IR Assignment-3

Vishnumolakala Nikhila (MT21103)
Madadi Vineeth Reddy (MT21046)

➢ Importing the necessary libraries such as pandas,numpy,matplotlib,etc.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import networkx as nx
```

Question 1 - Link Analysis

- Adjacency Matrix:

```
adj=[[0 for i in range(9000)] for i in range(9000)]
for e in edges:
    #print(e[0],e[1])
    adj[e[0]][e[1]]=1
adj
```

```
[[0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
```

➢ At first, we initialize with zeros. Then, if there is an edge between two nodes then we make that particular value in the matrix as 1.

- Edge list:

```
with open("Wiki-Vote.txt") as f:
    for l in f:
        l=list(map(int,l.split()))
        edges.append(l)
```

```
[[30, 1412],
 [30, 3352],
 [30, 5254],
 [30, 5543],
 [30, 7478],
 [3, 28],
 [3, 30],
 [3, 39],
 [3, 54],
 [3, 108],
 [3, 152],
 [3, 178],
 [3, 182],
 [3, 214],
```

➢ By printing edges, we get the edge list which specifies the starting and ending point of an edge as a pair of nodes.

- Number of nodes:

```
with open("Wiki-Vote.txt") as f:
    for l in f:
        l=list(map(int,l.split()))
        edges.append(l)
        nodes.append(l[0])
        nodes.append(l[1])
        nodes=list(set(nodes))
```

```
print(f"number of nodes = {len(nodes)}")
```

```
number of nodes = 7115
```

➢ The number of nodes obtained are 7115

- Number of edges:

```
print(f"number of edges = {len(edges)}")
```

```
number of edges = 103689
```

➢ The number of edges obtained are 103689.

- Avg In-degree:

```
#functions for calculating in degree and out degree
ind={}
outd={}
ngh={}

for n in nodes:
    ind[n]=0
    outd[n]=0
    ngh[n]=[]

for e in edges:
    ind[e[1]]+=1
    outd[e[0]]+=1

    ngh[e[0]].append(e[1])
    ngh[e[1]].append(e[0])

for n in nodes:
    ngh[n]=list(set(ngh[n]))
```

```
# 3. Avg In-degree
si=0
for n in nodes:
    si+=ind[n]
print(f"Average in-degree = {si/len(nodes)}")
```

```
Average in-degree = 14.573295853829936
```

➢ Average in-degree value obtained is 14.573295853829936.

- Avg Out-Degree:

```
# 4. Avg. Out-Degree
so=0
for n in nodes:
    so+=outd[n]
print(f"Average in-degree = {so/len(nodes)}")
```

```
Average in-degree = 14.573295853829936
```

➢ Average out-degree value obtained is 14.573295853829936.

- Node with Max In-degree:

```
# 5. Node with Max In-degree
mai=0
ni=0
for n in nodes:
    if(ind[n]>mai):
        ni=n
        mai=ind[n]
print(f"Node={ni} with Max In-degree = {mai}")
```

```
Node=4037 with Max In-degree = 457
```

➢ Node 4037 has the maximum in-degree with the value of 457.

- Node with Max out-degree:

```
# 6. Node with Max out-degree
mao=0
no=0
for n in nodes:
    if(outd[n]>mao):
        no=n
        mao=outd[n]
print(f"Node={no} with Max out-degree = {mao}")
```

```
Node=2565 with Max out-degree = 893
```

➢ Node 2565 has got maximum out-degree with the value of 893.

- The density of the network:

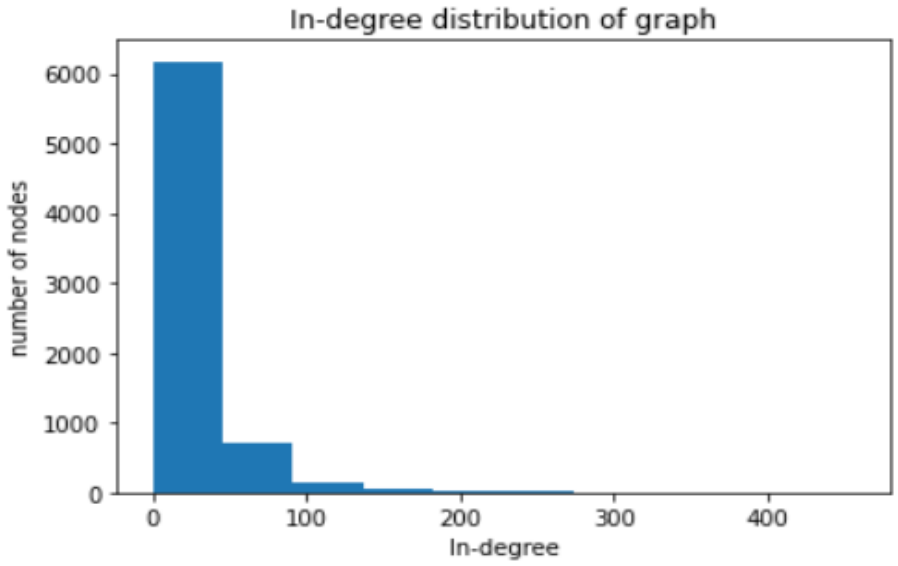
```
# 7. The density of the network
dg=len(edges)/(len(nodes)*(len(nodes)-1))
print(f"density = {dg}")
```

```
density = 0.0020485375110809584
```

➢ The density is calculated by taking the number of edges over the number of nodes whose value is 0.0020485375110809584.

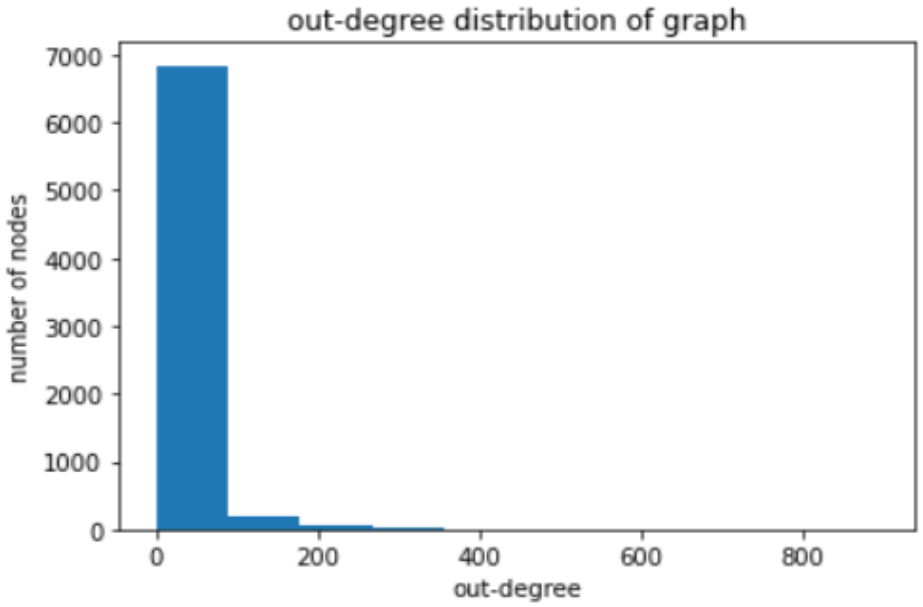
- Degree distribution of the network

```
# 1. Plot degree distribution of the network
dl=list(ind.values())

plt.hist(dl)
plt.title("In-degree distribution of graph")
plt.xlabel("In-degree")
plt.ylabel("number of nodes")
plt.show()
```



➢ Above figure shows the indegree distribution of the network.

```
dl=list(outd.values())

plt.hist(dl)
plt.title("out-degree distribution of graph")
plt.xlabel("out-degree")
plt.ylabel("number of nodes")
plt.show()
```

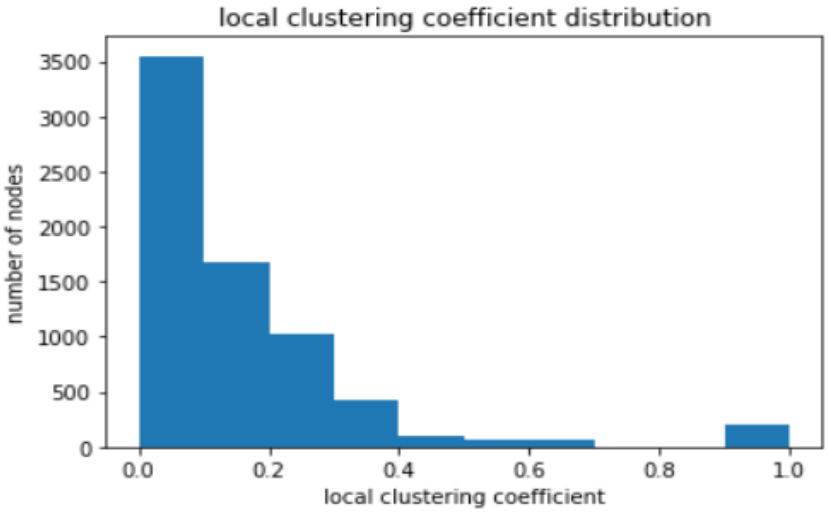


➢ Above figure shows the outdegree distribution of the network.

- <u>local clustering coefficient:</u>

```
# 2. Calculate the local clustering coefficient of each node
lcc={}
for node in nodes:
    neigh=ngh[node]
    n=len(neigh)
    e=0
    if n>1:
        for n1 in neigh:
            for n2 in neigh:
                if n2 in ngh[n1]:
                    e+=1
        e=e//2
        cc=(2*e)/(n*(n-1))
        lcc[node]=cc
    else:
        lcc[node]=0.0

lcc1=lcc.values()
```

```
plt.hist(lcc1)
plt.title("local clustering coefficient distribution")
plt.xlabel("local clustering coefficient")
plt.ylabel("number of nodes")
plt.show()
```



Question 2 - PageRank, Hubs and Authority

- PageRank score for each node:

```
# 1. PageRank score for each node
pr=nx.pagerank(g)
pr
```

```
{3: 0.00020539498232448016,
 4: 5.048782345863015e-05,
 5: 5.048782345863015e-05,
 6: 0.00031183250978437455,
 7: 5.048782345863015e-05,
 8: 0.00032663557615950425,
 9: 5.048782345863015e-05,
 10: 0.0004213996615598798,
 11: 5.048782345863015e-05,
 12: 5.048782345863015e-05,
 13: 5.048782345863015e-05,
 14: 5.048782345863015e-05,
 15: 0.00368122072952927,
 16: 5.048782345863015e-05,
 17: 5.048782345863015e-05,
 18: 5.048782345863015e-05,
 19: 0.00013112179292607272,
 20: 5.048782345863015e-05,
 21: 5.048782345863015e-05,
```

➢ Page rank score of each node can be seen in the above figure.

- Hubscore for each node:

```
hub,auth=nx.hits(g)
```

```
hubs=dict(sorted(hub.items(), key=lambda x: x[1], reverse = True))
hubs
```

```
{2565: 0.007940492708143142,
 766: 0.00757433529750125,
 2688: 0.006440248991029863,
 457: 0.006416870490261071,
 1166: 0.006010567902411203,
 1549: 0.005720754058269245,
 11: 0.004921182063808104,
 1151: 0.004572040701756409,
 1374: 0.004467888792711107,
 1133: 0.00391888173205735,
 2485: 0.003784460813080375,
 2972: 0.003517673976814717,
 3449: 0.003503558110460445,
 3453: 0.003449414861112209,
 4967: 0.003443340741834129,
 3352: 0.0033814231063450007,
 2871: 0.00323901670172771,
 5524: 0.0031957811103467968,
 3642: 0.003156068703698414,
```

➢ The hub score for each node can be seen from the above figure.

- Authority score for each node:

```
auths=dict(sorted(auth.items(), key=lambda x: x[1], reverse = True))
auths
```

```
{2398: 0.0025801471780088 75,
 4037: 0.0025732411242297 92,
 3352: 0.0023284150914976848,
 1549: 0.00230373148045718,
 762: 0.00225587485628714,
 3089: 0.0022534066884511636,
 1297: 0.0022501446366627238,
 2565: 0.002223564103953613,
 15: 0.002201543492565582,
 2625: 0.002197896803403074,
 2328: 0.0021723715453407406,
 2066: 0.002107040939609975,
 4191: 0.0020811941305289884,
 3456: 0.002050435521510778,
 737: 0.0020393826293356697,
 3537: 0.001957956707591018,
 2576: 0.0019547902768889515,
 4712: 0.001871635752056829,
 5412: 0.0018694113161489125,
```

➢ The authority score of each node is represented in the above figure.

- <u>Comparing the above results:</u>

➢ Page rank, authority score and hub score, all are used to evaluate the importance of a web page for their incoming links.
➢ We observe based on the results that some nodes in some metric get higher scores than the others.
➢ This difference is because page rank considers the node that has the highest in-degree whereas authority considers nodes that have incoming links from more hub nodes.
➢ Finally hub scores refer to the nodes with their outgoing nodes and thus there is variation.