Project 4: **Movie Recommendations**
Comp 4750 – Web Science
50 points

The goal of this project is to use the basic recommendation principles we have learned to analyze data from MovieLens. MovieLens (movielens.org) is a movie recommendation system, and GroupLens Research makes rating data sets from MovieLens available to the public.

MovieLens Files

Download ml-latest-small.zip from https://grouplens.org/datasets/movielens/. The zip file contains several CSV files. Two are described here:

1. **ratings.csv** – Each line represents one rating of one movie by one user.

   Format: userId,movieId,rating,timestamp (Timestamps are Unix seconds since 1/1/1970 UTC)

   Example:

```
1,1,4,964982703
1,3,4,964981247
1,6,4,964982224
…
610,168252,5,1493846352
610,170875,3,1493846415
```

2. **movies.csv** – Each line represents one movie.

   Format: movieId,title,genres (Genres are pipe-separated list.)

   Example:

```
1,Toy Story (1995),Adventure|Animation|Children|Comedy|Fantasy
2,Jumanji (1995),Adventure|Children|Fantasy
3,Grumpier Old Men (1995),Comedy|Romance
…
193585,Flint (2017),Drama
193587,Bungo Stray Dogs: Dead Apple (2018),Action|Animation
```

Python Movie Recommender

Code for reading from the ratings.csv and movies.scv files and creating recommendations is described in the book *Programming Collective Intelligence* and will be given to you with some minor modifications. The script assumes the CSV files are in a directory relative to the script called movielens.

You are to modify recommendations.py to answer the questions below. Each question your script answers correctly will award you **10 points** for a maximum of 50 points. You must have the question answered completely correct; partial credit will only be awarded if your answer is *very* close to the

correct one. In all the questions that involve comparing movie ratings, use the Pearson correlation coefficient to compute similarity.

Before answering the final two questions, modify your ratings.csv file to include at least 5 movie ratings of your own (the timestamp is not important).

1. What 5 movies received the *most* ratings? What 5 movies received the *fewest* ratings? Show the movies and the number of ratings sorted by number of ratings.

2. What 5 movies received ratings *most* like *The Princess Bride*? Which 5 movies received ratings that were *least* like *The Princess Bride* (negative correlation)?

3. Which 5 raters most disagreed with each other (negative correlation)? Show the raters' IDs and Pearson's *r*, sorted by *r*.

4. Which 5 raters are most correlated to you, according to the recommendations you added? Show the raters' IDs and Pearson's *r*, sorted by *r*.

5. Compute the ratings for all movies that you have not seen (assume these are movies not listed in the recommendations you added). Show the top 5 movies you should see and the bottom 5 movies you are likely to hate.

Your output should clearly indicate the answers from the question you answered. Use the formatting of the example output shown below. Note that the answers shown below are not correct, they are just shown to indicate the desired format for your answers.

```
Question 1:
   Most rankings:
   1. Die Hard (1988) – 223 rankings
   2. Return of the Jedi (1983) – 209 rankings
   …

Question 2:
   Most like Princess Bride:
   1. Top Gun (1986) - 0.6477
   2. Star Wars (1977) - 0.4232
   …

Question 5:
   Must see movies:
   1. Godfather, The (1972) – 5.000
   2. Dead Poets Society (1989) – 4.9114
   …
```

McChallenge Extra Credit

The MovieLens data contains a links.csv file containing each movie's IMDb ID. You can download IMDB's dataset from their website (google it) or use the OMDb API (omdbapi.com) to obtain the IMDB rating of each movie. Write a Python script that creates a scatterplot where each movie is a dot. The x-axis is the MovieLens ranking, and the y-axis is the current IMDB ranking. Determine the Person's r for the two lists.

You will receive an additional 1% added to your final grade for emailing me this script no later than Friday of Dead Week.

Scatterplot tutorial using Matplot: https://pythonspot.com/matplotlib-scatterplot/

Submit

Submit your working Python script to Canvas before it is due. If you pair-program with another person, only one needs to submit the script.

Please place the recommendations you added to ratings.csv in comments at the top of your Python script. I will run your script on modified data from MovieLense and compare your solution to mine. Therefore, do not hard-code anything in your script that would cause your script not to work correctly with different data files.