IR Assignment 3 Report

Question 1

The dataset name is “wiki-vote” and the link is provided below:

https://snap.stanford.edu/data/wiki-Vote.html

which consists of 7515 nodes and 103689 edges.

After downloading it we converted it into csv file by removing some unnecessary lines.

Then we need to find an adjacent matrix, for which we use the maximum of from_node and to_node.

Matrix is shown below:

	1	2	3	4	5	6	7	8	9	10	...	8288	8289	8290	8291	8292	8293	8294	8295	8296	8297
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	1	0	1	...	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	1	0	1	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8293	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
8294	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
8295	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
8296	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
8297	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

8297 rows × 8297 columns

Similarly, an edge list is created using from_node and to_node.

```
print(edge_list)
```

```
[(30, 1412), (30, 3352), (30, 5254), (30, 5543), (30, 7478), (3, 28), (3, 30), (3, 39)
```

1] Number of nodes

We merge the from_node list and to_node list after that we find uniques that will be the total number of nodes.

```
total_nodes=set(from_node+to_node)
print(len(total_nodes))
```

```
7115
```

2] Number of edges

We calculated a number of ones from the adjacent matrix.

```
edges=0
for i in range(len(adj_mtx)):
  for j in range(len(adj_mtx)):
    if(adj_mtx[i][j]==1):
      edges+=1
print(edges)
```

```
103689
```

3] Average in-degree

First of all, we calculate the in-degree of each and every node by using the column vector of the adjacent matrix

So we found a number of indegrees in the graph and the corresponding indegrees for each node.

```
n=len(in_deg)
print(n)
print(in_deg)
```

```
2381
{3: 31, 6: 20, 8: 44, 10: 15, 15: 361, 19: 22, 23: 22, 28: 122,
```

Average-in-degree = total_in-degree/total_nodes_indegree

```
avg_in_deg=total_in/n
print(avg_in_deg)
```

```
43.54850902981941
```

4] Average out-degree

Same as above, but instead of column here we take row to calculate total out-degrees.

```
n=len(out_deg)
print(n)
print(out_deg)
```

```
6110
{3: 23, 4: 29, 5: 23, 6: 302, 7: 24, 8: 182, 9: 81
```

Average-out-degree = total_out-degree/total_nodes_outdegree

```
avg_out_deg=total_out/n
print(avg_out_deg)
```

```
16.97037643207856
```

5] Maximum in degree

We find maximum value from the dictionary of indegrees of each node.

```
max_in_deg=max(in_deg.values())
print(max_in_deg)
ik = [key for key, val in in_deg.items() if val == max_in_deg]
print(ik)
```

```
457
[4037]
```

6] Maximum out degree

We find maximum value from the dictionary of outdegrees of each node.

```
max_out_deg=max(out_deg.values())
print(max_out_deg)
ok = [key for key, val in out_deg.items() if val == max_out_deg]
print(ok)
```

```
893
[2565]
```

7] The density of network

Density = Number_of_edges of graph / (n(n-1)/2)

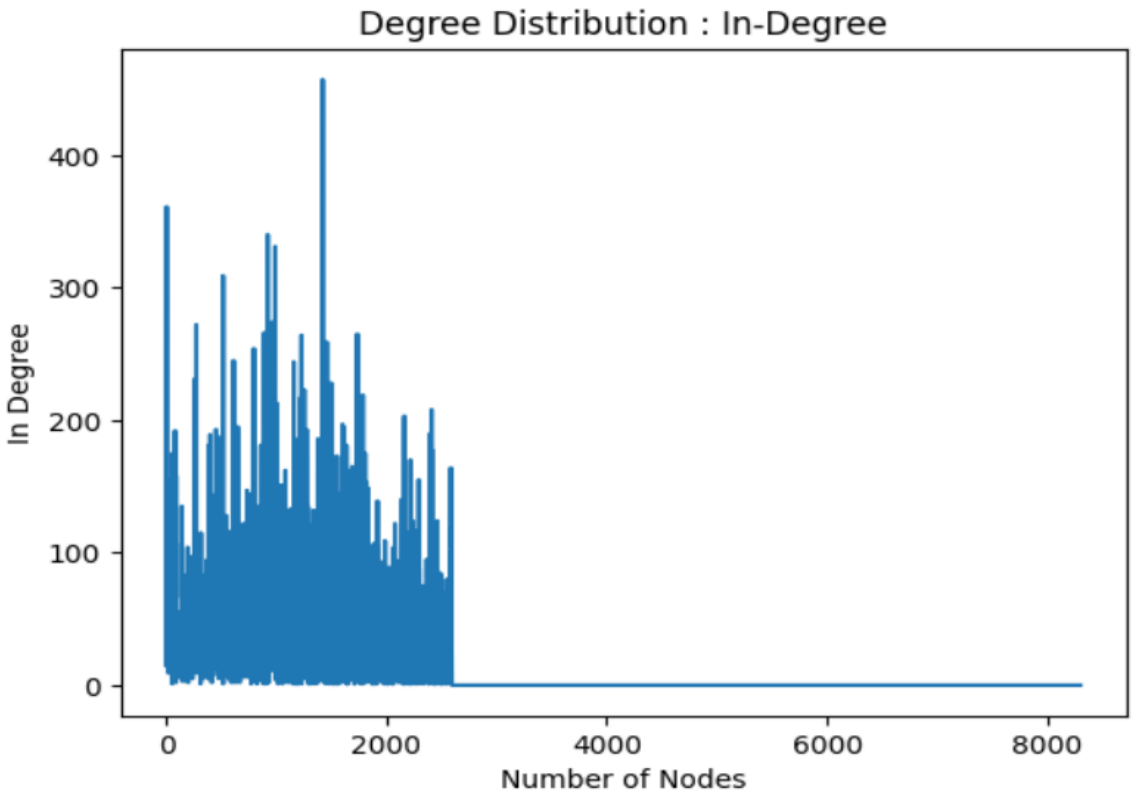
Where n is the number of node of graph

```
n=len(total_nodes)
max_poss_edge=(n*(n-1))//2
density=edges/max_poss_edge
print(density)
```

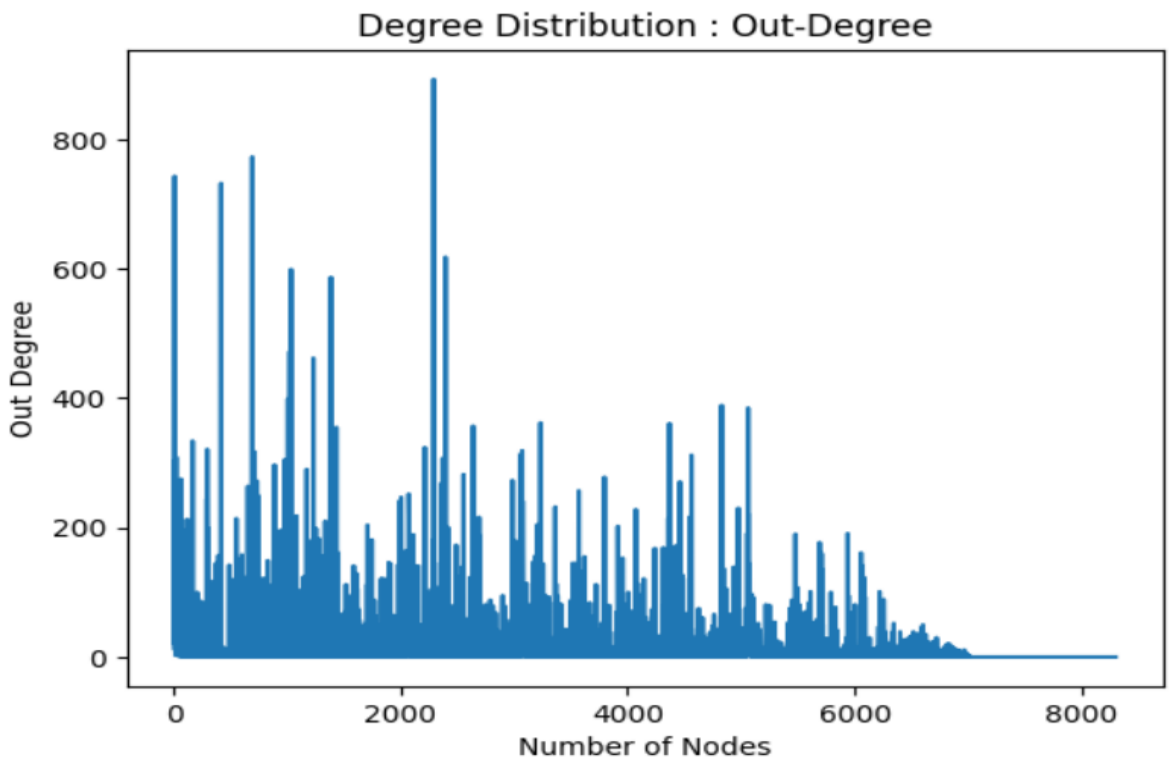
```
0.004097075022161917
```

(1) Plot degree distribution

i) In-degree plot

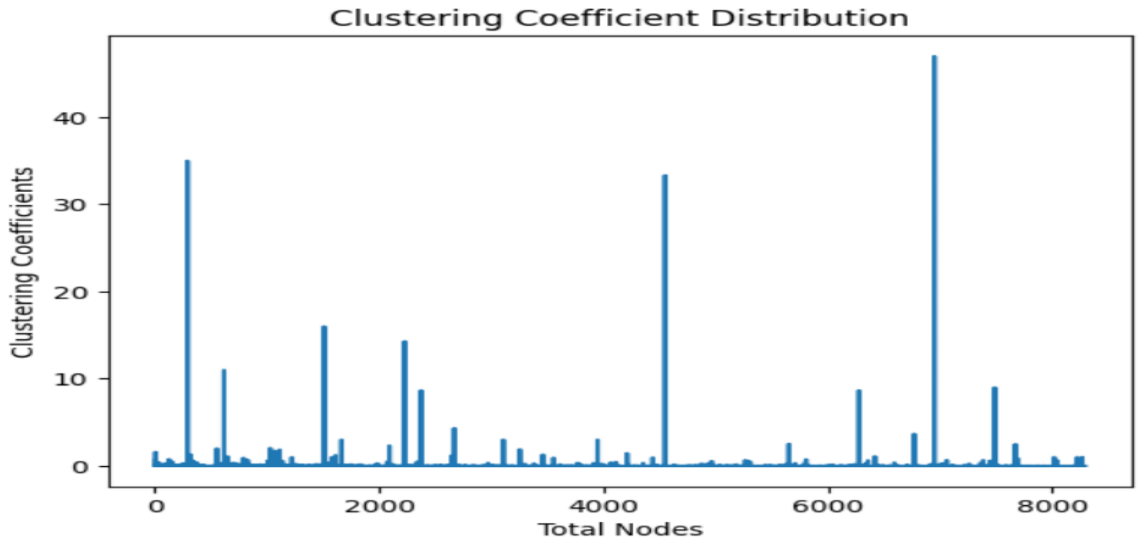


ii) Out-degree plot



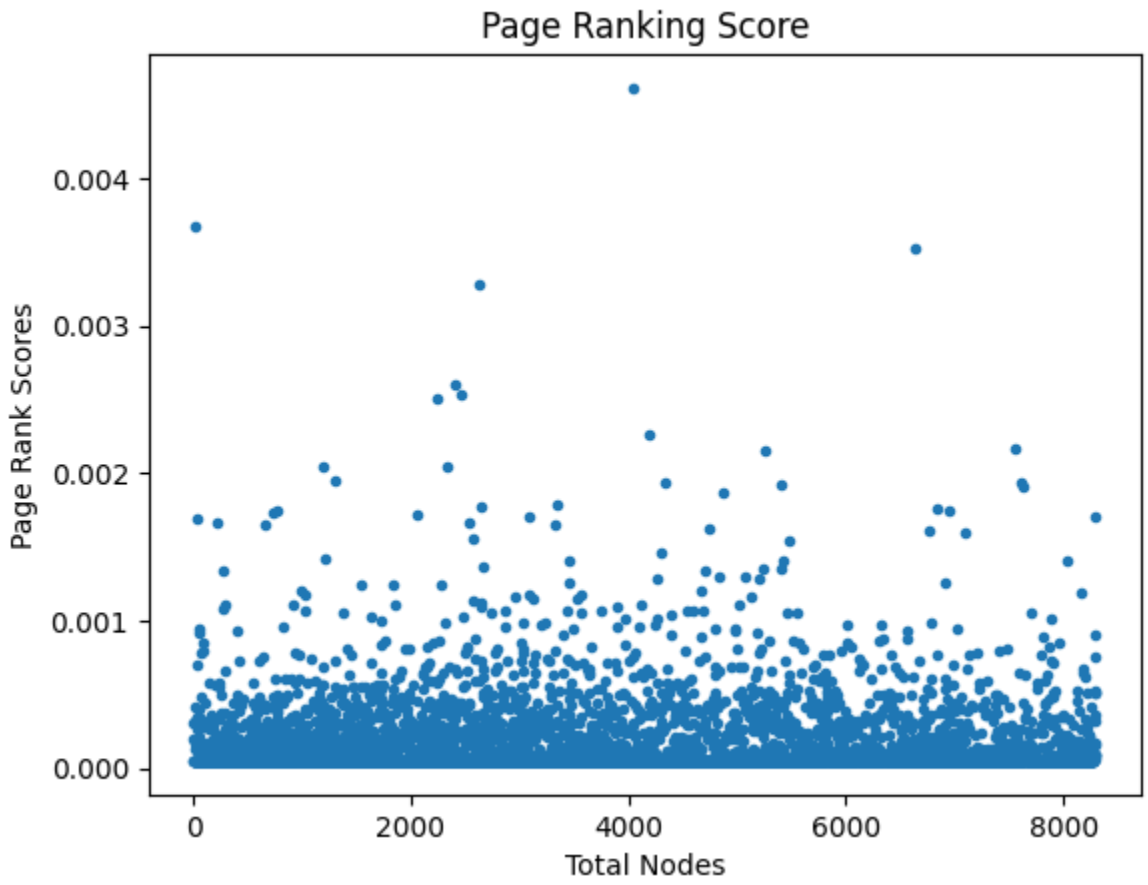
(2) Calculate the local clustering coefficient for each node

```
{1: 0.0, 2: 0.0, 3: 0.04946236559139785, 4: 0.0, 5: 0.0, 6: 1.5894736842105264, 7: 0.0, 8: 0.19238900634249473, 9: 0.0, 10: 0.8
```

