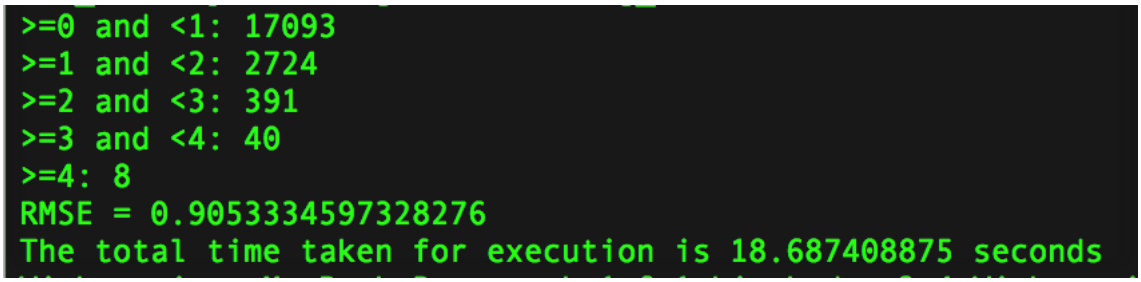
Assignment 3 - Recommendation System

Description:

Missing values in prediction were handled using the imputation technique of that user's average movie rating for both task1 and task 2.
The outliers, as in values lesser than 0 were equated to 0 and those values greater than 5 were equated to 5 in order to keep range of ratings between 0-5 for both task1 and task2.
I also used nearest neighbors approach while implementing a user based CF (task2) to pick the weights most similar to user and used only those weights for prediction by picking an N value.

TASK 1 –MODEL-BASED Algorithm:



How to run?
1. Move the input files (ratings.csv, testing_small.csv) and source code file (jar file) inside the spark-1.6.1-bin-hadoop2.4 folder in your machine
2. In terminal enter the same spark-1.6.1-bin-hadoop2.4 directory and run the following command.

Command to enter directory -->

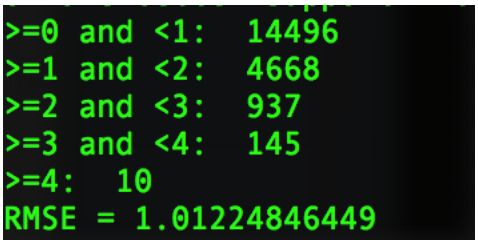
cd spark-1.6.1-bin-hadoop2.4

Command used to run source code -->

./bin/spark-submit --class Vishnupriya_Ravibalan_task1 Vishnupriya_Ravibalan_task1.jar ratings.csv testing_small.csv

TASK 2 – USER-BASED CF Algorithm using Pearson Correlation:

Accuracy obtained for task 2:



How to run?
1. Move the input files and source code file inside the spark-1.6.1-bin-hadoop2.4 folder in your machine
2. In terminal enter the same spark-1.6.1-bin-hadoop2.4 directory and run the following command.

Command to enter directory -->

cd spark-1.6.1-bin-hadoop2.4

Command used to run source code -->

./bin/spark-submit Vishnupriya_Ravibalan_task2.py ratings.csv testing_small.csv Vishnupriya_Ravibalan_result_task2.txt

<u>if you want to see the time taken for code to execute like task 1, please use :</u>

time ./bin/spark-submit Vishnupriya_Ravibalan_task2.py ratings.csv testing_small.csv Vishnupriya_Ravibalan_result_task2.txt

and Output will be: