**CMPT-741 Course Project – Recommender System**

**Task Description – Version 2 (Sep 18 2017)**

For project related questions, please contact the TA, "Jiaxi Tang" [jiaxit@sfu.ca](mailto:jiaxit@sfu.ca). His office hours: Thursday 2-3pm at ASB 9808.

**Introduction**

***Recommender Systems*** (RS) are a subclass of information filtering system that seek to predict the "rating" or "preference" that a user would give to an item.

In this project, you are asked to build a RS for Yelp, a company focusing on providing crowd-sourced business ratings and reviews. Briefly speaking, for a giving user-item pair, you will predict the rating (1-5) based on historical ratings given by many users. You can use any programming language you want, though we highly encourage you to use ***Python***, a widely-used programming language in data science and machine learning. This is a group project with each group having 2 or 3 students working together. You are highly recommended to start this project as early as possible. Counting on the last 1 or 2 week to finish the project will be unrealistic. You can use the discussion board on Coursys (<https://courses.cs.sfu.ca/>) to find a team member.

**Dataset**

We will use a ***Yelp Dataset*** in this course project. This data set, given in the file *train\_rating.txt*, is a collection of 2038130 user-item pairs and the corresponding rating. An example is

*293726,103641,30029,4,2015-09-23*

The first three columns are the training instance ID, user ID, and item ID respectively. 4 is the rating this user gives to this item chosen from {1,2,3,4,5}, and 2015-09-23 is the date on which the rating was given. Figure 1 shows an example of the raw data from which such rows are extracted, where the user may also give a text review, in addition to the rating. Typically a user rates only a small subset of all items. The goal of this data set is to learn a good recommender that can predict the rating of a user on unrated items.



Figure 1. An example of Yelp dataset

Note: The rating of a user provides basic information for building a RS, whereas date and text review provide additional information. Our class covers building a RS using only rating information. If you can improve the accuracy of RS by incorporating date and text information, you will get bonus points (for project report scores).

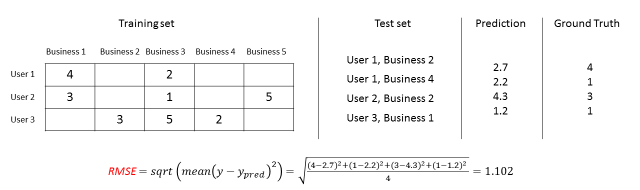
**Evaluation and Grading**

You can use any RS construction methods, i.e., item-based, content-based, matrix factorization, etc. The total 100 points of this project consist of:

(1) 40% for a 4-page final report: describe the key steps of your approach, discussion on techniques used and choices made, and the code. The report is due on Dec 12 2017.

(2) 60% for evaluating the prediction accuracy: We will host a Kaggle competition to evaluate the prediction accuracy at [*https://inclass.kaggle.com/c/cmpt-741-recommendation-contest*](https://inclass.kaggle.com/c/cmpt-741-recommendation-contest)*.* The evaluation is based on another data set *test\_rating.txt* containing 108024 user-item pairs with known ratings.

Your RS model is first learnt from *train\_rating.txt* and is evaluated using *test\_rating.txt.* For evaluation, the model will read each row in *test\_rating.txt* and make a prediction of rating, then output the results into an output file, and you submit the output file using Kaggle. The submitted ratings will be compared with the known ratings and you will receive a score for your prediction, called RMSE, which measures the error of predicted ratings. An example of computing RMSE is as follows, where “Ground true” is the known rating and “Prediction” is the predicted rating.



Before the submission deadline, you can evaluate your model using 30% of the test data, released to you at least two weeks before the deadline. The final score for grading is calculated by the remaining 70% of the test data, which will be released to you only at the submission deadline. Therefore, you should not overfit the first 30% of test data because your goal is predicting well on the remaining 70% test data. A better practice is to reserve some training data as validation set or to use cross-validation.

Summary of files:

train\_rating.txt - the training set

test\_rating.txt - the test set (30% will be released for your own evaluation, and the remaining 70% will be used for grading).

train\_review.json - the corresponding review of each user-item pair in training set

sampled\_submission.csv - a sample submission file in the correct format

All data sets are hosted at [*https://inclass.kaggle.com/c/cmpt-741-recommendation-contest*](https://inclass.kaggle.com/c/cmpt-741-recommendation-contest)

**Kaggle**

Kaggle is an online platform for data mining contest. People can freely host public or private contest on it, by giving the solution file (i.e., ratings of user-business pairs in test set) to Kaggle. People can submit their prediction, then Kaggle will judge your prediction and return a score based on your prediction. You can find more information from reference [1] and [2].

**Submission Guide:**

1. Once you form a team, create the Team Name on Kaggle following this format:

sfuID of team member 1\_sfuID of team member 2\_sfuID of team member 3

For example:

lza118\_ lza118\_ yza68

1. Using Kaggle, submit the sampled\_submission.csv file by the following contest start date.

(3) Each team could make at most 2 submissions per day.

**Rules**

contest start date: 12:00 am, 1st Oct 2017.

contest end date: 12:00 am, 12th Dec 2017.

You are free to use code from any other sources (e.g. Github), but be sure that you fully understand it before using it, otherwise you will get wired results with no reason in mind. You are also free to use other valid data sources (you can ask TA if you are not sure whether a data source is valid) for your model learning, for example, you can use Calendar to check if a given date is a holiday. Each group should work on the project independently, including coding, report, and submission. Sharing code and solution across different groups is strictly not permitted, and one will get the zero mark for sharing code and solution outside its group.

**Resources**

The following links contain some useful information on (1) Kaggle and (2) Recommender System.

* [1] Kaggle Leaderboard: https://www.quora.com/What-is-the-difference-between-public-and-private-leaderboard-in-Kaggle
* [2] Kaggle tutorials: https://www.kaggle.com/wiki/Tutorials
* [3] Matrix Factorization for Recommedation: https://datajobs.com/data-science-repo/Recommender-Systems-%5BNetflix%5D.pdf
* [4] HFT for review-based recommendation: https://cs.stanford.edu/people/jure/pubs/reviews-recsys13.pdf