MovieLens Report Project

Ernesto José Hernández Navarro

2024-03-25

Table of Contents

Introduction

Currently Machine Learning, one of the disciplines of artificial intelligence, has become one of the most important tools for public and private companies, in sectors such as health, economy, sports, traffic and movies, which is the case we are going to be analyzing this document.

Netflix organized a contest for a team that could create a movie recommendation system in 2006. This algorithm developed by BellKor's Pragmtic Chaos team in 2009 served as a reference for companies like Amazon and offer their customers more precisely the products that they have the most propensity to buy.

The goal of this project is to use the MovieLens dataset (which has 10 million ratings), R and RStudio, to achieve a root mean square error (RMSE) less than 0.86490. Due to the large amount of information the edX team has made available the code to split the data.

To help understand the data set, a data exploration will be carried out. For the algorithm tests, the information will be divided into a training and test set in order to validate the work done on the final model, the limitations and possible future potential.

Data exploration

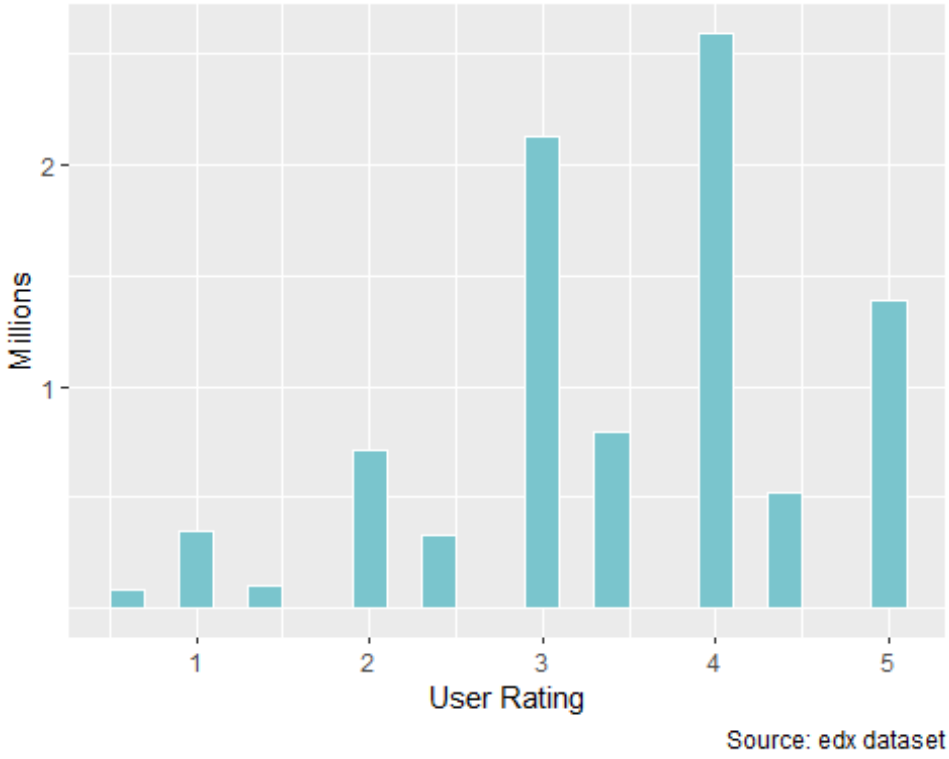
The data from edX data.frame has 9,000,055 rows and 6 columns, with ratings given by a total of 69,878 unique users, with a total of 10,677 unique movies. The data has structure and does not has nulls:

```
##    userId   movieId    rating timestamp     title    genres 
##         0         0         0         0         0         0 
```

On the top ten most rated films we see a clearly winner with Pulp fiction:

title	Count_Rating
Pulp Fiction (1994)	31362
Forrest Gump (1994)	31079
Silence of the Lambs, The (1991)	30382
Jurassic Park (1993)	29360
Shawshank Redemption, The (1994)	28015
Braveheart (1995)	26212
Fugitive, The (1993)	25998
Terminator 2: Judgment Day (1991)	25984
Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)	25672
Apollo 13 (1995)	24284

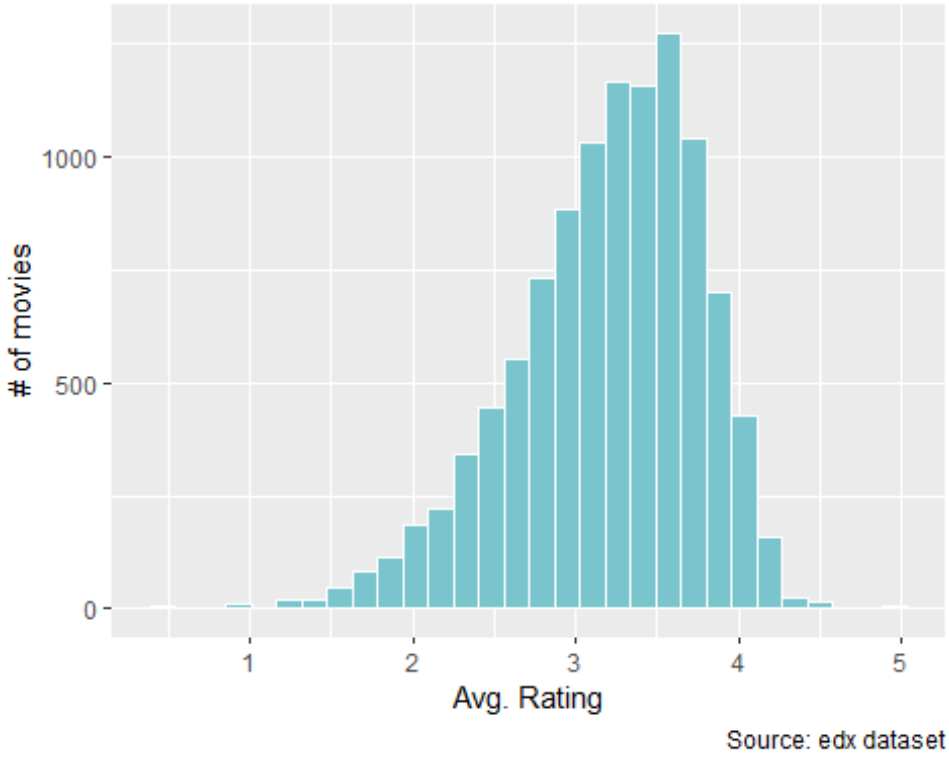
The rating most used by users is 4 representing the 28.76% of the total ratings, the minimum is 0.5, the maximum is 5. The integer rates were 7,156,885 which represents the 79.5% and the decimal ratings were 1,843,170 representing the 20.5%.



Ratings distribution

Movies

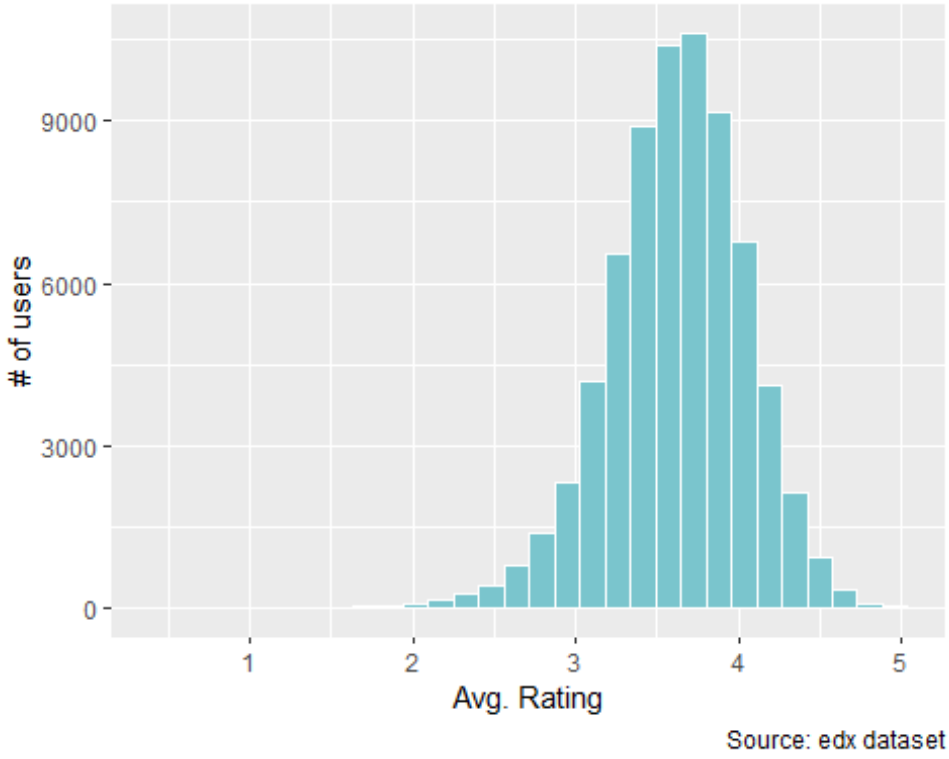
As we know, not all the users rates the received service, and in this case, the watched movie is not always rated, some are more rated than others, and 126 movies where rated with once. So, there is a bias that has to be consider on the training algorithm.



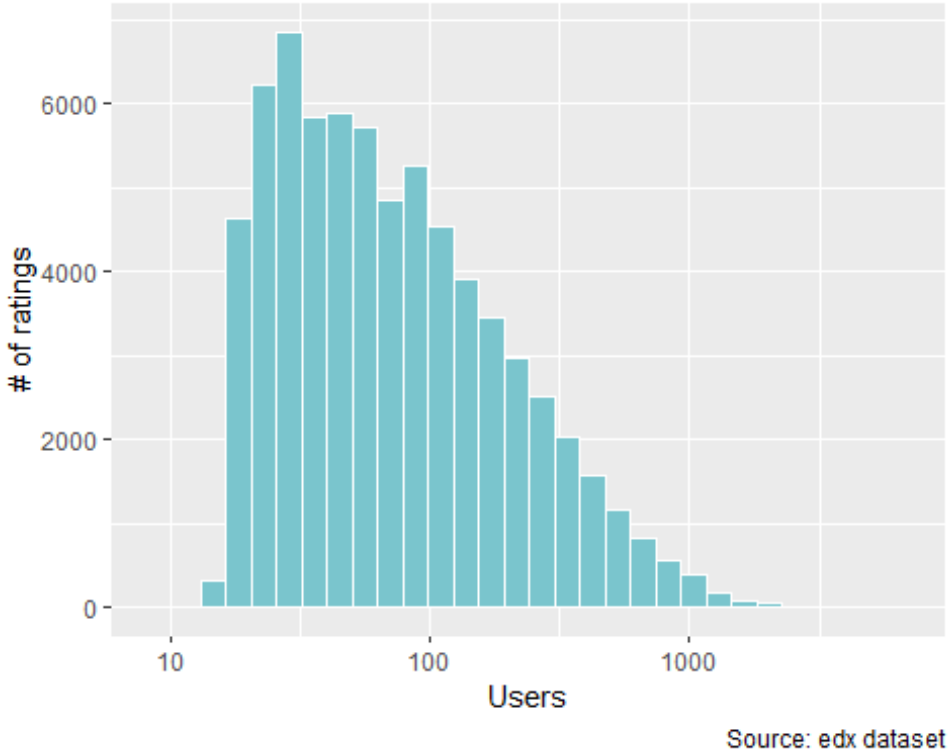
Distribution of rated movies in average

Users

Some users contribute more than others, and some of then a more benevolent (high ratings) with the movies. One user made 6616 and 2 did 1059 made less than 10 ratings. The bias is clearly with the following plot:



Distribution of users by Avg. Rating



# of ratings by user

Genre

Movies not always has one genre, so, in the edx data set we can see that the field "genre" assigns several genres and separates it with the symbol "|", and for our exploration the first one will be take. In the following table we can see that some genres have better ratings and a greater number of ratings:

```
## # A tibble: 20 x 3
##    genres               count rating
##    <chr>                <int>  <dbl>
##  1 Drama              3910127   3.67
##  2 Comedy             3540930   3.44
##  3 Action             2560545   3.42
##  4 Thriller           2325899   3.51
##  5 Adventure          1908892   3.49
##  6 Romance            1712100   3.55
##  7 Sci-Fi             1341183   3.4
##  8 Crime              1327715   3.67
##  9 Fantasy             925637   3.5
## 10 Children            737994   3.42
## 11 Horror              691485   3.27
## 12 Mystery             568332   3.68
## 13 War                 511147   3.78
## 14 Animation           467168   3.6
## 15 Musical             433080   3.56
## 16 Western             189394   3.56
## 17 Film-Noir           118541   4.01
## 18 Documentary          93066   3.78
## 19 IMAX                  8181   3.77
## 20 (no genres listed)       7   3.64
```

Drama and Comedy genre have clearly the most quantity of rates, Documentary and IMAX are the lowest rated. Also there are seven that don't have a genre.

Title

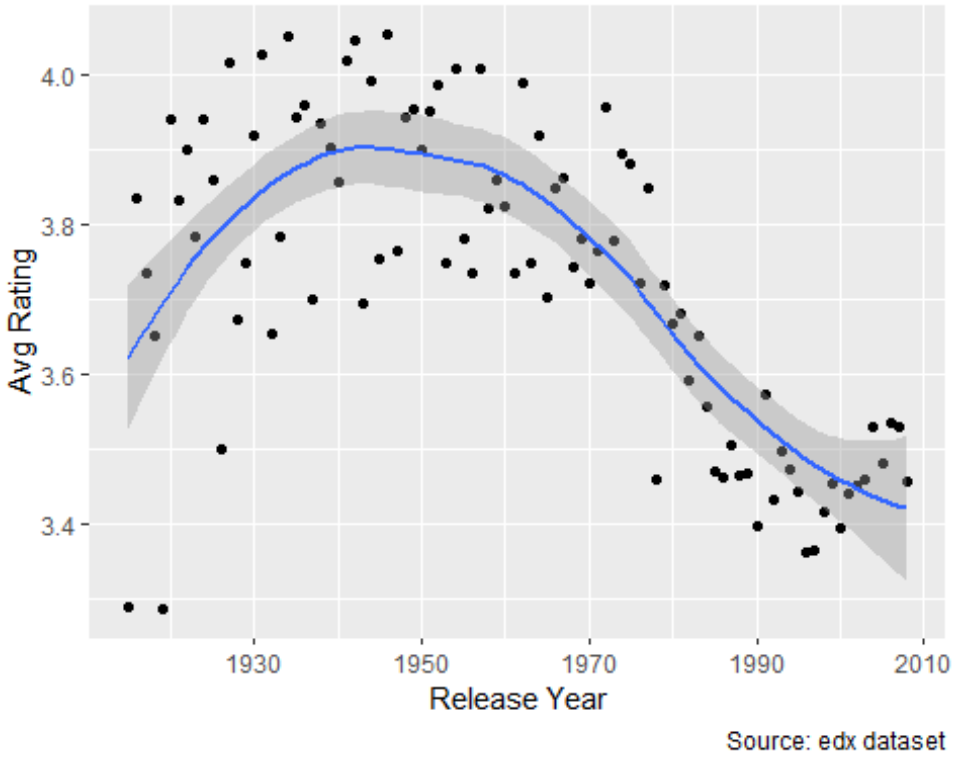
In the data set the column "Title" includes the release year of the movie as we can see in the following table:

```
## Selecting by n

## # A tibble: 10 x 2
##    title                                                     n
##    <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
```

