REPORT

Ranit Pal(MT22119)

Chitransh Bose(MT22096)

METHODOLOGY:

- **ADJACENCY MATRIX**

```
A = np.zeros((numnodes, numnodes))
for row in df.itertuples(index=False):
    start_node = row[0]
    end_node = row[1]
    A[start_node][end_node] = 1
```

FOR Adjacency matrix we are assigning row[0] value to the start node and we are also assigning row[1] value to the end_node and this is doing for the whole data-frame .

- **IN DEGREE AND OUT DEGREE**

```
for edge in  g_edges:
    out_node = edge[0]
    in_node = edge[1]
    out_deg[out_node] += 1
    inside_deg[in_node] += 1
```

For in and out degree we are iterating over the all edges and the in the case of on node we are assigning edge[1] value and for the out node we are assigning the edge[0] value and finally adding them iteratively.

- **CODE**

```
maximum_inside_degree = 0
mx_i_deg = 0
maximum_outside_degree = 0
mx_o_deg = 0

for i in range(numnodes):
    in_degree_index = 0
    out_degree_i = 0
    for j in range(numnodes):
        in_degree_index += A[j][i]
        out_degree_i += A[i][j]
    if in_degree_index > mx_i_deg:
        maximum_inside_degree = i
        mx_i_deg = in_degree_index
    if out_degree_i > mx_o_deg:
        maximum_outside_degree = i
        mx_o_deg = out_degree_i

nw_density = num_e / (numnodes * (numnodes - 1))
```

Here we are initializing the max in and out degree then walking over all the nodes and and calculating one by one and adding the values in the adjacency matrix and finally printing the out put in the next cell.

- **CODE**

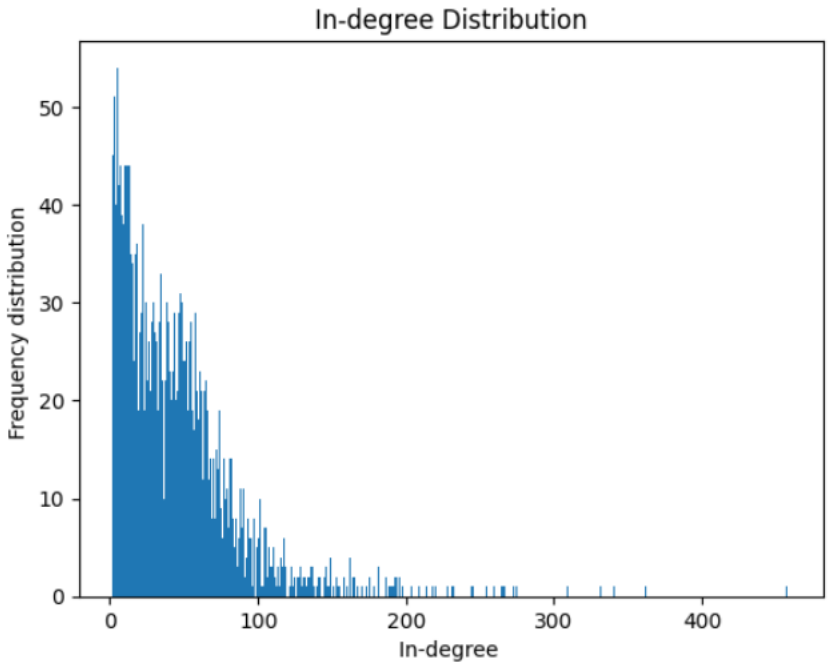
```
s_in_deg = 0
s_out_deg = 0
for i in range(numnodes):
    s_in_deg += inside_deg[i]
    s_out_deg += out_deg[i]

average_in_degree = s_in_deg / numnodes
average_out_degree = s_out_deg / numnodes


maximum_inside_degree = 0
mx_i_deg = 0
maximum_outside_degree = 0
mx_o_deg = 0

for i in range(numnodes):
    in_degree_index = 0
    out_degree_i = 0
    for j in range(numnodes):
        in_degree_index += A[j][i]
        out_degree_i += A[i][j]
    if in_degree_index > mx_i_deg:
        maximum_inside_degree = i
        mx_i_deg = in_degree_index
    if out_degree_i > mx_o_deg:
        maximum_outside_degree = i
        mx_o_deg = out_degree_i

nw_density = num_e / (numnodes * (numnodes - 1))
```

