**AIE425 Intelligent Recommender Systems, Fall Semester 24/25**

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

**B20000017, Seif Eldin Adel Mohamed**

**Core Idea:**

In this assignment, we will exercise one of core topics in the Intelligent Recommender Systems. Neighborhood (user and item- based) collaborative filtering algorithm. We have dataset of userId, movieId, ratings of these movies according to user’s rates. The goal is to calculate cosine similarity and pearson correlation of user and item-based.

**Solution 2.3:**

1/2- Companies that use recommender systems are (Amazon, Spotify, Facebook, Netflix, Google).

3- Companies that use recommender systems gather feedback from users in several ways to improve the accuracy and personalization of recommendations. Rating systems like star ratings, numeric scales, review texts. Implicit feedback types are clicks and views, watch time, search queries. Social media likes and shares, follow and subscription patterns.

4/5- We do preprocessing procedures to clean data such as drop missing values and unnecessary columns.

1. User Matrix:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 | Item 6 |
| User 1 | 3.5 | 4 | 3.5 | 3.5 | 3.5 | 3.5 |
| User 2 | 4 | 5 | ? | 4 | 3 | 5 |
| User 3 | ? | 3 | 4 | 5 | ? | 2 |
| User 4 | 3 | 4 | 3 | 1 | 3 | 3 |
| User 5 | 3 | ? | 3 | 3 | 5 | 2 |

7- I have two datasets one contain movieId, title and genres and the other contain userId, movieId, rating and timestamp.

8- Copy results.

9- User-based CF: Is to determine users who are similar to the target user and recommend rating by computing weighted averages of ratings.

Item-based CF: Get rating predictions for target item by user, first determine set of items that are similar to target item.

11- Pros of cosine similarity: Handle sparse data well, automatic Normalization. While cons are, ignores rating magnitude, not sensitive to rating scale difference.

Pros of pearson correlation: Accounts for rating scale differences, More accurate with continuous data. While cons are: sensitive to outliers, less effective for sparse data.

12- Copy results.

15- Copy results.

17- The libraries used are pandas, numpy, sklearn.metrics.pairwise to import cosine\_similarity.

18- A cosine similarity is a simple and very efficient method for calculating the similarities of ratings among users. Possibilities are that the ratings will differ due to different rating scales used by the users. Normally, two users who have very similar tastes may have different rating scales. For instance, one might give higher scores on average than the actual one, which makes the cosine similarity irrelevant. The centroid of each user's average rating, as the calculation starts with individual scales, is the basis for computing the Pearson Correlation transporter. As a result, user similarity scores can be more reliable in situations of similar user preferences, while dissimilar ratings indicate otherwise. Nevertheless, Pearson is sensitive to outliers and may call for more overlapping from users for valid outcomes.

In item-based filtering, cosine similarity is a well-established relatively best choice provided that the ratings of items show more constancy over time. It does not need the mean to be removed from data and hence it is a fast method. A query can be made in situations where the rating is for the same items most of the time. However, it might not be a subtle way to tell the difference as users may have different styles in rating items. The Pearson correlation coefficient in item-based CF is a great technique to find more nuanced relationships between objects, especially when user ratings fluctuate a lot. It covers the scale of the rating and uses Pearson which is more proper for cases of users rating items with a broad range of diversities. But it may be less effective when the dataset is very sparse and only a few users are rating the same items.

19- User-based CF done with Pearson correlation, which uses different measurement levels, improved prediction accuracy to some extent when similar scales were used. However, overlapping data has to be available. Item-based CF with cosine similarity overcomes the issues of being unstable and inefficient in sparse datasets. In general, the fountain of success of Pearson correlation is dense, consistent data, however, the cosine similarity is the one armed with stability and reliability in a sparse context.

20- User and item-based CF aggregation or using the cosine similarity adjustment could be the solutions to the poor accuracy. Matrix factorization or deep learning methods can be ways to provide personal touches by means of the sparity of datasets.

**Assignment Results:**

Average rating = (5\*2)+(4\*3)+(3\*1)+(2\*3)+(1\*3.5)/2+3+1+3+3.5 = 2.76











