Assignment - 3

Samyak Jain (2019098) and Sarthak Johari (2019099)

Question - 1

We were asked to pick a real-world network dataset with nodes > 100, so for this purpose, we have chosen wiki-vote to be our dataset. The network has information about Wikipedia voting (related to the promotion of user to adminship) and the data is collected from the period of the start of Wikipedia to January 2008. The nodes in the network represent the Wikipedia users, and a directed edge in the graph from i^{th} node to j^{th} node represents that user i voted for user j.

Methodology

Firstly, it was observed that the nodeIDs were not continuous in a particular range of integers thus we assigned a unique node number to each node and maintained a map from nodeID to nodeNumber and vice versa too. the nodeNumber is an integer in the range [0, N] where N = Number of nodes in the graph.
Following this, we form an ‘adjacency matrix’ and ‘edge list’ representing the network. The ‘adjacency matrix’ A is an NxN matrix where A_{ij}=1 if there is an edge from the node with nodeNumber ‘i’ to the node with

nodeNumber ‘j’ in the directed network otherwise, $A_{ij} = 0$. Edge list is a list of tuples where each tuple (i,j) represents that there is an edge from the node with nodeNumber ‘i’ to the node with nodeNumber ‘j’ in the directed network.

Following is the information about the dataset:

1. Number of Nodes: 7115
2. Number of Edges: 103689
3. Avg In-degree: 14.573295853829936
4. Avg. Out-Degree: 14.573295853829936
5. Node with Max In-degree : Node ID - 4037 | In-Degree = 457.0
6. Node with Max out-degree : Node ID - 2565 | Out-Degree = 893.0
7. The density of the network: 0.0020485375110809584

Metrics calculation:

- The number of nodes and number of edges were directly present in the dataset file. Still, we performed a sanity check and calculated the number of unique nodes and the number of edges (number of lines in the edge list part of the text file) and the results were consistent with the expected.

- Avg In-degree: Firstly indegree of all nodes is calculated and stored in a form of a dictionary with the key representing the node number and value representing the in-degree of that node. For calculating the in-degree of a node with node number ‘i’, we take the sum of elements of i^{th} column because that would have value 1 whenever there is an edge incoming to node ‘i’ from any other node. After getting the in-degree of all the nodes, we simply take the average of the in-degree values over all nodes.

$$\text{Average in-degree} = \frac{\text{Sum of in-degrees of all nodes}}{N}$$

- Avg out-degree: Firstly outdegree of all nodes is calculated and stored in a form of a dictionary with the key representing the node number and the value representing the out-degree of that node. For calculating the out-degree of a node with node number ‘i’, we take the sum of elements of i^{th} row because that would have value 1 whenever there is an edge outgoing from node ‘i’ to any other node. After getting the out-degree of all the nodes, we simply take the average of the out-degree values over all nodes.

$$\text{Average out-degree} = \frac{\text{Sum of out-degrees of all nodes}}{N}$$

- We calculated the in-degrees of all the nodes previously. So we select the node with the maximum value of indegree from the dictionary of indegree we have maintained.

- We calculated the out-degrees of all the nodes previously. So we select the node with the maximum value of out-degree from the dictionary of out-degree we have maintained.

- For network density we have used the formula :

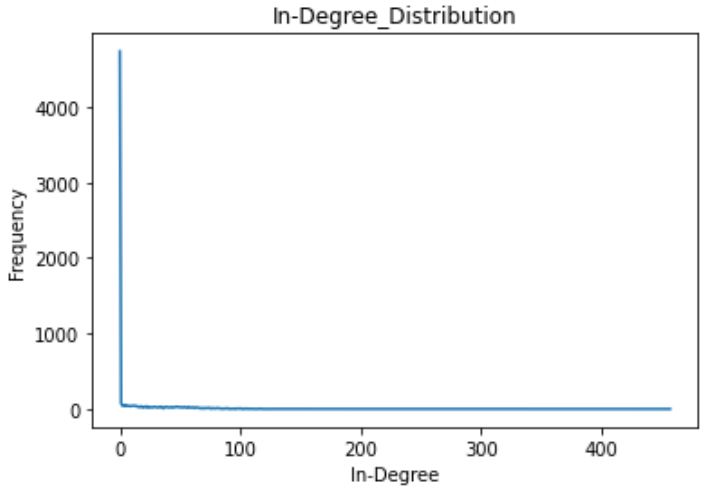
$$\text{Network-density} = \frac{\text{No. of edges present in the network}}{\text{Total possible edges in the network}}$$

