# Experiment No. 6

**Aim:** Develop Structure based social media analytics model for any business.

**Theory:**

We are performing social media analytics using data from Facebook. This social networkanalysis is mainly executed with the library of NetworkX.

In detail, the Facebook circles (friends lists) of ten people will be examined and scrutinized in order to extract all kinds of valuable information. The dataset has been downloaded from Stanford's website. It is known that a Facebook network is undirected and has no weights because one user can become friends with another user just once. Looking at the dataset from a graph analysis perspective:

- Each node represents an anonymized Facebook user that belongs to one of those ten friends lists.
- Each edge corresponds to the friendship of two Facebook users that belong to this network. In other words, two users must become friends on Facebook in order for them tobe connected in the particular network.

In order to perform the social media analytics, nodes 0, 107, 348, 414, 686, 698, 1684, 1912, 3437, 3980 are the ones whose friends list will be examined. That means that they are in the spotlight of this analysis. Those nodes are considered the spotlight nodes.

Specifically, we can analyze the network structure of the spotlight nodes by examining the following network measures:

1. Degree centrality: This measures the number of connections that each node has in the network. We can use this measure to identify the most connected nodes in the network, which may be important influencers or connectors.
2. Betweenness centrality: This measures the extent to which a node lies on the shortest pathbetween other nodes in the network. Nodes with high betweenness centrality may act as bridges or gatekeepers between different parts of the network.
3. Clustering coefficient: This measures the degree to which the neighbors of a node are also connected to each other. High clustering coefficients suggest the presence of tightly knit groups or communities in the network.
4. Network density: This measures the proportion of possible edges in the network that are actually present. A high density suggests a more tightly connected network, whereas alow density suggests a sparser network.

colab.research.google.com/drive/1AYtXtFzbjrDGkzC2bniyF1uINnWjPmDU#scrollTo=MJTgsWqBtB15

sma exp 6.ipynb

File Edit View Insert Runtime Tools Help Last saved at 12:02 PM

Comment Share

+ Code + Text

RAM Disk

```
pip install networkx matplotlib
```

```
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (3.2.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.49.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (24.0)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

```
import networkx as nx
import matplotlib.pyplot as plt

# Create a graph
G = nx.Graph()

# Add edges based on the spotlight nodes
edges = [(0, 107), (0, 348), (0, 414), (0, 686), (0, 698), (0, 1684), (0, 1912), (0, 3437), (0, 3980)]
G.add_edges_from(edges)
```

1s completed at 12:00 PM

sma exp 6.ipynb

File Edit View Insert Runtime Tools Help Last saved at 12:02 PM

Comment Share

+ Code + Text

RAM Disk

```
import networkx as nx
import matplotlib.pyplot as plt

# Create a graph
G = nx.Graph()

# Add edges based on the spotlight nodes
edges = [(0, 107), (0, 348), (0, 414), (0, 686), (0, 698), (0, 1684), (0, 1912), (0, 3437), (0, 3980)]
G.add_edges_from(edges)
```


```
# Calculate degree centrality
degree_centrality = nx.degree_centrality(G)
print("Degree Centrality:", degree_centrality)

# Calculate betweenness centrality
betweenness_centrality = nx.betweenness_centrality(G)
print("Betweenness Centrality:", betweenness_centrality)

# Calculate clustering coefficient
clustering_coefficient = nx.clustering(G)
print("Clustering Coefficient:", clustering_coefficient)

# Calculate network density
density = nx.density(G)
print("Network Density:", density)
```

sma exp 6.ipynb

File Edit View Insert Runtime Tools Help Last saved at 12:02 PM

Comment Share

+ Code + Text

RAM Disk

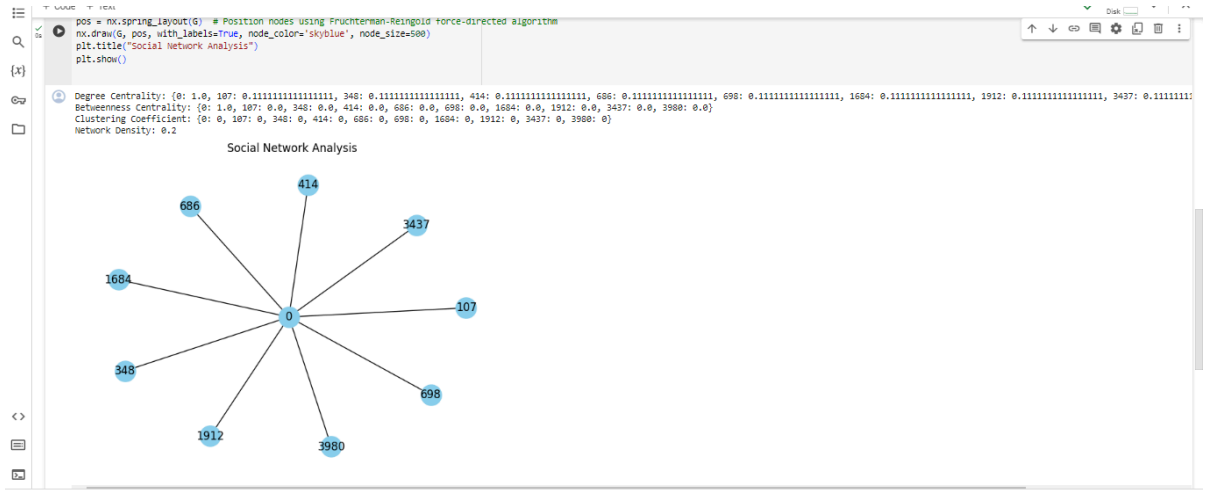
```
# Calculate betweenness centrality
betweenness_centrality = nx.betweenness_centrality(G)
print("Betweenness Centrality:", betweenness_centrality)

# Calculate clustering coefficient
clustering_coefficient = nx.clustering(G)
print("Clustering Coefficient:", clustering_coefficient)

# Calculate network density
density = nx.density(G)
print("Network Density:", density)

# Visualization
pos = nx.spring_layout(G)  # Position nodes using Fruchterman-Reingold force-directed algorithm
nx.draw(G, pos, with_labels=True, node_color='skyblue', node_size=500)
plt.title("Social Network Analysis")
plt.show()
```

```
Degree Centrality: {0: 1.0, 107: 0.1111111111111111, 348: 0.1111111111111111, 414: 0.1111111111111111, 686: 0.1111111111111111, 698: 0.1111111111111111, 1684: 0.1
Betweenness Centrality: {0: 1.0, 107: 0.0, 348: 0.0, 414: 0.0, 686: 0.0, 698: 0.0, 1684: 0.0, 1912: 0.0, 3437: 0.0, 3980: 0.0}
Clustering Coefficient: {0: 0, 107: 0, 348: 0, 414: 0, 686: 0, 698: 0, 1684: 0, 1912: 0, 3437: 0, 3980: 0}
Network Density: 0.2
```




**Conclusion:** Thus, by analyzing these network measures for the spotlight nodes, we can gain insights into the overall network structure and identify important nodes or subgroups within the network. This can be used to gain a deeper understanding of the relationship between thesenodes.