**User Guide**

**Step 1: Setting Up the Environment**

* Before running the code, make sure you have the necessary libraries installed. You will need pandas and numpy for this code to work. You can install them using pip:

bash

Copy code

pip install pandas numpy

**Step 2: Preparing the Dataset**

* Place your dataset file named u.data in the same directory as your Python script.
* Ensure the dataset file is in CSV format with the following columns: user\_id, movie\_id, rating, timestamp.

**Step 3: Running the Code**

* Open a terminal or command prompt and navigate to the directory containing the Python script and the dataset file.
* Run the Python script using the command:

bash

Copy code

python userbasedcollabfilter.py

**Step 4: Interacting with the Code**

* The code will prompt you for user input. Follow the prompts to interact with the code:
* For example, when prompted, enter a user ID and a movie ID to predict the rating. The validity of input is checked.
* The code will then calculate and display the predicted rating first using the Pearson function, and then alternatively the Cosine function.

**Explanation of Methods Used**

**A) Load the Dataset**

* ***pandas*** is used to read the dataset from a CSV file. It provides powerful tools for data manipulation and analysis.
* ***numpy*** is used for numerical computations. It is efficient for operations involving arrays and matrices.

**B) User-Based Collaborative Filtering**

* Pearson correlation is used as the similarity metric. It measures the linear correlation between two sets of ratings.

**C) Predict Movie Scores**

* Predicts a movie score for a specified user and movie using user-based collaborative filtering with Pearson correlation.

**D) Top Similar Users and Recommended Movies**

* Computes the top 10 most similar users to a selected user based on Pearson correlation.
* Recommends the top 10 movies for the selected user based on predicted ratings.

**E) New Similarity Function**

The cosine similarity function is advantageous for collaborative filtering in recommendation systems because it emphasizes the direction of user preferences rather than the specific numerical ratings. This makes it scale-invariant and well-suited for scenarios where relative preferences are more significant than absolute values. Additionally, it handles sparse data effectively and is computationally efficient, making it a practical choice for large datasets.

**Considerations of the Goals of the Recommendation System:**

* **Linear Relationships:** If the strength of linear relationships between ratings is a crucial factor for similarity, Pearson correlation might be a better choice. Pearson correlation is sensitive to both the direction and magnitude of the relationship between ratings. This can be important if users who rate movies similarly tend to have a linear relationship in their ratings.
* **Direction of Ratings (Likes/Dislikes):** On the other hand, if the direction of ratings (i.e., which movies users like or dislike) to be more significant, then cosine similarity is more appropriate. It focuses on the direction of user preferences and is not influenced by the absolute magnitude of ratings.