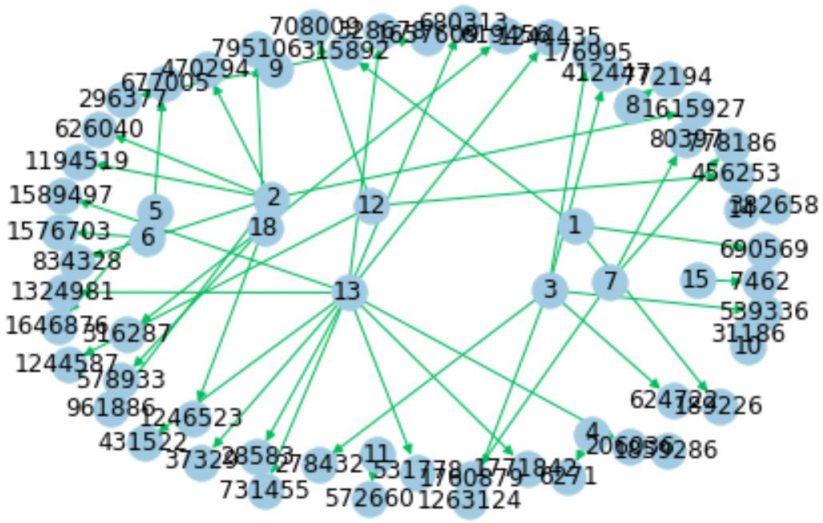
The problem definition demands prediction of friends with the list of mutual friends. This can be classified as a **supervised learning classification problem.** We need sound understanding of the networkx library to tackle this problem.

The first step would be data reading and EDA. We start by creating nodes, and join them by edges. The resultant graph would look like:



The next step is to identify unique profiles and followers. We can see the number of followers for each person, and from basic EDA, it is clear that 99% of all the data have only 40 followers. Further on, inspecting the number of people each person is following. Further on we can inspect the number of people each person is following, and the minimum and the maximum number of followers/ following.

We then generate the missing edges in the graph. Further on, we split the training and test data.

The next step is to find the ***Jaccard distance*** and ***cosine distance*** for the given dataset.

Jaccard distance, $j = \frac{X \cap Y}{X \cup Y}$ Cosine distance, $c = \frac{|X \cap Y|}{|X|.|Y|}$

Jaccard distance and cosine distance assist us to understand how likely two nodes are to be connected.

We further work on PageRank, which computes a ranking of the nodes in the graph based on the structure of the incoming links.

Further on, we calculate additional features, such as the shortest path between the two nodes, shared community of nodes, adar index, is the person following back, Katz centrality for nodes, and HITS score.

Next, the train and test data are taken and additional features are introduced , such as jaccard_followers, jaccard_followees, etc…

The final step is building a prediction model and improve upon its accuracy. The final accuracy obtained usin g XGBoost for training dataset is `0.99`, while for testing dataset, the accuracy is `0.91`.