Movie Lens Recommendation System

Tuong Nguyen

February 18, 2019

Abstract

This project is based on writing a movie recommender, since online videos watching has been increasing amount because more people are cutting their traditional cable TV. This project will be using multi linear regression modeling to give the predicted rating for a movie. It will show that the more independent variables that it used in the model, the more accurate the model will be. This can be verified with the RMSE.

Introduction

Online shoping and videos watching has been explosive within the last decades. It is only fitting that online companies such as Amazon, Netflix, etc. would want to cash in on it. A recommendation system from such companies would attract more customers.

A recommendation system is an important component part of online experiences for many users. It act as a filtering system which it recognizes the users' past ratings for products and make recommendations to which products the users would most likely to choose. This will enhance the users experiences with the online shoping and increase the value of the services.

This paper will explore a movie recommendation system which will be using the Multiple Linear Regression method. It will then uses the RMSE score to determine the prediction accuracy. RMSE (root mean squared error) is the measure of how well the model performed. It measures the difference between predicted values and the actual values. The error term is important because we want to minimize the error.

Data

The data for this paper was obtained from https://grouplens.org/datasets/movielens/10m/. The data set contains 1000054 ratings and *95580* tags applied to *10681* movies by *71567* users for online recommender service MovieLens. All users were selected at random and rated at least 20 movies. Each users represented with a unique id. The data file was in zipped format and downloaded using R programming function download.file(). The zipped file contains 3 different files: movies.dat, ratings.dat and tags.dat. The movie ratings range from 1-5. The data does not contain any empty or zero ratings. The following histogram in Figure 1 showed the ratings distribution where ratings 4 and 5 achieve the most. The data then partitioned into edx and validation set. The validation will be test set where the RSME will be tested against. The histograms also showed that there are some users give more ratings and some movies get rated more than others.

Methods

This paper will concentrate on the Multiple Linear Regression method to build the movie recommendation system. The model will be predicting the rating for movie i by user u and average ranking for movie i. So, the linear regression model looks as follows:

$$Y_{u,i} = \mu + b_i + b_u + \epsilon_{u,i}$$

where $Y_{u,i}$ denote as the dependent variable, rating for movie i by independent variables user u, μ is the average of ratings, b_i represent the average ranking for movie i and b_u is a user specific effect. ϵ_i are independent errors that are centered at 0 and μ and sampled at the same distribution. Since our data is too large to run under R studio with linear model function *lm()* and *predict()* using our current Windows 10 machine with [12.0 GB of RAM and processor with Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz, 2301

