HarvardX PH125.9x

EDX Project

Miguel Angel Bustos Sáez

2022-08-06

Contents

1 The introduction

The present project it´s "The Prediction Model Performance", based on The Sample Mean Square Error or RMSE. Using data analysis packages and the ten percent of the edx dataset: the validation set.

With the purpose to develop a recommendation system, here it´s presenting a series of data analysis where RMSE it´s the standard deviation of the residuals, and these residuals are a measure of how far from the regression line data points are and how spread out these residuals are. And shows how concentrated the data is around the line of best fit, having an objective reached below 0.8649

2 The Data

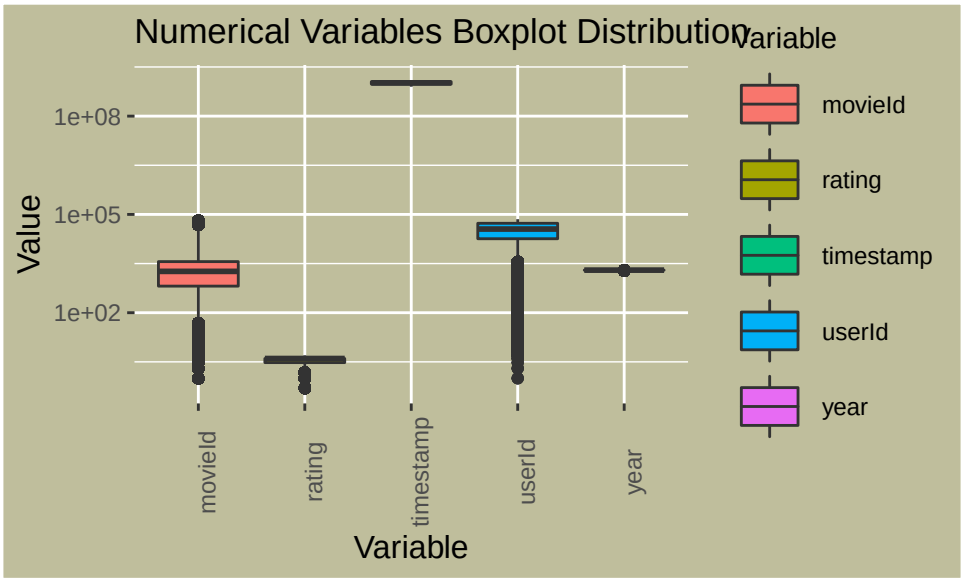
2.1 The Variables

The variables are; userId, movieId, rating, timestamp, title and genres. the timestamp passed to as date time format creating a date column. Additionaly it´s created a year variable, extracted from movie column:

The validation set

In the validation set, we have eight variables or columns:

- UserId: A number to identify a user
- MovieId: A number to identify a movie
- Rating: One to five, in a sequence of 0.5, are 10 ratings
- Timestamp: Record the time or date of
- Title: The name of the movie
- Genres: The gente of the movie, some movies have more than one
- Date: The format of hour, minute and second registration
- Year: The year of that registration ranking



And here we have a small boxplot that shows the numeric variables from validation data set

