Predicting Movie Rating

Mahalingam Kamalahasan

6/12/2019

Note: I am running version R3.6 To test the script against the dataset, please download the edx and validation dataset from https://drive.google.com/drive/folders/1IZcBBX0OmL9wu9AdzMBFUG8GoPbGQ38D. Look for readRDS commands (one for edx and other one for validation variable and replace the argument to the location where you downloaded the dataset)

Predicting Movie Rating

Overview:

We have approximately 10 millions records on movies rated by users. I would like to build a model that helps to predict a movie rating by a specific user. To determine effectiveness of the model we would use Root Mean Square Error (RMSE) as the means to rate the success of the model. The goal is to achieve root mean square error of less than 0.87750

About the data set: To be exact the training dataset contains 9000055 rows and 6 columns. The test data set contains 999999 rows and 6 columns.

```
#########################################################################
################## LOADING THE REQUIRED PACKAGES ####################
#########################################################################
# If tidyverse package does not exist, load from the following repo and install it
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")

# If caret package does not exist, load from the following repo and install it
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")



#########################################################################
########## LOADING THE TEST and VALIDATION DATA SETS ###############
#########################################################################
# Loading edx, the rds file was downloaded from google drive and stored locally
# The file was retreived from https://drive.google.com/drive/folders/1IZcBBX0OmL9wu9AdzMBFUG8GoPbGQ38D
edx <- readRDS("../dataset/edx.rds")

# running the dimension on edx to make sure I have correct number of rows and cols
dim(edx) # rows: 9000055      cols: 6
```

```
## [1] 9000055       6
```

```
# Loading Validation dataset, the rds file was downloaded from google drive and stored locally
# The file was retreived from https://drive.google.com/drive/folders/1IZcBBX0OmL9wu9AdzMBFUG8GoPbGQ38D
validation <- readRDS("../dataset/validation.rds")
```

```
# running the dimension on validation to make sure I have correct number of rows and cols
dim(validation) # rows: 999999     cols: 6
```

```
## [1] 999999      6
```

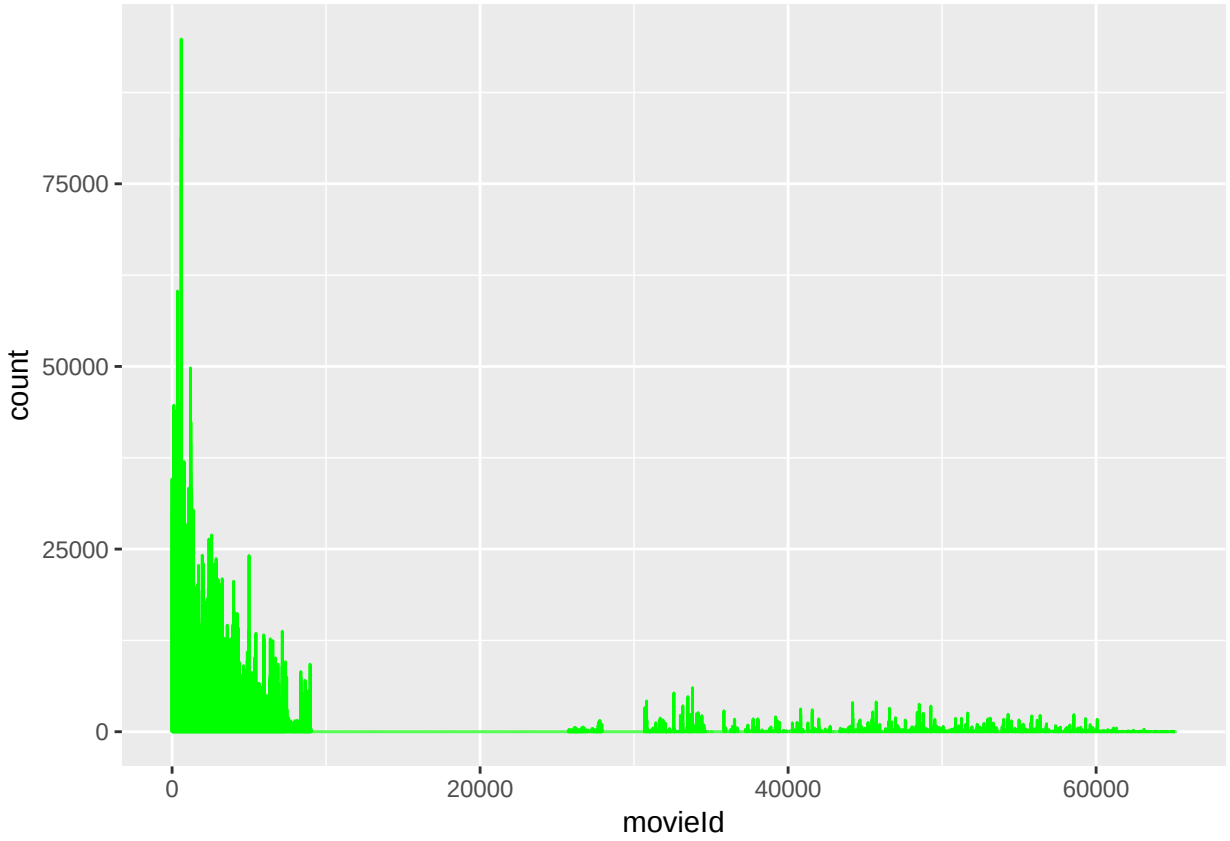
The dataset contains 6 columns, you can see their column names below.

```
## [1] "userId"    "movieId"   "rating"    "timestamp" "title"     "genres"
```

Method and Analysis

I would like to build the model that is simple and yet effective. We will do that by calculating first the average rating of all the movies given by the user. We then will apply the movie effect because some are blockbusters and others are not. You can see this from the histogram below.

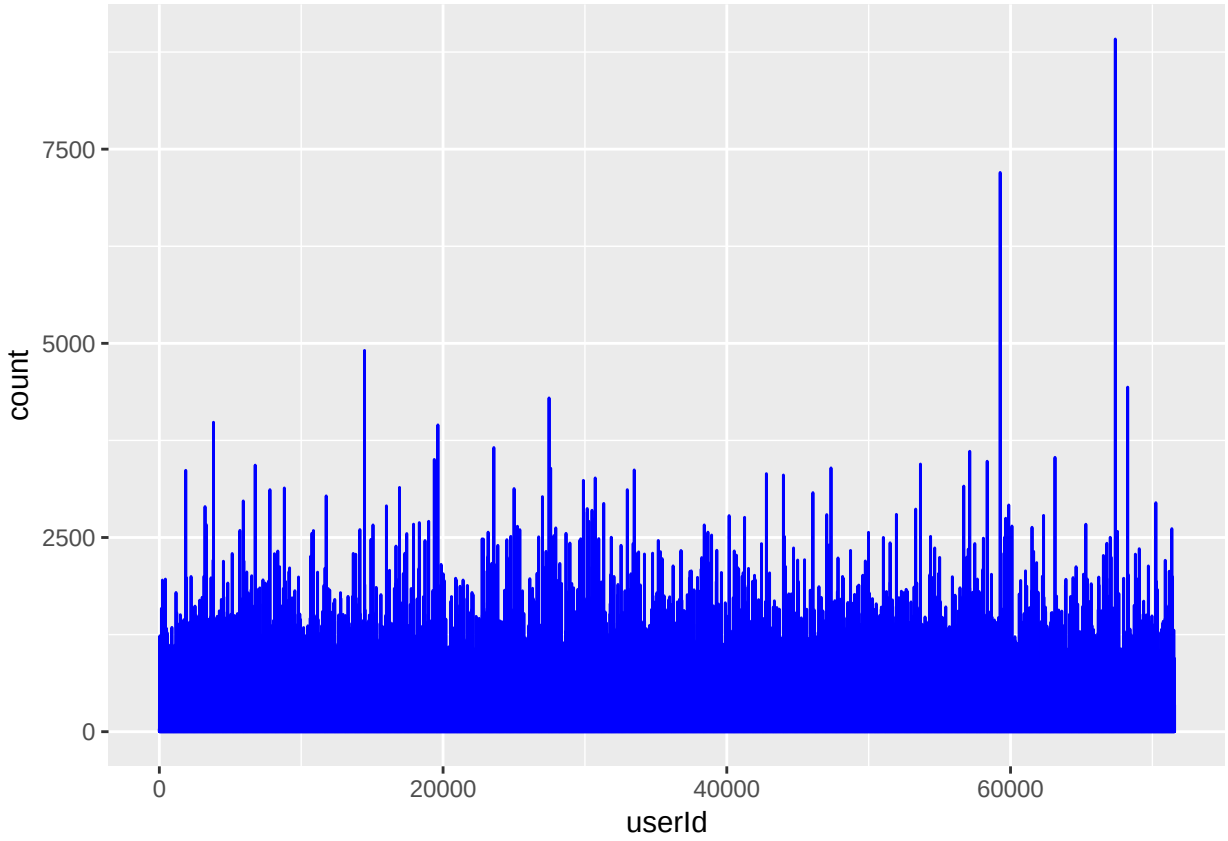
```
# You can see some movies are rated more than the others, because some are popular,
# blockbusters, critically acclaimed and etc. You can see them visually through the
# below plot
edx %>% ggplot(aes(movieId,color="movie ratings")) + geom_histogram(binwidth = 5, color="green")
```



Method and Analysis - User effect

Similarly you can see the user effect it is clear from the below plot tat not all users rate the same level and some rate higher than the others

```
edx %>% ggplot(aes(userId,color="user ratings")) + geom_histogram(binwidth = 5, color="blue")
```

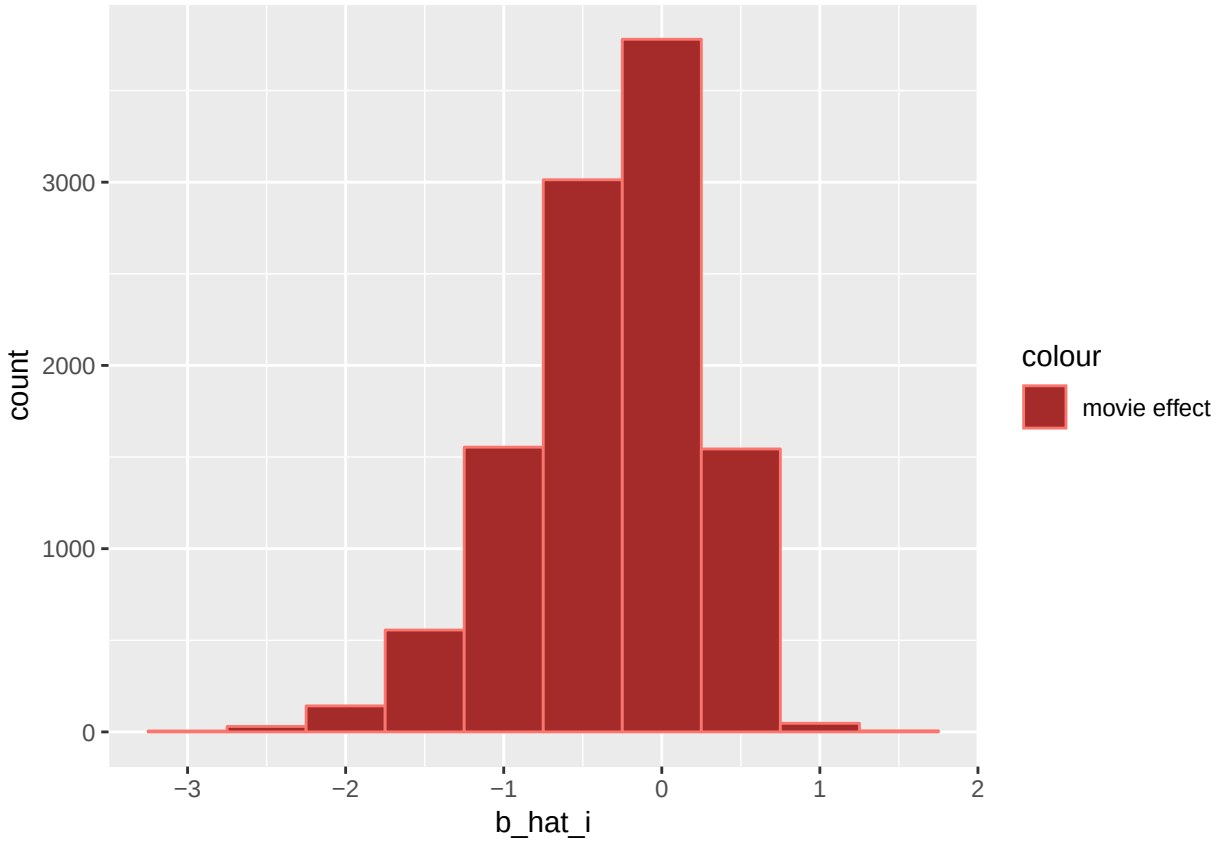


Method and Analysis - Building the equation

The above plots tell me that at the minimum to build a model that is effective to predict I need average of all movie ratins and add movie and user effect.So the equation is y_hat_iu = mu + bi + bu + epsilon, where mu is the average rating of all the movies, y_hat_iu is the predicted rating of a movie i by user u, bi is the average of movie effect of a movie i rated by all users and bu is the average of user effect on all movies rated by the user. Epsilon is the noise introduced, that is the residual obtained after determining all kinds of relationship we saw in the data. For simplicity we will assume that the epsilon is 0. Therefore the derived equation for the model would become: y_hat_iu = mu + bi + bu

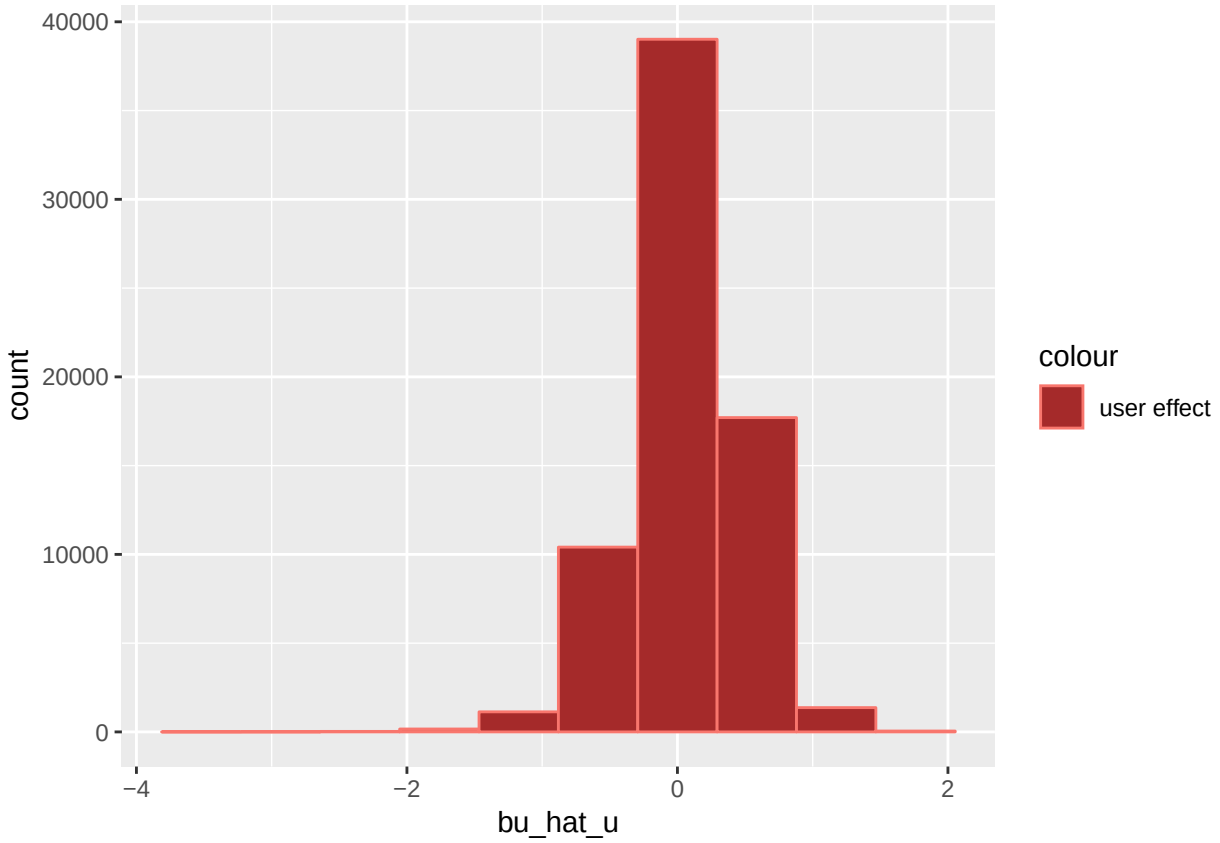
From the below histogram plot it is helpful to see the movie effect where we can see movie effect helping the good movies and discrediting the not so good movies.

```
bi %>% ggplot(aes(b_hat_i, color="movie effect")) + geom_histogram(bins=10,fill="brown")
```



At the same time you can also see from the below histogram plot it is helpful to see the user effect, i.e some users in general rate the movies they see optimistically than others. Some users more pessimistic than others.

```
bu %>% ggplot(aes(bu_hat_u, color="user effect")) + geom_histogram(bins=10,fill="brown")
```



Results

ROOT MEAN SQUARE ERROR (RMSE)

We will build a function for root mean square using the following code. Which we then call with predicted rating and actual rating in the test data set i.e validation dataset

```
RMSE <- function(predictedrating,actualrating)
{
  sqrt(mean((predictedrating-actualrating)^2))
}
```

Results - Contd

```
rmse_results <- RMSE(y_hat_iu,validation$rating)
rmse_results
```

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

Conclusion

We were able to build the model using the equation y_hat_iu = mu + bi + bu. We then were able to run this against the validation set and calculated the RMSE. We obtained RMSE of 0.8653488 which tells us to we are able to predict the rating for the movie pretty close to the actual user's rating for any movie.

RMSE RESULT: 0.8653488