# Performance Test of Enhancing User Experience Rcommender System

## Introduction

### Source of dataset

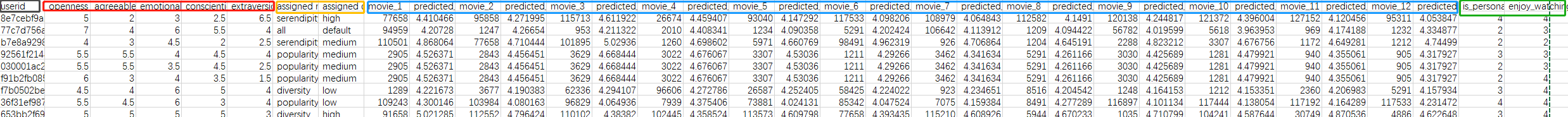
This dataset comes from <https://grouplens.org/datasets/personality-2018/> including two files (rating.csv and personality.csv)

And used in Article:

<Enhancing User Experience with Recommender Systems Beyond Prediction Accuracies> --by Joseph A. Konstan and Loren Terveen August, 2016

They used rating.csv train the model and output the prediction of recommender system with user’s feedback.

### Content of dataset

Personality.csv

Divided into 5 parts.

(1835 lines of record = number of users)

1. UserId
2. User Personality (1- 10): they use The Big Five Personality Traits Model (which is known as OCEAN Model) - According to the OCEAN Model, there are only five main components of personality. These five components are **Openness, Consciousness, Extraversion, Agreeableness and Neuroticism [**which is replaced by emotional\_stability to make all component positive related](huge number get better performance).
3. Attribute of Movie: Assigned Condition: (high, medium, low).
4. Prediction with results
5. Feedback from User (1-5)

Rating.csv



Totally 1028751 lines of record = number of test those users are experienced.

### Analysis of dataset

Although the prediction of 12 movies with its result is a perfect material to test the performances. But there are no records (the feedback from users) in rating.csv.

For example, the prediction rating with userid=8e7cebf9a234c064b75016249f2ac65e is 4.410466 with movieId 77658. However, in rating.csv which contains all feedback from user, userid=8e7cebf9a234c064b75016249f2ac65e never see this movie. The author just offer us the feedback of whole system for each user which are is\_personalized and enjoy\_watching.

As a result, I can only find another symbol to represent the predication of each userid in the first step.

## Evaluation model version 1.0 with Pig

### Data processing

As we already know some part of dataset are useless for my work. I want to build a new dataset with all information and the most important – **combine the userid, personality with movieId and ratings**

**What I need is a inner join and then generate the useful lines.**

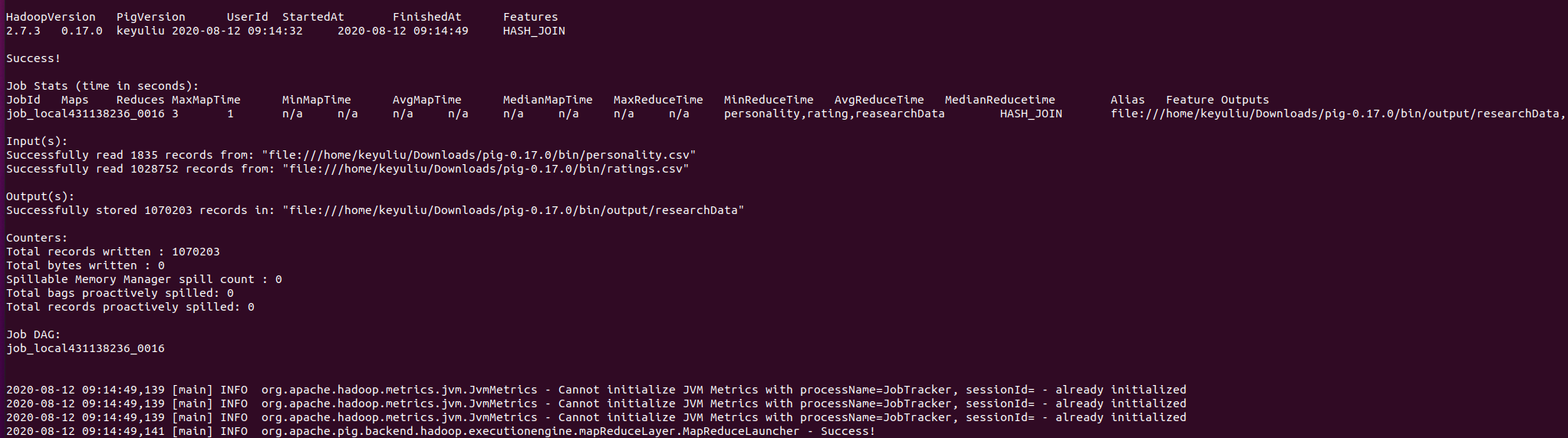
**In pig**

**Load File**

rating = LOAD 'ratings.csv' USING PigStorage(',') AS (useri, movie\_id, rating, tstamp);

personality = LOAD 'personality.csv' USING PigStorage(',') AS(userid, openness, agreeableness, emotional\_stability, conscientiousness, extraversion, assignedmetric, assignedcondition, movie\_1, predicted\_rating\_1, movie\_2, predicted\_rating\_2, movie\_3, predicted\_rating\_3, movie\_4, predicted\_rating\_4, movie\_5, predicted\_rating\_5, movie\_6, predicted\_rating\_6, movie\_7, predicted\_rating\_7, movie\_8, predicted\_rating\_8, movie\_9, predicted\_rating\_9, movie\_10, predicted\_rating\_10, movie\_11, predicted\_rating\_11, movie\_12, predicted\_rating\_12, is\_personalized, enjoy\_watching);

**reaserachData(included in project directory) - Data after inner join:**

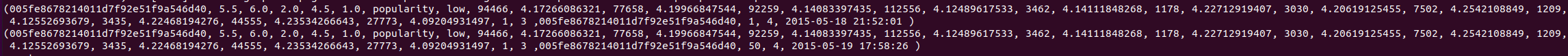


reasearchData = JOIN personality BY userid, rating BY useri;

STORE reasearchData INTO 'output/researchData' USING PigStorage(',');

Limit\_reasearchData = LIMIT reasearchData 2;

dump Limit\_reasearchData;



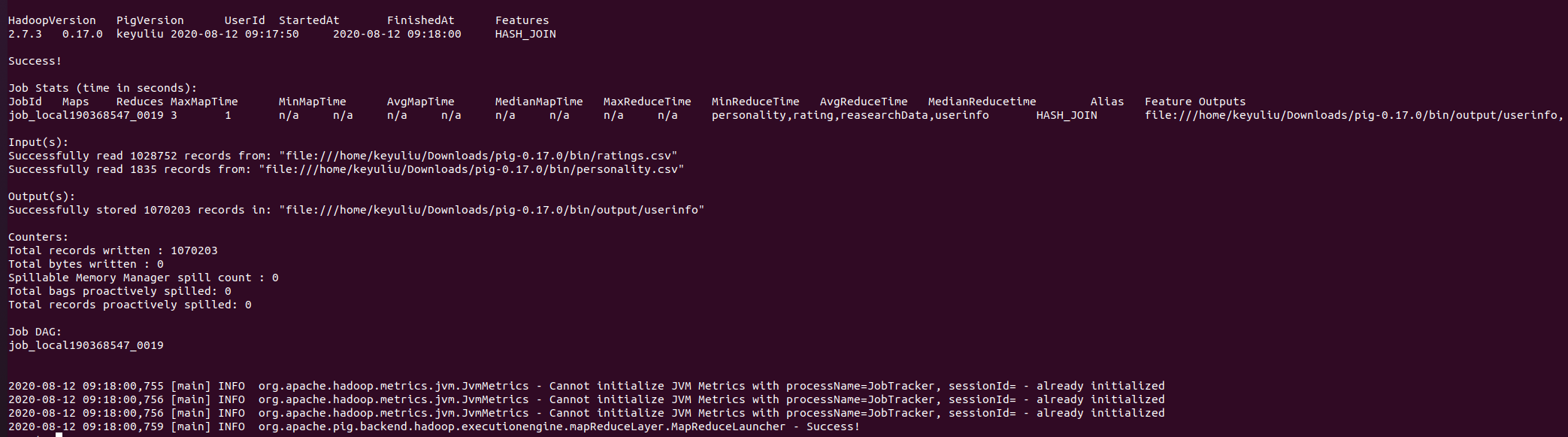
illustrate reasearchData;



**Userinfo: lines that are useful this time:(userid, openness, agreeableness, emotional\_stability, conscientiousness, extraversion, assignedmetric, assignedcondition, movie\_id, rating, is\_personalized, enjoy\_watching)**

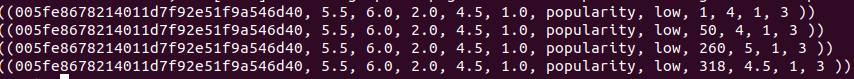
userinfo = FOREACH reasearchData GENERATE (userid, openness, agreeableness, emotional\_stability, conscientiousness, extraversion, assignedmetric, assignedcondition, movie\_id, rating, is\_personalized, enjoy\_watching);

STORE userinfo INTO 'output/userinfo’ USING PigStorage(',');



Limit\_userinfo = LIMIT userinfo 4;

dump Limit\_userinfo;



### Evaluation model 1.0

In version 1.0, I assume

Userinfo.Enjoy-watching is the representation of predicted result.

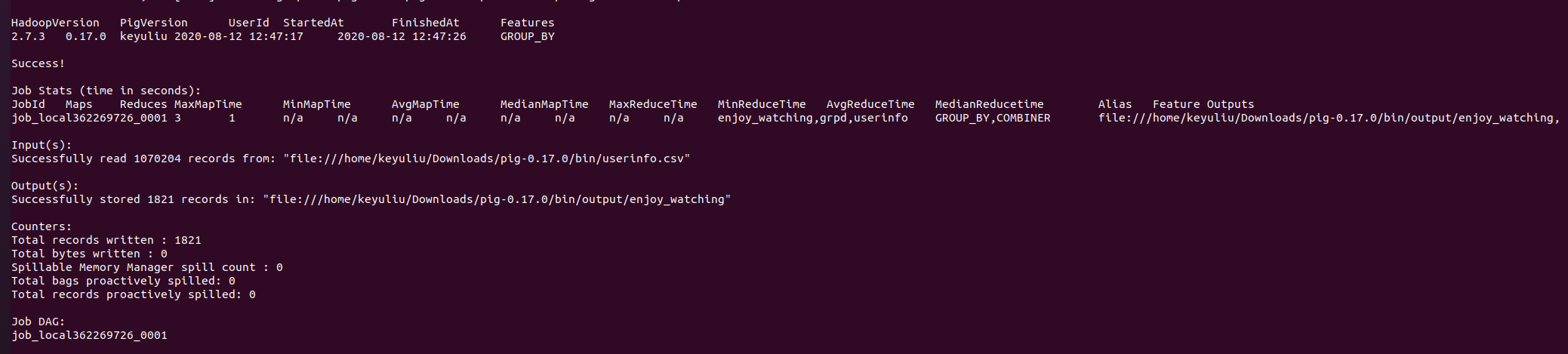
Userinfo.Rating is the representation of real feedback.

To calculate each user\_id’ s Enjoy-watching and Rating in userinfo. I manage the average mapreduce.

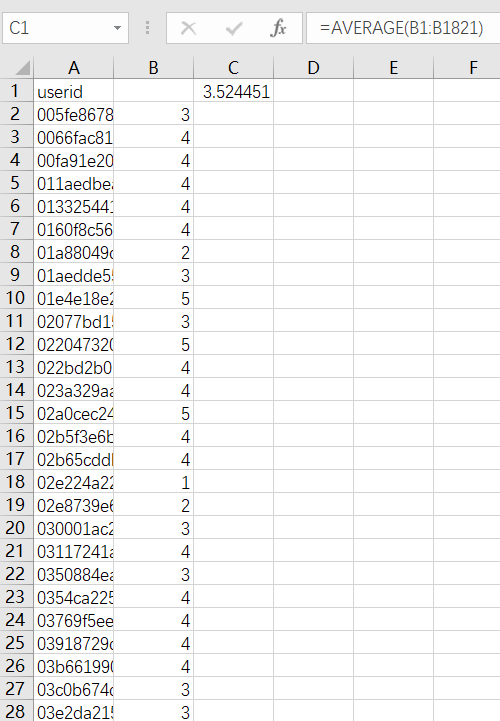
grpd = GROUP userinfo BY userid;

enjoy\_watching = FOREACH grpd GENERATE group, AVG(userinfo.enjoy\_watching);

STORE enjoy\_watching INTO 'output/enjoy\_watching' USING PigStorage(',');

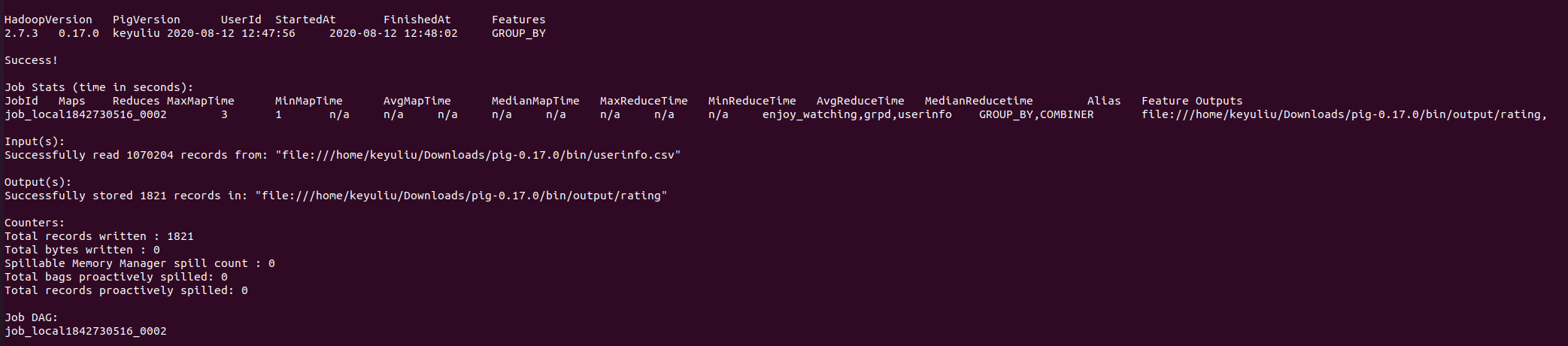


Then I read this .csv file with EXCEL and using =AVERAGE(B1:B1821) [In the screenshoot: the total records written = 1821] to get the Average of enjoy\_watching for all userid = 3.5244

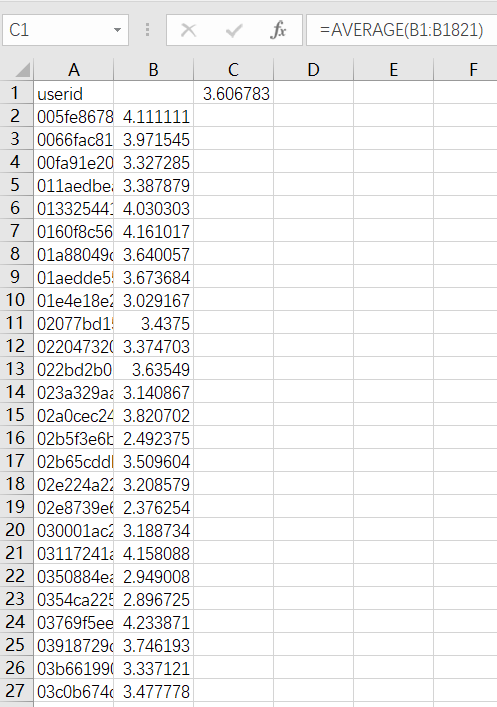
\

rating = FOREACH grpd GENERATE group, AVG(userinfo.rating);

STORE rating INTO 'output/rating' USING PigStorage(',');



Doing the same thing as enjoy\_watching. AVERAGE(B1:B1821) = 3.606783



**To calculate the average of all users, I assume all the users have the same property,** and in all assumptions above, the Average Absolute Different between prediction and feedback is :

****

This can give us a intuitional sense of the accuracy for this system – **The difference between the prediction of the system and the actual value is 0.082232**

## Evaluation model version 2.0 with HDFS MapReduce

### Limitation of model 1.0

In fact, people’s sense of values are different from one to another. So the Ocean model will be used to stratify people by their personality.

### Evaluation model version 2.0

* 1. New model for mapreduce

**Key :**

Using 5 factors (Openness, Consciousness, Extraversion, Agreeableness and emotional\_stability) as a key instead of userid.

**Value1 : representation of Predicted Result**

The userinfo.assignedcondition (high, midterm, low) will be assumed as the predictions for movies.

To transfer these new benckmarks from String to numbers in a resonable way. I will use quartering.

For example, there are totally 11 numbers from 0 to 10.

the value of assignedcondition (high, midterm, low) -> ()

High = (11+1) \* 0.75

miderm = (1+1) \* 0.5

low + (11+1) \* 0.25

**Value2 : representation of actual feedback**

is\_personalized is from and enjoy\_watching they are all from user’s feedback, So there gonna be a way to combine them together.

On the same time, Because is\_personalized is from (1-5) and enjoy\_watching is also from (1-5).

And they need to compare with the predicted result-assigned condition. So their value need to product 2。 And then, they are balanced when calculating absolute average difference.

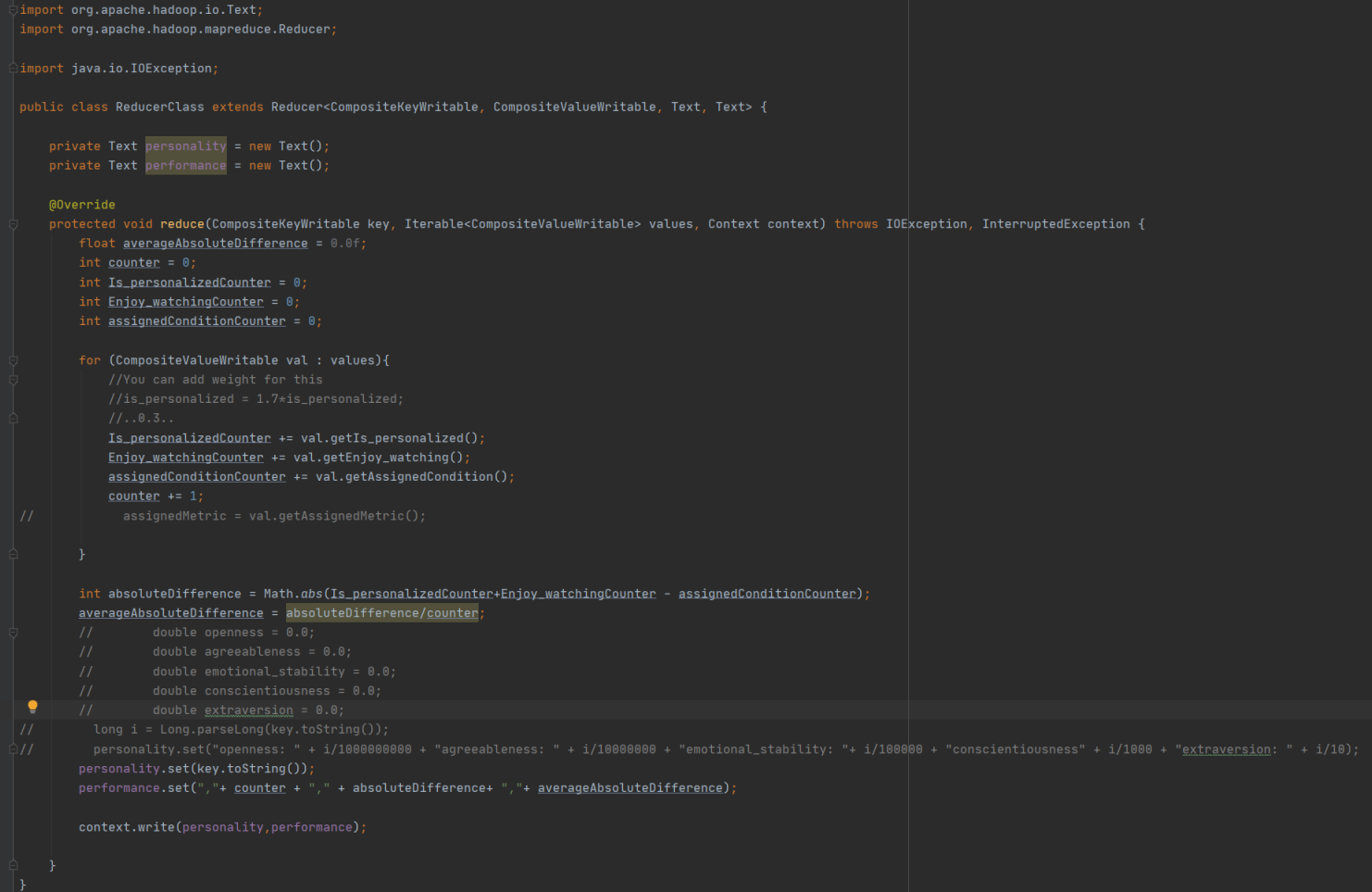
The representation of actual feedback = a\*is\_personalized +b\*enjoy\_watching (when a +b = 2)

* 1. Important Implementation (code is in EvaluationMapReduce\src\main\java)

compareTo and toString Method of custom writable comparable Key.

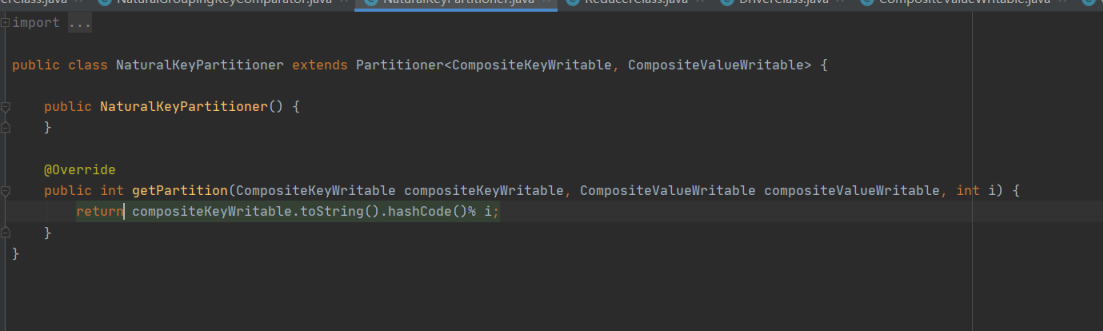


Reducer (the output is organized as .csv)

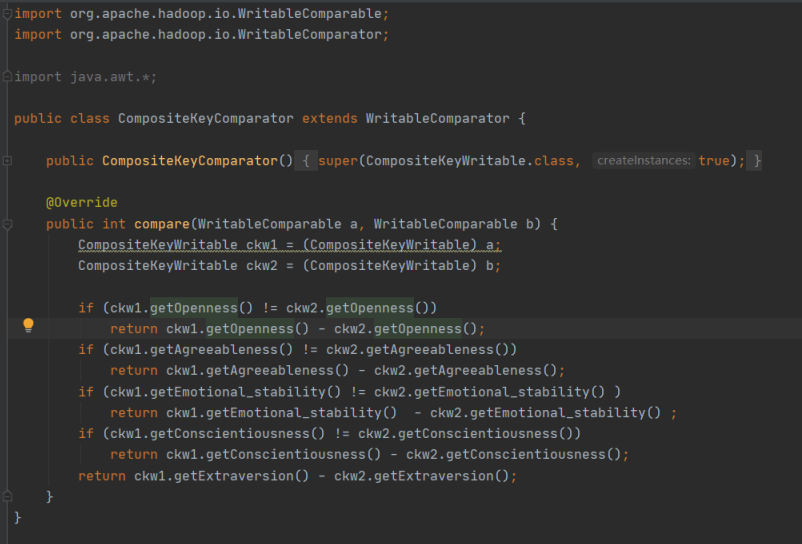


Change the rule of partition here if needed (by default is same)





Change the order of key here (by default is the same):

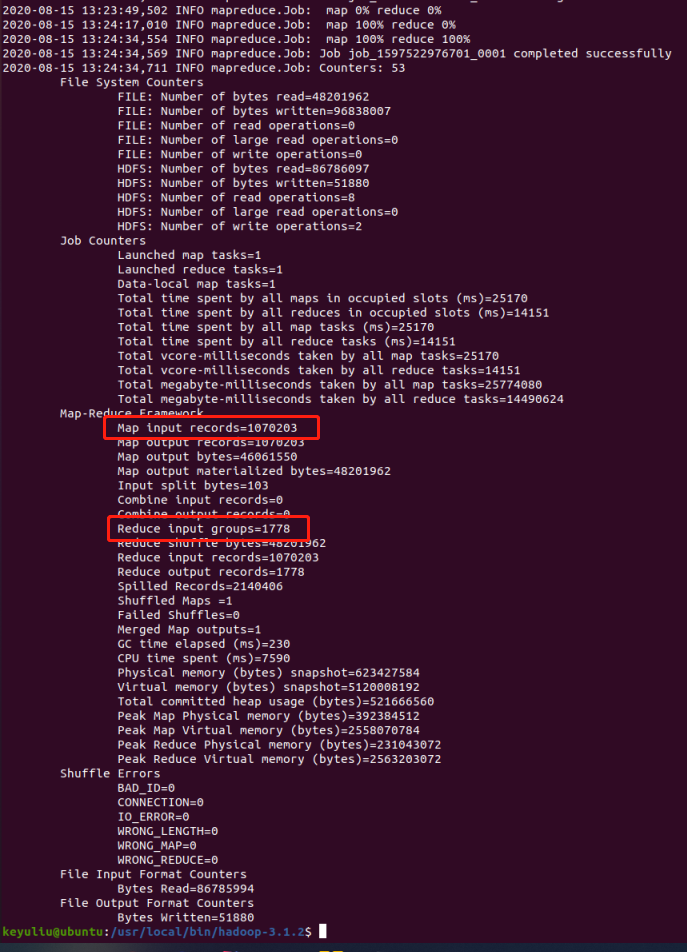


### 3.3 Implementation in HDFS

bin/hadoop fs -mkdir /EvaluationModel

bin/hadoop fs -copyFromLocal ~/Downloads/userinfo.csv /input

bin/hadoop jar ~/Downloads/SecondarySort-1.0-SNAPSHOT.jar DriverClass /input/userinfo.csv /result/output/Evaluation

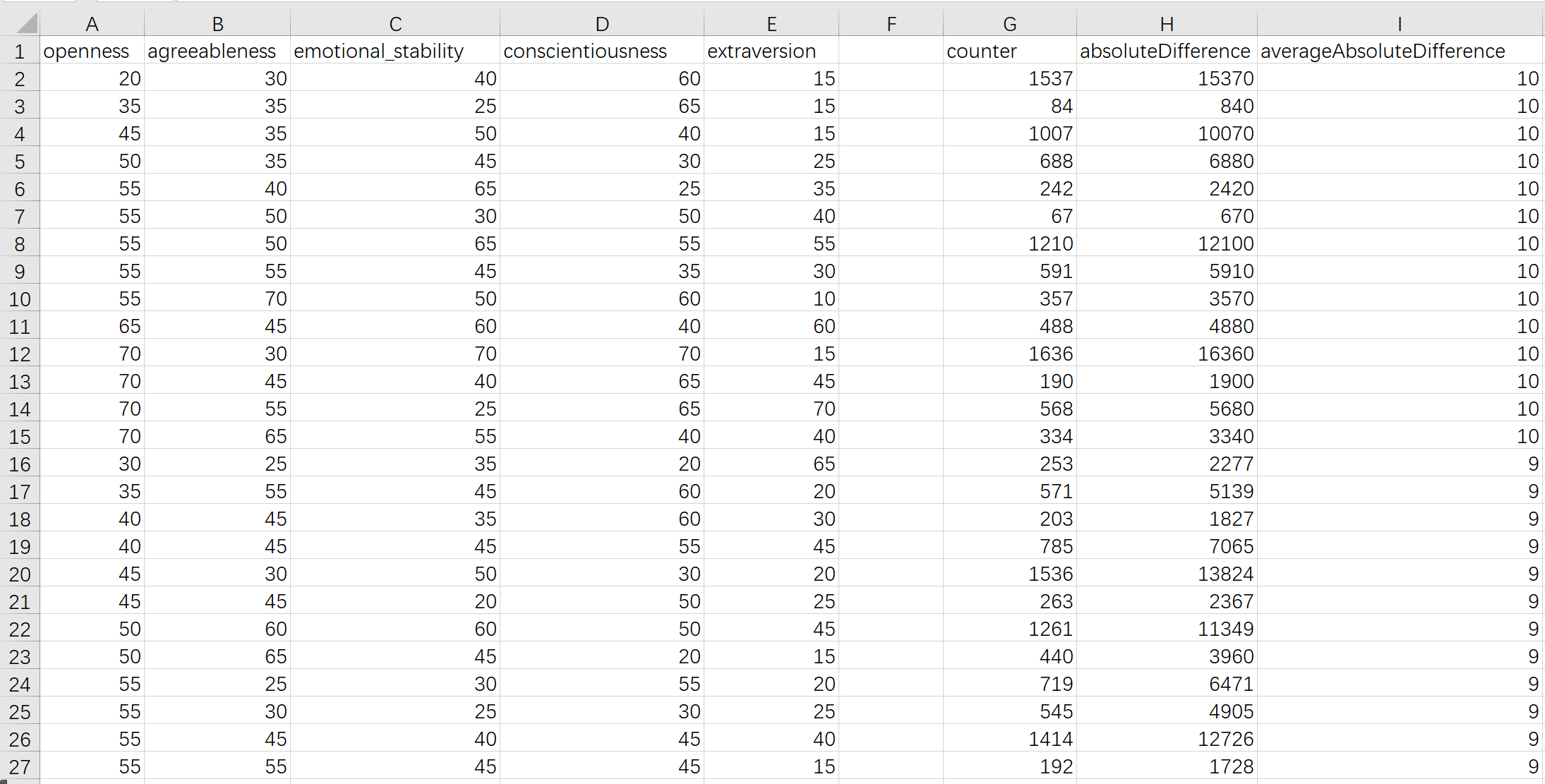


Tip: As you can see in screenshot, there are only 1778 splits after reducer. Why not 1821 – the same number of userid ?

It’s true that userid and 5 factor of personality are one-to-one relationsip, assigned condition actually has 4 value (default, high, low, medium). But default has no meaning when representing the prediction of system.

bin/hadoop fs -copyToLocal /result/output/Evaluation/part-r-00000 ~/Downloads/Project

### Output file part-r-00000 -> AAD.csv



Counter : number of userid who have this kind of personality

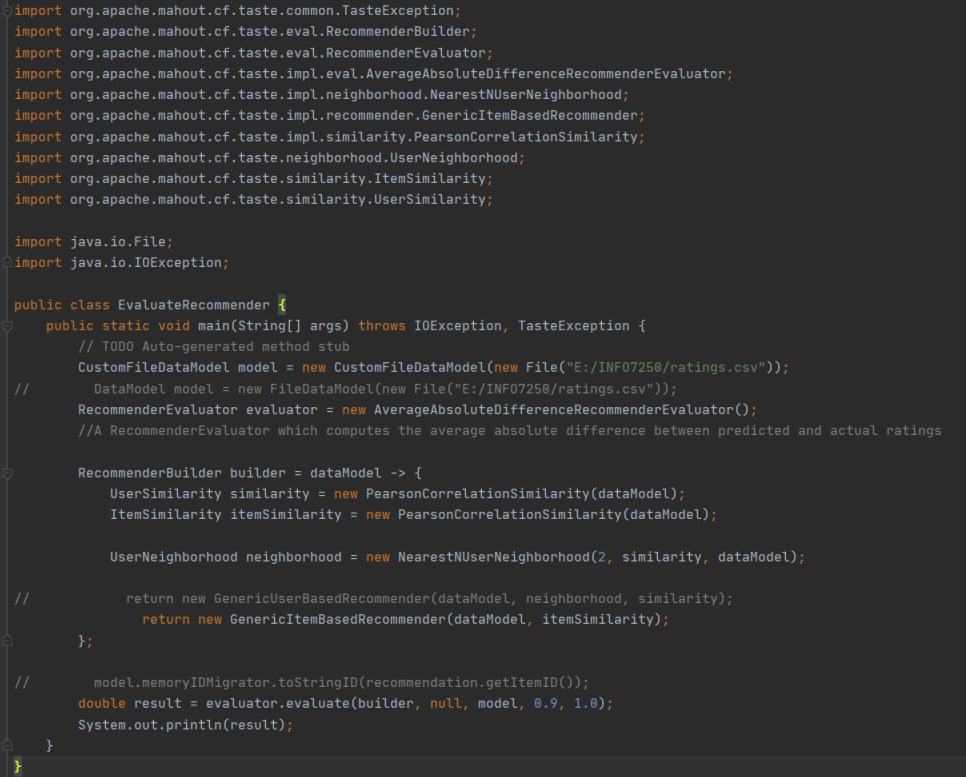
AbsoluteAverageDifference – **The difference between the prediction of the system and the actual value**

## Compare with generic recommender system by AverageAbsoluteDifference (Mahout)

### User/Item-based collaborative filtering Recommender Systems and AverageAbsoluteDifferenceRecommenderEvaluator

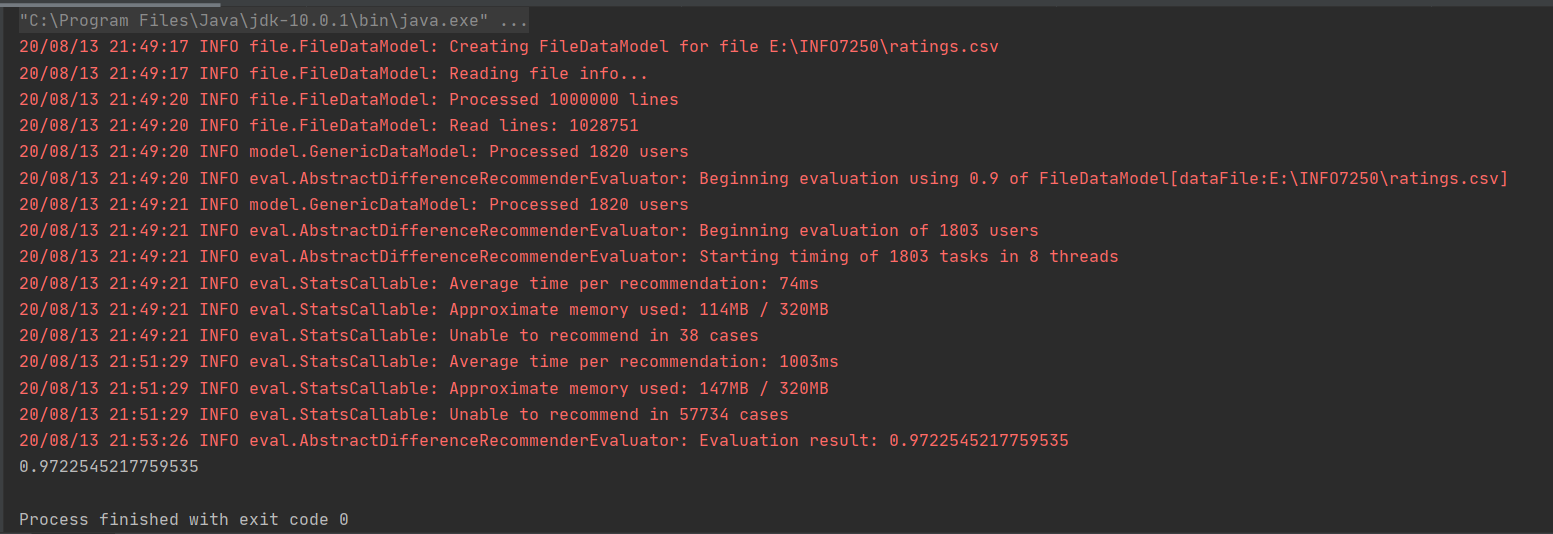
Implementation: as userid is hashcode of string not a long type. A custom data model is needed.

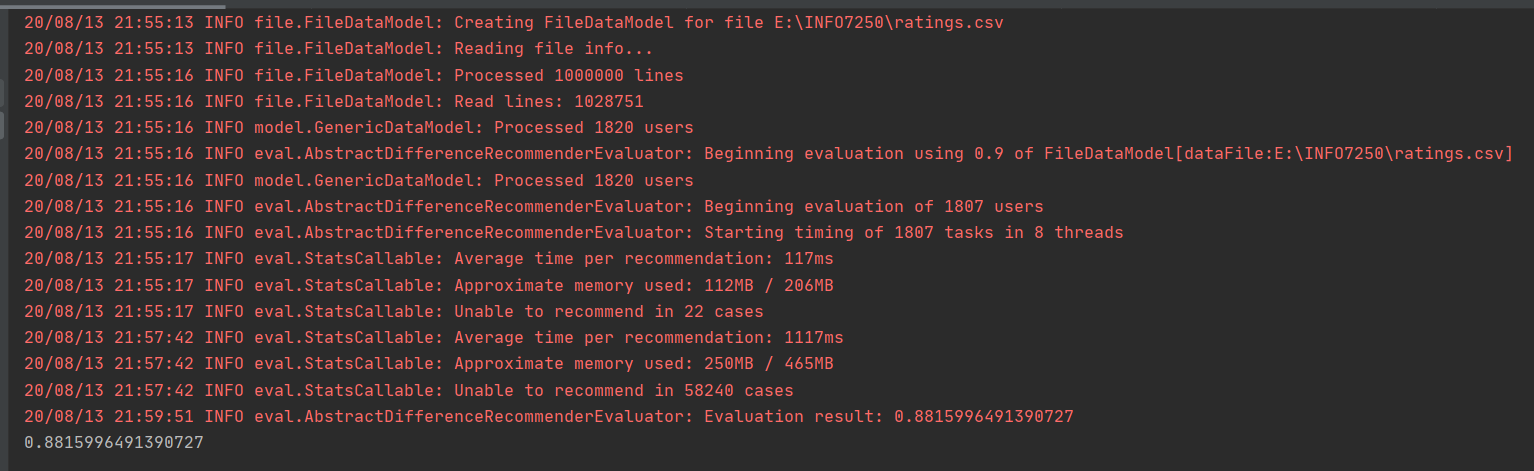




* 1. Output from User-based recommender system

Because the dataset are randomly separate as train and test. Repeat test is needed.



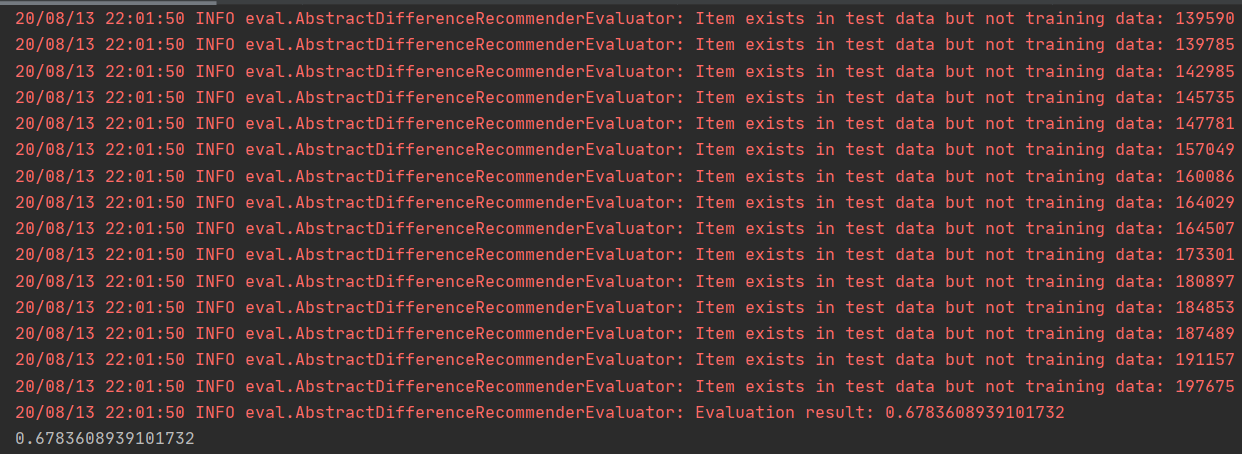
****

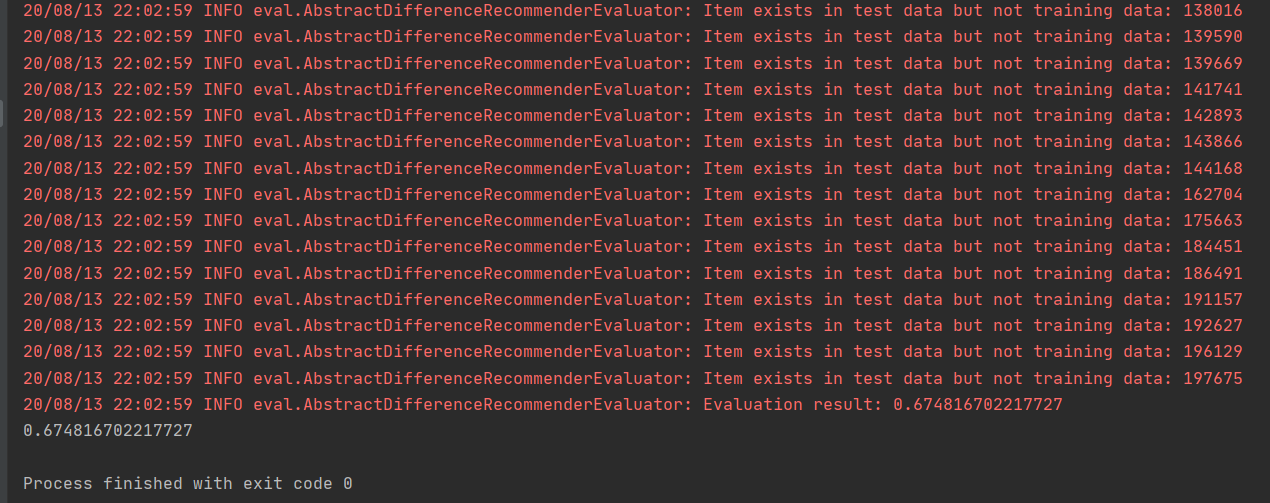
**After 20 times of test, the Average result of repeat test:**

**AverageAbsoluteDifference = 0.89973 which is higher than 0.082332.**

**We can see the model in article is far more accurate than generic model.**

* 1. Output from Item-based recommender system





It seems Absolute Average Difference is lower than user-based, does it mean item-based is more accurate ?

No. As I review the detail of rating.csv, I find most movies are not seen by different people. This means some movies are only seen by a single person. In generic item-based recommender of mahout, these “item” are used as “neighbors” in user-based recommender. And in this case, some neighbors

Only connected to one user. This is why the function runs not properly and the AverageAbsoluteDifference seems more stable and lower.

* 1. Connection with AAD.csv

We know that GenericUser-basedRecommender has AverageAbsoluteDifference = 0.8,

In AAD, we can find that the number of users(counter) with AverageAbsoluteDifference equals [0,1] = 314594.

And the total number of users is 1070203.

There are almost 1/3 users are located in predicted range. This also proved that the evaluation model 2.0 is reasonable.

## Future work

Using KNN-item-based recommender to test the AverageAbsoluteDifference.

Presentation using Prezi instead of PPT.

Sharing link : <https://prezi.com/view/KoUodk8HcfBLYS8Ron2i/>.

Or the pdf version in Project directory.