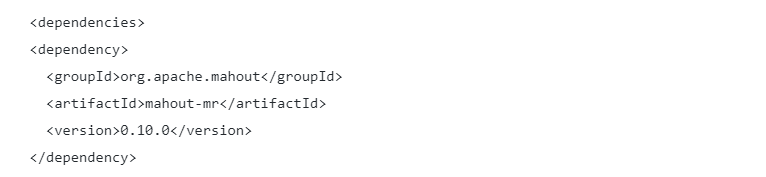
**Creating a User Based Recommendation System — Java | Apache Mahout**

Let’s create a User Based Recommendation System in Java using Apache Mahout. Apache Mahout is an open source project which is widely used to build recommendation Engines.

*To begin with, we need to have a Data with us. Below is a generic Data set which is in the format of UserID , Item Number , Rating.*[Example: “2,11,2.0” means User#2 has rated Item#11 as 2 stars] *Lets us save the below data set as MDist1.csv file, anywhere on our computer system.*

Now on your Java IDE, create a Maven Project, modify the pom.xml as below:



After this, introduce the below souce code file:

|  |
| --- |
| import java.io.File; |
|  | import java.util.List; |
|  |  |
|  | import org.apache.mahout.cf.taste.impl.model.file.FileDataModel; |
|  | import org.apache.mahout.cf.taste.impl.neighborhood.ThresholdUserNeighborhood; |
|  | import org.apache.mahout.cf.taste.impl.recommender.GenericUserBasedRecommender; |
|  | import org.apache.mahout.cf.taste.impl.similarity.CityBlockSimilarity; |
|  | 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.UserBasedRecommender; |
|  |  |
|  | public class ManhattenDistanceSimilarityExample { |
|  |  |
|  | public static void main(String[] args) { |
|  | try { |
|  | DataModel model = new FileDataModel(new File("D:\\DataScienceCollection\\mahout\\MDist1.csv")); |
|  | CityBlockSimilarity similarity = new CityBlockSimilarity(model); |
|  | UserNeighborhood neighborhood = new ThresholdUserNeighborhood(0.1,similarity, model); |
|  | UserBasedRecommender recommender = new GenericUserBasedRecommender(model, neighborhood, similarity); |
|  |  |
|  | // The First argument is the userID and the Second parameter is 'HOW MANY' |
|  | List<RecommendedItem> recommendations = recommender.recommend(2, 2); |
|  |  |
|  | for (RecommendedItem recommendation : recommendations) { |
|  | System.out.println(recommendation); |
|  | } |
|  | } catch (Exception e) { |
|  | System.out.println("Exception occured !"); |
|  | } |
|  |  |
|  | } |
|  |  |
|  | } |

In the above program we have mentioned our source CSV file name).

The notable point over here is the similarity threshold which is a lower limit for the similarity of two data records that belong to the same cluster. For example, if you set the similarity threshold to 0.25, data records with field values that are 25% similar are likely to be assigned to the same cluster.If you specify a similarity threshold of 1.0,then you are insisting that, for customers to appear in the same group, their characteristics must be identical.

Finally we are interested in calculating n recommendations (Here 2)for User#2. To proceed further we need to execute the above program. The results are as generated in decreasing order, as below:

Recommended Item[item:12, value:4.857143]

Recommended Item[item:14, value:3.357143]