Report MovieLens project

Introduction

In this project we use dataset Movielens included in the dslabs package. The Movilens is composed from users (userId) that give a rating between 0 and 5 (rating) in a specific date and time (timestamp) for the movies (movieId) that have a title and a genres associated. Our goal is using the inputs in one subset to predict movie ratings in the validation set that will compared with RMSE.

Data exploration

The first step, is to see the structur of our data (training dataset)

```
head(edx)
```

```
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 4      1     292      5 838983421               Outbreak (1995)
## 5      1     316      5 838983392               Stargate (1994)
## 6      1     329      5 838983392 Star Trek: Generations (1994)
## 7      1     355      5 838984474       Flintstones, The (1994)
##                          genres
## 1                Comedy|Romance
## 2         Action|Crime|Thriller
## 4  Action|Drama|Sci-Fi|Thriller
## 5       Action|Adventure|Sci-Fi
## 6 Action|Adventure|Drama|Sci-Fi
## 7       Children|Comedy|Fantasy
```

the data set edx, is composed by 6 six variable, of two type: 1) Quantitave variable:userId (number identify the user), movieId (number identify the movie), timestamp (number that iden identify date and time), rating (valutation of ranting movies - that is a discete variable, that hava a value from 0.5 to 5) 2) Qualitative variable: title (name movie title - not unique), genres (type genres associated with the movie).

```
str(edx)
```

```
## '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 838984474 838983653 838984885 83
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|Ac
```

this is summaries of every variable

```
summary(edx)
```

```
##      userId         movieId          rating        timestamp
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08
##  1st Qu.:18124   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08
##  Median :35738   Median : 1834   Median :4.000   Median :1.035e+09
##  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
```

```
##     title              genres
##  Length:9000055     Length:9000055
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

Count votes of genre

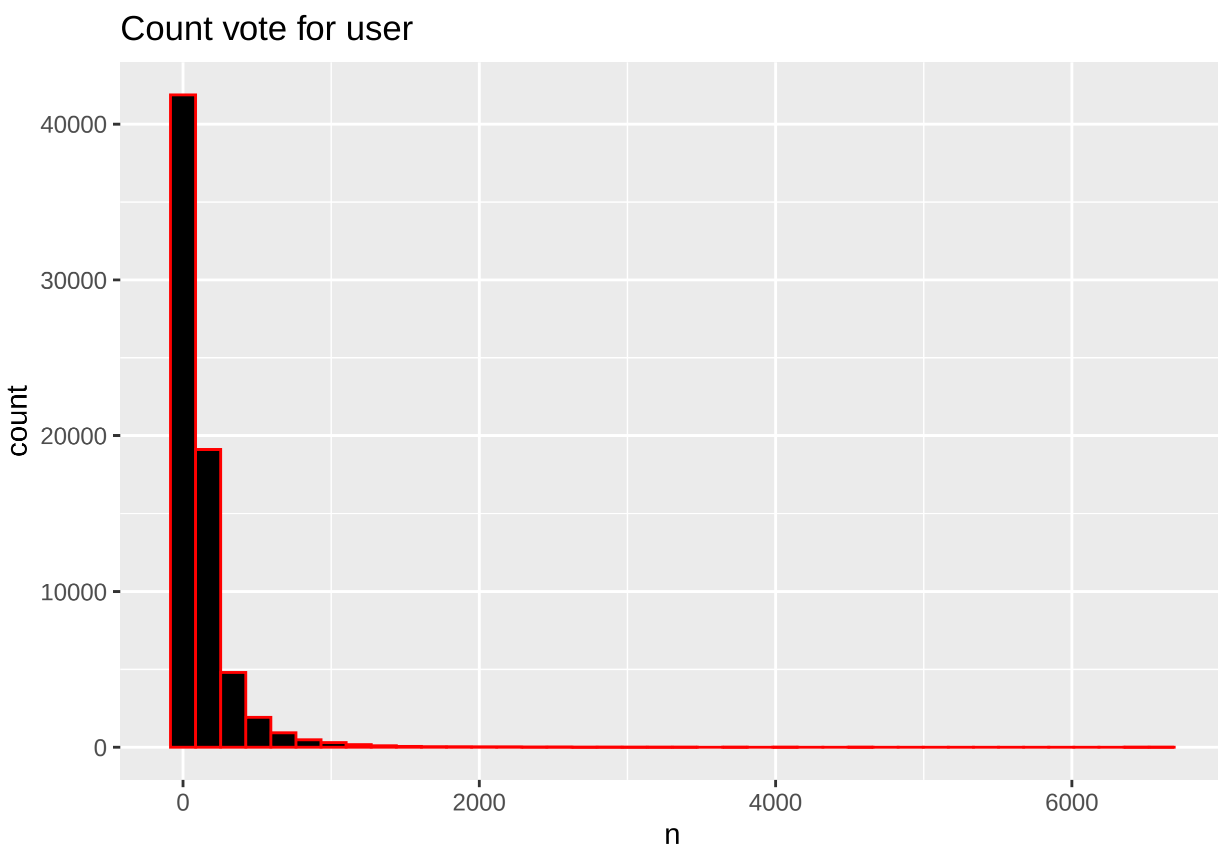
We see the top ten genres that have more review

