**Goal: Suggest products based on user ratings using Apache Mahout (Item-based recommendation).**

<dependencies>

<dependency>

<groupId>org.apache.mahout</groupId>

<artifactId>mahout-core</artifactId>

<version>0.9</version>

</dependency>

</dependencies>

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

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

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

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

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

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

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

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

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

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

Import java.io.File;

Import java.util.List;

Public class ProductRecommender {

Public static void main(String[] args) {

Try {

// Load data from CSV

DataModel model = new FileDataModel(new File(“data.csv”));

// Define similarity

UserSimilarity similarity = new PearsonCorrelationSimilarity(model);

// Define neighborhood

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

// Build recommender

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

// Recommend products for user 1

List<RecommendedItem> recommendations = recommender.recommend(1, 2);

// Display output

System.out.println(“Recommendations for User 1:”);

For (RecommendedItem recommendation : recommendations) {

System.out.println(“Product ID: “ + recommendation.getItemID() +

“, Predicted Rating: “ + recommendation.getValue());

}

} catch (Exception e) {

e.printStackTrace();

}

}

}