IR ASSIGNMENT 3
Group 65

Ques 1.

We picked the ‘Wiki-Vote’ as our real-world dataset with about 7000+ nodes which is related to the promotion of the user to administrator. The nodes in this Wikipedia dataset represents its user, and a directed edge in the graph from node i to node j represents that the ith user has voted jth user.

Methodology:

Since the NodeIDs which were given in the dataset were not present in a practical way of assigning the integer range, we made a unique number to the node and created a hashmap from the NodeID to nodeNumber and similarly for nodeNumber to NodeID, since the nodeNumbers are present in the range from 0 to N, where N represents the total number of nodes in our graph.

In this manner, we created our adjacency matrix and edge-list, which represents the network, where the A is an adjacency matrix with dimensions NXN, and any value 1 in the matrix represents the existence of an edge from node with nodeNumber ‘i’ to node with the nodeNumber ‘j’. In the same way, an edge-list is a list of tuples where each tuple (i,j) represents that an edge from the node with nodeNumber ‘i’ to node with nodeNumber ‘j’ exists in our graph.

We obtained the following information related to our dataset:

a. **Number of Nodes** : 7115
b. **Number of edges** : 103689
c. **Avg. In-Degree** : 14.573295853829936
d. **Avg. Out-Degree** : 14.573295853829936
e. **Max In-Degree Node** : Node ID - 4037 | In-Degree = 457.0
f. **Max Out-Degree Node** : Node ID - 2565 | Out-Degree = 893.0
g. **Network Density** : 0.0020485375110809584

Calculation of the Metrics:

- Number of Nodes and Edges: We mounted our Wiki-Vote dataset and iterated over the rows to compute the number of unique nodes and number of edges which are presented in our data. The numbers obtained matched with the information provided about the data on the web.
- Average In-Degree: In-degree refers to the presence of an edge which is incoming to some node i from any other node. We are storing the node number with its in-degree value in the form of a dictionary. We only need to count the number of 1’s present in the ith column. To

compute the average-value, we can take the average of all the in-degree values to obtain the average in-degree.

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

- Average Out-Degree: Out-degree refers to the presence of an edge which is outgoing to some node i from any other node. We are storing the node number with its out-degree value in the form of a dictionary. We only need to count the number of 1's present in the ith row. To compute the average-value, we can take the average of all the out-degree values to obtain the average out-degree.

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

- Node having max In-degree: To find the max in-degree value from all the nodes, we can run a loop over the dictionary to compare the maximum values between all the nodes and then print the highest value over all of them.

- Node having max Out-degree: To find the max out-degree value from all the nodes, we can run a loop over the dictionary to compare the maximum values between all the nodes and then print the highest value over all of them.

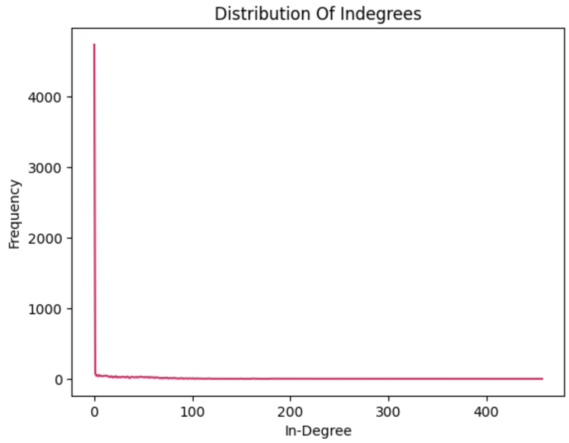
- Network Density:

$$\text{Network Density} = \frac{\text{Number of edges present over the network}}{\text{Total possible edges in the network}}$$

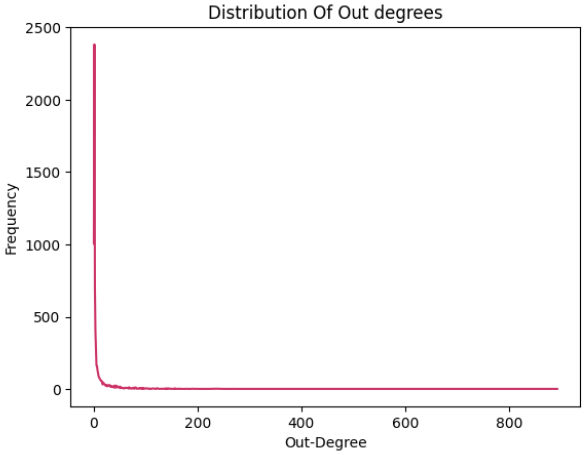
Total possible edges in the network for a directed graph can be taken as N*(N-1), where N refers to the nodes in the graph.

