Readme

CSE508: Information Retrieval Assignment 2

Question 1

Methodology:

1. Dataset taken: https://snap.stanford.edu/data/index.html
2. Represented the network in terms of its ‘adjacency matrix’ as well as ‘edge list’. Used the network to find out

[2 points] Represent the network in terms of its ‘adjacency matrix’ as well as ‘edge list’.
[28 points] Briefly describe the dataset chosen and report the following:
1. Number of Nodes
2. Number of Edges
3. Avg In-degree
4. Avg. Out-Degree
5. Node with Max In-degree
6. Node with Max out-degree
7. The density of the network
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 of the network.

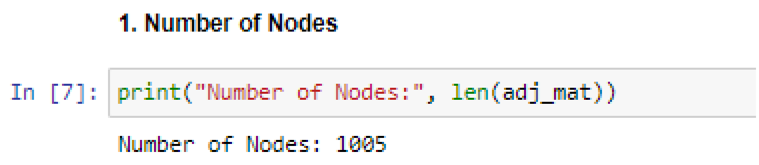
Preprocessing steps:
(i) Make the graph’s adjacency list
(ii) process all steps according to the given assignment.

Assumptions:

For the Local clustering coefficient, We take an undirected graph

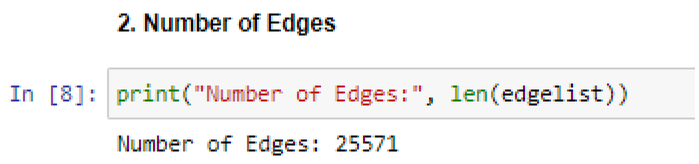
Results:

All the results are mentioned in the ipynb file itself.



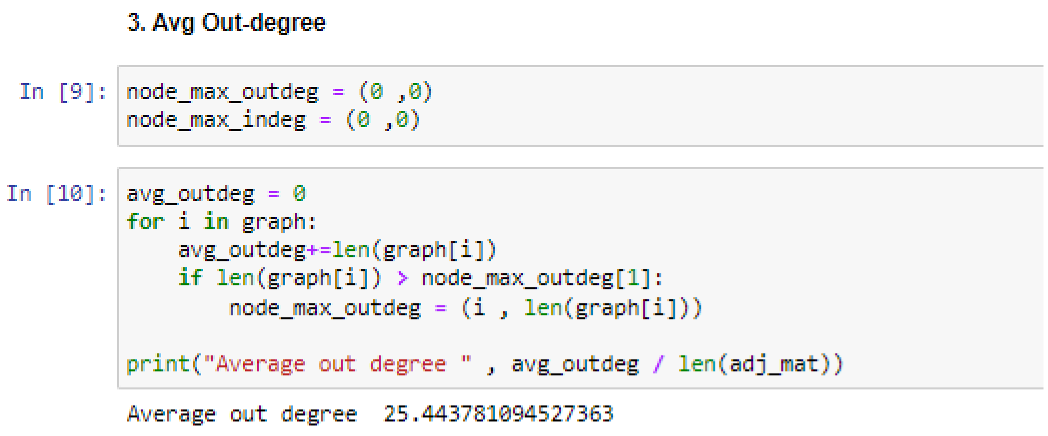
1. Number of Nodes

In [7]:
```
print("Number of Nodes:", len(adj_mat))
```
```
Number of Nodes: 1005
```



2. Number of Edges

In [8]:
```
print("Number of Edges:", len(edgelist))
```
```
Number of Edges: 25571
```

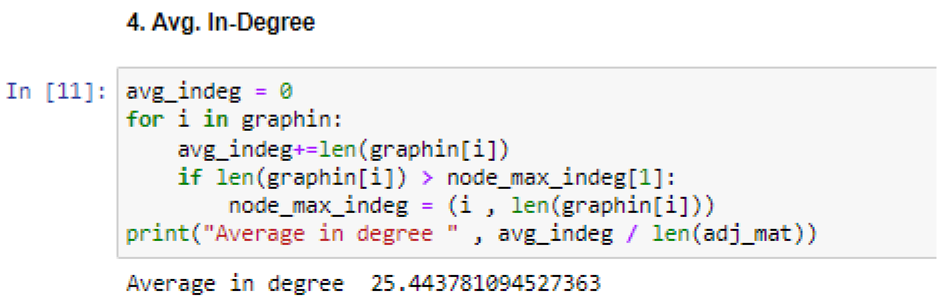


3. Avg Out-degree

In [9]:
```
node_max_outdeg = (0 ,0)
node_max_indeg = (0 ,0)
```

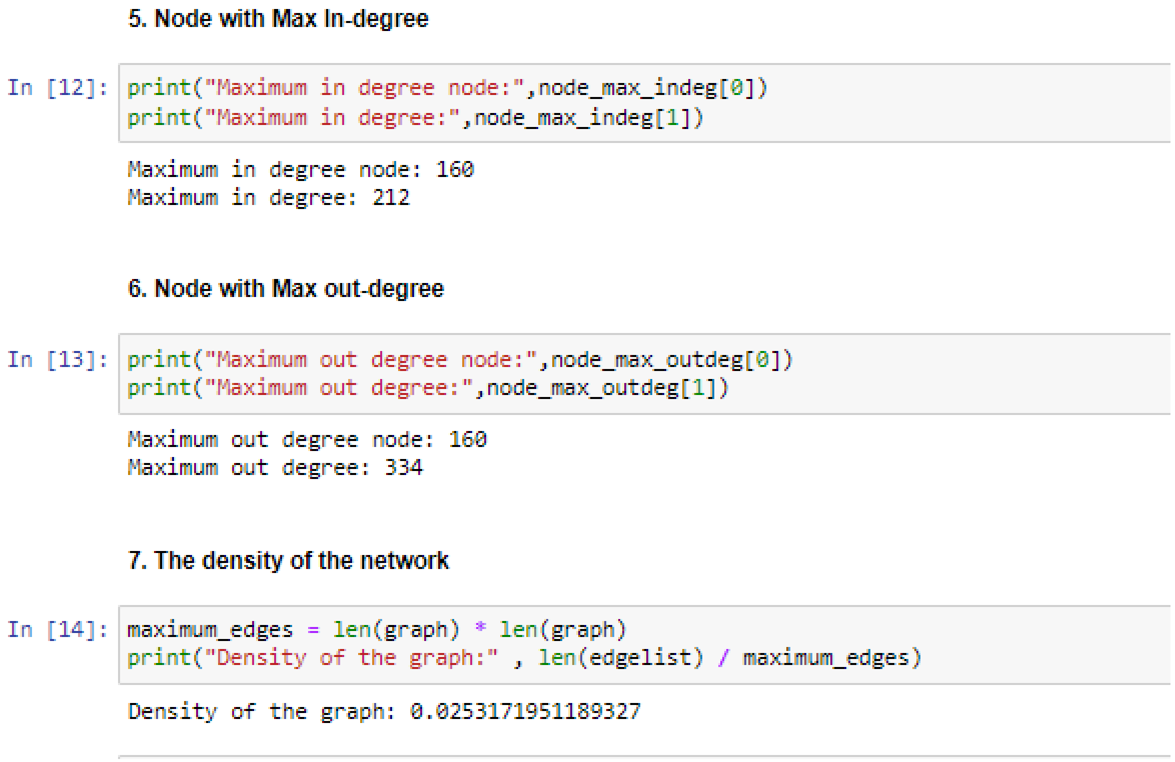
In [10]:
```
avg_outdeg = 0
for i in graph:
    avg_outdeg+=len(graph[i])
    if len(graph[i]) > node_max_outdeg[1]:
        node_max_outdeg = (i , len(graph[i]))

print("Average out degree " , avg_outdeg / len(adj_mat))
```
```
Average out degree  25.443781094527363
```



4. Avg. In-Degree

In [11]:
```
avg_indeg = 0
for i in graphin:
    avg_indeg+=len(graphin[i])
    if len(graphin[i]) > node_max_indeg[1]:
        node_max_indeg = (i , len(graphin[i]))
print("Average in degree " , avg_indeg / len(adj_mat))
```
```
Average in degree  25.443781094527363
```