```
## Selecting by count
```

```
## # A tibble: 10 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
```

Count vote for user

This is a histogram that rapresent the number votes give for every user.



Top 10 movies with most vote

Here we have the list of top ten movie with more movie review

```
## Selecting by count

## # A tibble: 10 x 2
##    title                                                          count
##    <chr>                                                          <int>
##  1 Pulp Fiction (1994)                                            31362
##  2 Forrest Gump (1994)                                            31079
##  3 Silence of the Lambs, The (1991)                               30382
##  4 Jurassic Park (1993)                                           29360
##  5 Shawshank Redemption, The (1994)                               28015
##  6 Braveheart (1995)                                              26212
##  7 Fugitive, The (1993)                                           25998
##  8 Terminator 2: Judgment Day (1991)                              25984
##  9 Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977) 25672
## 10 Apollo 13 (1995)                                               24284
```

Model prediction

We applicate different model to predict the rating movies and we select one that have more lower RMSE (Residual Mean Standard Error)

- In a first model we use the mean of rating for predict the rating of movies This model find the mean of training set of reating movies

```r
mu_edx <- mean(edx$rating)
mu_edx
```

```
## [1] 3.512465
```

```r
rmse_results <- data_frame(method = "Only Mean", RMSE = mu_edx)
```

```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

and the quality of model is:

```r
basic_rmse <- RMSE(validation_New$rating,mu_edx)
basic_rmse
```

```
## [1] 1.061202
```

- In a second model we applicate the penalty of the movie effect

```r
moviePenalty <- edx %>%
  group_by(movieId) %>%
  summarize(b_movie = mean(rating - mu_edx))

moviePenalty
```

