AIE 425 Intelligent Recommender systems , fall semester 24/25

Assignment #1: Neighborhood CF Models (User\_based, Item\_Based)

Student name : Maryam Elgohary

Student ID : A20000882

**Outlines**

1. **Introduction**
2. **Core Idea**
3. **Domains**
4. **Dataset Description**
5. **Design and Implementation of CF Models**
6. **Experiments and Results**
7. **Conclusion and Opinion**
8. **References**

**Introduction:**

In a world with high technology , Recommender systems plays a crucial role in matching users preferences .one of the main recommender system model is collaborative filtering. This approach mainly suggests Items like music, games, products and books. By using User- rating and item-rating to predict how likely an individual would prefer this item. In this assignment we will focus on Building CF (Item-Based and User-Based) and comparing the results with explaining the pros and cons of each.

**Core Idea**:

Collaborative filtering core idea is based on that users with similar taste most likely have same future interests. The collaborative filtering approach has two types :

User-based: recommends based on similar users preference. For example if user A and user B have same interest, it will most likely recommend same things for both

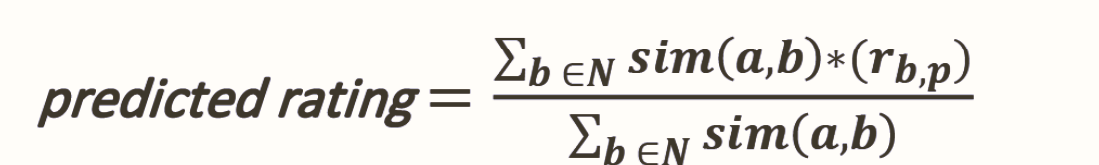
Item- based: recommends based on individual past interests. for example if user has liked or rates this item before, it will most likely recommend this item again

Analytical Solution:

The item based and user based similarity is calculated by 2 metrics approaches : Cosine Similarity and Pearson Correlation. The Cosine Similarity evaluates the users and items as vectors and calculates the difference angle. While in Pearson correlation it adds bias (the mean rating which helps more against the varying data causing it to be more accurate. The User-Based works on rows Meanwhile Item Based works on Columns.

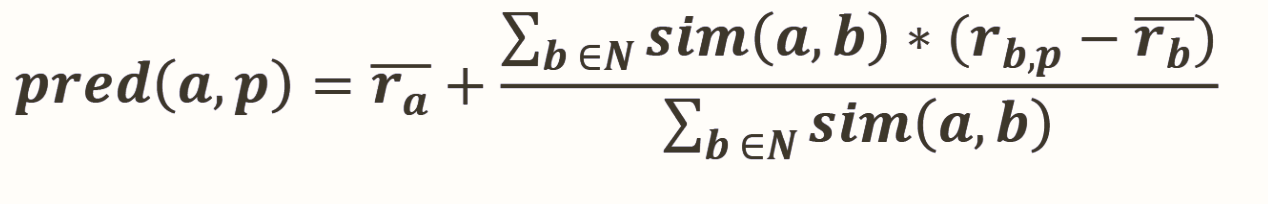
The user based differs in prediction rule:

In User-based Cosine Similarity Prediction Rule:

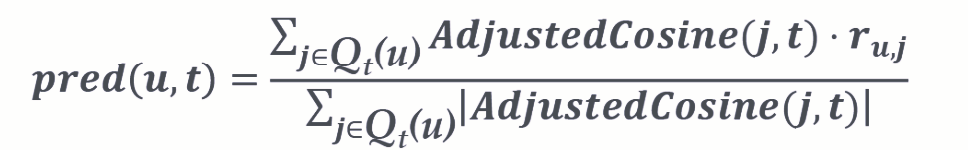


In User based Pearson Correlation Prediction Rule:

Note: the mean is the added weight



In Item-based Cosine similarity Prediction Rule:



**Domains**

* E-commerce: the recommender system was first made for e-commerce platforms as nt only enhancing the user experience, but also increasing the sales by increasing the visibility for the user. For example : Amazon, AliExpress.
* E-learning : Suggesting courses based on user past taken courses. For example: khan Academy , Coursera.
* Financial Services: recommender system is widely used in Stock Market , lately it recommends what’s best for you to invest in or what’s low risk for . For Example: thndr
* Health care: recommender system is used in mental health applications that it matches user with therapy exercises and peer communication
* Books and music: recommender systems is used to suggest books and music based on what user had read or listened to. For example: Spotify(music), Anghami(music), goodreads(book).

**Data description:**

Dataset is web scraped from good reads website, this website collects data by taking the user review, rating and collecting the average rating of all ratings. The rating type is interval rating from 1 to 5. By defining 1 with Very bad, 2 Bad, 3 with Neutral,4 with Good and 5 with Excellent:

The Book-rating dataset on consists of ratings of 10 books by.13 Users

The Book-Crossing dataset consists of :

* **Users:** Names of the users, Each one user must have rated more than 5 of the books
* **Books:**

1. Title of the book.
2. Rating of the book from 1 to 5.

This dataset is ideal for building User-Item Matrix and vice versa.

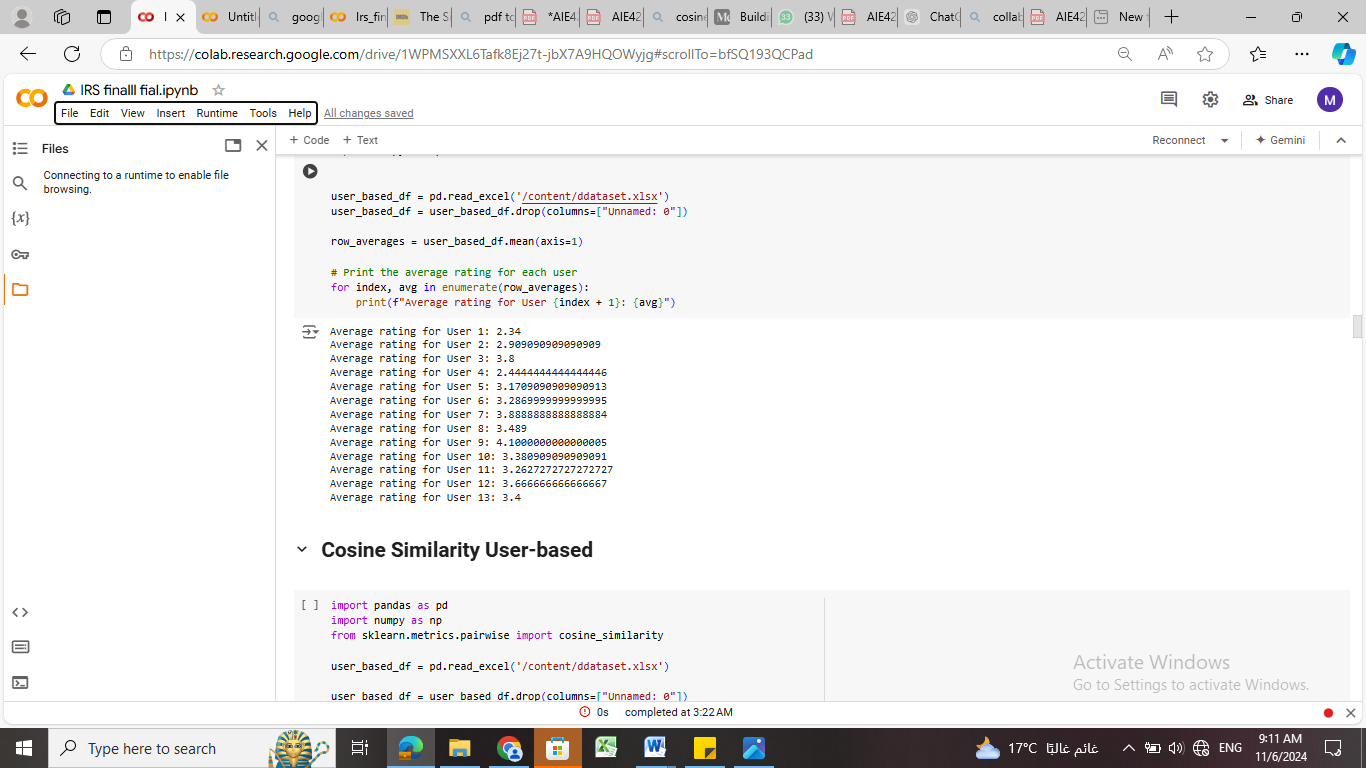
**Design and Implementation:**

* Data Loading and Preprocessing:
* Web scrap 10 Books, Ratings, and Usernames dataset using beautifulsoup library
