# HW#6 - Recommendation Systems

Ethan Landers
Due: Sunday, November 17, 2024, by 11:59 PM

## Q1

Find 3 users who are closest to you in terms of age, gender, and occupation.

For each of those 3 users:

*Q: What are their top 3 (favorite) films?*

*Q: What are their bottom 3 (least favorite) films?*

I created a file, q1.py, to answer these questions and utilized the function loadMovieLens() from the example code in recommendations.py.

I used loadMovieLens() to access each user's associated movie titles and ratings in the MovieLens dataset. To adapt it for my program, I made minor modifications to the function: adjusting the path parameter to match my file structure and adding ISO-8859-1 encoding when reading u.item to handle special characters.

In the program, I supplied my own age, gender, and occupation to several functions. The program's primary function is to find the three most similar users to me based on age, gender, and occupation. It then displays the top three (favorite) and bottom three (least favorite) movies for each of these similar users. The program identified three other users who were also 22 years old, male, and students and generated their top and bottom three movies.

Below is the output of my program:

```
User 245's Recommendations:
    Top 3 Movies:
        Grumpier Old Men (1995) - Rating: 5
        Homeward Bound II: Lost in San Francisco (1996) - Rating: 5
        Return of the Jedi (1983) - Rating: 4
    Bottom 3 Movies:
        Gone with the Wind (1939) - Rating: 2
        Beavis and Butt-head Do America (1996) - Rating: 1
        Home Alone 3 (1997) - Rating: 1


User 327's Recommendations:
    Top 3 Movies:
        Raging Bull (1980) - Rating: 5
        Full Metal Jacket (1987) - Rating: 5
        Jaws (1975) - Rating: 5
    Bottom 3 Movies:
        Bean (1997) - Rating: 1
        Devil's Own, The (1997) - Rating: 1
        Event Horizon (1997) - Rating: 1


User 359's Recommendations:
    Top 3 Movies:
        Titanic (1997) - Rating: 5
        Close Shave, A (1995) - Rating: 5
        Twelve Monkeys (1995) - Rating: 5
    Bottom 3 Movies:
        Saint, The (1997) - Rating: 3
        Twister (1996) - Rating: 3
        Dante's Peak (1997) - Rating: 3
```

After reviewing the results, I have determined that User 359 is most similar to me. Therefore, I will use User 359 as the substitute ”me” for the remainder of the assignment.

## Q2

Based on the ratings that users have given to the movies, answer the following questions:

*Q: Which 5 users are most correlated to the substitute you (i.e., which 5 users rate movies most similarly to the substitute you?)*

*Q: Which 5 users are least correlated (i.e., negative correlation)?*

*Q: Explain the general operation of any functions you use from recommendations.py.*

To answer these questions, I created a new script called q2.py. In this script, I used the loadMovieLens() function from recommendations.py to load and access movie ratings and titles from the MovieLens dataset. Additionally, I adopted the topMatches() function to create a new function called find_least_most_correlated_users(). This modified function calculates the similarity between my substitute user and all other users based on their movie ratings to determine the most and least correlated users. The output below shows the five users with the highest positive correlation and the five users with the highest negative correlation to User 359:

```
Top 5 Most Correlated Users:
    User ID: 809, Correlation: 1.00
    User ID: 755, Correlation: 1.00
    User ID: 155, Correlation: 1.00
    User ID: 97, Correlation: 1.00
    User ID: 929, Correlation: 1.00

Top 5 Least Correlated Users:
    User ID: 202, Correlation: -1.00
    User ID: 180, Correlation: -1.00
    User ID: 473, Correlation: -1.00
    User ID: 900, Correlation: -1.00
    User ID: 760, Correlation: -1.00
```

The top five most correlated users have scores of 1.00, indicating that these users rated movies exactly the same way as my substitute user. As a result, their movie recommendations will likely align closely with my substitute user's preferences. On the other hand, the top five least correlated users have scores of -1.00, meaning that they rated movies in exactly the *opposite* way as my substitute user. These users may not recommend movies that align with my substitute user's preferences.

## Q3

Compute ratings for all the films that the substitute you has not seen and answer the following questions:

*Q: What are the top 5 recommendations for films that the substitute you should see.?*

*Q: What are the bottom 5 recommendations (i.e., films the substitute you is almost certain to hate)?*

*Q: Explain the general operation of any functions you use from recommendations.py.*

I created a script file called q3.py to answer these questions. In this script, I used the functions loadMovieLens() and getRecommendations() from recommendations.py to find my substitute user's top five and bottom five recommendations. The getRecommendations() function returns a list of movies with predicted scores, allowing me to identify the highest and lowest rated recommendations for the user. Below is the output generated by my script:

```
Top 5 Recommendations for Substitute User:
    Movie: Saint of Fort Washington, The (1993), Predicted Score: 5.00
    Movie: They Made Me a Criminal (1939), Predicted Score: 5.00
    Movie: Star Kid (1997), Predicted Score: 5.00
    Movie: Someone Else's America (1995), Predicted Score: 5.00
    Movie: Santa with Muscles (1996), Predicted Score: 5.00
Bottom 5 Recommendations for Substitute User:
    Movie: Amityville: A New Generation (1993), Predicted Score: 1.00
    Movie: Amityville Curse, The (1990), Predicted Score: 1.00
    Movie: Amityville 3-D (1983), Predicted Score: 1.00
    Movie: Amityville 1992: It's About Time (1992), Predicted Score: 1.00
    Movie: 3 Ninjas: High Noon At Mega Mountain (1998), Predicted Score: 1.
```

The top five recommendations have scores of 5.00, indicating a high likelihood that the substitute user will enjoy these films. The bottom five recommendations have scores of 1.00, suggesting a very low likelihood that the substitute user would enjoy these films.

