**AIE425 Intelligent Recommender Systems**

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

**Student ID: A20000476, Student Name: Shahd Ahmed Maher Gawad**

1. **Introduction**

2

Built an intelligent recommender system for this assignment that makes game recommendations based on user reviews and ratings. Finding comparable games that fit each user's interests and analyzing trends in user preferences were the main goals. The algorithm is able to successfully recommend games that users are likely to like by analyzing user interactions and feedback on a variety of games. Collaborative filtering (CF) models—more especially, user-based and item-based strategies with both cosine and Pearson similarity measures—were used to complete this recommendation process.

1. **Recommender Systems Ideas**

When I went back to the lectures and searched, I found that there are many recommender systems like:

* Netflix Recommender System
* Amazon Recommender System
* Spotify Recommender System
* Google News Recommendations
* RAWG Game Recommender, and many more.

But for this assignment, I chose to work on the RAWG Game Recommender.

1. **Customer Feedback Collection and Rating System on RAWG**

Users can rate games on a scale (usually out of five stars) and write reviews about their experiences, which is how RAWG, a well-known game recommendation platform, gathers customer feedback. This information is crucial for understanding user satisfaction and game popularity because it provides direct insight into each user's preferences and opinions. The ratings are typically displayed in an aggregated average, which summarizes how the game is received by the larger user base.

1. **Prepare The Collected Data**

3

We use a lot of techniques to gather data, such as databases, web  scrapers, web crawlers, and APIs. I made the decision to develop an API. Getting the API key and registering for an account on the RAWG Video  Game Database API was the first step.After that, we created the code to  get information from the website.The coding then made it clear to me that I wanted the data organized with 7 games in columns and 50 users in rows

1. **Data Collection & Preparation**

We use a lot of techniques to collect data, such as databases, web scrapers, web crawlers, and APIs. After that, we created the code to get information from the website. The coding then made it clear to me that I wanted the data organized with seven games in columns and fifty users in rows.

**5.1. Data Collection**

Made the choice to develop an API. Getting the API key and registering for an account on the RAWG Video Game Database API was the first step.

* 1. **Data Structuring**

Aimed to create a dataset where 50 users in rows and 7 games in columns.

* 1. **Data Preprocessing**
     1. **Encoding**

Changed numerical values from category reviews.

Example:

* Poor = 1
* Average = 2
* Good = 3
* Very Good = 4
* Excellent = 5
  1. **Dataset Preparation**

4

50 users represented in the rows.

7 games represented in the columns.

1. **Comparison User-based and item-based CF algorithms**

**6.1. User-Based CF:** Using the preferences of other users who are similar to the user, user-based collaborative filtering suggests products to the user. It finds individuals who share similar preferences and makes recommendations for products they enjoyed.

**How It Operates:**

* Similarity Calculation: Determine how similar users are to one another using metrics such as the Jaccard index, cosine similarity, or Pearson correlation.
* Choose a neighborhood of people that are similar to you (for example, the top N similar users).
* Recommendation Generation: Create suggestions for the intended user by combining the ratings of users who are similar.
  1. **Item-Based CF:** Based on the similarities between items that the user has previously assessed, item-based collaborative filtering suggests items to the user. It finds and recommends products that are comparable to those the consumer enjoys.

**How It Operates:**

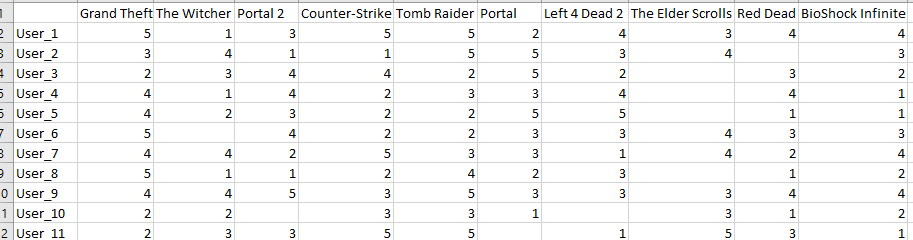
* Similarity Calculation: Use metrics such as adjusted cosine similarity or cosine similarity to determine how similar two things are.
* Recommendation Generation: Identify related products for each user-rated item and provide recommendations based on the ratings.

1. **Description about Dataset**

**7.1. Dataset Overview**

5

User reviews of a chosen 7 video games make up the dataset. It is organized in a matrix way, with the games shown in columns and users in rows. An integer number that represents a user's rating of a particular game is contained in each cell of the matrix. From 1 to 5, the ratings may be found where:

* Poor = 1
* Average = 2
* Good = 3
* Very Good = 4
* Excellent = 5
  1. **Sample From Dataset**
  2. **Data Collection**

We use a lot of techniques to collect data, such as databases, web scrapers, web crawlers, and APIs. After that, we created the code to get information from the website. The coding then made it clear to me that I wanted the data organized with seven games in columns and fifty users in rows.

* 1. **Data Preprocessing**

To make matrix operations and similarity computations easier, user and game IDs were numerically encoded.

**7.5. Rating Type**

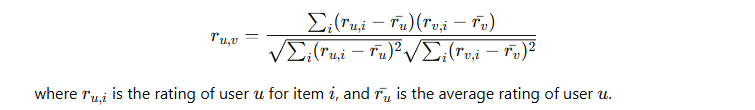
6

With integer values ranging from 1 to 5, the dataset employs a discrete rating system.

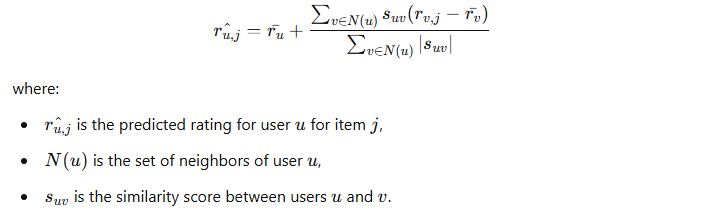
1. **User-Based & Item-Based CF (Analytical Solution)**

**8.1. User-Based CF**

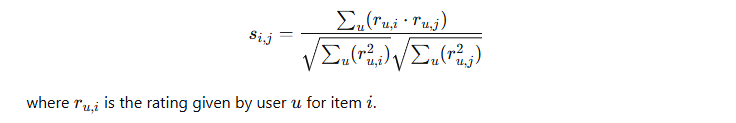
9.1.1. Similarity Calculation (Pearson Correlation)



8.1.2. Rating Prediction

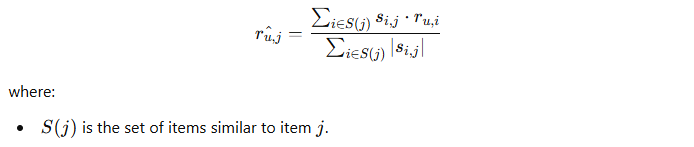


* 1. **Item-Based CF**
     1. Similarity Calculation (Cosine Similarity)



* + 1. Rating Prediction

7



1. **Cosine Similarity & Pearson Correlation Coefficient**

**9.1. Cosine Similarity**

**9.1.1 User-Based CF**

Cosine Similarity is Computed as:



A cosine similarity value close to 1 indicates that the users have similar rating patterns, while a value close to 0 indicates little to no similarity.

**9.1.2 Item-Based CF**

As with the user-based instance, each item is represented as a vector of ratings across all users, and you can identify which items are scored similarly by users by computing the cosine similarity between items based on how users have rated them.

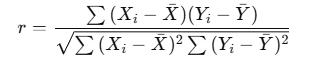
* 1. **Pearson Correlation Coefficient**

8

**9.2.1 User-Based CF**

This metric evaluates how well a linear function captures the relationship between two users. It takes into consideration the users' average ratings.

Pearson Correlation Coefficient is Computed as:

****

A Pearson correlation close to 1 indicates that the users tend to rate items similarly, while a value close to -1 indicates a strong negative correlation.

**9.2.2 Item-Based CF**

Items are connected depending on user ratings.

This draws attention to products that, in terms of user preferences, have a linear relationship.

**User-Based CF**

* Determine the Pearson correlation and cosine similarity between users.
* Determine a user peer group by looking at high similarity scores.

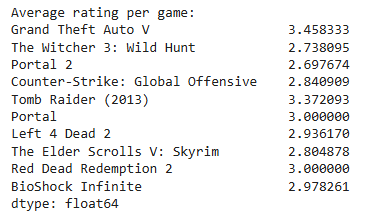
**Item-Based CF**

* Determine the Pearson correlation and cosine similarity between the items.
* Determine which items in the peer group have the highest similarity scores.

