**Long term project**

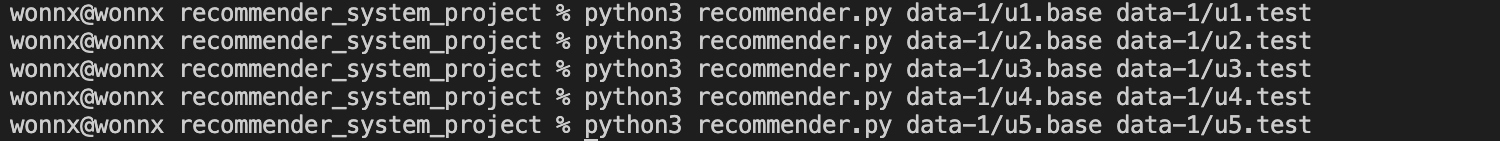
컴퓨터소프트웨어학과 2017029470 김종원

1. **Environment**

* Mac OS (Monterey) M1 chip
* Python 3.10.3 (release March 16, 2022)

1. **How to compile**

Before compiling dt.py, python version 3 must be installed in your system.



* Execution file name: recommender.py
* Training data file name: u#.test.txt
* Test data file name: u#.test.txt
* Output file name: u#.base\_prediction.txt

1. **Summary of algorithm**

I used the collaborate filtering method to predict the ratings of movies in test data. Collaborative filtering refers to a technology that predicts itself based on taste information collected from many users. Based on the user-user similarity neighbors that most similar to the corresponding user were extracted, and ratings were predicted based on the rates of the extracted neighbors.

* Using the pandas library, the ratings of movies left by users are transformed into a rating matrix.
* Based on the rating matrix, the Pearson correlation coefficient between user-user is obtained and converted into a similarity matrix.
* After sorting the similarity matrix, the neighbors corresponding to the top 5% of all users are obtained, and then the rating is predicted through a weighted sum.

1. **Detailed description of codes**
2. **get\_rating\_matrix**

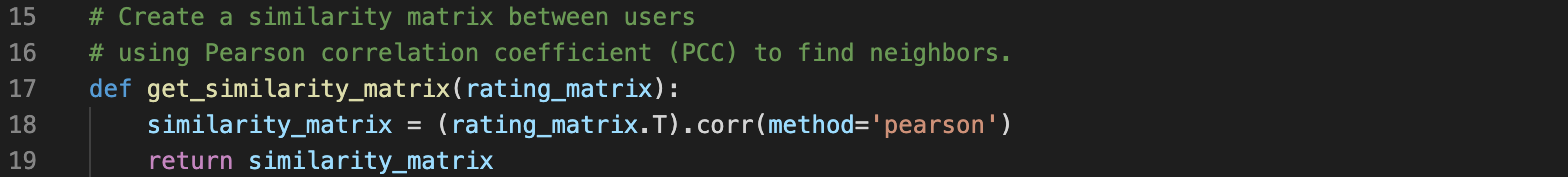
텍스트이(가) 표시된 사진

자동 생성된 설명

This is a function that transforms the given training data into a rating matrix. In the training data, the column corresponding to the time stamp was not used.

* Convert training data into dataFrame format of pandas library.
* Convert the dataFrame named **matrix** into the pivot\_table type **rating\_matrix**.

1. **get\_similarity\_matrix**



Based on the previously obtained rating matrix, it is a function that calculates the similarity between users and users though the Pearson correlation coefficient and stored as a similarity matrix.

* Using the built-in correlation method of pandas dataFrame, calculate similarity between users.

1. **predict**

텍스트이(가) 표시된 사진

자동 생성된 설명

Average ratings of movies evaluated by each user and neighbors of each user are obtained. Using the average, neighbor, and rating matrix, we predict the user’s rating for a movie that has not yet been rated. In addition, the output file name is set differently according to the input file, and the predicted rating is exported according to the format.

* **Average** and **neighbors** are in the form of dictionary, and each key contains average rating of the movies for the user and PCC for all users.

1. **estimate**

텍스트이(가) 표시된 사진

자동 생성된 설명

5% of all users are set as neighbors, and if the neighbor has previously rated the movie, it is multiplied by similarity and weighted summed.

* If the movie does not exist in the existing rating matrix, the rating is predicted using the average rating value of the user.
* If the prediction result is less than 1, it is corrected to 1, and if it is greater than 5, corrected to 5.

1. **Testing result**

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

For testing, I used a measure called RMSE (Root Mean Square Error). The bigger value means that the rating are predicted more incorrectly. All five testing results came out close to 1, indicating that the user actually evaluated the movie and the predicted value differed by about 1.