In a directed graph network,

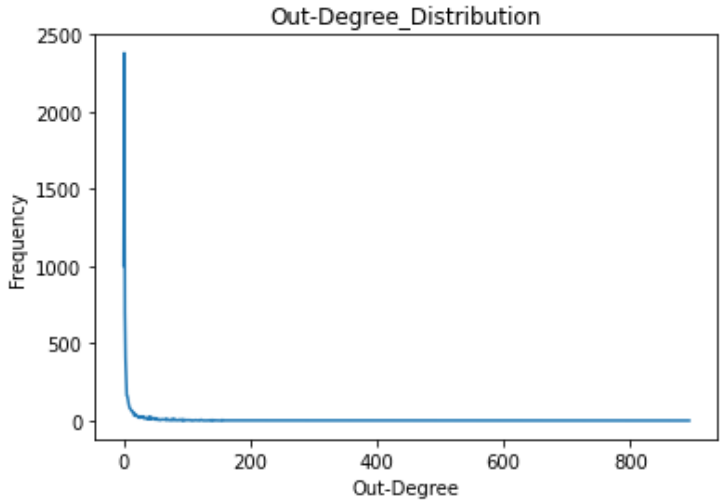
Total possible edges in network = (N)*(N - 1)
where, N = Number of nodes in the graph

Following are the degree distribution plots for the network:

- In-Degree Distribution:



- Out-Degree Distribution



Local Clustering Coefficient (LCC)

Note: The graph is converted to an undirected version as instructed for this part.

The approach followed for calculating the LCC for each node is as follows:

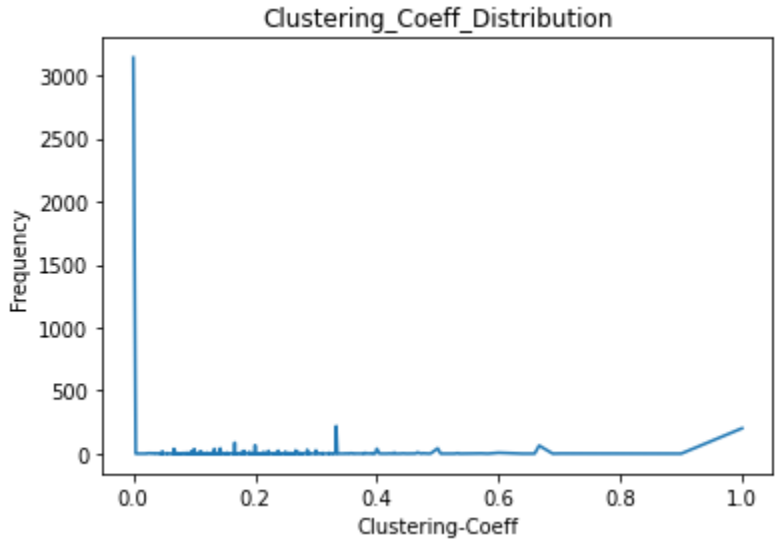
For a node, we first find the neighbour hood i.e the set of all the neighbour nodes that the given node is directly connected to. Let the number of neighbouring nodes for node ‘i’ be N_{V_i}. We then find the number of edges(links) existing between the neighbours of ‘i’, let it be N_{E_i}. Then the value of LCC for node ‘i’ is :

Total possible links between neighbours = $N_{T\_i} = (N_{V\_i})*(N_{V\_i} - 1) / 2$
[because we have considered undirected version here]

$$\mathbf{LCC_i = N_{E\_i} / N_{T\_i}}$$

This way, the local clustering coefficient of each node is calculated.

