Capstone Project

Movie Lens project

Ashraf Fashafsheh

February, 2022

Abstract.

This report of the Movie lens project Submitted in partial fulfillment of the requirements of the course Harvard X: PH125.9x Data Science:

In this project, we will predict ratings using Movie lens dataset.

Dataset:

The data set is the Movie lens dataset that we obtained from the https://grouplens.org/datasets/movielens/10m/, its contains 9000055 observations' and 6 variables.

```
dim(edx)
```

```
[1] 9000055      6
```

The data set consists of the following **features**: "userId" "movieId" "rating" "timestamp" "title" "genres".

```
names(edx)
```

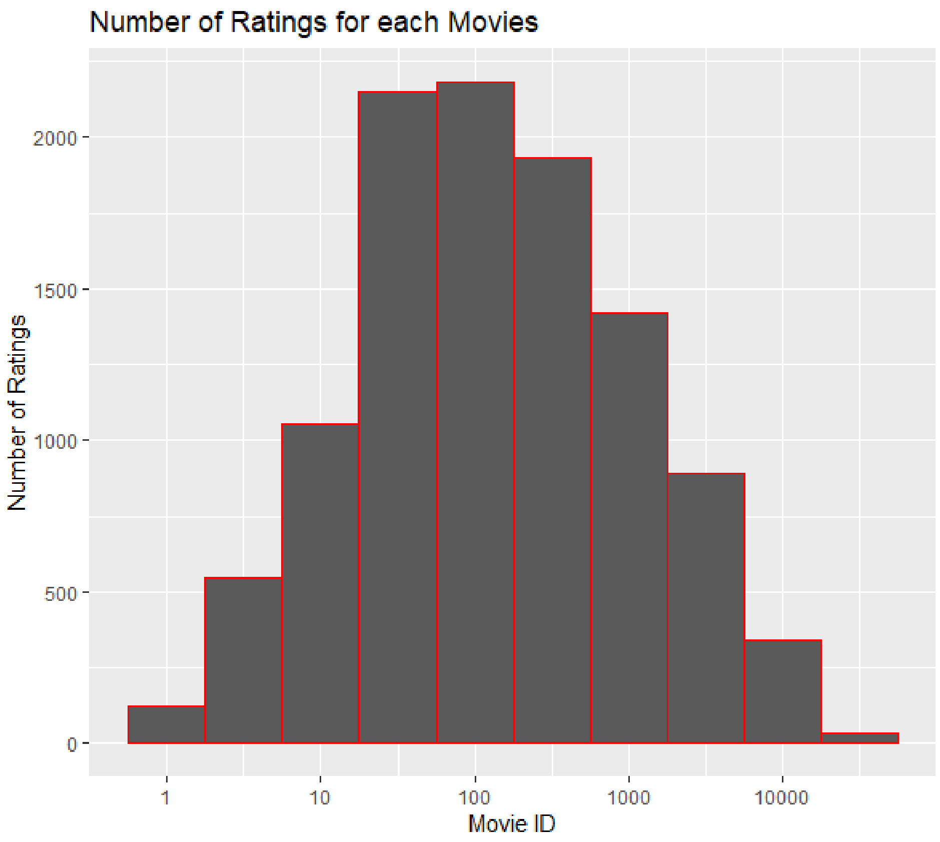
```
[1] "userId"    "movieId"   "rating"    "timestamp" "title"     "genres"
```

Data Loading

the data set is loaded using the code provided from our course, which split the data into edx set and 10% validation set. The edx set will be split into training and test set, and validation set will be used to final evaluation.

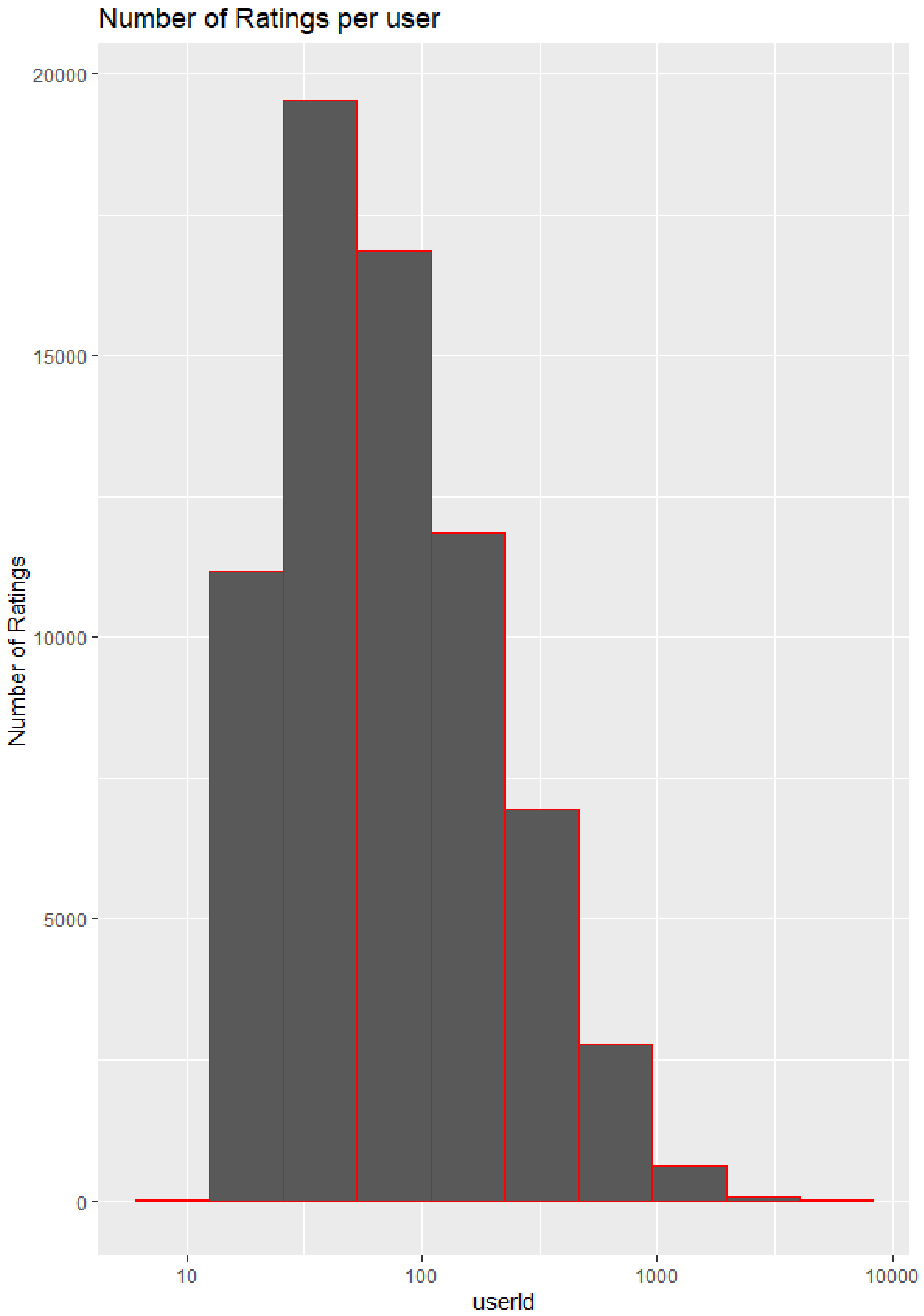
Data Summary and Exploratory Data Analysis

We will analyze and visualize each predictor to gain insight on how the ratings are distributed. This analyze will help us to make better predictions.



```
str(edx)
Classes ‘data.table’ and 'data.frame':9000055 obs. of  6 variables:
 $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ movieId  : num  122 185 292 316 329 355 356 362 364 370 ...
 $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
 $ timestamp: int  838985046 838983525 838983421 838983392 838983392 83
8984474 838983653 838984885 838983707 838984596 ...
 $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995
)" "Stargate (1994)" ...
 $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Dra
ma|Sci-Fi|Thriller" "Action|Adventure|Sci-Fi" ...
 - attr(*, ".internal.selfref")=<externalptr>
```

From result we have sex variables.



We will divided Our dataset into two subset 70% for training data and 30% for testing data, then we will build our model.

```
test_index <- createDataPartition(y = edx$rating, times = 1, p = 0.3, list = FALSE)

train_set <- edx[-test_index,]

test_set <- edx[test_index,]
```

The rating distribution is normally distributed in the predictors mostly, so we applied Central Limit Theorem so we can use the Linear Regression approach to make the predictions. We will use train dataset to build our linear model.

The formula for our linear model is Yu,i=μ+εu,i

Where Yu,i is the rating prediction

μ is the true rating

and εu,i is the independent errors sampled from the same distribution centered at zero.

Assuming all users give the same rating to all movies, μ+εu,i is the average of the ratings.

We can get the average rating using the following code;

```
Mu1
[1] 3.512596
```

So the RMSE for the average rating prediction as follows;

```
RMSE <- RMSE( Mu1,test_set$rating)
> RMSE
[1] 1.060597
```

In the first model we found that RMSE was 1.060597(as the average).

2nd Models

We will calculate RSME by adding the term bi to present the average rank of the movie as shown in the following equation is

$$Yu,i = \mu + bi$$

After run following codes

```
Mu2 <- mean(train_set$rating)

avg_movie <- train_set %>%

  group_by(movieId) %>%

  summarize(b_i = mean(rating - Mu2))

avg_movie

predicted_ratings <- Mu2 + test_set %>%

  left_join(avg_movie, by='movieId') %>%

  pull(b_i)

RMSE2 <- RMSE(predicted_ratings, test_set$rating)

RMSE2
```

```
RMSE2
[1] 0.9443758
```

We can see that since rating distribution in Movies and Users predictors are approximately normal, their bias improves the RMSE significantly.

Third models

In this model we adding movie average to determine movie bias bi , on each user's average to determine user bias bu, and on each genre "combo" average to determine genre bias bg, as the following equation is:

$$\mathbf{Yu,i = \mu + bi + bu + bg}$$

And after ran te following codes :

```
user_avgs <- train_set %>%
  left_join(avg_movie, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating - Mu2 - b_i))
user_avgs
```

#Now let's see how if RMSE improved with time

```
predicted_ratings <- test_set %>%
  left_join(avg_movie, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  mutate(pred = Mu2 + b_i + b_u) %>%
  .$pred
RMSE3 <- RMSE(predicted_ratings, test_set$rating)
```

We obtained te following result for third model

```
RMSE3
[1] 0.8673029
```

From this result is their bias improves the RMSE significantly.

Finally, I will train the algorithm using the full edx dataset and then to predict ratings within the validation dataset. prior to doing this it was necessary to incorporate the date of review and year of release variables in the validation set using the mutate function from dplyr package.

We will apply validation dataset to see the effect on the RMSE. I used following code

```
valid_pred_rating <- validation %>%

  left_join(avg_movie, by = "movieId" ) %>%

  left_join(user_avgs , by = "userId") %>%

  mutate(pred = Mu2 + b_i + b_u) %>%

  pull(pred)

RMSE4<- RMSE(valid_pred_rating,validation$rating)

RMSE4
```

We found a following result that is the best of the previous models

```
RMSE4
[1] 0.8665917
```

Results

RMSE	RMSE1	RMSE2	RMSE3	RMSE4
value	1.060597	0.9443758	0.8673029	0.8665917
Method	Just average	effect by Movie	effect by Movie + User	Validation

Conclusions

we built four model were used in our algorithm included: (1) the average rating of each movie, (ii) the specific-effect by Movie, (iii) the specific-effect by Movie + User. The final RMSE from our algorithm is 0.8665917. This result achieved our report have highest impact to the results (decrease the value of RMSE and increase accuracy**.**