Vijayakumar Gedigeri INF553 HW 3, 20th March 2017

Below is a brief description of implementation of task 1, task 2 and how to run the programs. First task is implemented in Scala for Spark 2.0.1. Second Task is implemented in Python 2.7 written for Spark 2.0.1

Task 1: Model-based CF Algorithm

Spark MLLIB is used to build the model based recommendation system. The model is trained using training records from ratings.csv file after removing the test data from testing_small.csv for small data and testing_20m.csv for large data. Once the mode is trained, prediction is done for testing data of (User, Movie) pair. For movies that do not have any ratings by any user in training data, and not predicted by the model, average rating of that user (From testing data predictions) is used to fill the missing predictions. Outliers in the predicted results are identified, and their ratings are replaced by average rating of user to improve the recommendation accuracy.

Instructions to run the program (With Scala file Vijayakumar_Gedigeri_Task1.Scala and jar Vijayakumar_Gedigeri_Task1.jar)

Output file will be created in the directory **progOutput** which will have a single file **part-00000**

1. Small Data

spark-submit --class Vijayakumar_Gedigeri_Task1 Vijayakumar_Gedigeri_Task1.jar ml-latest-small\ratings.csv testing_small.csv

Output: Accuracy

```
>=0 and <1: 14465
>=1 and <2: 4474
>=2 and <3: 1043
>=3 and <4: 241
>=4: 33
RMSE = 1.0519623554044326
```

2. Large Data

spark-submit --driver-memory 4g --class Vijayakumar_Gedigeri_Task1 Vijayakumar_Gedigeri_Task1.jar ml-20m\ratings.csv testing_20m.csv

Output: Accuracy

```
>=0 and <1: 3251572
>=1 and <2: 699569
>=2 and <3: 91254
>=3 and <4: 11410
>=4: 638
RMSE = 0.8256381186180838
```

Task 2: Item-based CF Algorithm

Here, Item-Based CF algorithm is used to predict the ratings for (User, Movie) pair. Like task1, test data is separated from training data. Pearson Correlation for all movies (From testing data) is first calculated for similarity among movie pairs. Co-rated users' rating is used as average rating for the movies while calculating the similarity. To calculate the rating for movie, nearest neighborhood of movies is followed with choosing around 40% of all movies that are like the movie being predicted. For cold-start movie problem, average rating of users is used to fill the missing rating.

Output will be written to file **Vijayakumar_Gedigeri_result_task2.txt** in the current directory.

Instructions to run the program (With Python file Vijayakumar_Gedigeri_Task2.py)

spark-submit Vijayakumar_Gedigeri_Task2.py ml-latest-small\ratings.csv testing_small.csv

Output: Accuracy

```
 >=0 and <1: 14755
>=1 and <2: 4343
>=2 and <3: 994
>=3 and <4: 160
>=4: 4
RMSE = 0.993447749861
```