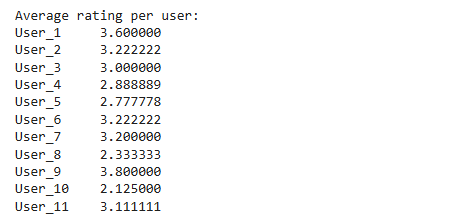
1. **Average Rating Overall Data**

9

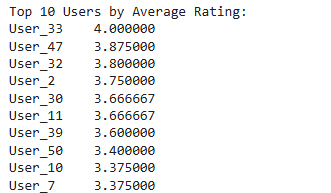
**10.1. Average Rating Per Game**

****

**10.2 Average Rating Per User (Sample)**



**10.3 Average Rating For Top 10 Users**

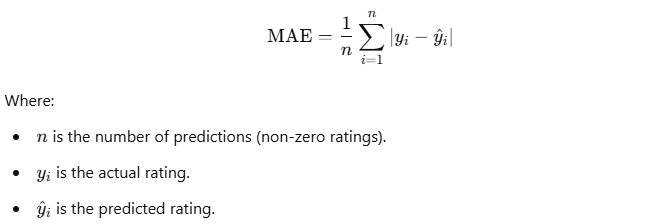


1. **Mean Absolute Error(MAE) & Root Mean Square Error(RMSE)**

10

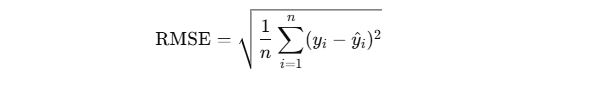
**11.1 Mean Absolute Error(MAE)**

MAE is preferred when you want a straightforward average error measurement that treats all errors equally.



**11.2 Root Mean Square Error(RMSE)**

RMSE is useful when you want to give more importance to larger errors, which is often the case in many applications.



1. **Results**

11

**12.1 User-Based CF By Pearson**

Mean Absolute Error (MAE) for User\_1: 0.44

Root Mean Square Error (RMSE) for User\_1: 0.61

Accuracy: 0.912

**12.2 Item-Based CF By Pearson**

Mean Absolute Error (MAE) for User\_1: 0.98

Root Mean Square Error (RMSE) for User\_1: 1.16

Accuracy: 0.804

**12.3 User-Based CF By Cosine**

Mean Absolute Error (MAE) for User\_1: 1.17

Root Mean Square Error (RMSE) for User\_1: 1.3

Accuracy: 0.766

**12.4 Item-Based CF By Cosine**

Mean Absolute Error (MAE) for User\_1: 1.18

Root Mean Square Error (RMSE) for User\_1: 1.41

Accuracy: 0.764

Due to its higher accuracy and lower error metrics, the User-Based CF that uses Pearson correlation is clearly the best model for making predictions.

1. **Conclusion**

The User-Based Collaborative Filtering (CF) method using Pearson correlation was shown to be the most successful collaborative filtering model in our research of models for the gaming recommender system. With a score of 0.912, the results showed excellent accuracy with a Mean Absolute Error (MAE) of 0.44 and a Root Mean Square Error (RMSE) of 0.61. This model was the best option for our dataset because it performed noticeably better than item-based strategies and cosine similarity techniques. The results confirm that Pearson's User-Based CF is the best model for our gaming recommender system, guaranteeing higher user happiness with correct game suggestions.

1. **Future Enhancements**

12

* Dynamic Recommendations: To keep the system up to date and applicable, apply real-time adjustments to the recommendation model in response to fresh user interactions or game releases.
* Performance Optimization: Make the system as fast and scalable as possible, particularly if the user base grows or the dataset grows substantially.
* Hybrid recommendation systems: Utilize both user preferences and item characteristics to improve the overall quality of recommendations by combining collaborative filtering with content-based filtering.

1. **References**

[**Explore RAWG Video Games Database API • RAWG**](https://rawg.io/apidocs)

[**https://ijresm.com/Vol.3\_2020/Vol3\_Iss2\_February20/IJRESM\_V3\_I2\_143.pdf**](https://ijresm.com/Vol.3_2020/Vol3_Iss2_February20/IJRESM_V3_I2_143.pdf)