

Social Media Link Predictions using graphs

By :


Vishal Kumar (MT20305)

Shubham Sharma (MT20315)

Mohit Ghai (MT20308)

Group No. 3

Problem Statement

- The problem of predicting the existence of a link between two entities in a network.
 - The problem can be coined as: the task of determining the likelihood that any two nodes that are not connected at time $t = t_0$, will be connected at time $t = t_i$ ($t_i > t_0$).
 - The objective of link prediction is to identify pairs of nodes that will either form a link or not in the future.
 - Our work is to find the links between users that can happen in future.
- 

Dataset

We have used kaggle link prediction 2019 challenge dataset.

It has :

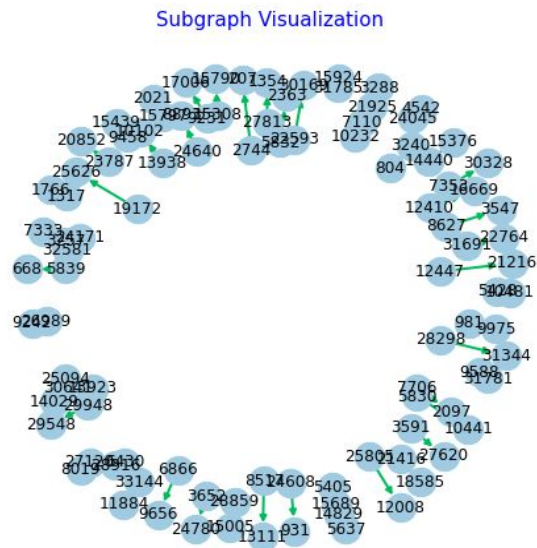
Source Node Id , Destination Node Id and direction variable.

A directed edge from node u to node v indicates that u follows v .



Data Visualization

As the graph is large. We have plotted only the subset of a graph i.e. shown below:



Analysis of Training Dataset

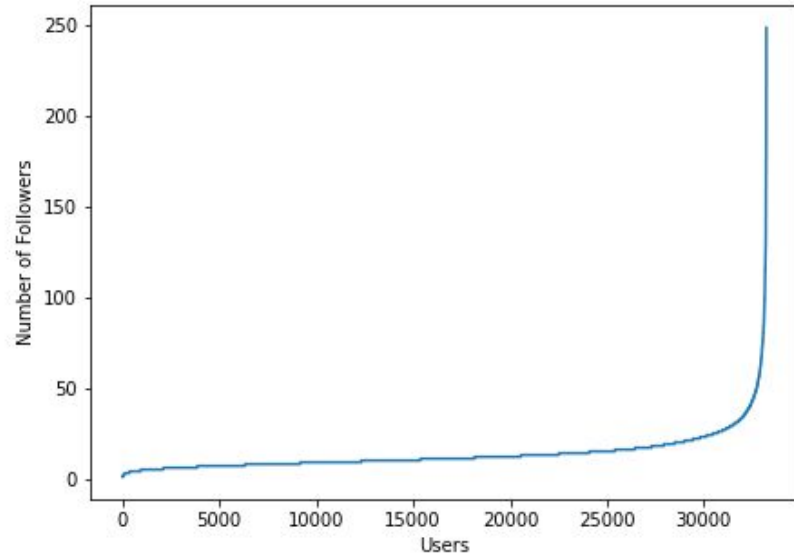
- It is a directed graph.
- It is a connected graph with single connected component.
- Number of Edges: 453797
- Number of Nodes: 33226
- Number of unique persons: 33226
- Number of people with no followers: 0
- Number of people with no following: 98 i.e. 0.29 % of overall.
- Number of people with no follower & followings: 0

Analysis

Percentile of Users & Their Followers	
Percentile of Users	Followers
90	23.0
91	24.0
92	25.0
93	26.0
94	28.0
95	30.0
96	33.0
97	37.0
98	45.0
99	60.0
100	248.0

90 % of users have 23 followers.

No. of Followers For Each Person



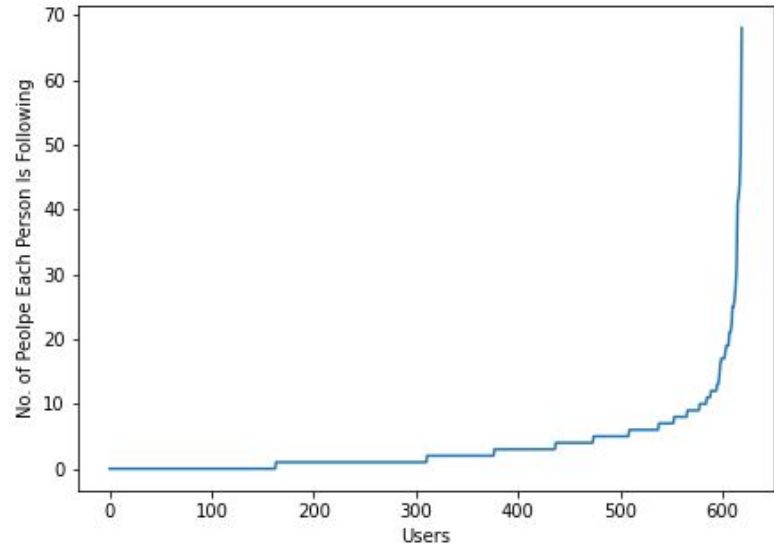
Many users have around 50 followers.

Analysis

Percentile of Users & Their Followings	
Percentile of Users	Followings
90	17.0
91	19.0
92	22.0
93	25.0
94	30.0
95	36.0
96	45.0
97	60.0
98	88.0
99	152.0
100	10226.0

90 % of users are following 17 other users.

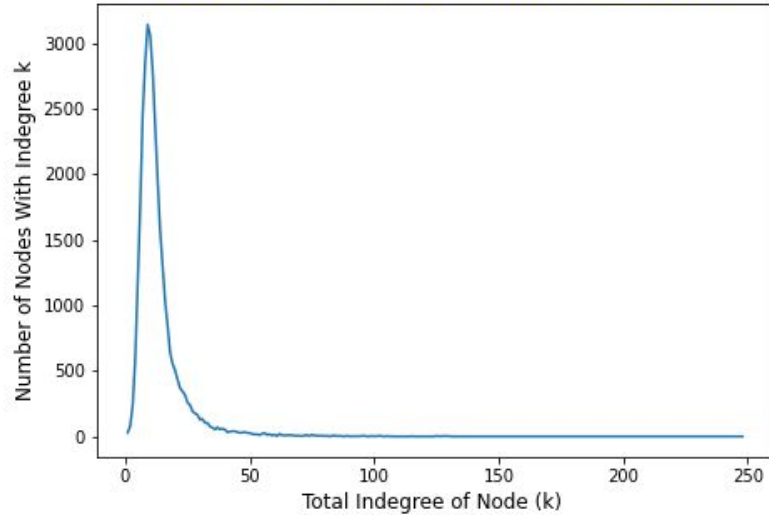
No. of People Each Person Is Following



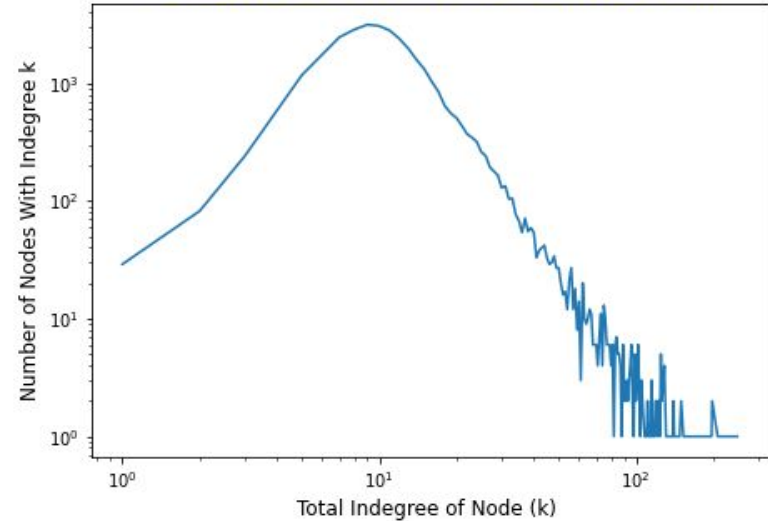
Most of users are following less number of persons.

