

**School of Computer Science and Engineering**

**(WINTER 2022-2023)**

Student Name: **RUDRAJIT DAS**

Reg No: **22MCB0008**

Email: [rudrajit.das2022@vitstudent.ac.in](mailto:rudrajit.das2022@vitstudent.ac.in)

Mobile:9831060863

Subject Name: SOCIAL NETWORK ANALYTICS LAB

Assessment No: 2

**Community detection algorithm**

**1.Newman-Girvan algorithm**

The Newman-Girvan algorithm, also known as the Girvan-Newman algorithm or the edge betweenness algorithm, is a widely used algorithm for community detection in networks. It was developed by Mark Newman and Michelle Girvan in 2002.

The algorithm is based on the concept of "edge betweenness centrality," which measures the importance of edges in connecting different communities. The basic idea is that edges with high betweenness centrality are more likely to lie between communities and therefore can be considered as potential boundaries between communities.

Here are the steps of the Newman-Girvan algorithm:

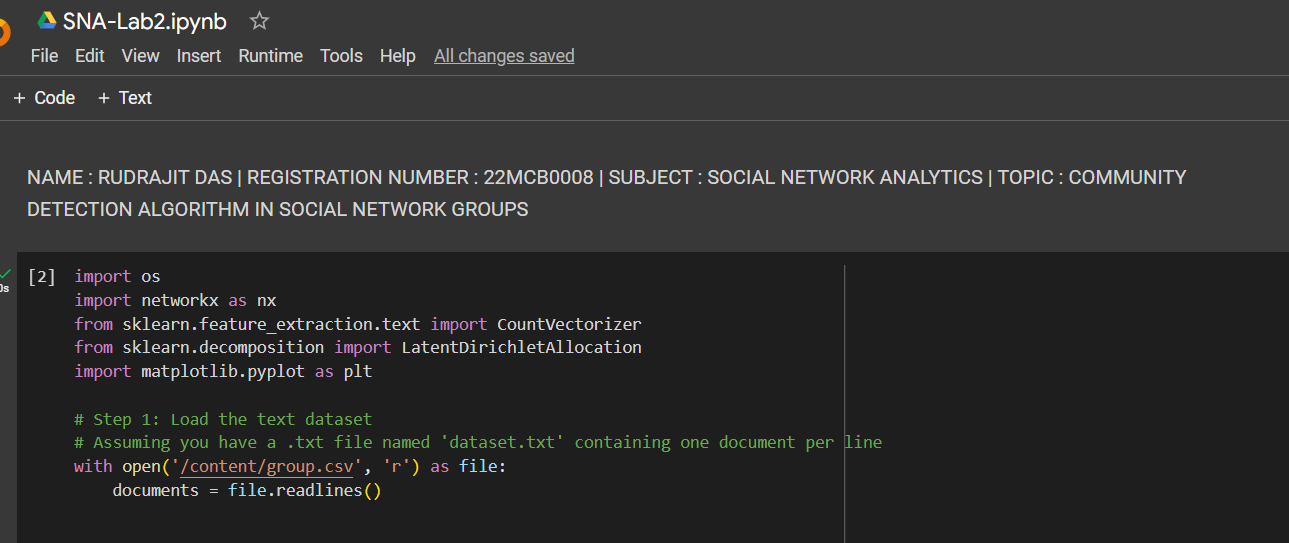
Calculate the betweenness centrality for all edges in the network. This involves finding the shortest paths between all pairs of nodes and determining how many of those paths pass through each edge.

