

Machine Learning - IV

JHANVI PAREKH

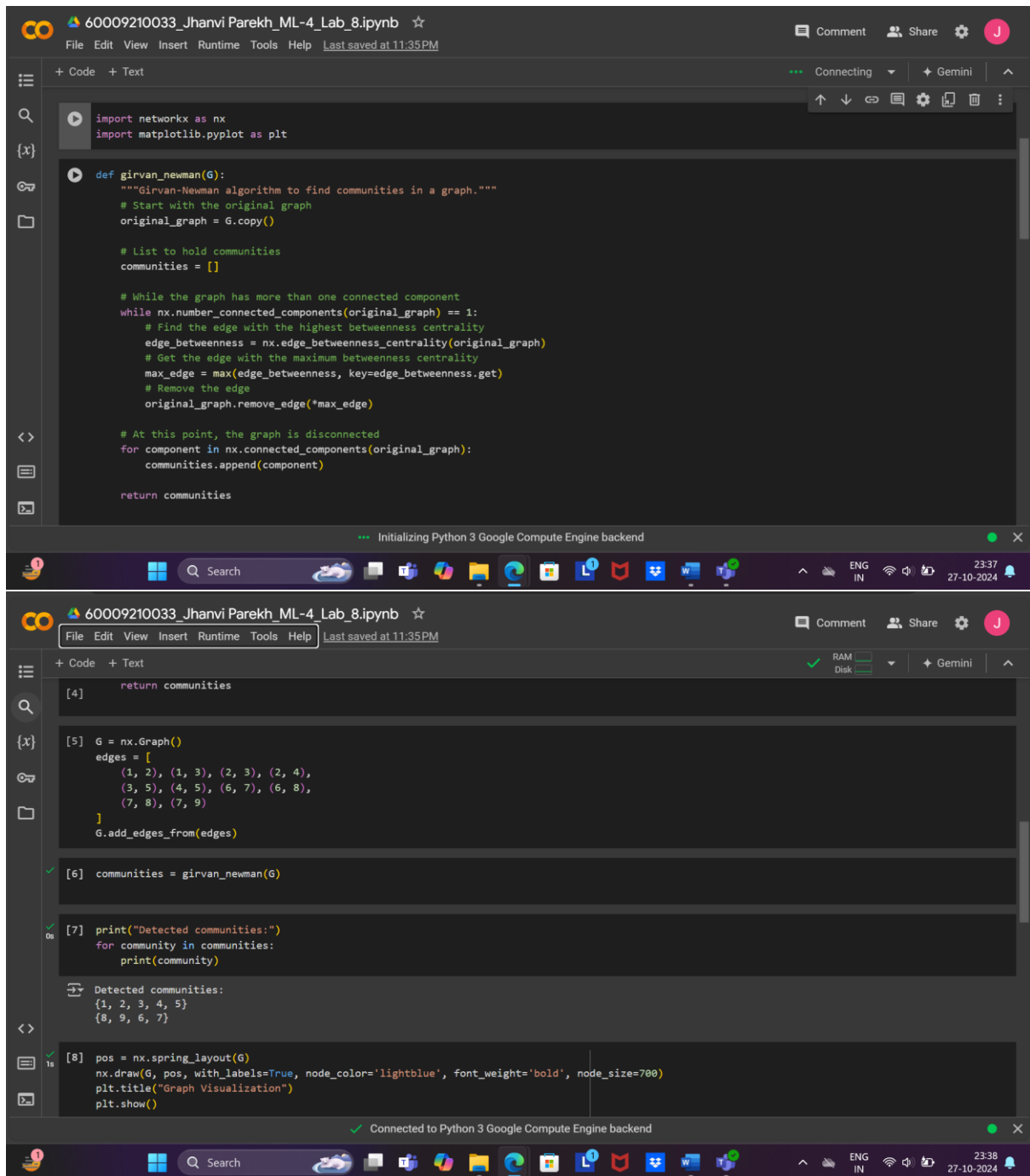
60009210033

CSE (DS)

Experiment 8

Link:

<https://colab.research.google.com/drive/1yRUjvNIQ7IVs46dDuHSbpelWUyfHeUGF?usp=sharing>



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

def girvan_newman(G):
    """Girvan-Newman algorithm to find communities in a graph."""
    # Start with the original graph
    original_graph = G.copy()

    # List to hold communities
    communities = []

    # While the graph has more than one connected component
    while nx.number_connected_components(original_graph) > 1:
        # Find the edge with the highest betweenness centrality
        edge_betweenness = nx.edge_betweenness_centrality(original_graph)
        # Get the edge with the maximum betweenness centrality
        max_edge = max(edge_betweenness, key=edge_betweenness.get)
        # Remove the edge
        original_graph.remove_edge(*max_edge)

    # At this point, the graph is disconnected
    for component in nx.connected_components(original_graph):
        communities.append(component)

    return communities

[4] return communities

[5] G = nx.Graph()
    edges = [
        (1, 2), (1, 3), (2, 3), (2, 4),
        (3, 5), (4, 5), (6, 7), (6, 8),
        (7, 8), (7, 9)
    ]
    G.add_edges_from(edges)

[6] communities = girvan_newman(G)

[7] print("Detected communities:")
    for community in communities:
        print(community)

Detected communities:
{1, 2, 3, 4, 5}
{8, 9, 6, 7}

[8] pos = nx.spring_layout(G)
    nx.draw(G, pos, with_labels=True, node_color='lightblue', font_weight='bold', node_size=700)
    plt.title("Graph Visualization")
    plt.show()
```

60009210033_Jhanvi Parekh_ML-4_Lab_8.ipynb

File Edit View Insert Runtime Tools Help

Comment Share Gemini

RAM Disk

1s

Graph Visualization

```
graph LR; 1 --- 2; 2 --- 3; 3 --- 4; 4 --- 5; 5 --- 1; 6 --- 7; 7 --- 8; 8 --- 9;
```

Connected to Python 3 Google Compute Engine backend

Search

ENG IN

23:38 27-10-2024