

Cosine Similarity-Based Collaborative Filtering

1. **Create the User-Item Matrix:** A matrix is created where rows represent users (User_ID), columns represent products (Product_ID), and the values are the ratings (Rating) users gave to products. Any missing values are filled with 0 to ensure the matrix is complete.
2. **Calculate Similarities:** Using the `cosine_similarity` function, the similarity between users is computed based on their ratings. This results in a similarity matrix, where each entry represents how similar two users are.
3. **Recommend Products for a Specific User:**
 - For a given user (`user_id`), the most similar users are identified.
 - Products that these similar users rated highly but the target user hasn't rated yet are selected.
 - These products are sorted by their ratings, and the top N recommendations are returned.

Deep Learning-Based Collaborative Filtering

1. **Prepare Data:**
 - The user-item matrix is converted to a NumPy array for compatibility with machine learning models.
 - The data is split into training (80%) and testing (20%) sets to evaluate performance.
 - The training data is normalized (mean-centered and scaled to have a standard deviation of 1) to improve the neural network's ability to learn.
2. **Define a Neural Network:**
 - A simple neural network is built using Keras with the following layers:
 - **Input Layer:** Takes the user's ratings as input.
 - **Hidden Layers:** Two dense layers with 128 and 64 neurons, respectively, using ReLU activation to capture complex patterns in the data.
 - **Output Layer:** Predicts ratings for all products.
 - The network is trained to reconstruct the user's ratings (autoencoder-style learning), using mean squared error (MSE) as the loss function.
3. **Train the Model:** The neural network is trained on the normalized training data for 10 epochs with a batch size of 32. Validation data (10% of the training set) is used to monitor the model during training.
4. **Generate Predictions:**
 - The trained model predicts normalized ratings for the test data.
 - Predictions are denormalized (scaled back to the original range) to match the actual rating scale.
5. **Recommend Products for a Specific User:**
 - For a given user, the model's predicted ratings for all products are retrieved.
 - The top N products with the highest predicted ratings are identified and returned as recommendations.