

**Chinese University of Hong Kong**

**Department of Computer Science**

**MSc in Computer Science**

CMSC5741 Big Data Tech. &Apps

**Final Report**

Topic: A Simple Movie Recommendation System

|  |  |  |  |
| --- | --- | --- | --- |
| **Professor:** | Michael R. Lyu | | |
| **Prepared by:** | Team 2  1155101156  1155105483  1155101677  1155105238 | Zhang Tuxin  YAN, YiXin  Liu ZhuoXi  Huang Anheng | [1155101156@link.cuhk.edu.hk](mailto:1155101156@link.cuhk.edu.hk)  [1155105483@link.cuhk.edu.hk](mailto:1155105483@link.cuhk.edu.hk)  [1155101677@link.cuhk.edu.hk](mailto:1155101677@link.cuhk.edu.hk)  [1155105238@link.cuhk.edu.hk](mailto:1155105238@link.cuhk.edu.hk) |
| **Submit Date:** | Nov. 1st, 2017 | | |

**Deliverables**:

*Introduction:*

Two different methods are used by our project so far. SVD and User-Based collaborative filtering. Both produced reasonable and acceptable results. After computing and training, we have two independent models that can be used to predict, or namely, just give movie recommendations by entering the information about the user we want to give suggestions.

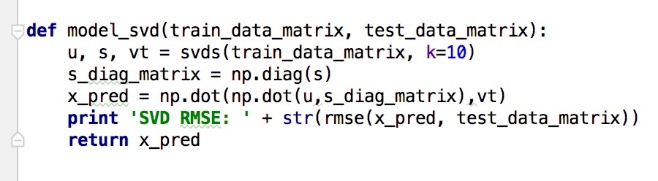
*Time consumed:*

Since the way we implemented Singular value decomposition and collaborative filtering are completely different. The time required to train the two models are also not the same. Clearly User-Based collaborative filtering consume far more less time than SVD. There are few reasons, firstly, we need to form a matrix for SVD while CF does not. Secondly, matrix computation is more expensive than array list computation. Use 1Million test dataset, the time required by SVD is roughly 3-4 minutes. On the other hand, line-based computation of CF only takes less than a minute using the same dataset.

**SVD:**

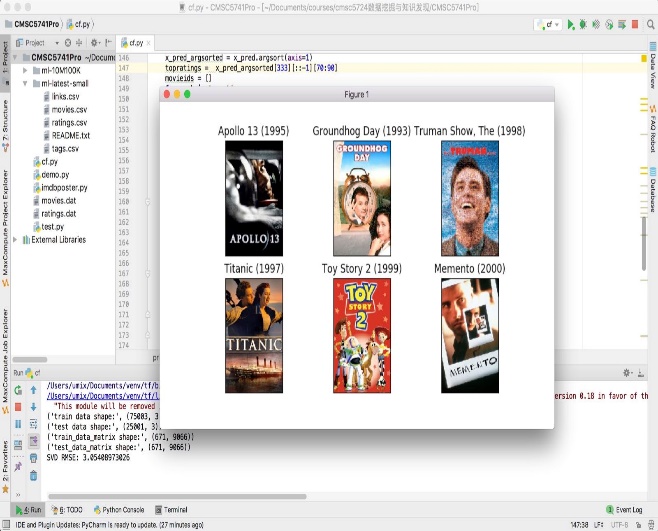
The Idea is again, form a rating matrix in the first hand, then manipulate the matrix to obtain the result we want. In this case, we want user-features matrix and movie

The partial code is shown below:



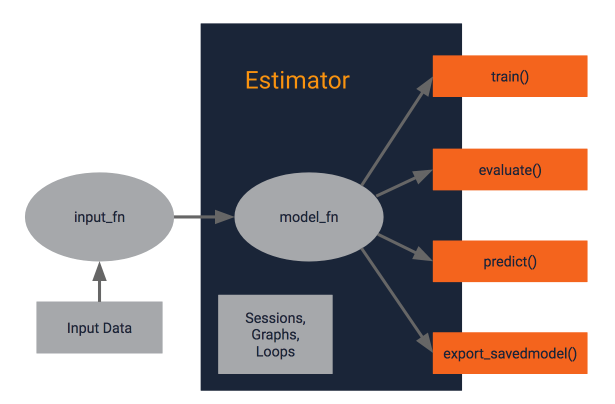
Setting K = 10 indicates that our desired recommendations are set to ten. After retrieving all the matrixes. We can obtain the estimate error by using root mean square error RMSE.

*User friendly result:*



As we can observe from the command window, the root mean square error is 3.054 which is considerably acceptable. By entering the user ID we want to predict

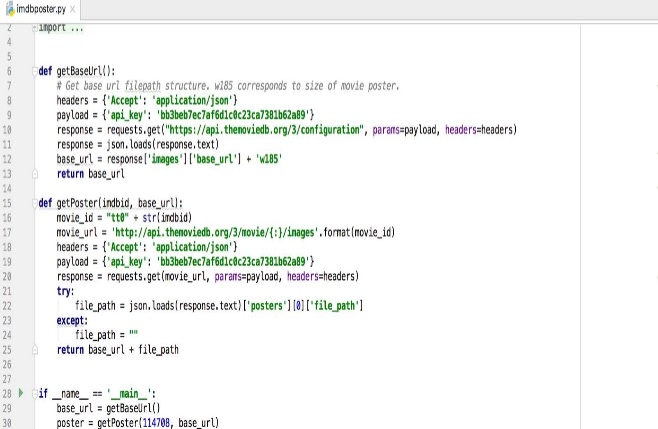
The result demonstrated a simple input-model-prediction model. As shown as the illustrations below.



*Input:* User ID

*Output:*

predicted rating list of that user, then base on the K value, we filter out those low predict rating movies, keep top 10 highest ratings movie. To make it more user friendly. We use IMDB-api to pull those movie posters from website.



This is the official recommendation code for pulling posters from the API.

*Conclusion for SVD:*

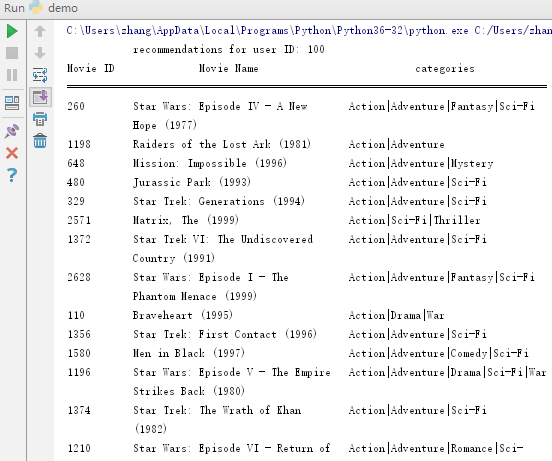
The principle is to calculate the corresponding matrixes. Then predict the ratings of that user by some matrix manipulations. Display result by sorting the ratings.

**User-Based collaborative filtering:**

*introduction:*

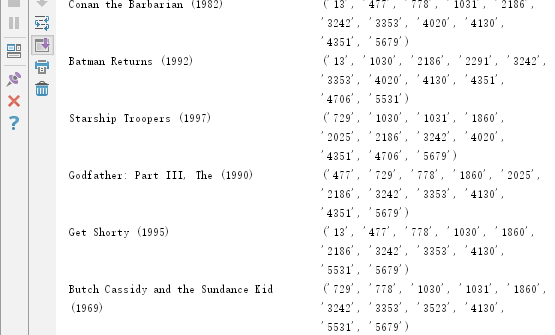
Briefly, this method required the user enter their user ID before doing any computation, this could be a point that we could make it better. But due to time constraining, the user id is required, and we hard coded it into the program before doing any computation.

*Result overview:*



After analyzing, we can surely conclude that we made some reasonable recommendations. For instance, the user with ID 100 watched and rated start wars 1-4 but not 5, and our program recommend him/her star war 5 which is completely reasonable. Since he/she is also an action, sci-fic movie lover, the program did not show any movie of other categories.

*Some analyzes:*



The result output by our CF program is separate into two parts. This first part is clearly the recommendation list. However, the second part is kind of for developer to implement more interesting features and gain some insights about what’s going on about the program. The image shown above demonstrate the reason why the program recommends those movies to the user.

**Conclusion:**

We have tried to use the same user-rating vector to test both programs. Surprisingly, they perform almost identical. Namely, they all recommend the similar movies to the user. Even though they are based on two completely different techniques. In our original plan, we were going to compare our result to others. But it seems that nobody has posted their results online, we were not able to obtain their results. But we still success to manage to compute the RMSE of SVD which is also kind of measurement of our work. Overall, our program still obviously can be implemented more beautifully. But the user interface feature combines with the recommendation we manage to obtained. Together they form a considerable working system.

==================================================两栏结束，一栏开始

**References:**

1. https://www.google.com/search?biw=1920&bih=949&tbm=isch&sa=1&ei=6f8nWtj-E4LO0ASr8JXAAw&q=input+model+predict&oq=input+model+predict&gs\_l=psy-ab.3...11418.12068.0.12233.6.6.0.0.0.0.113.405.5j1.6.0....0...1c.1.64.psy-ab..0.0.0....0.zmWBUlRTFP4#imgrc=4eOiABsmerkjfM:
2. http://blog.csdn.net/qq\_20282263/article/details/52692318