Degree Distribution Curves

In-Degree Distribution of Network (Linear Scale)

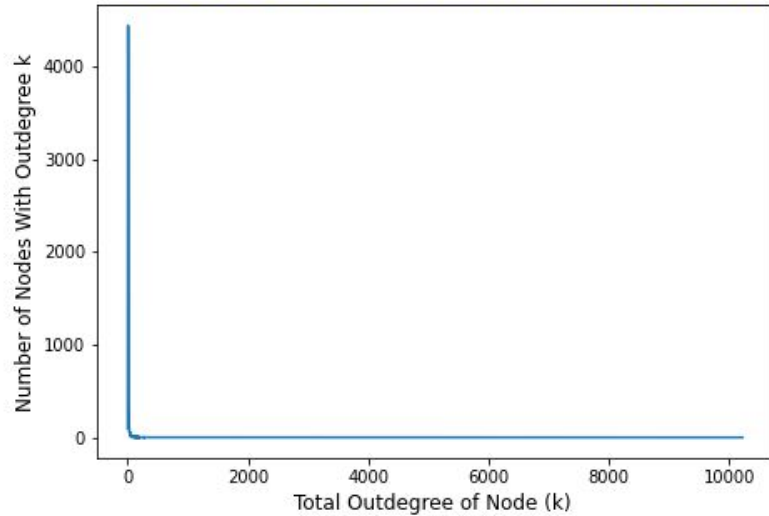


In-Degree Distribution of Network (Log Scale)



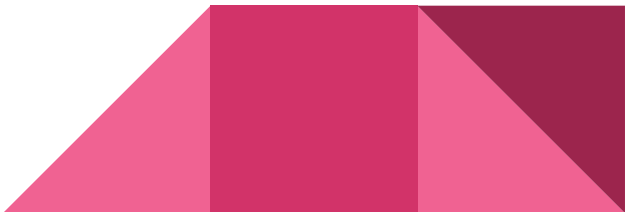
Degree Distribution Curves

Out-Degree Distribution of Network (Linear Scale)



Methodology

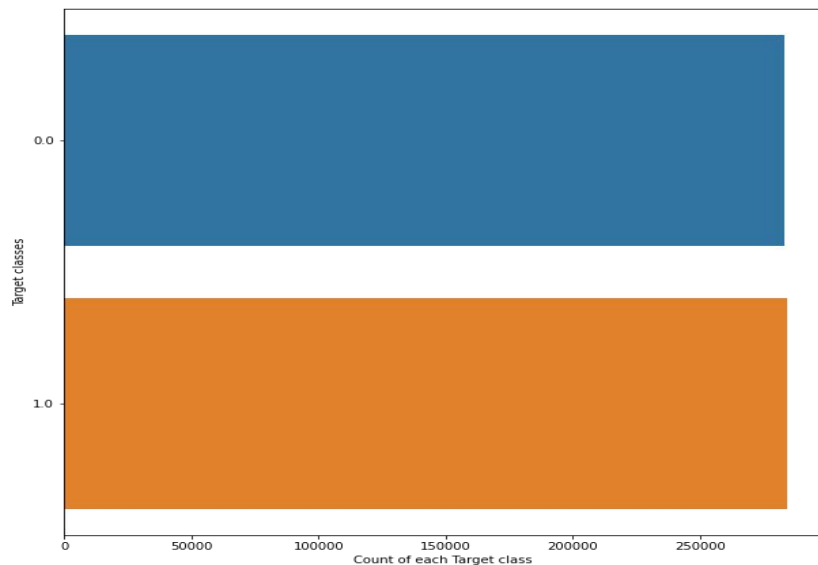
We have used the following heuristics to calculate the similarity scores between a pair of node.

- ❑ Jaccard Similarity
 - ❑ Cosine Similarity
 - ❑ PageRank
 - ❑ Adar Index
 - ❑ Preferential Attachment
 - ❑ HITS
 - ❑ Katz Centrality
- 

Dataset Preprocessing

	Source	Destination	jaccard_1	jaccard_2	cosine_1	cosine_2	Page_Rank0_s	Page_Rank0_d	shortest_path	adar	...	svdVFrom_3	svdVFrom_4	svdVFrom_5	svdVFrom_6	svdVTo_1	svdVTo_2	svdVTo_3	svdVTo_4	svdVTo_5	svdVTo_6
70639	30980.0	2503.0	0.034483	0.000000	0.111111	0.000000	0.000020	0.000027	-1.0	0.000000	...	-0.009251	0.000008	0.002838	-0.005958	-0.005080	-0.005023	0.005615	1.783801e-05	0.001107	-0.012009
308682	29415.0	5927.0	0.000000	0.013158	0.000000	0.034689	0.000018	0.000072	-1.0	2.715774	...	-0.009481	0.000009	0.002342	-0.005958	0.000072	-0.000298	-0.000095	6.290103e-07	-0.000388	-0.000080
60584	11972.0	10223.0	0.000000	0.000000	0.000000	0.000000	0.000019	0.000064	-1.0	0.000000	...	0.000005	0.000004	-0.000962	-0.000209	-0.003033	0.001627	0.000635	6.094223e-06	-0.004982	-0.001119
117396	320.0	19375.0	0.000000	0.000000	0.000000	0.000000	0.000046	0.000025	-1.0	0.000000	...	-0.003529	0.000018	-0.020183	-0.002809	-0.000089	0.000689	-0.000004	1.871903e-06	-0.000552	-0.000124
234008	1819.0	10200.0	0.003906	0.000000	0.017145	0.000000	0.000136	0.000022	-1.0	0.000000	...	-0.009688	0.000044	-0.080201	-0.012750	-0.003961	-0.003959	0.005499	1.915967e-05	0.002323	-0.011530

