# KSE 643 HW #2 (Individual Assignment)

Due: March 28, 2019

MovieLens dataset is one of the most popular benchmark datasets used to test the potency of various collaborative filtering models. The full version of the dataset consists of more than 26,000,000 ratings, 45,000 movies by 270,000 users. However, for the sake of fast execution, we will be using a smaller dataset. To do so, from https://www.kaggle.com/prajitdatta/movielens-100k-dataset/version/1, download the dataset, which is a snapshot of MovieLens dataset that contains 100,000 ratings applied by 1,000 users to 1,700 movies. Your task is to build a user-based collaborative filtering based on ratings of the users given in the dataset.

Here are brief descriptions of the data (copied from the above Kaggle site).

ml-data.tar.gz -- Compressed tar file. To rebuild the u data files do this: gunzip ml-data.tar.gz tar xvf ml-data.tar mku.sh

u.data -- The full u data set, 100000 ratings by 943 users on 1682 items. Each user has rated at least 20 movies. Users and items are numbered consecutively from 1. The data is randomly ordered. This is a tab separated list of user id | item id | rating | timestamp. The time stamps are unix seconds since 1/1/1970 UTC

u.info -- The number of users, items, and ratings in the u data set.

u.item -- Information about the items (movies); this is a tab separated list of movie id | movie title | release date | video release date | IMDb URL | unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | The last 19 fields are the genres, a 1 indicates the movie is of that genre, a 0 indicates it is not; movies can be in several genres at once. The movie ids are the ones used in the u.data data set.

u.genre -- A list of the genres.

u.user -- Demographic information about the users; this is a tab separated list of user id | age | gender | occupation | zip code The user ids are the ones used in the u.data data set.

u.occupation -- A list of the occupations.

u1.base -- The data sets u1.base and u1.test through u5.base and u5.test u1.test are 80%/20% splits of the u data into training and test data. Each of u1, ..., u5 have disjoint test sets. ; this if for 5 fold cross validation (where you repeat your experiment with each training and test set and average the results). These data sets can be generated from u.data by mku.sh.

ua.base -- The data sets ua.base, ua.test, ub.base, and ub.test ua.test split the u data into a training set and a test set with ub.base exactly 10 ratings per user in the test set. The sets ub.test ua.test and ub.test are disjoint. These data sets can be generated from u.data by mku.sh.

allbut.pl -- The script that generates training and test sets where all but n of a users ratings are in the training data.

mku.sh -- A shell script to generate all the u data sets from u.data.

After downloading the dataset and uncompressing it, you need to split the ratings dataset in such a way that 70% of a user's ratings is in the training dataset and 30% is in the testing dataset. Second, you need to build a ratings matrix where each row represents a user and each column represents a movie based on training dataset. Therefore, the value in the *i*th row and *j*th column denotes the rating given by user *i* to movie *j*. To measure the performance of the model, RMSE metric is used in this assignment. Mathematically, it is represented as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum(y_i - \hat{y}_i)^2}$$

where $y_i$ is the *i*th the real target value and $\hat{y}_i$ is the *i*th predicted value. Now, you need to answer the following four questions.

Q1. Build a simple version of the collaborative filter in such a way that outputs the mean rating for the movie by all the users who have rated it. Note that here the ratings of each user is assigned an equal weight. If some movies are available only in the test set and not in the training set, assign default to a rating of 3.0. What is the RMSE score obtained by this model? Provide a snapshot of the console output from your code.

Q2. Build a collaborative recommender system that utilizes the Peason correlation coefficient to give differential weights to the users. This is to give more preference to those users whose ratings are similar to the user in question than the other users whose ratings are not. If some movies are available only in the test set and not in the training set, assign default to a rating of 3.0, as you did before. What is the RMSE score obtained by this model? Provide a snapshot of the console output from your code.

Q3. Build an interactive prediction model in such a way that it returns top 3 movies (movie ID, movie title, predicted rating score, actual rating score) from the testing dataset when a user ID is entered. Do this for 10 random users and report the RMSE score obtained by the interactive session? Provide a summary of the results and the RMSE score computed.

Q4. For this question, build an item-based collaborative recommender system that utilizes the Cosine similarity measure instead of Pearson. What is the RMSE score obtained by your proposed model? Provide a snapshot of the console output from your code.

Document your answers using MS Word and name the file as ***individual_assignment2_yourstudentid_yoursurname.docx.*** Email me the word file and your working code. My TA will review your code and grade the questions. Note that python or Java is encouraged due to its convenience for available libraries; however, any programming language (including R) is allowed for implementation.

**Please submit the printed copy** of the report file on the due date. You will also need to **email me the report and code before you come to class on the due date**. **This assignment counts 100 points**.