Project 1 - MovieLens

R Markdown

Summary:

This is a prediction for the MovieLens data set. For this project a 10M version of the MovieLens dataset is used.

Matrix factorization is the final method used for this prediction. Matrix factorization is often better in terms of prediction accuracy and time needed to train the model conbared to methods like nearest neighboor. Matrix factorization is to factorize a matrix. Matrix factorization can be used to discover latent features underlying the interactions between two different kinds of entities. It is used to predict movie ratings in this particular work.

After testing the model using the course prediction method, Matrix factorization proved to give us a better RMSE. In this document, I have outlined some of predictions using the course model and then finish it with matrix factorization.

Analysis:

Using the MovieLens 10M dataset, I created an edx dataset and a validation dataset. Validation set is 10% of the MovieLens data. There was no data cleaning performed. For visualization, I created a table that shows the method used and the resulting RMSE. The table is given in the results section below.

For comparison purpose, the starting point was a naive approach of using just the average.

I started the prediction by checking RMSE on user effects compared to just the average. How did the rating of users affect the RMSE? This resulted in a much better outcome than just using the average. The RMSE went from 1.0612 to 0.8292. This was very good. However, I wanted to see if there were additional effects (biases) that could impact or improve the RMSE. Therefore, I started adding and removing various effects.

Adding Year to user effect did not affect the result. However, just using year effect increased the RMSE. This showed me user effect was signifant enough for additional validation. I continued to check user effect with month effect. Once again this resulted in a similar RMSE when doing just user effect by itself. Similar to year effect, when using just month effect, RMSE increased.

Next, I made the assumption that user with any additional bias would not change results from using only user effect. To make sure this was a correct assumption, I added genre effect to user effect. To my surprise, RMSE when significantly higher. Removing user from the last method improved RMSE but not to the point where it was better that user effect. Just the average was much better than the genre effect by itself.

At this point Movie plus User Effect became my baseline to validate against Matrix Faxctorization as this was my inital desire for the project. Will Matrix Factorization produce a lower RMSE?

Results:

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Method	RMSE
Just the average	1.0612018
Movie + User Effects Model	0.8292477
Movie + User Effects + Year Effects Model	0.8292408
Movie + Year Effects Model	0.9416836
Movie + User Effects + Month Effects Model	0.8292451
Movie + Month Effects Model	0.9437745
Movie + User Effects + Genre Effects Model	1.2448923
Movie + Genre Effects Model	1.1622867
Recosystem	0.8114974

Note - I added `echo = FALSE` parameter to the R code above to prevent the printing of the R code that generated the table.

Conclusion:

After trying out various effects and method, I discovered that MAtrix Factorization gave the best result. While movie + user effect had an acceptable RMSE, I discovered a different method that improved the result about 2.2%.

Matrix Factorization gave an RMSE of 0.8114. As a result this is my recommendaded method for movie recommendation system.