HarvardX: PH125.9x Data Science Capstone Project - Movielens

Pedro Cadilha

07/12/2020

Introduction

The MovieLens Dataset it's a huge database generated by the GroupLens research lab with over 20 million ratings for over 27,000 movies by more than 138,000 users.

In this project we are going to use a smaller subset, known as the 10M version of the MovieLens dataset, to make computation more easier.

The goal of this project is to build a recommendation system based on the users movie ratings, Which means being able to predict a rating an user would give to a movie.

We will compare the different models used to see how well they are performing using the Root Mean Squared Error (RMSE) as our loss function. We can interpret RMSE similar to standard deviation. Our final RMSE as to be less than 0.8649.

The code to generate the datasets to train ("edx") and validate ("validation") our model was provided.

Data exploration

First we are going to take a first look at the data, dimensions of the dataset and structure. We have a table with the dimensions:

```
## [1] 9000055       6
```

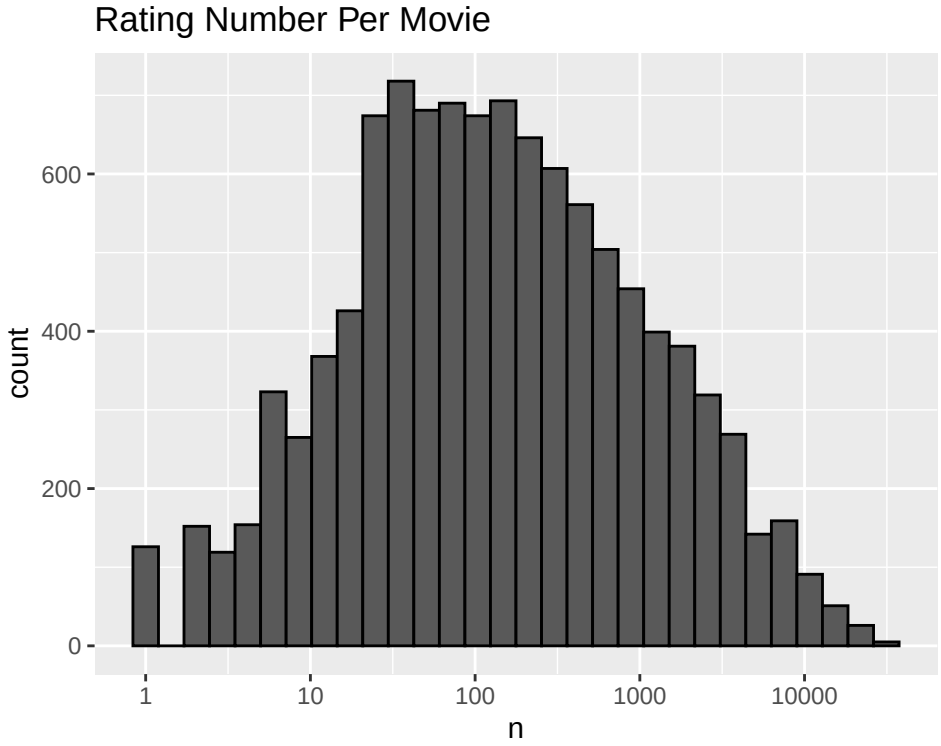
With the following structure:

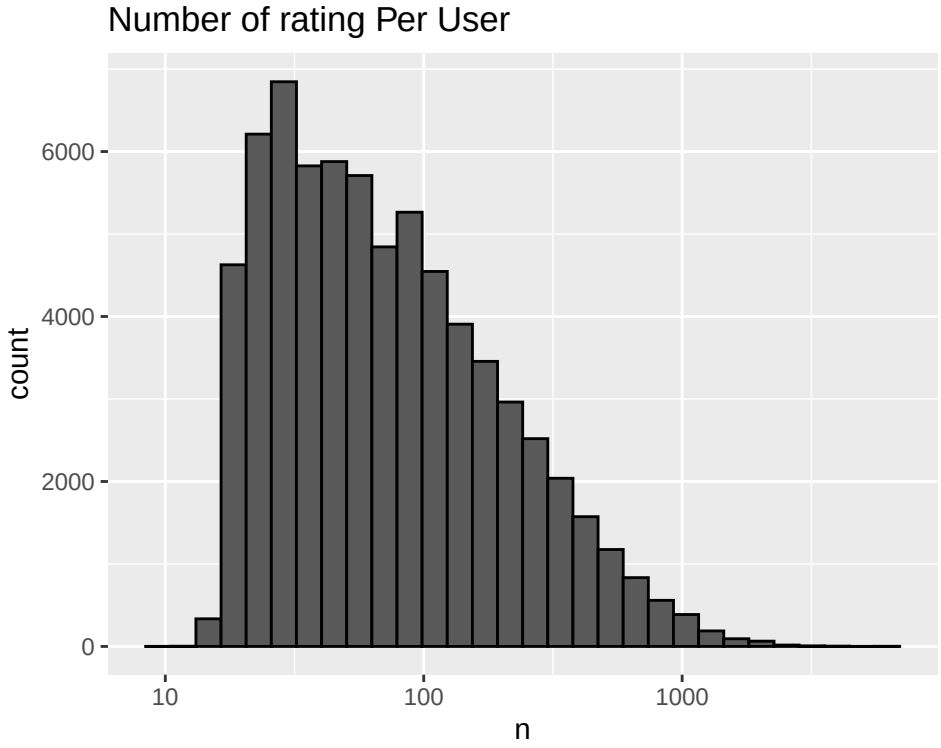
userId	movieId	rating	timestamp	title	genres
1	122	5	838985046	Boomerang (1992)	Comedy|Romance
1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
1	329	5	838983392	Star Trek: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

Six columns with the userID, movieID, rating, timestamp (time when the user rated the movie, which has to be converted to a readable format), the title of the movie (between brackets is the year the movie was released) and the genre classification.

Users and Movies

The activity of the users is variable, which means some users rate more movies than others and also the more popular movies receive more ratings than other more obscure movies.



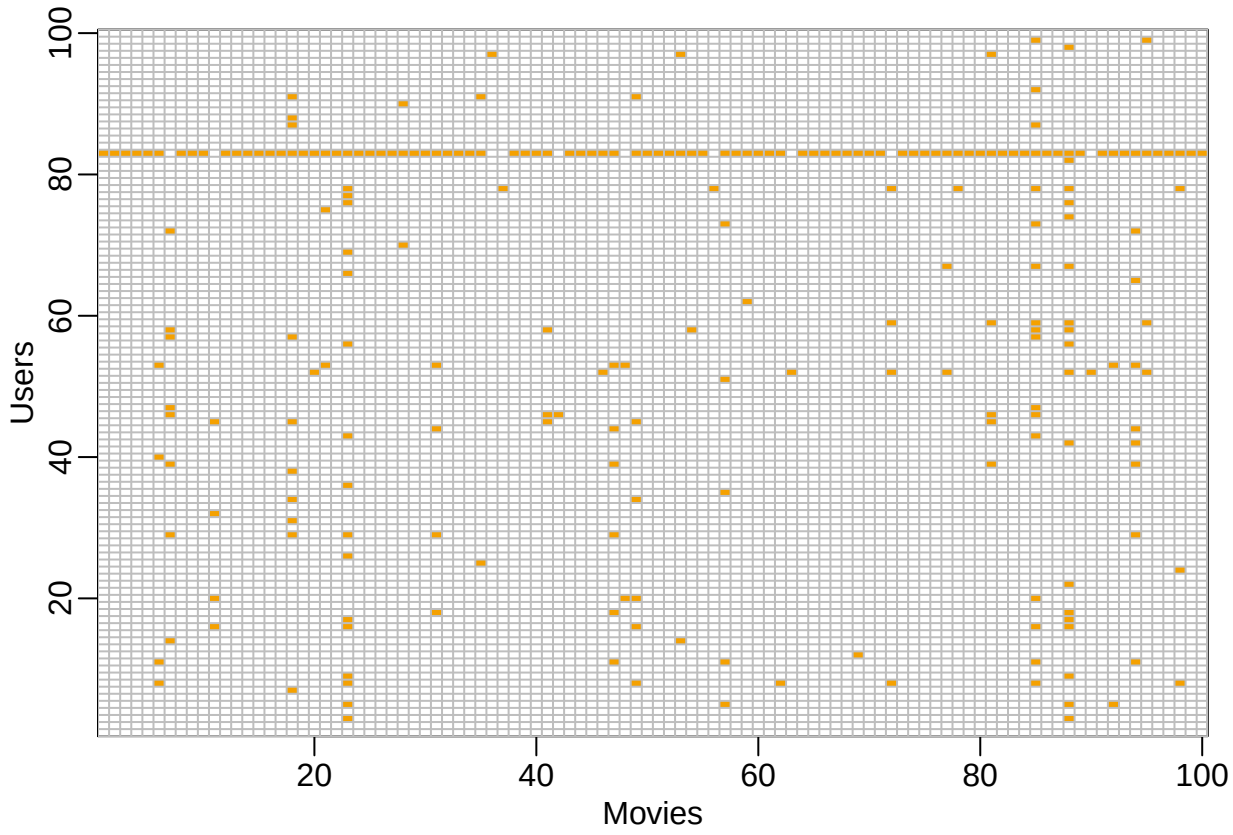


By this two plots we see that there are movies who get much more ratings than others, and there also users much more active than others.

