MOVIE RECOMMENDATION SYSTEM:

Part 1: Building the recommendation system

Step 1: Optimizing the Loss function with ALS:

First I divide the dataset into Training, Test and Validation and then use the product of factors technique and optimize the loss function by changing various features of the ALS model (mainly: rank).
I check for the lowest RMSE value and select that as our best model.

I then get train the best model and predict the test dataset with the best model and then get the optimal values for test dataset- RMSE, MSE.

Map to find the respective Files for this task:

Code: CodeFiles/ recommendation.py
Output: Output/Output_recomm_py

Sample Output:

For rank 5
MSE is 0.665895305582
RMSE is 0.816024083948

For rank 8
MSE is 0.656784463945
RMSE is 0.810422398472

For rank 12
MSE is 0.654998883731
RMSE is 0.809320013178
The best model was trained with rank 12

For testing data
MSE: 0.654945811622
RMSE is 0.809287224428

Step 2: Evaluating the Model (Cross Validation)

First I ran the shell script (which I modified to take ratings.csv file as its input) to get the 5 folds for cross validation.

File Names	Description
r1.train, r2.train, r3.train, r4.train, r5.train r1.test, r2.test, r3.test, r4.test, r5.test	The data sets r1.train and r1.test through r5.train and r5.test are 80%/20% splits of the ratings data into training and test data. Each of r1, ..., r5 have disjoint test sets; this if for 5 fold cross validation (where you repeat your experiment with each training and test set and average the results).

Reference: http://files.grouplens.org/datasets/movielens/ml-10m-README.html

Now, I repeat the experiment with each training and tests sets and average the result (RMSE, MSE and MAP)
For getting the MAP value, Rankingmetrics is used

Code: CodeFiles/crossValidation.py
Output: Output/Output_cv

Sample Output:

('The MSE value after Cross Validation is: ' 0.6183384684207281)
('The RMSE value after Cross Validation is: ' 0.7863454672983216)
('The MAP value after Cross Validation is: ' 0.8078338976328463)

Remarks:

Based on the results above, I have successfully evaluated my model using 5 – fold cross validation. The MSE and RMSE values closely match each other with the values of the best model (recommendation.py).

Part 2: Adding the user to the Database

Add a new user with User Id =0, (Because 0 is not there as a user Id in the database), and provide ratings to a few movies (I have rated almost 15 movies). Then use the best model (with rank =12) to get the predictions for this particular user for the movies he has not rated. Sample output below shows about 10 predicted recommendations.

Sample Output:

Rating(user=0, product=6400, rating=3.9777418987988806),
Rating(user=0, product=81100, rating=3.0627367990520633),
Rating(user=0, product=105040, rating=2.7796447539317217),
Rating(user=0, product=88400, rating=3.786483724058922),
Rating(user=0, product=7020, rating=4.037491579364013),
Rating(user=0, product=65845, rating=3.2159465610092024),
Rating(user=0, product=32170, rating=3.841737853358807),
Rating(user=0, product=1325, rating=2.168484237962208),
Rating(user=0, product=113470, rating=3.8084120904941567),
Rating(user=0, product=100270, rating=3.660690429186826 7),

CONTENTS OF ASSIGNMENT ZIP FILE:

1. Readme file

Directories:

1. CodeFiles: Contains all the code files
   - crossValidation.py
   - recommendation.py
2. Pseudo-code: Contains all the PseudoCode files
   - Cross Validation.txt
   - Recommendation_PseudoCode.txt
3. Output: Contains the sample output files.
   - Output_cv
   - Output_recomm_py