PCC-DS 391 Lab Assignment 9

Topic: Social Network Analysis

1. Write a program to create a simple social network graph, calculate centrality, and visualize it.

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

# Create a social network graph
G = nx.Graph()

# Add nodes (individuals)
G.add_nodes_from(["Alice", "Bob", "Charlie", "David", "Emma", "Frank"])

# Add edges (relationships)
G.add_edges_from([("Alice", "Bob"), ("Alice", "Charlie"), ("Bob", "David"),
            ("Charlie", "David"), ("Emma", "David"), ("Frank", "Alice")])

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

print("Degree Centrality:", degree_centrality)
print("Betweenness Centrality:", betweenness_centrality)
print("Closeness Centrality:", closeness_centrality)

# Draw the graph with labels
plt.figure(figsize=(8, 6))
pos = nx.spring_layout(G)
nx.draw(G, pos, with_labels=True, node_size=1000, node_color="lightblue", font_size=10,
font_weight="bold")
plt.title("Social Network Graph")
plt.show()
```

2. Write a program to compute PageRank on a sample social network.

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

# Create a directed graph (since PageRank is typically applied to directed graphs)
G = nx.DiGraph()

# Add edges (relationships) representing following relationships in a social network
edges = [("Alice", "Bob"), ("Alice", "Charlie"), ("Bob", "Alice"),
    ("Charlie", "Bob"), ("David", "Alice"), ("Eve", "David"),
    ("Eve", "Bob"), ("Frank", "Eve"), ("Frank", "Charlie")]
G.add_edges_from(edges)
```

PCC-DS 391 Lab Assignment 9

```
# Calculate PageRank
pagerank_scores = nx.pagerank(G, alpha=0.85)  # alpha is the damping factor
print("PageRank Scores:", pagerank_scores)

# Visualize the network with PageRank values
plt.figure(figsize=(10, 8))
pos = nx.spring_layout(G)
nx.draw(G, pos, with_labels=True, node_size=1000, node_color="skyblue", font_size=12, font_weight="bold")

# Display the PageRank values as node labels
pagerank_labels = {node: f"{rank:.2f}" for node, rank in pagerank_scores.items()}
nx.draw_networkx_labels(G, pos, labels=pagerank_labels, font_color="red")

plt.title("Social Network with PageRank Scores")
plt.show()
```

3. Write a program to use the Louvain method for community detection in a sample social network.

```
import networkx as nx
import matplotlib.pyplot as plt
import community as community_louvain  # Import the Louvain method

# Create a sample social network graph
G = nx.karate_club_graph()  # Using the classic karate club graph for example purposes

# Apply the Louvain method for community detection
partition = community_louvain.best_partition(G)

# Visualize the communities
plt.figure(figsize=(10, 8))
pos = nx.spring_layout(G)

# Assign colors to nodes based on the detected communities
colors = [partition[node] for node in G.nodes()]
nx.draw(G, pos, node_color=colors, with_labels=True, node_size=500, cmap=plt.cm.rainbow)

plt.title("Community Detection using Louvain Method")
plt.show()

# Print community structure
print("Detected communities:")
for community_id in set(partition.values()):
   community_nodes = [node for node, community in partition.items() if community == community_id]
   print(f"Community {community_id}: {community_nodes}")
```

install python-louvain package for the above code.