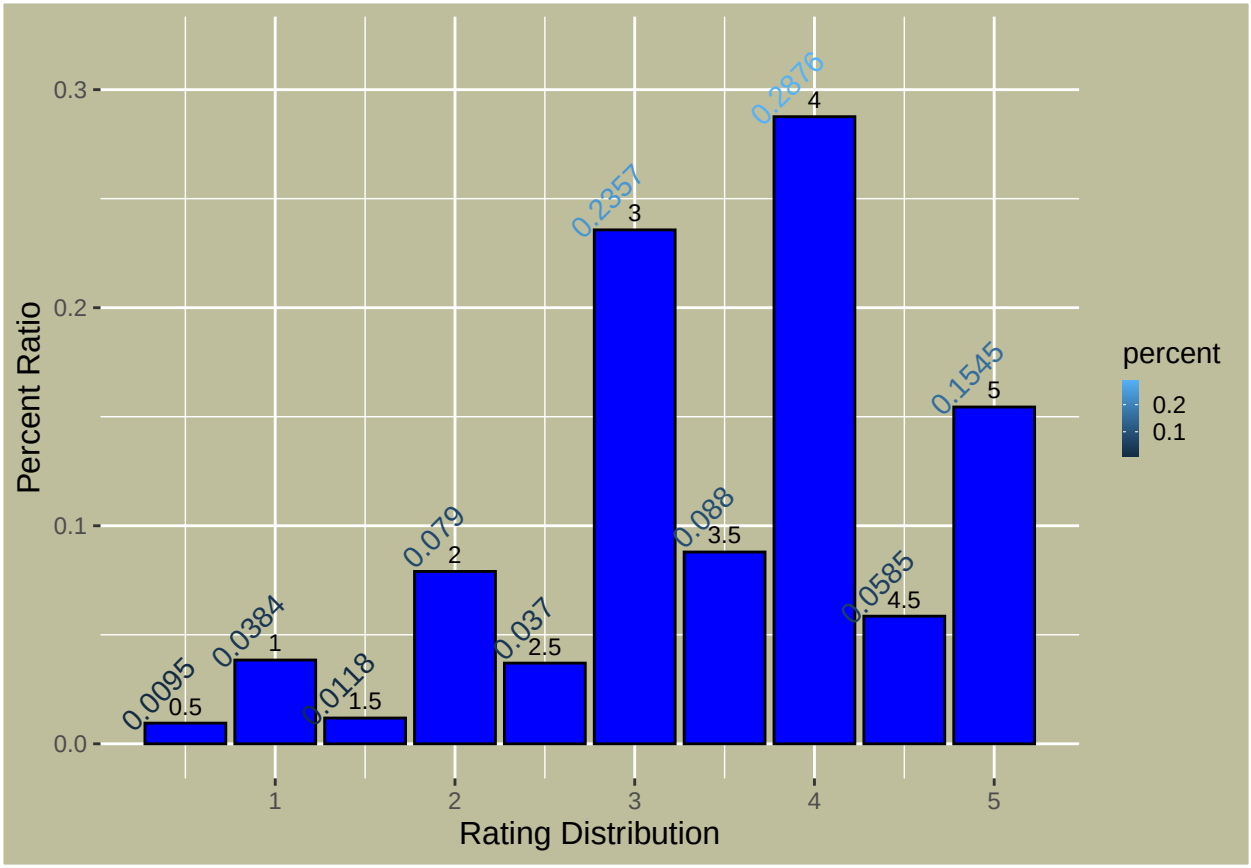
2.2 Rating Proportion

Mode & Mean: The mode it´s the rating 4 with more than two million and a half registration. And the mean it´s 3.5

```
## # A tibble: 10 x 3
##    rating       n percent
##     <dbl>   <int>   <dbl>
##  1    0.5   85374 0.00949
##  2    1    345679 0.0384
##  3    1.5  106426 0.0118
##  4    2    711422 0.0790
##  5    2.5  333010 0.0370
##  6    3   2121240 0.236
##  7    3.5  791624 0.0880
##  8    4   2588430 0.288
##  9    4.5  526736 0.0585
## 10    5   1390114 0.154
```

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

2.3 Rating and the ratio histogram distribution



Plot with the percent ratio and rating distribution.

2.4 The sample size

The issue of calculating the effective sample size it´s though this formula:

$$n < -(p * q) * z^2/e^2$$

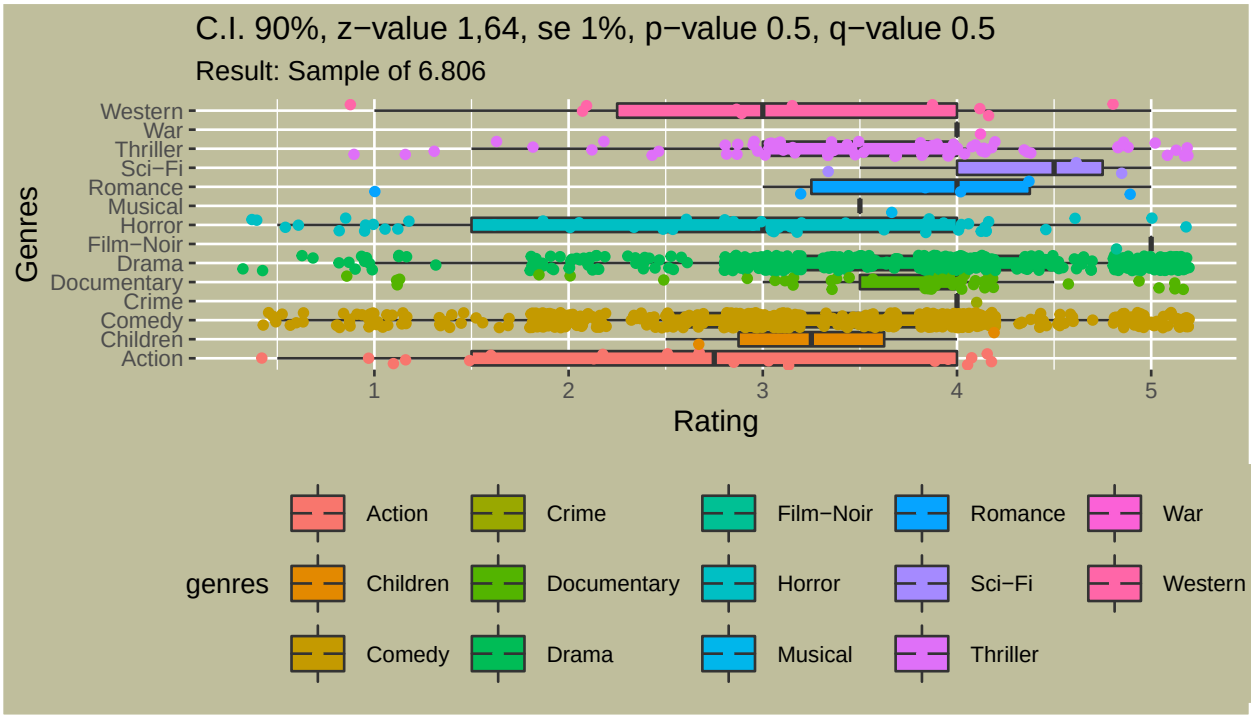
Example of how to calculate the finite sample size With this formula, the sample size is determined by calculating the sample. The confidence level corresponds to a Z-score. This is a constant value needed for this equation. Here are the Z-scores for the most common trust levels:

Z-score	p	q	e	z	Formula	Sample Result
90	0.5	0.5	0.01	1.65	n <- $(pq)z^{2/e}2$	6.806

Based on this example, and on our formula, the "N" will be the "Validation" quantity, we will choose our Z and it will be 1.65 (remember that the researcher assigned a confidence level of 90%) and "e" will be 1% . And since our example says that the probability of the event occurring is unknown, we assign 50% to "p" and 50% to "q".