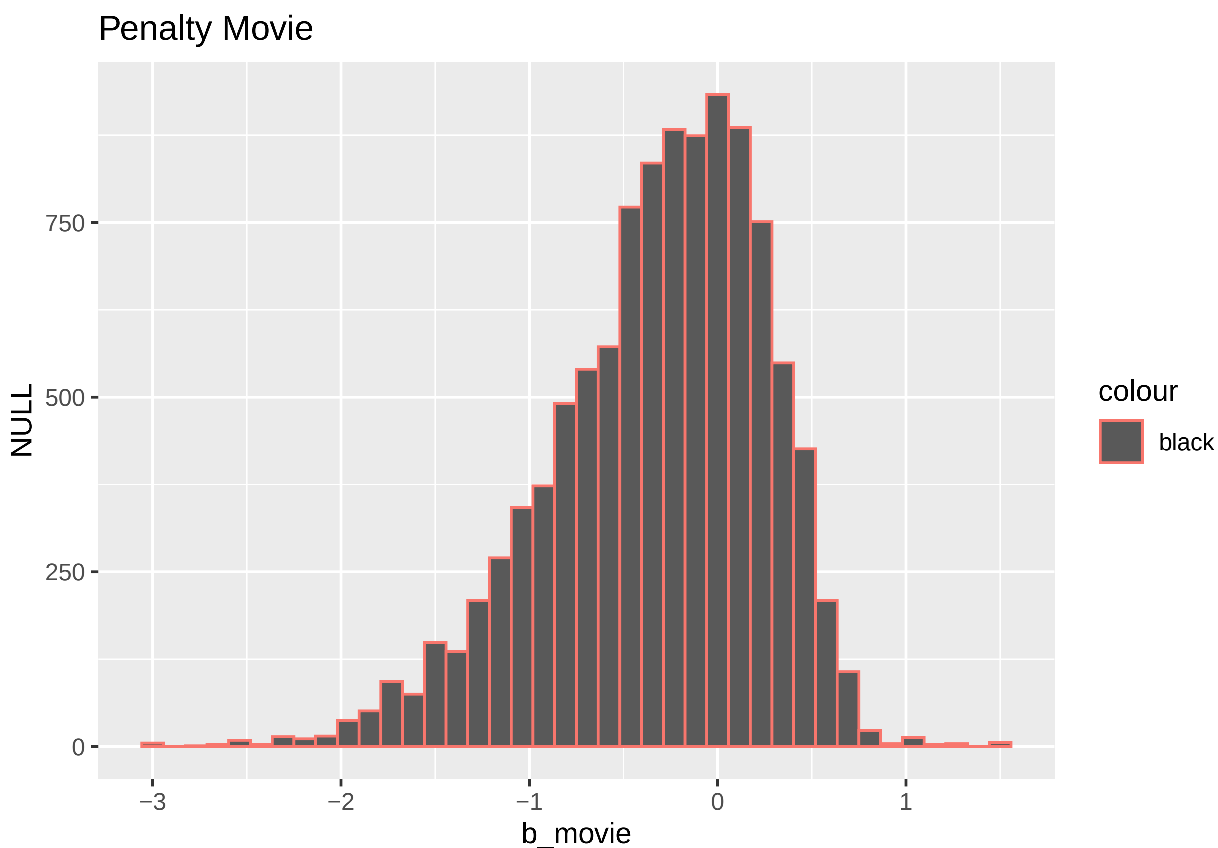
```
## # A tibble: 10,677 x 2
##    movieId b_movie
##      <dbl>   <dbl>
##  1       1  0.415
##  2       2 -0.307
##  3       3 -0.365
##  4       4 -0.648
##  5       5 -0.444
##  6       6  0.303
##  7       7 -0.154
##  8       8 -0.378
##  9       9 -0.515
## 10      10 -0.0866
## # ... with 10,667 more rows
```

and we have the quality of predict model is

```r
predict_ratings_movie<- validation %>%
  left_join(moviePenalty, by='movieId') %>%
  mutate(pred = mu_edx + b_movie)
modelMovies_rmse <- RMSE(validation_New$rating,predict_ratings_movie$pred)
rmse_results <- bind_rows(rmse_results, data_frame(method="Movie Effect Model", RMSE = modelMovies_rmse

modelMovies_rmse
```

```
## [1] 0.9439087
```




```
## # A tibble: 10,677 x 2
##    movieId b_movie
##      <dbl>   <dbl>
##  1       1  0.415
##  2       2 -0.307
##  3       3 -0.365
##  4       4 -0.648
##  5       5 -0.444
##  6       6  0.303
##  7       7 -0.154
##  8       8 -0.378
##  9       9 -0.515
## 10      10 -0.0866
## # ... with 10,667 more rows
```

- In the third model, for predict tha rating we use the penalty of movie effect (the previous model) and the penalty of users effect

Before we calculate the penalty of users

```
penaltyUser <- edx %>%
  left_join(moviePenalty, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_user = mean(rating - mu_edx - b_movie))

penaltyUser
```

```
## # A tibble: 69,878 x 2
```