5. Node with Max In-degree

In [12]:
```
print("Maximum in degree node:",node_max_indeg[0])
print("Maximum in degree:",node_max_indeg[1])
```

```
Maximum in degree node: 160
Maximum in degree: 212
```

6. Node with Max out-degree

In [13]:
```
print("Maximum out degree node:",node_max_outdeg[0])
print("Maximum out degree:",node_max_outdeg[1])
```

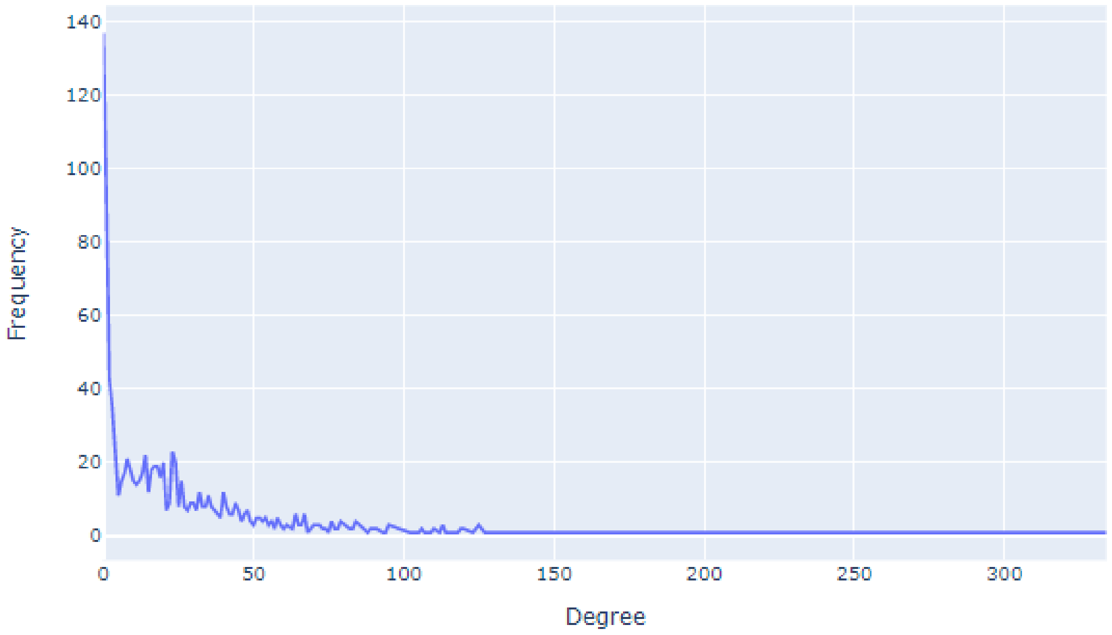
```
Maximum out degree node: 160
Maximum out degree: 334
```

7. The density of the network

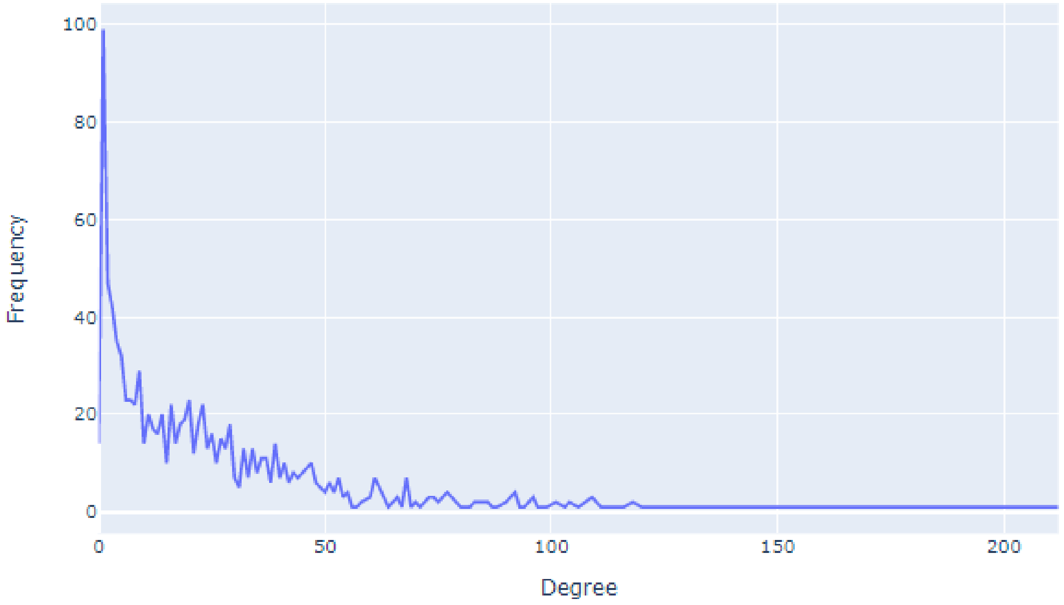
In [14]:
```
maximum_edges = len(graph) * len(graph)
print("Density of the graph:" , len(edgelist) / maximum_edges)
```

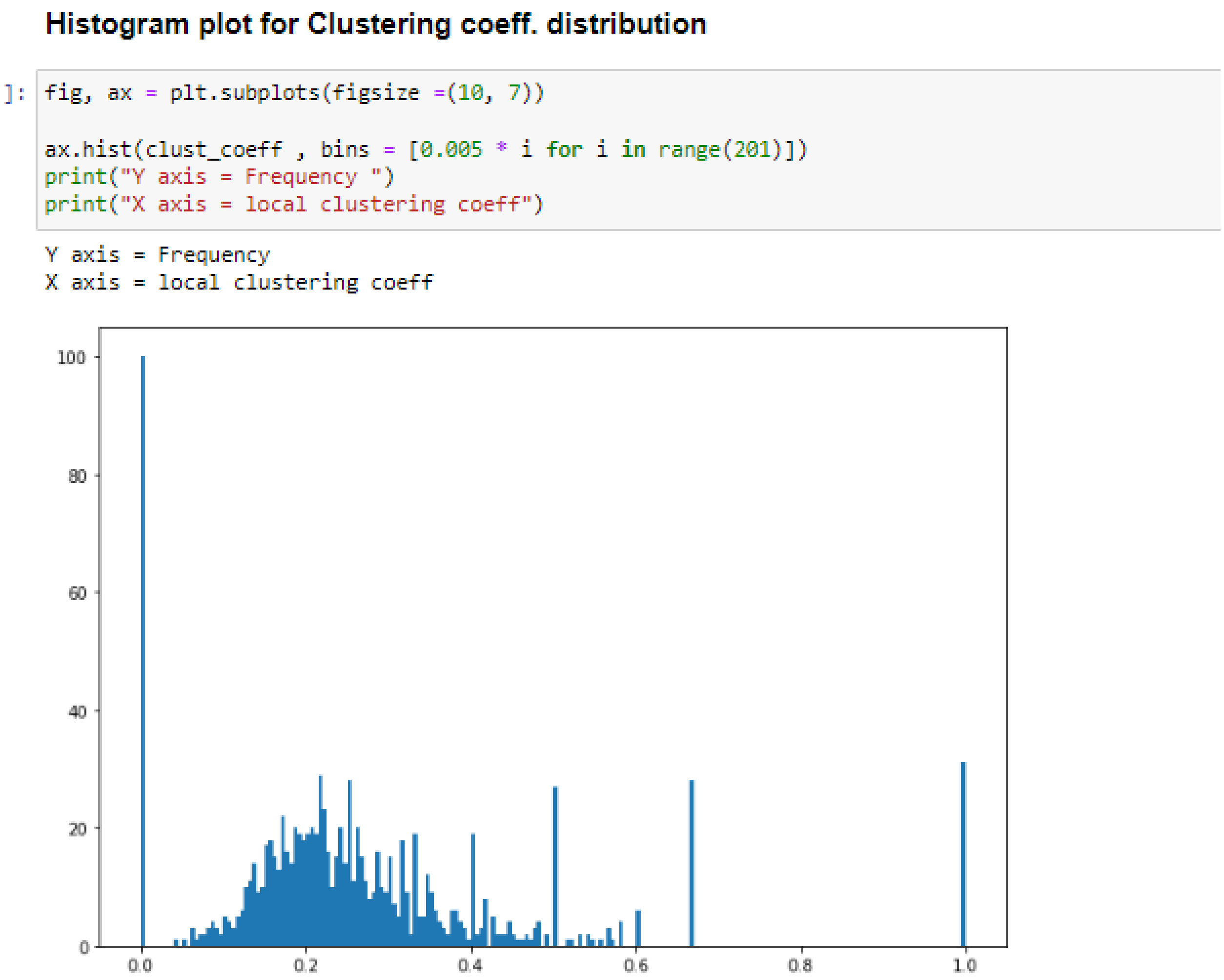
```
Density of the graph: 0.0253171951189327
```

OUT DEGREE DISTRIBUTION



IN DEGREE DISTRIBUTION



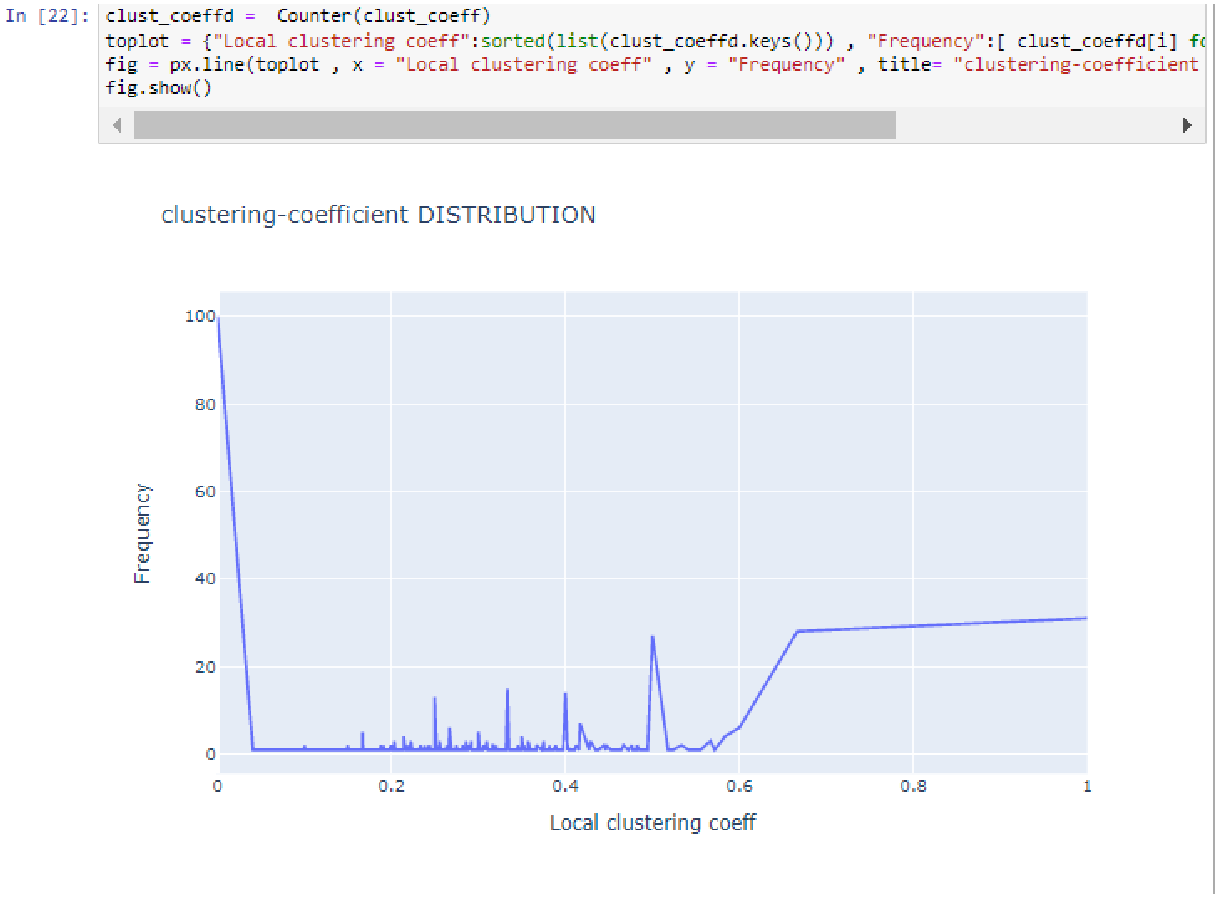


Histogram plot for Clustering coeff. distribution

```
]: fig, ax = plt.subplots(figsize =(10, 7))

   ax.hist(clust_coeff , bins = [0.005 * i for i in range(201)])
   print("Y axis = Frequency ")
   print("X axis = local clustering coeff")
```

```
Y axis = Frequency
X axis = local clustering coeff
```



```
In [22]: clust_coeffd =  Counter(clust_coeff)
         toplot = {"Local clustering coeff":sorted(list(clust_coeffd.keys())) , "Frequency":[ clust_coeffd[i] fo
         fig = px.line(toplot , x = "Local clustering coeff" , y = "Frequency" , title= "clustering-coefficient
         fig.show()
```

Question 2

Methodology:

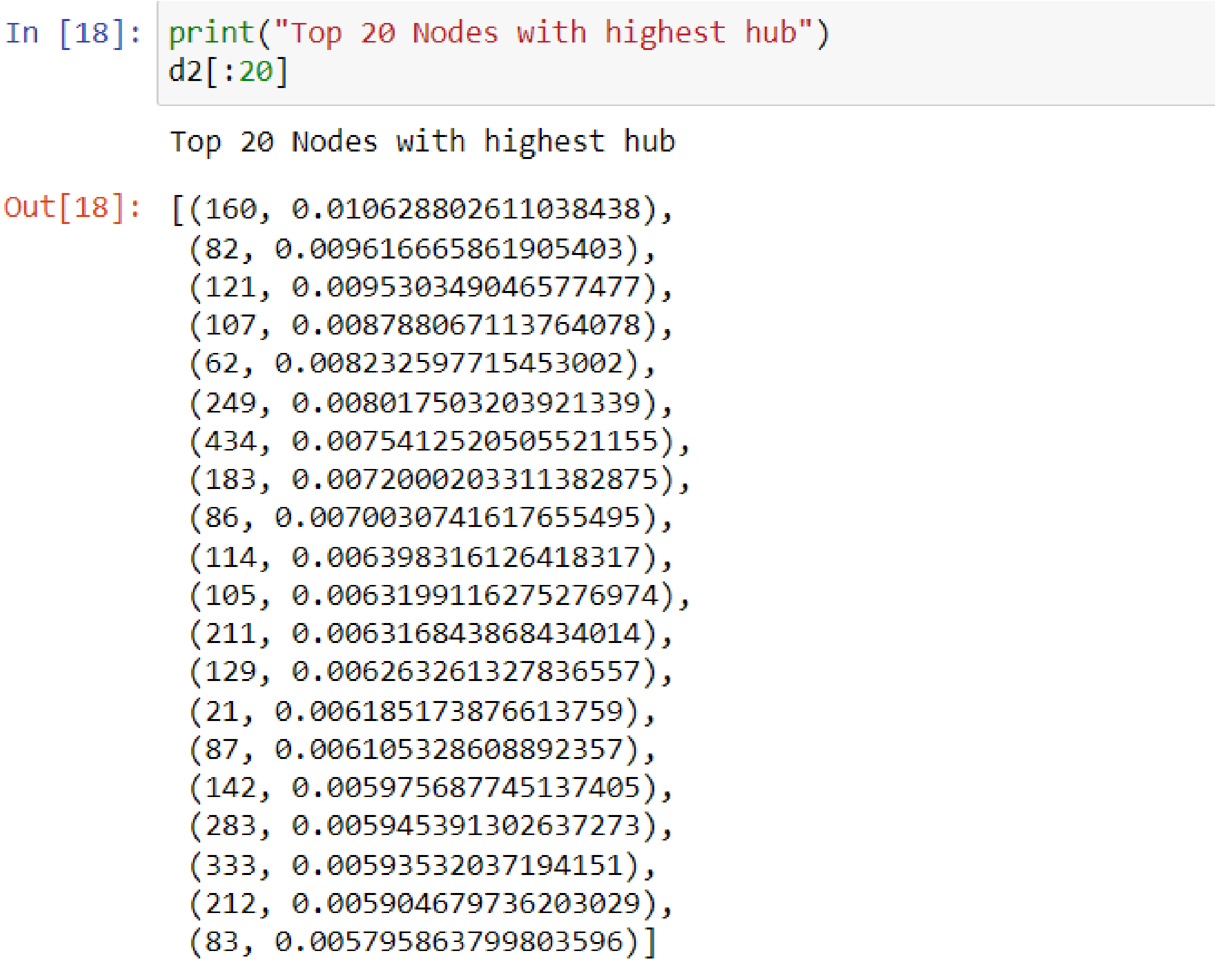
Follow the same steps as given in the assignment.