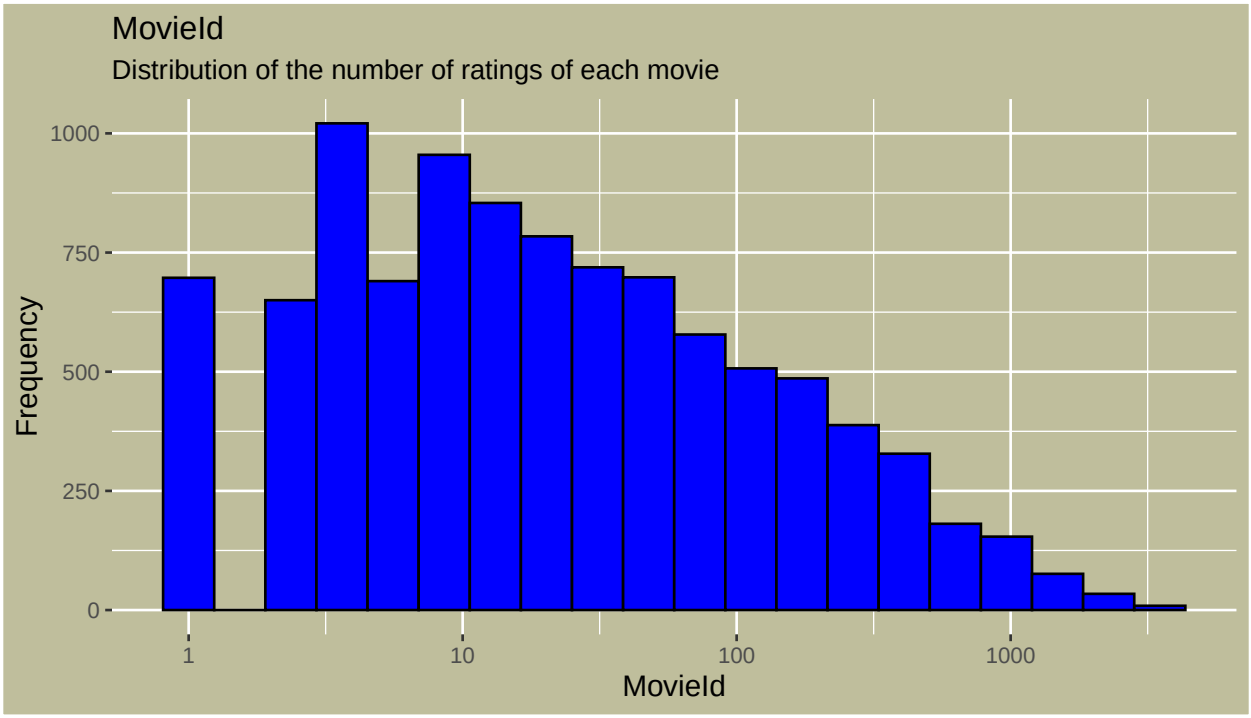
```
p <- 0.5 #(50%)      p = The probability that an event will occur
q <- 0.5 #(50%)      q = The probability that an event will not occur
e <-  0.01 #(0,1%)
z <- 1.65
n <- (p*q)*z^2/e^2
```

2.5 The sample size Plot



2.6 Movie ID movieId Frequency

The movieId frequency, it´s very notorious that are some movieId´s more frequent than others.



Here it is the MovieId independent variable and the Frequency as the dependent variable.

```
b_i_rating_avg <- validation %>%
  group_by(movieId) %>%
  summarise(b_i = mean(rating - mu))
head(b_i_rating_avg)
```

```
## # A tibble: 6 x 2
##   movieId    b_i
##     <dbl>  <dbl>
## 1       1  0.427
## 2       2 -0.281
## 3       3 -0.330
## 4       4 -0.683
## 5       5 -0.358
## 6       6  0.280
```

2.7 Sci-Fi and Film-Noir to SciFI and FilmNoir

Some fix in Sci-Fi and Film-Noir

```
validation %>% separate_rows(genres) %>% distinct(genres)
```

```
## # A tibble: 21 x 1
##    genres
##    <chr>
##  1 Comedy
##  2 Action
##  3 Adventure
##  4 Sci
##  5 Fi
##  6 Thriller
##  7 Children
##  8 Drama
##  9 Romance
## 10 War
## # ... with 11 more rows
```

```
# Rating average and Rating Standard Error

validation <- validation %>% separate_rows(genres)
validation$genres[validation$genres == "Sci-Fi"] <- "SciFi"
validation$genres[validation$genres == "Sci"] <- "SciFi"
validation$genres[validation$genres == "Fi"] <- "SciFi"
validation$genres[validation$genres == "Film-Noir"] <- "FilmNoir"
validation$genres[validation$genres == "Film"] <- "FilmNoir"
validation$genres[validation$genres == "Noir"] <- "FilmNoir"

head(validation)
```

```
## # A tibble: 6 x 8
##   userId movieId rating timestamp title         genres date                year
##    <int>   <dbl>  <dbl>     <int> <chr>         <chr>  <dttm>              <dbl>
## 1      1     231      5 838983392 Dumb & Dumbe~ Comedy 1996-08-02 10:56:32  1994
## 2      1     480      5 838983653 Jurassic Par~ Action 1996-08-02 11:00:53  1993
## 3      1     480      5 838983653 Jurassic Par~ Adven~ 1996-08-02 11:00:53  1993
## 4      1     480      5 838983653 Jurassic Par~ SciFi  1996-08-02 11:00:53  1993
## 5      1     480      5 838983653 Jurassic Par~ SciFi  1996-08-02 11:00:53  1993
## 6      1     480      5 838983653 Jurassic Par~ Thril~ 1996-08-02 11:00:53  1993
```