The plot for clustering-coefficient distribution is :



Question - 2

In this question, we were asked to give PageRank score and Authority and Hub score to each node and also calculate the hub and authority scores. We first create a directed network using (a DiGraph object).

- For calculating the PageRank score, we use the networkx.pagerank*()* function provided in the networkx library.
  The page rank algorithm was developed for the purpose of ranking web pages, it computes and gives a score to a node based on the structure of the incoming links to that node in a graph G. The idea behind the algorithm is that the node which has more incoming links from other nodes is likely to have more importance. It involves a random walk process over the nodes, where nodes are visited with some probability values. Nodes with more incoming edges tend to be visited more frequently and are thus more important. This random walk process is combined with a teleport operation where the web surfer can jump from the current node to any node in the web graph with equal probability. This teleport operation can be used in two ways : (i) when there are no outgoing edges from a node (dead end), (ii) At any node, the teleport operation is invoked with a certain probability of
  $0 < \alpha < 1$ and the normal random walk process is carried out with probability $1 - \alpha$. Thus in this random walk and teleportation process, each node '*u*' of the web graph is visited a fixed fraction of time $\pi(u)$ - the pagerank of *u.* In terms of equations, let π be the probability distribution of the web surfer across web pages (nodes), then after certain iterations, we arrive at the steady-state distribution such that
  $$\pi P = \pi$$

where P is the transition probability matrix. The left principal eigenvector of P (with the corresponding eigenvalue as 1) will give the pagerank values for the nodes.

- Now the nodes can have 2 types of scores to them. One is the authority score and the other is the hub score. They are basically metrics used for the evaluation of a node. Authorities are basically those nodes that contain useful information and its importance is measured by incoming links, whereas hubs are the nodes that point towards authorities.

  In mathematical terms, the authority score of a node X is the sum of hub scores of all the nodes that point to X. Hub score of a node X is the sum of authority scores of all the nodes that X points towards.

  To begin with authority and hub scores are initialized by 1 for each node. Then repeated iterations are made of the authority update rule and the hub update rule which are given below.

  For node X

$$Hub\,(X) \;=\; \sum_{q \in P} Authority(q),\ (P \;=\; are\ the\ set\ of\ nodes\ that\ X\ links\ to)$$

$$Authority\,(X) \;=\; \sum_{q \in P} Hub(q),\ (P \;=\; are\ the\ set\ of\ nodes\ that\ link\ to\ X)$$

  To prevent the values from diverging we normalize the values after each iteration to obtain converging values.
  For this question, we have used networkx [networkx.hits()] for the calculation of authority and hub scores for each node.

- After finding out the values of the PageRank scores, Authority and Hub scores for each node, we sort them in the decreasing order of the respective scores to make some observations. Following are top 10 (nodeID, score) pair for each type of score :

  - PageRank Score :

```
Top 10 NodeIDs based on PageRank score
[(4037, 0.0046127158911675 41),
 (15, 0.00368122072952927),
 (6634, 0.0035248136576402 56),
 (2625, 0.0032863743692309 01),
 (2398, 0.0026053331717250175),
 (2470, 0.002530105328384948),
 (2237, 0.0025047038004839886),
 (4191, 0.0022662633042363433),
 (7553, 0.0021701850491959575),
 (5254, 0.0021500675059293213)]
```

  - Authority Score :

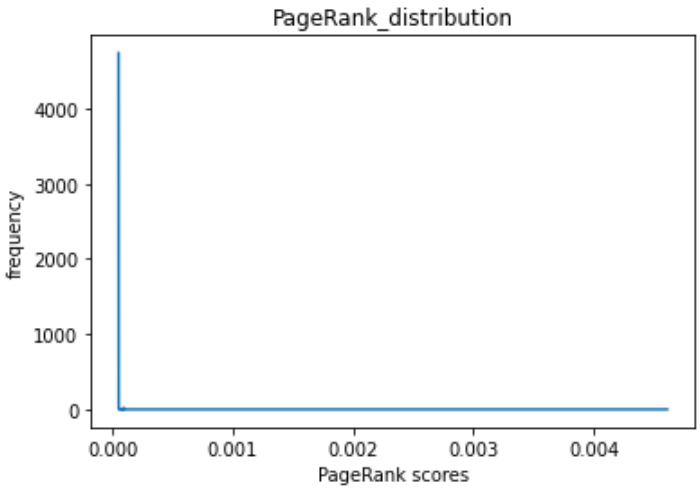
```
Top 10 NodeIDs based on Authority score
[(2398, 0.0025801471780088 74),
 (4037, 0.0025732411242297927),
 (3352, 0.0023284150914976835),
 (1549, 0.002303731480457179),
 (762, 0.00225587485628714),
 (3089, 0.0022534066884511 63),
 (1297, 0.0022501446366627 23),
 (2565, 0.0022235641039536134),
 (15, 0.002201543492565582),
 (2625, 0.0021978968034030 73)]
```

