SOCIAL NETWORK ANALYSIS

MARCH 9, 2025

Table of Contents

List of Figures:

Report: Social Network Analysis

1. Introduction

A thorough analysis of social network data occurs through the combination of Python programming language and networkx library. The Facebook combined network serves as the data source with its 4,039 node elements and 88,234 edges. The study examines Facebook network structure by building the network followed by clustering coefficient analysis and density measurements and connected components study and centrality analysis and random graph model comparisons (Erdős-Rényi, Barabási-Albert, and Watts-Strogatz). The report ends by asking a research question about how to locate central nodes within the network system.

2. Network Construction

2.1 Dataset Selection

Facebook network dataset chosen for this analysis, available publicly and can be accessed Here. The social network dataset consists of user entities connected through friendship relationships. The networkx library enabled the loading process of the dataset which serves as a tool to work with and modify complex network structures.

2.2 Network Creation

The construction of the network occurred when the program read the edge list data from its corresponding file. The networkx.read_edgelist function read the graph while converting all nodes into integer values. The final graph G possesses 4,039 nodes along with 88,234 edges that meet the necessary requirement of 1,000 nodes.

```
import networkx as nx

def load_graph(file_path):

    G = nx.read_edgelist(file_path, nodetype=int)

    return G

file_path = "C:/Users/ks615/Downloads/B107/facebook_combined.txt"

G = load_graph(file_path)
```

3. Network Analysis

3.1 Clustering Coefficient and Density

The average clustering coefficient evaluates the node tendency to develop clustered structures or cliques. The network density calculation determines the relationship between actual edges and maximum theoretical edges. Networkx functions enabled the assessment of these two metrics.

```
clustering_coeff = nx.average_clustering(G)
density = nx.density(G)
print(f"Average clustering coefficient: {clustering_coeff}")
print(f"Network density: {density}")
```

The average clustering coefficient was found to be **0.6055**, indicating a high level of clustering, which is common in social networks. The network density was **0.0108**, indicating a sparse network.

3.2 Connected Components

The size of the largest connected component was calculated to understand the network's connectivity. In this network, the largest connected component includes all 4,039 nodes, indicating that the network is fully connected.

```
largest_cc = max(nx.connected_components(G), key=len)
print(f"Size of largest connected component: {len(largest_cc)}")
```

3.3 Centrality Analysis

Centrality measures were used to identify the most influential nodes in the network. Degree centrality, which measures the number of connections a node has, was used to rank the nodes. The top 10 central nodes were identified.

```
centralities = nx.degree_centrality(G)
top_nodes = sorted(centralities, key=centralities.get, reverse=True)[:10]
print("Top 10 central nodes:", top_nodes)
```

The top 10 central nodes are: **[107, 1684, 1912, 3437, 0, 2543, 2347, 1888, 1800, 1663]**. These nodes are likely to be influential users in the network.

3.4 Cluster Visualization

The network visualization used cluster diagrams that displayed similar colors for nodes which belonged to the same community. The networkx package included greedy_modularity_communities function which detected communities.

```
def plot_clusters(G, title="Cluster Diagram"):
    plt.figure(figsize=(10, 8))
    pos = nx.spring_layout(G)
    clusters = list(nx.community.greedy_modularity_communities(G))
    colors = ["#" + ''.join(random.choices('0123456789ABCDEF', k=6)) for _ in
range(len(clusters))]
    for i, cluster in enumerate(clusters):
        nx.draw_networkx_nodes(G, pos, nodelist=list(cluster), node_size=50, alpha=0.8,
label=f'Cluster {i+1}', node_color=colors[i])
    nx.draw_networkx_edges(G, pos, alpha=0.5)
    plt.title(title)
    plt.legend()
    plt.show()
plot_clusters(G)
```

The cluster diagram reveals distinct communities within the network, indicating that users tend to form tightly-knit groups.

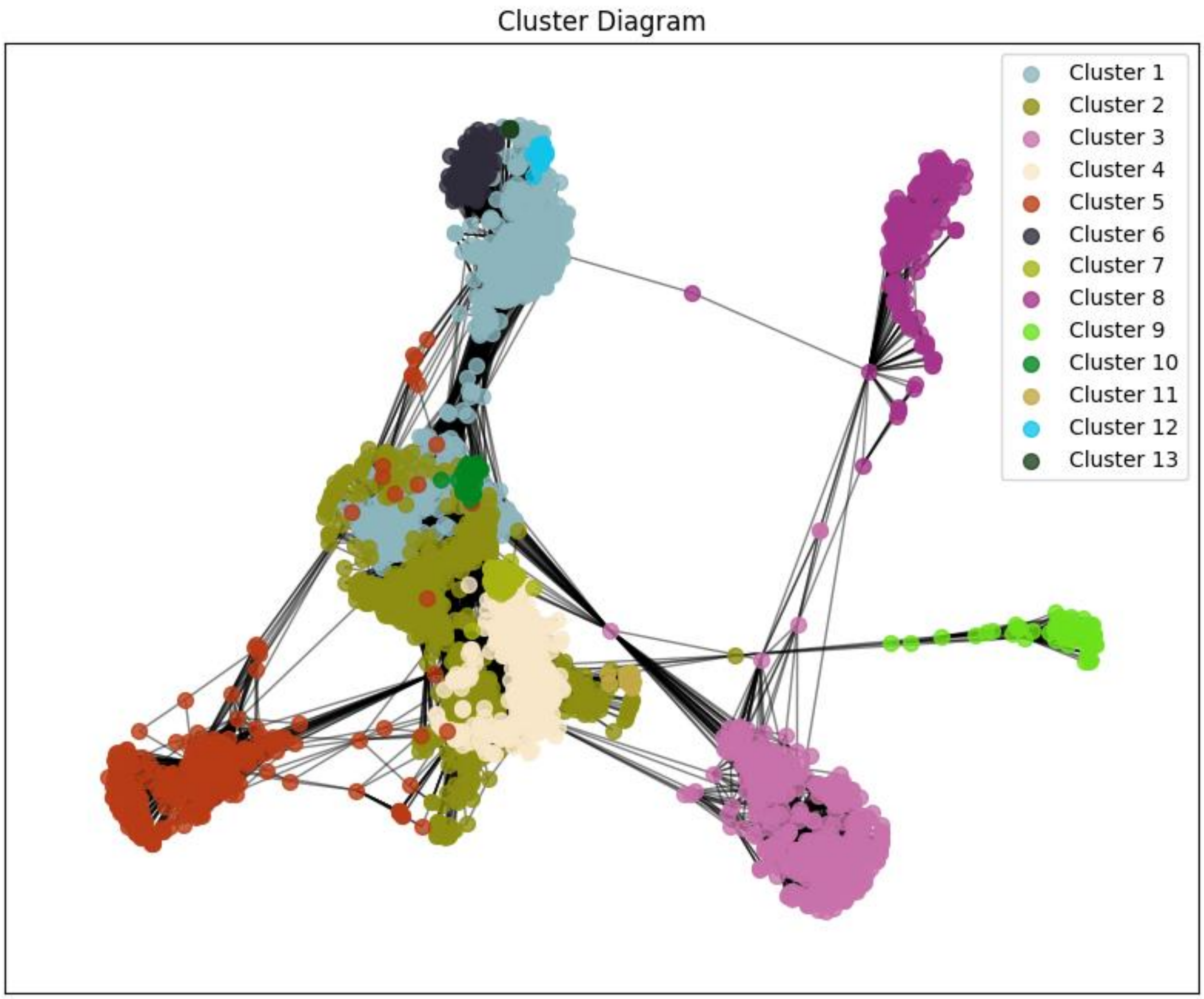

Fig1: Overall cluster visualization

4. Comparison with Random Graph Models

A comparison of Facebook network structure against Erdős-Rényi (ER), Barabási-Albert (BA) and Watts-Strogatz (WS) random graph models took place to analyze its structural composition. The models used the same node and edge configuration as the Facebook network for their construction.

4.1 Erdős-Rényi (ER) Model

Under the ER model all node pairs face equivalent chances to form connections with one another. The clustering coefficient together with density showed lower values in the ER model than in the Facebook network.

```
ER = nx.erdos_renyi_graph(n, m / (n * (n-1)/2))

print(f"ER Model - Clustering Coefficient: {nx.average_clustering(ER)}, Density:
{nx.density(ER)}")
```

ER Model Results:

- Clustering Coefficient: **0.0108**
- Density: **0.0108**

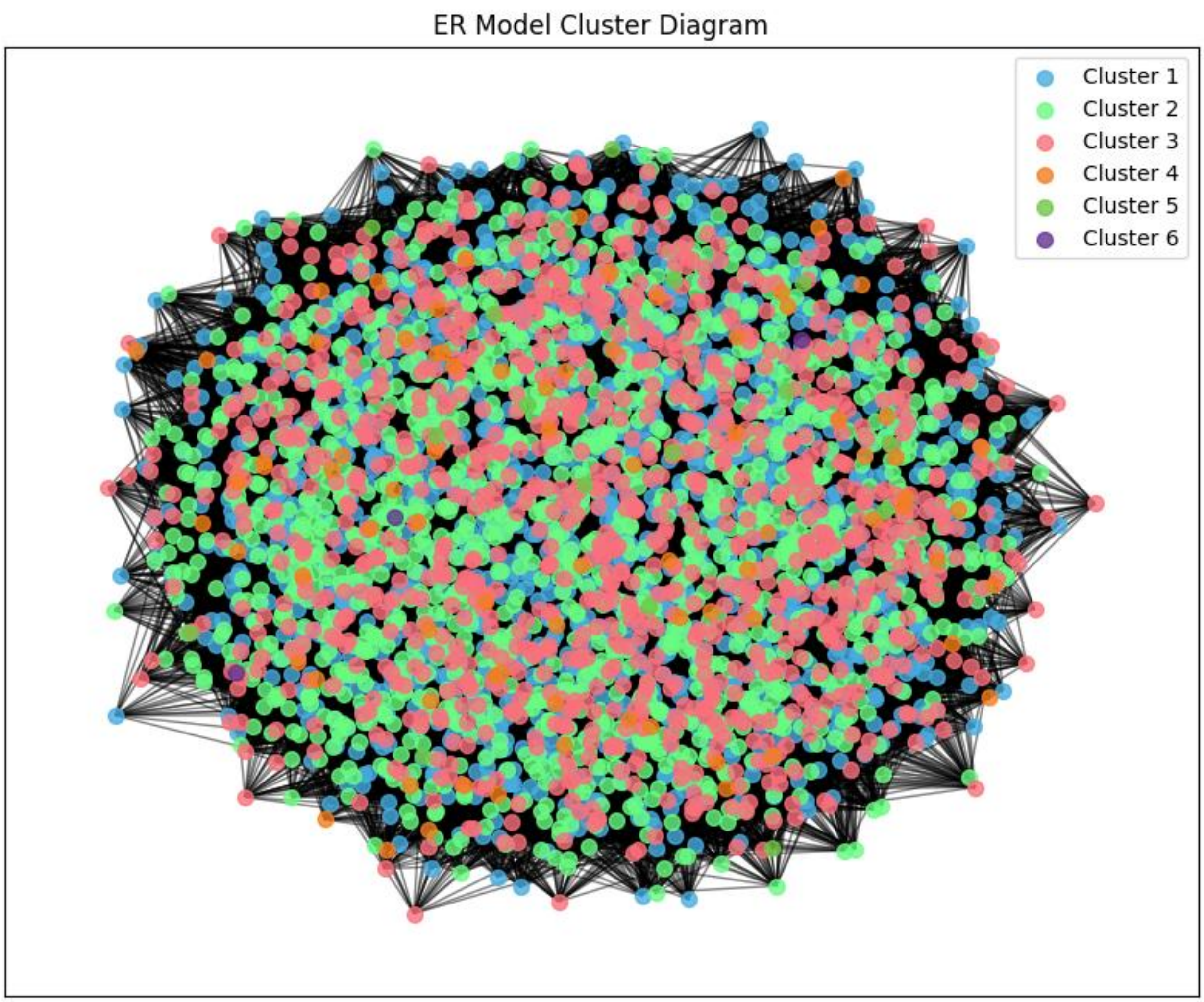

Fig2: ER Model visualization

4.2 Barabási-Albert (BA) Model

The BA model establishes scale-free connections by using preferential attachment during its growth process. The clustering coefficient value in the BA model surpassed the ER model's value yet remained below the clustering coefficient of the Facebook network.

```
BA = nx.barabasi_albert_graph(n, max(1, int(np.mean([deg for _, deg in G.degree()]) / 2))

print(f"BA Model - Clustering Coefficient: {nx.average_clustering(BA)}, Density:
{nx.density(BA)}")
```

BA Model Results:

- Clustering Coefficient: **0.0357**
- Density: **0.0103**

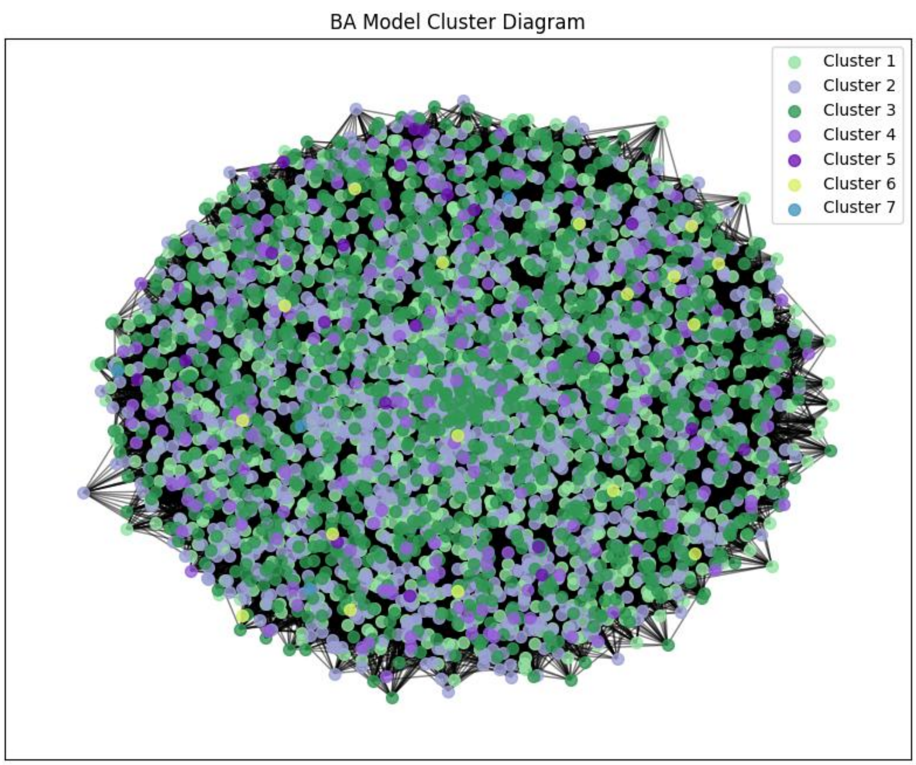

Fig3: BA Model visualization

4.3 Watts-Strogatz (WS) Model

The WS model manages to maintain high clustering while keeping path lengths short in its operation as a small-world network structure. The cluster coefficient measurement from WS model analysis demonstrated results that matched those of Facebook network analysis.

```
BA = nx.barabasi_albert_graph(n, max(1, int(np.mean([deg for _, deg in G.degree()]) / 2))
print(f"BA Model - Clustering Coefficient: {nx.average_clustering(BA)}, Density: {nx.density(BA)}")
```

WS Model Results:

- Clustering Coefficient: **0.5329**
- Density: **0.0104**

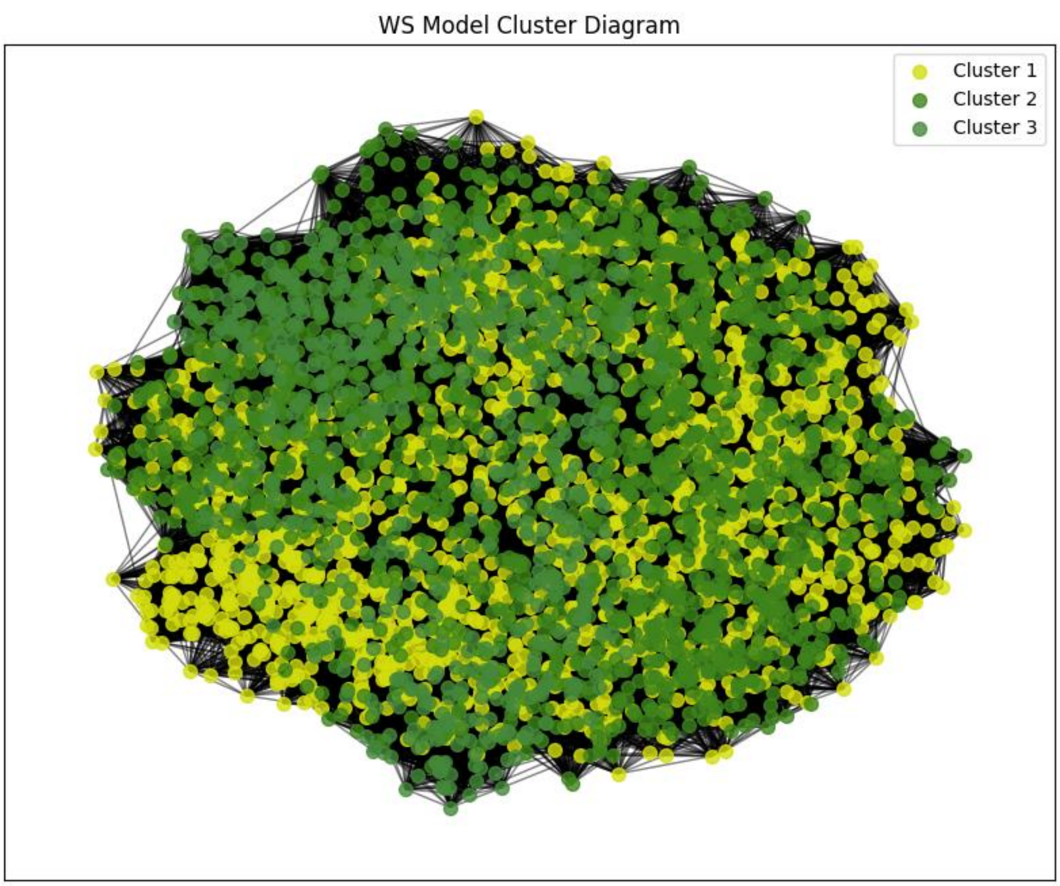

Fig4: WS Model visualization

4.4 Visualization of Random Graphs

Visualizations of the ER, BA and WS models allowed comparison of their structures to the Facebook network topology.

```
fig, axes = plt.subplots(1, 3, figsize=(18, 6))
for ax, model, name in zip(axes, [ER, BA, WS], ["Erdős-Rényi (ER)", "Barabási-Albert (BA)", "Watts-Strogatz (WS)"]):
    nx.draw_spring(model, node_size=20, alpha=0.6, ax=ax)
    ax.set_title(name)
plt.show()
```

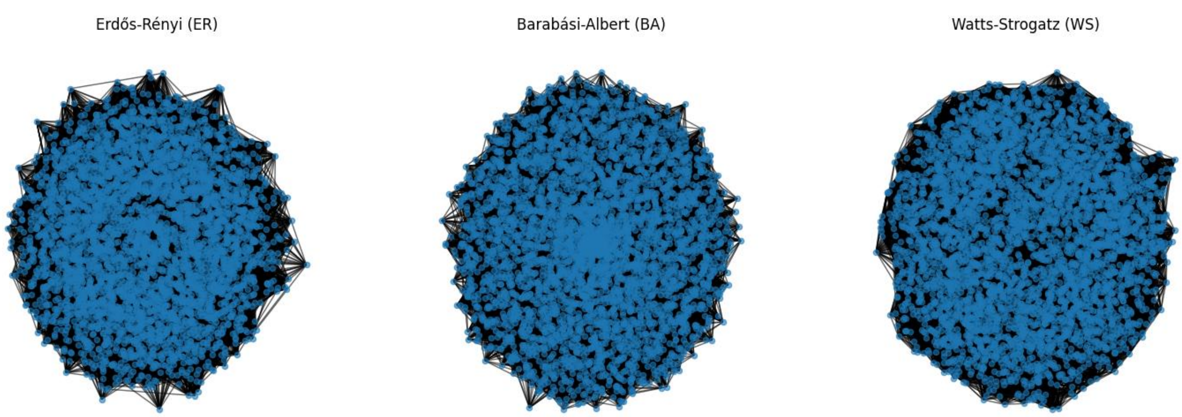

Fig5: Comparison of all models

The WS model matches Facebook network clustering patterns better than ER and BA models according to visual inspection results.

5. Research Question

5.1 Research Question

The research question: **"How does the structure of the Facebook social network influence the spread of information?"**

5.2 Analysis

The network analysis used degree centrality measures to locate the most influential nodes in the network system. The research analyzed the network roles of the selected top 10 central nodes. The selected nodes function as influential elements due to their extensive number of network connections.

5.3 Conclusion

The identification of important social network users relies on effective centrality measurement methods. The core nodes of the Facebook network likely serve as key elements for spreading information throughout the network.

6. Conclusion

The research used Python and networkx to perform an extensive evaluation of the Facebook network. The analysis examined the constructed network with attention to clustering coefficient density and centrality metrics. The Facebook network matched the cluster patterns of the Watts-Strogatz model when researchers compared it to random graph models. Researchers used centrality measures to answer the research question about identifying influential users as a final step.

7. GitHub Repository

Code is available on the following GitHub repository: https://github.com/Herr-Mahin/B107

References:

- NetworkX Documentation: https://networkx.org/
- Facebook Combined Dataset: https://snap.stanford.edu/data/ego-Facebook.html
- Erdős-Rényi Model: https://en.wikipedia.org/wiki/Erdős–Rényi_model
- Barabási-Albert Model: https://en.wikipedia.org/wiki/Barabási–Albert_model
- Watts-Strogatz Model: https://en.wikipedia.org/wiki/Watts–Strogatz_model