Subject: ***Information Retrieval and Analysis***
Student: ***Stanciu Iulia-Cristina***
Group: *12*
Practicals: ***Lab8 – Recommender – 19.12.2022***

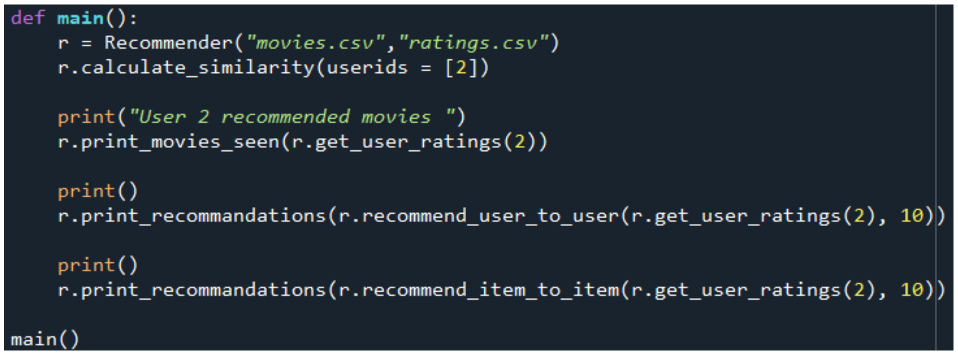
Collaborative filtering for Recommending movies based on ratings

Basic functionality of the recommender functions:

1. user-to-user recommendation
    - The parameters of the function are a list of pairs (movieid, rating) and an integer k indicating the desired number of recommendations.
    - Because the user may or may not be in the system, the user is added to the user list and their ratings to the list of pairs (movieid, rating).
    - The similarity matrix between users is calculated (explicitly a vector of similarities between the new user and all the other users. The similarity takes into account only the ratings of the movies seen by both users.
    - The recommended movies are given by the recommendation score, and the predicted rating is calculated as a weighted sum.
    - The recommended movies are the movies seen and rated (with good ratings) by most of the similar users.
    - In these calculations, only a limited number of users are considered (for this example I used the most similar 100 users).
    - The predicted movie score represents a weighted average of the ratings of the most similar users.
    - The function recommends the movies with the highest score (seen and liked by most of the similar users), not based on the highest movie. The reason for this is because otherwise, if only one of the similar users has seen a movie and rated it a 5, it would have a bigger chance to be recommended than one rated 4 or 5 by 99 out of the 100 users.

2. item-to-item recommendation
    - The parameters of the function are a list of pairs (movieid, rating) and an integer k indicating the desired number of recommendations.
    - Because the user may or may not be in the system, the user is added to the user list and their ratings to the list of pairs (movieid, rating).
    - The recommendations are based on how similarly the movies are rated compared to the movie best liked by the new user.
    - The similarity matrix between movies is calculated. (explicitly a vector of similarities between the movie and all the other movies.) The similarity takes into account all the ratings given by users. Also, if a user did not rate a movie, that movie automatically gets a 0 grade (Observation: I believe that this could work better with an average grade such as 2.5)
    - The recommendations are based on the predicted score. The score takes into account the similarity between the scores of the best rated movie of the user and the recommended movie, as well as the average rating of the users.

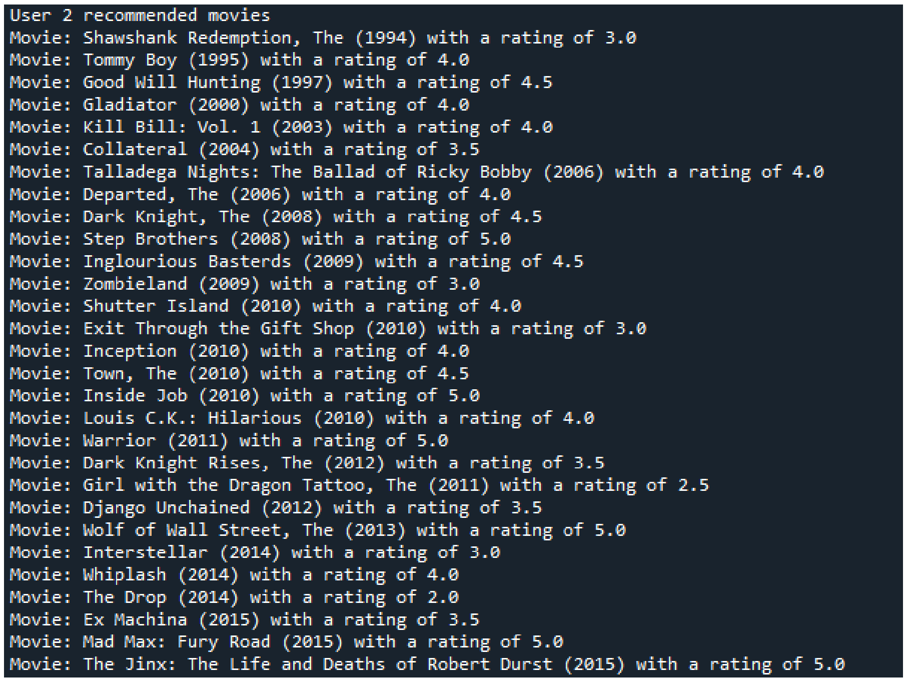
Subject: ***Information Retrieval and Analysis***
Student: ***Stanciu Iulia-Cristina***
Group: *12*
Practicals: ***Lab8 – Recommender – 19.12.2022***
The main function reads the following files: movies.csv and rating. csv and gives recommendations for a user or more.
Example of main function that shows the movies watched and 10 recommendations in each of the two ways: user-to-user and item-to-item:



```python
def main():
    r = Recommender("movies.csv","ratings.csv")
    r.calculate_similarity(userids = [2])

    print("User 2 recommended movies ")
    r.print_movies_seen(r.get_user_ratings(2))

    print()
    r.print_recommandations(r.recommend_user_to_user(r.get_user_ratings(2), 10))

    print()
    r.print_recommandations(r.recommend_item_to_item(r.get_user_ratings(2), 10))

main()
```

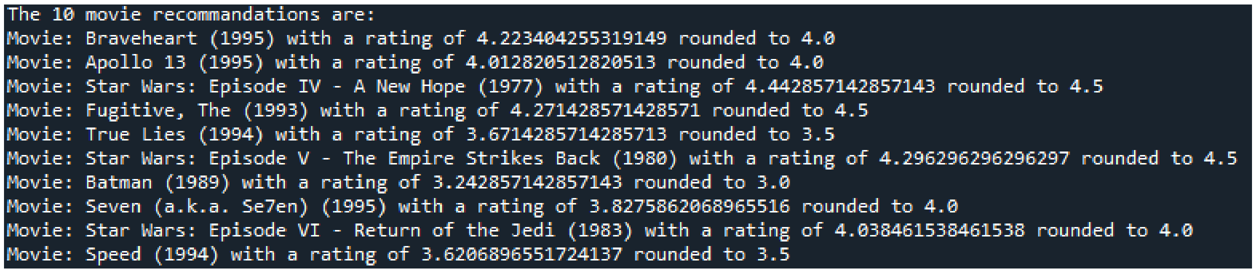
Results of the program for userid=2:

- Movies seen and the ratings:



```
User 2 recommended movies
Movie: Shawshank Redemption, The (1994) with a rating of 3.0
Movie: Tommy Boy (1995) with a rating of 4.0
Movie: Good Will Hunting (1997) with a rating of 4.5
Movie: Gladiator (2000) with a rating of 4.0
Movie: Kill Bill: Vol. 1 (2003) with a rating of 4.0
Movie: Collateral (2004) with a rating of 3.5
Movie: Talladega Nights: The Ballad of Ricky Bobby (2006) with a rating of 4.0
Movie: Departed, The (2006) with a rating of 4.0
Movie: Dark Knight, The (2008) with a rating of 4.5
Movie: Step Brothers (2008) with a rating of 5.0
Movie: Inglourious Basterds (2009) with a rating of 4.5
Movie: Zombieland (2009) with a rating of 3.0
Movie: Shutter Island (2010) with a rating of 4.0
Movie: Exit Through the Gift Shop (2010) with a rating of 3.0
Movie: Inception (2010) with a rating of 4.0
Movie: Town, The (2010) with a rating of 4.5
Movie: Inside Job (2010) with a rating of 5.0
Movie: Louis C.K.: Hilarious (2010) with a rating of 4.0
Movie: Warrior (2011) with a rating of 5.0
Movie: Dark Knight Rises, The (2012) with a rating of 3.5
Movie: Girl with the Dragon Tattoo, The (2011) with a rating of 2.5
Movie: Django Unchained (2012) with a rating of 3.5
Movie: Wolf of Wall Street, The (2013) with a rating of 5.0
Movie: Interstellar (2014) with a rating of 3.0
Movie: Whiplash (2014) with a rating of 4.0
Movie: The Drop (2014) with a rating of 2.0
Movie: Ex Machina (2015) with a rating of 3.5
Movie: Mad Max: Fury Road (2015) with a rating of 5.0
Movie: The Jinx: The Life and Deaths of Robert Durst (2015) with a rating of 5.0
```

