

**AIE425 Intelligent recommender systems, Fall Semester 24/25**

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

**221101218, Ahmed Osama Mahmoud**

**Core idea of Assignment:**

The primary objective of this assignment is to explore and apply Collaborative Filtering (CF) techniques—specifically, User-based and Item-based CF—to create personalized recommendations. This involves utilizing similarity measures like Cosine Similarity and Pearson Correlation. Additionally, this assignment aims to develop skills in web scraping to gather data and to understand the key differences between Cosine Similarity and Pearson Correlation, helping to determine when each measure is most suitable this revised version emphasizes both the technical skills you'll build (like personalized recommendation systems and web scraping) and the foundational knowledge of similarity measures. Let me know if you'd like further customization

**The solution of all steps in Section 2.3**

1. There are many companies use recommended Systems like:
2. **E-commerce:**

* **Amazon:** recommendation engine suggests products bases on user purchase history
* **Alibaba:** use recommendation system to suggest products tailored to user
* **eBay:** recommends items based on user’s search history and similar items

1. **Video:**

* **Netflix:** recommendation engine suggests movies and TV show based on user past viewing and ratings
* **Spotify:** recommendation engine suggests songs and playlist based on user history and listening patterns
* **YouTube:** use recommendation algorithms to suggests videos based on user search history and interactions

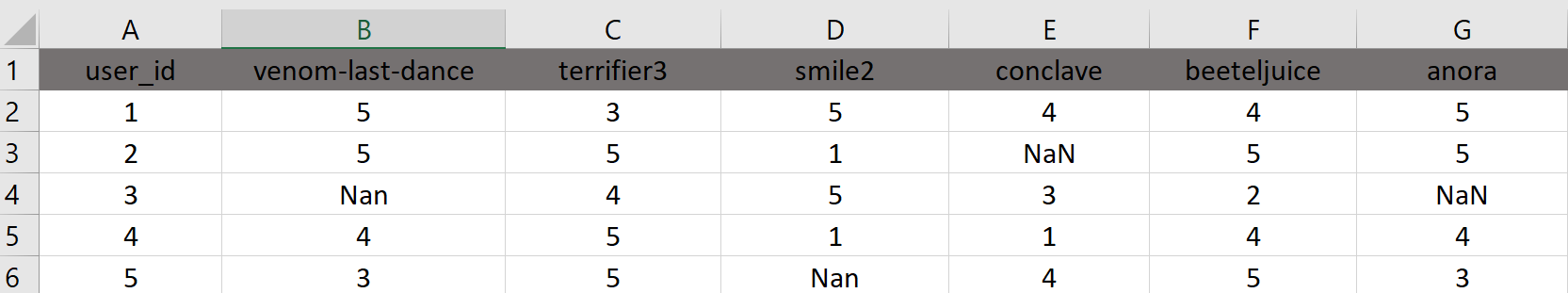
**3.Social Media:**

* **Facebook:** recommends friends, groups based on user interactions
* **LinkedIn:** Recommends jobs, connections, and articles based on user profile

1. I choose rotten tomatoes which is a movie website that recommend movies and TV shows based on user past viewing and ratings
2. Rotten tomatoes collect data from 2 resources which are critic reviews and users rating and rotten tomatoes rating consists of 3 types which are:

* **Tomato meter:** Percentage-based rating calculated from critic reviews
* **Audience Score:** Percentage-based rating calculated from audience ratings, reflecting how many users rated it 3.5 stars or above
* **Star Rating**: Audience members provide a star rating (0.5 to 5 stars), but this is not reflected in the Tomato meter

1. Collecting form rotten tomatoes website and cleaning data from any null values and converting rating values from stars to rounded numerical values
2. Firstly, I use web scrapping from rotten tomatoes which download the html file page and scrapping data that I needed to extract user matrix and extracting the rates of each movie after that collect each movie rating separated from other movie and finally, I merged it manually in user matrix csv

****

Columns:

* **user\_id**: Unique identifier for each user. This is a numeric column where each row corresponds to a different user
* **venom-last-dance, terrifier3, smile2, conclave, beetlejuice, anora**: These columns represent the names of movies that users have rated

Structure:

* Numerical values ratings: which ranged from 0.5 to 5 as this is rotten tomatoes rating scale type
* Missing rating: ‘NaN’ indicating that some users did not provide ratings for certain movies

1. I will take sample of this user matrix which are first 5 users and their ratings

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| user\_id | venom-last-dance | terrifier3 | smile2 | conclave | beeteljuice | anora | Avg-rating |
| 1 | 5 | 3 | 5 | 4 | 4 | 5 | 4.3 |
| 2 | 5 | 5 | 1 | NaN | 5 | 5 | 4.2 |
| 3 | Nan | 4 | 5 | 3 | 2 | NaN | 3.5 |
| 4 | 4 | 5 | 1 | 1 | 4 | 4 | 3.1 |
| 5 | 3 | 5 | NaN | 4 | 5 | 3 | 4 |

**User-based CF:**

* Similarity functions are computed between rows of ratings history to know the similar users
* Cosine similarity equation:

* Predicting ratings

**Item-based CF:**

* Similarity functions are computed between columns of rating history to know similar items
* Adjusted cosine similarity equation:
* Predicting rates



**User-based:**

**Cosine similarity:**

Cosine-similarity (1,3) = **=** 0.954

Cosine-similarity (2,3) ==0.730

Cosine-similarity (4,3) = =0.747

Cosine-similarity (5,3) == 0.960

**Person Correlation:**

Sim (1,3) = = 0.29

Sim (2,3) = -0.757

Sim (4,3) = -0.307

Sim (5,3) = 0.648

**Item-based:**

**Mean-centered-ratings**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| User-id | venom-last-dance | terrifier3 | smile2 | conclave | beeteljuice | anora |
| 1 | 0.7 | -1.3 | 0.7 | -0.3 | -0.3 | 0.7 |
| 2 | 0.8 | 0.8 | -3.2 | ? | 0.8 | 0.8 |
| 3 | ? | 0.5 | 1.5 | -0.5 | -1.5 | ? |
| 4 | 0.9 | 1.9 | -2.1 | -2.1 | 0.9 | 0.9 |
| 5 | -1 | 1 | ? | 0 | 1 | -1 |

Adj-cosine(venom-last-dance,terrifier3) = = 0.097

Adj-cosine (venom-last-dance, smile2) = -0.495

Adj-cosine (venom-last-dance, conclave) = -0.65

Adj-cosine (venom-last-dance, beeteljuice) = 0.087

Adj-cosine (venom-last-dance, anora) = 1

Adj-cosine (anora, venom-last-dance) = = 1

Adj-cosine (anora, beeteljuice) = 0.087

Adj-cosine (anora, conclave) = -0.688

Adj-cosine (anora, smile2) = -0.7

Adj-cosine (anora,terrifier3) = 0.09

1. **Cosine measurement** gives us a high similarity score close to 1 if the vectors point in the same direction regardless of their magnitude and the pros of cosine measurement Insensitive to the magnitude of vectors focuses on direction and also useful for sparse or high-dimensional data and from cons is doesn’t capture relationships based on value changes and can give high scores even if vectors have different mean levels

**Pearson Correlation**: This measure accounts for both the **linear relationship** and **direction** of change between the two vectors. A high positive correlation indicates that both vectors tend to increase and decrease in the same manner and the pros of Pearson correlation is considers both magnitude and direction and provides linear relationships and from cons is that assumes linearity there be misleading if data isn’t linear and sensitive to outliers

**13.**

person-pred (3,1) = 3.5 + = 3.02

cos-pred(3,1) = = 3.99

top N-list (user-based ) -> { user-1, user-2 }

top N-list (item-based) -> { anora, venom-last-dance}

**14.**

person-pred (3,1) = 3.5 + = 3.02

cos-pred (3,1) = = 3.99

which are closer to each other’s in there rating

**16.**

The results differed based on the similarity technique applied. For instance, the cosine similarity values varied compared to those obtained using Pearson similarity. This difference arises because Pearson similarity accounts for the inherent biases of neighboring data points or users, adjusting for individual tendencies. In contrast, cosine similarity focuses only on the angle between vectors, disregarding any mean-centered adjustments or personal biases. As a result, Pearson similarity can often provide a more nuanced measure of correlation by neutralizing these biases, while cosine similarity offers a straightforward measurement of vector alignment without such adjustments. This distinction highlights the importance of selecting a similarity measure that aligns with the goals of the analysis and the characteristics of the data

**17.**

BeautifulSoup is a Python library used for parsing HTML and XML documents. It provides methods for navigating and searching through the HTML tree, making it popular for web scraping tasks, JSON (JavaScript Object Notation) is a data-interchange format that’s lightweight and human-readable. The json library in Python provides functionality for working with JSON data, making it easy to read from and write to JSON files, Pandas is a powerful data manipulation and analysis library in Python, particularly well-suited for handling and processing large datasets , he csv library provides support for reading from and writing to CSV (Comma-Separated Values) files, which are a common format for storing tabular data

18. When comparing user-based and item-based collaborative filtering using cosine similarity and Pearson correlation, several key differences emerge due to the way each method interprets relationships and biases. Here are some remarks on the perceived differences between these approaches in bias sensitivity in cosine similarity less sensitive to user biases, as it only measures the angle between vectors without considering the absolute magnitude on the other hand, normalizes ratings by subtracting the mean rating for each user or item, thereby addressing the user or item bias.

**References:**

[1] BeautifulSoup Documentation. Beautiful Soup Documentation. [Online]. Available: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

[2] M. L. McKinney, Pandas Documentation. [Online]. Available: <https://pandas.pydata.org/pandas-docs/stable/>

[3] SciPy. SciPy Reference Guide. [Online]. Available: <https://docs.scipy.org/doc/scipy/>

[4] Python Software Foundation, "json — JSON encoder and decoder," Python Documentation. [Online]. Available: <https://docs.python.org/3/library/json.html>.