Project Goals

Source Location: Stanford Large Network Dataset Collection: Social Circles from Twitch

Format:

Target (Features):

- ID: unique ID number of the Twitch user
- Days: time duration of account being active on Twitch
- Mature: boolean value dictating whether the account uploads explicit content
- Partner: boolean value dictating whether respective user is partnered with Twitch
- New_ID: new_id created for the user

Edge:

- From: user in question
- To: user that is friended to the user in question

Project Idea

The foundational idea behind our project is to use the Twitch data from the Stanford Large Network Dataset Collection's social circles datasets in order to go through Twitch accounts and analyze which accounts in the set are most likely to follow one another based on a respective account's features. First, we would obtain the data regarding the users in the form of R Documentation, where each user would be represented by a node on the graph that is created, and each of the edges is represented by who a user follows. The graph in itself is undirected, as if two users follow each other, there will simply be two edges between them. Using this information, we would create a graph (again, users are nodes and edges are who is followed). The features of each user are stored in the nodes and the features will be represented as dummy variables (represented as 1 for having the feature and 0 for not having). Each of the edges will then be weighted by some arbitrary means. In order to traverse the unweighted version of the graph, we will use a **Breadth First Search (BFS Traversal) algorithm**, as this method allows us to go through each of the users, or nodes, and collect the needed feature data about them. However, we would then use **Dijkstra's algorithm** to do a traversal of the weighted graph and also to find the nearest neighbor of the current node, despite its worse time-space complexities because of how it can be applied to weighted edges. Afterwards, we will implement a variant of the **PageRank algorithm**. Utilizing this algorithm, the last step will be to find the most likely followers based on common features and the shared edges.