CSC478 Final Project Report

Dachi Xu

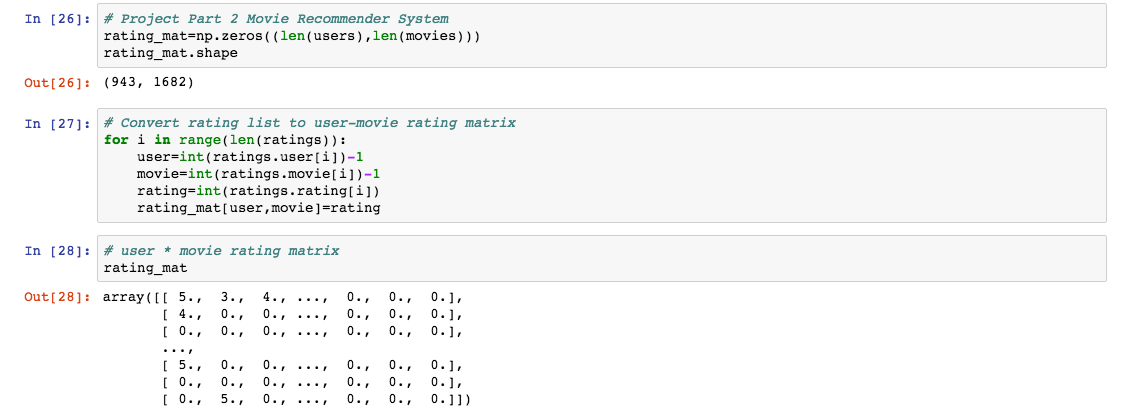
Project Application Part 1 Report:

The structure of the project is organized as three parts, the first part is data analysis, the second part is application 1 (recommendation system using collaborative filtering with user-based), the third part is application 2(recommendation system using collaborative filtering with item-based). Below paragraph are mainly discussing the application 1.

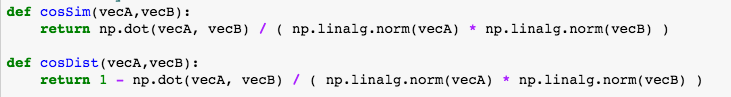
The purpose of application 1 is to apply k-nearest-neighbor algorithm on the user rating data and predict users rating for the selected movies. The overall workflow of this application is organized as follows:

1. Prepare the user rating data
2. Convert the original user rating data to rating matrix
3. Select user and movie for the rating prediction
4. Find the k nearest neighbors of the selected users based on the distances(cosine) of their rating vectors
5. Calculate the rating by averaging the rating for the selected movies from top k neighbors
6. Compare the prediction rating with the actual rating (if the user had rated it already)

As mentioned previously in the data analysis part, this dataset from movieLens contains 943 users and 1682 movies, the original format of rating data is (user – movie-rating), we converted the data to user x movie ratings matrix for the computation convenience.



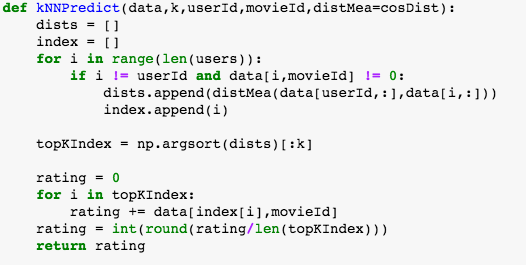
To calculate the similarity/distance between users, we adopted the cosine algorithm because it is compatible with the vector computation and would not be distorted by a scale problem (even this problem does not exhibit in the rating data)



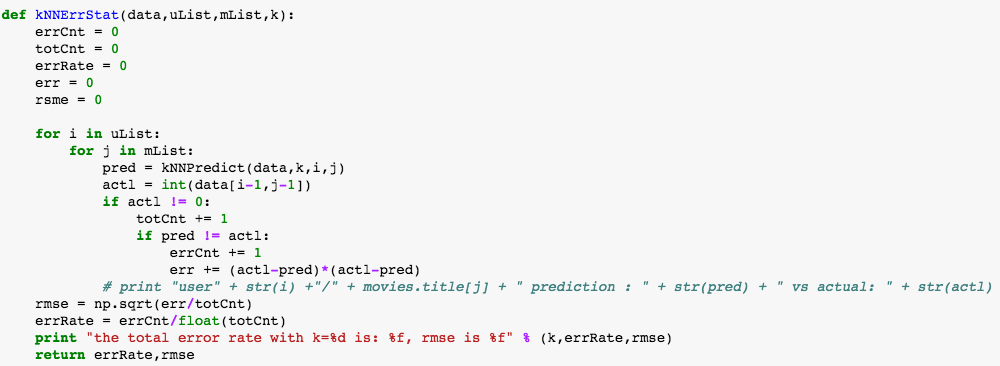
To have a glimpse of the user profile, we wrote a function to present a user’s age, gender, occupation and some of his/her favorite movies.



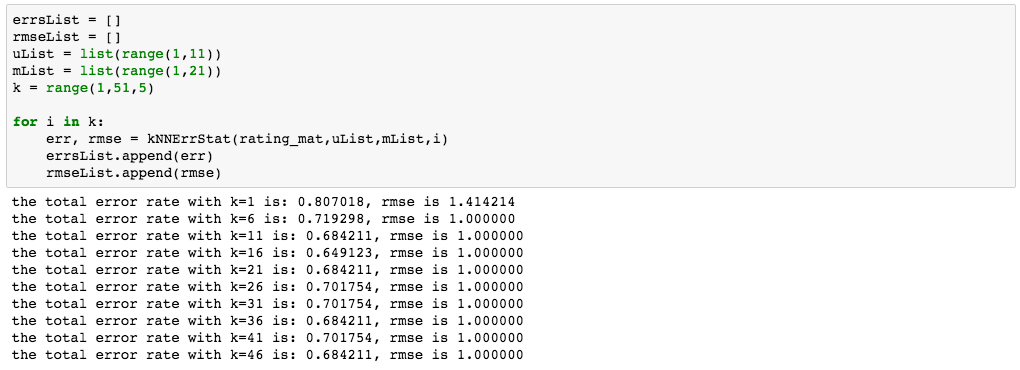
We have implemented the kNN algorithm, the input parameter will include the rating matrix, k number, user id, movie id and distance function (default distance function is cosine distance)

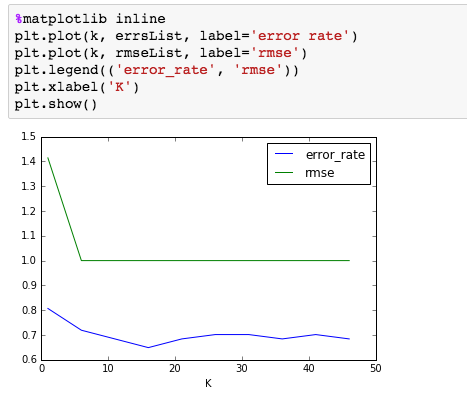


We also wrote a function to loop through different kNN parameters and keep the statistics of kNN error rate and rmse.



Below is a sample run of the application part 1 on the movieLens data, we selected 10 users (user 1 to 10) and 20 movies (movie 1 to 20) and we tried 10 different k (1 to 46, steps = 5).





As the test data indicates, that the accuracy went up as the k becomes larger, the RMSE also went lower as the k increased in the test run. When k = 1, the error rate is relatively high (80.7%) and the error rate went down to about 70% as the k increased, the best error rate is 64.91% when k = 16, the error rate fluctuated around 70% but remained stable even the k increased.

Appendix:

In addition to the report above, several supplement materials are included:

app1.ipynb (The ipython notebook for the demonstration of the system)

app1.py (The interactive python program that user can test on the recommendation system)

app1\_readme.txt (Instructions on how to run the ipython notebook and interactive python program)

Reference:

1. CSC478 course material: <http://facweb.cs.depaul.edu/mobasher/classes/csc478/>
2. Machine Learning In Action
3. Michael J. Pazzani , Daniel Billsus: Content-based Recommendation Systems, <http://www.fxpal.com/publications/FXPAL-PR-06-383.pdf>
4. Wikipedia on Recommendation System: <https://en.wikipedia.org/wiki/Recommender_system>