**Programming Assignment #4 Report**

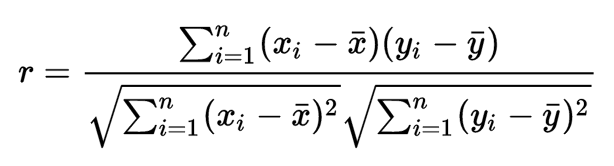
Computer Science and Engineering

2010003970

Namwoo Kim

1. **Summary of algorithm**

This program predicts item’s rating by using given base data sets. First, calculate similarity score between two users among all users by using Pearson Correlation Coefficient(PCC).

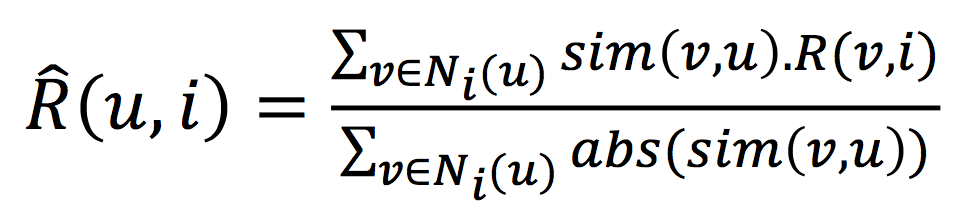


Xi refers rating of item i of user X

x-bar refers average rating of user x

* Items must be shared between two users (common items)

Second, predict score using formula below.

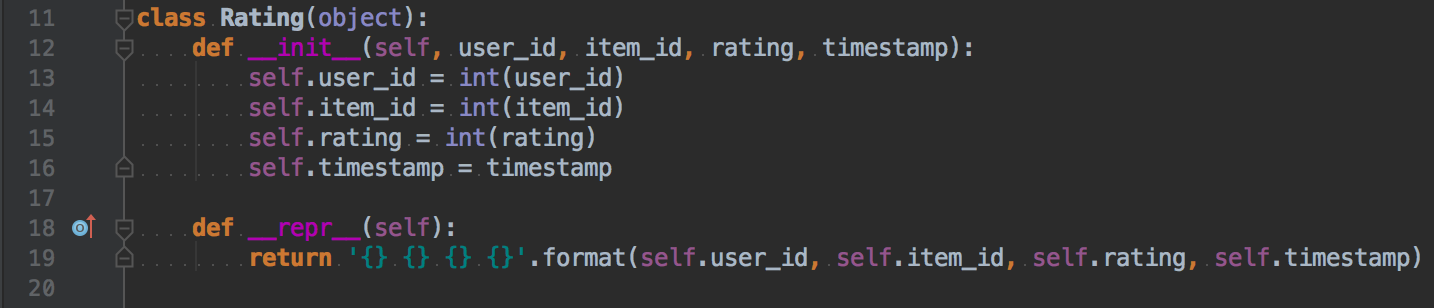


sim(v, u) implies that similarity score between user v and u.

R(v, i) implies that rating for item i of user v.

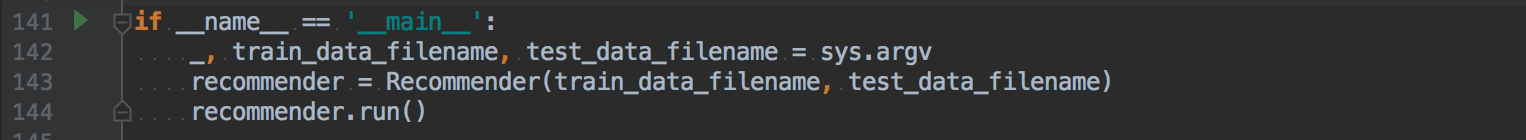
1. **Detailed description of my codes**

**<basic data structure Rating>**



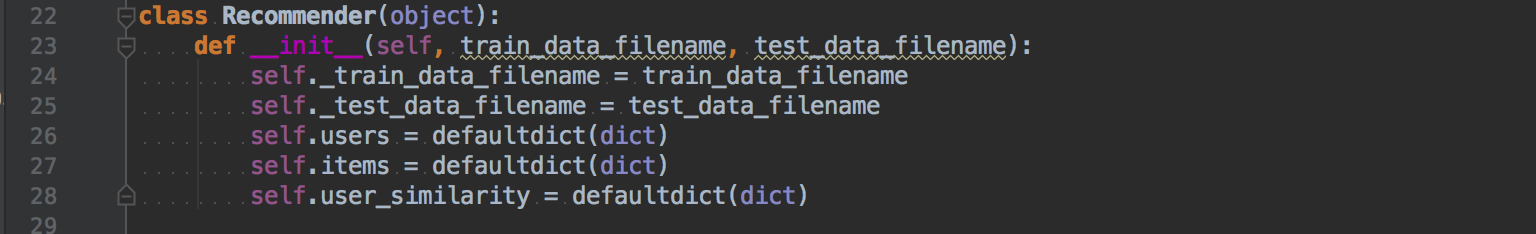
This class contains data from u#.base file. User\_id, item\_id, rating, timestamp.

**<main entry of this program>**



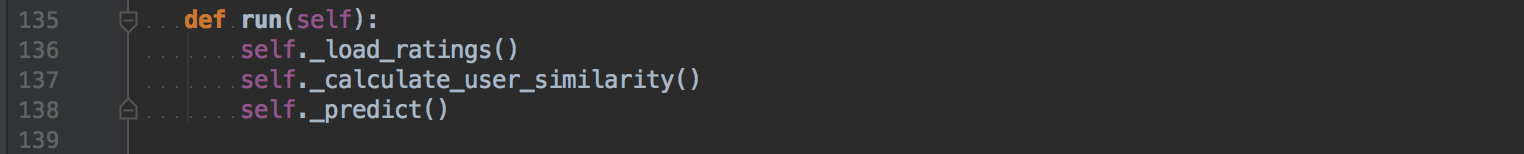
This part only initialize recommender class and then call run function.

**<Recommender constructor>**



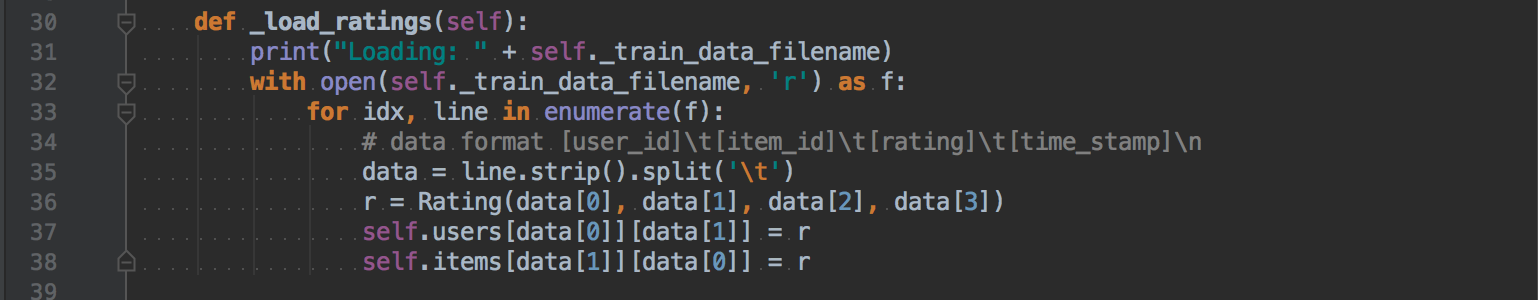
Constructor initialize default variable’s data type and store filenames(train\_data(u#.base), test\_data(u#.test))

**<Recommender function run()>**



This is the main part of running this recommender. This function running as follows: \_load\_ratings(), \_calculate\_user\_similarity() then \_predict()

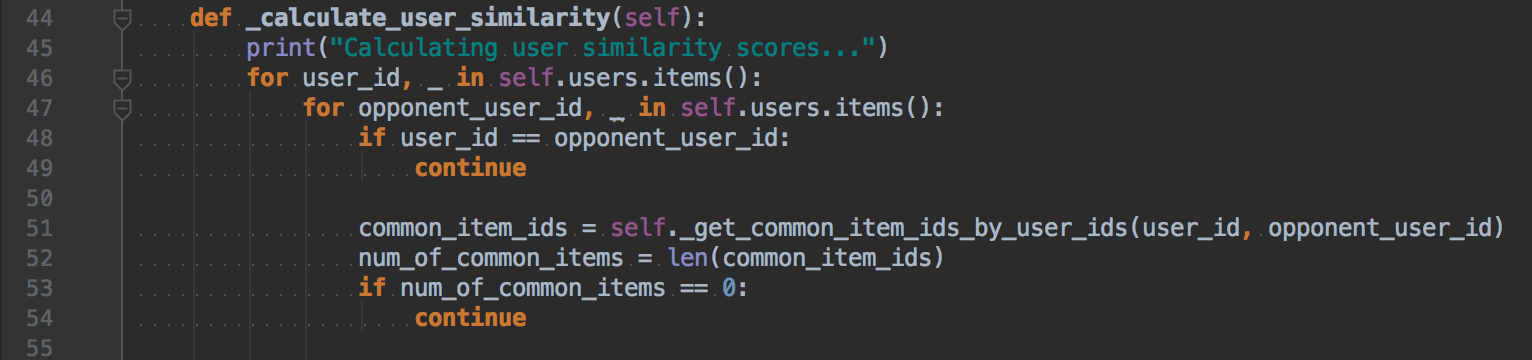
**<Recommender function \_load\_ratings()>**



This function read train\_data file which is same as ‘u#.base’ file and then insert Rating class instance into users and items dictionary(which is same as array). The reason why I add Rating class instance variable into two arrays is to retrieve user or items as fast as possible.

