MovieLens Recommendation System Project

HarvardX Data Science Professional Certificate: PH125.9x Capstone 1

Kanin Limniyakul

3/31/2022

Contents

Executive Summary **4**

Exploratory Data Analysis (EDA) **4**

General Rating Information 4

General Movies information 6

General User information 7

General Genres Information 8

Methodology and Analysis **11**

Data partitioning of edx dataset 11

Residual Mean Square Error(RMSE) 11

Simple Model 11

Add Movie Effect 12

Add User Effect 12

Add Genres Effect 12

Regularization 13

Matrix Factorization **14**

Model Validation **15**

Conclusion **16**

List of Figures

1 Rating Distribution 5

2 Histrogram of ratings for each movies 6

3 Rating vs Release Date 10

List of Tables

1 Summary of edx dataset 4

2 Rating Summary 4

3 Top 5 rated movies 6

4 Top 10 most reviewed genres 8

5 RSME Results - Simple Average 11

6 RSME Results - Add Movie Effect 12

7 RSME Results - Add user effect 12

8 RSME Results - Add genres effect 13
9 RSME Results - Regularised 14
10 RSME Results - Matrix Factorization 15
11 RSME validation 15
12 RSME validation with Matrix Factorization 16

Executive Summary

This is the a part of HarvardX professional certificate in Data Science capstone project.The target of the project is to predict movie ratings from 10 million MovieLens dataset which trying to minimize the root mean square estimate (RMSE) of the validation set as much as possible. The report started from performing the exploratory analysis (EDA) to see the bias effects on each parameters including movies, users, genres and released year.

Then we built a linear model from the naive model (simple average all the rating) then add each effect parameters except released year (it's has a little effect on the ratings). The training set RSME is gradually reduced from 1.0601524 to 0.8650721. After that the regularized method was used to control the variance of effects size so the RSME is finally reduced to 0.8650721.

Matrix Factorization is another methodology used in this report, the basic concept is to reshape the matrix size of 9,000,055 x 6 to be 69,878 x 10,677 and use recosystem library to estimate the rating. the training RMSE is 0.7909077.

The final validation test is conducted and got the final the linear model RSME of 0.8644514 and the final matrix factorization model is 0.7830619.

Exploratory Data Analysis (EDA)

Firstly, the split movieLens 10M dataset into edx dataset and valiation set. Our edx dataset is further divided into the test set and the training set. The edx dataset comprises of 9,000,055 rows, 6 variables, 69,878 unique users and 10,677 unique movies which means that not all users rated all the movies.

Table 1: Summary of edx dataset

np_of_rows	no_of_column	no_of_users	no_of_movies	avg_rating	no_of_genres
9000055	6	69878	10677	3.51	797

General Rating Information

The movies are rated in the scale of 0 to 5 with the incremental of 0.5. From the below histogram and the table, rating 4 is the majority of the dataset (2,588,430) and average rating of all the movies is 3.512 (red dashed-line).

Table 2: Rating Summary

rating	n()
0.5	85374
1.0	345679
1.5	106426
2.0	711422
2.5	333010
3.0	2121240
3.5	791624
4.0	2588430
4.5	526736
5.0	1390114

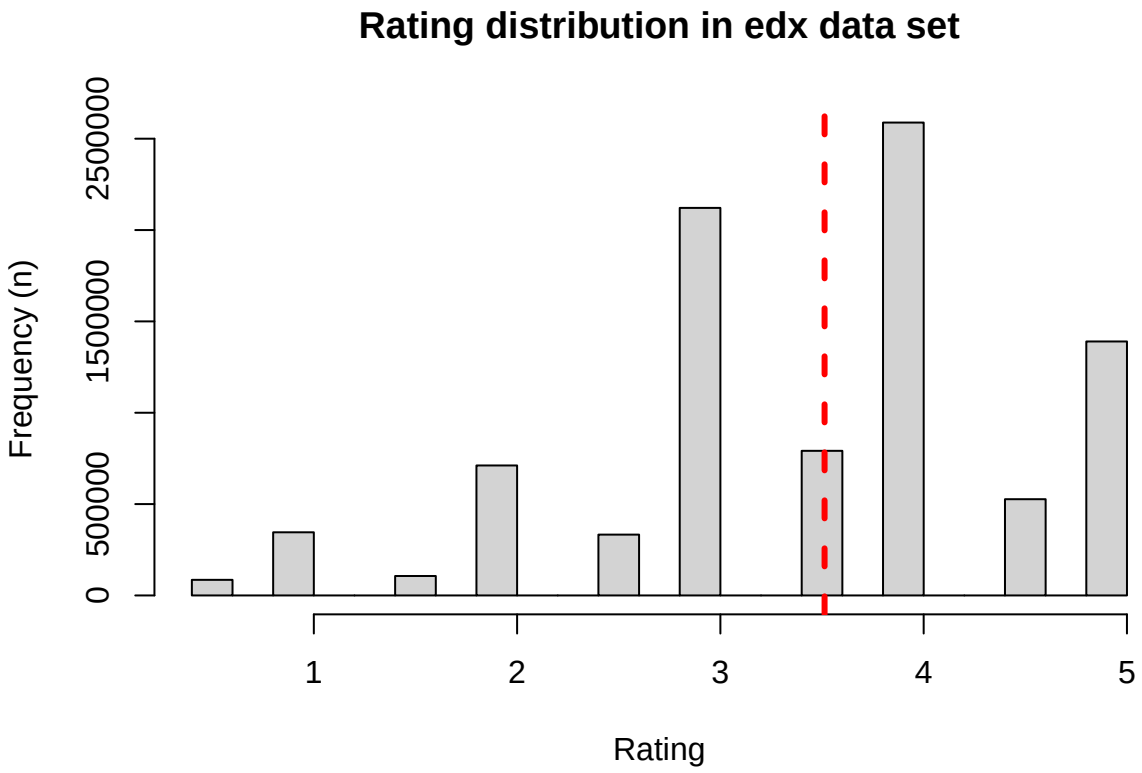


Figure 1: Rating Distribution

General Movies information

There are 10,677 movies in the edx dataset. It is well understandably that some movies have got more numbers of review more than others which could be depending on the popularity. We can see the distribution of no of ratings as below histogram. Pulp fiction (1994) got the highest no of rating of 31,326 and there are 126 movies got 1 rating only. We can see the top 5 number of rating as below,

```
## Selecting by n
```

Table 3: Top 5 rated movies

movieId	n
296	31362
318	28015
356	31079
480	29360
593	30382

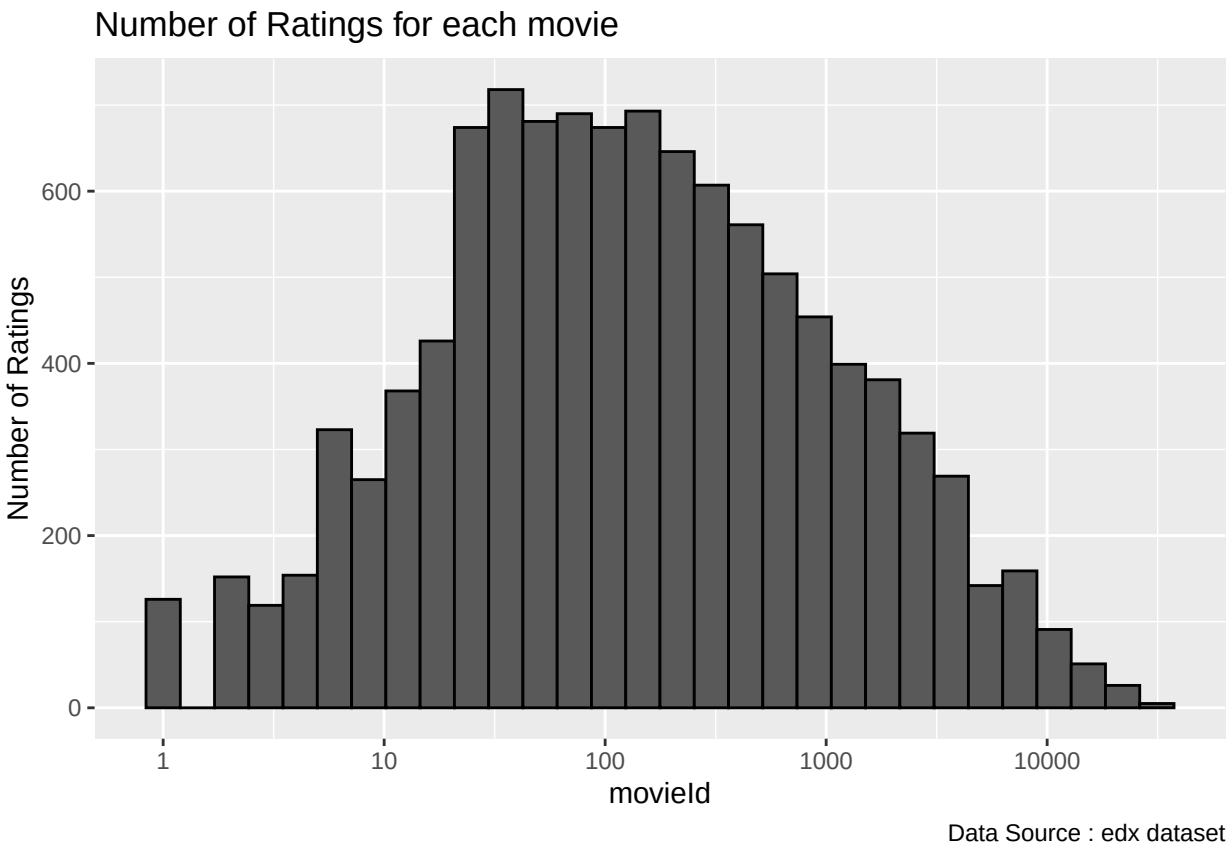
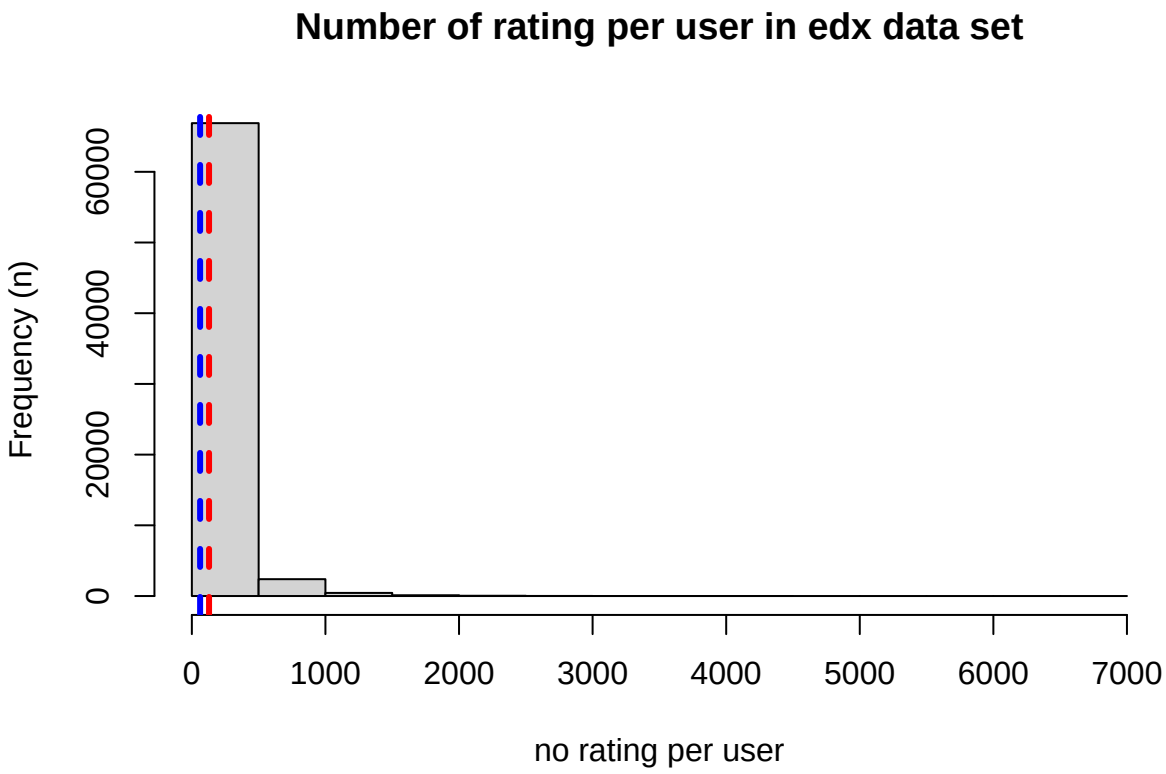
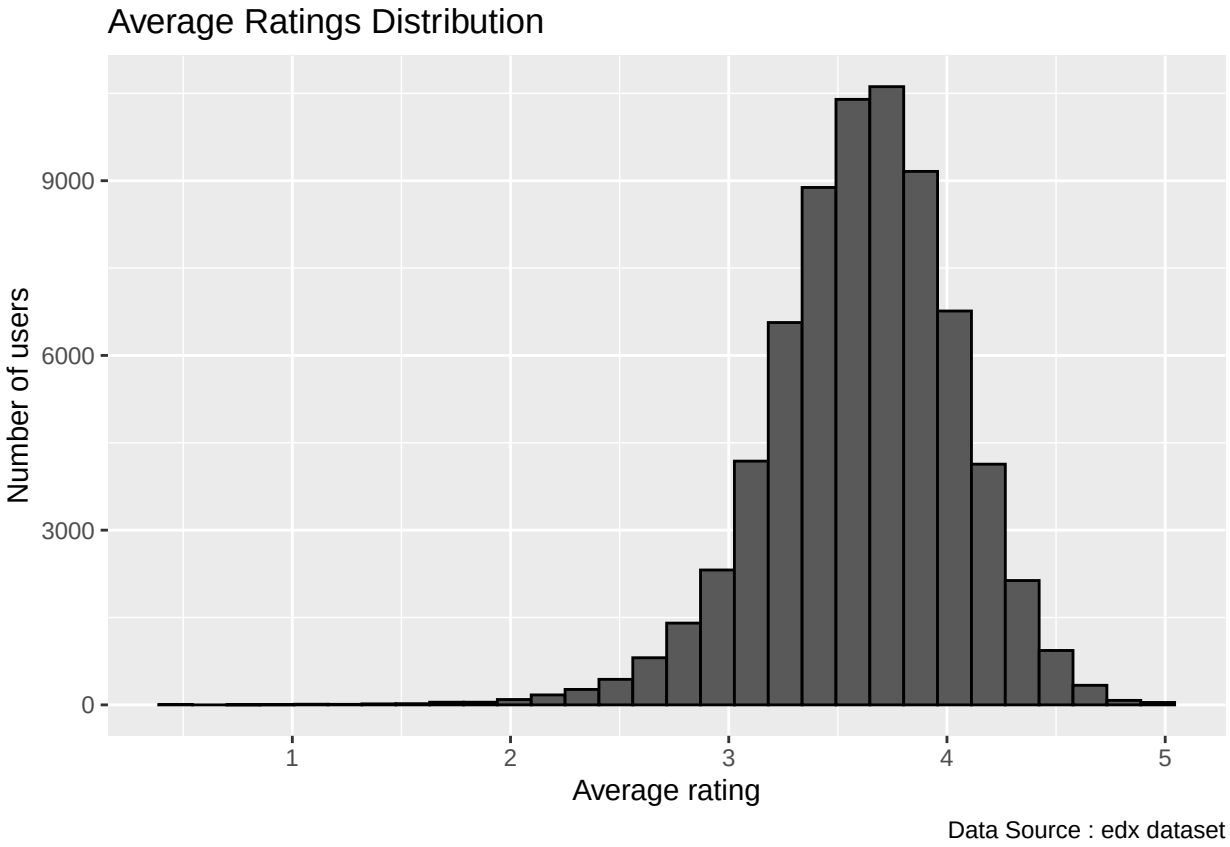


Figure 2: Histrogram of ratings for each movies

General User information

Some users are more active than others, we can see the distribution which the minimum no of rating of 10 per user, the most active user reviewed up to 66,616 movies, mean no rating is 128.8 and median of 62 times. In addition, the rating scores are varies across users as well, some users tended to give more rating compare to other users.



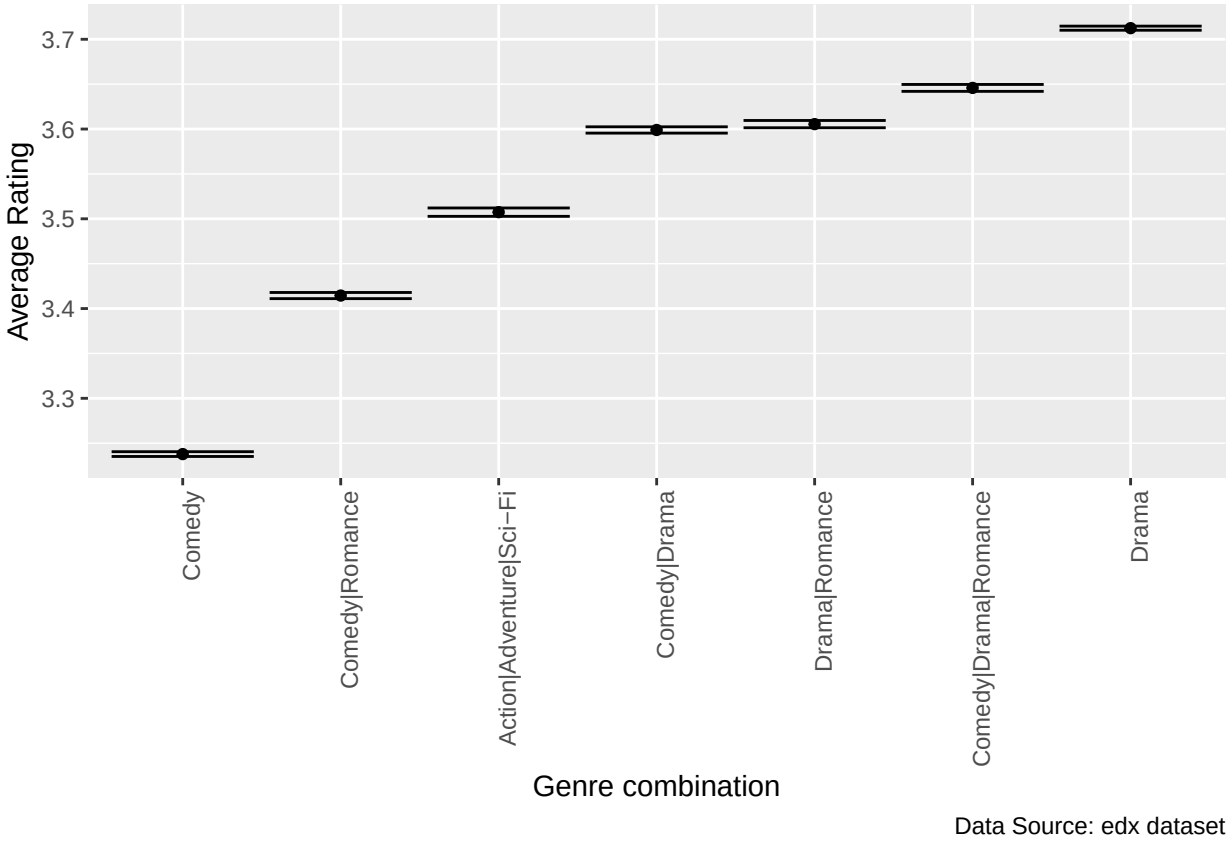


General Genres Information

From the data, it is observed that some movies genres also vary in terms of number of rating and average rating as per the table and figure below. The table shows the top 10 most reviewed genres. whereas the plot shows the rating average of the movies having more than 150,000 reviews.

Table 4: Top 10 most reviewed genres

genres	no_of_review	avg_rating
Action|Adventure|Sci-Fi	219938	3.507407
Action|Adventure|Thriller	149091	3.434101
Comedy	700889	3.237858
Comedy|Drama	323637	3.598961
Comedy|Drama|Romance	261425	3.645824
Comedy|Romance	365468	3.414486
Crime|Drama	137387	3.947135
Drama	733296	3.712364
Drama|Romance	259355	3.605471
Drama|Thriller	145373	3.446345



General Release date information

From the plot below, there is a little effect on release date but not so strong. The average ratings were in between 3.25 to 4.25.

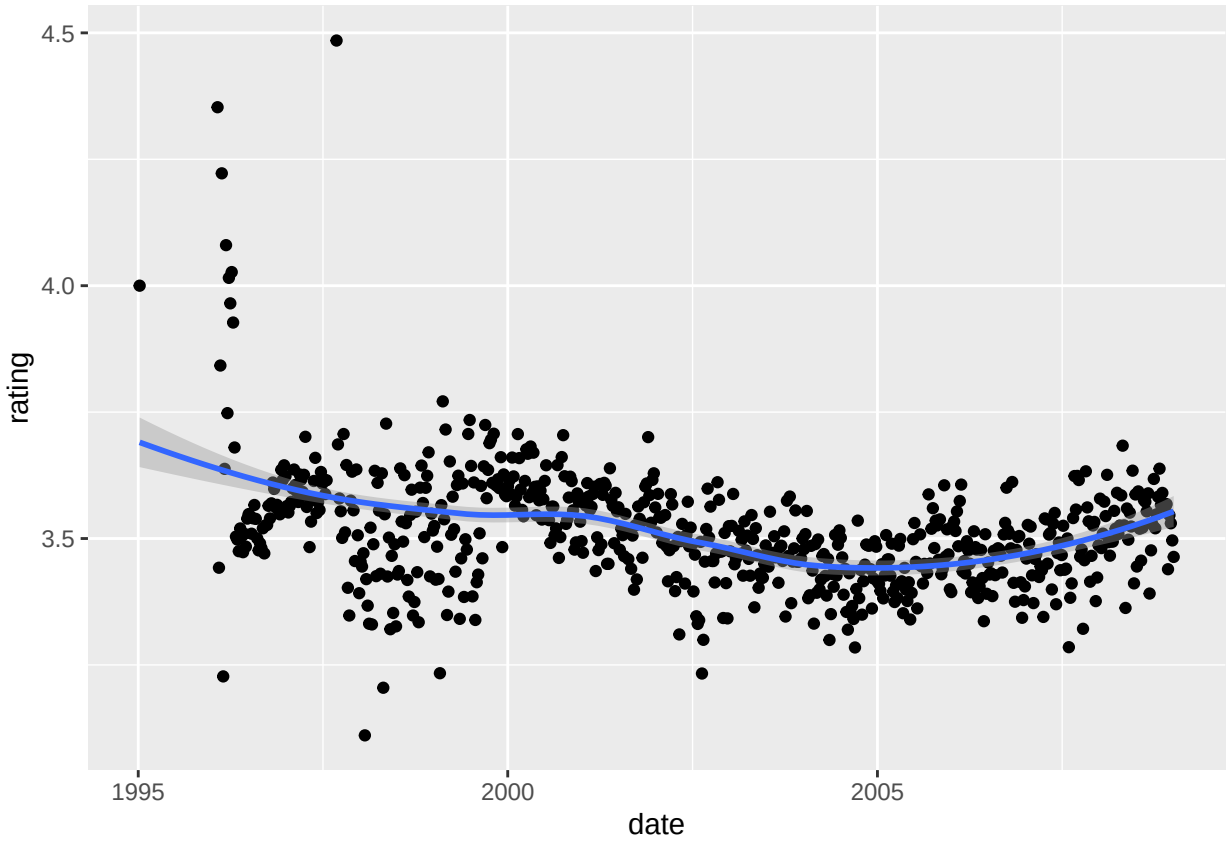


Figure 3: Rating vs Release Date

Methodology and Analysis

As we have explored the effects on parameters of movies, users, genres and release date. We will determine the bias from 3 parameters (except time whuch has a little or no effect to ratings.) and then perform regularise analysis on those effects.

Data partitioning of edx dataset

Firstly, the edx data set will be divided into the training set (80%) and the test set (20%) as we reserve the validation set for the final hold-out test set. The logic is to provide the sufficient train dataset to train the model

```
## Joining, by = c("userId", "movieId", "rating", "timestamp", "title", "genres", "date")
```

Residual Mean Square Error(RMSE)

Residual Mean Square Error(RMSE) is calculated by the standard deviation of the difference between predicted rating and the actual rating as below equation,

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

where,

$y_{u,i}$ is defined as the actual rating provided by user i for movie u $\hat{y}_{u,i}$ is the predicted rating for the same N is the total number of user/movie combinations

The goal is this project is to create a machine learing model that can achieve RSME $<$ 0.xxxx

Simple Model

The first and the very simple method is to average all the movie ratings and apply to all the movies. Not surprisingly the RSME is as high as 1.06152

The first model is shown below,

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

where, $Y_{u,i}$ is the actual rating for movie i by user u μ is the average of all ratings $\epsilon_{u,i}$ is independent errors

Table 5: RSME Results - Simple Average

method	RMSE
Average	1.060152

Add Movie Effect

As per exploratory data analysis section, movies themselves have their own biases due to the popularity, so we need to add a term to take into account from movies biases. We add b_i into the first model as below,

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

where the added b_i is movie bias

the least square estimate bi can be calculated by the average of $Y_{u,i} - \hat{\mu}$ for each movie i

$$\hat{y}_{u,i} = \hat{\mu} + \hat{b}_i$$

we can improve our RSME into 0.9435666.

Table 6: RSME Results - Add Movie Effect

method	RMSE
Average	1.0601524
Average + b_i	0.9435666

Add User Effect

Following the same logic as movie effect, the user effect (b_u) to the model

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

where the added b_u is user bias

the least square estimate b_u can be calculated by the average of

$$\hat{b}_u = mean\left(\hat{y}_{u,i} - \hat{\mu} - \hat{b}_i\right)$$

RSME is now further improved into 0.8660905

Table 7: RSME Results - Add user effect

method	RMSE
Average	1.0601524
Average + b_i	0.9435666
Average + b_i + b_u	0.8660903

Add Genres Effect

The model is further improved by add genres effect

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

where the added b_g is genres bias

the least square estimate b_g can be calculated by the average of

$$\hat{b}_g = mean\left(\hat{y}_{u,i} - \hat{\mu} - \hat{b}_i - \hat{b}_u\right)$$

RMSE is now reduced to 0.8657490.

Table 8: RSME Results - Add genres effect

method	RMSE
Average	1.0601524
Average + b_i	0.9435666
Average + b_i + b_u	0.8660903
Average + b_i + b_u + b_g	0.8657489

Regularization

Regularization is to "control" the effect sizes variability. In our dataset in the EDA section, there are some movies got rated many times where as some movies got reviewed only one. SO the equation we try to minimize is as below,

$$\frac{1}{N}\sum_{u,i}\left(y_{u,i} - \mu - b_i\right)^2 + \lambda\sum_i b_i^2$$

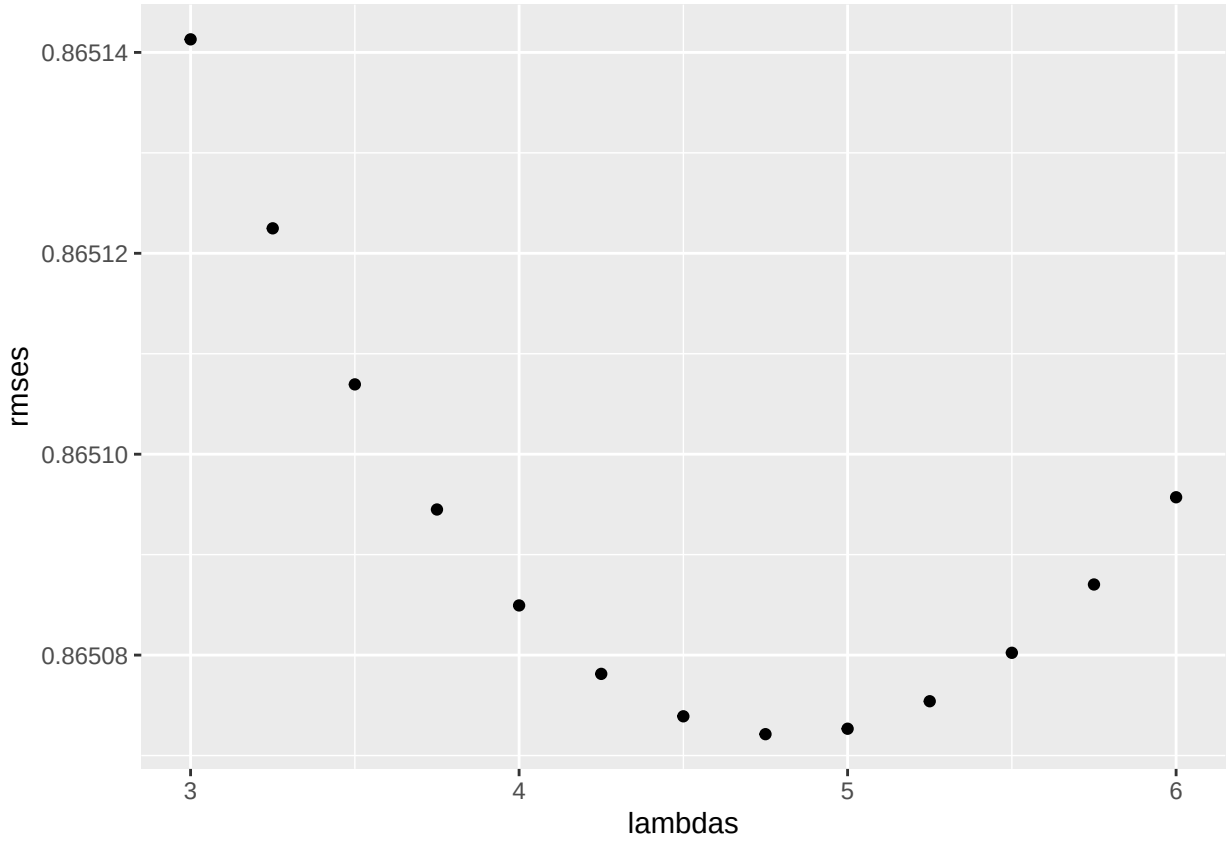
Thus, the regularized estimate of b_i is

$$\hat{b}_i\left(\lambda\right) = \frac{1}{\lambda + n_i}\sum_{u=1}^{n_i}\left(Y_{u,i} - \hat{\mu}\right)$$

The incorporated regularize to all the effects is as below,

$$\frac{1}{N}\sum_{u,i}\left(y_{u,i} - \mu - b_i - b_u - b_g\right)^2 + \lambda\left(\sum_i b_i^2 + \sum_u b_u^2 + \sum_g b_g^2\right)$$

The Lampda that yields the lowest RMSE is 4.75.




```
## [1] 4.75
```

The RMSE is now 0.8650722.

Table 9: RSME Results - Regularised

method	RMSE
Average	1.0601524
Average + b_i	0.9435666
Average + b_i + b_u	0.8660903
Average + b_i + b_u + b_g	0.8657489
Regularised average + b_i + b_u + b_g	0.8650721

Matrix Factorization

The concept of Matrix Factorization is try to reduce the large matrix dimension into the product of two matrices."recosystem" package is used to train the model, find the penalty term, exporting metrices and make prediction.

```
## Loading required package: recosystem
```

```
## Warning in set.seed(2022, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
## iter      tr_rmse          obj
##    0       0.9925   1.0019e+07
##    1       0.8802   8.0891e+06
##    2       0.8479   7.5217e+06
##    3       0.8248   7.1539e+06
##    4       0.8076   6.9011e+06
##    5       0.7941   6.7195e+06
##    6       0.7831   6.5777e+06
##    7       0.7740   6.4648e+06
##    8       0.7661   6.3736e+06
##    9       0.7593   6.2968e+06
##   10       0.7532   6.2317e+06
##   11       0.7480   6.1771e+06
##   12       0.7430   6.1281e+06
##   13       0.7386   6.0860e+06
##   14       0.7346   6.0478e+06
##   15       0.7309   6.0148e+06
##   16       0.7276   5.9851e+06
##   17       0.7245   5.9600e+06
##   18       0.7216   5.9319e+06
##   19       0.7189   5.9096e+06

##  [1] 4.938781 4.857063 5.146074 5.098394 3.709553 3.183384 3.016438 2.927537
##  [9] 3.564188 3.902695
```

Table 10: RSME Results - Matrix Factorization

method	RMSE
Average	1.0601524
Average + b_i	0.9435666
Average + b_i + b_u	0.8660903
Average + b_i + b_u + b_g	0.8657489
Regularised average + b_i + b_u + b_g	0.8650721
Matrix Factorizarion	0.7909007

Model Validation

Now the whole edx dataset is used as the training set with lampda (4.75) to test with the final validation set. The final RMSE is

Table 11: RSME validation

method	RMSE
Validate Regularised average + b_i + b_u + b_g	0.8644514

```
## Warning in set.seed(2022, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used

## iter      tr_rmse          obj
##    0       0.9715   1.2006e+07
```

```
##    1       0.8728   9.8879e+06
##    2       0.8385   9.1711e+06
##    3       0.8165   8.7506e+06
##    4       0.8013   8.4722e+06
##    5       0.7898   8.2761e+06
##    6       0.7802   8.1255e+06
##    7       0.7719   8.0055e+06
##    8       0.7649   7.9056e+06
##    9       0.7590   7.8262e+06
##   10       0.7538   7.7590e+06
##   11       0.7493   7.7025e+06
##   12       0.7452   7.6526e+06
##   13       0.7416   7.6084e+06
##   14       0.7382   7.5720e+06
##   15       0.7352   7.5361e+06
##   16       0.7324   7.5060e+06
##   17       0.7298   7.4794e+06
##   18       0.7274   7.4549e+06
##   19       0.7251   7.4315e+06

## [1] 0.7830619
```

Table 12: RSME validation with Matrix Factorization

method	RMSE
Validate Regularised average + b_i + b_u + b_g	0.8644514
Validate matrix factorization	0.7830619

Conclusion

The model using the regularized on movies, users and genres effect can achieve RMSE of 0.8644514 whereas the matrix factorization model can achieve RMSE of 0.7826499.

The limitation is the computing power of personal laptop on the very large dataset which the recommenderlab package cannot be implemented in this report. It has various algorithms to estimate the ratings.

Future works with more computing power, recommenderlab can be tried.