**AIE425 Intelligent Recommender Systems, Fall Semester 24/25**  
**Assignment #1: Neighborhood CF Models (User, Item-Based CF)**

**Student ID:** A20001136  
**Full Name:** Abdelrahman El-Sayyad

**Table of Contents**

1. Introduction
2. Core Idea of the Assignment
3. Dataset Description
4. Data Collection and Preprocessing
5. Methodology and Implementation
6. Assignment Results
7. Conclusion and Opinion
8. References

**1. Introduction**

**Purpose:** This assignment aims to explore the foundational concepts of Intelligent Recommender Systems, particularly focusing on Neighborhood Collaborative Filtering (CF) models, including user-based and item-based approaches. In this context, we utilize data scraping techniques to build and analyze a user-item matrix with recommendations.

**2. Core Idea of the Assignment**

The main objective of this assignment is to develop user-based and item-based collaborative filtering algorithms by using a dataset collected from a web source, transforming the data into a user-item matrix, and calculating recommendations based on similarity measures. This project emphasizes understanding collaborative filtering through hands-on data collection, preprocessing, matrix generation, and similarity computations.

**3. Dataset Description**

**Source:** The dataset used for this assignment is scraped from the "quotes.toscrape.com" website.  
**The dataset contains the following columns:**

* **Quote:** Text of the quote.
* **Author:** Name of the author.
* **Tags:** Keywords or "genres" associated with each quote, representing the themes.

**Data Structure:** The dataset is transformed into a user-item matrix where:

* Each row represents a simulated user.
* Each column represents a unique "tag" from the scraped quotes.
* Each cell contains a rating from 1 to 5, representing the user's hypothetical preference for each tag
* 

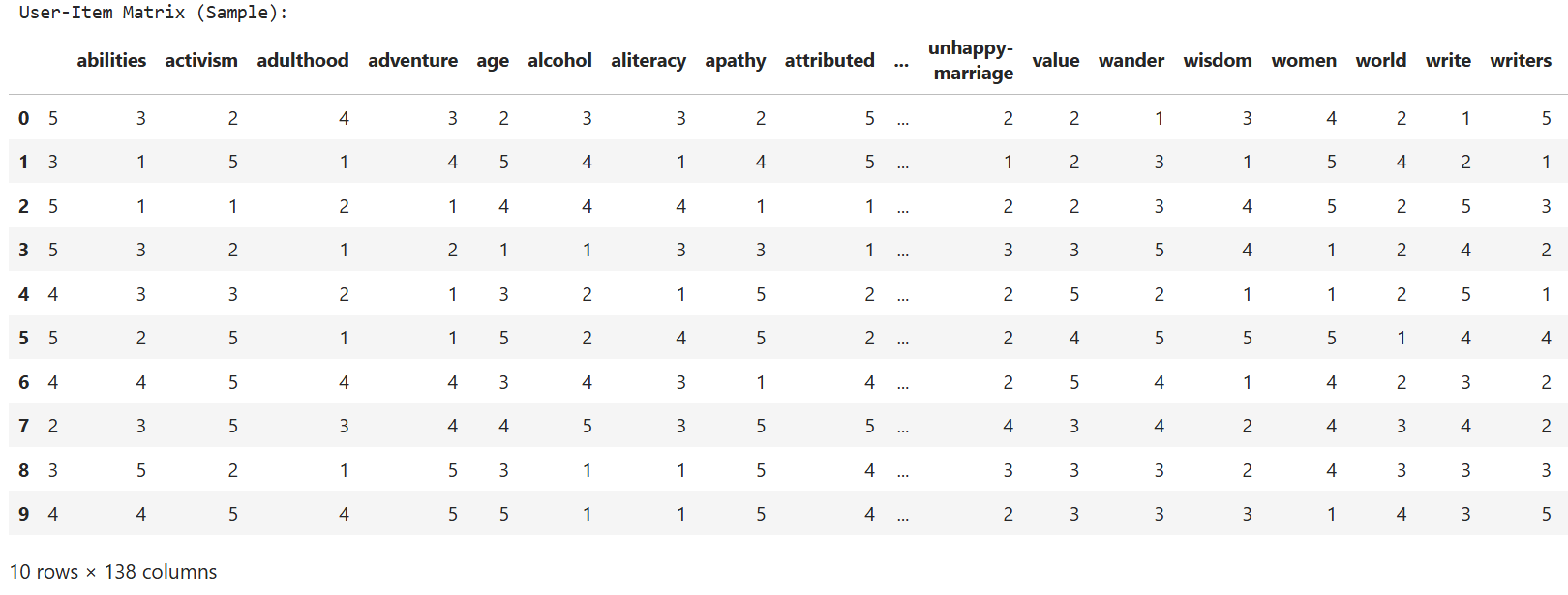
**Figure 1:**  
This table lists the unique tags (genres) extracted from the Quotes to Scrape dataset. These tags represent different themes or categories associated with each quote and serve as items in the user-item matrix.

**4. Data Collection and Preprocessing**

**Data Collection Process:**  
Data was collected using Python and the BeautifulSoup library to scrape quotes, authors, and tags from the quotes website. The data scraping process iterated through 10 pages, gathering all quotes and their associated tags.

**Unique Tags Extraction (Figure 1):**After data collection, the unique tags associated with each quote were extracted to serve as the item features for the user-item matrix.

**User-Item Matrix Generation (Figure 2):**A user-item matrix was simulated with random integer values ranging from 1 to 5, representing user ratings for each tag. This matrix serves as the primary dataset for testing the collaborative filtering algorithms.

* 

**Figure 2:**

This matrix is a simulation of user ratings for different tags (genres). Each row represents a user, and each column represents a tag. The matrix values range from 1 to 5, indicating hypothetical user preferences for each genre. This matrix is used to calculate similarities and generate recommendations.

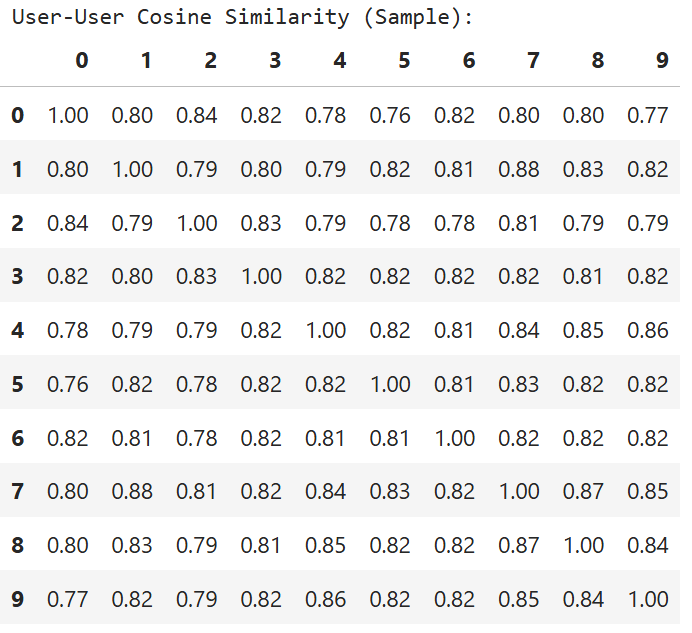
**5. Methodology and Implementation**

**User-Based and Item-Based CF Algorithms:**

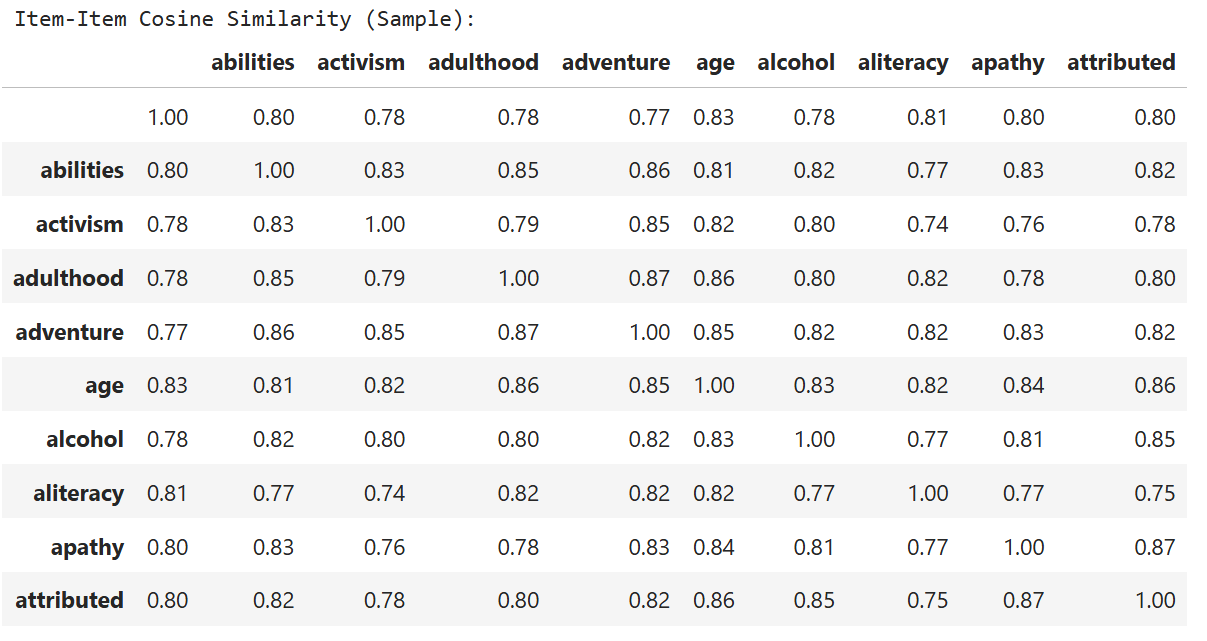
* **User-Based CF:** Predicts a user’s interest based on similar users.
* **Item-Based CF:** Recommends items based on similarities between items.

**Similarity Measures (Figure 3, 4, 5, and 6):**Cosine similarity and Pearson correlation coefficient were applied to compute the similarity for both users and items.

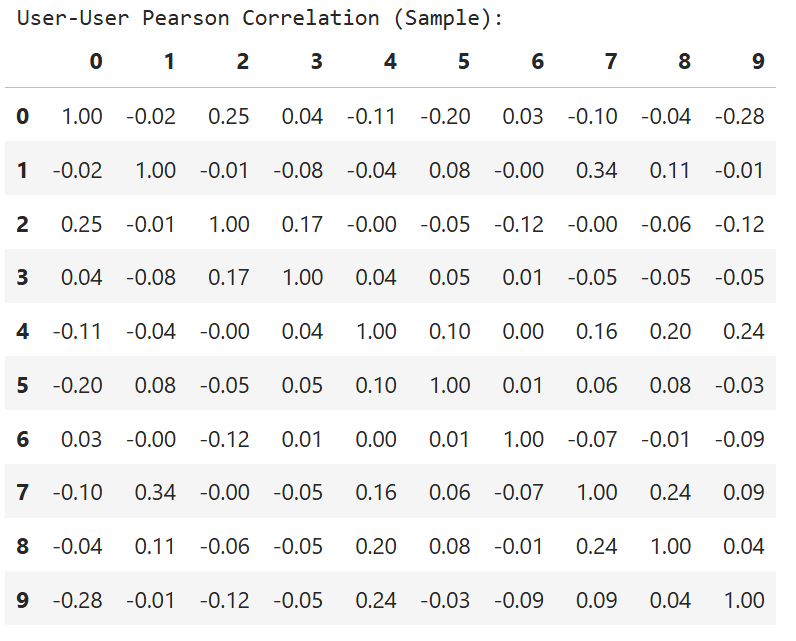
**Rating Prediction Calculation (Figure 7):**   
Using the similarity matrices, predicted ratings were calculated to assess user preferences and generate top-N recommendations.

* 

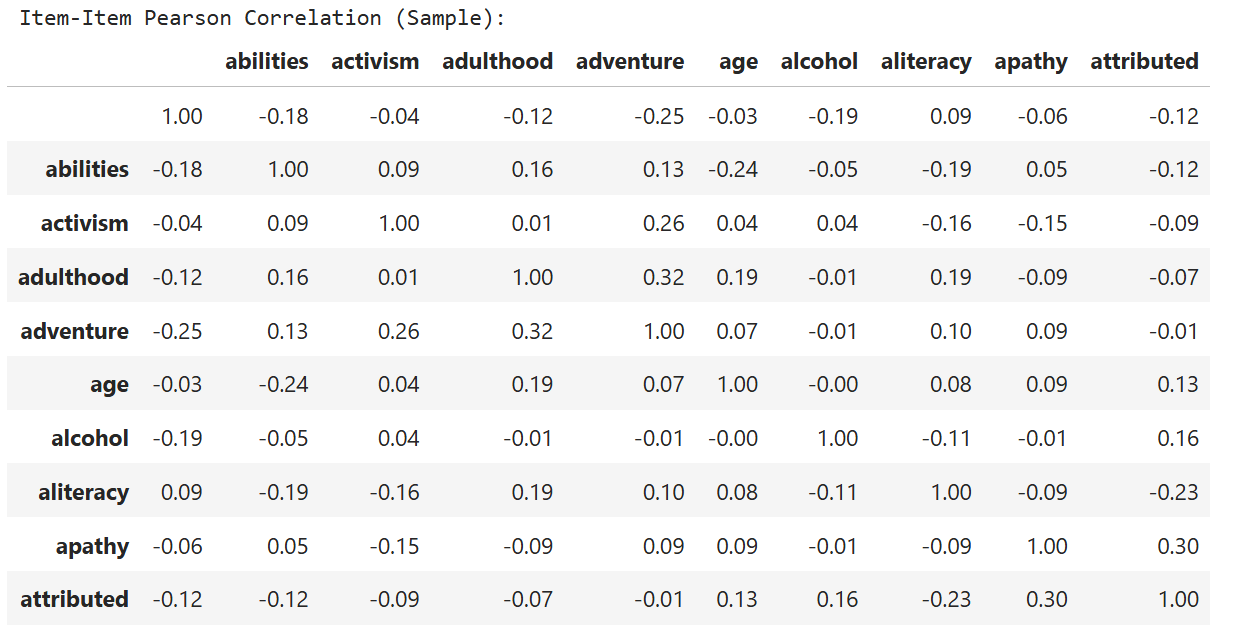
**Figure 3:**  
This matrix displays the cosine similarity between users based on their tag preferences. Higher similarity values indicate users with more similar tastes. This snapshot shows a 10x10 sample for readability.snapshot displays a 10x10 sample.quote and serve as items in the user-item matrix.

* 

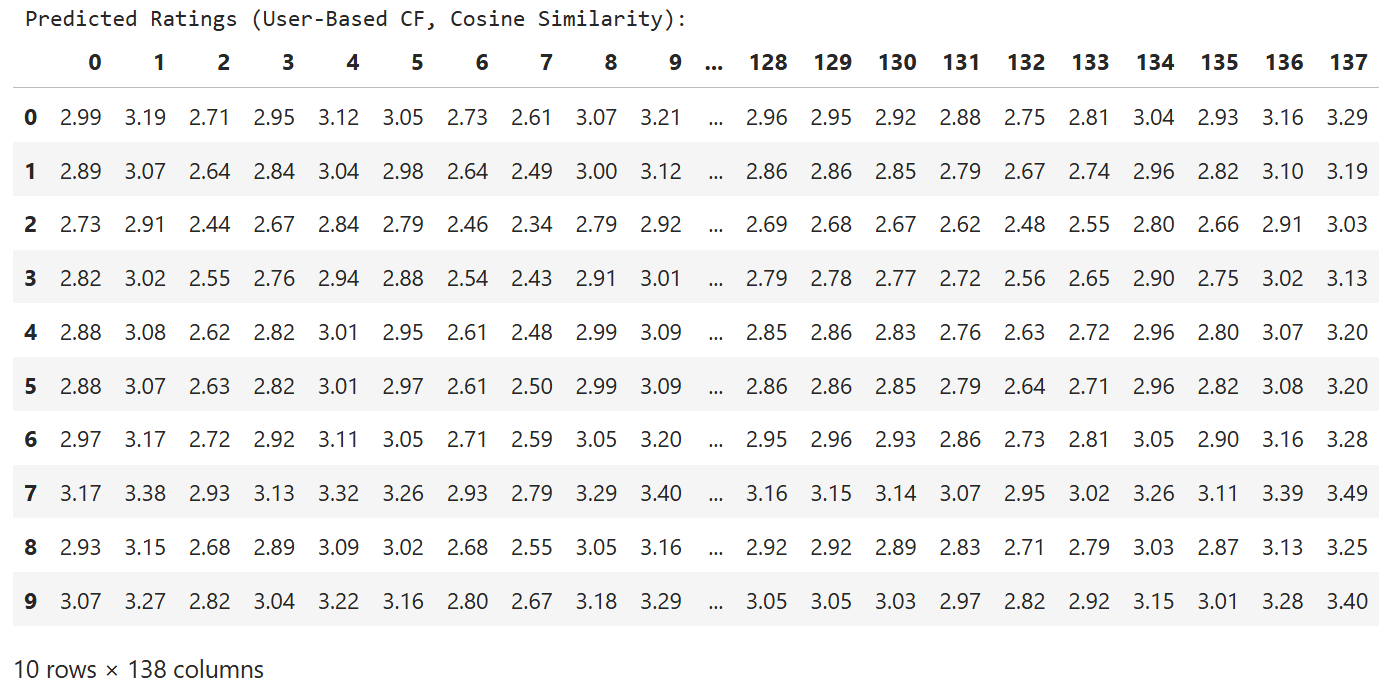
**Figure 4:**  
This matrix presents a sample of the item-item cosine similarity matrix, which measures the similarity between different tags based on user ratings. High values indicate greater similarity between tags, aiding in item-based collaborative filtering to recommend tags similar to those a user already likes.

* 

**Figure 5:**  
This sample showcases the Pearson correlation matrix for users, reflecting how user preferences align. Positive correlations indicate similar taste, while negative ones suggest opposite preferences. This is crucial for identifying peer groups in user-based collaborative filtering.

* 

**Figure 6:**  
This matrix displays Pearson correlation coefficients for different tags, measuring linear relationships between them. Positive values suggest tags are liked together, while negative values imply a differing preference pattern, which informs item-based collaborative filtering.



**Figure 7:**  
This table shows predicted ratings using User-Based Collaborative Filtering with Cosine Similarity. Each row represents a user, and each column represents a tag or item. The values indicate how much a user is predicted to like each tag, helping to suggest items with higher scores for personalized recommendations.

**6. Assignment Results**

**Average Rating Calculation (Figure 8):**  
Average ratings for each tag were calculated across users, providing insights into general preferences.

