IR Assignment 3

Methodology followed for Question 1(Link Analysis)

Firstly, we downloaded Gnutella peer-to-peer network, August 4 2002 from the asked SNAP database,

1. Data was downloaded in form of a tab separated txt files, we iterated over each line and appended the tab separated value in a list of list.

2. Converted list of list to dataframe, then removed redundant rows from the top of dataframe.

3. For **counting number of nodes** we stored all unique nodes in the list, then finally the length of list leads to the number of nodes.

4. For **counting number of edges,** we detected total count of from node and to node pairs.

5. For creating **adjacency matrix,** firstly created empty dataframe with number of rows as unique number of nodes and columns as unique number of nodes plus one extra column for storing node number in it.

6. Populated adjacency matrix, using earlier created dataframe with row number as from node and column number as to node. Output for adjacency matrix is shown as follows:

	Node	0	1	2	3	4	5	6	7	8	...	10869	10870	10871	10872	10873	10874	10875	10876	10877	10878
0	0	0	1	1	1	1	1	1	1	1	...	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10871	10874	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
10872	10875	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
10873	10876	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
10874	10877	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	1	0	0
10875	10878	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

10876 rows × 10877 columns

7. For creating **edge list,** containing from node and to node pair in them. Partial extract from which is shown as follows:

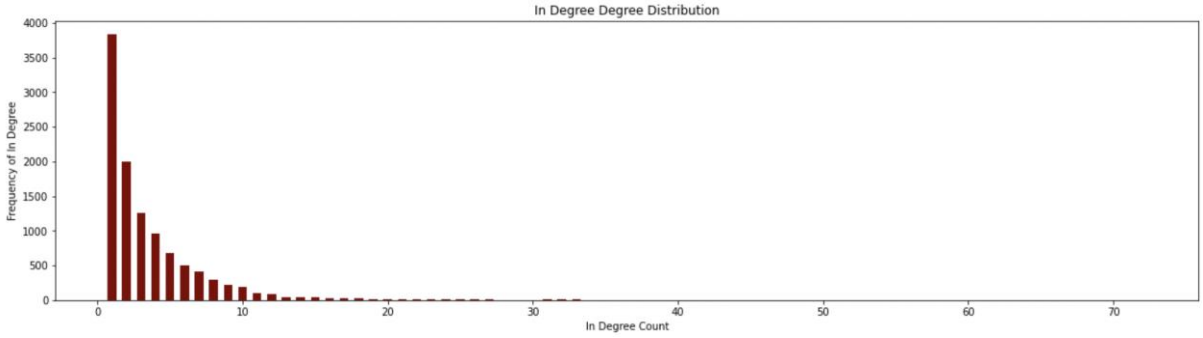
```
[[0, 1],
 [0, 2],
 [0, 3],
 [0, 4],
 [0, 5],
 [0, 6],
 [0, 7],
 [0, 8],
 [0, 9],
 [0, 10],
 [1, 2],
 [1, 11],
```

8. For calculating **average in degree,** created dictionary with node as the key and number of in-degrees as its value, then took sum of all those values and divided it by number of nodes.

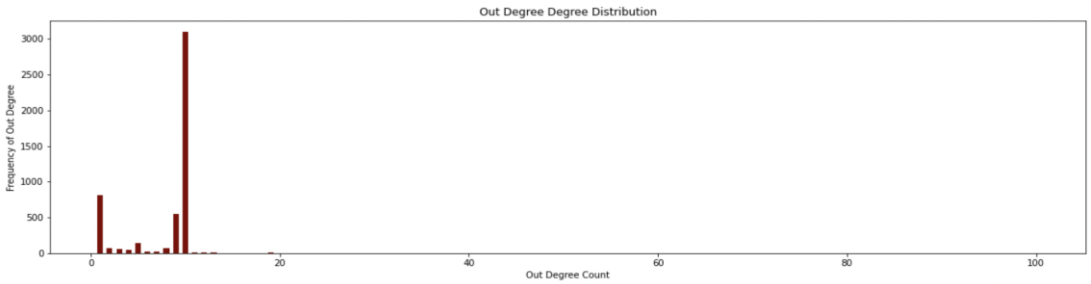
9. For calculating **maximum in degree,** extracted key with maximum value.

10. For calculating **average out degree,** created dictionary with node as the key and number of out-degrees as its value, then took sum of all those values and divided it by number of nodes.

11. For calculating **maximum out degree,** extracted key with maximum value.
12. Calculated **density of the network** using the formula (number of edges)/((number of nodes)*(number of nodes-1))
13. For getting **in** - **degree distribution of network,** we created dictionary with count of indegree as key, and their count as respective value.
14. Then, we plotted of in-degree count v/s their respective frequency on bar plot and got the result as follows:



15. For getting **out** - **degree distribution of network,** we created dictionary with count of outdegree as key, and their count as respective value.
16. Then, we plotted of out-degree count v/s their respective frequency on bar plot and got the result as follows:

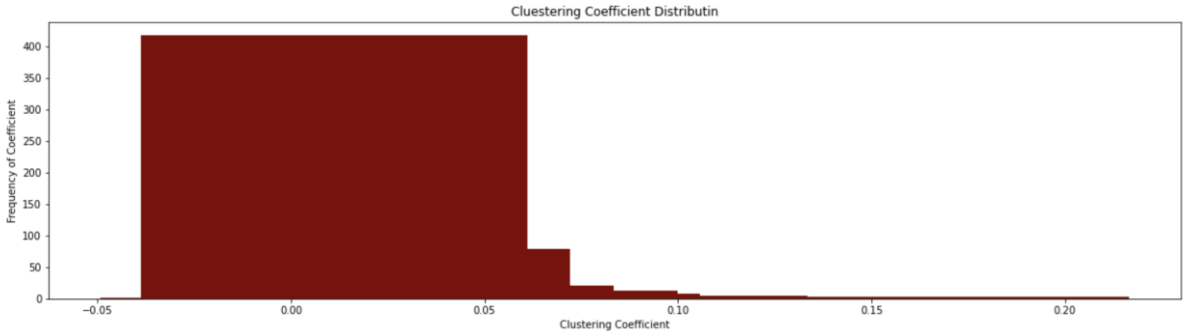


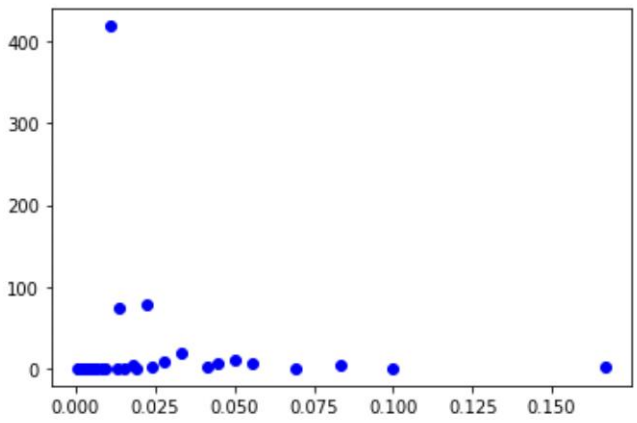
17. For calculating **local clustering coefficient**, we iterated over each node of the graph, then found out it's neighboring nodes and stored in the list using from-to node dataframe, then iterated over the dataframe while placing each of the neighbouring node as 'from node' one by one, and checking if the 'to node' also lies from the list created earlier, since our dataset is directed in nature we explicitly took connection from a to b, and , b to a as single edge only.

    Then, while iterating over each node, we calculated clustering coefficient as – (number of connection in neighboring nodes)/(number of possible connection between neighboring node)

    Stored node as key and it's coefficient value in dictionary.

18. Finally for **plotting clustering-coefficient distribution of the network,** created another dictionary with clustering coefficient as the key and it's respective count as the value.
    Plotted the same in form of a bar graph as well as scatter plot, output for which is depicted as follows:





Methodology followed for Question 2 (PageRank, Hubs and Authority)

Library Used: - Networkx

Firstly, we downloaded Gnutella peer-to-peer network, August 4 2002 from the asked SNAP database,

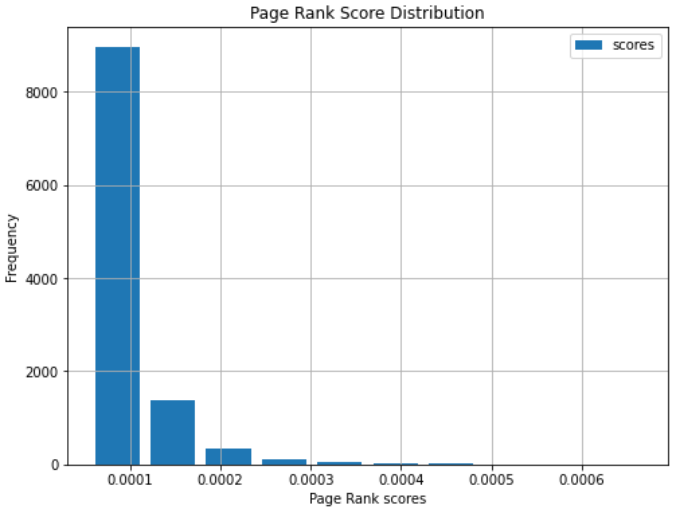
1. The data was then read and a graph was constructed using the '.txt' data using the '**read_edgelist()**' function of the '**networkx**' library. The graph contains **10876 nodes** and **39994 edges.**

2. **Page Rank Score**: -

   After constructing the graph, the Page Rank scores were computed for each node in the graph, using the '**pagerank()**' method of '**networkx**' library and then the nodes were ranked based on their page rank scores.
   We got the following results: -

```
{'1056': 0.0006711727183638692,
 '1054': 0.0006625823159671494,
 '1536': 0.0005496656851576059,
 '171': 0.0005434801433858492,
 '453': 0.0005243733925984653,
 '407': 0.0005097076151434906,
 '263': 0.0005079084313868783,
 '4664': 0.0005023514218978612,
 '1959': 0.0004892066518182482,
 '261': 0.0004858173126082881,
 '410': 0.0004849712292876587 4,
 '165': 0.0004841257759138581,
 '1198': 0.0004610584770426449,
 '127': 0.0004487289379563434 7,
 '4054': 0.0004377941409535 53,
 '2265': 0.0004322961966903 9003,
 '345': 0.0004310608487193 0077,
 '763': 0.0004304581456241818,...
```

We have also plotted the Page rank score vs frequency plot to show the score distribution, as shown: -



3. **Authority Score and HUB Score**: -

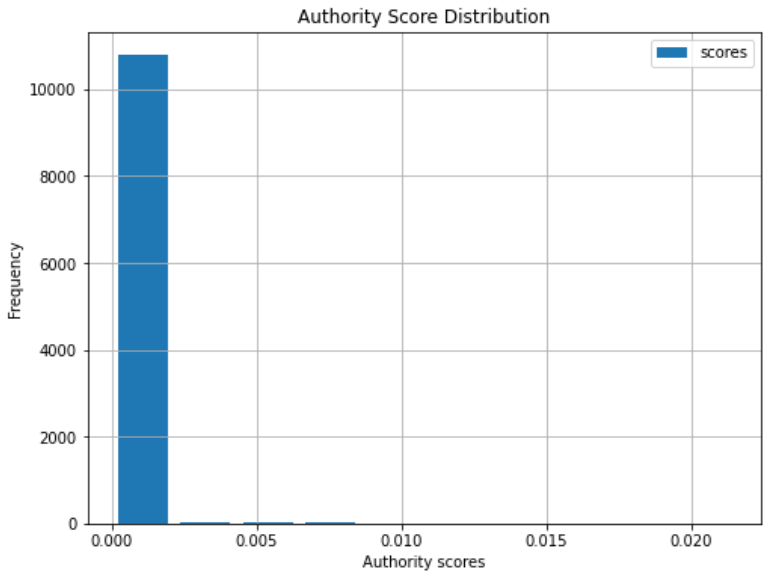
We have used the **HITS (Hyperlink Induced Topic Search) algorithm** of the '**networkx**' library to compute the Authority and the Hub Scores for each node.

