Decription:

1. Implementation:
   a) Task 1:

      As for task1, it is simply invoking Scala's collaborative filtering methods which are in mlib library. The most important thing is to choose parameters. As for this task, author choose rank = 4, numInterations = 10 and lamda = 0.01 to get the result. As for the missing users, author set the predict rate for them as the mean rate of all training data set. The outlier ratings are set to 5 if it is bigger than 5. There is no one smaller than 0.

```
>=0 and <1: 14280
>=1 and <2: 4493
>=2 and <3: 1142
>=3 and <4: 264
>=4: 77

Rooted Mean Squar Error = 1.0884995149716175
total running time16.436538665s
```

   b) Task 2:

      As for task2, author wrote a user-based collaborative filtering because there are 671 distinct users and more than 5000 distinct movies. Using user-based recommender can reduce running time.

      Secondly, author use 2 matrix to save data. Each column of the utility matrix represent a movie and each row represent a user. The correlation matrix store the Pearson Correlation between 2 users. These 2 matrix are obtained after training stage. Note that when calculating correlation, denominator could be 0. If so, the correlation is set to 0.

      Thirdly, to calculate the predict rate, function given in class is used. The denominator here is rounded to 2 decimal. If the prediction rate is bigger than 5, it is set to 5. Neighbor number is equal to 15, which means the top 15 nearest users of the predicting one is chose to calculate the predicting rate.

      As for the missing users, the predict rate is set to mean rate of training set.

```
>=0 and <1: 15064
>=1 and <2: 4228
>=2 and <3: 811
>=3 and <4: 135
>=4: 18

RMSE::0.9565171828193761
duration::126.968683039s
```

2. Command line:

Task1:

Task2: