Experiment No: 4

Name: Alston Fernandes **Roll No. : 19**

Batch: C **Performance Date : 07/03/2024**

Topic:	Exploratory Data Analysis and visualization of Social Media Data for any network.
Prerequisite:	Knowledge of Social Media concepts, Network measures
Mapping With COs:	CSDL8023.4
Objectives:	- To develop and study social network graphs using NetworkX.
Outcomes:	Students should be able to manipulate and study social network graphs.
Instructions:	This experiment is a compulsory experiment. All the students are required to perform this experiment individually.
Deliverables:	**Submission on elearn :** 1. Design and develop your own social network based on friendship, similar interests etc. **Consider a network …** In this social network, friendships are formed through shared interests, mutual acquaintances, and common goals. Hrishikesh and Sahil Shelke became friends through their shared passion for web development and machine learning. Sahil, in turn, introduced Ambuj Pandey to Hrishikesh, fostering a connection based on their interests in C++ programming and artificial intelligence. Leona, with her expertise in Java and game development, found common ground with both Sahil and Ambuj, solidifying their friendships. Siddhanth, intrigued by Gaurav Samanta's knowledge in networking and cybersecurity, formed a bond with him. Siddhanth's friend, Shrikrishna, shares his interest in mobile app development and UI/UX design, leading to their friendship. Shrikrishna, in turn, introduced Akshay Shekate, Ashish, and Ashmit to the group, forging connections rooted in their interests in cloud computing and big data. Additionally, Pratik and Kaushik betmutha bonded over their shared fascination with cloud computing and big data analytics. Aleron and Alston, drawn together by their expertise in networking and game development, established a close friendship. Alston also connected with Anvith, who shares his passion for Ruby programming and big data analysis.

Code:	**Cell 1:**
	import networkx as nx
	import matplotlib.pyplot as plt
	
	class User:
	  def __init__(self, user_id, name, interests=None):
	  self.user_id = user_id
	  self.name = name
	  self.interests = interests if interests is not None else []

```
class SocialNetwork:
    def __init__(self):
        self.users = {}
        self.friendships = {}

    def add_user(self, user):
        self.users[user.user_id] = user
        self.friendships[user.user_id] = set()

    def add_friendship(self, user_id1, user_id2):
        self.friendships[user_id1].add(user_id2)
        self.friendships[user_id2].add(user_id1)

    def get_friends(self, user_id):
        return self.friendships[user_id]

# Create the social network
social_network = SocialNetwork()

# Add users
users_data = {
    1: ("Hrishikesh", ["Java", "Web Developement"]),
    2: ("Sahil Shelke", ["Web Development", "Machine Learning"]),
    3: ("Ambuj", ["C++", "Artificial Intelligence"]),
    4: ("Boris", ["Java", "Web Development"]),
    5: ("Siddhanth", ["JavaScript", "Frontend Development"]),
    6: ("Shrikrishna", ["Networking", "Cybersecurity"]),
    7: ("Siddhesh", ["Mobile App Development", "UI/UX Design"]),
    8: ("Akshay Shekate", ["Cloud Computing", "Big Data"]),
    9: ("Ashish", ["Distributed Computing", "Data Science"]),
    10: ("Bipin", ["Python", "Data Science"]),
    11: ("Sahil Godse", ["Cloud Computing", "Big Data"]),
    12: ("Shubham", ["JavaScript", "UI/UX Design"]),
    13: ("Aleron", ["Networking", "Big Data"]),
    14: ("Alston", ["Ruby", "Game Development"]),
    15: ("Om", ["GOLang", "Big Data"]),
}

for user_id, (name, interests) in users_data.items():
    user = User(user_id, name, interests)
    social_network.add_user(user)

# Add friendships
friendships = [
    (1, 2),(1, 4), (1, 5),(1, 13), (1, 14),
    (2, 3), (2, 4),
    (3, 4), (3, 15),
    (4, 5),
    (5, 6),(5, 11), (5, 12),
    (6, 7),
    (7, 8), (7, 9), (7, 10),
    (8, 9),
    (9, 10),
```

```
    (11, 12),
    (13, 14),
    (14, 15)
]

for user_id1, user_id2 in friendships:
    social_network.add_friendship(user_id1, user_id2)

# Create a NetworkX graph
graph = nx.Graph()
for user_id, friends in social_network.friendships.items():
    for friend_id in friends:
        graph.add_edge(user_id, friend_id)

# Draw the graph
plt.figure(figsize=(8, 6))
nx.draw(graph, with_labels=True, font_weight='bold', node_size=1000,
node_color='lightblue')
plt.title("Social Network Graph")
plt.show()
```