3 RMSE Preparation

3.1 The Rating Average of Each Movie b_i_rating_avg

3.512033 it´s the mean validation rating:

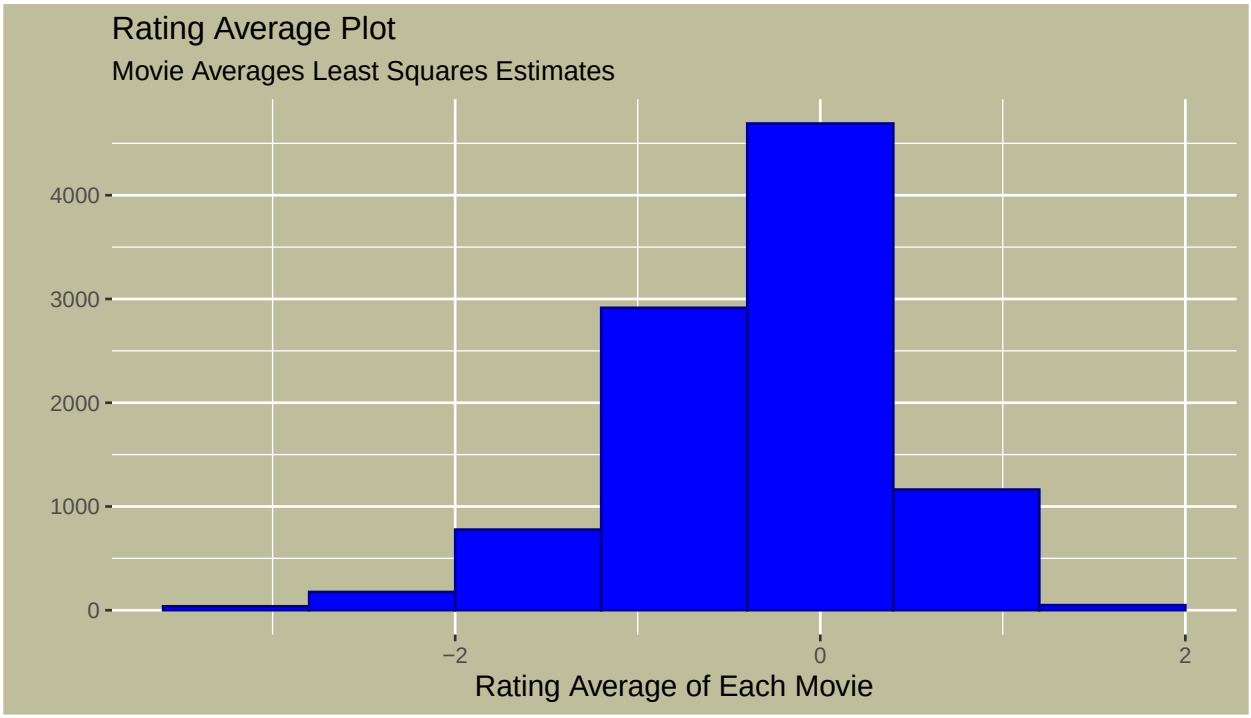
```
## [1] 3.521781
```

The Rating Average of Each Movie: Here it´s the head validation rating of each registration, With the rating average of each movie, it´s significant for the RMSE formula, so for that reason it´s bresented this value.

```
head(b_i_rating_avg)
```

```
## # A tibble: 6 x 2
##   movieId    b_i
##     <dbl>  <dbl>
## 1       1  0.417
## 2       2 -0.291
## 3       3 -0.340
## 4       4 -0.693
## 5       5 -0.368
## 6       6  0.270
```

The Rating Average of Each Movie Plot: The b_i_rating_avg it´s the Rating Average of Each Movie and Percentile Distribution



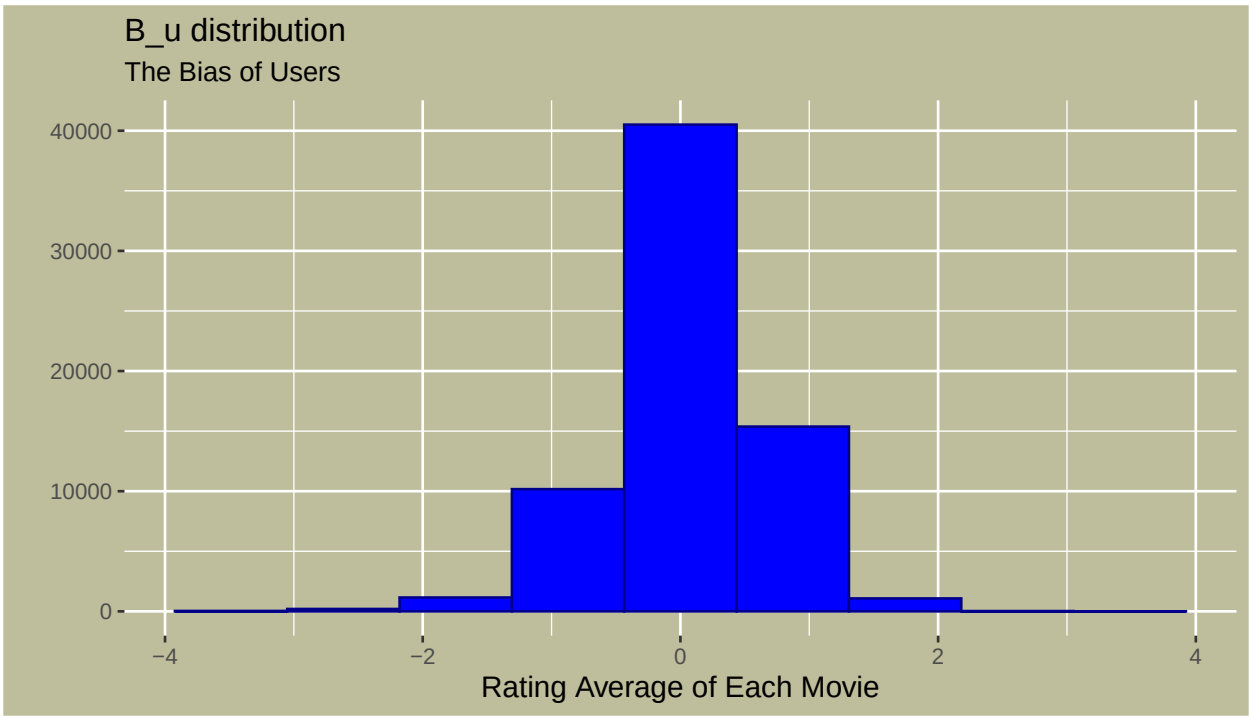
3.2 The User Average of each userId: b_u_user_avg

The User Average: Shows the average of each user.

```
head(b_u_user_avg)
```

```
## # A tibble: 6 x 2
##   userId    b_u
##    <int>  <dbl>
## 1      1  1.58
## 2      2 -0.545
## 3      3  0.176
## 4      4  0.593
## 5      5 -0.316
## 6      6  0.266
```

The User Average of Each User Plot: Here it is the rating average of movies.