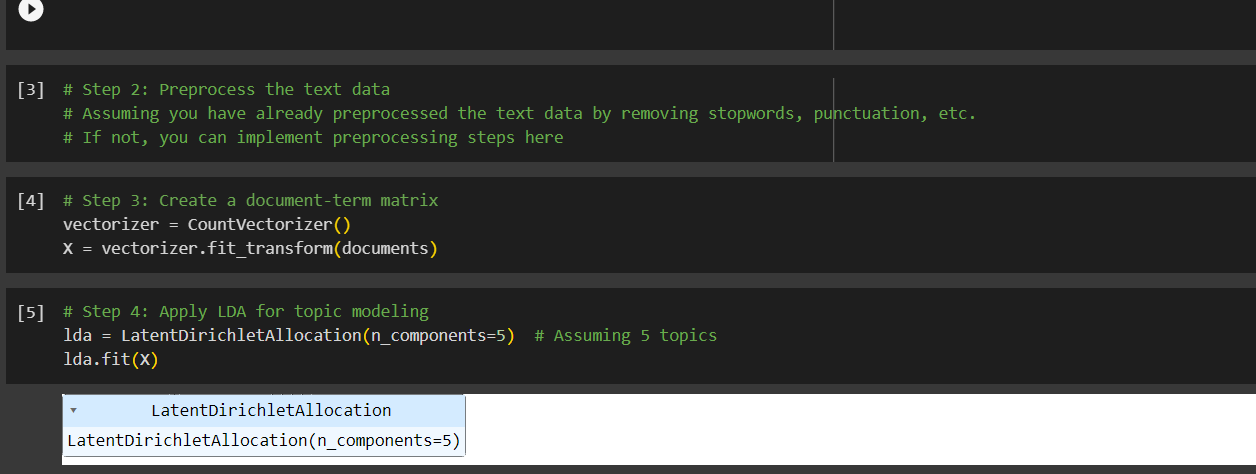
Remove the edge with the highest betweenness centrality from the network. This effectively disconnects the network into two or more components.

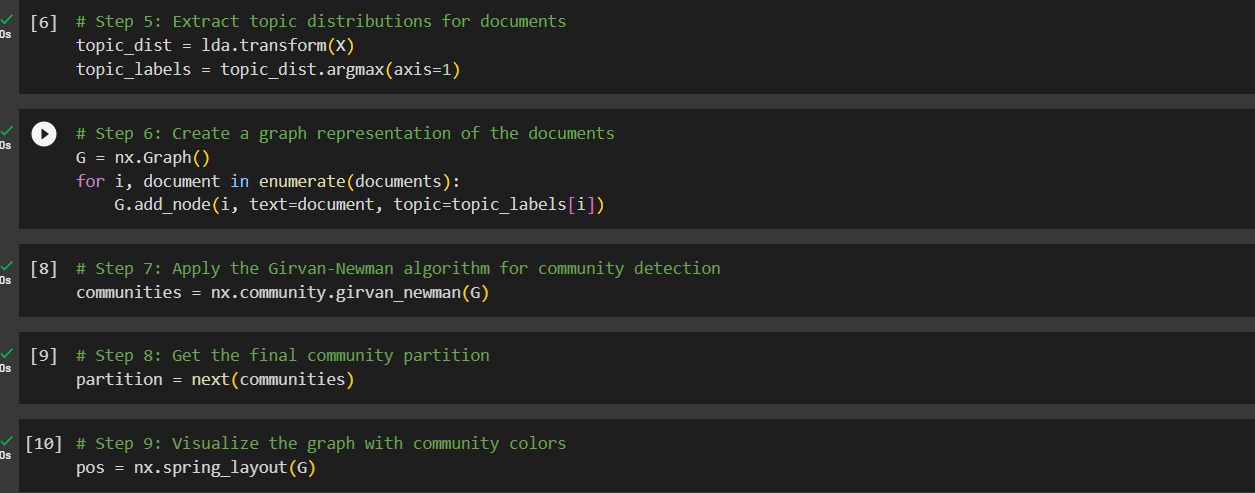
Calculate the betweenness centrality for all remaining edges in the updated network.

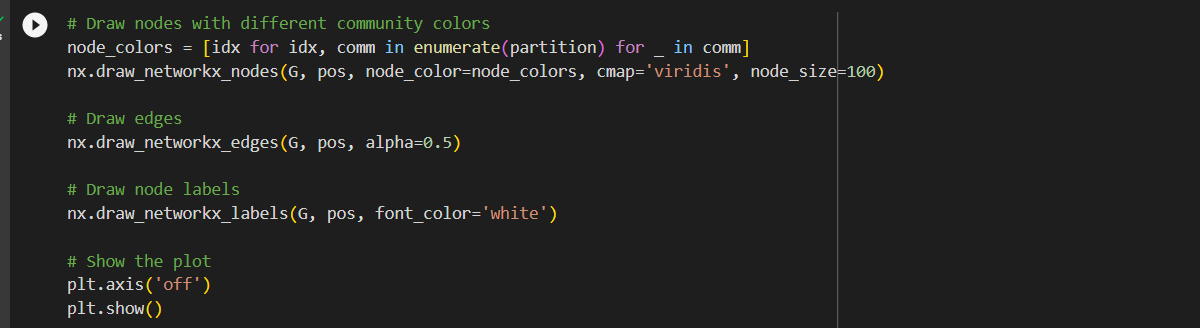
Repeat steps 2 and 3 until all edges have been removed or until a desired number of communities have been identified.