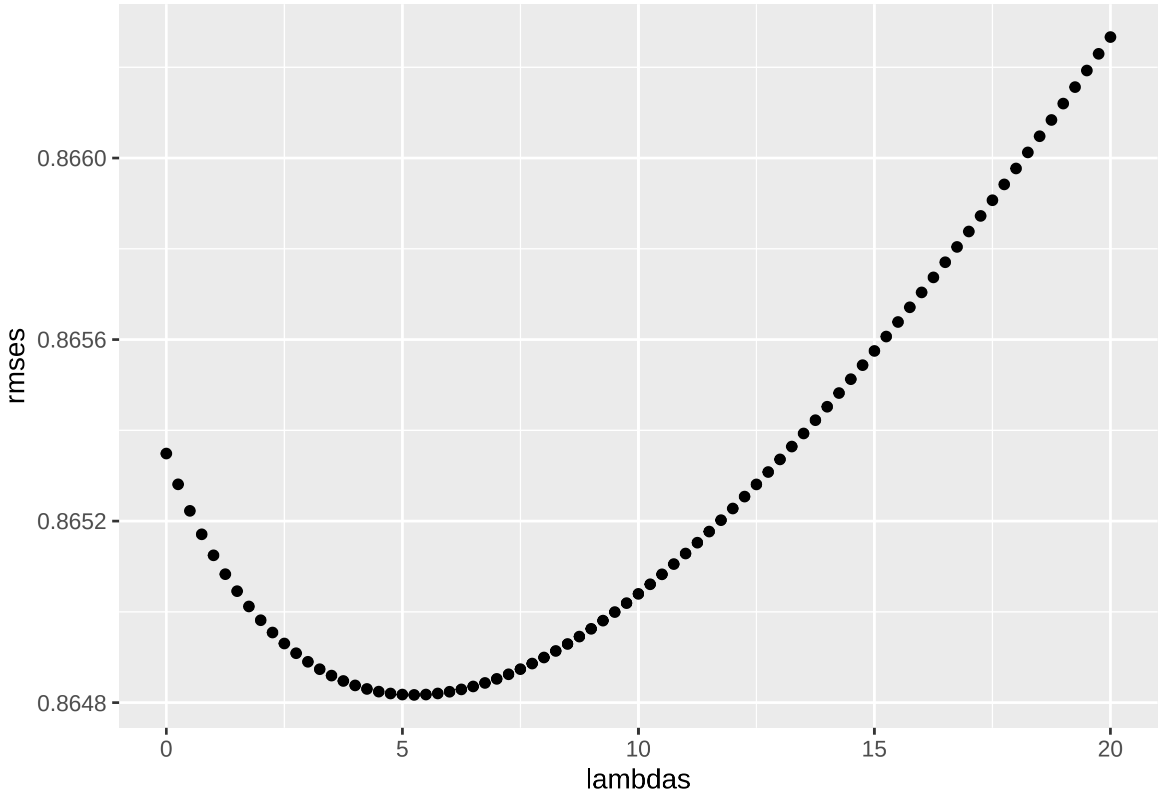
```
##    userId  b_user
##     <int>   <dbl>
##  1      1  1.68
##  2      2 -0.236
##  3      3  0.264
##  4      4  0.652
##  5      5  0.0853
##  6      6  0.346
##  7      7  0.0238
##  8      8  0.203
##  9      9  0.232
## 10     10  0.0833
## # ... with 69,868 more rows
```

and now we can calculate the RMSE of this model, that is

```
predicted_ratings_user <- validation %>%
  left_join(moviePenalty, by='movieId') %>%
  left_join(penaltyUser, by='userId') %>%
  mutate(pred = mu_edx + b_movie + b_user)
# test rmse results
model_MoviesUsers_rmse <- RMSE(validation_New$rating,predicted_ratings_user$pred)
rmse_results <- bind_rows(rmse_results,data_frame(method="Movie and User Effect Model", RMSE = model_Mov
model_MoviesUsers_rmse
```

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

- The fourth model, we consider users,movies,years, genres and than applicate the regularization




```
## [1] 5.25
```

the RMSE in this model is:

```
min(rmses)
```

```
## [1] 0.864817
```

```
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Regularized Movie and User Effect Model",RMSE = min(rmses)
```

Results

We see the results with different model applicate and relative RMSE

```
rmse_results %>% knitr::kable()
```

method	RMSE
Only Mean	3.5124652
Movie Effect Model	0.9439087
Movie and User Effect Model	0.8653488
Regularized Movie and User Effect Model	0.8648170

Conclusion

We see the best model to predict the rating movies is the fourth model where we applicate the regularization