Joyce Bozeko
joyceb@vt.edu

Assignment 2 CS 5664

1. **Part 1: Facebook community analysis**

a) Modularity analysis

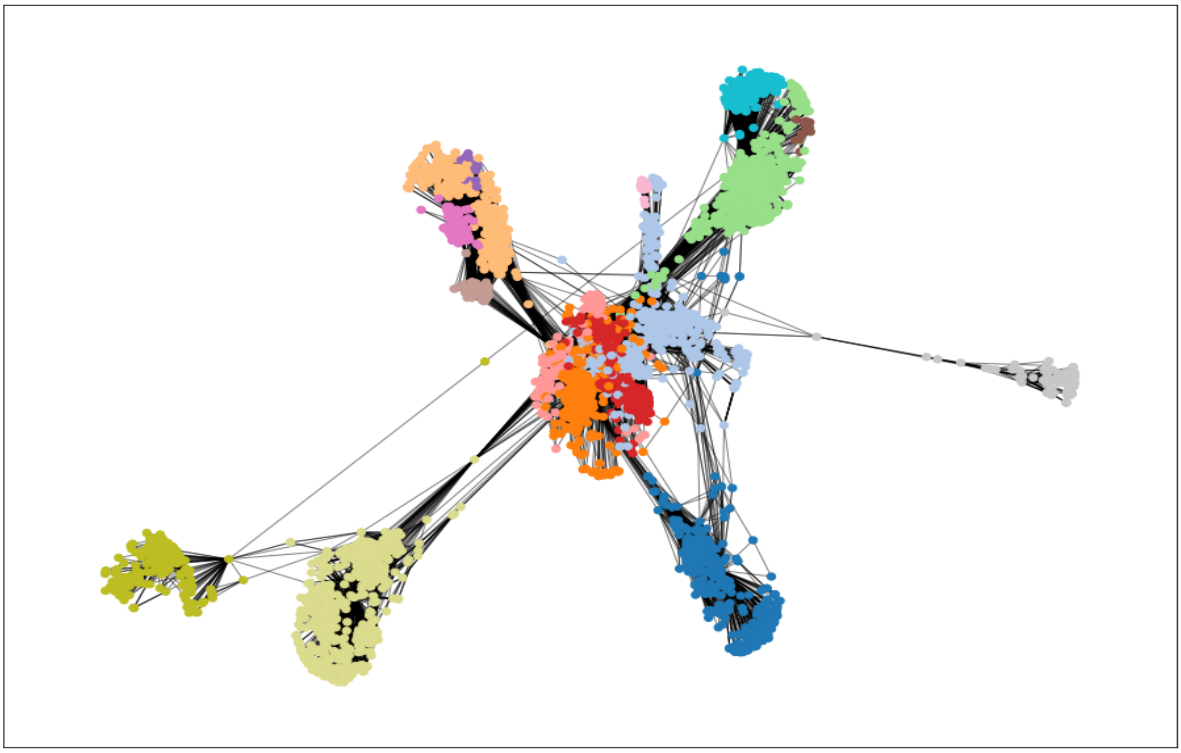
We use the **community. best partition ()** function from the python-Louvain library to detect communities within the graph. This function returns a dictionary where each node is assigned to a community. We then calculate the modularity score using the **community. Modularity ()** function, which takes the community assignment and the graph as input.

The modularity using the **Louvain heuristics** for the Facebook graph is **0.8349291779896184.**

b) Graph clustering

We use the **Network**Xlibrary to visualize the graph, with nodes colored according to their community membership. The nx. spring_layout () function is used to generate a layout of the nodes, and the nx.draw_networkx_nodes () function is used to draw the nodes with different colors based on their community membership. The nx.draw_networkx_edges () function is used to draw the edges between nodes. Finally, the plt. show () function is used to display the graph.

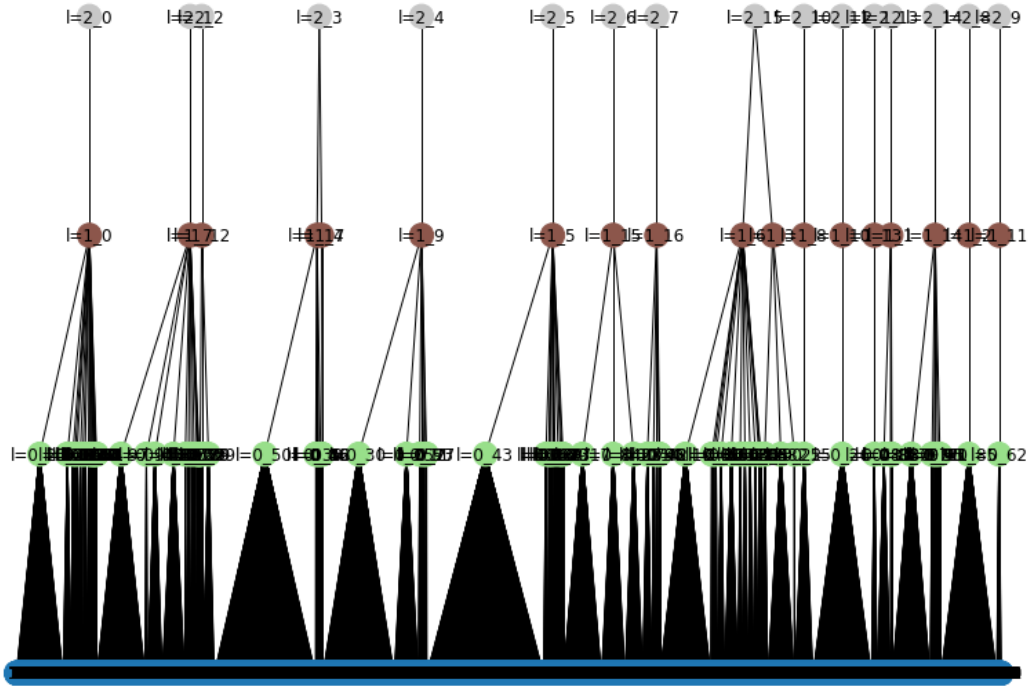
In the following image we can see the **16 clusters** of the graph plotted in different colors



c) hierarchical graph clustering

We create directed graph **pg.** to store the **dendrogram** information. The dendrogram contains the output of the Louvain algorithm, which is a nested dictionary where each key-value pair represents a node and its corresponding community at a given level of the dendrogram. We then iterate through the levels of the dendrogram, creating a new node for each community at that level and connecting it to the nodes corresponding to its sub-communities at the previous level. We then use the **graphviz_layout ()** function to generate a layout of the dendrogram. Nodes are labeled with their community ID and colored according to their level in the dendrogram using the node_color

The figure bellow shows the 3 levels of the hierarchical clustering for the Facebook graph.



d) Finding K-cliques

We tried finding 20 users that connected each to another (20-cliques) in the Facebook network, but the network is too large and the complexity of the algorithm is exponential. That lead us to a huge ram consumption. We then run the code for a random graph with params n=50 (number of nodes in the graph) and p = 0.33 (the probability of an edge between any two nodes)

All 5-cliques in the graph:
[3, 45, 35, 16, 31]
[24, 41, 49, 20, 42]
[24, 41, 49, 20, 26]
[24, 43, 20, 49, 42]
[37, 35, 0, 16, 2]
[37, 35, 0, 16, 45]
[37, 35, 0, 1, 34]
[37, 35, 0, 1, 45]

[37, 35, 0, 2, 34]
[37, 4, 1, 19, 36]

2. Part 2 : Facebook network Plot using the Gephi

Bellow we plot the Facebook network using Gephi with the **Force Atlas 2** algorithm. The node size depends on its number of link in the network. We produced the output shown in the image by filtering only nodes that have at least 20 links (users which have at least 20 friends). That filter reduced the number of visible nodes by 50%, making the graph clearer.

We also computed all important features (degree, density, centrality measures, clustering coefficient …) of the network. All this can be found the Gephi file.