- **RESULTS**

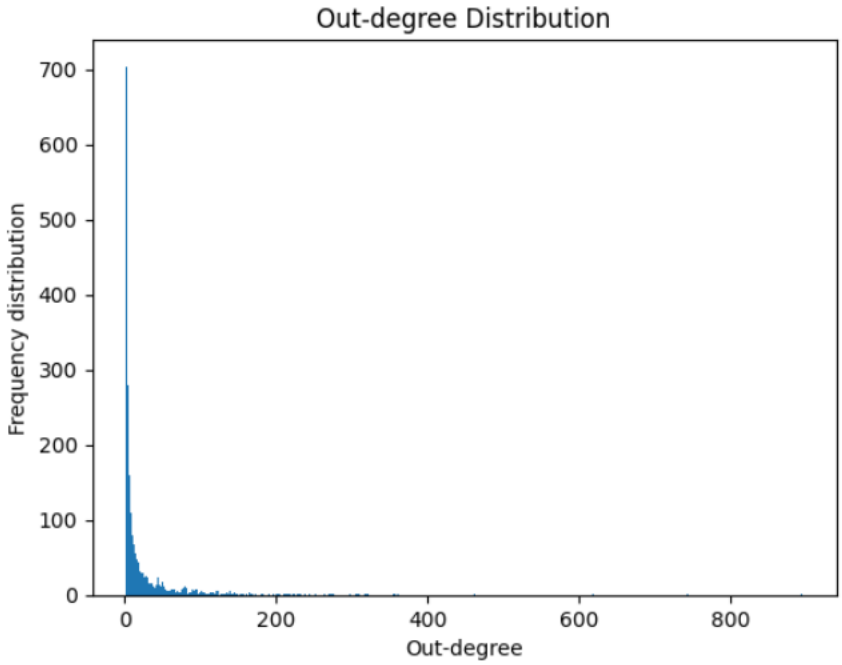
```
NUMBER OF NODES 8298
NUMBER OF EDGES: 103689
AVERAGE IN DEGREE: 12.495661605206074
AVERAGE OUT DEGREE: 12.495661605206074
NODE WITH MAX IN DEGREE: 4037
NODE WITH MAX OUT DEGREE: 2565
NETWORK DENSITY: 0.0015060457521039019
```

- **DIAGRAM**

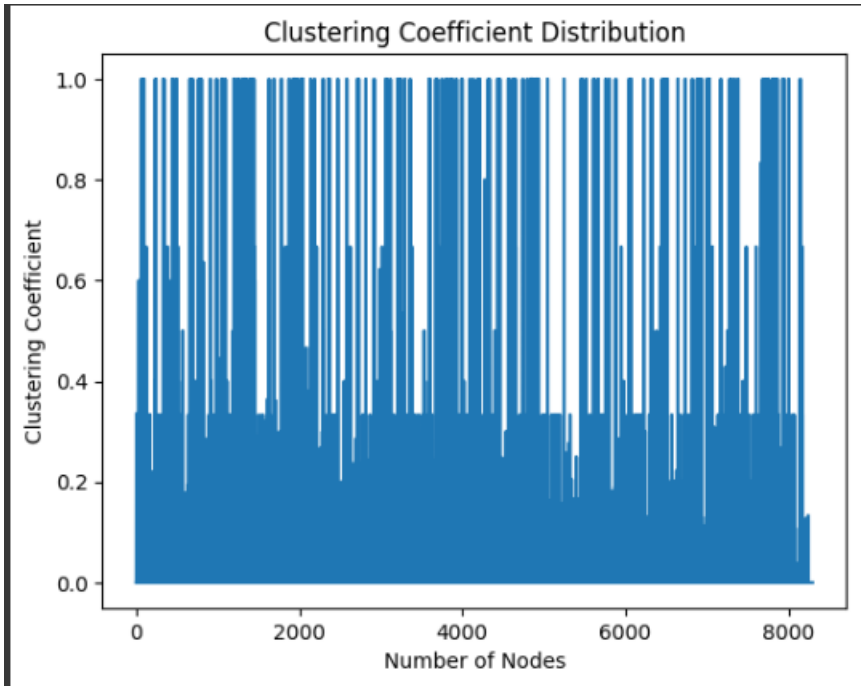
IN DEGREE DISTRIBUTION



OUT DEGREE DISTRIBUTION :



