```
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.recommender.RecommendedItem;

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

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

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

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

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

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

import java.io.File;

import java.util.List;

public class RecommenderSystem {


 public static void main(String[] args) throws Exception {

 // Load data from CSV
       File ratingsFile = new File("data.csv");

  DataModel model = new FileDataModel(ratingsFile);

       // Compute user similarity
       UserSimilarity similarity = new PearsonCorrelationSimilarity(model);

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

       // Build the recommender
       UserBasedRecommender recommender = new GenericUserBasedRecommender(model,
neighborhood, similarity);

       // Recommend items for user 1
```

```
        int userId = 1;
        List<RecommendedItem> recommendations = recommender.recommend(userId, 3);

        System.out.println("Recommendations for user " + userId + ":");

 for (RecommendedItem recommendation : recommendations) {

 System.out.println("Item: " + recommendation.getItemID() + ", Score: " +
recommendation.getValue());
        }
    }
}
```