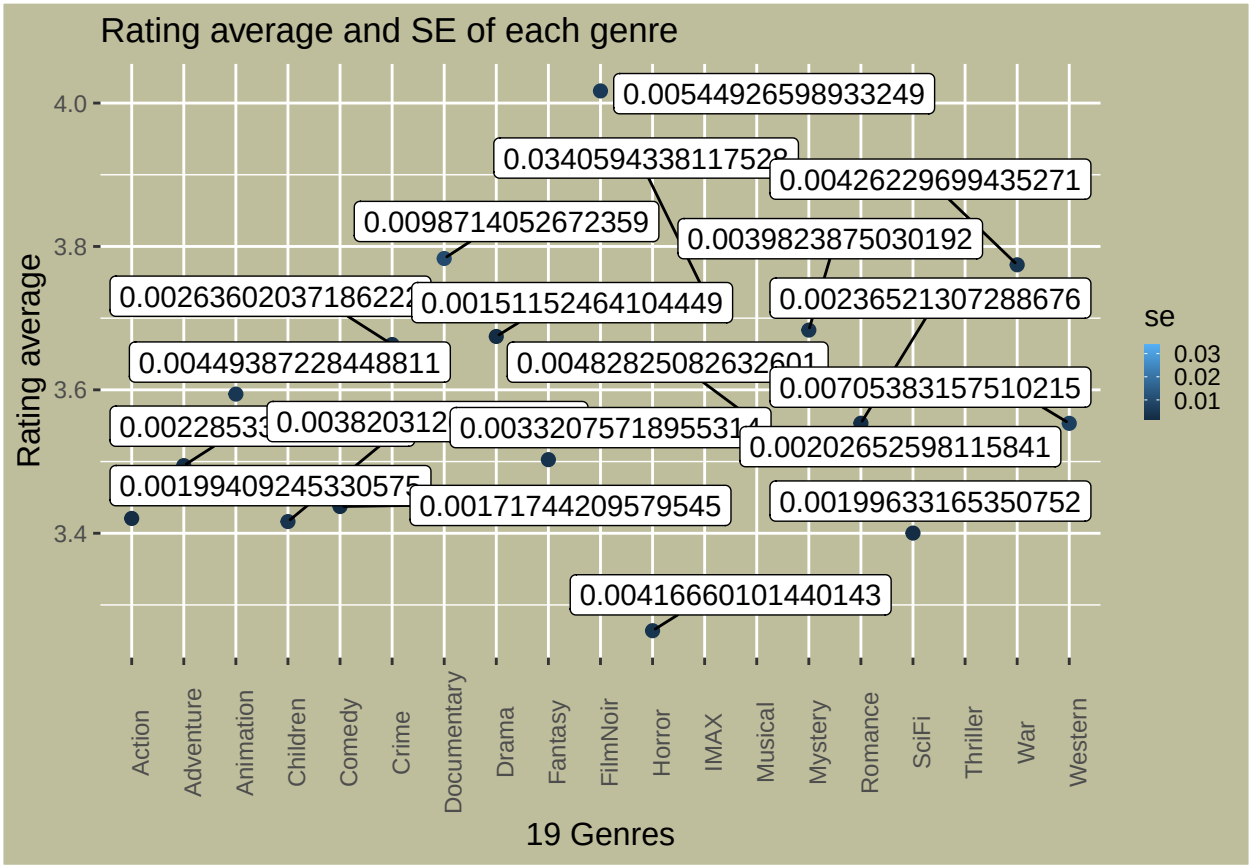
Averages are necessary to take an objective prediction:

- The Rating Average of Each Movie: b_i_rating_avg
- The User Average of each userId: b_u_user_avg

Until here average movie and userId averages are represented in the data.

3.3 The Rating average and SE of each genre & plot

```
## # A tibble: 19 x 4
##    genres           n average      se
##    <chr>        <int>   <dbl>   <dbl>
##  1 Action      284804    3.42 0.00199
##  2 Adventure   212182    3.49 0.00229
##  3 Animation    51944    3.59 0.00449
##  4 Children     82155    3.42 0.00382
##  5 Comedy      393138    3.44 0.00172
##  6 Crime       147242    3.66 0.00264
##  7 Documentary  10388    3.78 0.00987
##  8 Drama       434071    3.67 0.00151
##  9 Fantasy     102845    3.50 0.00332
## 10 FilmNoir     26102    4.02 0.00545
## 11 Horror       76740    3.26 0.00417
## 12 IMAX           899    3.74 0.0341
## 13 Musical      48094    3.56 0.00483
## 14 Mystery      62612    3.68 0.00398
## 15 Romance     189783    3.55 0.00237
## 16 SciFi       298612    3.40 0.00200
## 17 Thriller    258536    3.50 0.00203
## 18 War          56916    3.77 0.00426
## 19 Western      21065    3.55 0.00705
```



