SOCIAL NETWORK ANALYTICS LAB
ASSESSMENT: -1

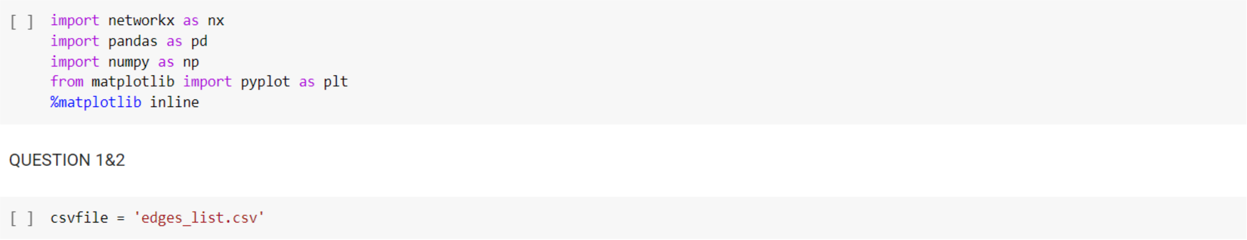
NAME: - AMAN KUMAR SHARMA
REG. NO.: - 22MCB0021
COURSE CODE: - MCSE618P
DATE: - 23 MARCH 2023

Social-Network-Analytics-Graph-Attributes-And-Centrality-Measures

NetworkX:

NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. It is used to study large complex networks represented in form of graphs with nodes and edges. Using networkx we can load and store complex networks. We can generate many types of random and classic networks, analyze network structure, build network models, design new network algorithms and draw networks.

Importing the libraries



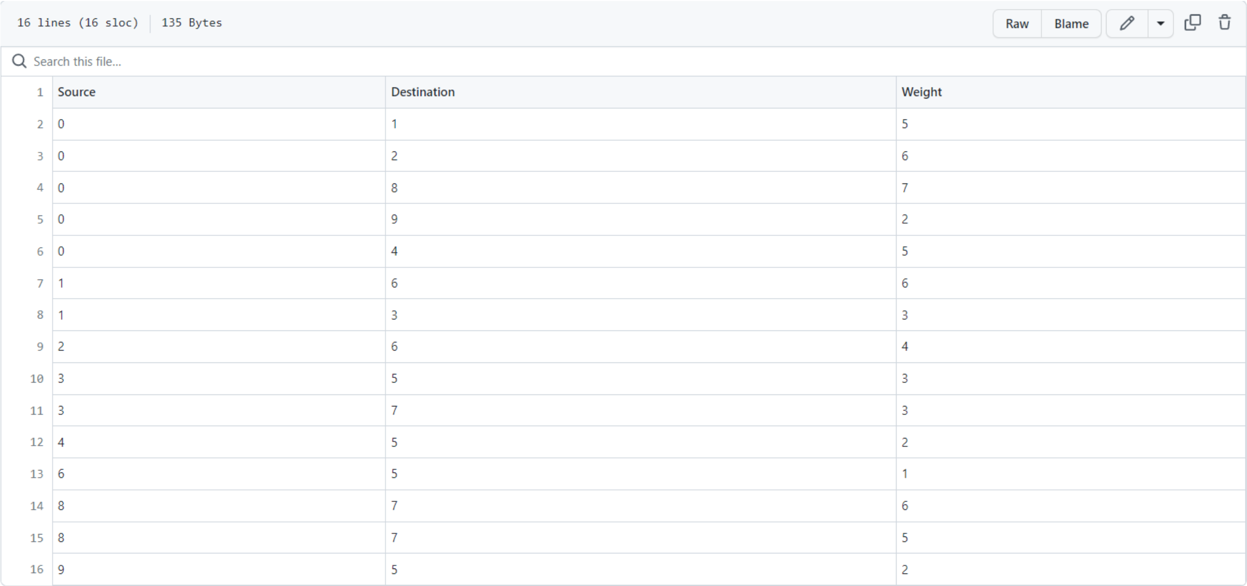
```
[ ] import networkx as nx
    import pandas as pd
    import numpy as np
    from matplotlib import pyplot as plt
    %matplotlib inline
```