## Q4

Choose your (the real you, not the substitute you) favorite and least favorite film from the data.

*Q: What are the top 5 most correlated films to your favorite film? Bottom 5 least correlated?*

*Q: What are the top 5 most correlated films to your least favorite film? Bottom 5 least correlated?*

*Q: Based on your knowledge of the resulting films, do you agree with the results? In other words, do you personally like/dislike the resulting films?*

*Q: Explain the general operation of any functions you use from recommendations.py.*

To answer these questions, I created a script named q4.py. In the script, I selected my favorite movie, *Aliens* (1986), and my least favorite movie, *Sleepless in Seattle* (1993), to find correlated films. To achieve this, I utilized the calculateSimilarItems() function from recommendations.py. This function generates a dictionary of item-item similarities, which served as the foundation for identifying related movies. I accessed the similarities for each chosen movie, retrieved the most and and least correlated films, and displayed the results to output as shown below:

```
Top 5 Most Correlated Movies to Favorite Movie:
    Movie: Yankee Zulu (1994), Correlation: 1.00
    Movie: Woman in Question, The (1950), Correlation: 1.00
    Movie: Witness (1985), Correlation: 1.00
    Movie: Wings of Courage (1995), Correlation: 1.00
    Movie: Wend Kuuni (God's Gift) (1982), Correlation: 1.00
Top 5 Least Correlated Movies to Favorite Movie:
    Movie: Vie est belle, La (Life is Rosey) (1987), Correlation: 1.00
    Movie: Vermont Is For Lovers (1992), Correlation: 1.00
    Movie: Two Deaths (1995), Correlation: 1.00
    Movie: Touki Bouki (Journey of the Hyena) (1973), Correlation: 1.00
    Movie: To Have, or Not (1995), Correlation: 1.00
Top 5 Most Correlated Movies to Least Favorite Movie:
    Movie: Wonderland (1997), Correlation: 1.00
    Movie: Witness (1985), Correlation: 1.00
    Movie: Wings of Courage (1995), Correlation: 1.00
    Movie: Wife, The (1995), Correlation: 1.00
    Movie: Unhook the Stars (1996), Correlation: 1.00
Top 5 Least Correlated Movies to Least Favorite Movie:
    Movie: Three Lives and Only One Death (1996), Correlation: 1.00
    Movie: Temptress Moon (Feng Yue) (1996), Correlation: 1.00
    Movie: Small Faces (1995), Correlation: 1.00
    Movie: Sleepover (1995), Correlation: 1.00
    Movie: Scarlet Letter, The (1926), Correlation: 1.00
```

I do not believe the correlated movies are similar to my favorite and least favorite films. After watching trailers for many of the movies listed in the results, I found that they often did not align with the categories in which they were ranked, whether tone or genre. Also, I noticed that some movies appeared in multiple categories, like *Wings of Courage* (1995). Many of the movies also seemed relatively obscure or not well-known.

## Q5

Filter the MovieLens dataset so that it only contains movies with at least 10 ratings.

*Q: How many movies are in the filtered dataset?*

Re-do Q3 and Q4 using this dataset.

*Q: Do you think the recommendations have improved?*

To address these questions, I created a script called q5.py. This script includes a function called filter_movies_by_ratings() that filters the dataset to only include movies with at least 10 ratings. The function itereates through the dataset, counts the number of ratings for each movie, and excludes movies with fewer than 10 ratings.

The methodology used for recommendations and item similarity analysis is similar to what was employed in Q3 and Q4. However, the filtered dataset alters the inputs, potentially affecting the results. Below are the results from the q5.py script after applying the modified dataset:

```
Number of movies in filtered dataset: 1144
```

```
(Q3) Top 5 Recommendations for Substitute User:
    Movie: Wrong Trousers, The (1993), Predicted Score: 4.56
    Movie: Casablanca (1942), Predicted Score: 4.52
    Movie: Shawshank Redemption, The (1994), Predicted Score: 4.49
    Movie: 12 Angry Men (1957), Predicted Score: 4.48
    Movie: Schindler's List (1993), Predicted Score: 4.48

(Q4) Top 5 Most Correlated Movies to Favorite Movie:
    Movie: Wonderland (1997), Correlation: 0.50
    Movie: To Live (Huozhe) (1994), Correlation: 0.50
    Movie: Telling Lies in America (1997), Correlation: 0.50
    Movie: Smile Like Yours, A (1997), Correlation: 0.50
    Movie: Shooting Fish (1997), Correlation: 0.50
```

Filtering reduced the dataset to 1,144 movies.

For the Q3 results, the recommendations appeared significantly more accurate, as the suggested movies are popular and well-renowned. This indicates that the filtering process likely enhanced the quality of user-based recommendations by focusing on widely rated films.

However, the Q4 results are less impressive. Many of the recommended movies are still obscure and not well-known. As a result, I would conclude that the filtering did not substantially improve the recommendations in this case.