**<Recommender function \_calculate\_user\_similarity()>**

* Very first part

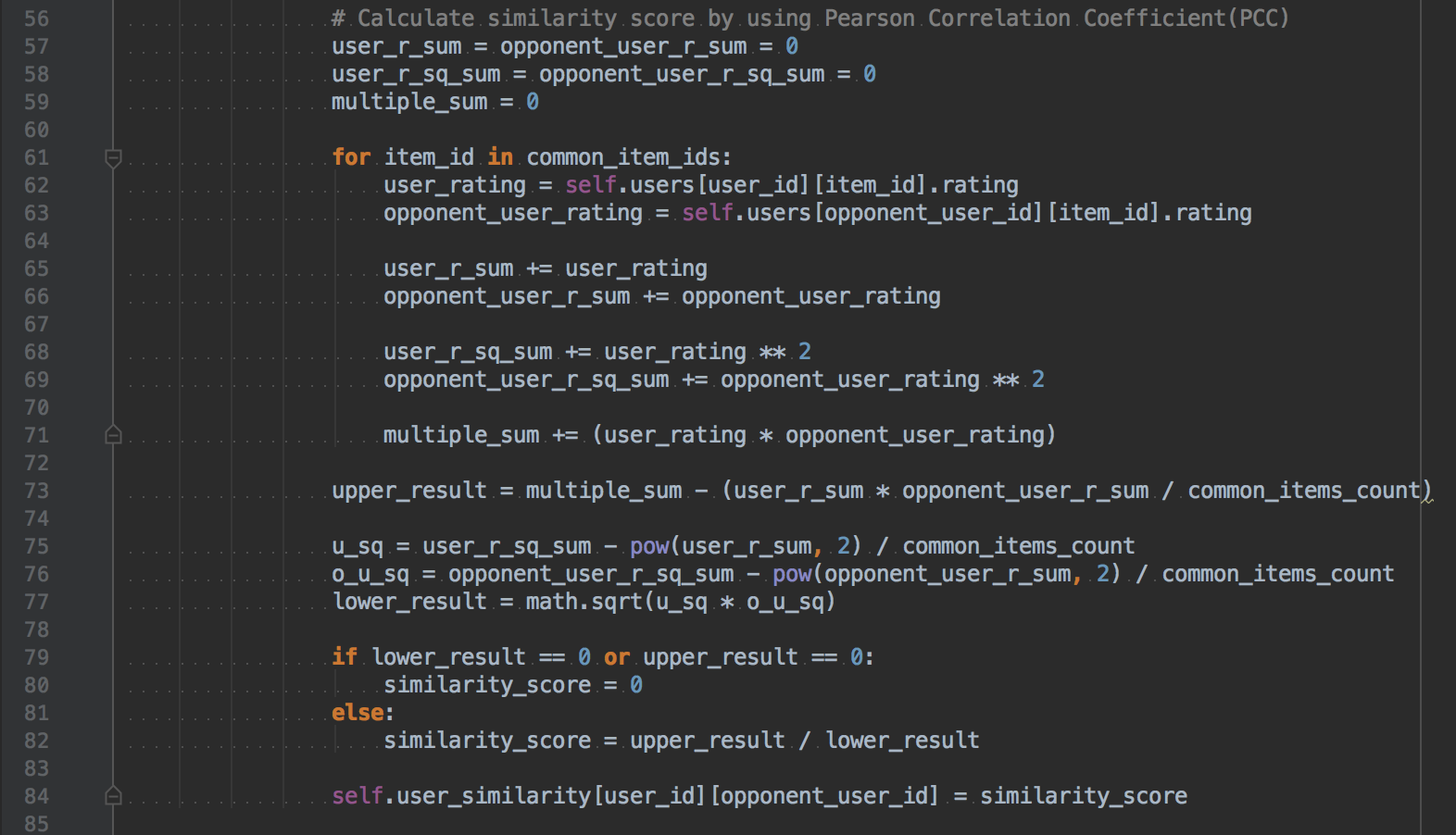


Line 46~47: Iterating all users by all users which is O(n^2).

Line 48~49: Ignore same user (which is calculating similarity between me and me)

Line 51~54: Prevent to calculate when there’s no common items between two uses.

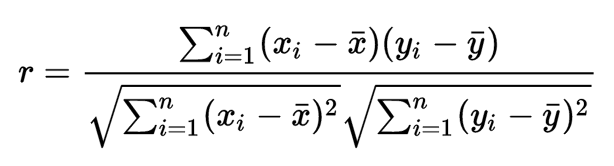
* Main part



Line 57~59: Initialize numeric variables

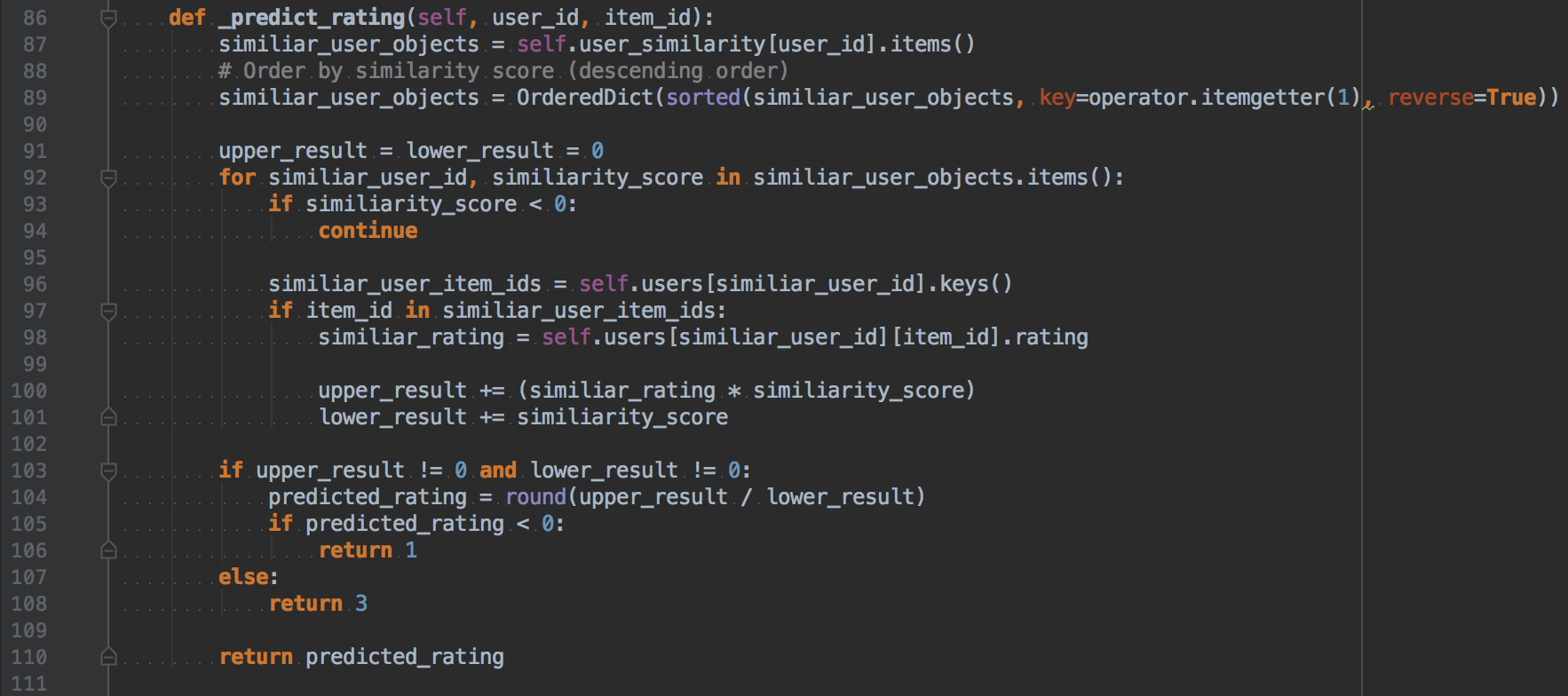
Line 65~69: Adding all rating scores for each user respectively

