Information Retrieval - CSE508 - Assignment 3

Name: Debnath Kundu (MT22026), Snehal Buldeo (MT22074)

GitHub Link: https://github.com/debnathkundu/CSE508_Winter2023_A3_98

DATASET: Gnutella peer-to-peer network, August 5 2002

Dataset information

A sequence of snapshots of the Gnutella peer-to-peer file sharing network from August 2002. There are total of 9 snapshots of Gnutella network collected in August 2002. Nodes represent hosts in the Gnutella network topology and edges represent connections between the Gnutella hosts.

Dataset statistics	
Nodes	8846
Edges	31839
Nodes in largest WCC	8842 (1.000)
Edges in largest WCC	31837 (1.000)
Nodes in largest SCC	3234 (0.366)
Edges in largest SCC	13453 (0.423)
Average clustering coefficient	0.0072
Number of triangles	1112
Fraction of closed triangles	0.002546
Diameter (longest shortest path)	9
90-percentile effective diameter	5.3

Assumptions:

1. The chosen dataset has the required distribution of scores, and above mentioned information about the dataset is correct.
2. The node numbers present in the dataset are in the range (0, dataset.size)
3. All the node numbers (integers) in the above range are present in the dataset
4. The dataset is in the form of a “name.txt” file
5. The first 4 lines in the .txt file are dataset description and the node information is present from the 5th line in the text file.
6. From the 5th line the .txt file is in the format '0\t1\n', where 0 is the source node and 1 is the destination node.

Question 1 - [45 Points] Link Analysis

Pick a real-world directed network dataset (with number of nodes > 100) from here. [2 points] Represent the network in terms of its 'adjacency matrix' as well as 'edge list'.

Adjacency matrix

	0	1	2	3	4	5	6	7	8	9	...	8836	8837	8838	8839	8840	8841	8842	8843	8844	8845
0	0	1	1	1	1	1	1	1	1	1	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8841	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
8842	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
8843	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	1	0	0	0	0
8844	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
8845	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

8846 rows × 8846 columns

```
[15]  1  print("list_of_edges" , list_of_edges) # list_of_edges
      2  print("Length of list_of_edges" , len(list_of_edges) , len(set(list_of_edges)))
```

```
list_of_edges [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (0, 8), (0, 9), (0, 10), (1, 310), (
Length of list_of_edges 31839 31839
```

[28 points] Briefly describe the dataset chosen and report the following:

1. Number of Nodes

```
[16]  1  print("Number of list_of_nodes:", len(adjacency_matrix))
```

```
Number of list_of_nodes: 8846
```

2. Number of Edges

```
[17]  1  print("Number of Edges:", len(list_of_edges))
```

```
Number of Edges: 31839
```

3. Avg In-degree

```
[19]  1  average_in_degree = 0
      2  for i in graphin:
      3      average_in_degree+=len(graphin[i])
      4      if len(graphin[i]) > node_with_max_indegree[1]:
      5          node_with_max_indegree = (i , len(graphin[i]))
      6  print("Average in degree " , average_in_degree / len(adjacency_matrix))
```

```
Average in degree  3.5992539000678274
```

4. Avg. Out-Degree

```
[20]  1  average_out_degree = 0
      2  for i in graph:
      3      average_out_degree+=len(graph[i])
      4      if len(graph[i]) > node_with_max_outdegree[1]:
      5          node_with_max_outdegree = (i , len(graph[i]))
      6
      7  print("Average out-degree " , average_out_degree / len(adjacency_matrix))
```

```
Average out-degree  3.5992539000678274
```

Avg **In-Degree** & **Out-Degree** are same for a dircted graph.

5. Node with Max In-degree

```
[21]  1  print("Node with Max In-degree:",node_with_max_indegree[0])
      2  print("Node with Max In-degree:",node_with_max_indegree[1])
```

```
Node with Max In-degree: 842
Node with Max In-degree: 79
```

6. Node with Max out-degree

```
[22]  1  print("Node with Max out-degree:",node_with_max_outdegree[0])
      2  print("Max out-degree:",node_with_max_outdegree[1])
```

```
Node with Max out-degree: 3002
Max out-degree: 65
```

7. The density of the network

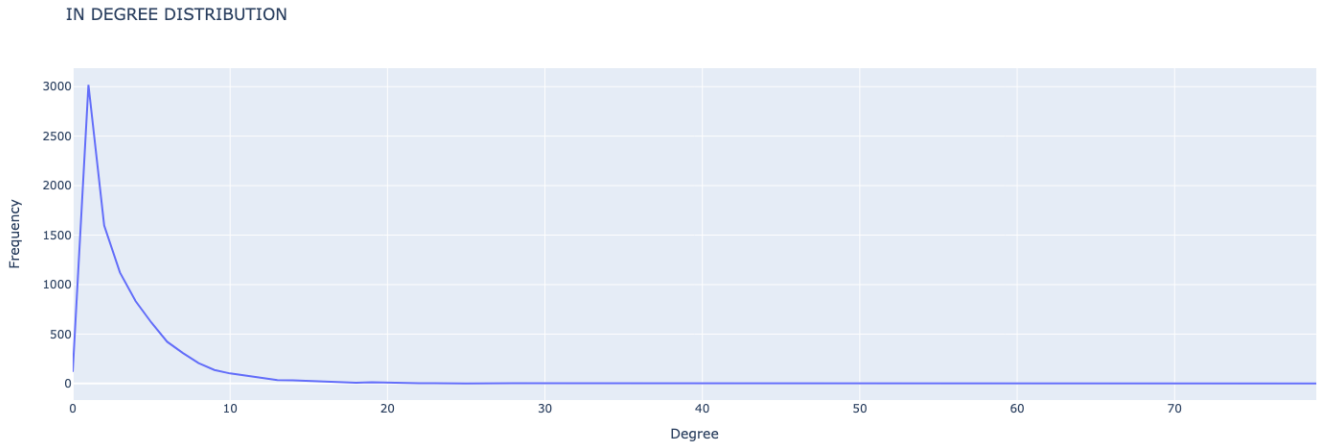
```
[23]  1  maximum_edges = len(graph) * len(graph)
      2  print("The density of the network:" , len(list_of_edges) / maximum_edges)
```

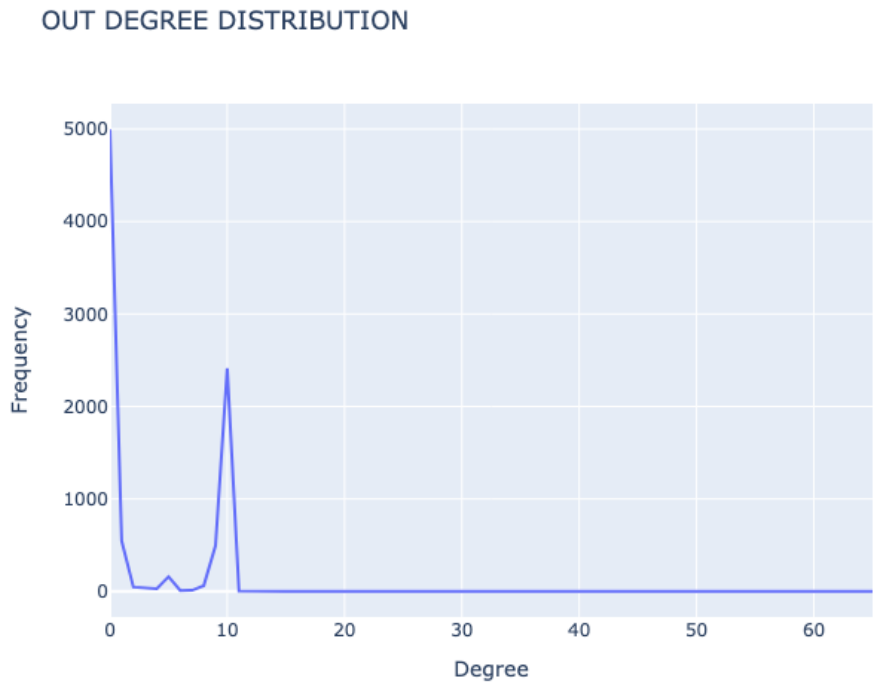
```
The density of the network: 0.00040687925616864426
```

Hence it is a very sparse graph.

Further, perform the following tasks:

1. [5 points] Plot degree distribution of the network (in case of a directed graph, plot in-degree and out-degree separately).





2. [10 points] Calculate the local clustering coefficient of each node and plot the clustering-coefficient distribution (lcc vs frequency of lcc) of the network.

NOTE:

1. You are NOT allowed to use any library to perform the tasks for this question.

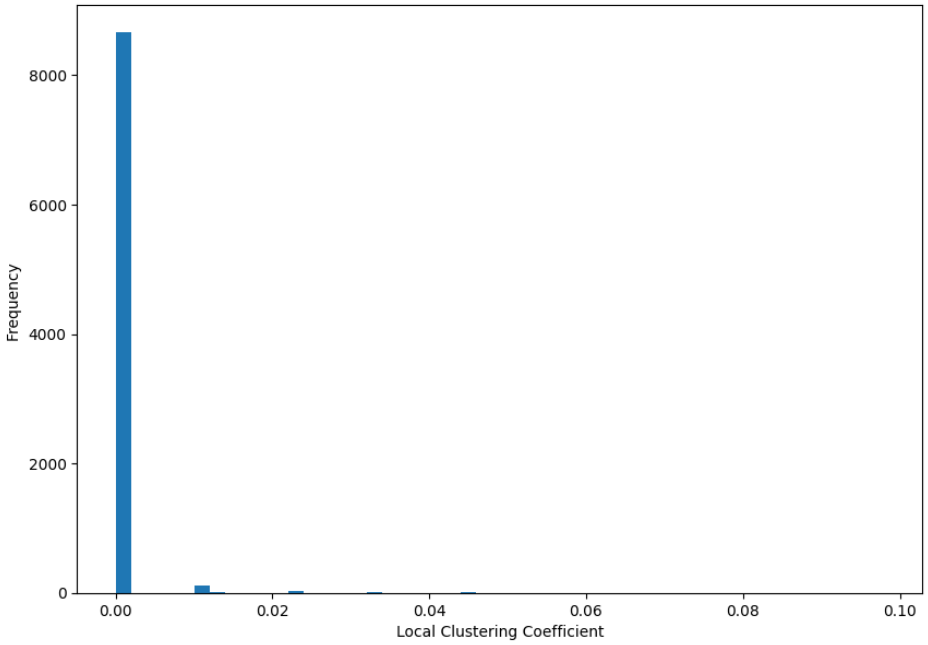
2. Mention the formula for calculating the metrics in your report.

```
[28]  1  clustering_coefficient = []
      2  for i in graph:
      3      t = 0
      4      for j in graph[i]:
      5          for k in graph[j]:
      6              if k in graph[i] :
      7                  t += 1
      8      len_ = len(graph[i])
      9      if len_ < 2:
     10          clustering_coefficient.append(0)
     11      else:
     12          t >>= 1
     13          clustering_coefficient.append(t / ((len_ * (len_-1)  ) ))
```