The degree-distribution plots can be as follows:

1. In-degree distribution:



2. Out-degree distribution:



3. **Local-Clustering Coefficient:**

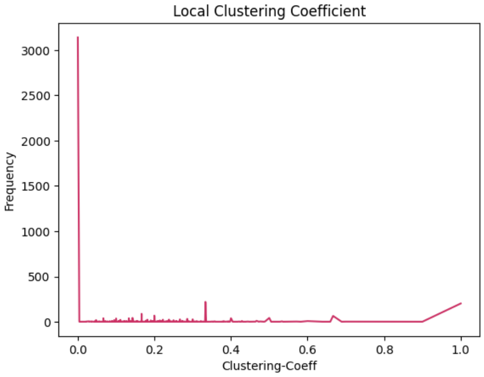
The graph has been constructed to an undirected format, by removing the direction of an edge from Node i to node j by considering that an edge also exists from node j to node i.

The approach which has been followed to implement the LCC for each node has been as follows:

For a node, we need to find the neighborhood, i.e. all the possible neighboring nodes which were directly connected to the given node. Consider the number of neighbors for the node i as N_{Vtx_i}, and the number of edge-links which exist between the neighbors of ‘i’ let it be NE_i.

Then the value of LCC for a node ‘i’ is :
Total possible links between neighbors = N_{T_i} = $((N_{Vtx\_i})*(N_{Vtx\_i} - 1))/ 2$, as we have an undirected graph. $LCC_i = N_{E\_i} / N_{T\_i}$

The local clustering coefficient(LCC) of each node can be computed in this manner. The plot for clustering-coefficient distribution is :



Ques 2.

We are supposed to compute the 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 an object of a Digraph.

To calculate the PageRank score, we use the inbuilt function pagerank() 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. The algorithm states that the node which has more incoming links(in-degree value) as compared to 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 become 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.
There are two ways of doing this teleport operation:
(i) when there are no outgoing edges from a node (dead end).
(ii) The teleport operation gets invoked at any node with a certain probability of $0 < α < 1$ and the normal random walk process is carried out with probability $1 - α$.

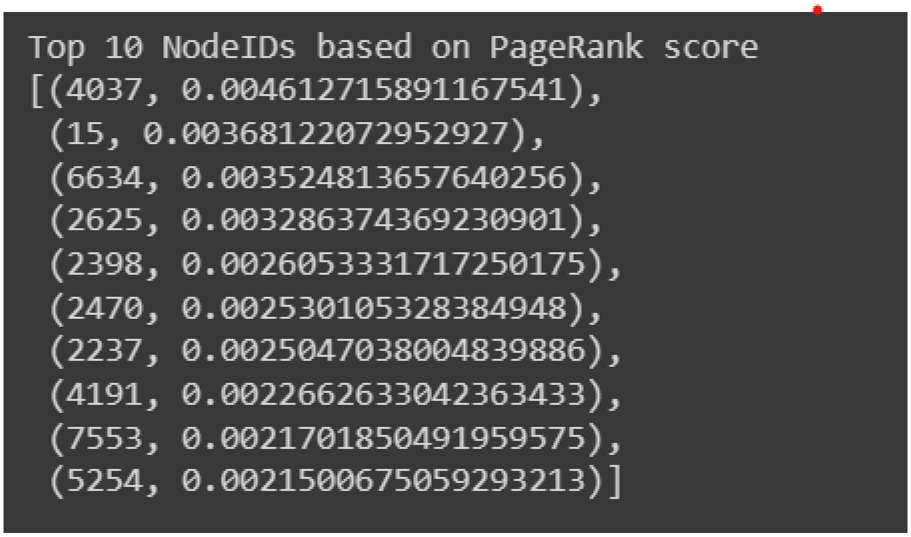
In this random walk and teleportation process, each node ‘u’ of the web graph gets visited in a fixed fraction of time π(u) - which is the pagerank of u. In terms of equations, let π be the probability distribution of the web surfer across web pages (nodes), then after certain iterations, we arrive at the steady-state distribution such that $πP = π$, 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.