We have used all the heuristics mentioned in previous slide and generated a dataset with 45 features.



We have scaled the 'Class' Feature such that the number of 0 and 1's become equal.

Dataset Preprocessing

```
Relevant_Features Index(['Source', 'Destination', 'Class', 'cosine_1', 'cosine_2',  
                        'Page_Rank0_d', 'katz_s', 'katz_d', 'hubs_s', 'authorities_s',  
                        'pref_attachment_1', 'common_neighbour_1', 'svdUFrom_6', 'svdVFrom_2',  
                        'svdVFrom_5', 'svdVFrom_6', 'svdVTo_5'],  
                        dtype='object')
```

We have applied the Pearson Correlation function and generated 17 features that were highly correlated to the target 'Class' feature.

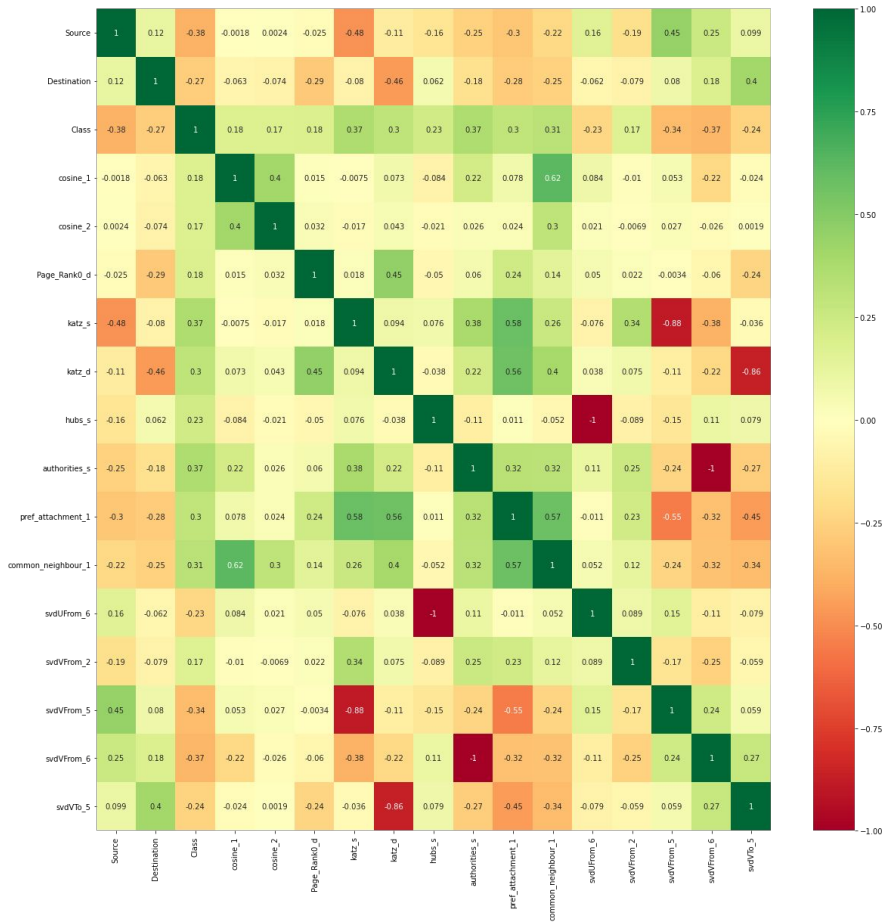
After this we have removed the 'Source' and 'Destination' features.

```
Train data: ((397072, 14), (397072,))  
Test data: ((170174, 14), (170174,))
```

Splitting the dataset in 70:30 such that 70% of the data is used for training the model and 30% of data is used for testing

Important Note: A system with high precision but low recall will return very few results, but most of its predicted labels are correct when compared to the training labels.

Dataset Preprocessing

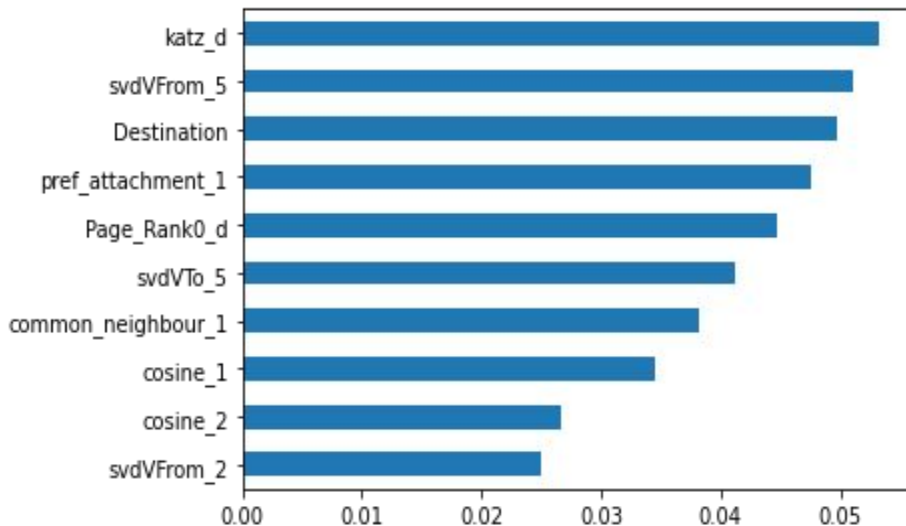


Correlation Matrix with Heat Map Visualization

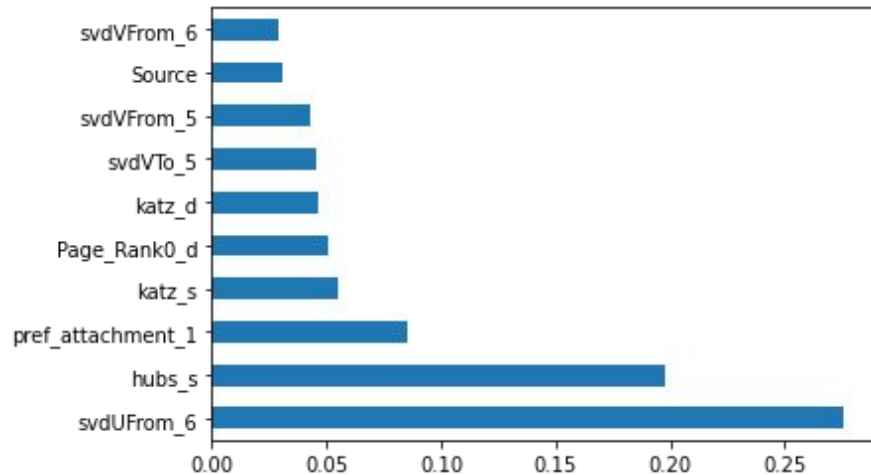
A correlation heatmap is a **graphical representation of a correlation matrix representing the correlation between different variables**. The value of correlation can take any value from -1 to 1. Correlation between two random variables or bivariate data does not necessarily imply a causal relationship.

The correlation matrix mentioned here is after the pearson correlation has been done and the features with low correlation to the target has been removed.

