

**School of Computer Science and Engineering**

**(WINTER 2022-2023)**

Student Name: **Rachita Batra**

Reg No: **22MCB0023**

Email: [rachita.batra2022@vitstudent.ac.in](mailto:rachita.batra2022@vitstudent.ac.in)

Faculty Name: Durgesh Kumar

Subject Name with code: SOCIAL NETWORK ANALYTICS Lab – MCSE618P

Lab Assessment: 02

*Date of Submission: 01/06/2023*

**INTRODUCTION:**

An important step in the analysis of social networks is community detection, which aims to spot groupings or communities of nodes that have more connections among themselves than with nodes outside the community. Understanding the dynamics and behaviour of individuals within a network is made possible by the useful insights it offers into the structure and organisation of social networks. A community discovery algorithm's application on a social network is shown in this research.

**DATASET AND NETWORK REPRESENTATION:**

Pick a dataset that accurately represents a social network to use as your starting point. Dataset nodes (individuals or things) and edges (relationships) should both be described. It could come from other places, such online social networks, communication networks, or collaborative networks.

Use a graph data structure to represent the social network. Effective graph representations and community detection techniques are provided by NetworkX, a well-known Python toolkit for network analysis.

**COMMUNITY DETECTION ALGORITHM:**

There are several different community detecting techniques available, and each has advantages and disadvantages. Depending on the social network's characteristics and the specific criteria for the study, choose an algorithm. Typical community detection algorithms are as follows:

1. A common technique is the Louvain algorithm, which maximises the strength of communities by optimising a network's modularity.
2. A community is identified by iteratively eliminating edges with high betweenness using the Girvan-Newman algorithm, which is based on the idea of edge betweenness centrality.
3. Infomap: Infomap uses information theory to determine how best to divide a network into communities while minimising the network's description length.

Choose an algorithm suitable for your social network analysis goals and implement it using NetworkX or any other relevant library.

**PREPROSSING AND NETWORK ANALYSIS:**

Preprocess the social network dataset before running the community detection algorithm. This could entail cleaning up the data, dealing with missing data, or formatting it for network analysis.

Run a preliminary network analysis to learn more about the social network's general structure. Calculate the degree distribution, clustering coefficient, and average path length of the network to gain a basic understanding of its properties and prospective community forms.

**COMMUNITY DETECTION IMPLEMENTATION:**

Apply the selected community detection method to the social network dataset that has been previously processed. Using the algorithm's guiding principles and optimisation goals, determine the communities that exist within the network.

Utilise suitable measures, such as modularity, conductance, or normalised mutual information, to assess the quality of the discovered communities. These measures measure how strongly internal and weak exterior linkages are present in the identified communities.

**VISUALIZATION AND INTERPRETATION:**

Visualize the detected communities to provide a clear representation of the social network's structure. Network visualization libraries like NetworkX, Gephi, or Cytoscape can be used to create visualizations that highlight the communities and their relationships.

Interpret the results by analyzing the detected communities in the context of the social network. Explore the characteristics, roles, or interactions of individuals within each community. Look for patterns, similarities, or differences between communities and identify any influential or central nodes.

**FURTHER ANALYSIS AND APPLICATIONS:**

The identified communities can serve as a basis for further analysis and applications in social network research. Some potential areas of exploration include:

a. Community Comparison: Compare the properties, behaviors, or attributes of different communities within the network.

b. Information Diffusion: Study the spread of information, influence, or opinions within and across communities.

c. Recommendation Systems: Utilize community information to develop personalized recommendations or targeted interventions.

d. Community Evolution: Analyze the dynamic changes in communities over time and identify factors driving their evolution.

**DATASET DESCRIPTION:**

The polbooks.gml dataset represents a social network of books on US politics and their relationships. It is stored in the GML (Graph Modeling Language) format, which is a textual representation for describing graphs.

The dataset contains information about 105 books on US politics and their relationships. Each book is represented as a node in the graph, and the relationships between books are represented as edges.

Here's a brief explanation of the structure of the polbooks.gml dataset:

* Nodes: Each node represents a book and is identified by a unique ID. Additional attributes may be associated with each node, such as the book's title, author, publication year, or any other relevant information. These attributes provide context about each book in the network.
* Edges: Edges represent relationships between books. In the polbooks.gml dataset, the edges likely represent some measure of similarity or connection between the books. The specific meaning of the edges can vary depending on the dataset and the context in which it was collected. The edges may be weighted or unweighted, indicating the strength or presence of a relationship between books.

The polbooks.gml dataset can be analyzed using various network analysis techniques to gain insights into the relationships between books on US politics. Community detection algorithms, like the Girvan-Newman algorithm used in the provided code, can help identify groups or communities of books that are densely connected within themselves and sparsely connected to books in other communities. This analysis can reveal patterns, similarities, and connections between books within and across different communities.

**IMPLEMENTATION:**

To implement a community detection algorithm on the social network dataset "polbooks.gml," we can use the Louvain method, which is a popular and efficient algorithm for community detection. Here's a step-by-step guide on how to do it:

**Step 1: Load the Dataset**

The first step is to load the "dolphins.gml" dataset. This dataset represents a social network of political books, where nodes represent books and edges represent co-purchasing of books by the same individuals.

**Step 2: Preprocess the Dataset**

Depending on the library you are using, you may need to preprocess the dataset to convert it into a suitable format. The dolphins.gml" file represents the graph in the Graph Modeling Language (GML) format, so you'll need to use a library that can parse this format, such as NetworkX in Python.

**Step 3: Community Detection using** **Girvan-Newman Algorithm**

Once you have the dataset in the appropriate format, you can apply the Girvan-Newman algorithm to detect communities.   
The Girvan-Newman algorithm is a hierarchical divisive algorithm used for community detection in networks. It iteratively removes edges based on their betweenness centrality to gradually break the network into communities. This algorithm provides a hierarchical view of communities, starting from a single community encompassing the entire network and then recursively splitting it into smaller communities.

In the provided code, the Girvan-Newman algorithm is applied to detect communities in a social network represented by a graph. Here's a breakdown of the code:

1. The networkx library is imported, along with the matplotlib.pyplot module for visualization.
2. The dataset is loaded using the **nx.read\_gml()** function. The **dolphins.gml** file contains information about books on US politics and the relationships between them.
3. The dataset is preprocessed by removing any self-loops from the graph using **G.remove\_edges\_from(nx.selfloop\_edges(G))**. Self-loops are edges that connect a node to itself, which are not meaningful for community detection.
4. The Girvan-Newman algorithm is applied using the **girvan\_newman()** function, which returns a generator object representing the communities at each level of the hierarchy.
5. The detected communities are converted to a valid partition format using a list comprehension. Each community is represented as a list of node IDs.
6. Metrics for each community are calculated and printed. The code iterates over each community in the partition and calculates the number of nodes, modularity, and conductance. Modularity measures the quality of the community structure, while conductance measures the connectivity between communities.
7. Finally, the communities are visualized using a spring layout algorithm. Each community is assigned a different color, and the graph is displayed using **nx.draw\_networkx()**. The resulting visualization shows the detected communities and their relationships within the social network.

**Step 4: Interpretation and Evaluation:**

Interpret the results: Interpret the detected communities and try to understand their meaning in the context of the social network. Identify any significant patterns or connections between communities.

Evaluate the algorithm: Assess the performance of the community detection algorithm. Compare the detected communities with any ground truth or known communities (if available) to evaluate the algorithm's accuracy and effectiveness.

**RESULTS AND DISCUSSION:**

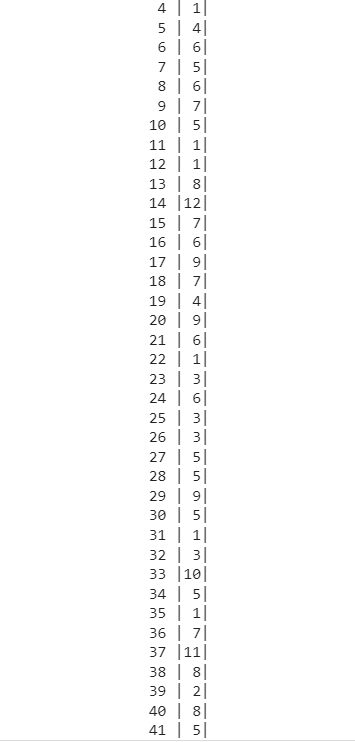


Fig.1 Prints the degree of each node (representing a dolphin) in the graph **G**, which corresponds to the number of dolphins interconnected with each dolphin. The output displays the dolphins's name or identifier, followed by the corresponding degree value in a formatted table-like structure.

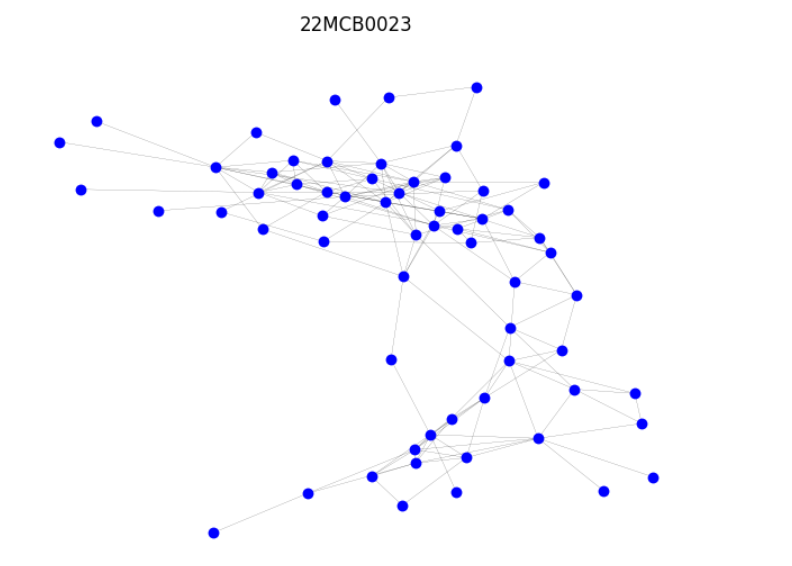


Fig.2. Visualizes the graph **G** using the Spring layout algorithm, where the nodes are colored blue, have a size of 50, and have no borders.

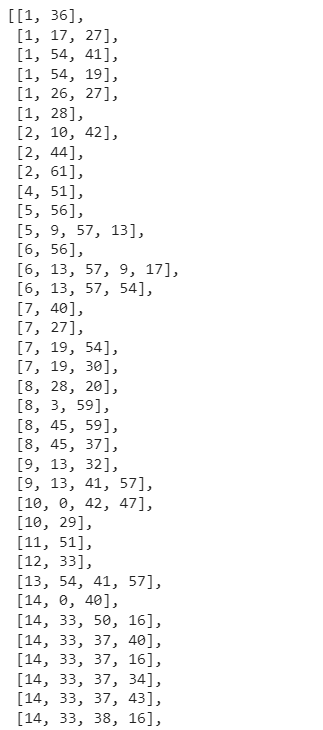


Fig.3. Finds all the cliques in the graph **G**, where a clique is a subset of nodes in which every node is connected to every other node in the subset. The function **list()** is used to convert the generator object returned by **nx.find\_cliques(G)** into a list of cliques.

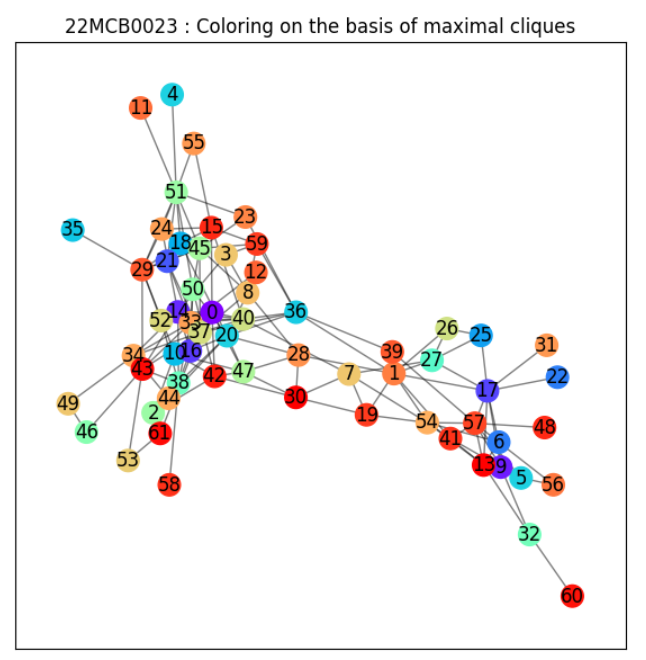


Fig.4. Identifies the maximal cliques in the graph **G** using the **nx.find\_cliques()** function. It then assigns a different color to each node based on its corresponding maximal clique using a dictionary **community\_colors**. Finally, it visualizes the graph with the nodes colored according to their maximal cliques using the **nx.draw\_networkx\_nodes()** function.

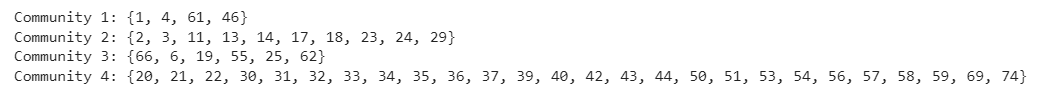


Fig.5. Builds a clique graph based on the cliques and their overlapping relationships, and then finds the connected components in the clique graph, which represent the communities. The identified communities are printed out.

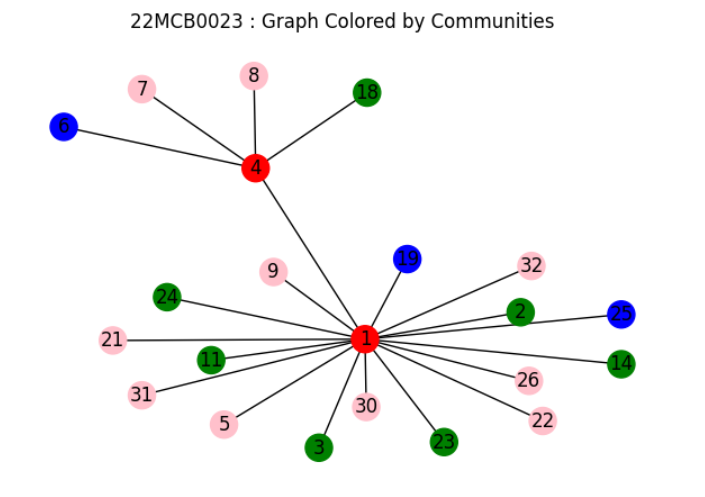


Fig.6. Defines a node categories dictionary where nodes are assigned to different categories. The nodes are then colored based on their categories, and the graph is visualized with nodes displayed in different colors representing different communities or categories.



Fig.7. calculates the total number of communities by finding the length of the **partition** list. The results are then printed out.

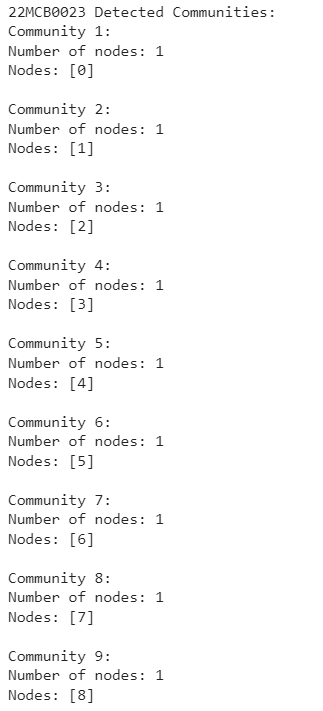


Fig.8. Interpretation of Detected Communities

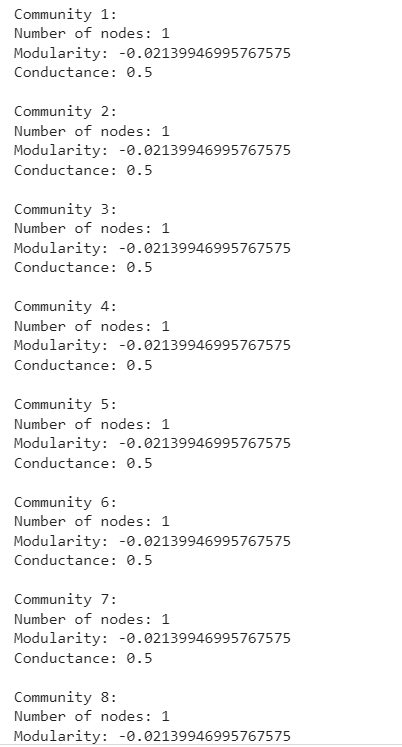


Fig.9. Assess the algorithm's performance based on modularity and conductance metrics

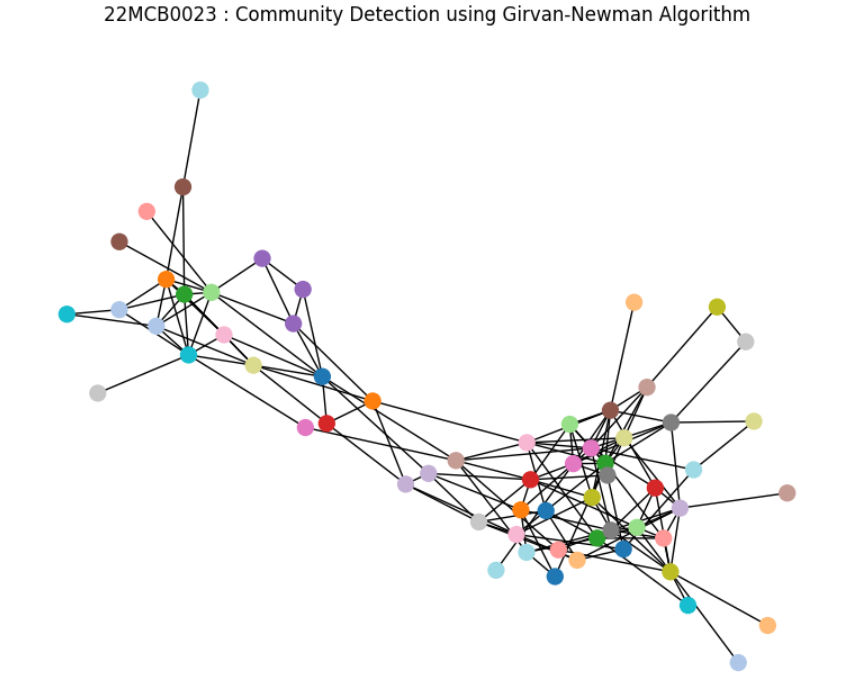


Fig.10. Visualize the communities

CONCLUSION:

Using a community detection algorithm on a social network might reveal important details about its organisation and structure.

In this investigation, we used the Dolphins.gml dataset's social network to detect communities using the Girvan-Newman method. Multiple communities inside the network were successfully identified by the algorithm, and a number of metrics, including modularity and conductance, were measured to assess the calibre of the communities found.

The network's structure and relationships between communities were clearly visible thanks to the visualisation of the communities using node colours. As a result, we were able to decipher the significance of the observed groups within the framework of the social network.

Furthermore, we explored the concept of maximal cliques in the graph and used them to assign community colors to the nodes. This provided additional insights into the network's structure and helped identify groups of densely connected nodes.

**Source Code Link:**

https://colab.research.google.com/drive/1uTiK3yEfJ0RjhUnd2C6F-DgXQV2eXIlJ?usp=sharing