* Dataset is saved in excel sheet
* Dataset has 10% missing data which can be used to be calculated by .recommender systems
* Tranposing data for item-Based purpose
* Data Visualization:
* Plot matrix correlation in Cosine similarity and Pearson correlation.
* Implementing cosine similarity user-based and item-based:
* Assuming Nan values equal zero and calculating cosine similarity built in function of an inserted user With all users and then making prediction based on weighted rate
* Implementing Pearson correlation user-based and item-based:
* Identifying the missing values, assuming Nan Value equal zero , calculating the average, creating a function to get the top 2 similar users, using weights to make predictions

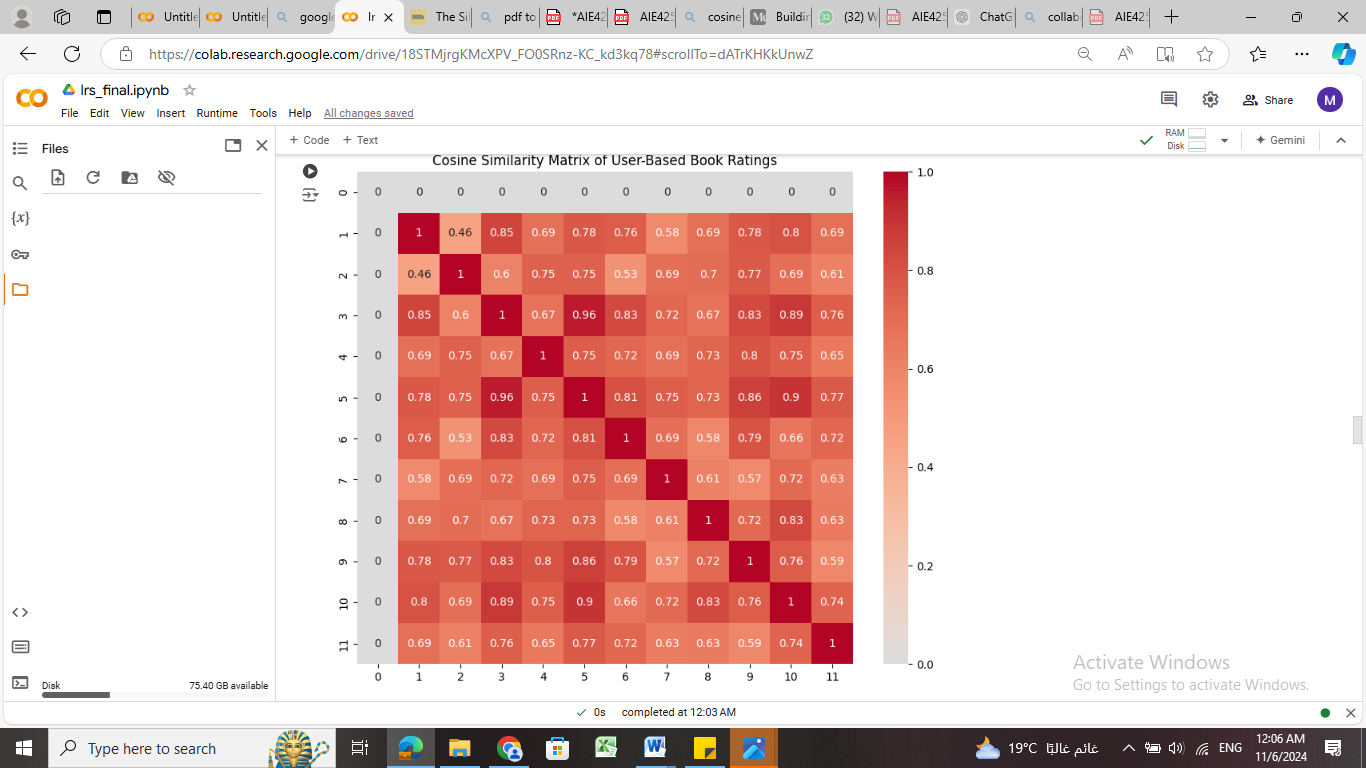
**Experiments and results:**

* + - 1. Data Overview and Visualizations:

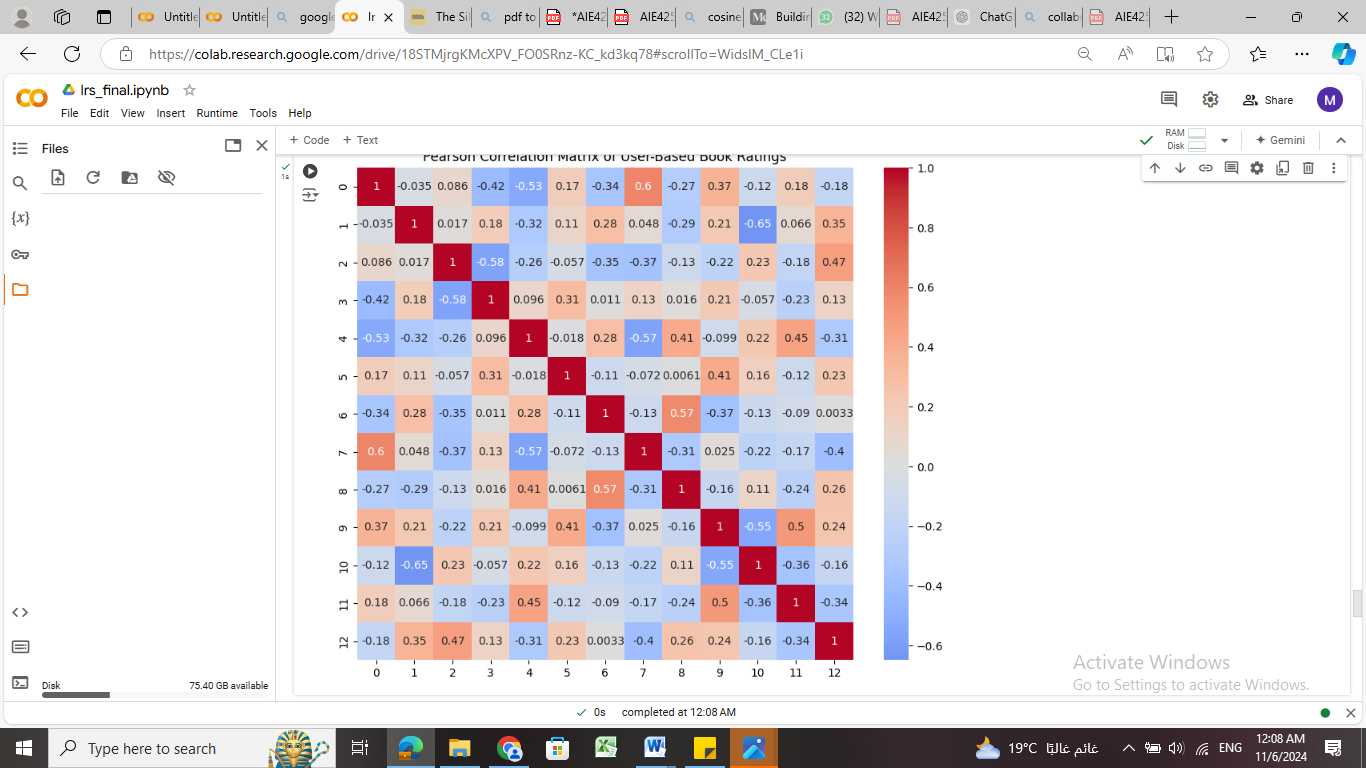
Average of each row

****

Cosine similarity graph



Pearson Similarity



Cosine Similarity

* Pros :

1. focuses on direction
2. doesn’t require mean centered
3. works well with many nulls

* Cons:

1. Doesn’t take into account user bias
2. Doesn’t take average rating into account for example two users could rate items similarly relative to each other, but cosine similarity does not account for how much they agree or disagree with the average rating

Pearson correlation:

* Pros:

1. Takes into account average rating
2. More robust to variance
3. Consider proportional behavior

* Cons:

1. Require mean center
2. Sensitive to outliers

c- Evaluation:

* Evaluate recommendations based on MAE:
* Manual solution:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | * B1 | * B2 | * B3 | * B4 | * B5 | * B6 |
| * U1 | * 1 | * 5 | * 1 | * 4 | * 3 | * 3 |
| * U2 | * ? | * 5 | * 3 | * 5 | * ? | * 1 |
| * U3 | * 5 | * 5 | * 3 | * 3 | * 3 | * 1 |
| * U4 | * 5 | * 2 | * 1 | * 4 | * 1 | * 3 |
| * U5 | * 5 | * 4 | * 4 | * 1 | * 3 | * 1 |