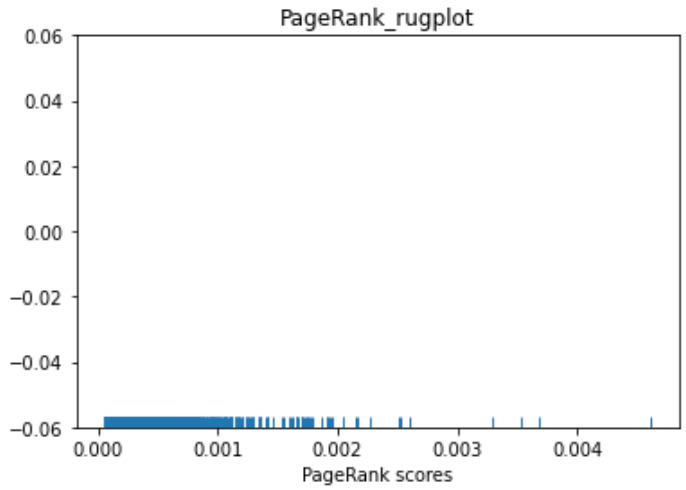
- Hub Score :

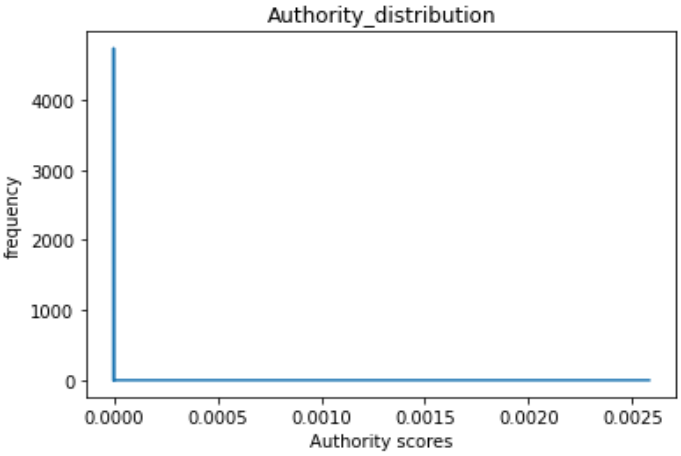
```
Top 10 NodeIDs based on Hub score
[(2565, 0.007940492708143143),
 (766, 0.007574335297501252),
 (2688, 0.006440248991029864),
 (457, 0.006416870490261073),
 (1166, 0.006010567902411202),
 (1549, 0.005720754058269244),
 (11, 0.004921182063808105),
 (1151, 0.004572040701756409),
 (1374, 0.004467888792711108),
 (1133, 0.003918881732057349)]
```

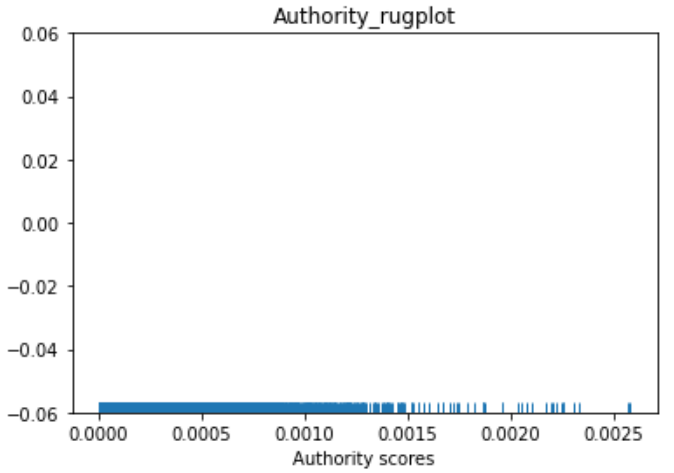
On observing, we can observe that there are a lot of common nodeIDs in the top 10 based on PageRank and Authority score for eg - 4037, 15, 2625, 2398. This is because both of the scoring techniques treat the nodes with more incoming links as more important and compute the rankings based in the incoming links to a node. PageRank is based on the structure of incoming links and is likely to give more importance to nodes with more incoming links (because the nodes with the most incoming links will be visited more often in the random walk process) and the authority score is directly based on inbound (incoming) links. Thus because of this kind of similarity, they share some common nodes in the top 10. Also, we can observe that the node with the maximum indegree (in-degree of = 457) as we found in the first question i.e - nodeID 4037 is very highly rated by both the algorithms (top in PageRank scores, and second in Authority score). HITS algorithm as we know computes the hub score for the node based on outgoing links and thus we can observe that the node with max out-degree (out-degree = 893) - nodeID 2565 (as found from the first question) has the highest hub score over here. Also, the values of authority score and hub

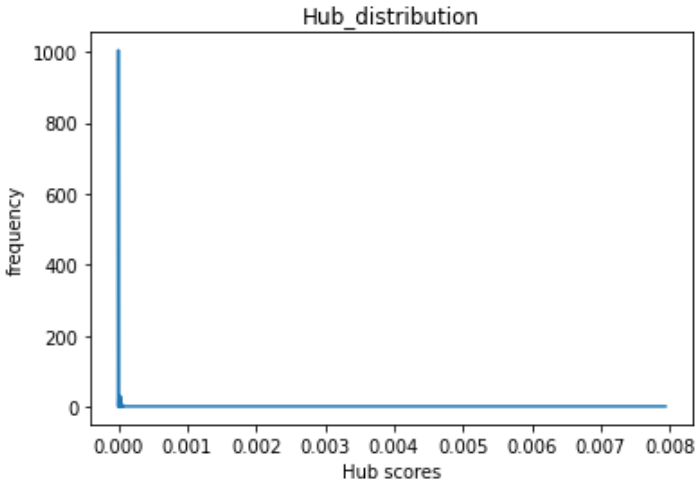
score are normalized over here (by default in inbuilt networkx implementation) to prevent them from diverging. We further try to visualize the distribution of these scores using rugplots, in which the value of the single variable is displayed as marks/ticks along a single axis) [barplot was too cluttered due to the value of scores being around very small values):

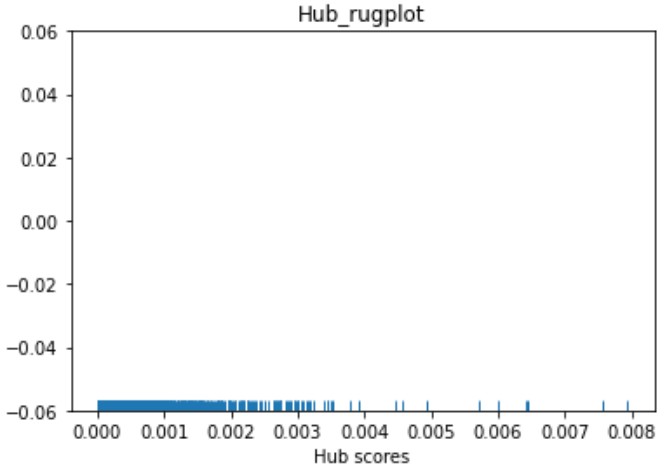












The distribution further gives insights about how the distribution is dense around certain values or between a range of values (the range can be seen from the graph). All of them have certain outlying high values for some of the nodes, but most of the nodes have scores lying in a specific range for each.