QUESTION 1&2

```
[ ] csvfile = 'edges_list.csv'
```

Description of CSV File:

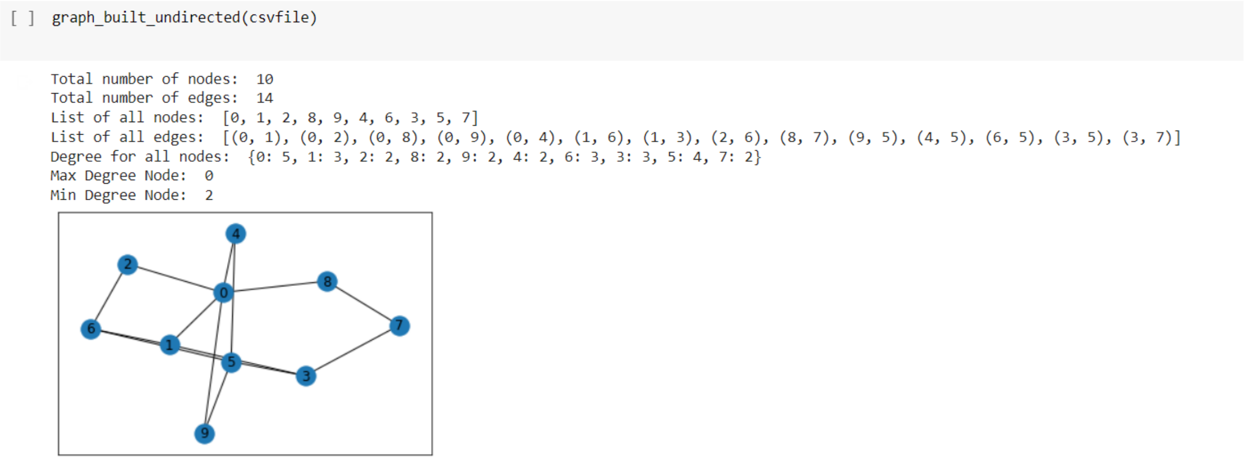
A CSV (comma-separated values) file is a text file with a specific format allowing data to be saved in a table-structured format.



Source	Destination	Weight
0	1	5
0	2	6
0	8	7
0	9	2
0	4	5
1	6	6
1	3	3
2	6	4
3	5	3
3	7	3
4	5	2
6	5	1
8	7	6
8	7	5
9	5	2

Undirected Graph:

An undirected graph is a graph, i.e., a set of objects (called vertices or nodes) that are connected, where all the edges are bidirectional. An undirected graph is sometimes called an undirected network.

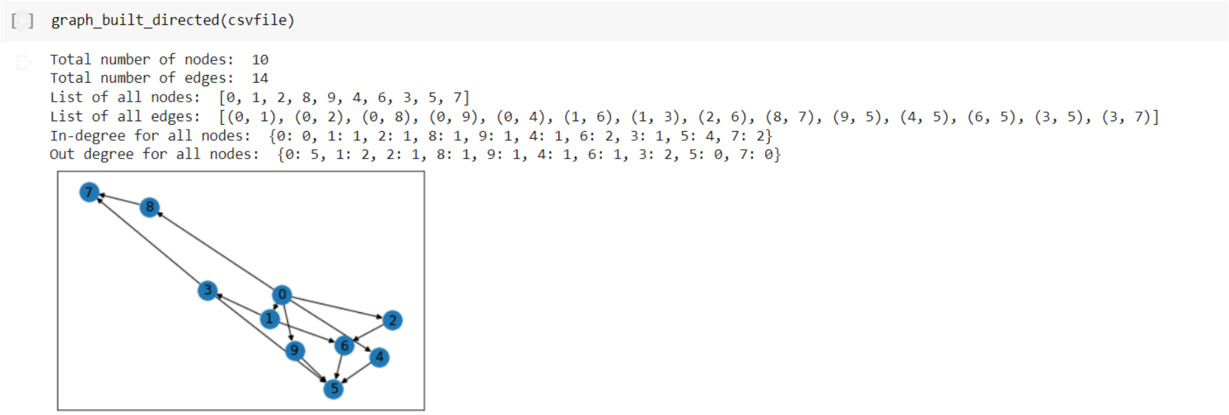


```
[ ]  graph_built_undirected(csvfile)
```

```
Total number of nodes:  10
Total number of edges:  14
List of all nodes:  [0, 1, 2, 8, 9, 4, 6, 3, 5, 7]
List of all edges:  [(0, 1), (0, 2), (0, 8), (0, 9), (0, 4), (1, 6), (1, 3), (2, 6), (8, 7), (9, 5), (4, 5), (6, 5), (3, 5), (3, 7)]
Degree for all nodes:  {0: 5, 1: 3, 2: 2, 8: 2, 9: 2, 4: 2, 6: 3, 3: 3, 5: 4, 7: 2}
Max Degree Node:  0
Min Degree Node:  2
```