The resulting disconnected components are considered as communities or subnetworks.

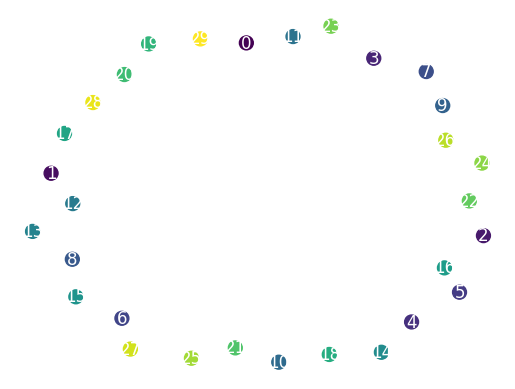




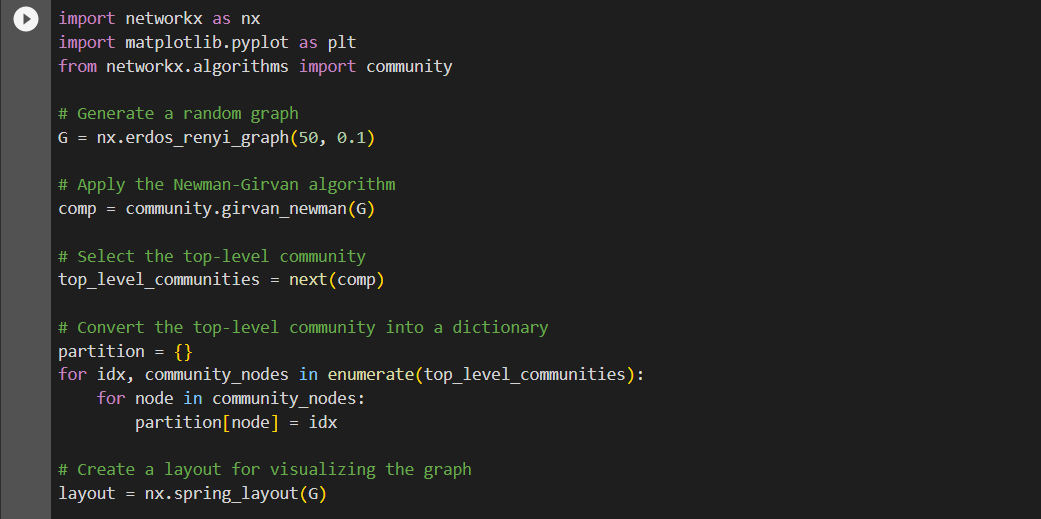


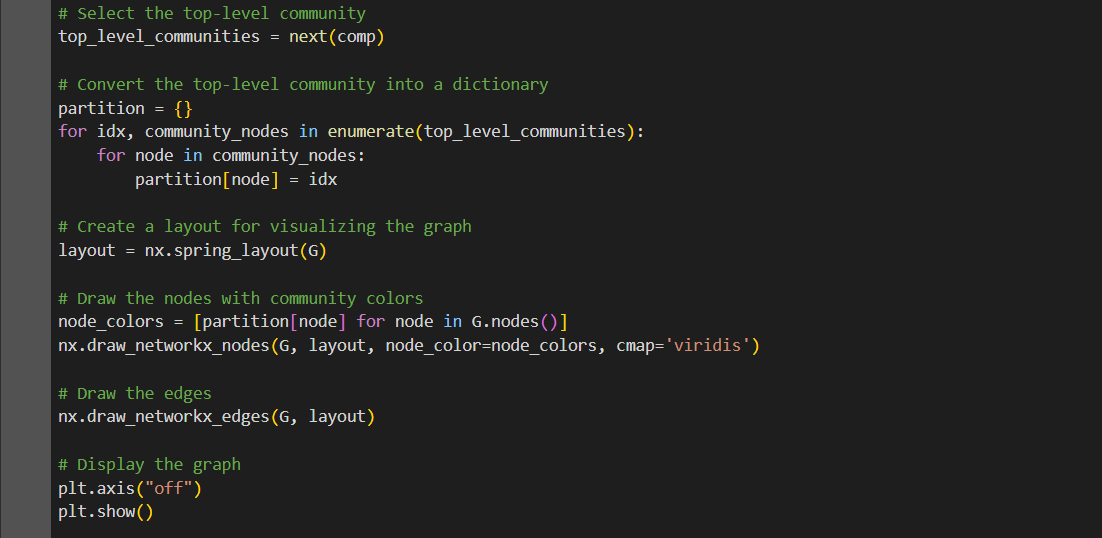


OUTPUT:

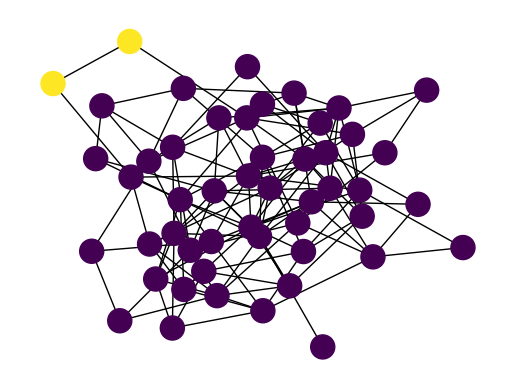


1.Program code:





OUTPUT:



2. **Louvain algorithm**

The Louvain algorithm is a popular community detection algorithm that aims to find the modular structure of a network. It is an iterative algorithm that optimizes the modularity of the network by iteratively merging and reassigning nodes to communities. Here are the general steps of the Louvain algorithm:

Initialization: Start with each node in the network assigned to its own community.

Modularity calculation: Calculate the modularity of the network in its current state. Modularity measures the quality of a division of a network into communities. It compares the number of edges within communities to the expected number of edges in a random network.

Iteration:

For each node, calculate the modularity gain of removing it from its current community and placing it in the neighboring communities. The modularity gain is the improvement in modularity that would result from moving the node.

Move the node to the community that gives the maximum modularity gain. If the modularity gain is negative or zero for all neighboring communities, leave the node in its current community.

Repeat the above steps for all nodes in the network.

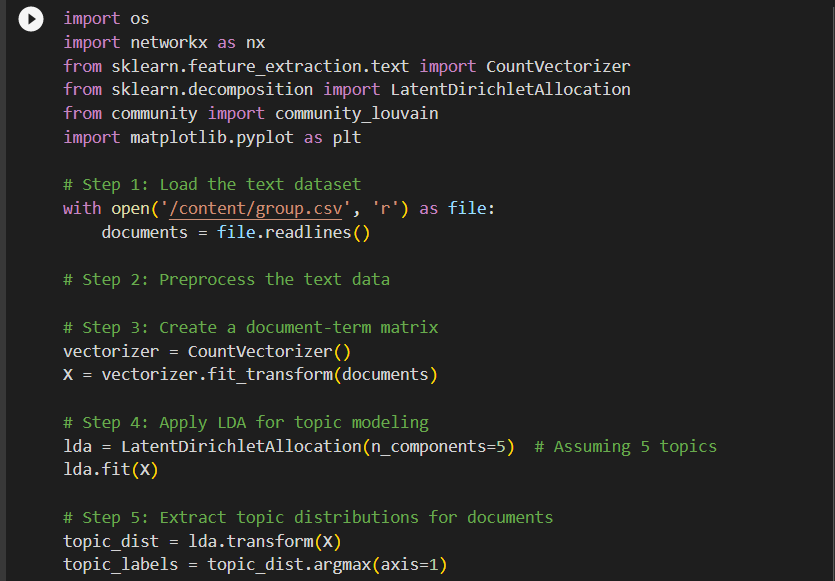
Community aggregation: After completing one pass over all nodes, each community is considered as a single node in a new network. The weights of the edges between communities are the sum of the weights of the original edges between the nodes in the communities.