Authority Score: -

After obtaining the Authority scores, the nodes were ranked based on their calculated Authority scores.
We got the following results: -

```
{'1054': 0.02155377863120839,
 '261': 0.01684254000613121,
 '453': 0.015861410734500224,
 '407': 0.01494611752902302,
 '410': 0.012339436489592001,
 '699': 0.011927472691451424,
 '1056': 0.011347590531563295,
 '3076': 0.011194144989682055,
 '989': 0.010582621487363296,
 '2195': 0.009938456915497641,
 '1198': 0.00978808400463488,
 '2196': 0.009138135533327688,
 '412': 0.009072144189599033,
 '2197': 0.008805829358698529,
 '165': 0.00875627419359887,
 '763': 0.008638965879016256,...
```

We have also plotted the Authority score vs frequency plot to show the score distribution, as shown below: -

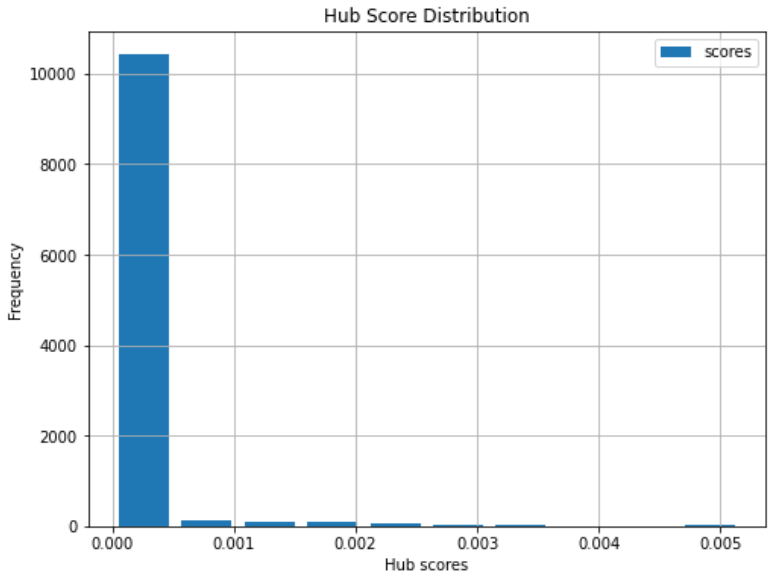


Hub Score: -

After obtaining the Hub scores, the nodes were ranked based on their calculated Hub scores.
We got the following results: -

```
{'3154': 0.005167046979753697,
 '4645': 0.004990291476323975,
 '4866': 0.004990291476323975,
 '5256': 0.004990291476323975,
 '4942': 0.004944090430452508,
 '3020': 0.004839221057079094,
 '6083': 0.004839221057079094,
 '4745': 0.004823144762597137,
 '4990': 0.004823144762597137,
 '2443': 0.004798257833167894,
 '3831': 0.004715355472993939,
 '5722': 0.004673300708676307,
 '5431': 0.0045395620087731,
 '2628': 0.004374690793200526,
 '3899': 0.00436323166576853,...
```

We have also plotted the Hub score vs frequency plot to show the score distribution, as shown below: -



Comparison of Results: -

On analyzing and comparing the results of all the scoring methods, we found out that different nodes were assigned the highest rank/score in each of the scoring methods.

This is due to the fact that, **PageRank** computes the ranking/scoring of the graph nodes, based on the structure of the incoming links, therefore it assigns maximum score to the node, having the maximum in-degree i.e., **the node 1056,** as shown in our output.

Whereas, in **Authority Scores**, instead of assigning higher values to the node with maximum in-degree, it rather assigns higher value to those nodes, which have incoming links from maximum number of HUB nodes. In this case, all the incoming nodes to the concerned node, may not be considered since all of them may not satisfy the maximum HUB nodes criteria. In our case, **the node 1054 was assigned the maximum Authority Score.**

In **Hub Scores**, the score of a node is computed based on the outgoing links from that node, therefore the node with maximum number of outgoing links is assigned the highest Hub score. In our case, **the node 3154 was assigned the maximum Hub Score.**

Scoring Method	Highest Ranked Node	Highest Score
Page Rank	1056	0.0006711727183638692
Authority	1054	0.02155377863120839
Hub	3154	0.005167046979753697

Comparison plot to compare different scoring methods used →