Directed Graph:

A directed graph is a graph, i.e., a set of objects (called vertices or nodes) that are connected, where all the edges are directed from one vertex to another. A directed graph is sometimes called a digraph or a directed network.



```
[ ]  graph_built_directed(csvfile)
```

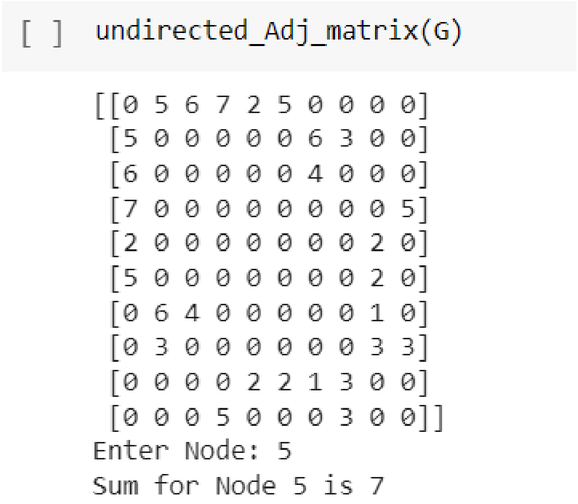
```
Total number of nodes:  10
Total number of edges:  14
List of all nodes:  [0, 1, 2, 8, 9, 4, 6, 3, 5, 7]
List of all edges:  [(0, 1), (0, 2), (0, 8), (0, 9), (0, 4), (1, 6), (1, 3), (2, 6), (8, 7), (9, 5), (4, 5), (6, 5), (3, 5), (3, 7)]
In-degree for all nodes:  {0: 0, 1: 1, 2: 1, 8: 1, 9: 1, 4: 1, 6: 2, 3: 1, 5: 4, 7: 2}
Out degree for all nodes:  {0: 5, 1: 2, 2: 1, 8: 1, 9: 1, 4: 1, 6: 1, 3: 2, 5: 0, 7: 0}
```

QUESTION 3

Adjacency Matrix:

The adjacency matrix also called the connection matrix, contains rows and columns that represent a simple labeled graph, with 0 or 1 in the position of (Vi, Vj) according to the condition of whether Vi and Vj are adjacent or not.

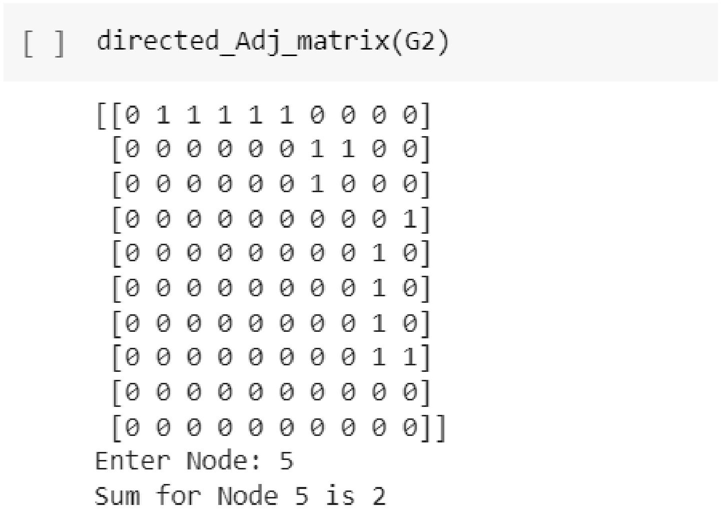
For Undirected Graph:



```
[ ]  undirected_Adj_matrix(G)
```

```
[[0 5 6 7 2 5 0 0 0 0]
 [5 0 0 0 0 0 6 3 0 0]
 [6 0 0 0 0 0 4 0 0 0]
 [7 0 0 0 0 0 0 0 0 5]
 [2 0 0 0 0 0 0 0 2 0]
 [5 0 0 0 0 0 0 0 2 0]
 [0 6 4 0 0 0 0 0 1 0]
 [0 3 0 0 0 0 0 0 3 3]
 [0 0 0 0 2 2 1 3 0 0]
 [0 0 0 5 0 0 0 3 0 0]]
Enter Node: 5
Sum for Node 5 is 7
```