Cell 2:

```
# Export the graph to XML format
nx.write_graphml(graph, "social_network_graph.xml")
```

Cell 3:

```
# Calculate centrality measures
degree_centrality = nx.degree_centrality(graph)
betweenness_centrality = nx.betweenness_centrality(graph)
closeness_centrality = nx.closeness_centrality(graph)
```

Cell 4:

```
#Printing Centrality measures
from tabulate import tabulate
# Data
centrality_measures = []
for node in range(1, 16):
    centrality_measures.append((f"Node {node}", degree_centrality[node],
betweenness_centrality[node], closeness_centrality[node]))

print("Centrality Measures:")
print(tabulate(centrality_measures, headers=["Node", "Degree Centrality",
"Betweenness Centrality", "Closeness Centrality"], floatfmt=".2f"))
```

Cell 5:

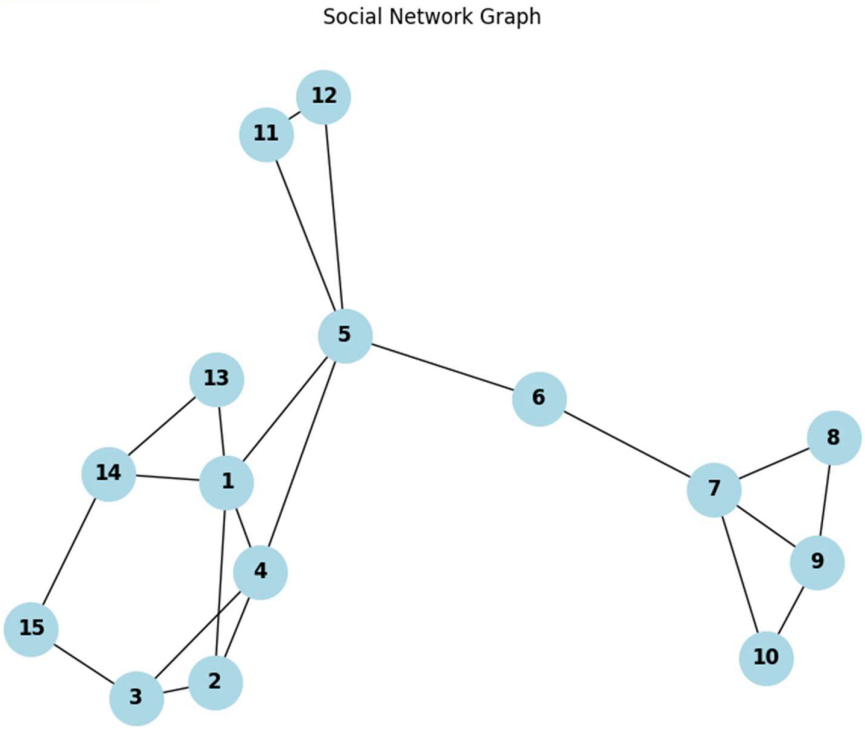
```
# Calculate centralization and density
degree_centralization = max(degree_centrality.values()) -
sum(degree_centrality.values())
density = nx.density(graph)
```

Cell 6:

```
# Graph metrics
graph_metrics = [
    ("Degree Centralization", degree_centralization),
    ("Density", density)
]

print("\nGraph Metrics:")
print(tabulate(graph_metrics, headers=["Metric", "Value"], floatfmt=".2f"))
```

Output of cell 1:



Output of cell 2

```
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
<graph edgedefault="undirected">
<node id="1"/>
<node id="2"/>
<node id="4"/>
<node id="5"/>
<node id="13"/>
<node id="14"/>
<node id="3"/>
<node id="15"/>
<node id="6"/>
<node id="11"/>
<node id="12"/>
<node id="7"/>
<node id="8"/>
```

```
<node id="9"/>
<node id="10"/>
<edge source="1" target="2"/>
<edge source="1" target="4"/>
<edge source="1" target="5"/>
<edge source="1" target="13"/>
<edge source="1" target="14"/>
<edge source="2" target="3"/>
<edge source="2" target="4"/>
<edge source="4" target="3"/>
<edge source="4" target="5"/>
<edge source="5" target="6"/>
<edge source="5" target="11"/>
<edge source="5" target="12"/>
<edge source="13" target="14"/>
<edge source="14" target="15"/>
<edge source="3" target="15"/>
<edge source="6" target="7"/>
<edge source="11" target="12"/>
<edge source="7" target="8"/>
<edge source="7" target="9"/>
<edge source="7" target="10"/>
<edge source="8" target="9"/>
<edge source="9" target="10"/>
</graph>
</graphml>
```

Output of cell 4

Centrality Measures:

Node	Degree Centrality	Betweenness Centrality	Closeness Centrality
Node 1	0.36	0.32	0.47
Node 2	0.21	0.01	0.36
Node 3	0.21	0.07	0.35
Node 4	0.29	0.18	0.45
Node 5	0.36	0.65	0.52
Node 6	0.14	0.44	0.44
Node 7	0.29	0.37	0.36
Node 8	0.14	0.00	0.27
Node 9	0.21	0.01	0.28
Node 10	0.14	0.00	0.27
Node 11	0.14	0.00	0.36
Node 12	0.14	0.00	0.36
Node 13	0.14	0.00	0.34
Node 14	0.21	0.07	0.36
Node 15	0.14	0.01	0.29

Output of cell 6

Graph Metrics:

Metric	Value
Degree Centralization	-2.79
Density	0.21

	Q5) Share your network analysis findings based on above centrality measures. 1. **Degree Centrality:** • Nodes 1, 5, and 4 have the highest degree centrality, indicating they are connected to a large number of other nodes in the network. • Nodes 6, 8, 9, 10, 11, 12, 13, and 15 have relatively lower degree centrality, suggesting they have fewer connections compared to other nodes. 2. **Betweenness Centrality:** • Node 5 has the highest betweenness centrality, indicating it plays a critical role in connecting other nodes in the network. • Nodes 1, 4, 6, and 7 also have notable betweenness centrality, suggesting they serve as important bridges or intermediaries between different parts of the network. 3. **Closeness Centrality:** • Node 5 has the highest closeness centrality, indicating it is the most central node in terms of proximity to other nodes in the network. • Nodes 1, 4, 6, and 7 also have relatively high closeness centrality, suggesting they are well-connected and can quickly interact with other nodes in the network. 4. **Graph Metrics:** • The degree centralization value of -2.79 suggests that the distribution of connections among nodes in the network is relatively uneven, with a few nodes having significantly higher degrees compared to others. • The density of 0.21 indicates that the network is relatively sparse, with only 21% of possible edges present in the graph. This suggests that not all nodes are directly connected to each other, leading to a more modular or decentralized structure.
Conclusion:	Students will be able to develop Structure based social media analytics models for any business using NetworkX.
References:	https://networkx.org/documentation/stable/tutorial.html#analyzing-graphs https://www.cantorsparadise.com/measuring-network-centrality-2a76b0045410

Don Bosco Institute of Technology

Department of Computer Engineering

Assessment Rubric for Experiment No. 4

Title of Experiment : Exploratory Data Analysis and visualization of Social Media Data for any network.

Year and Semester : 4th Year and VIIIth Semester

Sr. No.	Criteria	1 Marks	2 Marks	3 Marks	4 Marks	5 Marks
1	Productivity	Not Satisfactory	Satisfactory	Good	Very Good	Excellent
2	Performance (Implementation)	Not Satisfactory	Satisfactory	Good	Very Good	Excellent
3	Viva	Satisfactory	Good	Very Good		
4	Submission on Time	Submitted after the given deadline	Submitted before the given deadline			