Project 1

In this project, we perform a comparative study of two movie recommendation systems. The first system, i.e., the base system, implements the user-based recommendation system that uses Pearson correlation for similarity computation between users. Name this system recommendationA.py. This corresponds to the approach discussed in class.

The second recommendation system will be one that is modified based on recommentationA.py. Name this system recommendationB.py. **Some suggestions on how you may want to modifying the base program:**

- Use Jaccard coefficient, Manhattan distance, Cosine similarity, or other similarity measure to compute how similar two users' tastes of movies are to each other;
- Which user's data to be used for weighted rank computation? In the base system, all users' ratings, except for those with negative similarity value, are used for computing the ranking. How may this step be modified? (i.e., only the top n most similar user's ratings are to be used);
- Currently, users who share only one or very few movies may participate in weighted similarity ranking computation. One possible way of modification is to allow only users share at least 25% or 50% of the movies to be used for ranking computation;
- Incorporate demographic information of the users in the data sets.
- others

Evaluation of System Performance

To quantitatively compare the performances of these two systems, an objective evaluation is to be performed. To compute the performance of each system, first we hold out a subset of data to be used solely for testing purpose, we call this data the "test data". The original data subtract the test data is referred to as the "training data". For this project, the test data is formed by randomly selecting 100 users from the original data. The rest of the data forms the training data.

For each user in the test data, we will use the recommendation system to "guess" the most likely rating the user would give based on the rankings of all the remaining movies this user has watched, i.e., all except the movie currently being tested, together with the training data. With this approach, one by one, we can "guess" the predicted rating of all the movies this user has already watched and have given a rating.

Therefore, for each user in the test data, we have two vectors: one with the original ratings of the user, and one with the "guessed" ratings of the user. We can then compare how good the recommendation system did in matching the movie ratings by comparing the similarity of these two vectors. Use the Pearson correlation measure to compute the similarity between these two vectors. This results in the recommendation quality value for this user. Repeat the above process for every user in the test data. Then, compute the average recommendation quality value across the 100 users in the test data.

For example:
Assume Test set contains users "1" and "2"; Training set contains users "3","4", and "5"

	Toy Story	Golden Eye	Four Rooms	Get Shorty	Copycat
"1"		3		3	
"2"	2	3	2	4	
"3"		4	2	5	
"4"			4	3	4
"5"	3	3		5	5

Pseudo code for the evaluation process:

```
For each user in the test data
    For each movie rated by this user
        Remove the rating of that movie for the user
```

Compute a rating for the movie
Add it to the list of movie ratings
Compute the pearson correlation of the estimated ratings with the actual ratings
Compute the average pearson correlation of all the users in the test data

- For user "1":
  - Compute rating for 'Golden Eye'→ record rating=[4.3]
  - Compute rating for 'Get Shorty' → record rating=[4.3, 2.5]
  - R1=Correlation between [4.3, 2.5] and [3, 3]
- For user "2":
  - Compute rating for 'Toy Story'→ record rating=[3]
  - Compute rating for 'Golden Eye'→ record rating=[3, 4]
  - Compute rating for 'Four Rooms'→ record rating=[3, 4, 2.5]
  - Compute rating for 'Get Shorty' → record rating=[3, 4, 2.5, 3.2]
  - R2=Correlation between [3, 4, 2.5, 3.2] and [2, 3, 2, 4]

- Compute the average of R1 and R2

Data

Down the data file (movies100K.tar.gz) and the original program (recommendationDemo.py) from the course web site.

- uncompress the file: unzip movies100K.tar.gz
- open the tar file: tar -xvf movies100K.tar

This creates a directory "movies100K" containing all the components of the data file.

Prepare the project report

The project report (PDF file) should include the following:

1. Describe the approach used in the base recommendation system.
2. Report the average correlation values of the predicted rankings against the original rankings for the test data for the base system;
3. Describe the modification you designed and implemented in recommendationB.py. Explain the Rationale for your design scheme;
4. Report the average correlation values of the predicted rankings for recommendationA.py and recommendationB.py for the test data;
5. Given discussion/explanation of the differences you observe in the results obtained from these two systems.

Submit your programs

Submit your program and project report through D2L Dropbox labeled "Project 1".