Dataset Preprocessing

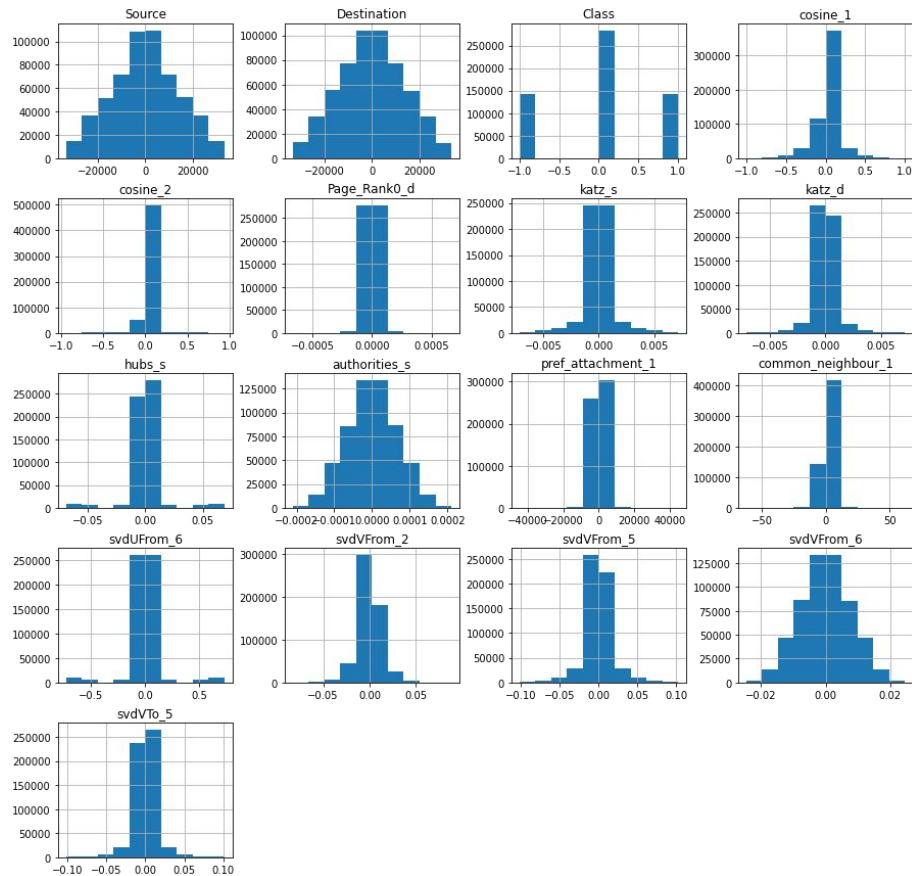


The above figure tells us about the features selected among the 14 features which are least correlated to the target variable. This has been found using Extra Tree Classifier.



The above figure tells us about the features selected among the 14 features which are highest correlated to the target variable. This has been found using Extra Tree Classifier.

Dataset Preprocessing



This figure shows the range of values of all the features in the preprocessed dataset.

Machine Learning Classifiers used

Based on above heuristics we obtained the feature vector. Used the following classifiers for prediction:

- Logistic Regression
- Decision Tree
- Random Forest
- XGBClassifier
- K Nearest Neighbors
- SVM



Results on Test Dataset

Classifier	Accuracy	Roc-Auc	Precision	Recall	F1 Score	MCC Score	Kappa Score
LR	82.86	82.86	92.84	71.19	80.59	0.68	0.66
KNN	88.81	88.80	90.14	87.14	88.61	0.78	0.78
DT	95.99	95.99	96.39	95.56	95.97	0.92	0.92
RF	97.21	97.21	97.48	96.92	97.20	0.94	0.94
XGB	91.33	91.33	90.86	91.88	91.37	0.83	0.83



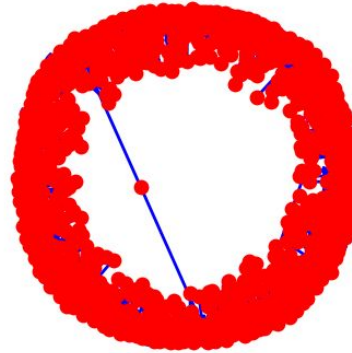
Python Visualization Application

E:/IIIT Delhi Coursework/NS/negative_edges_features.csv

Upload File

S No.	User name	Recommendation
1	beenthrownwillgoaway	nathan_1288
2	DandifiedDolphin	ObeseBlindDog
3	WillINeverAgainBeFat	Oirodoit
4	brian_topp	nevercriesalone45
5	ulovenecunter	dlock21
6	mankey101	Zombicide
7	binupdd	piero_cornejo_12
8	travelwithus04	AgentZeno
9	Cjbaccam	MultibandDynamics
10	carlosqmanjr	supersticiouswriting
11	DumbQuestionIdk	tigerman111
12	Bluemantan	ncmailperson
13	Zombicide	pefe
14	fossman83	ndelta
15	Napsonz	raymondsux
16	ScienceZest	qyvleader
17	shioox	SolarisNight
18	zhengdon	rdmfresno
19	argyles122	f1outsourcing
20	Walucas94	Meganight

Visualization test of Subgraph



This application is made using tkinter library in python. It uses matplotlib for plotting the graph



Thank You