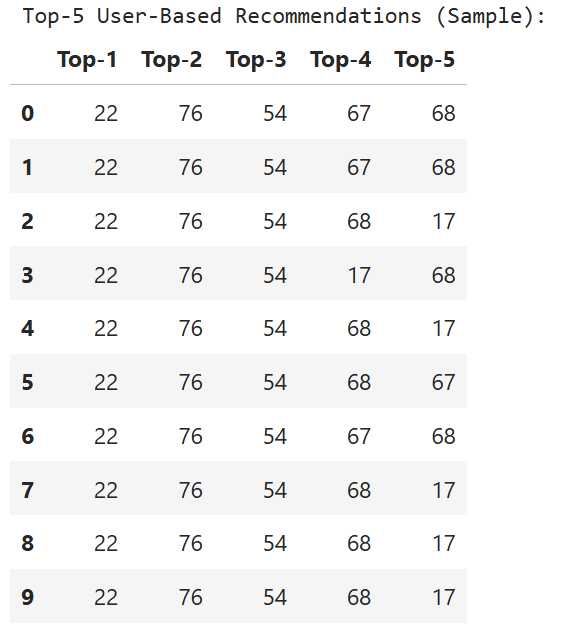
**Cosine Similarity vs. Pearson Correlation Results:**The similarity was computed using both cosine and Pearson methods. Cosine similarity focused on angle-based similarity, while Pearson captured linear correlation.

The cosine similarity and Pearson correlation results differ in how they capture relationships between users and items. Cosine similarity values are generally higher because this measure only considers the angle between ratings, focusing on the direction of preferences without accounting for rating scale differences. This makes it effective for identifying general similarities but may overlook nuanced rating behaviors. Conversely, Pearson correlation adjusts for individual rating tendencies, reflecting how closely users' rating patterns linearly align, including variations in rating levels. As a result, Pearson correlation values tend to be lower but offer a more refined view of rating relationships, particularly for users or items with distinct rating styles. Overall, cosine similarity provides a broader similarity measure, while Pearson correlation reveals subtler, bias-adjusted similarities.

**Top-N Recommendations:**Top-5 recommendations for each user were derived based on cosine similarity for both user-based and item-based approaches.



**Figure 8:**  
This table shows the average ratings assigned to each tag in the dataset, calculated across all users. Each row represents a specific tag (or genre), with the "Average Rating" column indicating the mean rating value that users have collectively assigned to that tag. This provides insights into the overall preference levels for different themes, with higher average ratings suggesting that users generally find content with those tags more appealing. Tags such as "abilities," "adventure," and "attributed" have higher average ratings, indicating positive user interest in these themes.



**Figure 9:**  
This table displays the top-5 recommendations for a sample of 10 users based on user-based collaborative filtering. Each row represents a user, and the columns labeled "Top-1" through "Top-5" indicate the recommended items, ranked in order of predicted relevance or preference. These recommendations are generated using a similarity-based approach, which identifies users with similar preferences and suggests items that these similar users rated highly. The table provides insight into the algorithm’s suggestion patterns, with certain items (e.g., items 22, 76, 54) frequently appearing across different users, indicating these items are popular or highly relevant among similar users in the dataset.

.

**7. Conclusion and Opinion**

**Summary:**The collaborative filtering models successfully recommended items by analyzing similarities between users and items. While cosine similarity performed well in identifying comparable preferences, Pearson correlation provided a deeper analysis of the relationships.

**Opinion:**User-based CF is more adaptable for systems with diverse users but fewer items, while item-based CF is suitable for static item catalogs. Future improvements could involve incorporating more detailed user demographics and context-aware factors.

**8. References**

[1] *BeautifulSoup Documentation*, available: [www.crummy.com/software/BeautifulSoup/bs4/doc/](http://www.crummy.com/software/BeautifulSoup/bs4/doc/)

[2] *Scikit-learn Cosine Similarity Documentation*, available: scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine\_similarity.html

[3] *Quotes to Scrape*, available: quotes.toscrape.com

[4] *OpenAI's ChatGPT*, assistance in generating, formatting, and structuring content