import org.apache.mahout.cf.taste.eval.DataModelBuilder;

import org.apache.mahout.cf.taste.impl.model.file.FileDataModel;

import org.apache.mahout.cf.taste.model.DataModel;

import org.apache.mahout.cf.taste.neighborhood.UserNeighborhood;

import org.apache.mahout.cf.taste.impl.neighborhood.NearestNUserNeighborhood;

import org.apache.mahout.cf.taste.similarity.UserSimilarity;

import org.apache.mahout.cf.taste.impl.similarity.PearsonCorrelationSimilarity;

import org.apache.mahout.cf.taste.recommender.Recommender;

import org.apache.mahout.cf.taste.impl.recommender.GenericUserBasedRecommender;

import org.apache.mahout.cf.taste.recommender.RecommendedItem;

import java.io.File;

import java.util.List;

/\*\*

\* AI-Based Recommendation System

\* Recommends products to users based on user-item preferences using Apache Mahout.

\*/

public class ProductRecommender {

public static void main(String[] args) {

try {

// Load CSV file

File file = new File("recommender.csv");

DataModel model = new FileDataModel(file);

// Compute similarity between users

UserSimilarity similarity = new PearsonCorrelationSimilarity(model);

// Find nearest neighbors (e.g., top 2)

UserNeighborhood neighborhood = new NearestNUserNeighborhood(2, similarity, model);

// Create user-based recommender

Recommender recommender = new GenericUserBasedRecommender(model, neighborhood, similarity);

// Recommend 3 products for user with ID = 3

List<RecommendedItem> recommendations = recommender.recommend(3, 3);

// Output recommendations

System.out.println("=== Recommendations for User 3 ===");

for (RecommendedItem item : recommendations) {

System.out.printf("Recommended Item: %d | Score: %.2f\n", item.getItemID(), item.getValue());

}

} catch (Exception e) {

System.err.println("Error: " + e.getMessage());

}

}

}