Let's explore this a bit more exemplifying with a small sample of users:

userId	Forrest Gump (1994)	Jurassic Park (1993)	Pulp Fiction (1994)	Silence of the Lambs, The (1991)
1	5	NA	NA	NA
4	NA	5	NA	NA
7	NA	NA	NA	3
8	NA	3	NA	4
10	3	NA	2	3

We can see that even very popular movies are missing rating by the users.
If we take a random sample of 100 unique users and their respective rating over a sample of 100 movies, it's possible to visualize how sparse is this matrix:



Only the filled squares correspond to a rating.
In fact we can see how many unique users and movies exist in the dataset:

n_users	n_movies
69878	10677

And if all of them rated every single movie we would have a table with 746087406 entries.

Instead there is only a small fraction, 0.012063, of all possible rates.

Genres

The movies are categorized by genres, but a lot of them have a combination of genres, we can see how many of these combinations exist and also if there is some relation between the genres combination and the users rating.

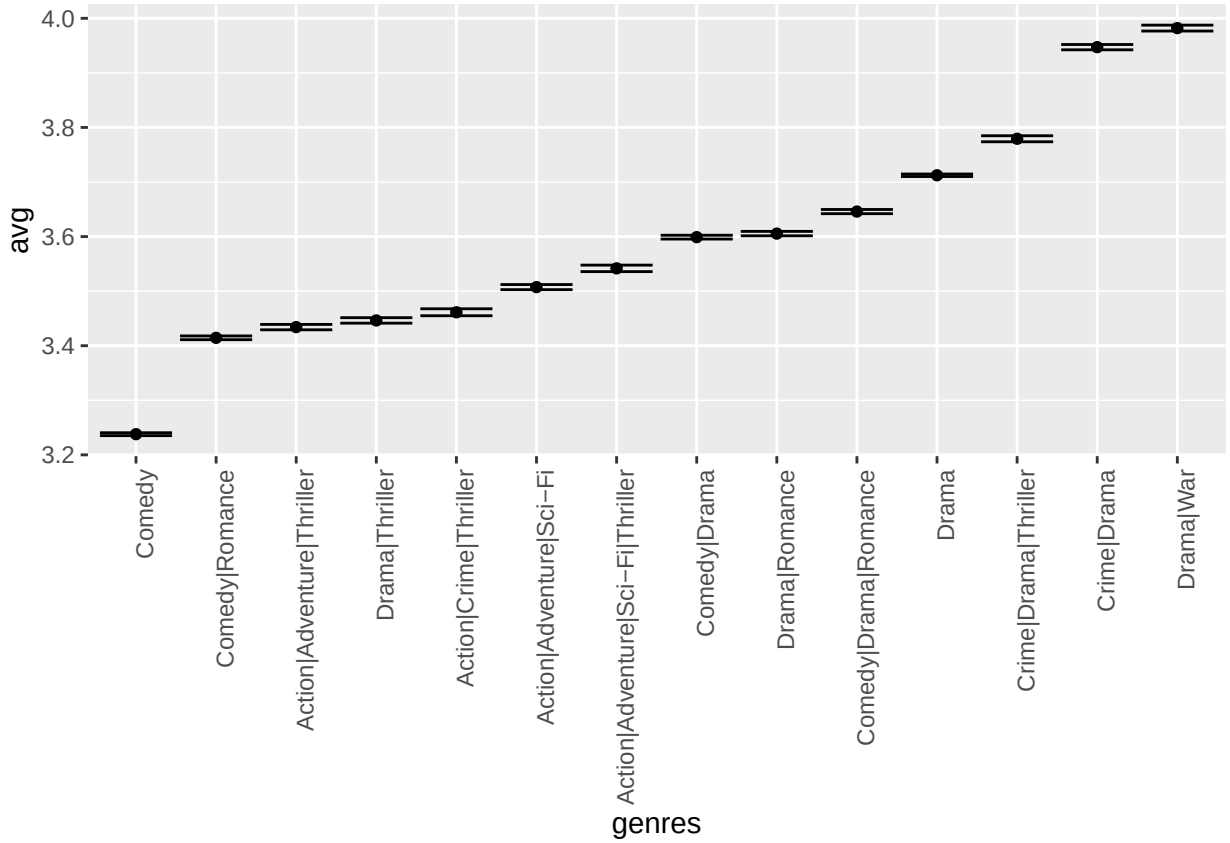
The number of distinct genre combinations are 797.