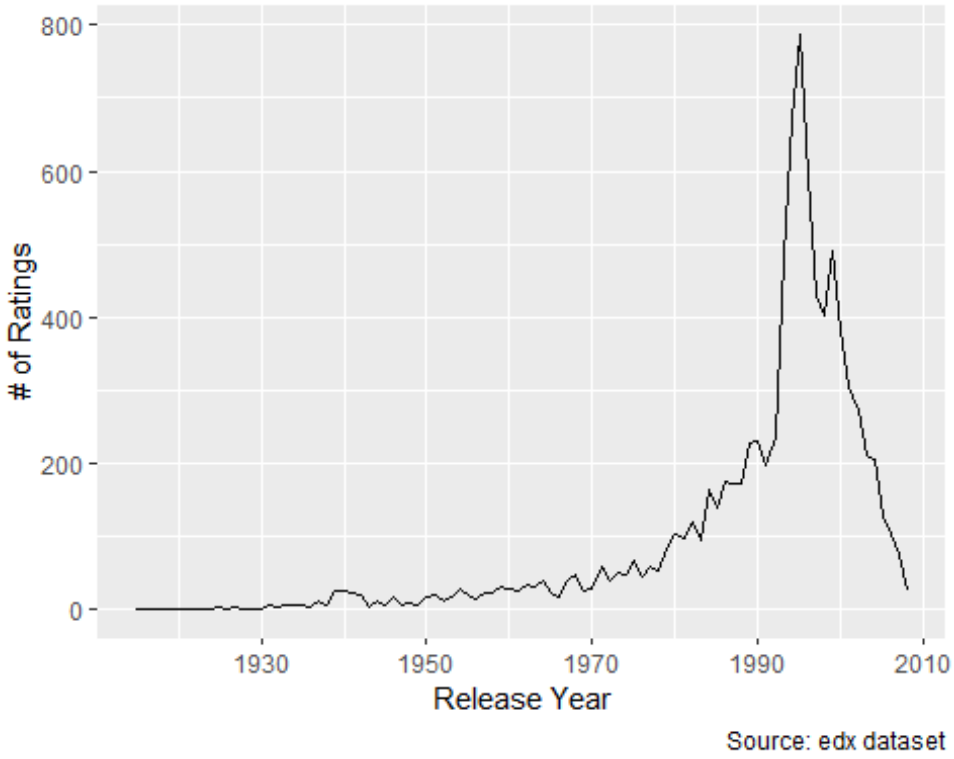
```
##  7 Fugitive, The (1993)                                          25998
##  8 Terminator 2: Judgment Day (1991)                             25984
##  9 Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977) 25672
## 10 Apollo 13 (1995)                                              24284
```

It looks like there is a bias on the year, on the following chart the curve increase between 1940 and 1950, after that and starting from 1970 the curve decrease the average rating. The number of rates starts it peak on 1990 and maximum number is on 1995.

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Plot with year of release:



Methods

```
## Warning in set.seed(2024, sample.kind = "Rounding"): non-uniform
'Rounding'
## sampler used

## Joining with `by = join_by(userId, movieId, rating, timestamp, title,
year,
## genres)`
```

As mentioned at the introduction our goal is to reach a RMSE less than 0.86490, to do it the edx dataset needs to be split in two parts, the training set and the test set. This together with cross-validation methods allows to prevent over-training.

Doing the same steps learned from professor Irizarry courses the data will be partitioned, 80% for training and 20% for testing using the libraries “caret”, “tydiverse” and “dplyr”.

Error loss

The RMSE is like the standard deviation of the residual predictors. The RMSE represent the error loss between the predicted ratings from applying the algorithm and actual ratings in the test set. The formula shown below, $y_{u,i}$ is defined as the actual rating provided by a user u for a movie i, $\hat{y}_{u,i}$ is the predicted rating for the same, and N is the total number of user/movie combinations.

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

Developing the algorithm

The goal set for this project is to achieve a RMSE equal or less than 0.86490, then it will be about reaching the goal step by step. We need to the fit the model with this formula:

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

The common method is to use the rating mean of every user on movies.

$$Yu, i = \mu$$

```
mu_hat <- mean(train_set$rating)
RMSE(test_set$rating, mu_hat)

## [1] 1.059951
```

And any number will be higher than $\hat{\mu}$ mean as we can see below

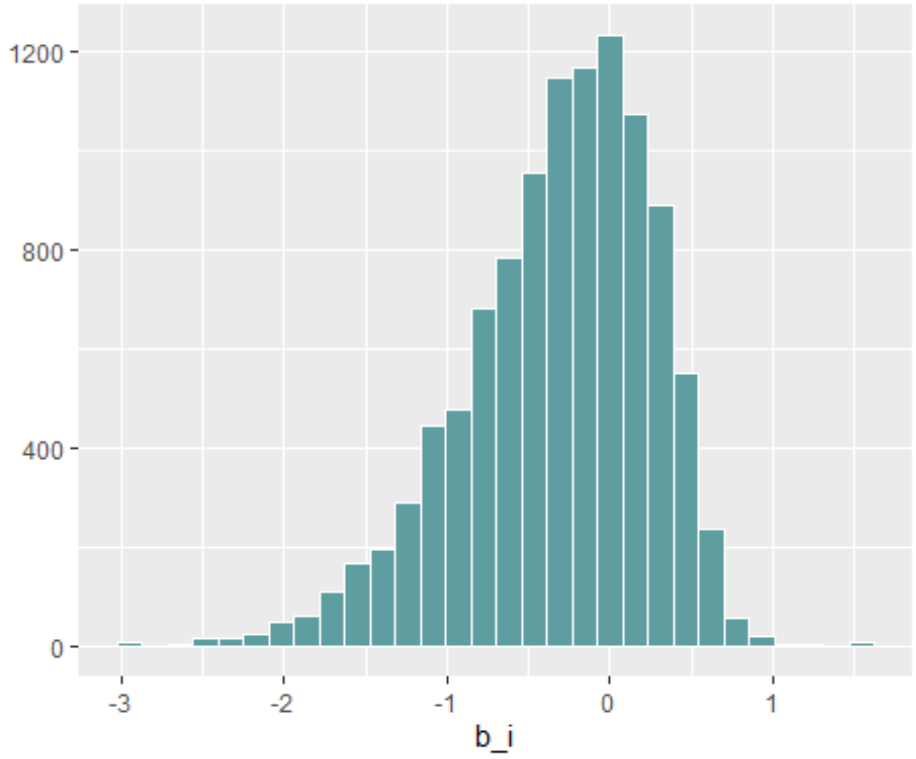
```
predictions <- rep(2.5, nrow(test_set))
RMSE(test_set$rating, predictions)

## [1] 1.465849
```

Movie effect

Due to the large dataset for this project a linear regression would take several time, a better option to work with is the least square estimate of the movie effect $\hat{b}_i$, it can be take from the average of $Y_{u,i} - \hat{\mu}$ in every movie i. The following formula was used to get the movie effect:

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

And we can see the effect of the movies with the following plot:



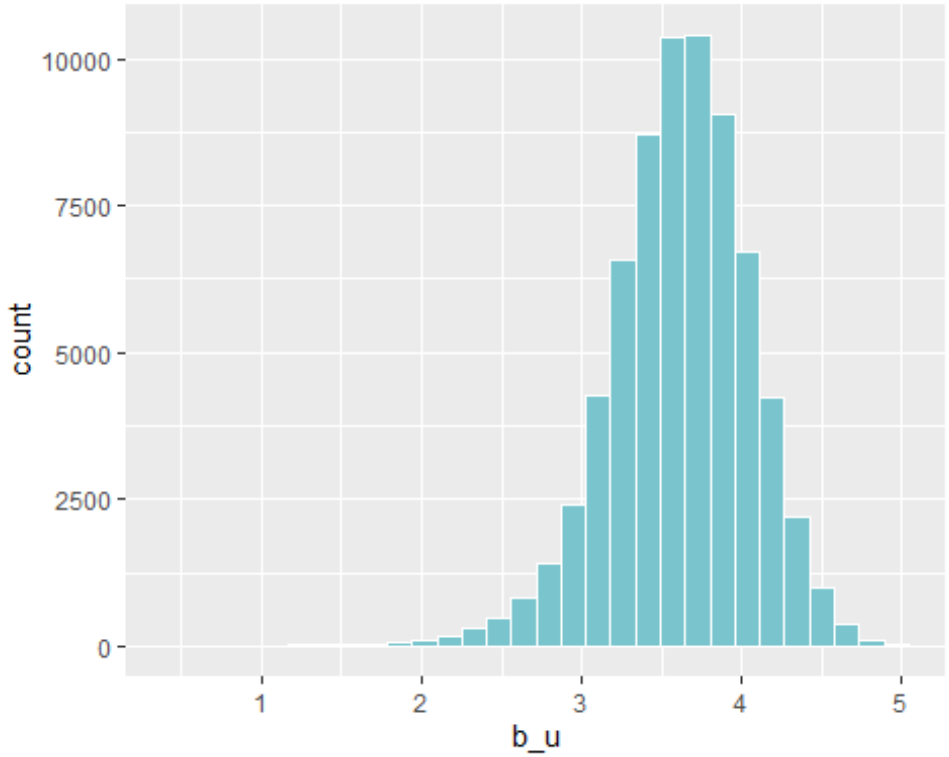
Distribution movie effect

The RMSE is lowest than the mean and it was a good improve, as it was shown in data exploration some movies get more rated than others, we still need improve more the result to get our objective.

Method	RMSE	Difference
Objective	0.8649	-
Movie Effect	0.94367	0.07877

User effect

Some users tend to rate more than others (quantity) and other are more critical than others, so adding the user effect is slightly better than only use the movie effect



Distribution user effect

Method	RMSE	Difference
Objective	0.8649	-
Movie Effect	0.94367	0.07877
Movie + User Effect	0.86561	0.00071

Gender effect

The movies acclaimed by critics then to be more rated by users thats why we saw an improve using movies and users, but this happens with the genre too. The table below shows the results.

Method	RMSE	Difference
Objective	0.8649	-
Movie Effect	0.94367	0.07877
Movie + User Effect	0.86561	0.00071
Movie, User, Genre Effect	0.86526	0.00036

Year effect

On the data exploration it was shown than it was a clearly year effect reaching it highest peak on the 90's. It adds a modest improve of 0.00036.

Method	RMSE	Difference
Objective	0.8649	-
Movie Effect	0.94367	0.07877
Movie + User Effect	0.86561	0.00071
Movie, User, Genre Effect	0.86526	0.00036
Movie, User, Genre, Year Effect	0.86509	0.00019

