**Predicting Movie rating based on User Based Collaborative Filtering(UBCF)**

*- by Ratul Ramchandani*

This is a case study of predicting user ratings for various movies not rated by the user. There is

a data file ‘ratings.txt’ which has the movie reviews from movielens. There are 1,00,000 lines

of data. The first few lines are shown below:



The idea is to split the data into training and test data set by means of stratified sampling. We

will predict the ratings for the test data set and result would look like:



To achieve predicted rating, we will apply user based collaborative technique (UBCF). The assumption is that users with similar preferences will rate items similarly. Thus missing ratings for a user can be predicted by first finding a neighbourhood of similar users and then aggregate the ratings of these users to form a prediction.

The neighborhood is defined in terms of similarity between users. We will apply the popular similarity measures Jaccard similarity, Pearson correlation coefficient and the Cosine similarity. We will use the R-package: **recommenderlab**. A highly recommended read on the recommendation algorithms can be found [here](https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf).

On the training data we will apply the prediction algorithm, which will predict a rating for every

movie that a user has not rated. Once the prediction is done, then we will parse the testing data row wise. For every user ID and movie ID combination we will extract the predicted rating.

We will perform the above 2 steps for all the 3-similarity metric, one at a time. Post this we

will calculate the Normalized Mean Absolute Error (NMAE) for all the three models and perform

comparison.



NMAE scores for models generated by applying different similarity measure:



Analysis of different models:

For the given data set the NMAE scores for different models are very similar and hence a best model cannot be decided. On an average the NMAE is 18% for all three models. This translates to an absolute error value of around 0.8 rating on an average.

Details of attached files:



The github link for the files can be found here -- <https://github.com/creativecoderr/recommenderSystems.git>