Course name: Data Science (ITE4005)

Professor: Sang-Wook Kim (email: wook@hanyang.ac.kr)

TAs: Dong-hyuk Seo (email: hyuk125@agape.hanyang.ac.kr)
Jiwon Son (email: tinybeing@agape.hanyang.ac.kr)

< Long term project>

10 May 2022

Due Date: 18 June 2022, 11:59 pm

1. Environment

- OS: Windows, Mac OS, or Linux
- Languages: C++, Java, or Python (any version is ok)

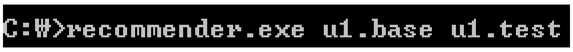
2. Goal: Predict *the ratings of movies in test data* by using the given training data containing movie ratings of users. You can choose any algorithm to predict (ex. content-based and collaborative filtering algorithms). For a content-based algorithm, you can refer to the web page to get the content related to our training and test data (http://grouplens.org/datasets/movielens/).

(Note) This assignment is to predict ratings for each user-item pair only within test data.

3. Requirements

The program must meet the following requirements:

- Execution file name: **recommender.exe**
- Execute the program with two arguments: training data name, test data name
  - Example:



```
C:₩>recommender.exe u1.base u1.test
```

  - training data name = ‘u1.base’, test data name = ‘u1.test’

- File format for a training data

[*user_id*]\t[*item_id*]\t[*rating*]\t[*time_stamp*]\n

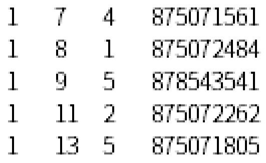
[*user_id*]\t[*item_id*]\t[*rating*]\t[*time_stamp*]\n

[*user_id*]\t[*item_id*]\t[*rating*]\t[*time_stamp*]\n

[*user_id*]\t[*item_id*]\t[*rating*]\t[*time_stamp*]\n

…

  - Row: a record that was already rated by a user for an item
  - Example:



```
1	7	4	875071561
1	8	1	875072484
1	9	5	878543541
1	11	2	875072262
1	13	5	875071805
```

Figure 1. An example of a training data.

  - ■ Five training data will be provided: ‘u1.base’, ‘u2.base’, ‘u3.base’, ‘u4.base’, and ‘u5.base’

- File format for a test data

  [*user_id*]\t[*item_id*]\t[*rating*]\t[*time_stamp*]\n

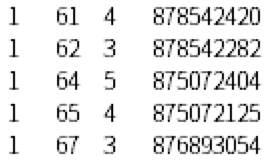
  [*user_id*]\t[*item_id*]\t[*rating*]\t[*time_stamp*]\n

  [*user_id*]\t[*item_id*]\t[*rating*]\t[*time_stamp*]\n

  [*user_id*]\t[*item_id*]\t[*rating*]\t[*time_stamp*]\n

  …

  - ■ Row: a record that needs to be predicted by using your algorithm
  - ■ Example:



```
1	61	4	878542420
1	62	3	878542282
1	64	5	875072404
1	65	4	875072125
1	67	3	876893054
```

Figure 2. An example of a test data.

  - ■ Five test data will be provided: ‘u1.test’, ‘u2.test’, ‘u3.test’, ‘u4.test’, and ‘u5.test’

- Output file format
  - ■ You must print an output file for each test data
  - ■ File format for the output of ‘u#.test’
    - ‘u#.base_prediction.txt’

      [*user_id*]\t[*item_id*]\t[*rating*]\n

      [*user_id*]\t[*item_id*]\t[*rating*]\n

      [*user_id*]\t[*item_id*]\t[*rating*]\n

      [*user_id*]\t[*item_id*]\t[*rating*]\n

      …
  - ■ ‘u#.base_prediction.txt’ should contain all user-item pairs in the test data and ratings that were predicted for the pairs by using your algorithm
  - ■ Supposed to follow the naming scheme for the output file as above

4. Evaluation measure

- Compute the difference between each predicted data (u1~u5.base_prediction.txt) and each test data (u1~u5.test)

- Test method
    - ■ For testing, we will use a measure called RMSE (Root Mean Square Error) defined as follows

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (p_i - a_i)^2}{n}}$$

(p_i: predicted rating for item i, a_i: original rating for item i, $\boldsymbol{n}$: the number of ratings)

    - ■ Because RMSE means error rates, the bigger value means that the ratings are predicted more incorrectly

5. Note

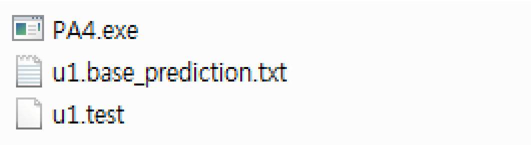
- This is a *competition project*
- If the accuracy of your model is higher, you will get a higher score
    - ■ We will first give a minimum score at least 75 if (1) you submit your program before the deadline, (2) your program is correctly performed without any errors, and (3) all requirements for this project are satisfied.
    - ■ Then, we will assign the additional scores from 0 to 25 based on your rank.

6. Submission

- Please submit the program files and the report to *GitLab*
    - ■ Report
        - File format must be *.pdf
        - Guideline
            - ✔ Summary of your algorithm
            - ✔ Detailed description of your codes (for each function)
            - ✔ Instructions for compiling your source codes at TA's computer (e.g. screenshot) (*Important!!*)
                - ■ If TAs read your instructions but cannot compile your program, you will get a penalty. Please write the instructions carefully.
            - ✔ Any other specification of your implementation and testing
    - ■ Program and code
        - An executable file (.exe or .py)
            - ✔ If you are not in the following two cases, please submit alternative files (e.g., .py file, .jar file, makefile)
            - ✔ You MUST SUBMIT instructions for compiling your source codes. If TAs read your instructions but cannot compile your program, you will get a penalty. Please, write the instructions carefully.
        - All source files

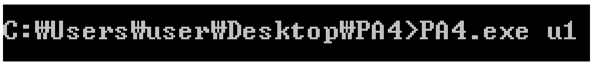
6. Testing program

- Please put the following files in a same directory: Testing program (PA4.exe), your output files (u1.base_prediction.txt), given test file (u1.test)



PA4.exe
u1.base_prediction.txt
u1.test

- Execute the testing program with one argument (input file name)



```
C:₩Users₩user₩Desktop₩PA4>PA4.exe u1
```

- Check your RMSE for each input file

7. Penalty

- This assignment does *not allow late submission*!!
- Significant penalty up to 30% will be given when the requirements are not satisfied