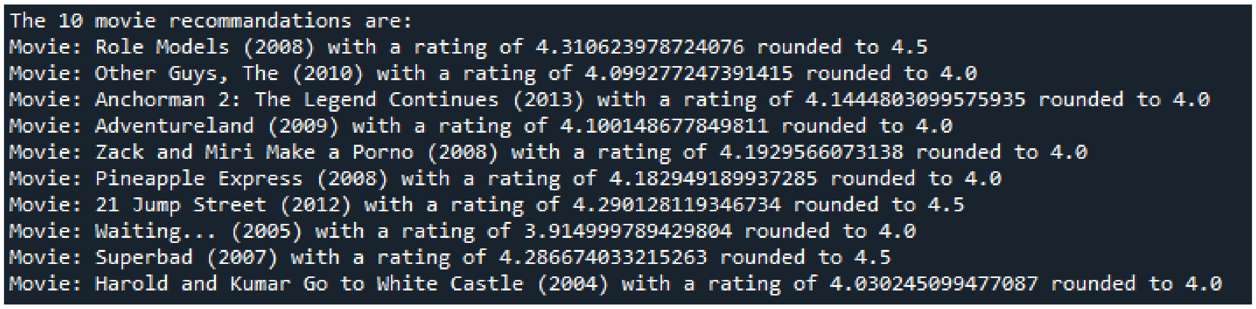
- User to user recommendations:



```
The 10 movie recommandations are:
Movie: Braveheart (1995) with a rating of 4.223404255319149 rounded to 4.0
Movie: Apollo 13 (1995) with a rating of 4.012820512820513 rounded to 4.0
Movie: Star Wars: Episode IV - A New Hope (1977) with a rating of 4.442857142857143 rounded to 4.5
Movie: Fugitive, The (1993) with a rating of 4.271428571428571 rounded to 4.5
Movie: True Lies (1994) with a rating of 3.6714285714285713 rounded to 3.5
Movie: Star Wars: Episode V - The Empire Strikes Back (1980) with a rating of 4.296296296296297 rounded to 4.5
Movie: Batman (1989) with a rating of 3.242857142857143 rounded to 3.0
Movie: Seven (a.k.a. Se7en) (1995) with a rating of 3.8275862068965516 rounded to 4.0
Movie: Star Wars: Episode VI - Return of the Jedi (1983) with a rating of 4.038461538461538 rounded to 4.0
Movie: Speed (1994) with a rating of 3.6206896551724137 rounded to 3.5
```

Subject: ***Information Retrieval and Analysis***
Student: ***Stanciu Iulia-Cristina***
Group: *12*
Practicals: ***Lab8 – Recommender – 19.12.2022***

- Item to item recommendations



```
The 10 movie recommandations are:
Movie: Role Models (2008) with a rating of 4.310623978724076 rounded to 4.5
Movie: Other Guys, The (2010) with a rating of 4.099277247391415 rounded to 4.0
Movie: Anchorman 2: The Legend Continues (2013) with a rating of 4.1444803099575935 rounded to 4.0
Movie: Adventureland (2009) with a rating of 4.100148677849811 rounded to 4.0
Movie: Zack and Miri Make a Porno (2008) with a rating of 4.1929566073138 rounded to 4.0
Movie: Pineapple Express (2008) with a rating of 4.182949189937285 rounded to 4.0
Movie: 21 Jump Street (2012) with a rating of 4.290128119346734 rounded to 4.5
Movie: Waiting... (2005) with a rating of 3.914999789429804 rounded to 4.0
Movie: Superbad (2007) with a rating of 4.286674033215263 rounded to 4.5
Movie: Harold and Kumar Go to White Castle (2004) with a rating of 4.030245099477087 rounded to 4.0
```

Conclusions:

- The recommendations from the user-based recommendation system differ a lot from the one based on movies.
- I tasted the program for more users. The recommended movies usually have predicted ratings of over 3 (on average number of recommendations), which is good because we don't want to recommend the user movies with a bad rating.
- It was hard to find a way to calculate the predicted rating, especially for the item-to-item function. I believe that it would have been better to calculate it also based on similar users, but I wanted to try a way in which to only consider the average rating and the similarity between movies.
- Optional work 3: The liked/disliked system would be easier to recommend based on a user similarity, than a item similarity. The problem of this system and the only likes system also is that most users are indifferent to a lot of movies. Those are usually good, but would not get recommended in this kind of system.

Questions, ideas and things taken into account:

- I believe it would have been better to consider both ratings and genres for the item-to-item function.
- I was not sure if the predicted rating should be on the same .5 scale as the given ratings from the .csv files.
- A combined recommendation system (of the two functions) would be more efficient.
- A testing system of the algorithm was started, but not finished. The idea behind it was to divide the data between training data and test data based on the timestamp and see if the movies a user is going to rate good in the future (from the division moment) are the same as the recommended ones.
- There are corner cases not discussed such as: a very popular movie rated as 1 by a user with the same/similar taste as the new user: I believe that should modify the predicted movie rating and recommendations more.