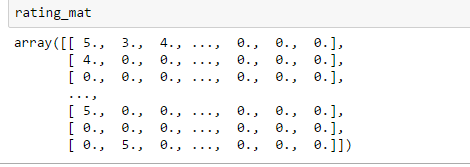
1. Introduction

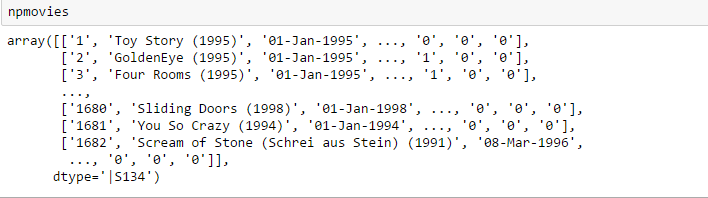
This is an Item-based collaborate filter system application, which has 3 main functions: “***rec\_most\_similar\_movies***” --- Giving a movie, recommend the top n similar movies, ” ***PredictRateForUser***”-----Given an unrated movie, predict the rating for a specific user, and “***PredictRateForUserMovie***”-------Given an user, find all his/her unrated movies and predict the rating for him/her.

2. Parameters:

This application will take generally 3 parameters, which are movie rating table(***rating\_mat***), movie database(***npmovies***), and similarity function(**c*osSim2,pearsSim, ecludSim***) to use. The movie rating table is a table contains lists of ratings for each user and is got from “u.data “, which contains user id and the user’s rating for each movie.

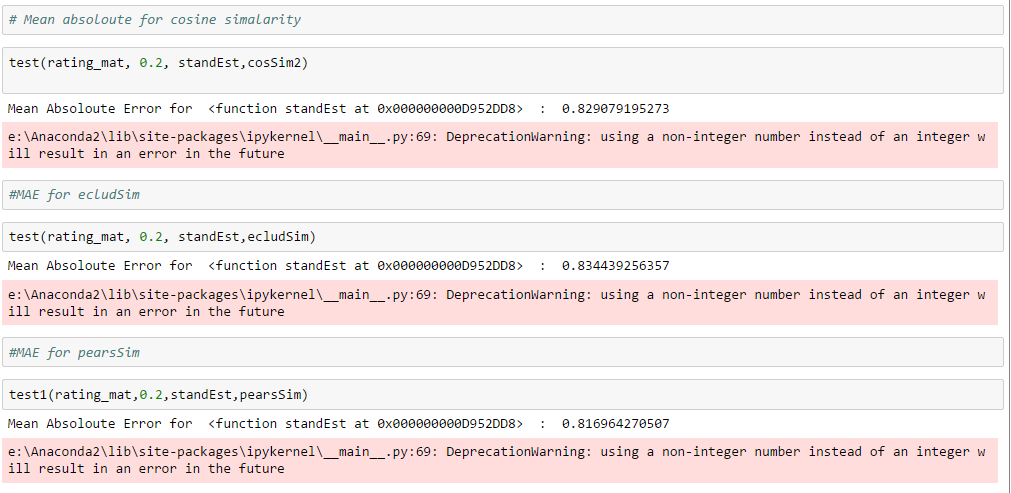
Movie rating table:

Movie database:



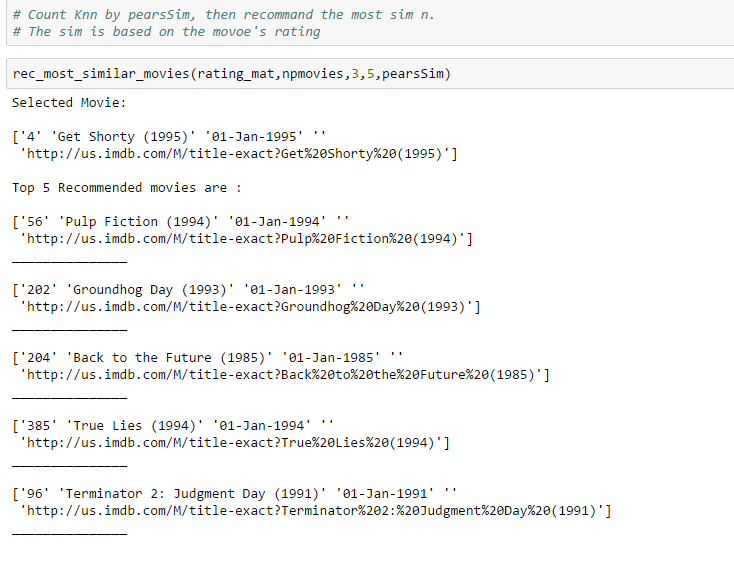
3. Function Details

To build this application, we first import “itemBasedRec.py”, and then built a test function to test the mean absolute error for each similarity mothed, the results are:



From the test result we found that although pears method seems slight better (0.817 vs 0.834 and 0.829), there is not a significant difference using anyone of those three similarity method. Then we build the function ***rec\_most\_similar\_movies,*** which will recommend top n similar movies for that user. This function takes 5 parameters: the movie rating table, movie database, the movie used as simple ,number of movie user wants to be recommend, and the similarity method. The basic idea is KNN search. We find the top n similar movies, and recommend them to that user.

The result of the function is like:

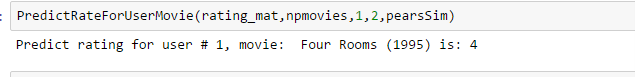


The ***PredictRateForUser*** function will automatically predict all unrated movie for a user, and the prediction is based by that user’s rating on most similar movies. This function takes 4 parameters:

the movie rating table, movie database, user id , and the similarity method. And the output of this function is :



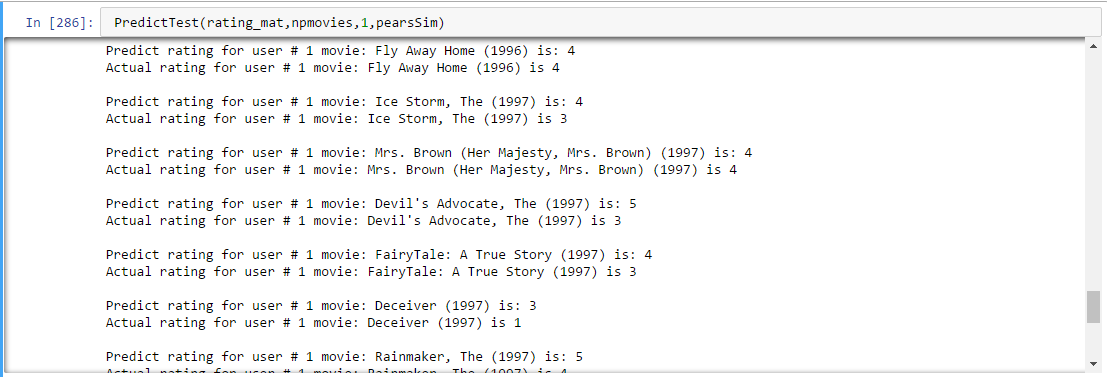
Function ***PredictRateForUserMovie*** is pretty much like ***PredictRateForUser,*** but specified which movie the user want to predict:



4. Test

In the test part we built 2 test functions, which are ***PredictTest*** and ***PredictTest2***. The idea is using the same way to predict the rated movie for a specific user, and comparing the predict result with actual rating. And for accuracy, we used the distance between the predict rating and actually rating.

Example for PredictTest:



Next, we take 20% or the total user as test target (189 users), and use ***PredictTest2*** to test the accuracy with different similarity method, the result is:



This result is consist with the MAE test result. We can see that all three similarity method gives the accuracy around 0.79, and pears similarity gives the highest accuracy, with is 0.7964 and just slightly higher than other 2 methods.

The overall test results are around 78% and are acceptable.