CSCI 6600/7350 Project 1 (Due: midnight, Monday Sept 7th)

You are to design a movie recommendation system based on what has been discussed in class. What are some other approaches you may use to modify the systems to potentially improve the quality of the movie recommendations? Show two ways of modifying the system. Name the two systems recommendationA.py and recommendationB.py. To assess the performance of these two modifications as they compare with the original system, an objective evaluation is to be performed to quantitatively compare the qualities of these systems.

1. **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.

2. **Some suggestions on how you may want to modifying the original program:**
    - Use Jaccard coefficient, Manhattan distance, cosine similarity, or other similarity measure you have identified to compute how similar two users' movie tastes are to each other,
    - Which users' data to use for weighted rank computation? Currently, all users' ratings, except for those with negative similarity value, are used for computing the ranking. How can this be modified to improve the way ranking maybe computed? (i.e., only the top n most similar user's rating should be used),
    - Currently, users who share only one or very few movies may participate in weighted similarity ranking computation. One possible way of improving is to allow only users share at least 25% or 50% of the movies to be used for ranking computation.
    - Try to incorporate demographic information of the users in the data sets

3. **How to evaluate the performance of the systems?**
    To compute the performance of the 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, lets make the test data containing movie ratings for 100 users. To make the test data more general, these 100 users should be randomly selected from the original data.

    For each of the movies a user in the test data has rated, apply the recommendation system to estimate the ranking of that movie. This estimation will use the rankings from all the remaining movies this user has made, together with the training data. Compare this set of computed/predicted weighted rankings to the original/true rankings of the user using the Pearson correlation measure. This results in the correlation value for this user.

    Repeat the above process for every user in the test data. Then, compute the average Pearson correlation values across the 100 users in the test data.

4. **Prepare the project report**
    Describe the results from the project in the project report. The report should include the following:
    1. The base system used for comparison purpose is the user based recommendation system that uses Pearson correlation similarity for computing as discussed in class. Report the average correlation values of the predicted rankings against the original rankings for the test data for the base system.
    2. Describe the two modifications you programmed in recommendationA.py and recommendationB.py:
        a. Rationale for your scheme
    3. Report the average correlation values of the predicted rankings for recommendationA.py and recommendationB.py for the test data.

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