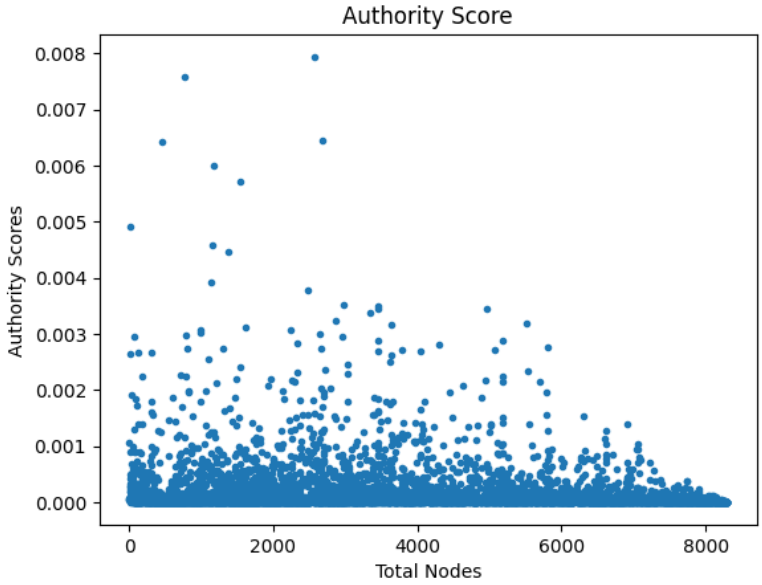


Question 2

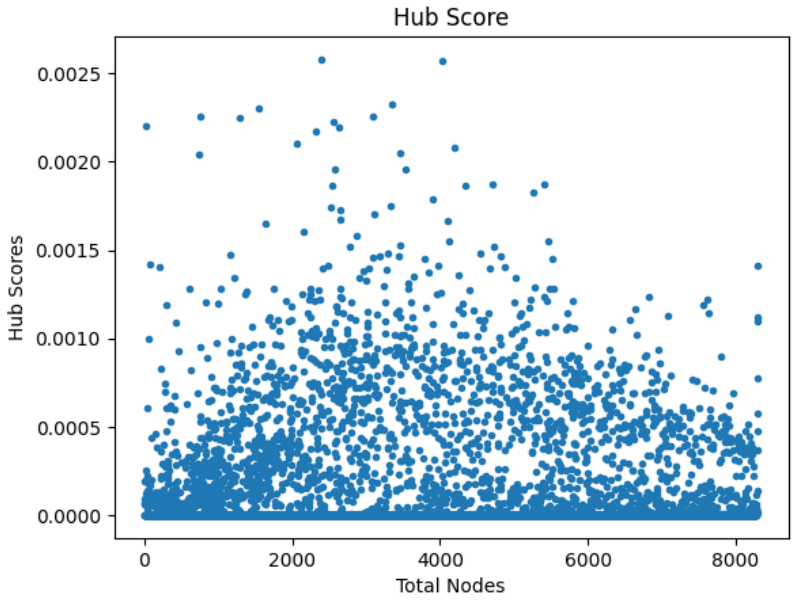
Page Rank Scores



Authority Scores



Hub Score



Comparison

	Pg_score	Authority_score	Hub_score
1	0.004613	0.007940	0.002580
2	0.003681	0.007574	0.002573
3	0.003525	0.006440	0.002328
4	0.003286	0.006417	0.002304
5	0.002605	0.006011	0.002256
6	0.002530	0.005721	0.002253
7	0.002505	0.004921	0.002250
8	0.002266	0.004572	0.002224
9	0.002170	0.004468	0.002202
10	0.002150	0.003919	0.002198

Image shows the top 10 highest values for Page Rank, Authority Score and Hub score.

Highest page rank score is 0.004613 in entire network it means that node corresponding to the highest page rank has higest number of incoming links connected with it.

Highest authority score is 0.007940 in the network. it means that node corresponding to the highest authority score is most important node in terms of providing valuable information or resources to other nodes in the network.

Highest Hub score is 0.002580 in the network. it means that node corresponding to the highest hub score is highly connected to other nodes that are highly connected and therefore considered a hub in the network.