Date Review effect

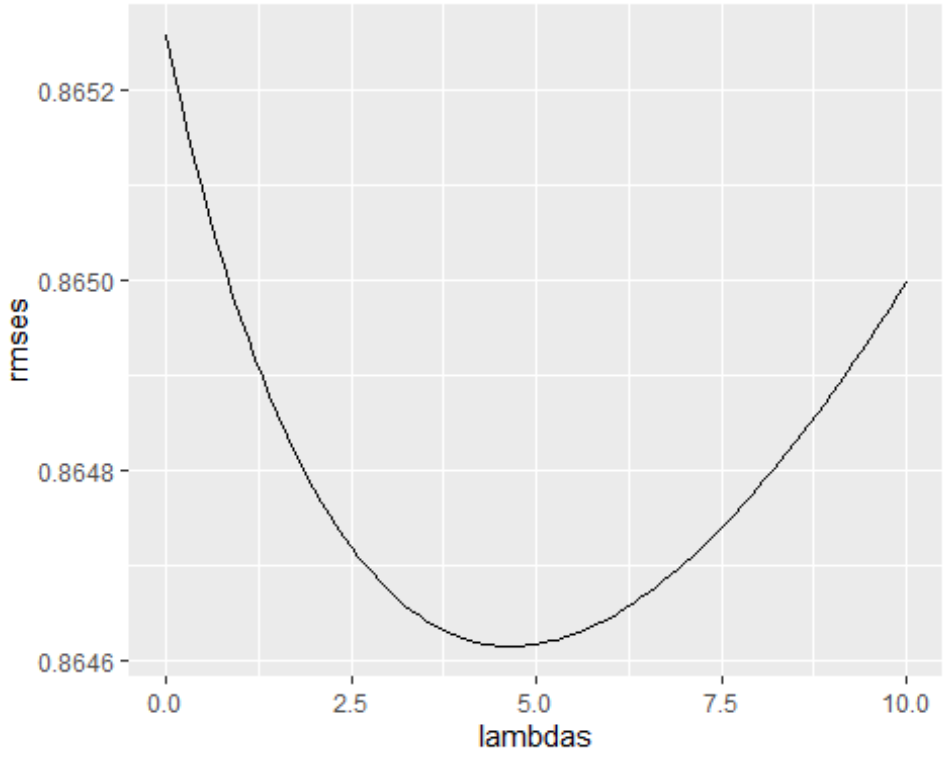
Our final bias is the date review, this column is a exact date with hours, to get a better approximation it was rounded to week.

Method	RMSE	Difference
Objective	0.8649	-
Movie Effect	0.94367	0.07877
Movie + User Effect	0.86561	0.00071
Movie, User, Genre Effect	0.86526	0.00036
Movie, User, Genre, Year Effect	0.86509	0.00019
Movie, User, Genre, Year, Date Review Effect	0.8649	0

This leads to the RMSE objective, but, can it be better?

Regularization effect

Answering the last question, yes, we can improve the last result with regularization. The following plot shows the RMSE tested for every λ value tested. The optimal parameter is 4.7 which minimized the RMSE to 0.86462.



Selecting best lambda

Method	RMSE	Difference
Objective	0.8649	-
Movie Effect	0.94367	0.07877
Movie + User Effect	0.86561	0.00071
Movie, User, Genre Effect	0.86526	0.00036
Movie, User, Genre, Year Effect	0.86509	0.00019
Movie, User, Genre, Year, Date Review Effect	0.8649	0
Regularized Movie, User, Genre, Year, Date Review Effect	0.86462	-0.00028

Final testing

With our algorithm defined and using using the training set and the test set to validate the RMSE, for our final step, the date review was modify to enhance the running time. The final result for the RMSE is 0.85621 that is less than the objective o the project:

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

Method	RMSE	Difference
Objective	0.8649	-
Final RMSE	0.85621	-0.00869

Conclusions

Analyzing the MovieLens dataset we found bias that were reduce to a RMSE lower than the objective, this was thanks to regularization to.

We could improve this result using matrix factorization, singular value descomposition (SVD) and principal component analysis (PCA). It quantify residuals within this error loss based on patterns observed between groups of movies or groups of users such that the residual error in predictions can be further reduced.

Reference

Irizarry, Rafael A. 2020. Introduction to Data Science: Data Analysis and Prediction Algorithms

with R. CRC Press.

http://rafalab.dfci.harvard.edu/dsbook/