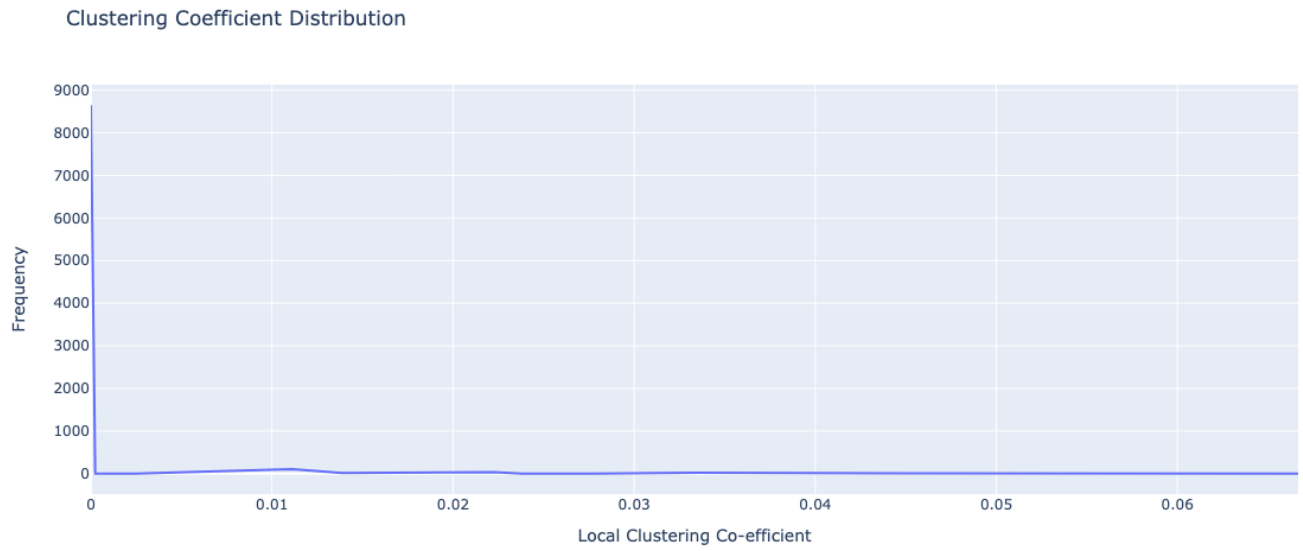
```
[29]  1  for i in range(len(graph)):
      2      print("The local clustering coefficient Node "+ str(i)+"'s  is" , clustering_coefficient[i])
```

```
The local clustering coefficient Node 5618's  is 0.011111111111111112
The local clustering coefficient Node 5619's  is 0.0
The local clustering coefficient Node 5620's  is 0
The local clustering coefficient Node 5621's  is 0.011111111111111112
The local clustering coefficient Node 5622's  is 0
The local clustering coefficient Node 5623's  is 0.0
The local clustering coefficient Node 5624's  is 0
The local clustering coefficient Node 5625's  is 0
The local clustering coefficient Node 5626's  is 0.011111111111111112
The local clustering coefficient Node 5627's  is 0.0
The local clustering coefficient Node 5628's  is 0.0
The local clustering coefficient Node 5629's  is 0.0
The local clustering coefficient Node 5630's  is 0.0
The local clustering coefficient Node 5631's  is 0
The local clustering coefficient Node 5632's  is 0
The local clustering coefficient Node 5633's  is 0
The local clustering coefficient Node 5634's  is 0.0
The local clustering coefficient Node 5635's  is 0.0
The local clustering coefficient Node 5636's  is 0.0
The local clustering coefficient Node 5637's  is 0
The local clustering coefficient Node 5638's  is 0
The local clustering coefficient Node 5639's  is 0
The local clustering coefficient Node 5640's  is 0.0
The local clustering coefficient Node 5641's  is 0
The local clustering coefficient Node 5642's  is 0
The local clustering coefficient Node 5643's  is 0.0
The local clustering coefficient Node 5644's  is 0.022222222222222223
The local clustering coefficient Node 5645's  is 0
The local clustering coefficient Node 5646's  is 0
```

Histogram:



Line Plot distribution:



Question 2: [35 points] PageRank, Hubs and Authority

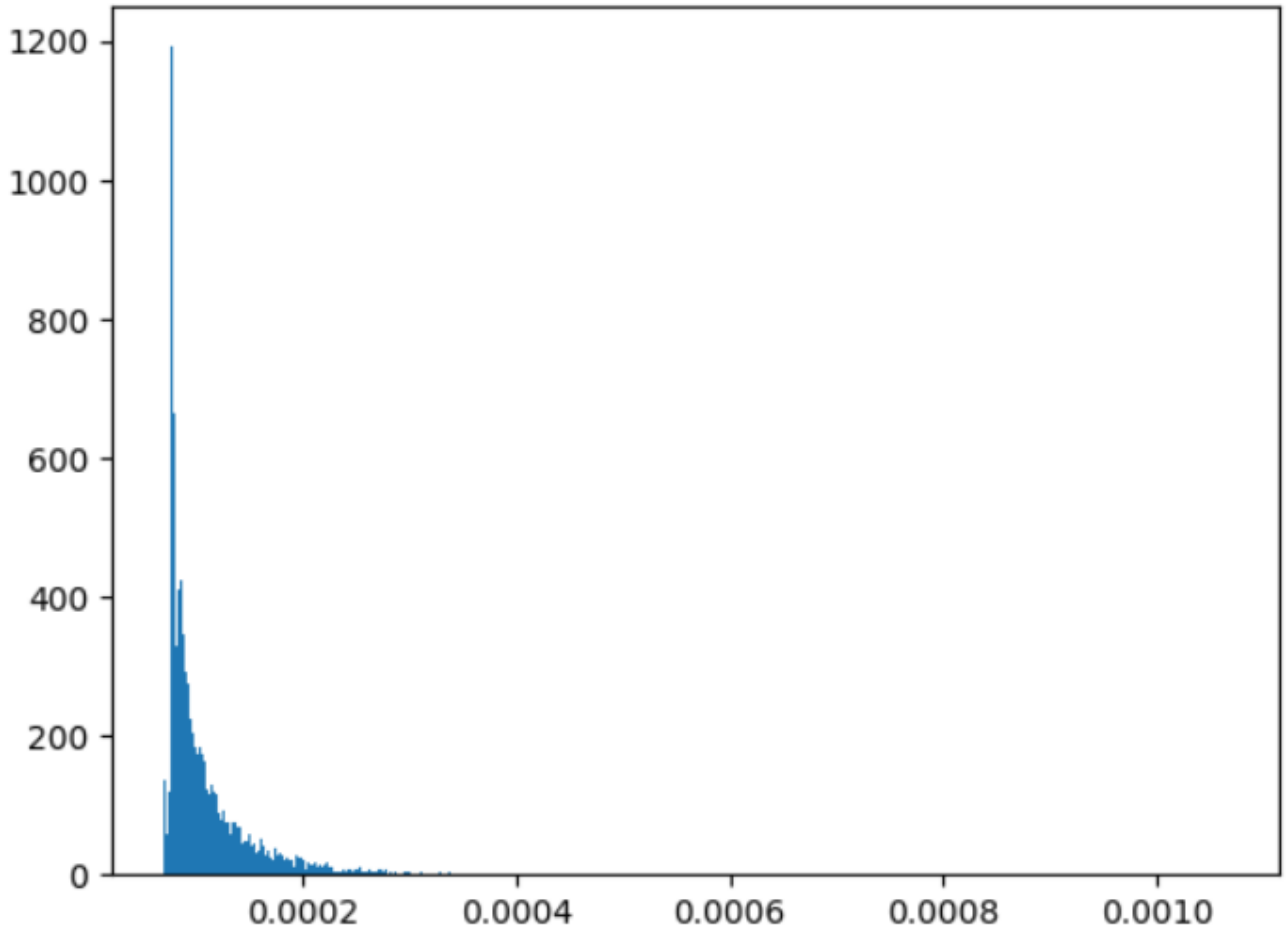
For the dataset chosen in the above question, calculate the following:

1. [15 points] PageRank score for each node

Methodology:

1. An object of Digraph is created to represent a graph using networkx library
    - As the considered graph is a directed graph
2. Page rank score is calculated using the pagerank() function in networkx library

Distribution of page rank values for all the nodes



Observations:

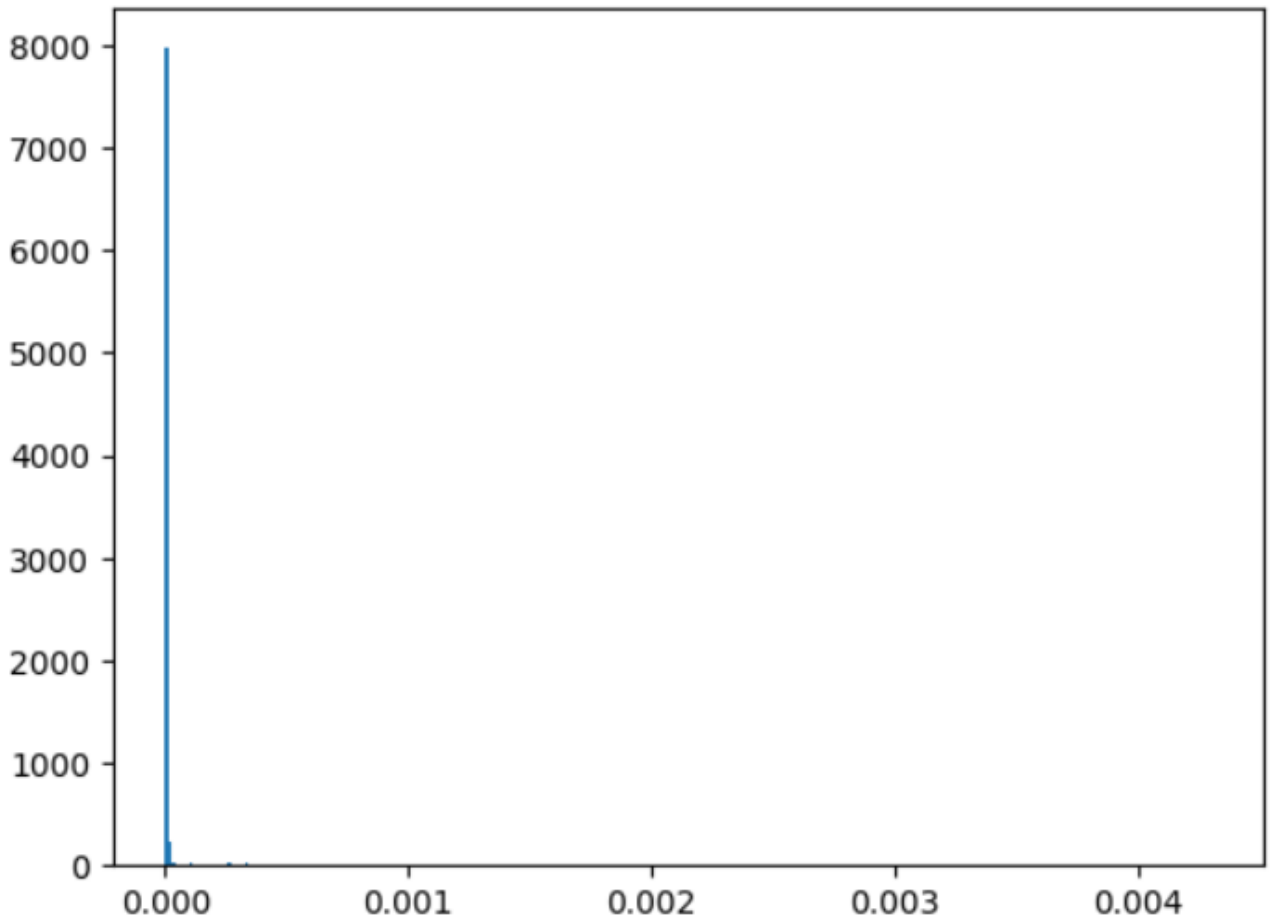
1. The page rank score lie within the range `( 6.989799880100975e-05 - 0.0010650431352832407 )`
2. Most of the page rank values are closer to 0

2. [15 points] Authority and Hub score for each node

Methodology:

1. An object of Digraph is created to represent a graph using networkx library
    - As the considered graph is a directed graph
2. Hub scores and Authority scores are calculated using the hits() function in networkx library

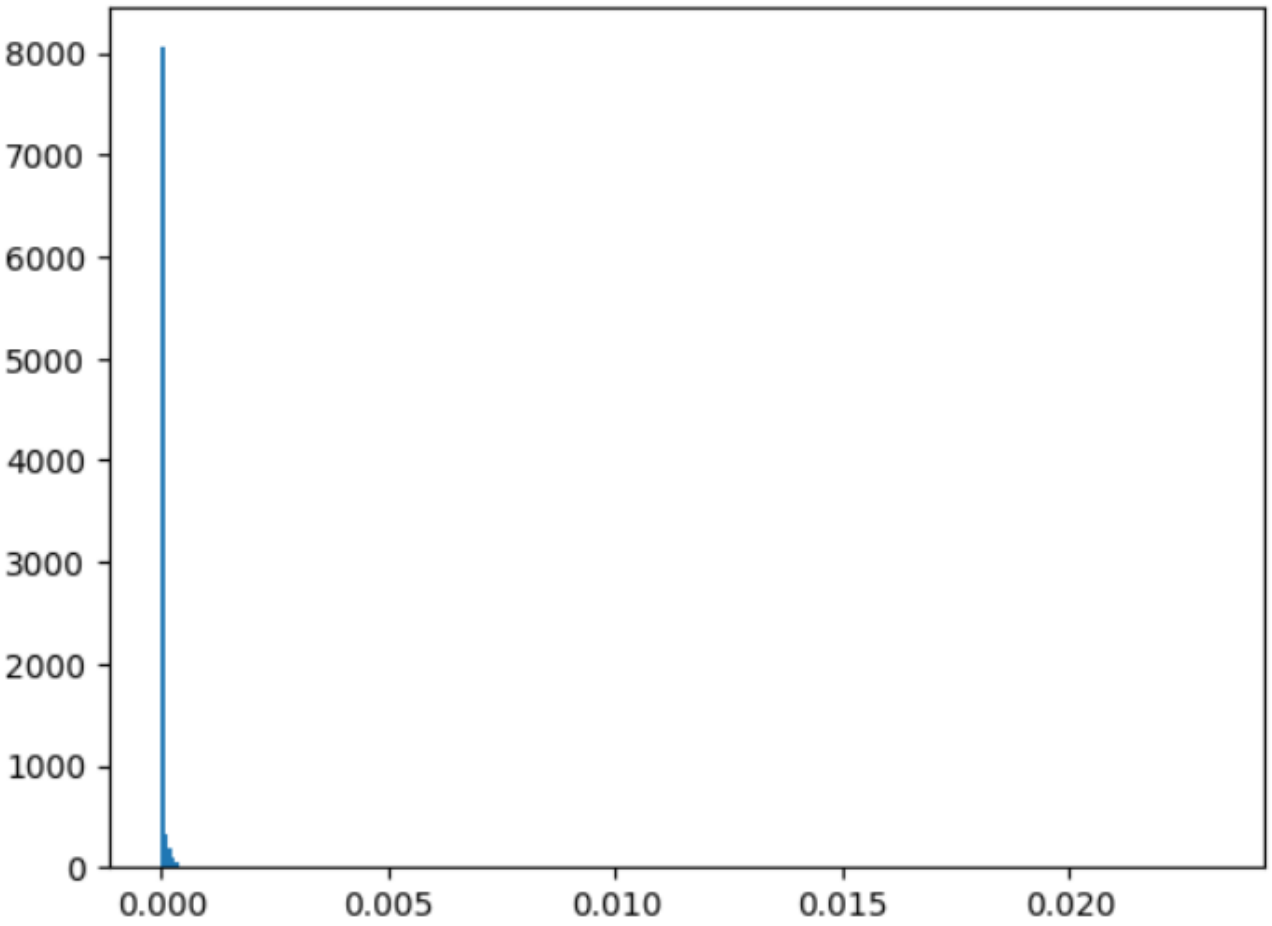
Distribution of Hub score values for all the nodes



Observations:

1. The page rank score lie within the range `( -1.9789912682262837e-20 - 0.004297779755596079 )`
2. Most of the Hub score values are closer to 0

