**AIE 425 Intelligent Recommender Systems, Fall Semester 24/25**

**Assignment #1: Neighborhood CF models (user,item-based CF)**

**Student ID:** A20000947

**Full Name:** Mohamed Yasser Saad Ahmed

This project aims to apply the neighborhood collaborating filtering models on the chosen dataset “MovieLens” from GroupLens. The dataset consists of 100873 row x 4 columns. 610 Unique users have given their ratings to 170875 movies.

The type of rating used in this dataset is a **Interval-based ratings**, a discrete bounded set of numbers from 1 to 5 where 5 is the highest rating and 1 represents the lowest.

Now after illustrating the data, I constructed a 4x6 user item matrix, where I made sure to have missing values in order to apply the model to fill them:

| movieId | 1 | 3 | 6 | 47 | 50 | 70 |
|---|---|---|---|---|---|---|
| userId | | | | | | |
| 1 | 4.0 | 4.0 | 4.0 | NaN | 5.0 | 3.0 |
| 2 | 4.0 | 4.0 | NaN | 5.0 | 5.0 | 3.0 |
| 3 | 4.0 | 4.0 | 4.0 | 5.0 | 5.0 | 3.0 |
| 4 | NaN | 4.0 | 4.0 | 2.0 | NaN | 3.0 |

This figure shows the created user item matrix which has 4 nan values that will be filled throughout the work.

**User-based CF:**

In the first step we find users that are similar to our target user in order to predict the rating our target user might give to a certain movie based on the ratings given by the users similar to our target. This level of similarity can be determined by the cosine similarity and pearson correlation coefficient. Similarities are computed between the **rows.** In this sector we calculated the similarities upon users and the results is shown under assignment results.

**Item-based CF**:

On the other hand, in the item-based CF we tend to apply our similarity calculations between the **columns** to find similarities between items to try to predict what rating the target user will give to a certain item based on the neighborhood items.

**Assignments results and calculations:**

First, we needed to calculate the average rating to find on average how every movie is rated using the following formula:

$$Average\ Rating = \frac{5 \times r_5 + 4 \times r_4 + 3 \times r_3 + 2 \times r_2 + 1 \times r_1}{r_5 + r_4 + r_3 + r_2 + r_1}$$

Where (r) represents the number of users who have rated this movie. After doing the calculations the following results has been aquired:

| Movie ID | Average Rating |
|---|---|
| 1 | 4.0 |
| 3 | 4.0 |
| 6 | 4.0 |
| 47 | 4.0 |
| 50 | 5.0 |
| 70 | 3.0 |

Then the code has been run to calculate both the cosine similarity (also the adjusted cosine similarity) and the Pearson correlation coefficient using both user-based and item-based CF the following results has been shown:

```
[10] User-Based Cosine Similarity:
          1         2         3         4
1  1.000000  0.996289  0.996289  0.972050
2  0.996289  1.000000  0.999632  0.956258
3  0.996289  0.999632  1.000000  0.956258
4  0.972050  0.956258  0.956258  1.000000

Item-Based Cosine Similarity:
           1         3         6        47        50        70
1   1.000000  0.996769  0.996769  0.976160  0.996833  0.996392
3   0.996769  1.000000  0.999391  0.958915  0.987429  0.999771
6   0.996769  0.999391  1.000000  0.958915  0.987429  0.999771
47  0.976160  0.958915  0.958915  1.000000  0.988486  0.956183
50  0.996833  0.987429  0.987429  0.988486  1.000000  0.986486
70  0.996392  0.999771  0.999771  0.956183  0.986486  1.000000

User-Based Pearson Similarity:
          1         2         3         4
1  1.000000  0.845154  0.845154  0.106600
2  0.845154  1.000000  0.985714 -0.306319
3  0.845154  0.985714  1.000000 -0.306319
4  0.106600 -0.306319 -0.306319  1.000000

Item-Based Pearson Similarity:
           1         3         6        47        50  70
1   1.000000  0.333333  0.333333  0.942809  1.000000 NaN
3   0.333333  1.000000 -0.333333  0.471405  0.333333 NaN
6   0.333333 -0.333333  1.000000  0.471405  0.333333 NaN
47  0.942809  0.471405  0.471405  1.000000  0.942809 NaN
50  1.000000  0.333333  0.333333  0.942809  1.000000 NaN
70       NaN       NaN       NaN       NaN       NaN NaN
```

Calculations has been made following this formula:

$$cosine(u,v) = \frac{\sum_{p \in com(u,v)=P}(r_{u,p})(r_{v,p})}{\sqrt{\sum_{p \in com(u,v)=P}(r_{u,p})^2}\sqrt{\sum_{p \in com(u,v)=P}(r_{v,p})^2}}$$

Now all the information required to make the predictions and recommendations is available. Which allows to proceed to make predictions and fill the nan values with the adjusted ratings using the information we have. The predictions were the following:

```
User-Based CF Predictions using Cosine Similarity:
          1         3         6        47        50   70
1  3.816115  4.050259  4.050259  4.012228  4.570934  3.0
2  3.818532  4.050586  4.050605  4.022042  4.576575  3.0
3  3.818532  4.050605  4.050586  4.022042  4.576575  3.0
4  3.806928  4.049234  4.049234  3.977479  4.549499  3.0

Item-Based CF Predictions using Cosine Similarity:
          1         3         6        47        50        70
1  4.000074  3.997923  3.997923  4.005533  4.002272  3.997724
2  4.197211  4.192931  4.192952  4.209652  4.201707  4.192406
3  4.197211  4.192952  4.192931  4.209652  4.201707  4.192406
4  3.254336  3.258575  3.258575  3.241321  3.249817  3.259165

User-Based CF Predictions using Pearson Similarity:
          1         3         6        47        50        70
1  3.971415  4.060435  4.060435  4.528122  4.933301  3.000000
2  3.292101  3.281711  3.282622  4.047113  4.194460  2.414153
3  3.292101  3.282622  3.281711  4.047113  4.194460  2.414153
4  0.713020  1.113626  1.113626 -0.370389  0.418682  0.861945
```

Now it is meant to be able to recommend top 3movies to the users, after reaching the results that allows us to make recommendations upon it and after running the code the recommendations were :

I

```
Top-N Recommendations (User-Based CF, Cosine Similarity):
 1    [50, 3, 6]
2    [50, 6, 3]
3    [50, 3, 6]
4    [50, 3, 6]
dtype: object

Top-N Recommendations (Item-Based CF, Cosine Similarity):
 1    [47, 50, 1]
2    [47, 50, 1]
3    [47, 50, 1]
4    [70, 3, 6]
dtype: object

Top-N Recommendations (User-Based CF, Pearson Similarity):
 1    [50, 47, 3]
2    [50, 47, 1]
3    [50, 47, 1]
4    [3, 6, 70]
dtype: object

Top-N Recommendations (Item-Based CF, Pearson Similarity):
 1    [1, 3, 6]
2    [1, 3, 6]
3    [1, 3, 6]
4    [1, 3, 6]
```

From this output we see in both cases of user-based either in cosine or Pearson similarity, the recommendations upon the user were similar and some of them has the

same recommendation which justifies the results obtained in the user-user cosine similarity we calculated earlier as the similarities upon them were high.

In the item-based in both cases we also see how similar they are. In the case of cosine similarity, an exceptionally low level of difference was observed. But when it came to the Pearson similarity, we found out that 4 users got the same recommendations.

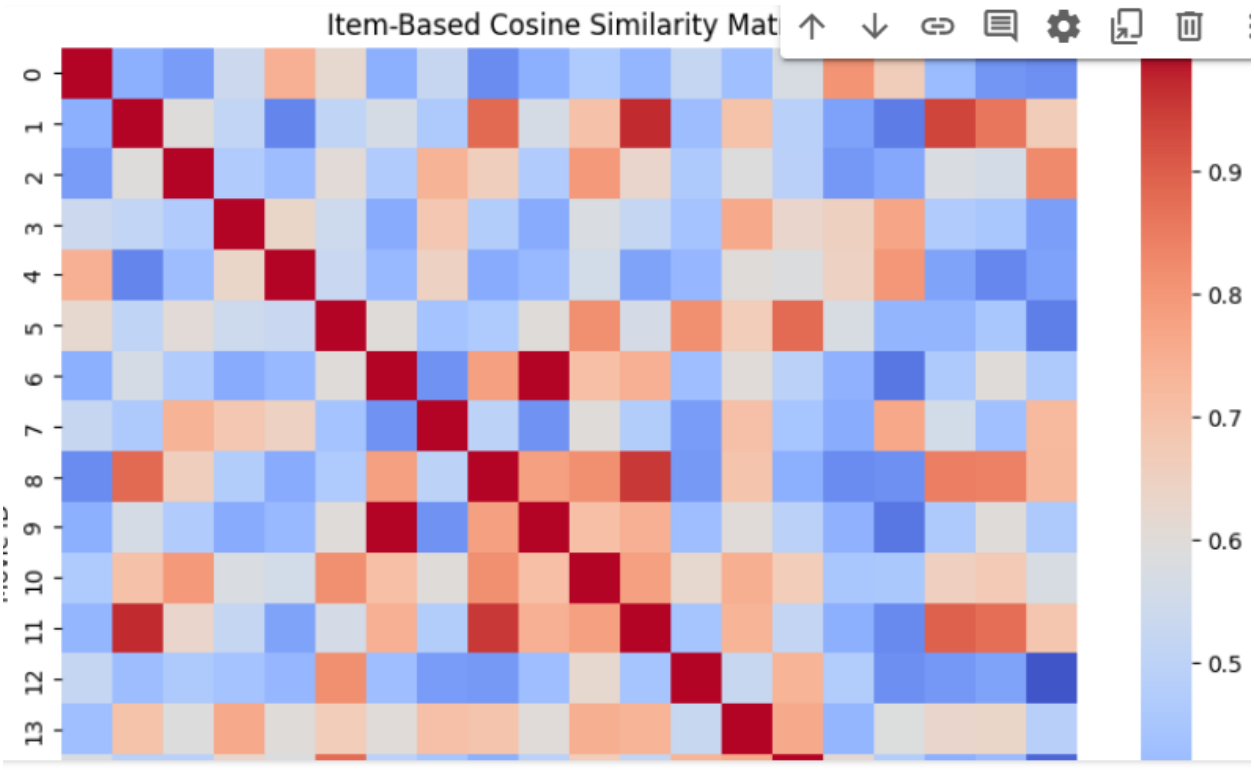
**Libraries and tools used:**

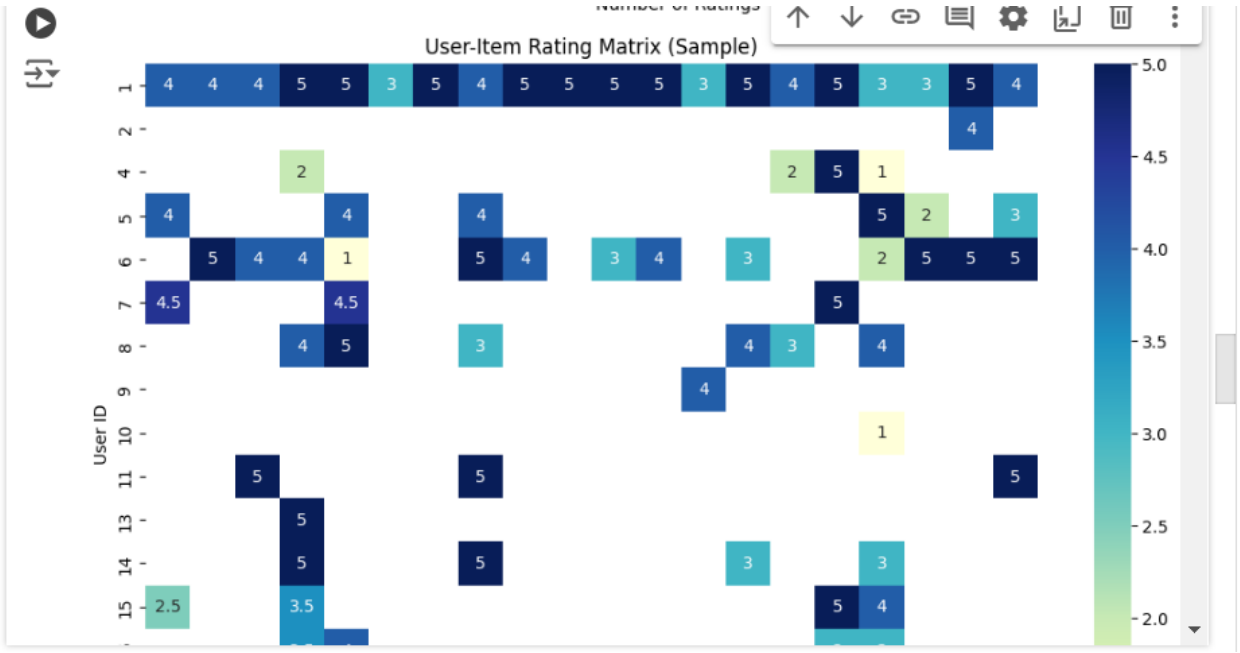
**Numpy:** to be able to do the calculations on the dataset.

**Pandas:** to load, read and manipulate the dataset.

**Sklearn.metrics.pairwise:** From which we import the **cosine_similarity** which is essential for our calculations.

**Matplot and seaborn:** to analyze and visualize the data and get some knowledge about the correlation between the datapoints in advance before applying the CF to it.


Item-Based Cosine Similarity Mat
0
1
2
3
4
5
6
7
8
9
10
11
12
13
0.9
0.8
0.7
0.6
0.5


User-Item Rating Matrix (Sample)
User ID
1
2
4
5
6
7
8
9
10
11
13
14
15
5.0
4.5
4.0
3.5
3.0
2.5
2.0

**Process:**

The dataset has been loaded without the need to preprocess it as it was already dataset. We constructed a subset of 4 rows and 6 columns to construct a user-movie matrix while considering keeping nan values to calculate and predict them. After filling in the using the mathematical approaches explained above, we finally were able to recommend the top 3 movies for each user.

The possible future enhancement is to revise the recommendation of the top 3 movies recommendation in the item-based CF approach using Pearson similarity as the results showed the same recommendation for every user which is something to keep an eye on as it makes a conflict with item-item similarity calculated using Pearson correlation coefficient as some items had a coefficient of 0.333 which mathematically means there is an inverse correlation and with the commercial language means that they are totally not similar.

**References**

[1] GroupLens, "MovieLens dataset (ratings.csv)", Available at: https://grouplens.org/datasets/movielens/.
[2] Piazza, "AIE425 course lectures", Fall 2024, Available at: https://piazza.com/gu.edu.eg/fall2024/aie425.