DISTRIBUTION OF CLUSTERING COFFICIENT:



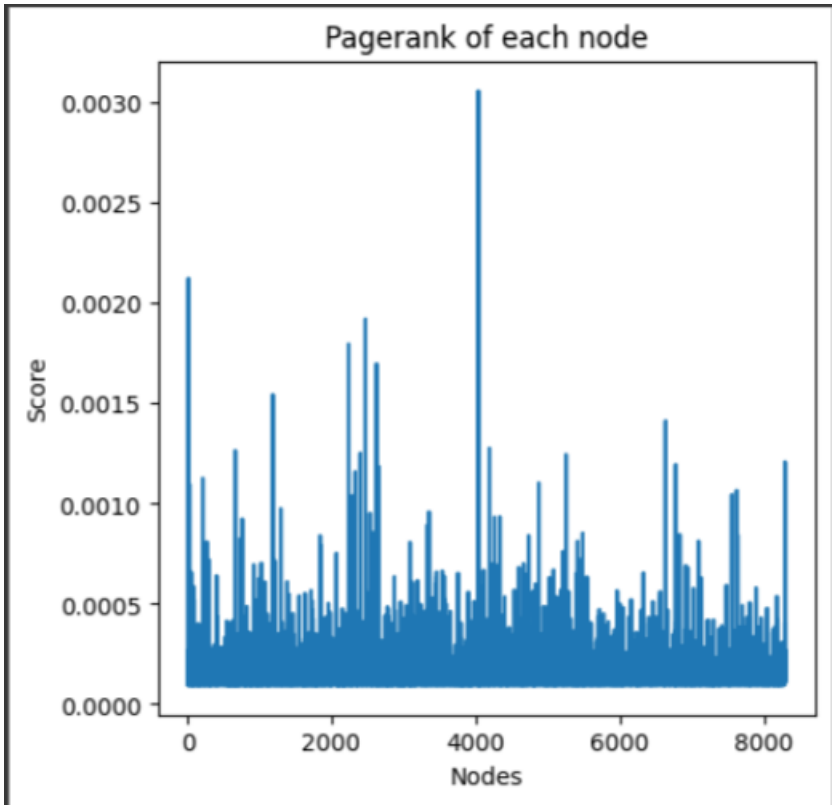
Ques 2.
Assumption: Graphs are directed and have atleast 100 nodes.

Methodology:

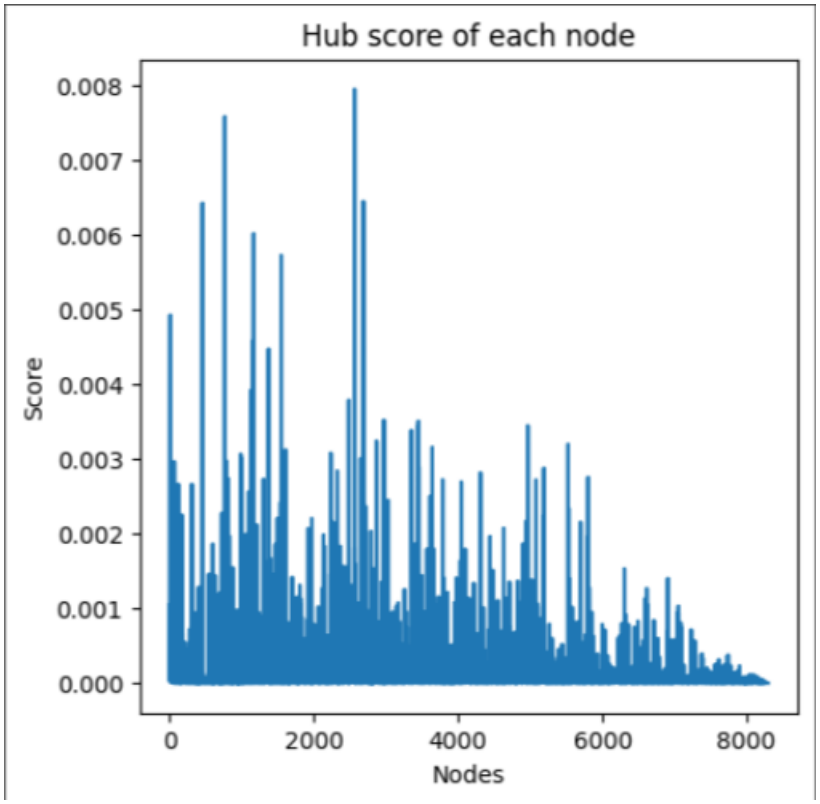
1. Create an empty graph first and as per the dataset add a node from FromNodeId to ToNodeId.
2. Calculate the PageRank on the graph.
3. Calculate hub score and authority score using HITS algorithm.
4. Compare the results.

Results:

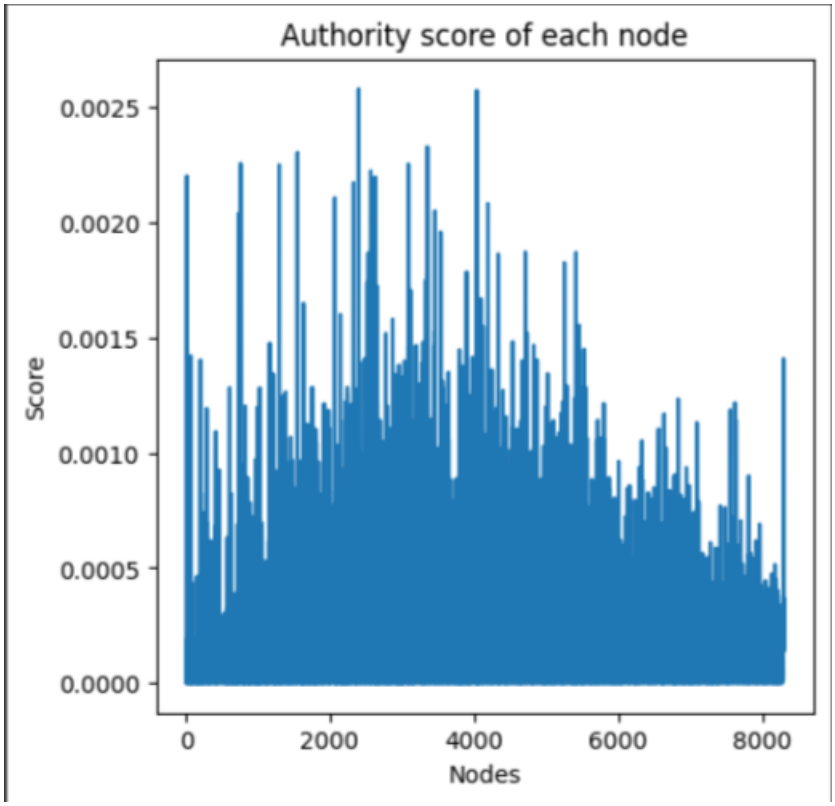
1. Pagerank of each node.



