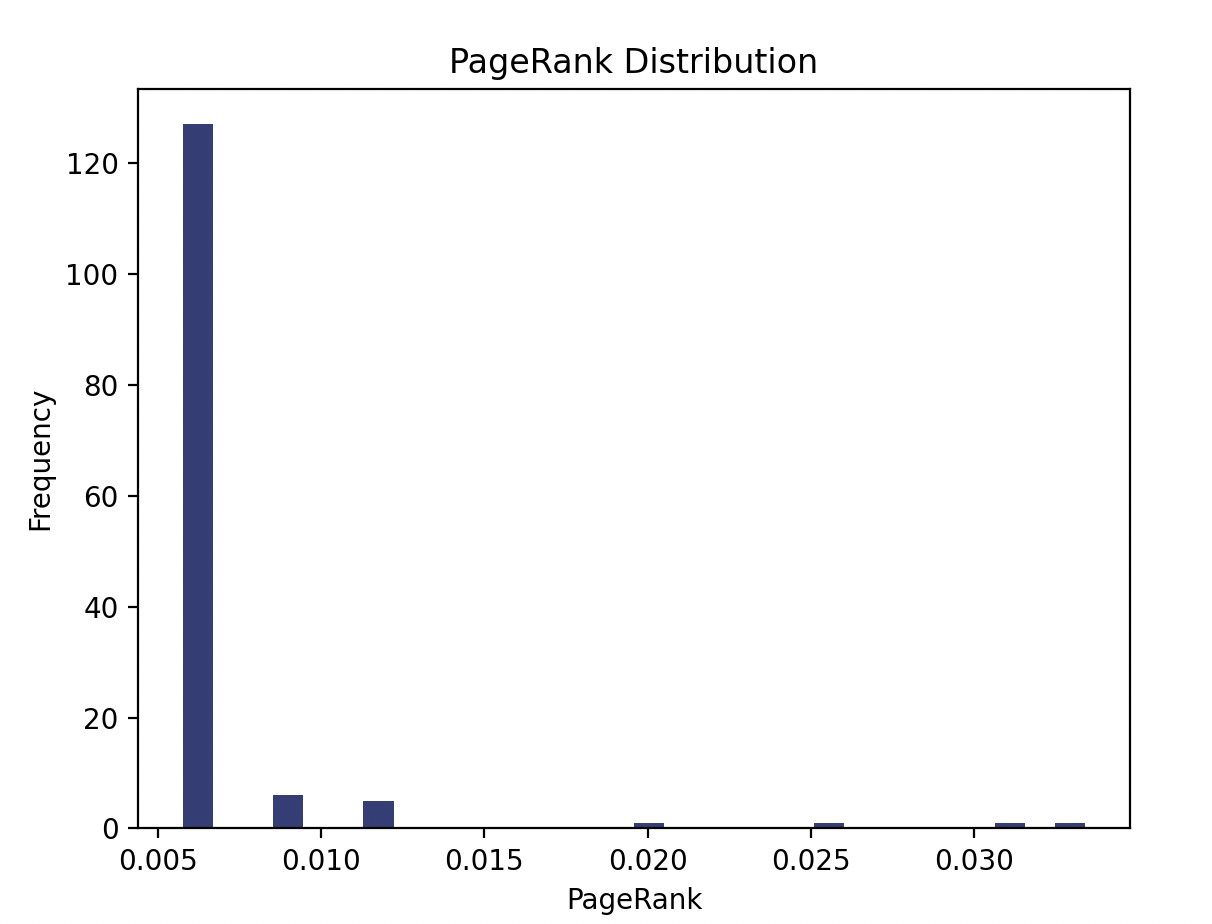
Ci =1 means the nodes are fully connected.

Ci = 0 means none of the nodes are connected to each other.



1. Pagerank Distribution

It is a measure of importance of nodes in a network. The usage of pagerank distribution is vividly seen in most influential nodes in a network. The pagerank algorithm works by assigning a numerical number to every page based on number of links pointed to each node.



1. Closeness centrality

Closeness centrality is a measure of closeness or importance in a network based on the average distance between the nodes.

It is calculated as follows:

C(i) = (n - 1) / Σd(i, j)

Closeness centrality is used in social network analysis to identify the well-known individuals who have the highest connections with the other individuals. By this we can also be able to know the key locations for infrastructure and many other.

Chart, histogram

Description automatically generated

1. Betweenness centrality

It is a measure of shortest paths between the nodes.

Mathematically, the betweenness centrality of a node is derived from the below equation:

C(i) = Σ(s, t) ∈ σ(i) (n(s,t,i) / n(s,t))

Betweenness centrality is commonly used in social network analysis to identify individuals who act as brokers or gatekeepers in the network, connecting otherwise disconnected groups.

Graphical user interface

Description automatically generated with medium confidence

## References

1. Tweepy - https://docs.tweepy.org/en/stable/getting\_started.html#introduction
2. Networkx - https://networkx.org/documentation/stable/reference/algorithms/centrality.html
3. Matplotlib - https://matplotlib.org/stable/index.html
4. Pandas - https://pandas.pydata.org/docs/
5. https://towardsdatascience.com/customizing-networkx-graphs-f80b4e69bedf

## Team Members

Sri Sai Teja Narra - A20514763

Swathi Puskoori - A20513538