Two different types of scores are possible for the nodes. The hub score is one, and the authority score is another. They are essentially measures that are used to assess a node. Hubs are nodes that point to authorities, whereas authorities are nodes that hold meaningful information whose value is determined by incoming links. The authority score of a node X is defined mathematically as the sum of the hub scores of all the nodes that point to X. The authority scores of all the nodes that node X points towards make up the hub score of that node. Authority and hub scores are initially initialized at 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, $Hub\ (X) = q \in P \sum Authority(q)$, ($P$ refers to the set of nodes which links to X)
$Authority\ (X) = q \in P \sum Hub(q)$, ($P = are\ the\ set\ of\ nodes\ that\ link\ to\ X$)
To prevent the values from diverging we normalize the values after each iteration to obtain converging values. Here, we have used networkx [networkx.hits()] for the calculation of authority and hub scores for each node.

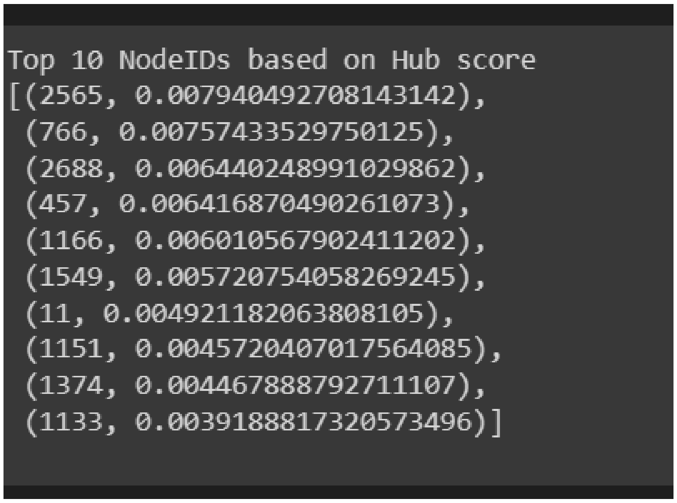
After determining the numbers for each node's PageRank, Authority, and Hub scores, we sorted the results by decreasing score in order to draw some conclusions. The top 10 (nodeID, score) pairs for each type of score are listed below:

PageRank Score:



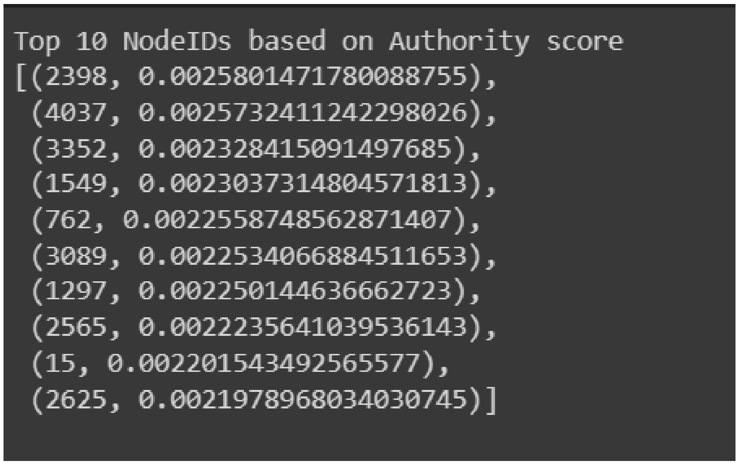
```
Top 10 NodeIDs based on PageRank score
[(4037, 0.004612715891167541),
 (15, 0.0036812207295292 7),
 (6634, 0.003524813657640256),
 (2625, 0.003286374369230901),
 (2398, 0.0026053331717250175),
 (2470, 0.002530105328384948),
 (2237, 0.0025047038004839886),
 (4191, 0.0022662633042363433),
 (7553, 0.0021701850491959575),
 (5254, 0.0021500675059293213)]
```

Hub Score:



```
Top 10 NodeIDs based on Hub score
[(2565, 0.007940492708143142),
 (766, 0.007574335297501 25),
 (2688, 0.006440248991029862),
 (457, 0.006416870490261073),
 (1166, 0.006010567902411202),
 (1549, 0.005720754058269245),
 (11, 0.004921182063808105),
 (1151, 0.0045720407017564085),
 (1374, 0.004467888792711107),
 (1133, 0.0039188817320573496)]
```

Authority Score:

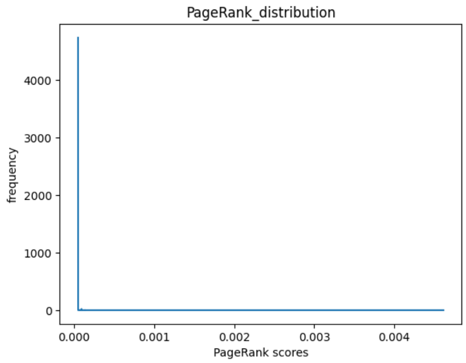


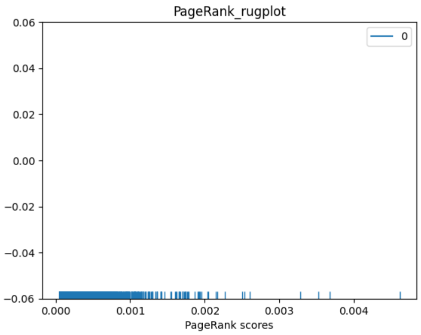
```
Top 10 NodeIDs based on Authority score
[(2398, 0.002580147178008 8755),
 (4037, 0.002573241124229 8026),
 (3352, 0.002328415091497685),
 (1549, 0.0023037314804571813),
 (762, 0.002255874856287 1407),
 (3089, 0.0022534066884511653),
 (1297, 0.002250144636662723),
 (2565, 0.0022235641039536143),
 (15, 0.002201543492565577),
 (2625, 0.0021978968034030745)]
```

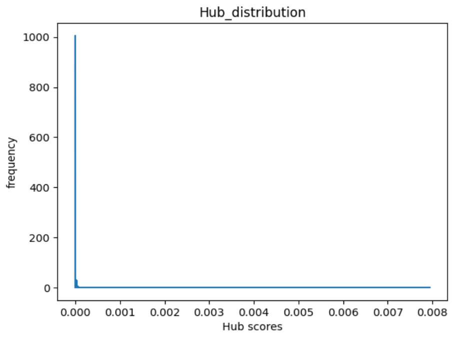
Based on the PageRank and Authority score, we can see that the top 10 nodeIDs share a lot of similarities, for example, 4037, 15, 2625, and 2398. This is due to the fact that both scoring methods assign a higher importance to nodes with more incoming links and base their ranks on these links. The authority score is directly dependent on inbound (incoming) connections, whereas PageRank is based on the structure of incoming links and is likely to assign more weight to nodes with more incoming links (since the nodes with the most incoming links will be visited more frequently in the random walk process). They therefore share some common nodes at the top of the graph as a result of this commonality.

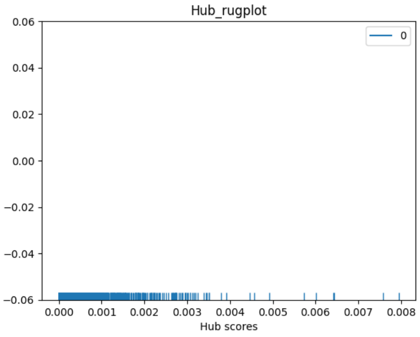
Furthermore, we can see that nodeID 4037, which has the highest indegree (in-degree of = 457) as determined by the first query, is ranked very well by both algorithms (first in PageRank scores and second in Authority scores). We can see that the node with the greatest hub score over here is nodeID 2565 (as determined from the first question), which has the maximum out-degree (out-degree = 893) as determined by the HITS algorithm. In order to prevent divergence, the values of the authority score and hub score are also normalized here (by default in the built-in networkx implementation).

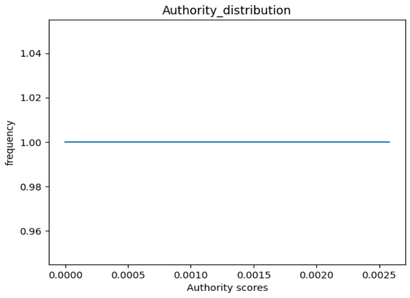
We try to visualize the distribution of the scores using rugplots, where the value of the single variable is displayed as marks/ticks along a single axis) [barplots were too cluttered due to the value of scores being around very small).

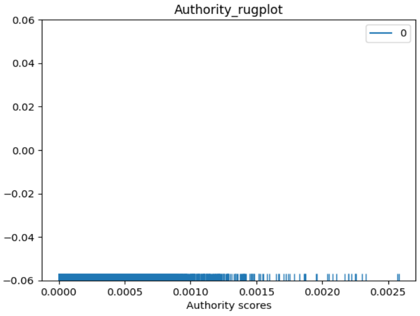

PageRank_distribution


PageRank_rugplot


Hub_distribution


Hub_rugplot


Authority_distribution


Authority_rugplot

The distribution also reveals how the distribution is dense between a range of values or around particular values (the range is shown on the graph). Most of the nodes have scores that fall within a given range for each, however some of them all have certain outlying high values for some of the nodes.