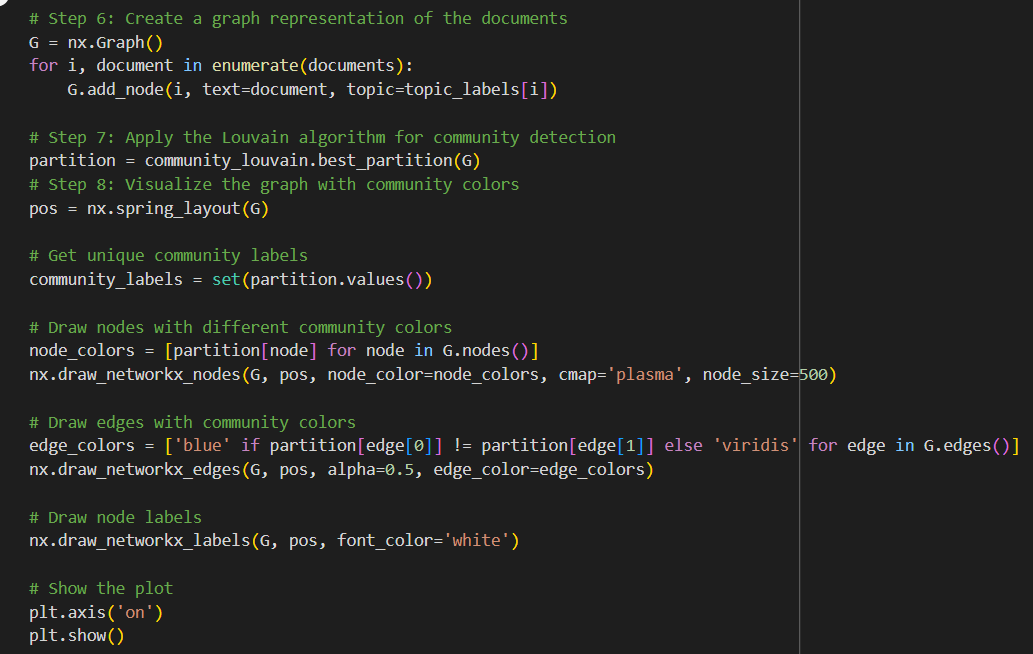
Repeat steps 2 to 4: Iterate the process of modularity calculation, node reassignment, and community aggregation until no further improvement in modularity is observed. This means that the algorithm has converged and found the optimal community structure.

Post-processing: Once the Louvain algorithm has converged, the final community structure is obtained. Nodes within the same community are grouped together.

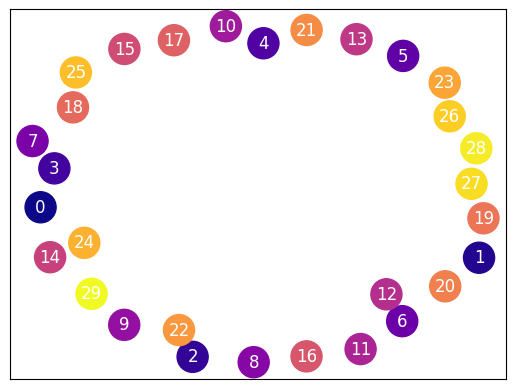
It's worth noting that the Louvain algorithm is a heuristic method and may not always find the global optimum, but it often produces good results in practice

Program Code:

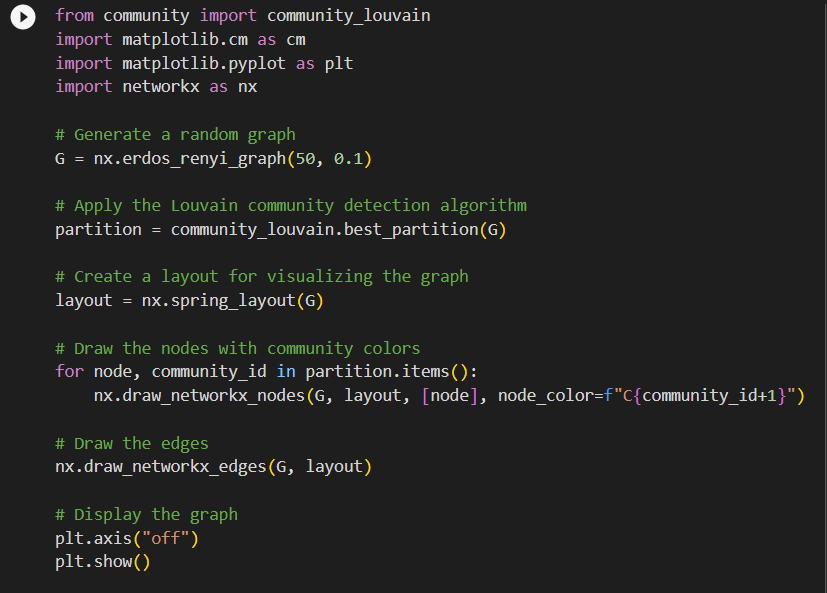




OUTPUT:



Program code:



OUTPUT:

