

HarvardX: PH125.9x Data Science

MovieLens Project

April 25, 2019

Sanskriti Anurag Srivastava

I. Introduction

A recommender system is a subclass of information filtering system that seeks to predict the “rating” or “preference” a user would give to an item. Recommender systems are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general. There are also recommender systems for experts, collaborators, jokes, restaurants, garments, financial services, life insurance, online dating, and Twitter pages. This project will build a movie recommendation system using the 10M MovieLens Dataset collected by GroupLens Research, which includes 10,000,000 ratings on 10,000 movies by 72,000 users.

II. Executive Summary

The objective of this project is to train a machine learning algorithm that predicts user ratings (from 0.5 to 5 stars) using the inputs of a provided subset (edx dataset) to predict movie ratings in a provided validation set. The root-mean-square error (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. RMSE is a measure of accuracy, to compare forecasting errors of different models for a dataset and not between datasets, as it is scale-dependent. RMSE is always non-negative, and a value of 0 (almost never achieved in practice) would indicate a perfect fit to the data. In general, a lower RMSE is better than a higher one. The following key steps are followed in order to perform the analysis and make the conclusion:

- Prepare Data
- Summarize Dataset
- Visualize Dataset
- Evaluate Algorithm
- Evaluate Validation set In the project, three models (“Simple Average”, “Movie_Effect” and “Movie+User_Effect”) are developed and their accuracy is assessed using their resulting RMSE. Finally, the best resulting model, “Movie + User_Effect Model” with RMSE of 0.8426, is ran directly on the validation set to predict the movie ratings. The RMSE result on validation dataset of 0.8294 is

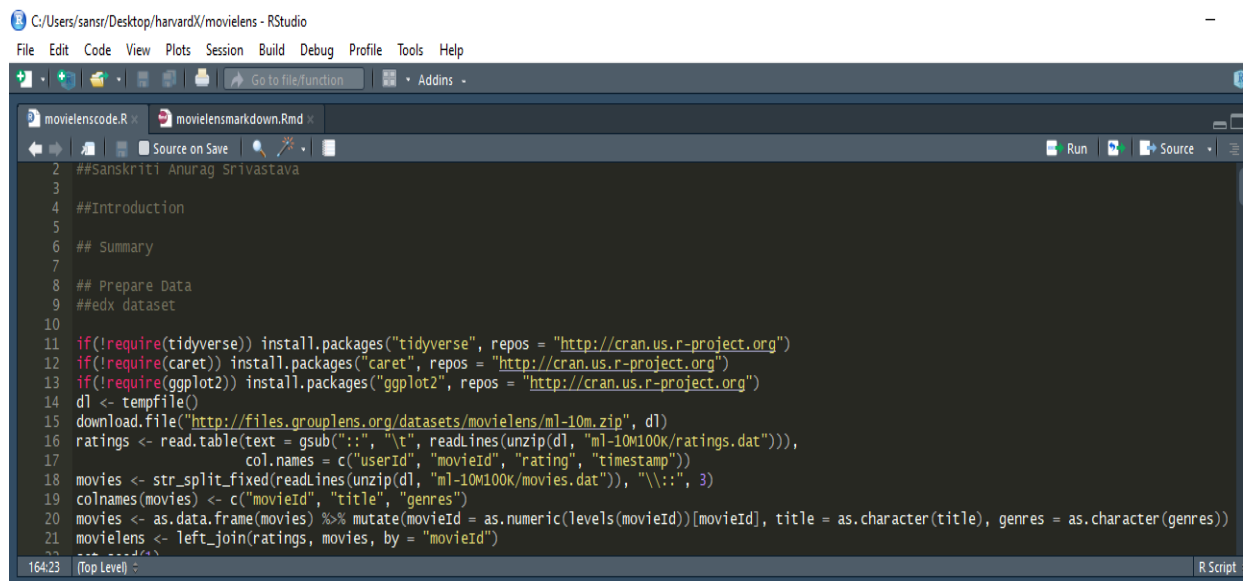
lower than the results on test dataset of 0.8426, suggesting that the **“Moive+User Effect”** model is likely a reliable prediction model.

III. Prepare Data

(edx dataset)

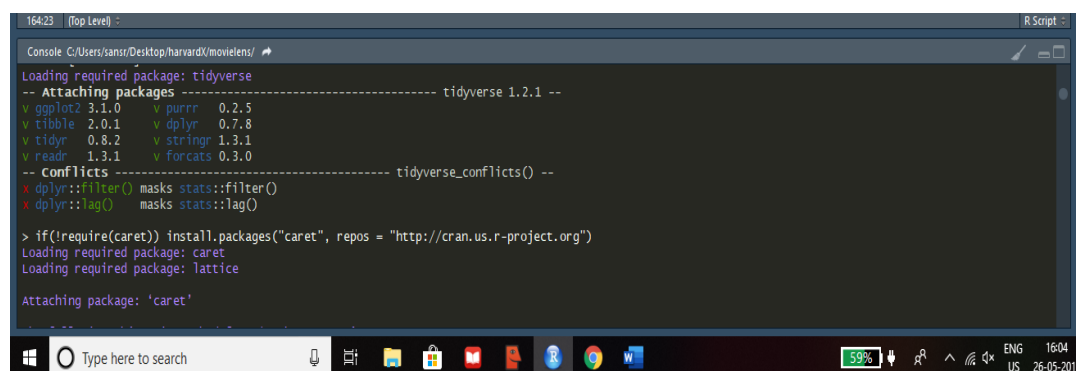
The following edx section is used to perform the analysis in this project. The ggplot2 package is added to the edx set.

CODE TO PREPARE DATASET:



```
1 ##Sanskriti Anurag Srivastava
2
3
4 ##Introduction
5
6 ## Summary
7
8 ## Prepare Data
9 ##edx dataset
10
11 if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
12 if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
13 if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
14 dl <- tempfile()
15 download.file("http://files.grouplens.org/datasets/movielens/ml-10m.zip", dl)
16 ratings <- read.table(text = gsub(":", "\t", readLines(unzip(dl, "ml-10M100K/ratings.dat"))),
17   col.names = c("userId", "movieId", "rating", "timestamp"))
18 movies <- str_split_fixed(readLines(unzip(dl, "ml-10M100K/movies.dat")), "\\:", 3)
19 colnames(movies) <- c("movieId", "title", "genres")
20 movies <- as.data.frame(movies) %>% mutate(movieId = as.numeric(levels(movieId))[movieId], title = as.character(title), genres = as.character(genres))
21 movielens <- left_join(ratings, movies, by = "movieId")
22
23
24 164/23 (Top Level) | R Script
```

OUTPUTS:



```
164/23 (Top Level) | R Script
Console C:/Users/sansr/Desktop/harvardX/movielens/
Loading required package: tidyverse
-- Attaching packages ----- tidyverse 1.2.1 --
v ggplot2 3.1.0 v purrr 0.2.5
v tibble 2.0.1 v dplyr 0.7.8
v tidyr 0.8.2 v stringr 1.3.1
v readr 1.3.1 v forcats 0.3.0
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag() masks stats::lag()

> if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
Loading required package: caret
Loading required package: lattice
Attaching package: 'caret'
```

```
16423 (Top Level)
Console C:/Users/sansr/Desktop/harvardX/movielens/
lift

> if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
> d1 <- tempFile()
> download.file("http://files.grouplens.org/datasets/movielens/ml-10m.zip", d1)
trying URL 'http://files.grouplens.org/datasets/movielens/ml-10m.zip'
content type 'application/zip' length 65566137 bytes (62.5 MB)
downloaded 62.5 MB

> ratings <- read.table(text = gsub(":", "\t", readLines(unzip(d1, "ml-10m100K/ratings.dat"))),
+ col.names = c("userId", "movi ..." ... [TRUNCATED])
```

TRAINING AND TESTING DATA

```
C:/Users/sansr/Desktop/harvardX/movielens - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help

movielenscode.R x movielensmarkdown.Rmd x
Source on Save
29 removed <- anti_join(temp, validation)
30 edx <- rbind(edx, removed)
31 rm(d1, ratings, movies, test_index, temp, movielens, removed)
32
33 ##Training and Testing dataset
34 set.seed(1)
35 train_index <- createDataPartition(y = edx$rating, times = 1, p = 0.8, list = FALSE)
36 train_set <- edx[train_index,]
37 temp <- edx[-train_index,]
38 test_set <- temp %>%
39   semi_join(train_set, by = "movieId") %>%
40   semi_join(train_set, by = "userId")
41 removed <- anti_join(temp, test_set)
42 train_set <- rbind(train_set, removed)
43 rm(temp, removed)
44
```

The test and training datasets are derived using edx set: 80% sample for training, and 20% sample for testing.

IV. Summarize Dataset

The following ways are used to look at the raw data from different perspectives: shape, size, type, general layout.

```
16423 (Top Level)
Console C:/Users/sansr/Desktop/harvardX/movielens/

> ##Summarize Dataset
>
> summary(edx)
  userId      movieId      rating      timestamp      title
Min.   : 1      Min.   : 1      Min.   :0.500      Min.   :7.897e+08      Length:9000055
1st Qu.:18124    1st Qu.: 648    1st Qu.:3.000    1st Qu.:9.468e+08      Class :character
Median :35738    Median : 1834    Median :4.000    Median :1.035e+09      Mode  :character
Mean   :35870    Mean   : 4122    Mean   :3.512    Mean   :1.033e+09
3rd Qu.:53607    3rd Qu.: 3626    3rd Qu.:4.000    3rd Qu.:1.127e+09
Max.   :71567    Max.   :65133    Max.   :5.000    Max.   :1.231e+09
 genres
Length:9000055
Class :character
Mode  :character
```

```

164:23 (top level)
Console C:/Users/sansr/Desktop/harvardX/movielens/

> str(edx)
'data.frame': 9000055 obs. of 6 variables:
 $ userId : int 1 1 1 1 1 1 1 1 1 ...
 $ movieId : num 122 185 292 316 329 355 362 364 370 ...
 $ rating : num 5 5 5 5 5 5 5 5 5 ...
 $ timestamp: int 838985046 838983525 838983421 838983392 838983392 838984474 838983653 838984885 838983707 838984596 ...
 $ title : chr "Boomerang (1992)" "Net, The (1995)" "outbreak (1995)" "Stargate (1994)" ...
 $ genres : chr "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|Adventure|Sci-Fi" ...

> dim(edx)
[1] 9000055 6

> summary(validation)

```

```

> summary(validation)
      userId      movieId      rating      timestamp      title
Min.   : 1      Min.   : 1      Min.   :0.500      Min.   :7.897e+08      Length:999999
1st Qu.:18096    1st Qu.: 648    1st Qu.:3.000    1st Qu.:9.467e+08      Class :character
Median :35768    Median : 1827    Median :4.000    Median :1.035e+09      Mode  :character
Mean   :35870    Mean   : 4108    Mean   :3.512    Mean   :1.033e+09
3rd Qu.:53621    3rd Qu.: 3624    3rd Qu.:4.000    3rd Qu.:1.127e+09
Max.   :71567    Max.   :65133    Max.   :5.000    Max.   :1.231e+09
      genres
Length:999999
Class :character
Mode :character

```

```

Console C:/Users/sansr/Desktop/harvardX/movielens/

> str(validation)
'data.frame': 999999 obs. of 6 variables:
 $ userId : int 1 1 1 2 2 2 3 3 3 4 4 ...
 $ movieId : num 231 480 586 151 858 ...
 $ rating : num 5 5 5 3 2 3 3 3 5 4 5 5 3 ...
 $ timestamp: int 838983392 838983653 838984068 868246450 868245645 868245920 1136075494 1133571200 844416936 844417070 ...
 $ title : chr "Dumb & Dumber (1994)" "Jurassic Park (1993)" "Home Alone (1990)" "Rob Roy (1995)" ...
 $ genres : chr "Comedy" "Action|Adventure|Sci-Fi|Thriller" "Children|Comedy" "Action|Drama|Romance|War" ...

> dim(validation)
[1] 999999 6

```

Displaying Top 10 Genres

```

> edx %>% separate_rows(genres, sep = "\\|")%>%
+   group_by(genres) %>%
+   summarize(count = n()) %>%
+   arrange(desc(count))
# A tibble: 800 x 2
   genres                                count
  <chr>                                <int>
1 Drama                                733296
2 Comedy                              700889
3 Comedy|Romance                      365468
4 Comedy|Drama                       323637
5 Comedy|Drama|Romance                261425
6 Drama|Romance                      259355
7 Action|Adventure|Sci-Fi            219938
8 Action|Adventure|Thriller          149091
9 Drama|Thriller                     145373
10 Crime|Drama                       137387
# ... with 790 more rows

```

Displaying Top 10 Movies

```
> edx %>% group_by(movieId, title)%>%
+   summarize(count = n()) %>%
+   arrange(desc(count))
# A tibble: 10,677 x 3
# Groups:   movieId [10,677]
  movieId title count
  <dbl> <chr> <int>
1 296 Pulp Fiction (1994) 31362
2 356 Forrest Gump (1994) 31079
3 593 Silence of the Lambs, The (1991) 30382
4 480 Jurassic Park (1993) 29360
5 318 Shawshank Redemption, The (1994) 28015
6 110 Braveheart (1995) 26212
7 457 Fugitive, The (1993) 25998
8 589 Terminator 2: Judgment Day (1991) 25984
9 260 Star Wars: Episode IV - A New Hope (a.k.a. Star wars) (1977) 25672
10 150 Apollo 13 (1995) 24284
# ... with 10,667 more rows
```

Displaying top 10 Movies By rating

```
> edx %>% group_by(rating, title)%>%
+   summarize(count = n()) %>%
+   arrange(desc(count))
# A tibble: 88,248 x 3
# Groups:   rating [10]
  rating title count
  <dbl> <chr> <int>
1 5 Shawshank Redemption, The (1994) 14769
2 5 Pulp Fiction (1994) 13441
3 5 Silence of the Lambs, The (1991) 11805
4 5 Schindler's List (1993) 11533
5 5 Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977) 11276
6 4 Fugitive, The (1993) 10948
7 5 Forrest Gump (1994) 10466
8 3 Batman (1989) 10399
9 4 Silence of the Lambs, The (1991) 10289
10 5 Usual Suspects, The (1995) 10088
# ... with 88,238 more rows
```

V. Visualize Dataset

Data visualization is most efficient, the fastest and most useful way to summarize and learn about the data.

Visualization refers to creating charts and plots from the raw data.

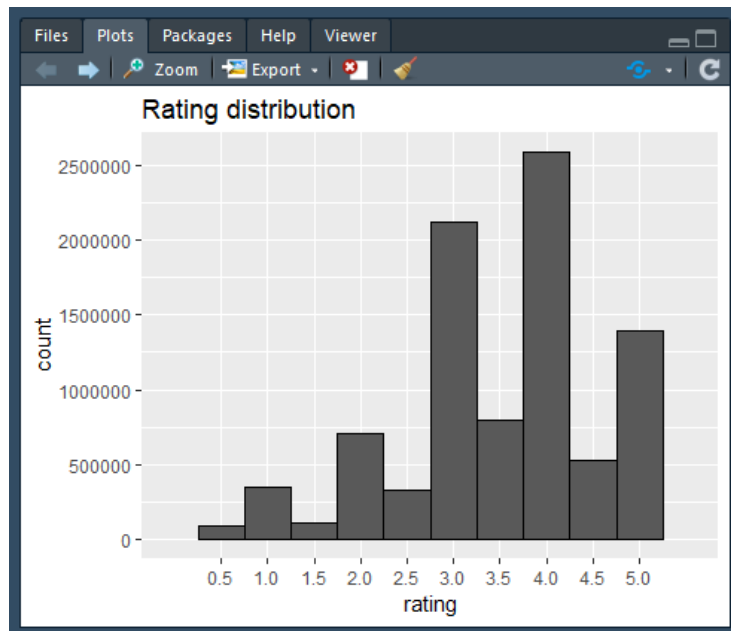
Plots of the distribution or spread of attributes can help spot outliers or invalid data.

Rating Distribution---Users give full-star ratings more frequently than half-star ratings.

CODE:

```
68
69   edx %>%
70     ggplot(aes(rating)) +
71     geom_histogram(binwidth = 0.5, color = "black") +
72     scale_x_discrete(limits = c(seq(0.5, 5, 0.5))) +
73     scale_y_continuous(breaks = c(seq(0, 3000000, 500000))) +
74     ggtitle("Rating distribution")
75
```

VISUALISED DATA:

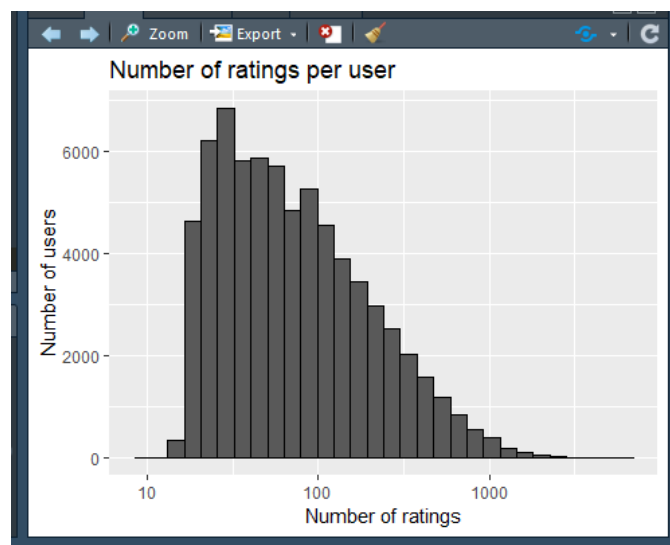


Numbers of ratings per user--- A lot of users rate hundreds of movies.

CODE:

```
75  
76   edx %>% count(userId) %>%  
77   ggplot(aes(n)) +  
78   geom_histogram(bins = 30, color = "black") +  
79   scale_x_log10() +  
80   xlab("Number of ratings") +  
81   ylab("Number of users") +  
82   ggtitle("Number of ratings per user")  
83
```

VISUALISED DATA:

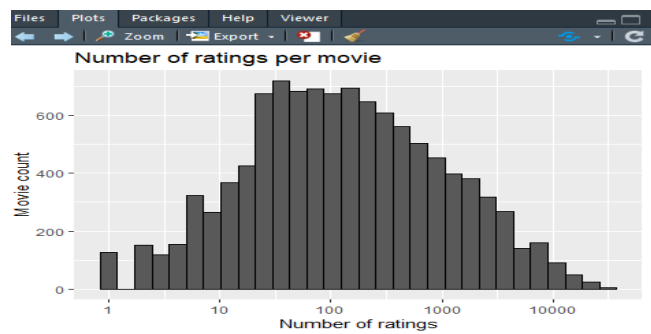


Number of ratings per movie--- Most movies were rated several times. Hundreds and even thousands.

CODE:

```
83
84   edx %>%
85     count(movieId) %>%
86     ggplot(aes(n)) +
87     geom_histogram(bins = 30, color = "black") +
88     xlab("Number of ratings") +
89     ylab("Movie count") +
90     scale_x_log10() +
91     ggtitle("Number of ratings per movie")
92
```

VISUALISED DATA:

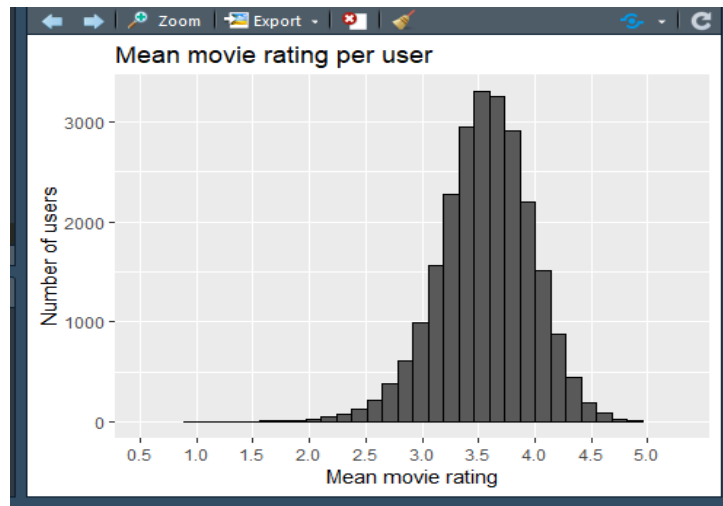


Mean Movie Rating Per User--- After shortlisting those users that have rated at least 100 movies, it is found that most users gave ratings of 3.0, 3.5 and 4.0.

CODE:

```
92
93   edx %>%
94     group_by(userId) %>%
95     filter(n() >= 100) %>%
96     summarise(mean_rating = mean(rating)) %>%
97     ggplot(aes(mean_rating)) +
98     geom_histogram(bins = 30, color = "black") +
99     xlab("Mean movie rating") +
100    ylab("Number of users") +
101    ggtitle("Mean movie rating per user") +
102    scale_x_discrete(limits = c(seq(0.5,5,0.5)))
103
```

VISUALISED DATA:



VI. Evaluate Algorithm

The following RMSE function is used to assess three model algorithms in this section.

```

103
104 ## Evaluate Algorithm
105
106 RMSE <- function(true_ratings, predicted_ratings){
107   sqrt(mean((true_ratings - predicted_ratings)^2))
108 }
109

```

MODEL 1: SIMPLE AVERAGE MODEL

(Reported RMSE- 1.059735)

(Improves in the next model)

The 1st model predicts rating using the dataset's mean rating, and all differences in movie ratings are explained by random variation. Following is the equation used for the calculation:

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

CODE:


```

109
110 ##1st model: simple Average Model
111
112 mu_hat <- mean(train_set$rating)
113 model_1_rmse <- RMSE(test_set$rating, mu_hat)
114 rmse_results <- data_frame(Model = "Simple Average", RMSE = model_1_rmse)
115 rmse_results%>%knitr::kable()

```

Model	RMSE
Simple Average	1.059735

MODEL 2: MOVIE EFFECT MODEL

(Reported RMSE- 0.943203)

(Improves in the next model)

Using the average mean rating of all movies may not be appropriate as popular movies are likely rated more than unpopular movies. Hence, in order to improve prediction, the average mean rating of each movie is compared to the average mean rating, and the estimation deviation and the resulting variables ("b" or bias) are used to predict using the following equation:

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

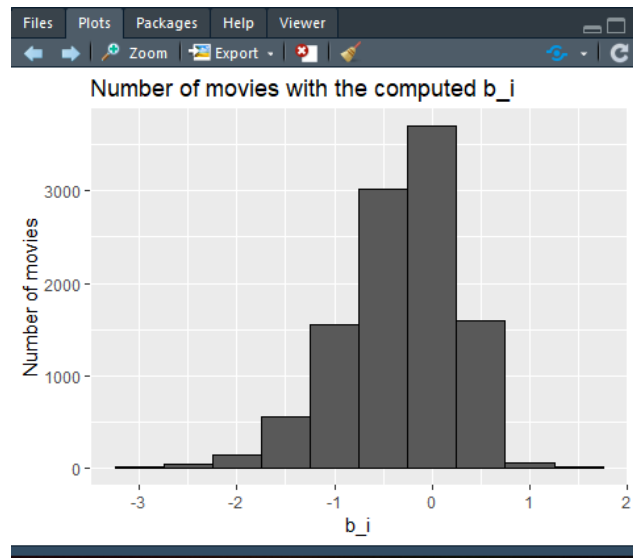
CODE:

```

116
117 ##2nd model: Movie_Effect Model
118
119 mu <- mean(train_set$rating)
120 movie_avgs <- train_set %>%
121   group_by(movieId) %>%
122   summarize(b_i = mean(rating - mu))
123 movie_avgs %>% qplot(b_i, geom = "histogram", bins = 10, data = ., color = I("black"),
124   ylab = "Number of movies", main = "Number of movies with the computed b_i")
125
126 predicted_ratings <- mu + test_set %>%
127   left_join(movie_avgs, by='movieId') %>%
128   .$b_i
129 model_2_rmse <- RMSE(predicted_ratings, test_set$rating)
130 rmse_results <- bind_rows(rmse_results,
131   data_frame(Model="Movie_Effect",
132     RMSE = model_2_rmse ))
133 rmse_results %>% knitr::kable()
134

```

VISUALISED DATA:



The above histogram shows that the rating data skew to the left, which is a result of a lower boundary in a dataset, suggesting that most ratings are higher than the mean rating of all movies.

Model	RMSE
Simple Average	1.059735
Movie_Effect	0.943203

The above result shows that in the 2nd model, the RMSE improves.

MODEL 3: Movie+User Effect Model (Reported RMSE- 0.8426298)

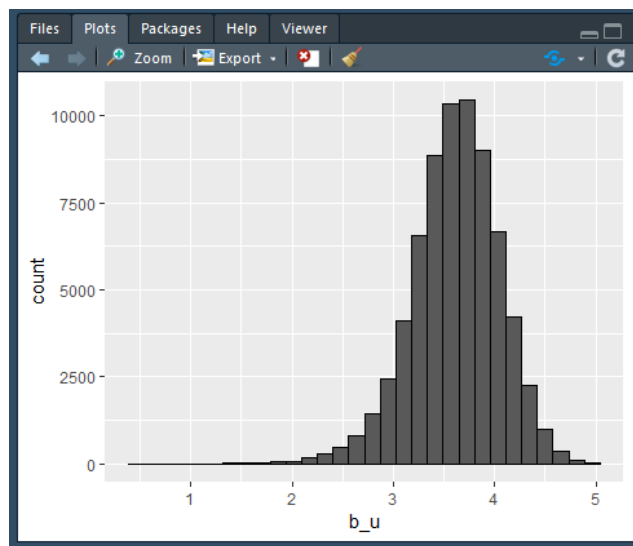
It is found that there is substantial variability across users: some users rate many movies while others are selective. Hence, the average rating for user μ is only computed for those that have rated over 100 movies, and the following equation is used for the prediction:

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

CODE:

```
134
135 ##3rd model: Movie+User_Effect Model
136
137 train_set %>%
138   group_by(userId) %>%
139   summarize(b_u = mean(rating)) %>%
140   filter(n()>=100) %>%
141   ggplot(aes(b_u)) +
142     geom_histogram(bins = 30, color = "black")
143
144 user_avgs <- test_set %>%
145   left_join(movie_avgs, by='movieId') %>%
146   group_by(userId) %>%
147   summarize(b_u = mean(rating - mu - b_i))
148
149 predicted_ratings <- test_set %>%
150   left_join(movie_avgs, by='movieId') %>%
151   left_join(user_avgs, by='userId') %>%
152   mutate(pred = mu + b_i + b_u) %>%
153   .$pred
154 model_3_rmse <- RMSE(predicted_ratings, test_set$rating)
155 rmse_results <- bind_rows(rmse_results,
156                           data_frame(Model="Movie + User_Effect",
157                                       RMSE = model_3_rmse ))
158 rmse_results %>% knitr::kable()
159
```

VISUALISED DATA:



The above histogram shows that the rating data are more normally distributed compared to the 2nd Model, suggesting that the 3rd Model may produce more reliable results than the 2nd Model.

Model	RMSE
Simple Average	1.0597347
Movie_Effect	0.9432030
Movie + User_Effect	0.8426298

It is shown that the RMSE is further reduced using the 3rd model.

VII. Evaluate validation set

Based on the results from the preceding section, the best resulting model, “Movie + User_Effect Model”, is ran directly on the validation set to predict the movie ratings. It is found that the **RMSE of the validation set is 0.8294.**

```
159
160 ## Evaluate validation set
161
162 user_avgs_validation <- validation %>%
163   left_join(movie_avgs, by='movieId') %>%
164   group_by(userId) %>%
165   summarize(b_u = mean(rating - mu - b_i))
166 predicted_ratings <- validation %>%
167   left_join(movie_avgs, by='movieId') %>%
168   left_join(user_avgs_validation, by='userId') %>%
169   mutate(pred = mu + b_i + b_u) %>%
170   .$pred
171 model_rmse_validation <- RMSE(predicted_ratings, validation$rating)
172 model_rmse_validation
173
```

```
> model_rmse_validation
[1] 0.8294231
```

VIII. Conclusion

In this project, three models (“Simple Average”, “Movie_Effect” and “Movie+User_Effect”) are developed and used to predict movie rating, and their accuracy is assessed using their resulting RMSE. The best resulting model, “Movie + User Effects Model” with RMSE of 0.8426, is ran directly on the validation set to predict the movie ratings. The RMSE result on validation dataset of 0.8294 is lower than the best results on test dataset of 0.8426 (3rd model), suggesting that the “Movie+User_Effect” model is likely a reliable prediction model.