* Cosine similarity Rule:

1. Cosine (1, 2) = 0.9219
2. Cosine (2, 2) = 1.0000
3. Cosine (3, 2) = 0.9731
4. Cosine (4, 2) = 0.8485
5. Cosine (5, 2) = 0.8413

* Cosine similarity Prediction Rule:

1. Top 2 highest similarity U1 and U3
2. Pred(2,1)= 3
3. Pred(2,5) = 3

-both should be recommended equally

* Mean values

1. U1 = 2.31
2. U2 = 2.9
3. U3 =3.8
4. U4 =2.4
5. U5 = 3.1

* Pearson correlation Rule

1. Pearson (1,2)= 0.5606
2. Pearson (2,2)= 1
3. Pearson (3,2)= 0.8528
4. Pearson (4,2)= 0.1348
5. Pearson (5,2)= 0.3015

* Pearson Prediction

1. Top similar users = 1 ,3
2. Pred (2,1) =3.4
3. Pred(2,5) =3

-so the priority here for item 1

* Conclusion : we can conclude here that cosine similarity is not good with varying data and Pearson predict more precisely due to the bias
* Cosine similarity Item based:
* Mean centered matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * -1.3 | * 2.6 | * -1.3 | * 1.7 | * 0.7 |
| * ? | * 2.1 | * 0.1 | * 2.1 | * -1.9 |
| * 1.2 | * 1.2 | * -0.8 | * -0.8 | * -2.5 |
| * 2.6 | * -0.4 | * -1.4 | * 1.6 | * 0.6 |
| * 1.9 | * 0.9 | * 0.9 | * -2.1 | * -2.1 |

* Adjusted cosine similarity

1. Adjcosine(1,2)=-0.1141
2. Adjcosine(1,3)=-0.1830
3. Adjcosine(1,4)=--0.2519
4. Adjcosine(1,5)=-0.50861

* Conclusion : the correlation between item-item is very weak that will lead to insufficient prediction

**Conclusion and opinion:**

* In conclusion implementing the Collaborative Filtering approach to estimate the null values based on similar interests leveraging Pearson correlation and Cosine similarity to rate prediction has shown a reasonable results. This can provide a valuable insights and experience for the user. Its important to consider the limitation of the this model such as depending on taking feedback from user who can give a random value or not give a feedback at all. Building a recommender system that take data from various resource will enhance our model , integrating machine learning model to CF model will help increasing the accuracy of the recommendation Ultimately, The Collaborative Filtering approach will remain one of the core systems in Recommending system .

**References:**

[1] G. Goodreads, "Goodreads: Meet your next favorite book," Goodreads, 2024. [Online]. Available: <https://www.goodreads.com/>.

[2B. K. Excel, "Building Movie Recommender Systems Using Cosine Similarity in Python," Medium, 2024. [Online]. Available: <https://medium.com/@bkexcel2014/building-movie-recommender-systems-using-cosine-similarity-in-python-eff2d4e60d24>

[3] K. Najmani, E. H. Benlahmar, N. Sael, and A. Zellou, "Collaborative Filtering Approach: A Review of Recent Research," in Advanced Intelligent Systems for Sustainable Development (AI2SD’2020), AI2SD 2020, First Online: 10 February 2022, pp. 151–163.

**Pandas**

[4] "pandas: Powerful Python data analysis toolkit," [Online]. Available:

https://pandas.pydata.org/pandas-docs/stable/. [Accessed: November 10 2024]

**Matplotlib**

[5] "Matplotlib: Visualization with Python," [Online]. Available: https://matplotlib.org/stable/contents.html. [[Accessed: November 10 2024]

**Seaborn**

[6] "Seaborn: Statistical data visualization," [Online]. Available: https://seaborn.pydata.org/. [Accessed: November 10 2024]

**BeautifulSoup**

[7] "Beautiful Soup: Python library for parsing HTML and XML documents," [Online]. Available: https://www.crummy.com/software/BeautifulSoup[Accessed: November 10 2024]

**Request**

[7] "Requests: HTTP for Humans," [Online]. Available: <https://requests.readthedocs.io/en/latest/>. [Accessed: November 10 2024]