Line 73~77: This part implements Pearson Correlation Coefficient(PCC).



Line 79~84: Adding similarity score into dictionary

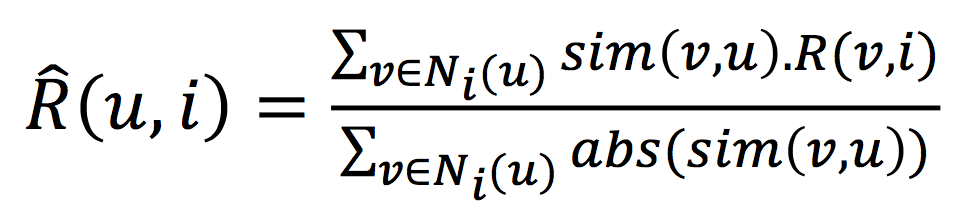
**<Recommender function \_predict\_rating()>**

****

Line 89: Ordering similar users by descending order of similarity score

Line 93~94: Ignoring user which has lower similarity score ( under 0 ) since PCC returns -1 to 1 range similarity score.

Line 97~101: Implement predict formula

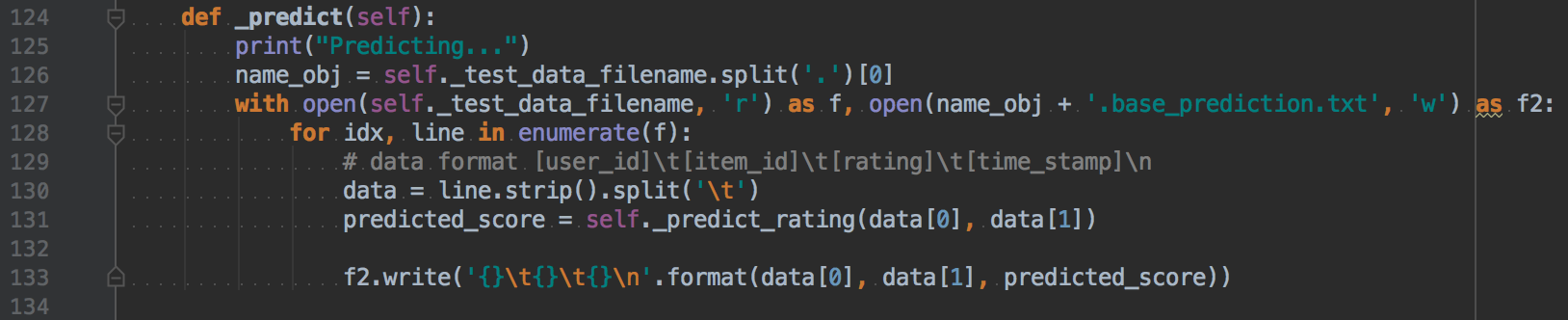


(But, I didn’t use abs for this situation. Assume that we’re using similarity score over zero. We don’t need to use abs for now.)

Line 105~106: Return 1 if predicted score is under zero. ( returning 1 to set value into range 1-5)

Line 108: Return 3 if similar user has no rating for given item and there’s nobody ratings item (Rare case). Just return mid value for range 1-5.

**<Recommender function \_predict()>**

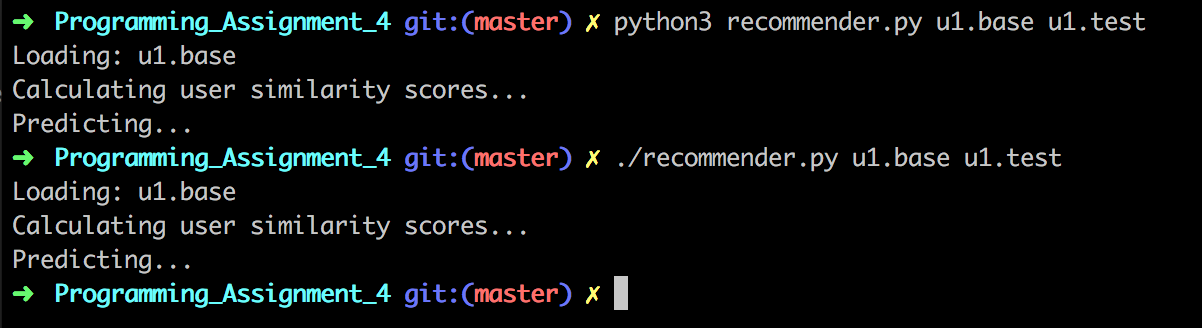


This function simply writes a new file with predicted scores.

1. **Instructions for compiling my source code at TA’s computer**

This program works with **Python 3.6 only**. You can easily run this program in any other environments which has python3 because python program doesn’t need to compile for run.

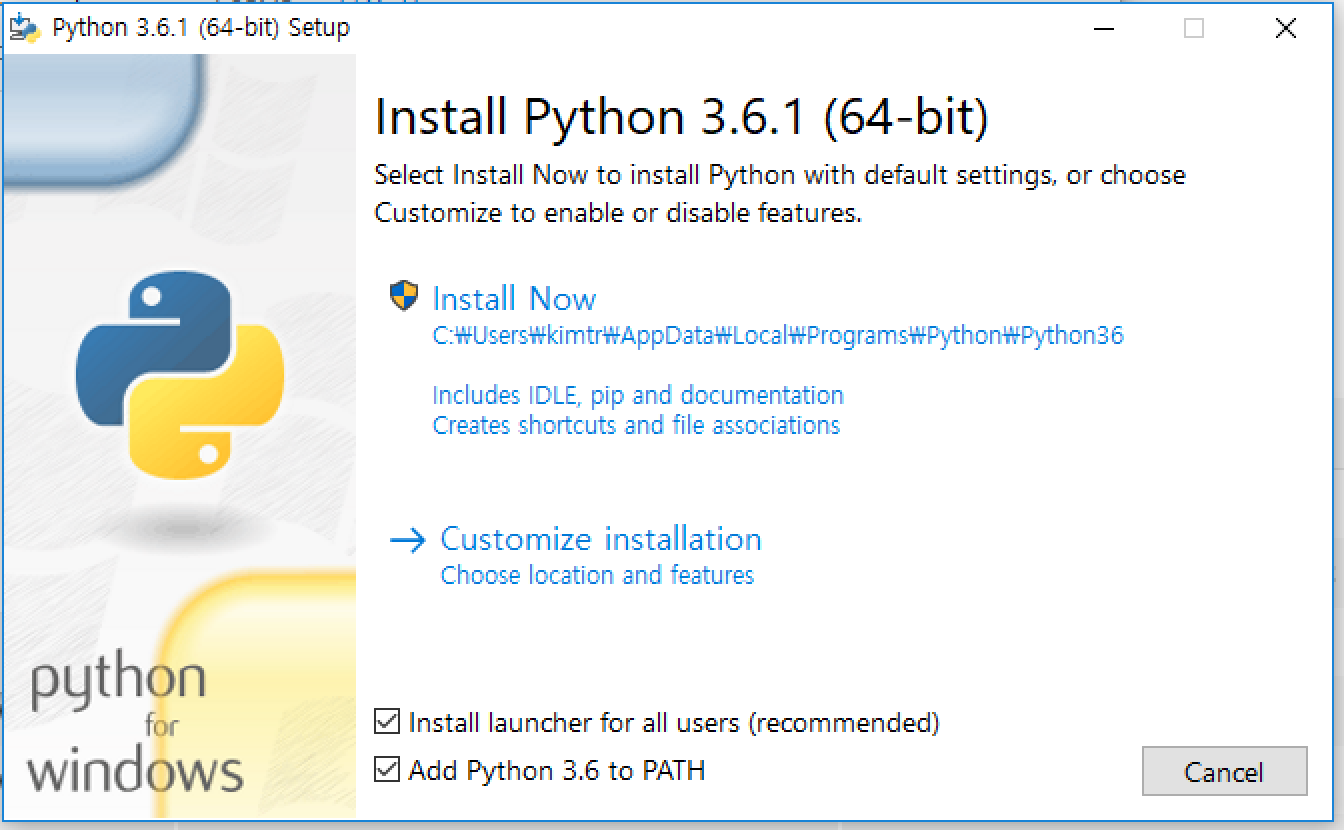
In mac OS, there is a preinstalled Python2 but you have to install Python3.



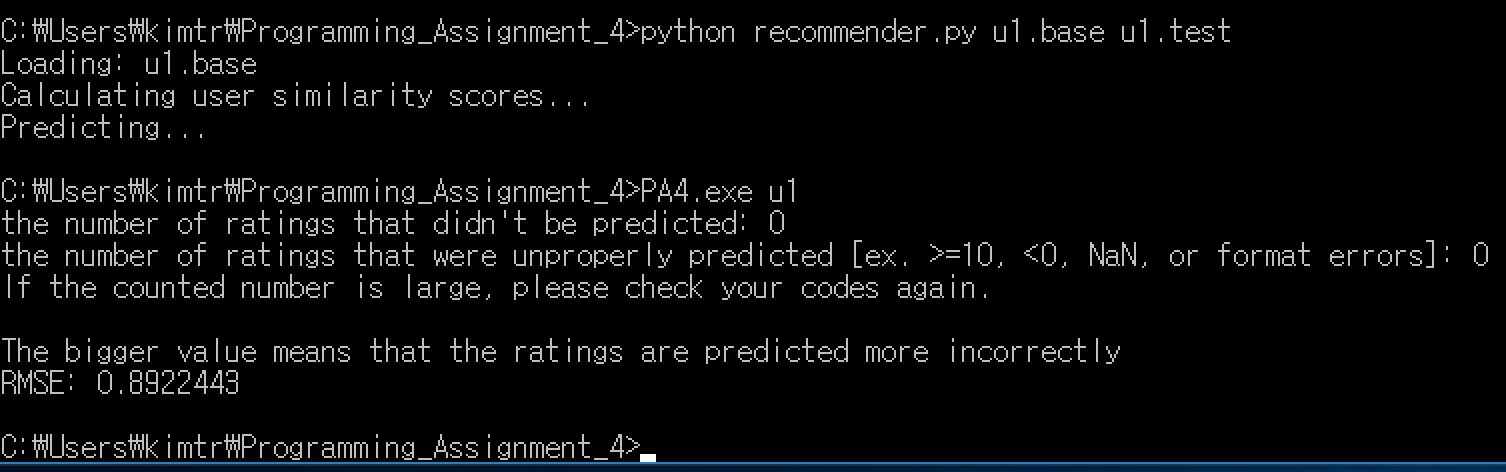
OK for both ways.

In windows, install python 3.6 first.

**Note that you must check “Add Python 3.6 to PATH”**



Then, Just type “python recommender.py u1.base u1.test” from command line utility.



1. **Any other specification of my implementation and testing**

I’ve tested both mac OS and Windows environments and successfully tested with the program that TA given to me. I think there’s nothing to mention for any specifications.