# Summary:

**1.Displaying a sub-graph:**

- Create a new file-train.csv without headers.(nx.info()) gives certain information about the graph.
  Type: DiGraph
  Number of nodes: 1862220
  Number of edges: 9437519
  Average in degree(average number of edges coming into a node): 5.0679
  Average out degree(average number of edges going out to a node): 5.0679
- Plot a sub-graph selecting only top 20 rows from train.csv file

**2.Follower Stats:**

- Number of followers can be valued using in-degrees.Hence calculating in-degrees gives follower count of every user.
- We try plotting follower count of each people.
- we use box plot to check for outliers.
- to have a better picture we plot percentile values and we understand that.
  - 99% of people having 40 or fewer followers.

**3.Following Stats:**

- Number Of people each person is following can be valued using out-degrees.Hence calculating out-degrees gives each people following count.
- We try plotting following count of each user.
- we use box plot to check for outliers.
- to have a better picture we plot percentile values and we understand that.
  - 99% of people having 40 or fewer followers.

**4.Follower+Following Stats:**

- Now we take both follower and following count(in_out count) of each people.
- We try plotting in_out count of each user.
- we use box plot to check for outliers.
- to have a better picture we plot percentile values and we understand that.
  - 99% of people having 40 or fewer followers.

**5.Binary Classification Task:**

- The data we have consists only connected edges,hence while performing classification task we will have only label 1.
- It is necessary that we have 0 labelled data i.e not connected edges/links having shortest path greater than 2.

**6.Data Split:**

- We split obtained data into 80:20
- positive links and negative links are split seperatly because we need positive training data only for creating graph and for feature generation

**7.Featurizations:**

- **Jaccard Distance:**
  - Lets say u1,u2,u3,u4,u5,u6 are connected,such that u1 followers-{u3,u4,u5} and u2 followers-{u3,u4,u6}.
  - We could find how disimilar these two sets are using Jaccard distance.
  - The Jaccard similarity index (sometimes called the Jaccard similarity coefficient) compares members for two sets to see which members are shared and which are distinct. It's a measure of similarity for the two sets of data, with a range from 0% to 100%. The higher the percentage, the more similar the two populations. Although it's easy to interpret, it is extremely sensitive to small samples sizes and may give erroneous results, especially with very small samples or data sets with missing observations.

$$j = \frac{|X \cap Y|}{|X \cup Y|}$$

- **Cosine distance(Otsuka-Ochiai coefficient):**
  - Like Jaccard distance we could also calculate another distance which is an extension to cosine similarity.

$$CosineDistance = \frac{|X \cap Y|}{SQRT(|X| \cdot |Y|)}$$

- **Page Rank:**
  - Named after Larry page.
  - Imagine whole internet as directed graph where each web page is node.
  - Each page is given a rank depending on number and quality/importance of links/edges.
  - It gives you a probability value for each of the web page that represents the likelihood of a random person clicking on a page to arrive at another page. <img width = "350" src= 'https://upload.wikimedia.org/wikipedia/en/8/8b/PageRanks-Example.jpg'/>

- **Shortest Path:**
  - Getting Shortest path between two nodes, if nodes have an edge i.e, trivially connected then we are removing that edge and calculating the shortest path.
  - else assign -1

- **Connected Component:**
  - A digraph is strongly connected if every vertex is reachable from every other following the directions of the arcs. I.e., for every pair of distinct vertices u and v there exists a directed path from u to v.
  - A digraph is weakly connected if when considering it as an undirected graph it is connected. I.e., for every pair of distinct vertices u and v there exists an undirected path (potentially running opposite the direction on an edge) from u to v.
  - Here (1)is strongle connected component (2) is weakly connected component <img width = "350" src = "https://i.stack.imgur.com/iJffU.png" />

- **Adamic/Adar Index:**
  - Introduced by Lada Adamic and Eytan Adar to predict links in a social network.
  - Adamic/Adar measures is defined as inverted sum of degrees of common neighbours for given two vertices.

$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{log(|N(u)|)}$$

  - As u is high (high neighbours),then there is less probability that N(x) and N(y) being connected are less
  - As u is less(less neighbours), then there is more probability that N(x) and N(y) being connected are more.

- **Kartz centrality:**
  - introduced by Leo Katz in 1953 and is used to measure the relative degree of influence of an actor (or node) within a social network.
  - Katz centrality similarly like page rank that measures influence by taking into account the total number of walks between a pair of actors

- **HITS algorithm:**
  - Hyperlink-Induced Topic Search (HITS; also known as hubs and authorities) is a link analysis algorithm that rates Web pages, developed by Jon Kleinberg.
  - HITS algorithm gives you two scores;Authority and Hub score.Authorities estimates the node value based on the incoming links. Hubs estimates the node value based on outgoing links.
  - At the start both the scores are 1.We gradually update both authority and hub scores.We then normalize both scores so the it does becomes infitely large.

- **SVD features:**
  - We compute adjacency matrix
  - We then decompose it using svd of components 6.which results in two matrices left singular and right singular matrix.
  - Now both the matrices could be used as features containing vector of 6-dimensional

- **Weight feature:**
  - Taken from a research paper - Graph-based Features for Supervised Link Prediction by William Cukierski, Benjamin Hamner, Bo Yang
  - Intuitively, consider one million people following a celebrity on a social network then chances are most of them

never met each other or the celebrity. On the other hand, if a user has 30 contacts in his/her social network, the chances are higher that many of them know each other.
- In order to determine the similarity of nodes, an edge weight value was calculated between nodes. Edge weight decreases as the neighbor count goes up.
- Since the graph is directed,weighted in and weighted out are differenlty calculated.

$$W = \frac{1}{\sqrt{1+|X|}}$$

**8.Data Preparation:**

**we will create these each of these features for both train and test data points**

1. jaccard_followers
2. jaccard_followees
3. cosine_followers
4. cosine_followees
5. num_followers_s
6. num_followees_s
7. num_followers_d
8. num_followees_d
9. inter_followers
10. inter_followees
11. Page Ranking of source
12. Page Ranking of dest
13. shortest path between source and destination
14. belongs to same weakly connect components
15. is following back
16. adar index
17. katz of source
18. katz of dest
19. hubs of source
20. hubs of dest
21. authorities_s of source
22. authorities_s of dest
23. SVD features for both source and destination
24. Weight Features
    - weight of incoming edges
    - weight of outgoing edges
    - weight of incoming edges + weight of outgoing edges
    - weight of incoming edges * weight of outgoing edges
    - 2*weight of incoming edges + weight of outgoing edges
    - weight of incoming edges + 2*weight of outgoing edges

**9.Load data:**

- Load train and test data.
- Drop indicator_link.

**10.Models:**

- Trained using randomised radom forest.
- Output : Train f1 score 0.9652533106548414,Test f1 score 0.9241678239279553
- Plot confusion matrix
- plotted roc-auc curve