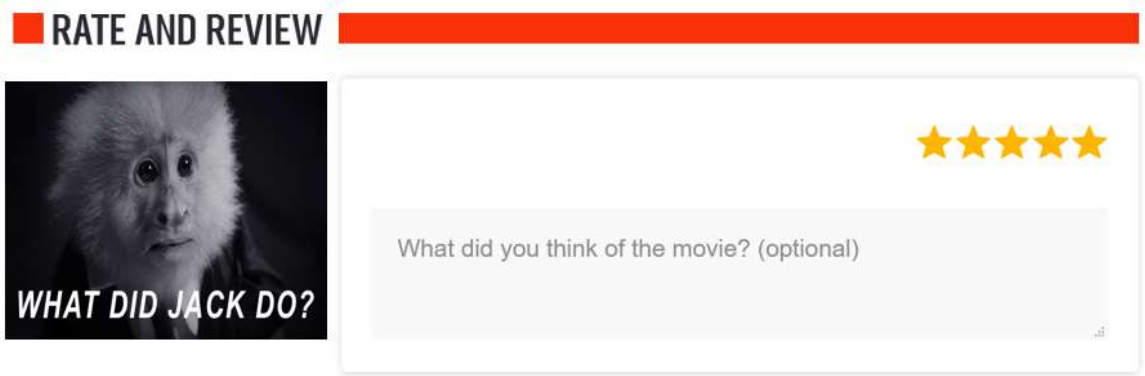
CS 5001 – Applied Social Network Analysis
Spring 2021
HW #3

25 points

Submit to Canvas by 11:59 p.m. on Monday, Feb. 22, 2021



What to Do:

For this assignment you are to write a **Python** program that will **analyze a social network of people who have rated movies.** The network is represented as an undirected graph. There are nodes for people and nodes for movies. Each edge is between a person and a movie; such an edge denotes that the person has seen the movie, and has given it a rating (which should be an attribute of the edge). Yes, this is a bipartite graph.

Posted on Canvas is a text file[1] (**userRatedMovie.csv**) that contains data for this network; there are 17 nodes (7 people and 10 movies) and 23 edges. Each line in the file contains a user's name (name), the title of a movie (title), and the integer rating the user gave to the movie (userRating). Python provides functions to read csv files and create graphs; for example, you could read the csv file into a Pandas dataframe and then build a graph from the dataframe (see Networkx.from_pandas_edgelist).

Here's what you are to do to analyze this network:

(1) Partition the nodes into 2 sets: the people and the movies. **Display the graph** with the nodes for the people in red and the nodes for the movies in blue (if you're color blind, you can pick different colors than red and blue, as long as the colors for each group of nodes is different; we don't want the nodes all the same color!).

(2) **List the "most important" person and the "most important" movie based on each of the following types of centrality: degree, closeness, and betweenness.** If there is a tie for a score, choose one of the nodes in the tie using any method that you want. You must write code to determine the winners in each of the centrality categories; you will lose points if you just look at the centrality results and (manually) pick the highest-valued nodes!

[1] This file was created on a Windows machine. If you're using it on a Unix/Linux machine, you may want to run *dos2unix* on it first to remove any weird characters like '\r'

(3) **List movies that were seen by at least 3 people.** Hint: Consider computing an event-by-actor matrix.

(4) **Create a social network of the people**. There should only be "people" nodes (no "movie" nodes) in this graph. There should be an (undirected) edge between 2 people nodes if the 2 people have seen at least one movie in common; there should not be any self-loops (e.g., Joe should not have an edge to himself). Each edge should have an attribute representing the number of movies in common the 2 people have seen; this represents the "strength" of their social tie. **Display this graph including the edge weights.** Hint: Consider computing an actor-by-actor matrix.

(5) **List pairs of movies that were seen by 2 or more people** (e.g., if "What Did Jack Do?" and "Rabbits" were seen by 4 different people, then output that pair of movies). Don't include duplicate results in your output. Hint: Consider computing an **event-by-event matrix**.

What to Turn In:

You need to submit (via Canvas) the following (all as a **single** pdf file):

- A listing of **your source code**. **COMMENT YOUR CODE** so that the grader can follow what you are doing!!!
- The specified **output** for tasks (1)-(5).

Grading:

Here's how many points each task is worth:

Task	Points Possible
Create graph from csv file (with edge weights)	1
Bipartite graph display	3
Most important movie and user by degree centrality	2
Most important movie and user by closeness centrality	2
Most important movie and user by betweenness centrality	2
Movies seen by at least 3 people	4
Social network of the people	6
Pairs of movies seen by 2 or more people	5
Total	**25**