2. Hub score of each node.



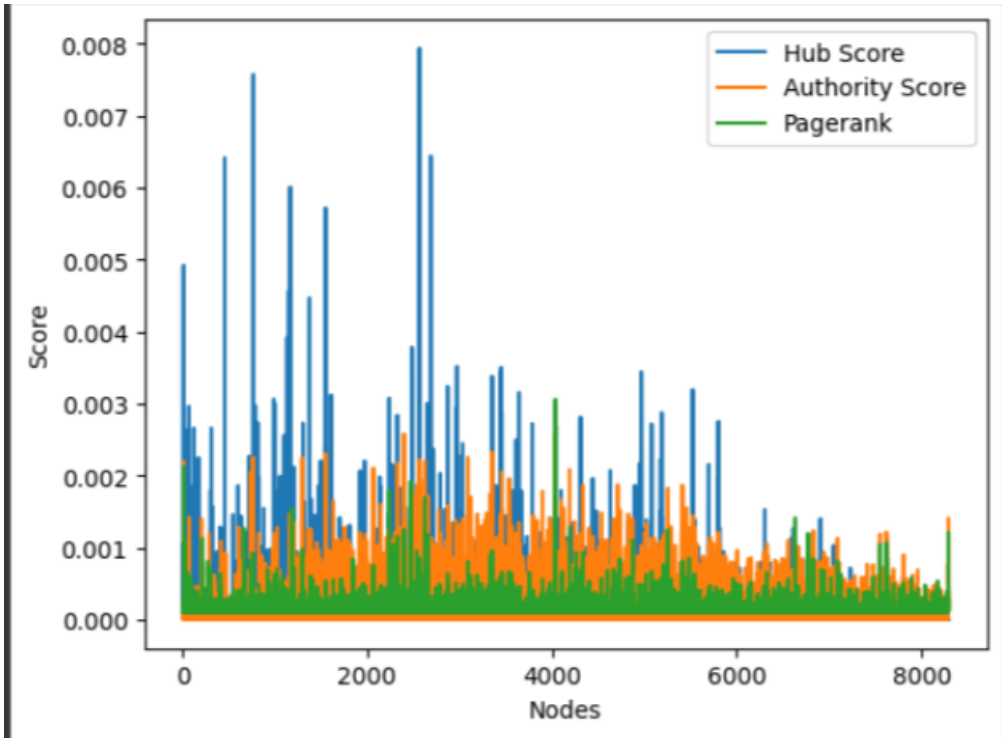
3. Authority score of each node.



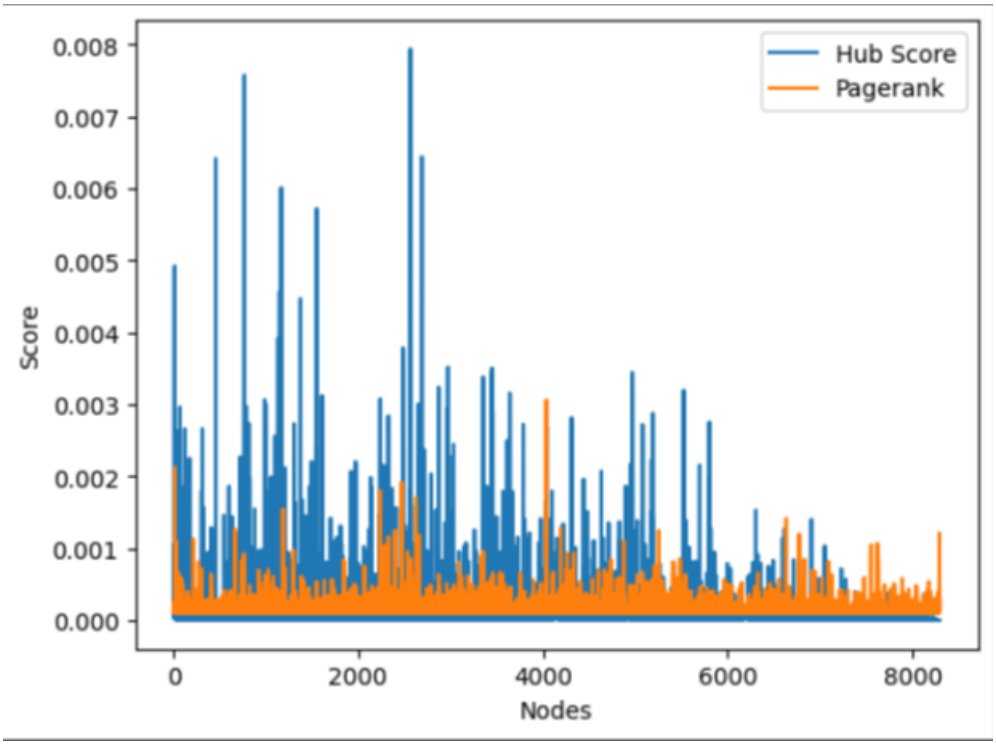
4. Comparison of different scores numerically.

	Nodes	Pagerank	Hub Score	Authority Score
0	3	0.000168	0.000040	9.501172e-05
1	4	0.000095	0.000073	3.937246e-20
2	5	0.000095	0.000035	2.581388e-20
3	6	0.000245	0.001054	6.398066e-05
4	7	0.000095	0.000082	-1.280203e-19
...	...	...	...	...
7110	8293	0.001209	-0.000000	1.408707e-03
7111	8294	0.000422	-0.000000	1.121717e-03
7112	8295	0.000428	-0.000000	1.093686e-03
7113	8296	0.000109	-0.000000	1.374990e-04
7114	8297	0.000265	-0.000000	3.649261e-04

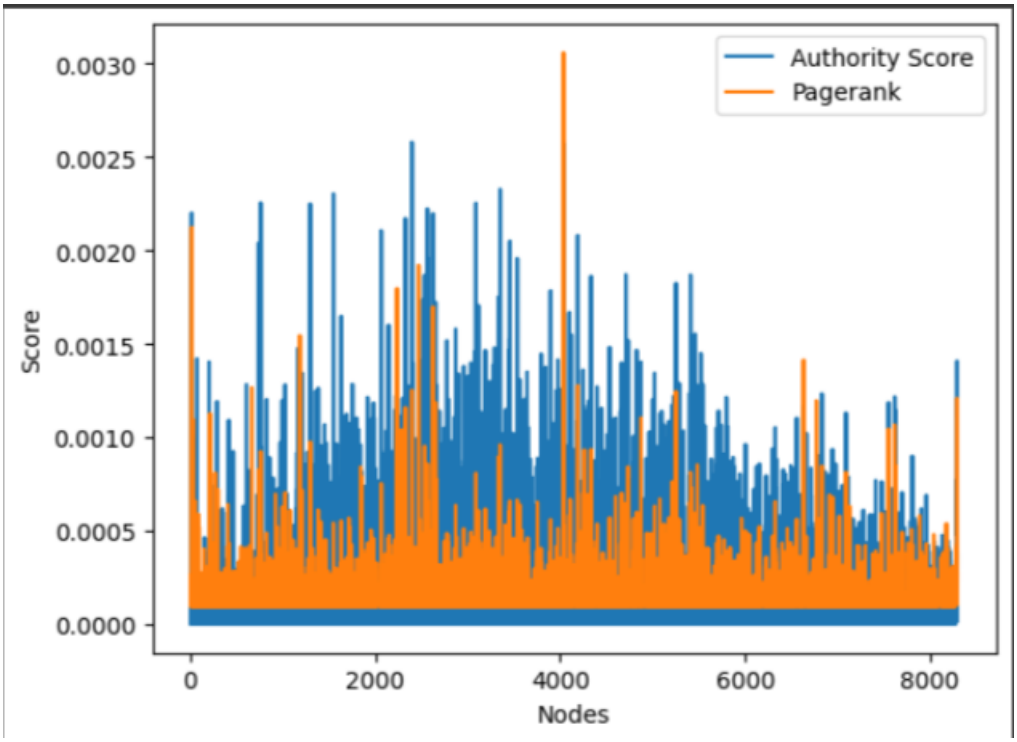
5. Comparison of different scores graphically.
    a. Comparison between pageRank, hub score and authority score.



b. Comparison of pagerank and hub score.



c. Comparison of pagerank and authority score.



There is a difference in all these score because:
1. PageRank rank pages based on the structure of the incoming links whereas hub is calculated based on the outgoing links and authority based on the incoming links.

2. PageRank uses probability to visit random pages to rank the pages whereas HITS uses hub and authority score. Hub contains information about how many links are there that contain relevant information and authority contribute to the pages that actually contain content.

References:

1. GeeksforGeeks
2. StackOverflow