For directed Graph:



```
[ ]  directed_Adj_matrix(G2)

      [[0 1 1 1 1 1 0 0 0 0]
       [0 0 0 0 0 0 1 1 0 0]
       [0 0 0 0 0 0 1 0 0 0]
       [0 0 0 0 0 0 0 0 0 1]
       [0 0 0 0 0 0 0 0 1 0]
       [0 0 0 0 0 0 0 0 1 0]
       [0 0 0 0 0 0 0 0 1 0]
       [0 0 0 0 0 0 0 0 1 1]
       [0 0 0 0 0 0 0 0 0 0]
       [0 0 0 0 0 0 0 0 0 0]]
      Enter Node: 5
      Sum for Node 5 is 2
```

<u>Centrality Measures:</u>

Centrality measures are scalar values given to each node in the graph to quantify its importance based on an assumption.

1. <u>Degree Centrality</u>

Degree Centrality is a measure of node importance in a network based on the number of links or edges that a node has with other nodes in the network. The degree centrality of a node can be calculated using the following formula:

Degree Centrality of Node i = number of edges connected to Node i / (N-1)

where N is the total number of nodes in the network. This formula calculates the degree centrality of a node by dividing the number of edges that connect to the node by the maximum number of possible edges that the node could have in the network, which is N-1 (since a node cannot be connected to itself). The resulting value ranges from 0 to 1, with higher values indicating a higher degree of centrality.

2. Betweenness centrality:

Betweenness Centrality is a measure of node importance in a network based on the extent to which a node lies on the shortest paths between other pairs of nodes in the network. The betweenness centrality of node i is calculated as follows:

For each pair of nodes (j,k) in the network, find the shortest path(s) between them.

For each node, i in the network, calculate the fraction of these shortest paths that pass-through i. This fraction is called the betweenness centrality of node i.

The betweenness centrality of node i is given by the following formula:

Betweenness Centrality of Node i = $\Sigma\ (s, t) \in \Phi(i)\ \sigma(s, t \mid i) / \sigma(s, t)$

where $\Phi(i)$ is the set of all pairs of nodes in the network that include node i, $\sigma(s, t)$ is the total number of shortest paths between nodes s and t, and $\sigma(s, t \mid i)$ is the number of those paths that pass through node i.

In other words, the betweenness centrality of a node i is the sum of the fraction of all shortest paths in the network that pass-through i. This measure gives an indication of the extent to which a node acts as a "bridge" between other nodes in the network and is often used to identify nodes that play a critical role in the flow of information or resources within the network.

3. Closeness centrality:

Closeness centrality is a measure of centrality in a network that calculates the average shortest distance from each vertex to every other vertex. It is calculated as the reciprocal of the sum of the length of the shortest paths between a node and all other nodes in the graph.

The formula for closeness centrality is 1/ (average distance to all other vertices).

In a connected graph, closeness centrality (or closeness) of a node is calculated as the sum of the length of the shortest paths between the node and all other nodes in the graph.

Closeness centrality measures how central a node is in a network. The more central a node is, the closer it is to all other nodes. Closeness centrality can be used to identify important nodes in a network that are critical for information flow or communication. It can also be used to compare different nodes' importance within a network.

There are three different ways to measure network centrality: degree, closeness, and betweenness. Degree centrality measures how many connections each node has, while betweenness centrality measures how often each node appears on the shortest paths between pairs of other nodes.

4. Eigenvector Centrality:

Eigenvector centrality is a measure of the influence of a node in a network. It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes.

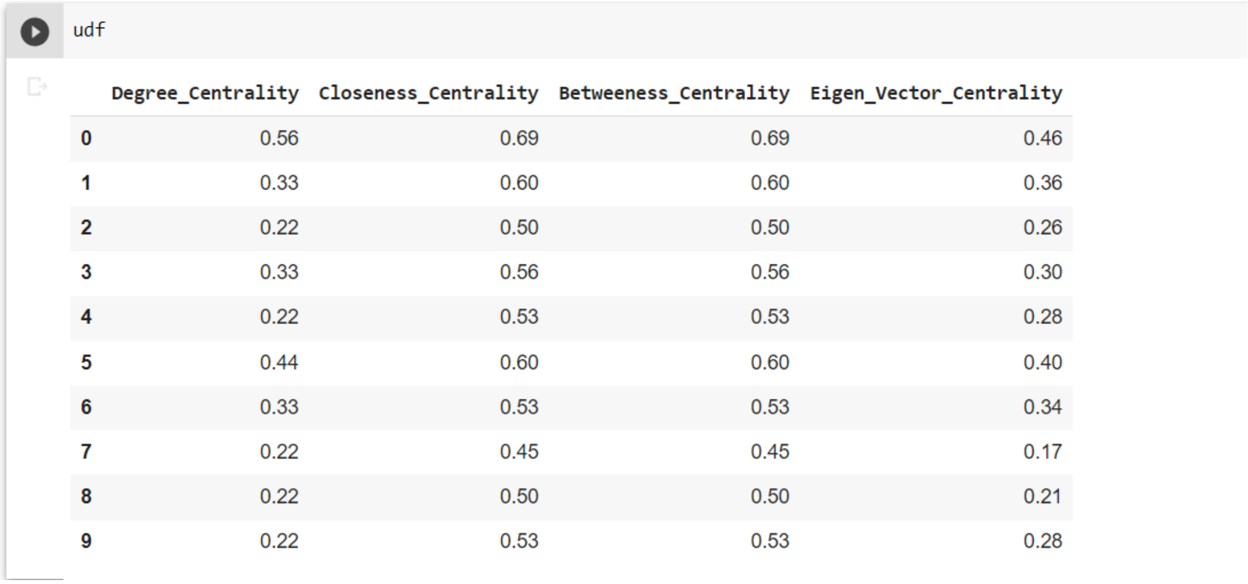
Eigenvector centrality is calculated using eigenvectors, which are a set of vectors associated with a linear system of equations. The eigenvector centrality score for each node is proportional to the sum of its neighbors' scores, with weights given by the adjacency matrix.

Mathematically, eigenvector centrality can be represented as:

$Ax = \lambda x$

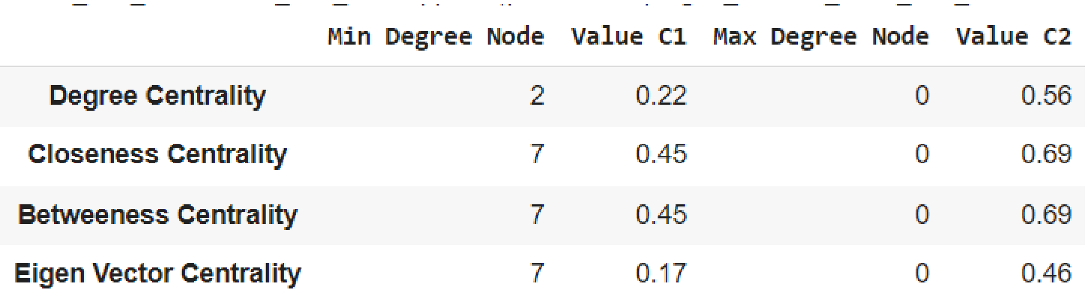
where A is the adjacency matrix of the graph, x is an eigenvector corresponding to eigenvalue λ, and Ax represents matrix multiplication between A and x.