And also, the more popular and unpopular genre combinations:

```
## # A tibble: 797 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 787 more rows
```

```
## # A tibble: 797 x 2
##    genres                                      count
##    <chr>                                       <int>
##  1 Action|Animation|Comedy|Horror                  2
##  2 Action|War|Western                              2
##  3 Adventure|Fantasy|Film-Noir|Mystery|Sci-Fi      2
##  4 Adventure|Mystery                               2
##  5 Crime|Drama|Horror|Sci-Fi                       2
##  6 Documentary|Romance                             2
##  7 Drama|Horror|Mystery|Sci-Fi|Thriller            2
##  8 Fantasy|Mystery|Sci-Fi|War                      2
##  9 Action|Adventure|Animation|Comedy|Sci-Fi        3
## 10 Horror|War|Western                             3
## # ... with 787 more rows
```

We can plot the average rating by genre, for genres or genres combinations with more than 100000 rates.



It's clear that some genres combinations got better ratings by the users, so this variable, might be useful to consider in our prediction model.

Rating

Proceeding to analyze the ratings, we see that the most popular are the movies more rated and that there is a preference by the users in rating the movies with a whole number instead of half numbers.

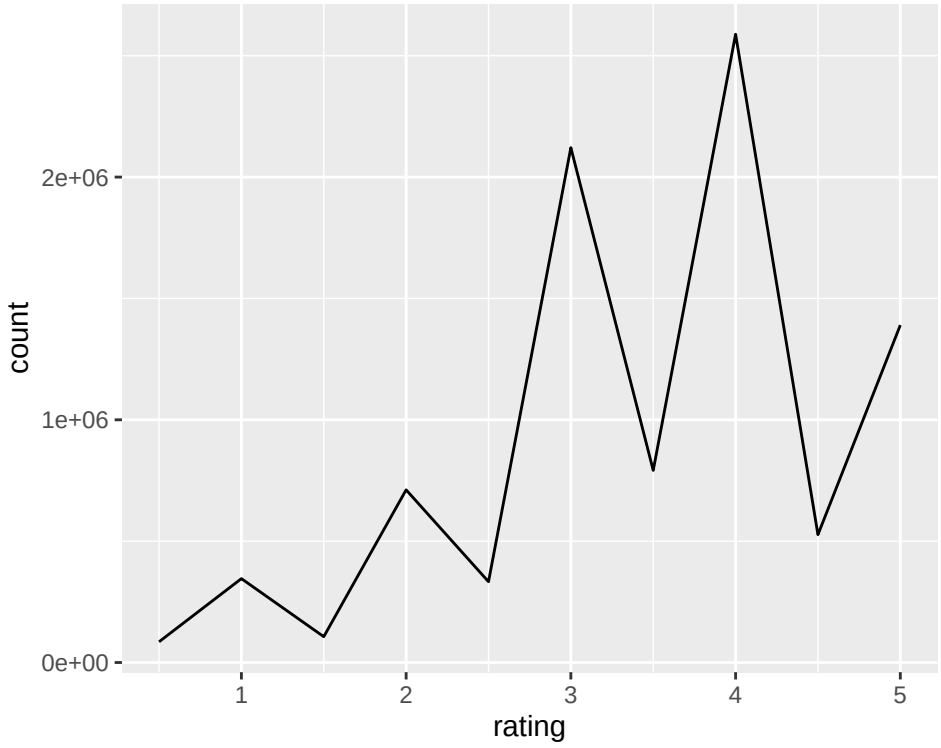
Movies with more ratings:

```
## # 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
```

The rating distribution:

```
## # A tibble: 5 x 2
##   rating   count
##    <dbl>   <int>
## 1    4   2588430
## 2    3   2121240
## 3    5   1390114
## 4    3.5  791624
## 5    2    711422
```

The plot of the rating distribution:



Year released

To explore a bit more our data, we are going to analyse the relation between the year a movie was released and the rating. We need, firstly, to separate the movie name from the year in the column movie title of our both datasets, **edx** and **validation**.