Calculated the PageRank score, and Authority and Hub score using the HITS algorithm for each node. Compared the results obtained from both the algorithms

For calculating PageRank, authority and hubs networkx library is used.

Analysis:

Top nodes with highest hub.

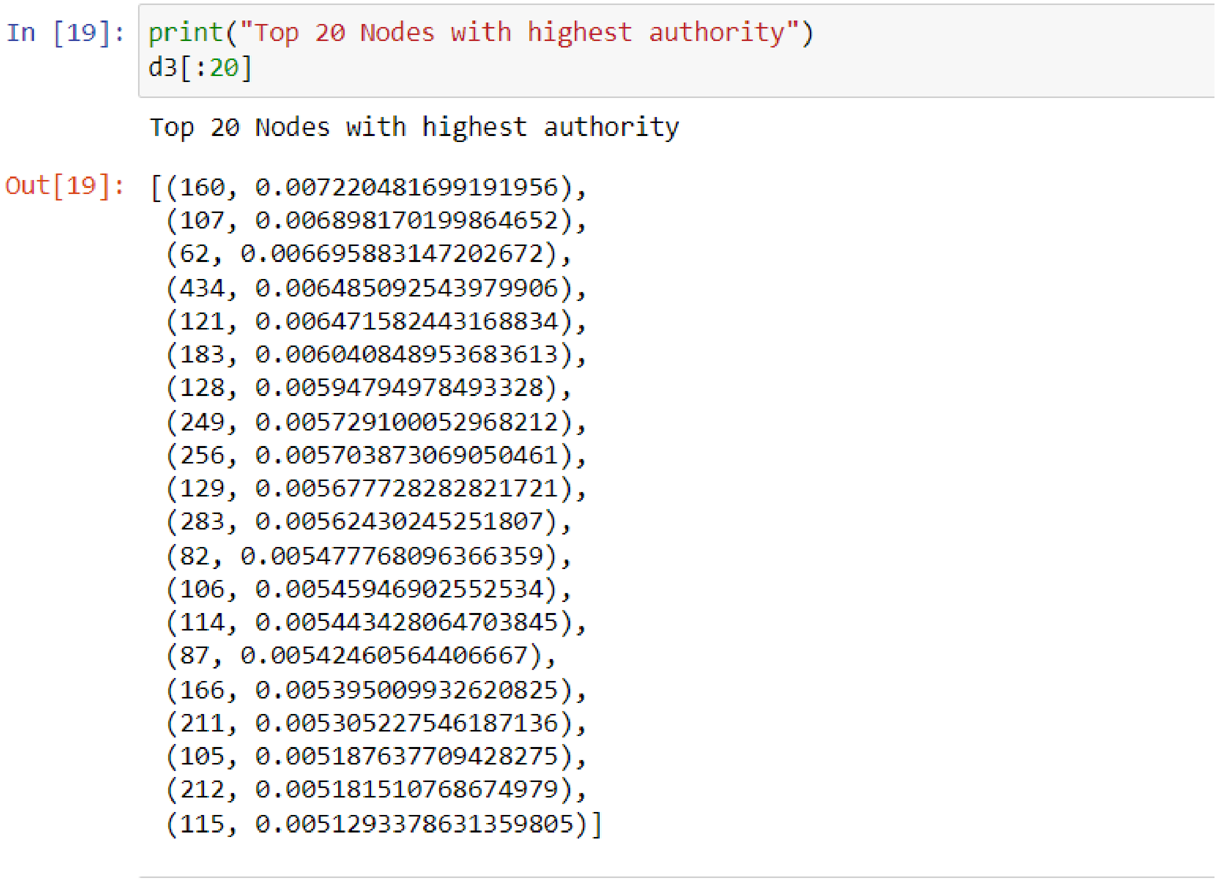


```
In [18]: print("Top 20 Nodes with highest hub")
         d2[:20]

         Top 20 Nodes with highest hub

Out[18]: [(160, 0.010628802611038438),
          (82, 0.009616665861905403),
          (121, 0.009530349046577477),
          (107, 0.008788067113764078),
          (62, 0.008232597715453002),
          (249, 0.008017503203921339),
          (434, 0.0075412520505521155),
          (183, 0.0072000203311382875),
          (86, 0.007003074161765495),
          (114, 0.006398316126418317),
          (105, 0.0063199116275276974),
          (211, 0.006316843868434014),
          (129, 0.006263261327836557),
          (21, 0.006185173876613759),
          (87, 0.006105328608892357),
          (142, 0.005975687745137405),
          (283, 0.005945391302637273),
          (333, 0.005935320371941510),
          (212, 0.005904679736203029),
          (83, 0.005795863799803596)]
```

Top nodes with the highest authority

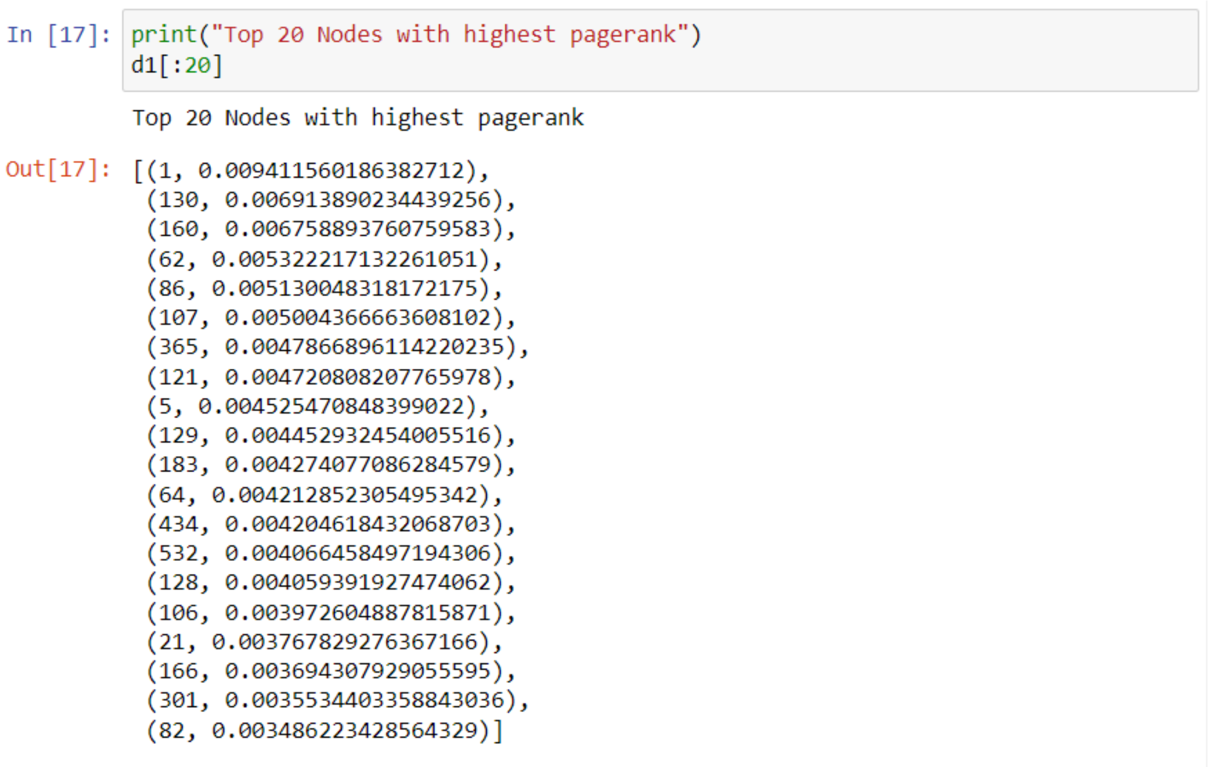


```
In [19]: print("Top 20 Nodes with highest authority")
         d3[:20]

         Top 20 Nodes with highest authority

Out[19]: [(160, 0.007220481699191956),
          (107, 0.006898170199864652),
          (62, 0.006695883147202672),
          (434, 0.006485092543979906),
          (121, 0.006471582443168834),
          (183, 0.006040848953683613),
          (128, 0.00594794978493328),
          (249, 0.005729100052968212),
          (256, 0.005703873069050461),
          (129, 0.005677728282821721),
          (283, 0.00562430245251807),
          (82, 0.005477768096366359),
          (106, 0.00545946902552534),
          (114, 0.005443428064703845),
          (87, 0.00542460564406667),
          (166, 0.005395009932620825),
          (211, 0.005305227546187136),
          (105, 0.005187637709428275),
          (212, 0.005181510768674979),
          (115, 0.0051293378631359805)]
```

Top nodes with highest pagerank



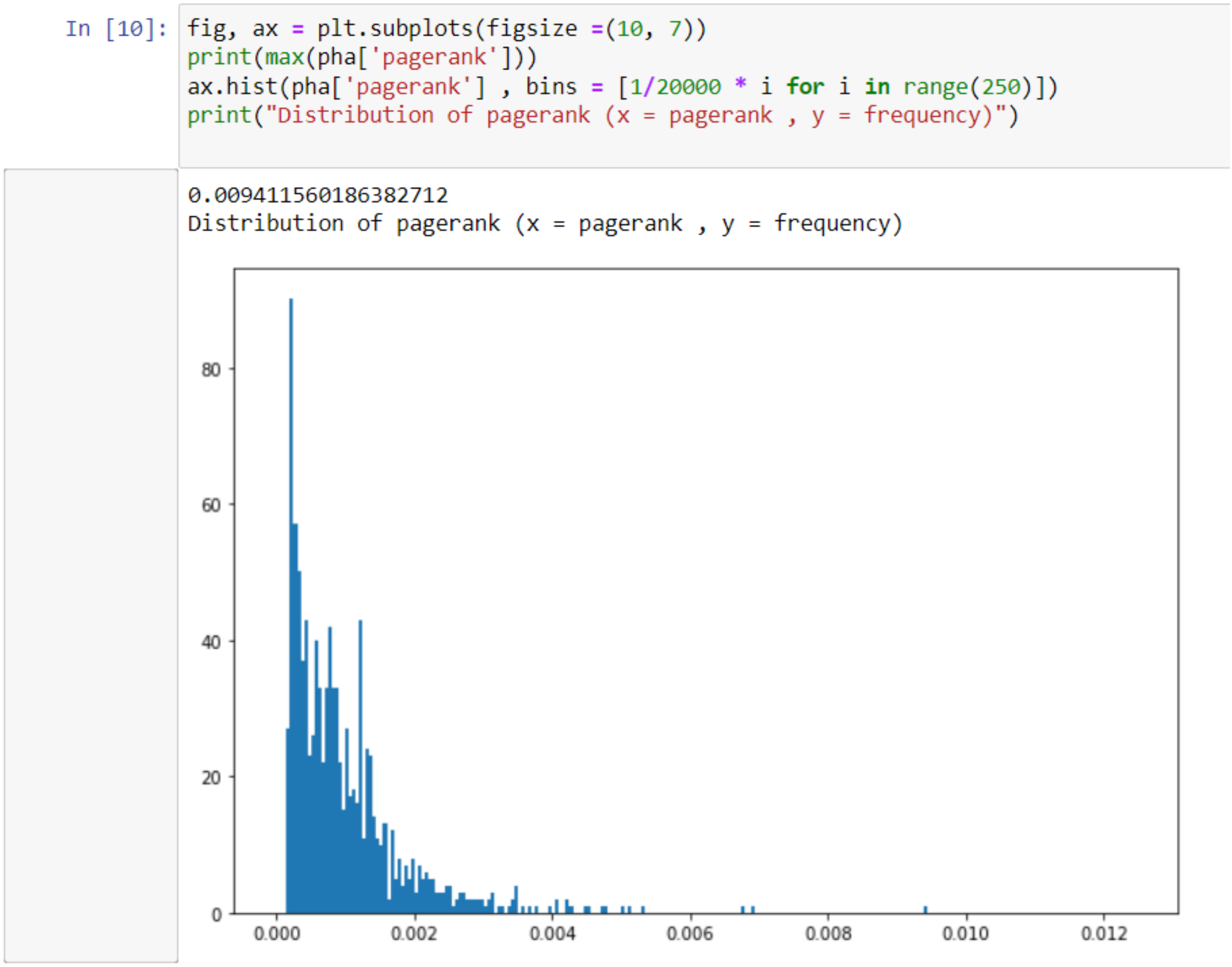
```
In [17]: print("Top 20 Nodes with highest pagerank")
         d1[:20]

         Top 20 Nodes with highest pagerank

Out[17]: [(1, 0.009411560186382712),
          (130, 0.006913890234439256),
          (160, 0.006758893760759583),
          (62, 0.005322217132261051),
          (86, 0.005130048318172175),
          (107, 0.005004366663608102),
          (365, 0.0047866896114220235),
          (121, 0.004720808207765978),
          (5, 0.004525470848399022),
          (129, 0.004452932454005516),
          (183, 0.004274077086284579),
          (64, 0.004212852305495342),
          (434, 0.004204618432068703),
          (532, 0.004066458497194306),
          (128, 0.004059391927474062),
          (106, 0.003972604887815871),
          (21, 0.003767829276367166),
          (166, 0.003694307929055595),
          (301, 0.0035534403358843036),
          (82, 0.003486223428564329)]
```

Graphs

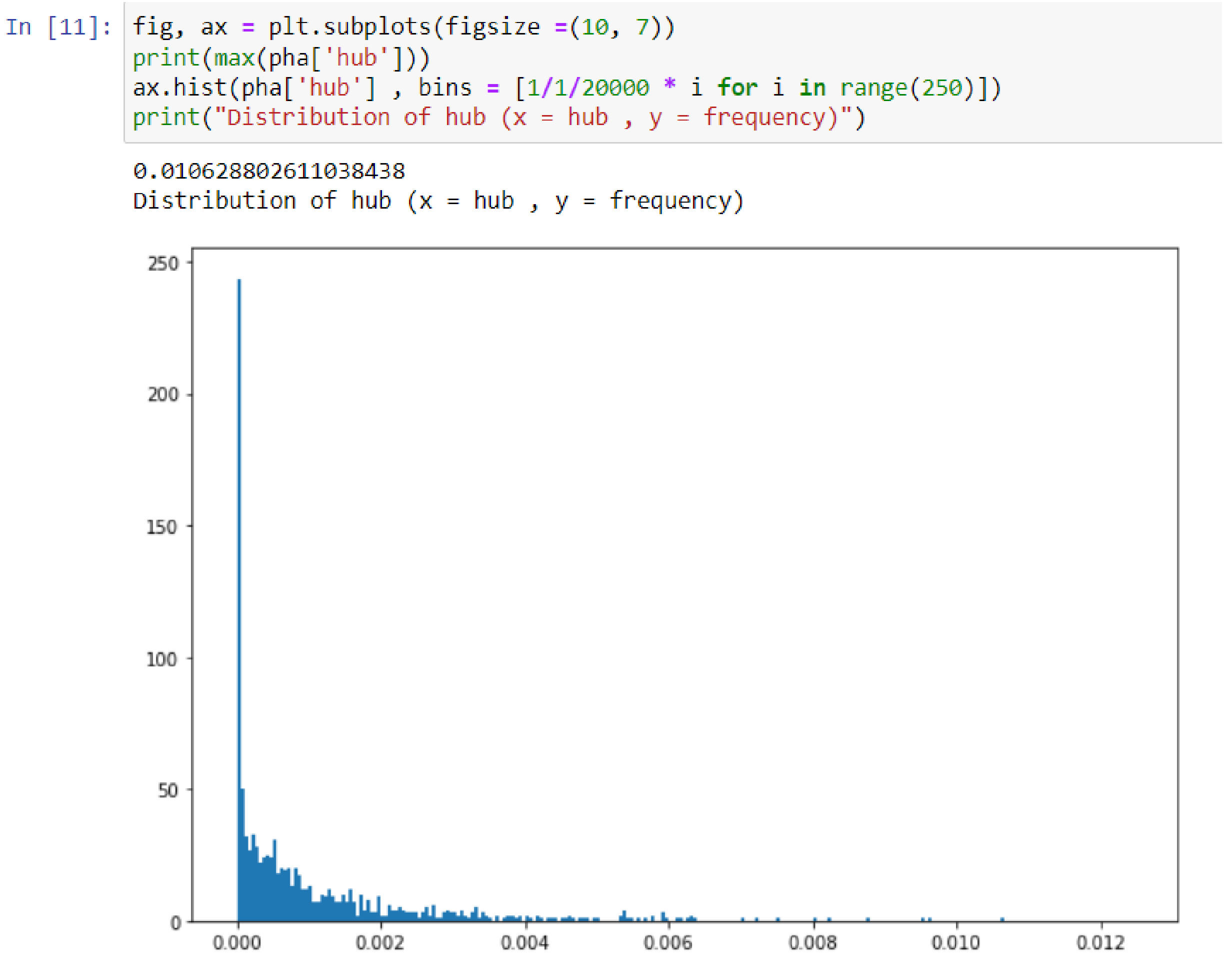
Distribution graph for Pagerank



```
In [10]: fig, ax = plt.subplots(figsize =(10, 7))
         print(max(pha['pagerank']))
         ax.hist(pha['pagerank'] , bins = [1/20000 * i for i in range(250)])
         print("Distribution of pagerank (x = pagerank , y = frequency)")
```

```
0.009411560186382712
Distribution of pagerank (x = pagerank , y = frequency)
```

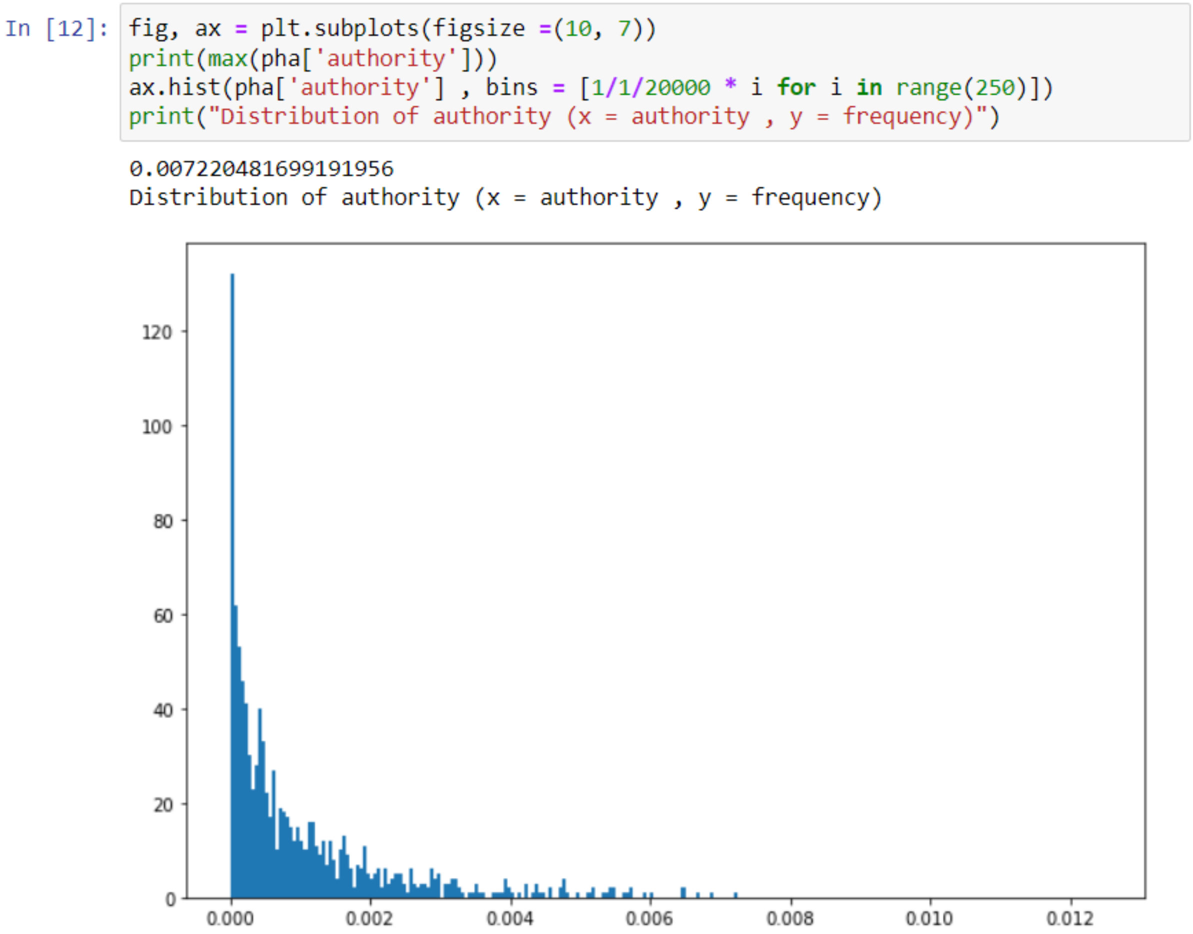
Distribution graph for hub



```
In [11]: fig, ax = plt.subplots(figsize =(10, 7))
         print(max(pha['hub']))
         ax.hist(pha['hub'] , bins = [1/1/20000 * i for i in range(250)])
         print("Distribution of hub (x = hub , y = frequency)")
```

```
0.010628802611038438
Distribution of hub (x = hub , y = frequency)
```

Distribution graph for authority



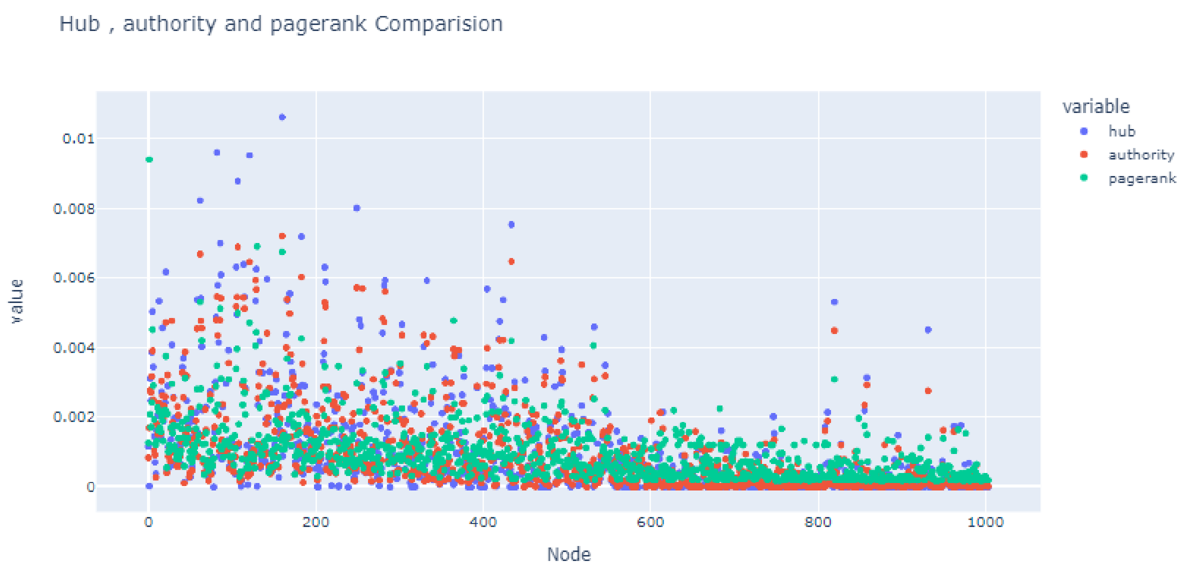
```
In [12]: fig, ax = plt.subplots(figsize =(10, 7))
         print(max(pha['authority']))
         ax.hist(pha['authority'] , bins = [1/1/20000 * i for i in range(250)])
         print("Distribution of authority (x = authority , y = frequency)")
```

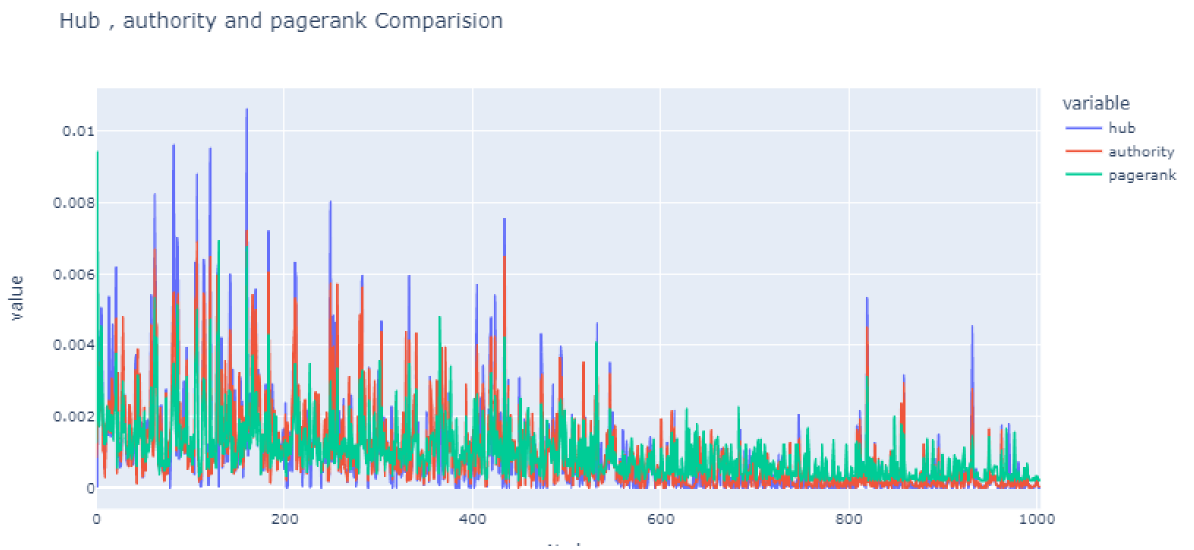
```
0.007220481699191956
Distribution of authority (x = authority , y = frequency)
```

Observation:-

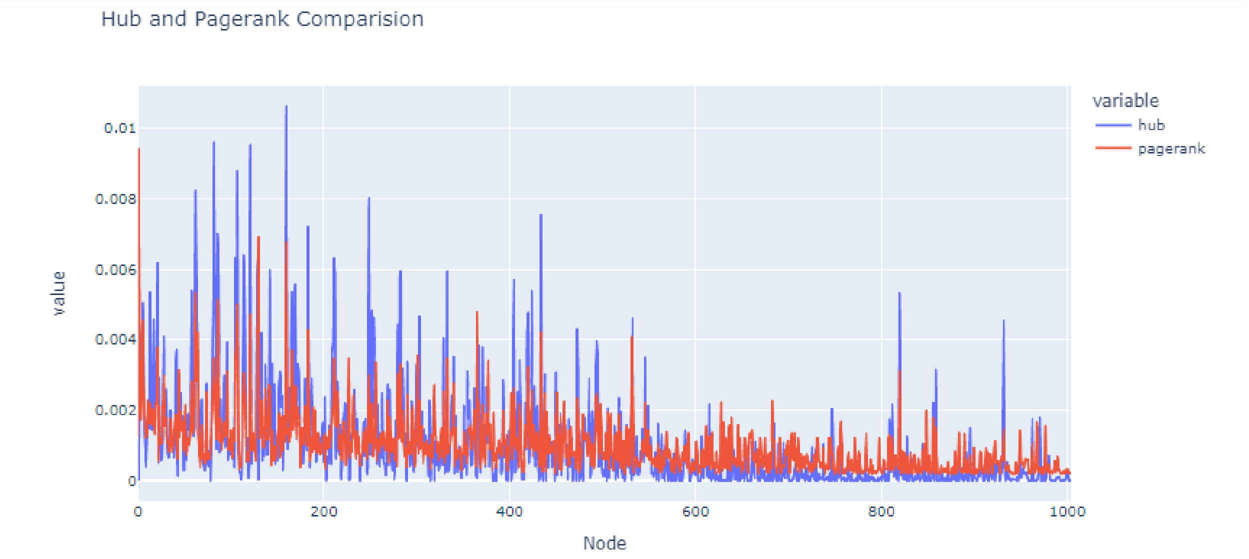
- Hub has more 0 numbered value nodes and the distribution graph falls faster than any other two(PageRank and authority) of the graphs.
- The Hub distribution graph has the longest tail.
- The Hub distribution graph has the shortest tail.
- The maximum value of the hub is 0.010628802611038438
- The maximum value of the PageRank is 0.009411560186382712
- The maximum value of the Authority is 0.007220481699191956

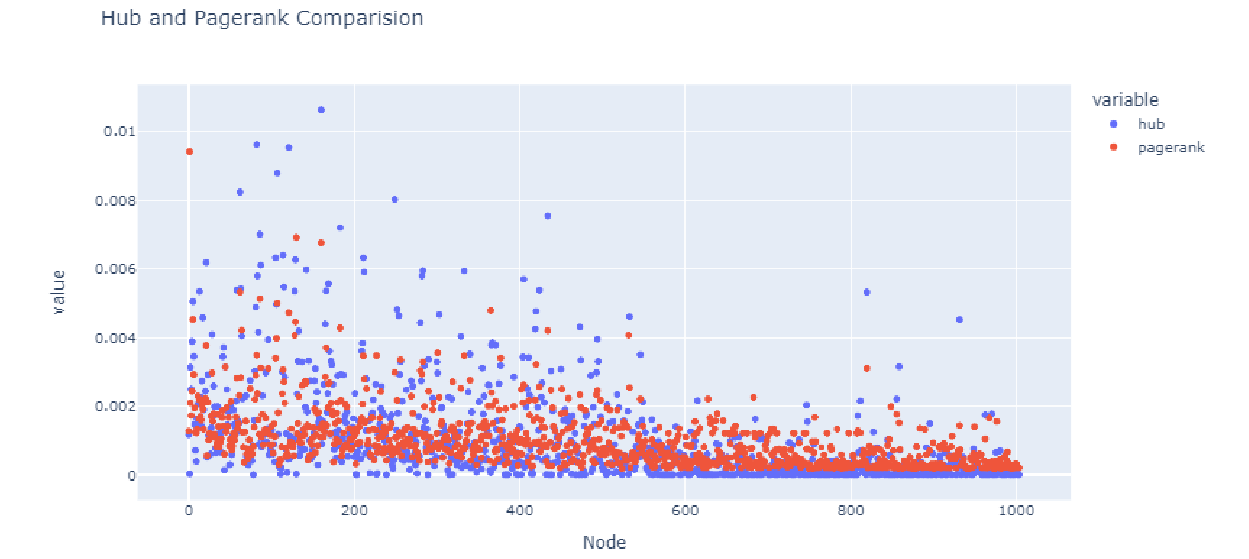
Comparison graphs(both line and scatter):



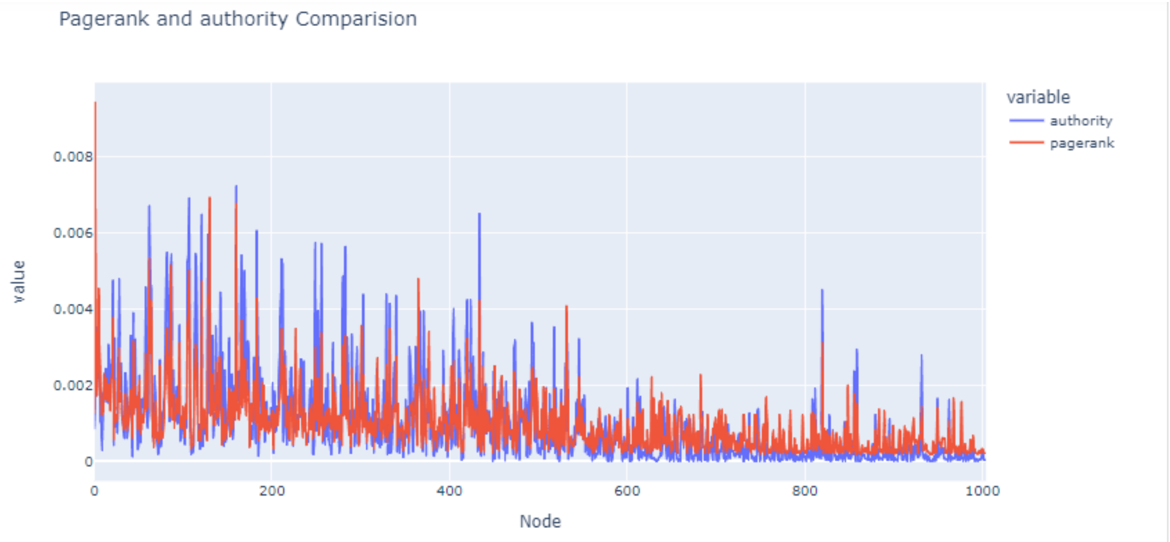


Pairwise Comparision graph (line and scatter):-

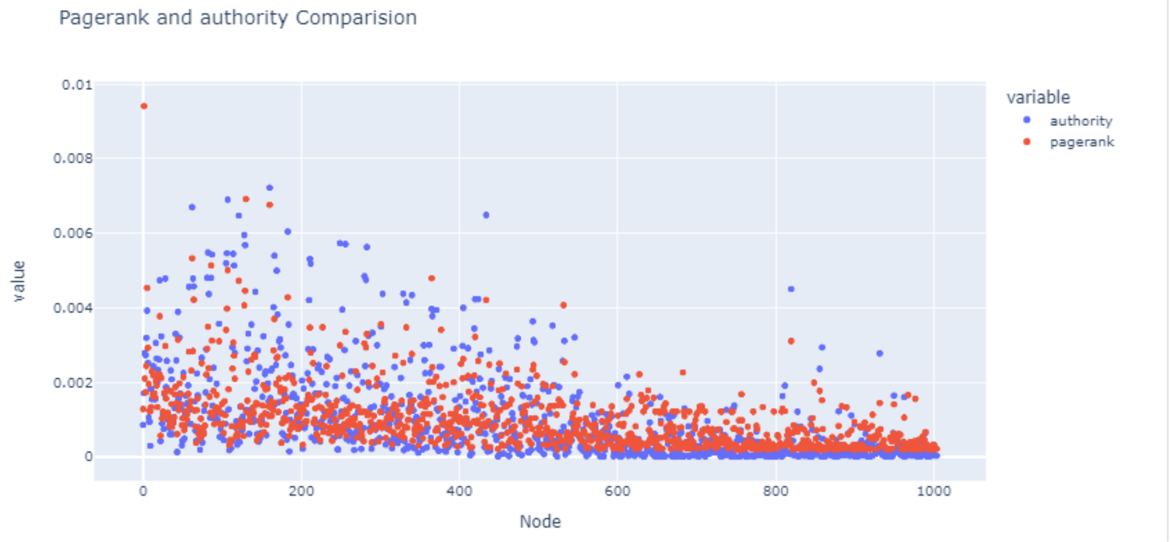


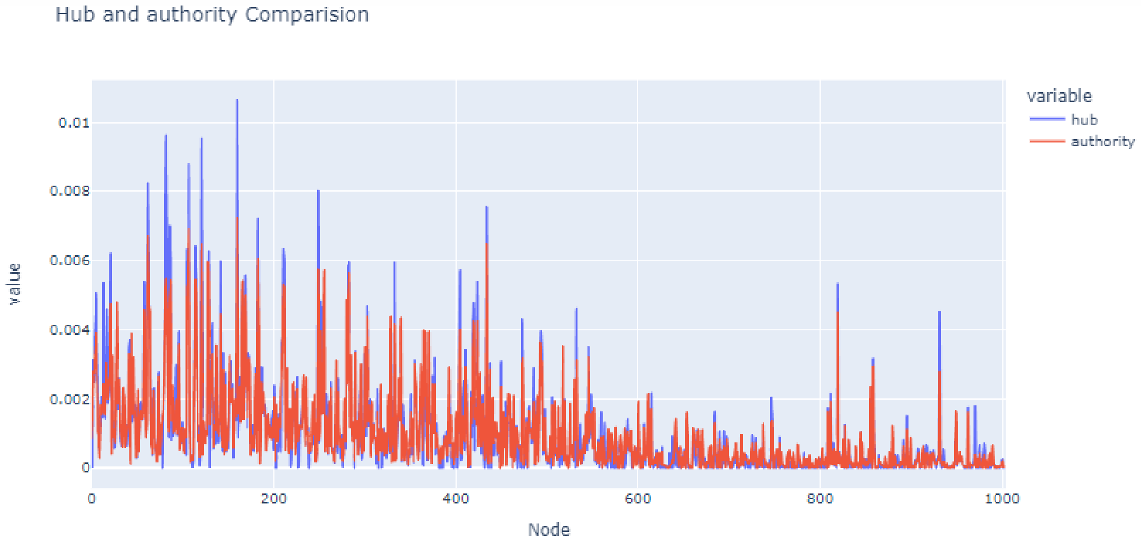


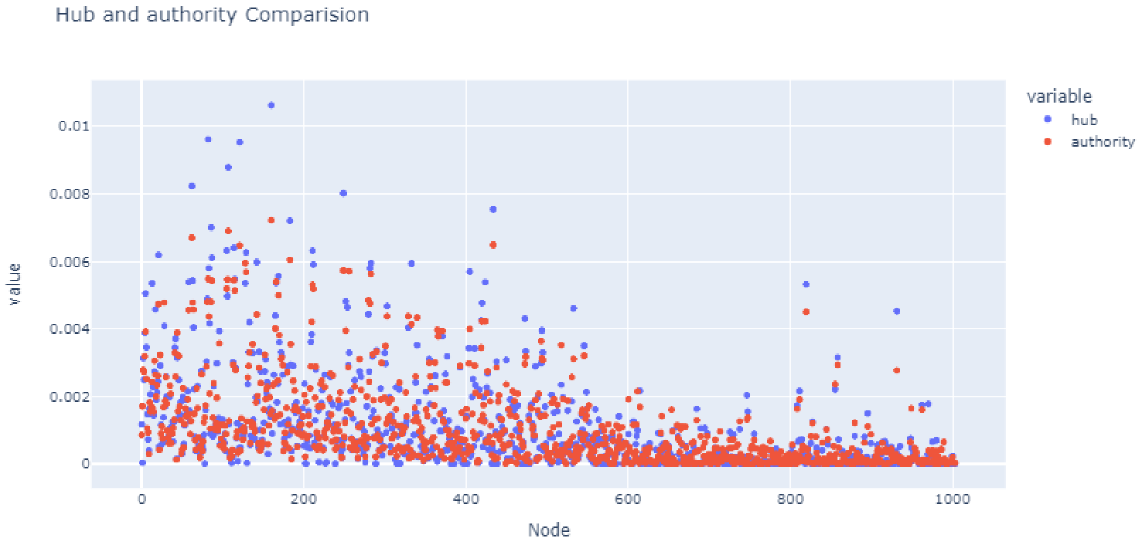
Pagerank and authority Comparision



Pagerank and authority Comparision




Hub and authority Comparision
variable
hub
authority
value
0.01
0.008
0.006
0.004
0.002
0
0
200
400
600
800
1000
Node


Hub and authority Comparision
variable
hub
authority
value
0.01
0.008
0.006
0.004
0.002
0
0
200
400
600
800
1000
Node