Centrality Measures for given Undirected graph:



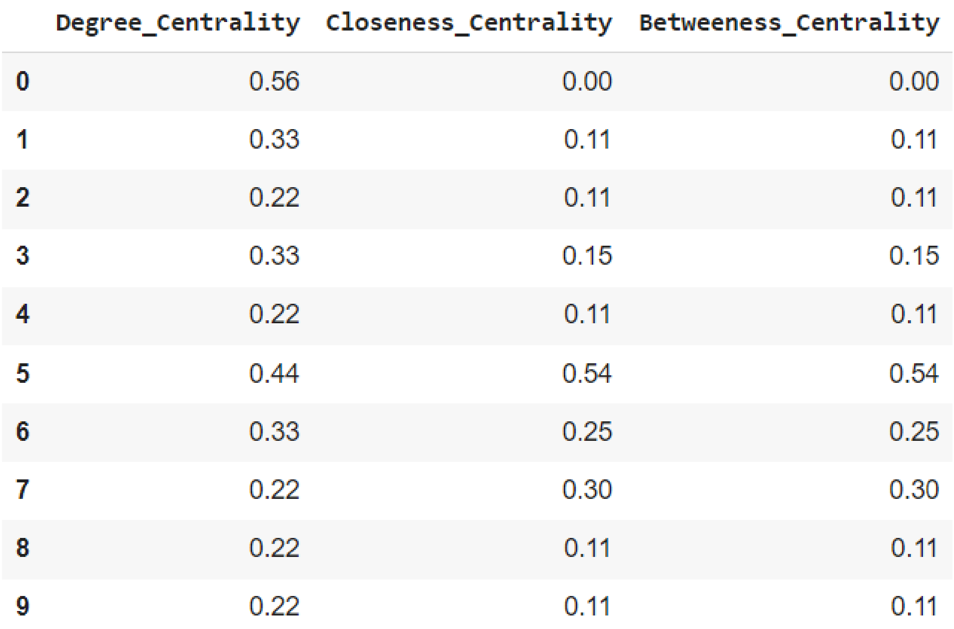
```
udf
```

	Degree_Centrality	Closeness_Centrality	Betweeness_Centrality	Eigen_Vector_Centrality
0	0.56	0.69	0.69	0.46
1	0.33	0.60	0.60	0.36
2	0.22	0.50	0.50	0.26
3	0.33	0.56	0.56	0.30
4	0.22	0.53	0.53	0.28
5	0.44	0.60	0.60	0.40
6	0.33	0.53	0.53	0.34
7	0.22	0.45	0.45	0.17
8	0.22	0.50	0.50	0.21
9	0.22	0.53	0.53	0.28

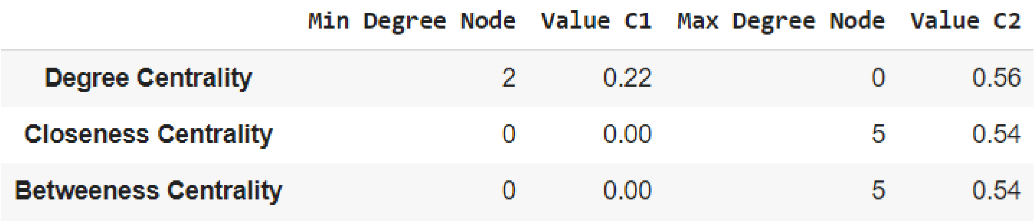


	Min Degree Node	Value C1	Max Degree Node	Value C2
Degree Centrality	2	0.22	0	0.56
Closeness Centrality	7	0.45	0	0.69
Betweeness Centrality	7	0.45	0	0.69
Eigen Vector Centrality	7	0.17	0	0.46

Centrality Measures for given directed graph:



	Degree_Centrality	Closeness_Centrality	Betweeness_Centrality
0	0.56	0.00	0.00
1	0.33	0.11	0.11
2	0.22	0.11	0.11
3	0.33	0.15	0.15
4	0.22	0.11	0.11
5	0.44	0.54	0.54
6	0.33	0.25	0.25
7	0.22	0.30	0.30
8	0.22	0.11	0.11
9	0.22	0.11	0.11



	Min Degree Node	Value C1	Max Degree Node	Value C2
Degree Centrality	2	0.22	0	0.56
Closeness Centrality	0	0.00	5	0.54
Betweeness Centrality	0	0.00	5	0.54