Distribution of Authority score values for all the nodes

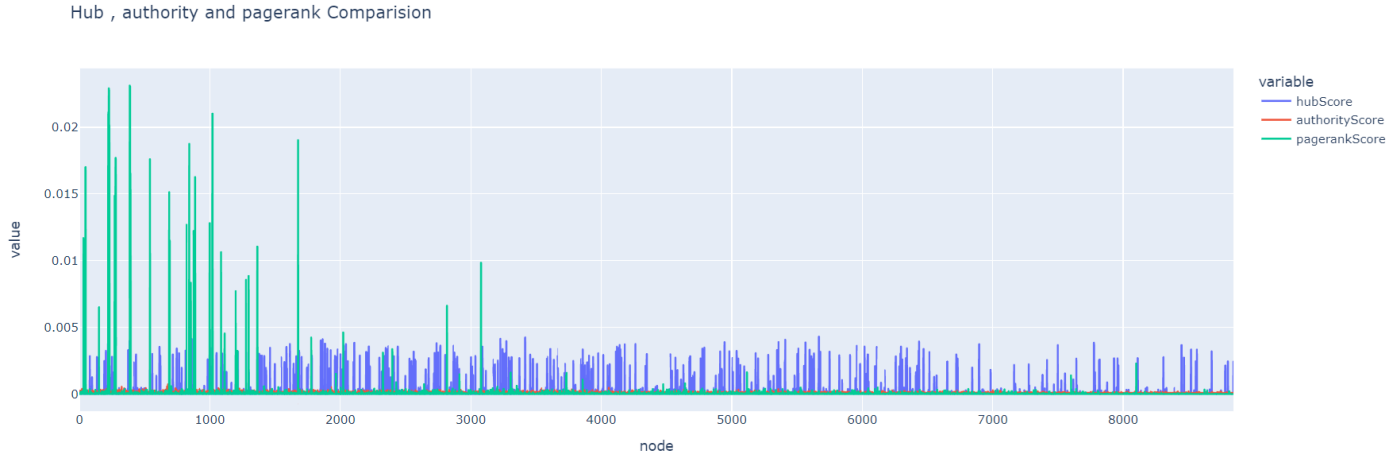


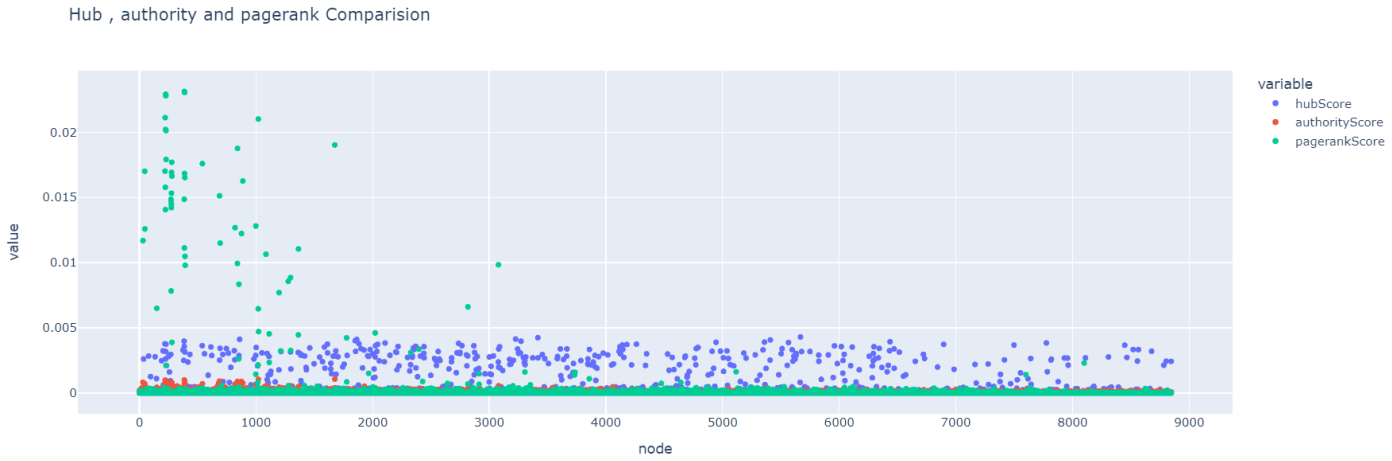
Observations:

1. The page rank score lie within the range `( -9.122141878126134e-19 - 0.023124000691889458 )`
2. Most of the Hub score values are closer to 0

3. [5 points] Compare the results obtained from both the algorithms in parts 1 and 2 based on the node scores.

The observed result was as follows:





Observations:

1. The range of page rank scores is much higher than authority score and hub score
2. The scores for node number 121 are

```
Page rank:  0.0001137378133544726 9
Hub score:  5.6514501031072 76e-06
Authority score:  2.1493047978181387e-05
```