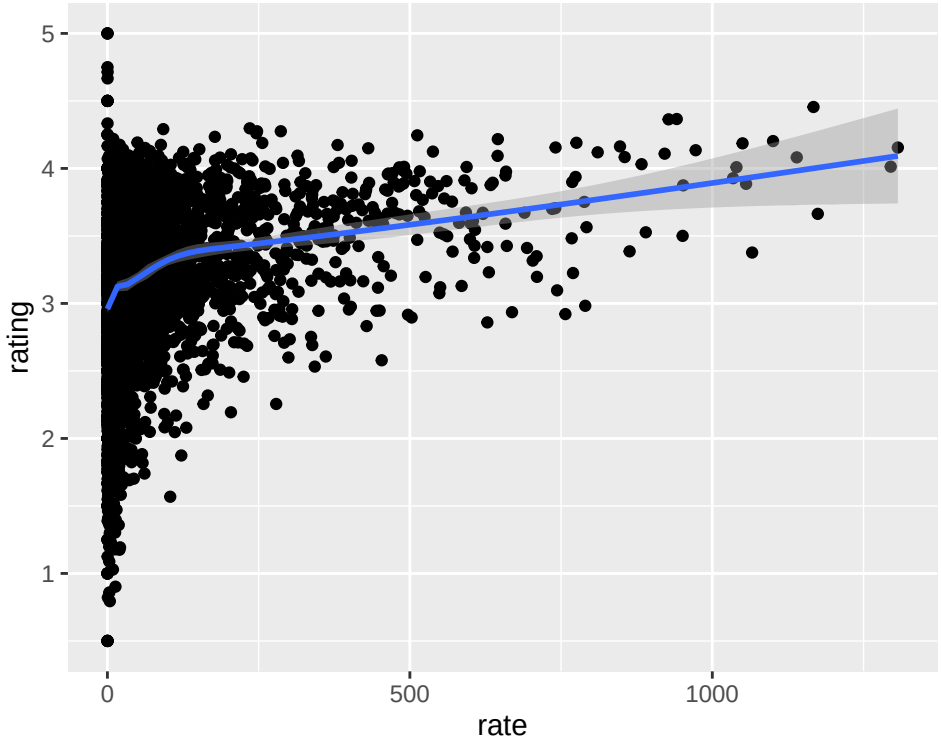
And now it looks like this:

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

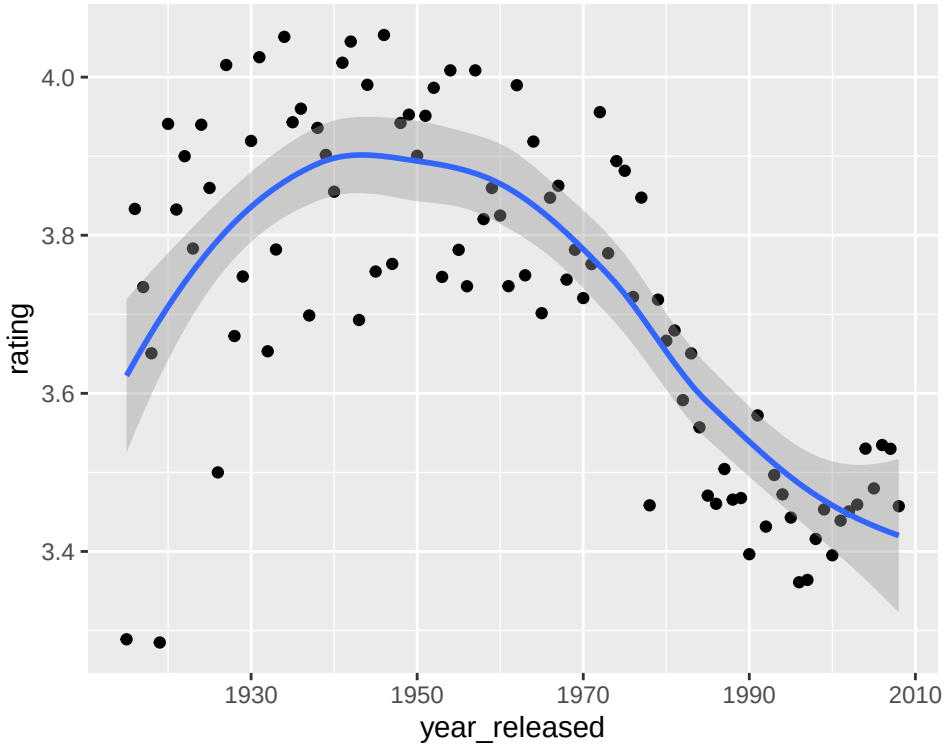
It's possible to verify that the most frequent rated movies have above average rating:

```
## # A tibble: 25 x 6
##    movieId     n years title                          rating  rate
##      <dbl> <int> <dbl> <chr>                           <dbl> <dbl>
##  1     296 31362    24 Pulp Fiction                     4.15 1307.
##  2     356 31079    24 Forrest Gump                     4.01 1295.
##  3     480 29360    25 Jurassic Park                    3.66 1174.
##  4     318 28015    24 Shawshank Redemption, The        4.46 1167.
##  5     110 26212    23 Braveheart                       4.08 1140.
##  6    2571 20908    19 Matrix, The                      4.20 1100.
##  7     780 23449    22 Independence Day (a.k.a. ID4)    3.38 1066.
##  8     150 24284    23 Apollo 13                        3.89 1056.
##  9    2858 19950    19 American Beauty                  4.19 1050
## 10     457 25998    25 Fugitive, The                    4.01 1040.
## # ... with 15 more rows
```

And, with this plot we can see clearly that movies who are rated more by year, have better rating, and the ones who are rated less, don't have a clear tendency.



We can also analyze if the rating has some connection with the year the movie was released, with this plot:



Seems that the classic old movies have better ratings.

We already took a view on the dataset, and from that, we can think that:

- Users rate differently, some are more picky on the movies than others.
- Popular, blockbuster or acclaimed movies are rated differently.
- There are some genres who are better rated than others.
- There is some relation between the year the movie was released and the rating

Modeling

Now we can start modeling using some of this variables. First we are going to split our dataset **edx** in a train ("train_set") and test set ("test_set").

And we are going to define our function to calculate our model performance, as we stated before we are going to use Root Mean Squared Error (RMSE):

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

with $\hat{y}_{u,i}$ and $y_{u,i}$ being the predicted and actual ratings, and N, the number of possible combinations between user u and movie i.

First Approach

Predict the same rating for all movies and all users with all the differences being explained by random error.

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

Where μ is the average of all ratings in edx dataset and $\varepsilon_{u,i}$ corresponds to the independent errors sampled from the same distribution. The estimate that minimizes the RMSE is just the average of all the ratings.

The average is 3.5125267 and the RMSE is 1.0605613. Naturally this value is far away from our requirements of RMSE=0.8649.

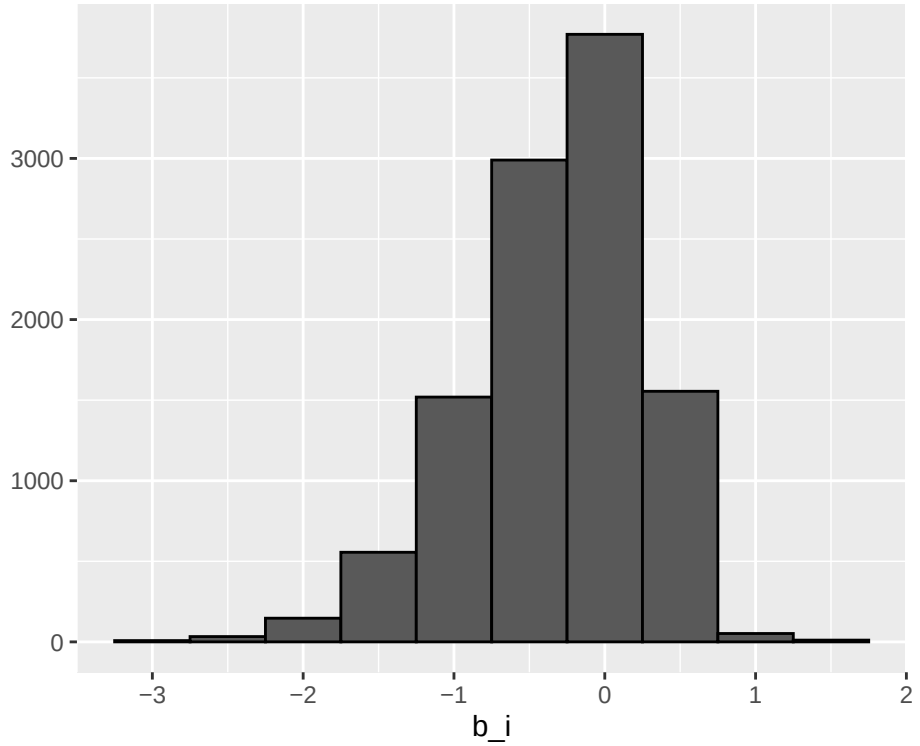
Along the way we are going to use a table to add our models results.

Movie effect model

This model adds the movie effect to the basic approach, the term b_i, represents the average rating for movie i.

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

We are going to evaluate how likely is a movie rating to be away from the mean rating.



Remember the mean from all the ratings is 3.5125267.
From the plot, the b_i (movie effect) range from -3 (rating of 0.5) to 1.5 (rating of 5).

The RMSE for this model is 0.9439868.
Adding results to our summary table:

method	RMSE
Just the average	1.0605613
Movie Effect Model	0.9439868

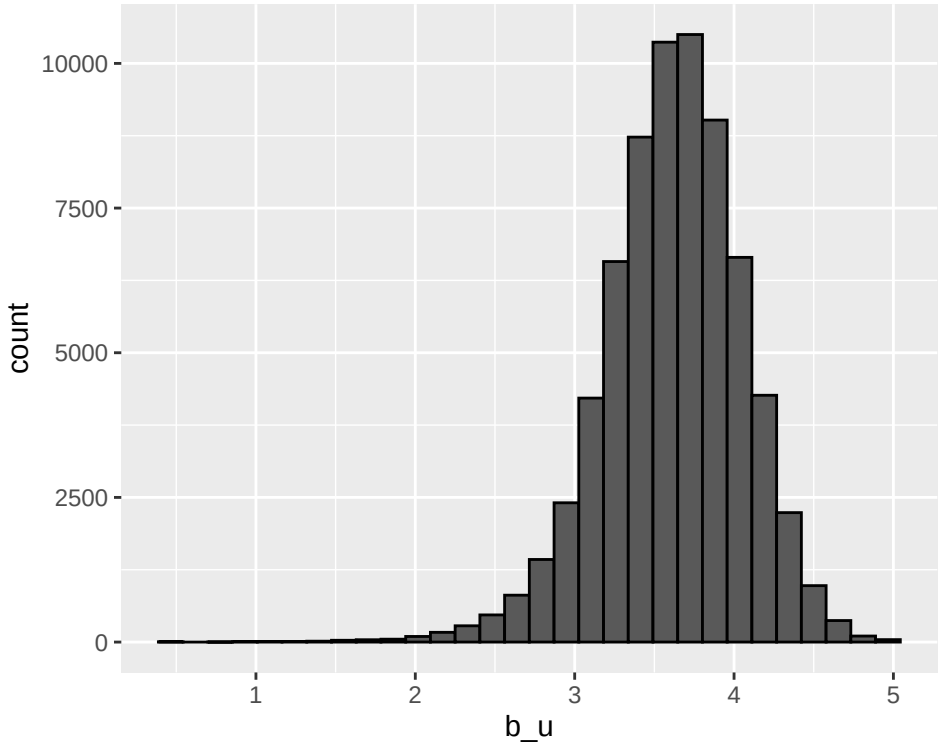
Movie + User Effect Model

Now we are going to add a user effect to our model:

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

where b_u is the user effect parameter. Which means the average rating given by a unique user to all the movies that he rated.

We can plot the distribution of this effect:



The RMSE for this model is 0.8666408.
Adding results to our summary table:

method	RMSE
Just the average	1.0605613
Movie Effect Model	0.9439868
Movie + User Effects Model	0.8666408

Our model is improving perfomance, but we didn't reach our goal still! Our error comes from the fact that in our prediction, we gave the same weight, to movies with a lot or a reduced number of rates. So the movies with better ratings:

```
##  [1] "Hellhounds on My Trail"
##  [2] "Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva)"
##  [3] "Satan's Tango (SÃ¡tÃ¡ntangÃ³)"
##  [4] "Fighting Elegy (Kenka erejii)"
##  [5] "Sun Alley (Sonnenallee)"
##  [6] "Along Came Jones"
##  [7] "Angus, Thongs and Perfect Snogging"
##  [8] "Bullfighter and the Lady"
##  [9] "Blue Light, The (Das Blaue Licht)"
## [10] "Constantine's Sword"
```

or lower:

```
##  [1] "Besotted"                                "Grief"
```

```
##  [3] "Altered"                              "Accused (Anklaget)"
##  [5] "Confessions of a Superhero"           "War of the Worlds 2: The Next Wave"
##  [7] "Karla"                                "SuperBabies: Baby Geniuses 2"
##  [9] "Disaster Movie"                       "From Justin to Kelly"
```

are quite obscure, because they didn't have a meaningful number of ratings, as we can see for the top movies:

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

These movies were rated only once!

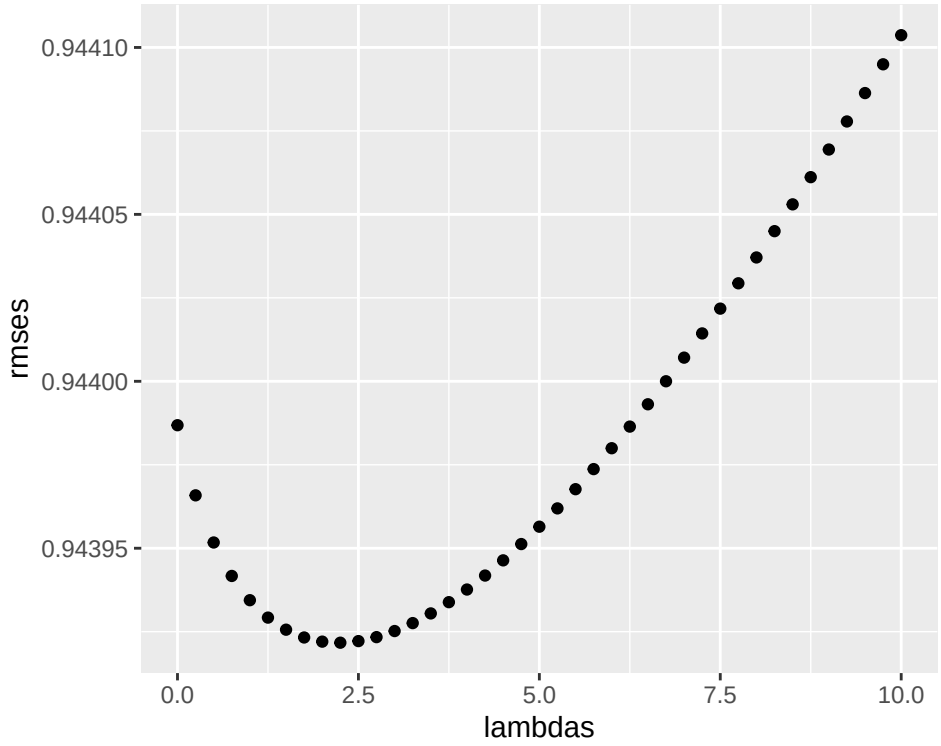
Movie Effect Regularized

That's why we need to "regularize the effect", which means, giving a penalty term to reduce the weight of weak predictions (reduced number of ratings).

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

Where n_i is the number of ratings per movie.

We will try a sequence of λ to find the one who minimizes the RMSE by plotting the values of RMSE for each iteration:



From the plot the λ value is 2.25 and the RMSE is 0.9439217. Adding the results to our table:

method	RMSE
Just the average	1.0605613
Movie Effect Model	0.9439868
Movie + User Effects Model	0.8666408
Movie Effects Regularized	0.9439217

Our performance improved slightly comparing with the movie effect without regularization.

Movie + User Effect regularized

Let's do the same, but now in the combined movie+user model. We will apply regularization to both movie and user effect.

Now the optimal λ is 4.75 and the corresponding RMSE is 0.8659626. The summary of our results until now:

method	RMSE
Just the average	1.0605613
Movie Effect Model	0.9439868
Movie + User Effects Model	0.8666408
Movie Effects Regularized	0.9439217
Movie + User Effects Regularized	0.8659626

Our RMSE is getting better but not good enough still!

Movie + User + Genre effect regularized

Since, we saw previously that, some genres have better ratings than others and some of them also have much bigger number of ratings than others, we are going to add also the genre effect regularization to our model.

Now the optimal λ is 4.75 and the corresponding RMSE is 0.865651. The summary of our results until now:

method	RMSE
Just the average	1.0605613
Movie Effect Model	0.9439868
Movie + User Effects Model	0.8666408
Movie Effects Regularized	0.9439217
Movie + User Effects Regularized	0.8659626
Movie + User + Genres Effects Regularized	0.8656510

Movie + user + genre + year released effects regularized

We saw before, that there was a relation between the year a movie was released and it's rating, therefore we are going to use the variable "year_released" also as a feature in our model.

Now the optimal λ is 4.5 and the corresponding RMSE is 0.8654882. The summary of our results until now:

method	RMSE
Just the average	1.0605613
Movie Effect Model	0.9439868
Movie + User Effects Model	0.8666408
Movie Effects Regularized	0.9439217
Movie + User Effects Regularized	0.8659626
Movie + User + Genres Effects Regularized	0.8656510
Movie + User + Genres + Year Released Effects Regularized	0.8654882

Calculation of final RMSE

We can now use our optimal λ, 4.5, in our validation set to calculate the final **RMSE** of our model.
Now we are going to use the entire **edx** dataset to calculate the predictions of our model, which then will be used to calculate the error of our predictions against the **validation** dataset.

Our final results are summarized in the following table:

method	RMSE
Just the average	1.0605613
Movie Effect Model	0.9439868
Movie + User Effects Model	0.8666408
Movie Effects Regularized	0.9439217
Movie + User Effects Regularized	0.8659626
Movie + User + Genres Effects Regularized	0.8656510
Movie + User + Genres + Year Released Effects Regularized	0.8654882
RMSE_Validation_Set	0.8642948

The combination of the feautures, movie, user, genres and year_released regularized, improved our model to a value of RMSE of 0.8643, inferior to 0.8649 that was demanded, in the validation dataset.

Conclusion

By using a combination of features, it was possible to predict the rating of the movies in our validation set. We minimized our RMSE applying regularization to our features.
It would be possible to improve this model working on the genres effect, exploring it a bit more, by splitting each movie genre in all the categories.
Another approach would be working with the year the movie was rated or even analyzing the effect on the rating of the difference between the year the movie was rated and the the year it was released.
Furthermore a model utilizing matrix factorization approach can also be used for deeper exploration.