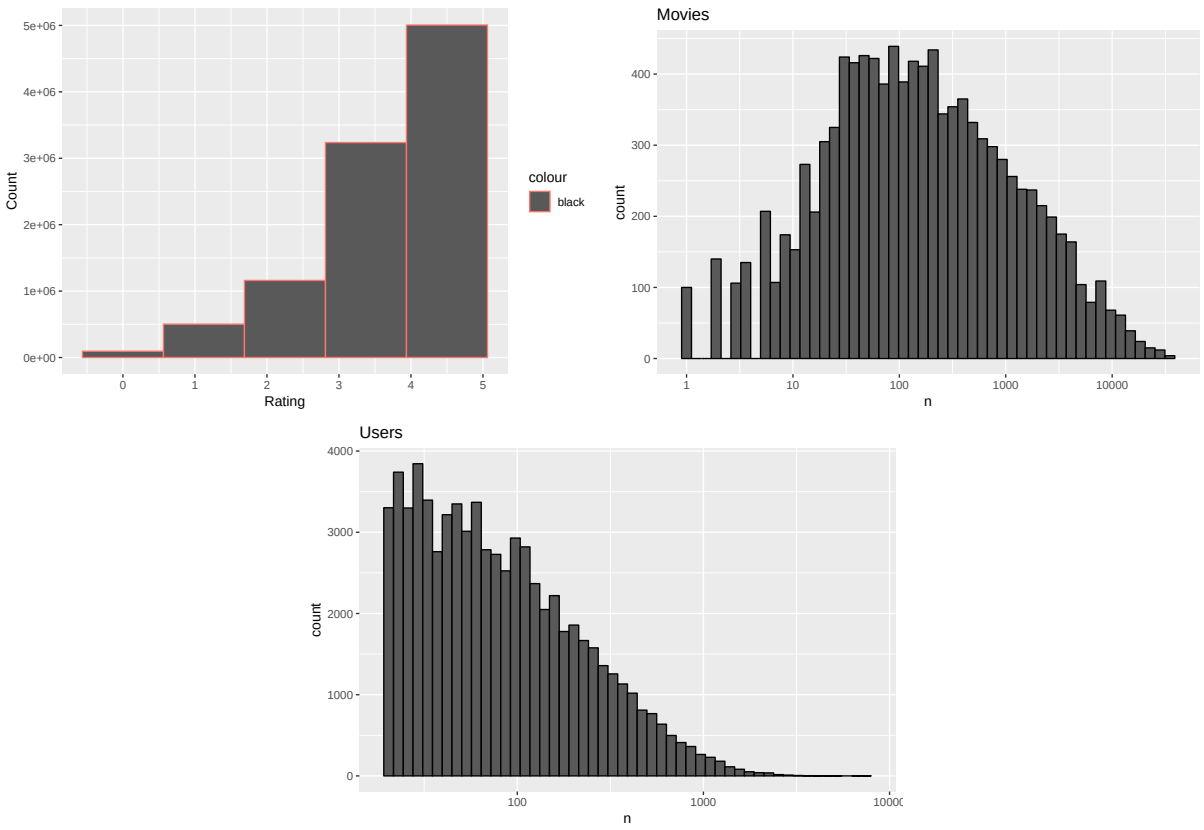

Figure 1: MovieLens Ratings/movies/users distributions

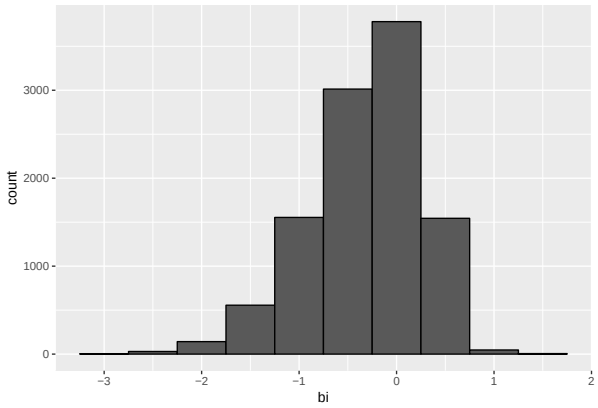

Figure 2: Movie Averages distributions

Mhz, 2 Core(s), 4 Logical Processor(s)] , we will have to resort estimate those values using R programming languages.

```
lm(rating ~ as.factor(userId) + as.factor(movieId), data = edx)
```

Analysis

The μ represent the average of all ratings in training set, edx, and b_i represent the average ranking for all the movies. We calculate these values using R with results of $\mu = 3.512465$ and histogram plot of all b_i. If you rewrite the equation above, assuming the user is not affecting the rating outcome, the $\widehat{b_i} = y_{u,i} - \widehat{\mu}$.

Since this is a multi linear regression, the users will also have an affect the movie ratings outcome since some of the users rated more than others. The calculation for the average users' ratings, can represent with b_u, so the new equation, $\widehat{b_u} = y_{u,i} - \widehat{\mu} - \widehat{b_i}$. Now, with the joining the movie averages and user rating averages to obtain the predicted ratings. The table below will show the first few movies with the predicted ratings. From the table, it showed that "Dumb and Dumber" movie has a five ratiting. The users average rating, $b_u = 1.679234$, movie average rating, $b_i = -0.5773440$, and the predicted rating $y_{u,i} = 4.614356$.

Table 1: Predicted Rating Results sample

title	bi	bu	pred
Dumb & Dumber (1994)	-0.5773440	1.6792347	4.614356
Jurassic Park (1993)	0.1510566	1.6792347	5.342757
Home Alone (1990)	-0.4568130	1.6792347	4.734887
Rob Roy (1995)	0.0175932	-0.2364086	3.293650
Godfather, The (1972)	0.9029008	-0.2364086	4.178957

Results

Now, are these predictions accurate? To answer this question, RMSE was used to determine the accuracy of the predictions. The formula for RMSE(root mean squared error) is as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(\widehat{y}_{u,i} - y_{u,i})^2}$$

where $\widehat{y}_{u,i}$ is the prediction and $y_{u.i}$ is the actual. Since RMSE is measured errors, it is best to achieve a low value as possible. In R Studio, the following code was used to calculate the RMSE and its results, showed

0.8653488.

```
# Calculate the RMSE of prediction using both movie and fuser effect
rmsewith2var <- RMSE(predicted_ratings$pred, validation$rating)
print(rmsewith2var)
```

```
## [1] 0.8653488
```

If users effect was not taken into account, the RMSE would be much larger from the table shown below. However, adding genres to the equation model, it reduce the RMSE down .001. Thus, user and movie effect has the most impact on the multi linear model of prediction.

Table 2: Summary of RMSEs

method	RMSE
User and Movie Effect	0.8653488
Movie Effect	0.9439087
Movie, User, and Genres Effect	0.8649469

Conclusion

In conclusion, the multi linear model does a good job in predicting the rating for the movie. Adding the extra independent variables effect such as user and movie will lower the RMSE if it were just user or movie effect in the model. However, with an additon to genres to the linear model, it reduces the RMSE by only .001. Thus, genres doesn't have much of an effect on the linear model. As limitation to computing power, the linear model or predict function can't be used, so the project was forced to used normal calculation using R studio. For future project, with enhance computing power, I would like to test out whether *Knn* and K-mean clustering method will do a better job with reducing the RMSE. In addition, implement residuals and errors to see if the model will improve.