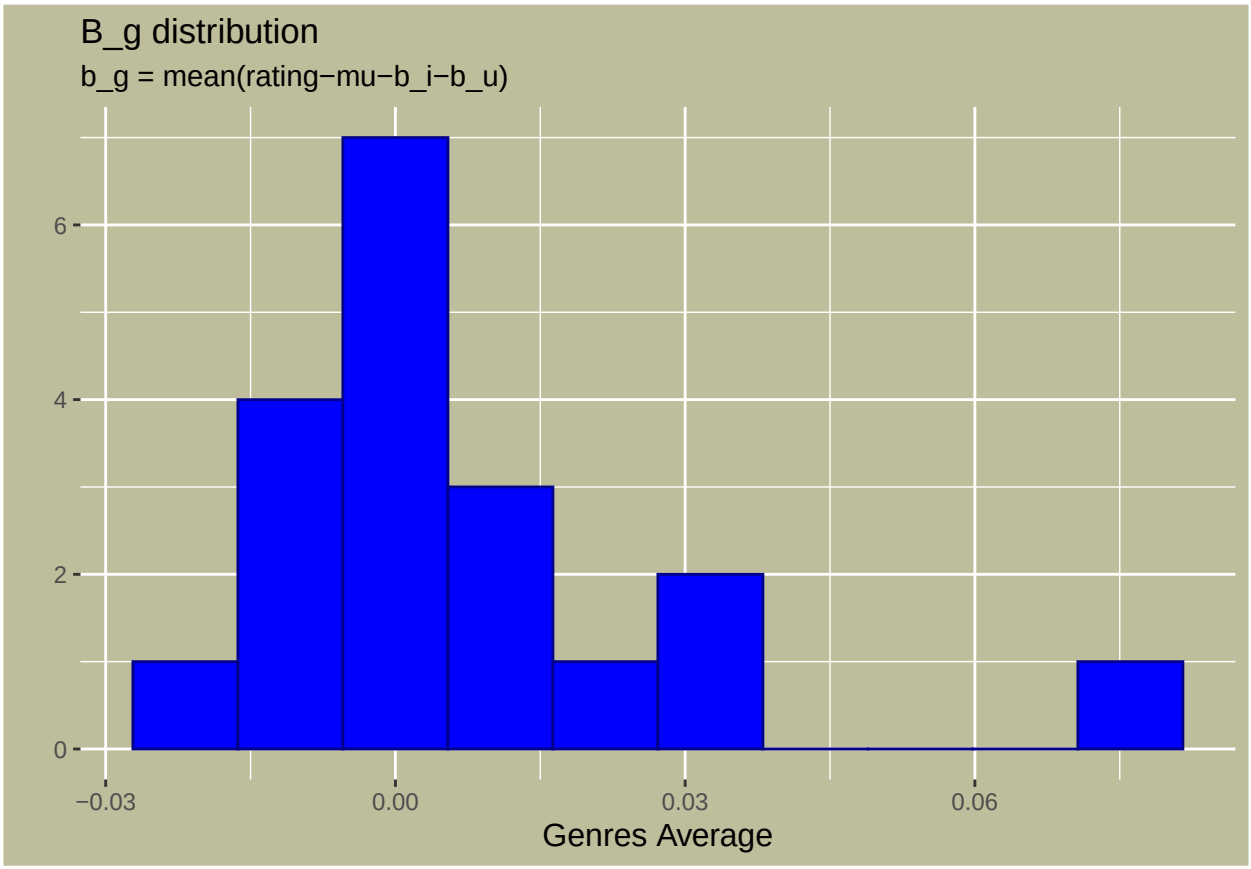
3.4 Distribution of b_g

Now it´s necessary to obtain the average of rating minus mu, b_i and b_u, taking the others averages, now b_g is added to the team:

- The Rating Average of Each Movie: **b_i_rating_avg**
- The User Average of each userId: **b_u_user_avg**
- The Rating less mu less b_i and less b_u : **b_g_genres_avg**

```
## # A tibble: 6 x 2
##   movieId       b_u
##     <dbl>     <dbl>
## 1       1  3.05e-17
## 2       2 -2.05e-17
## 3       3  2.68e-17
## 4       4  2.50e-18
## 5       5 -5.69e-18
## 6       6  3.94e-17
```

Distribution of b_g averages



Now all the variables that were created are accepted to make a prediction

4 Prediction

Taking all these: mu + b_i + b_u + b_g, now it´s prediction:

Prediction Result

Here it is the prediction head and summary result:

```
## [1] 4.540275 5.217528 5.216096 5.217288 5.217288 5.226295
```

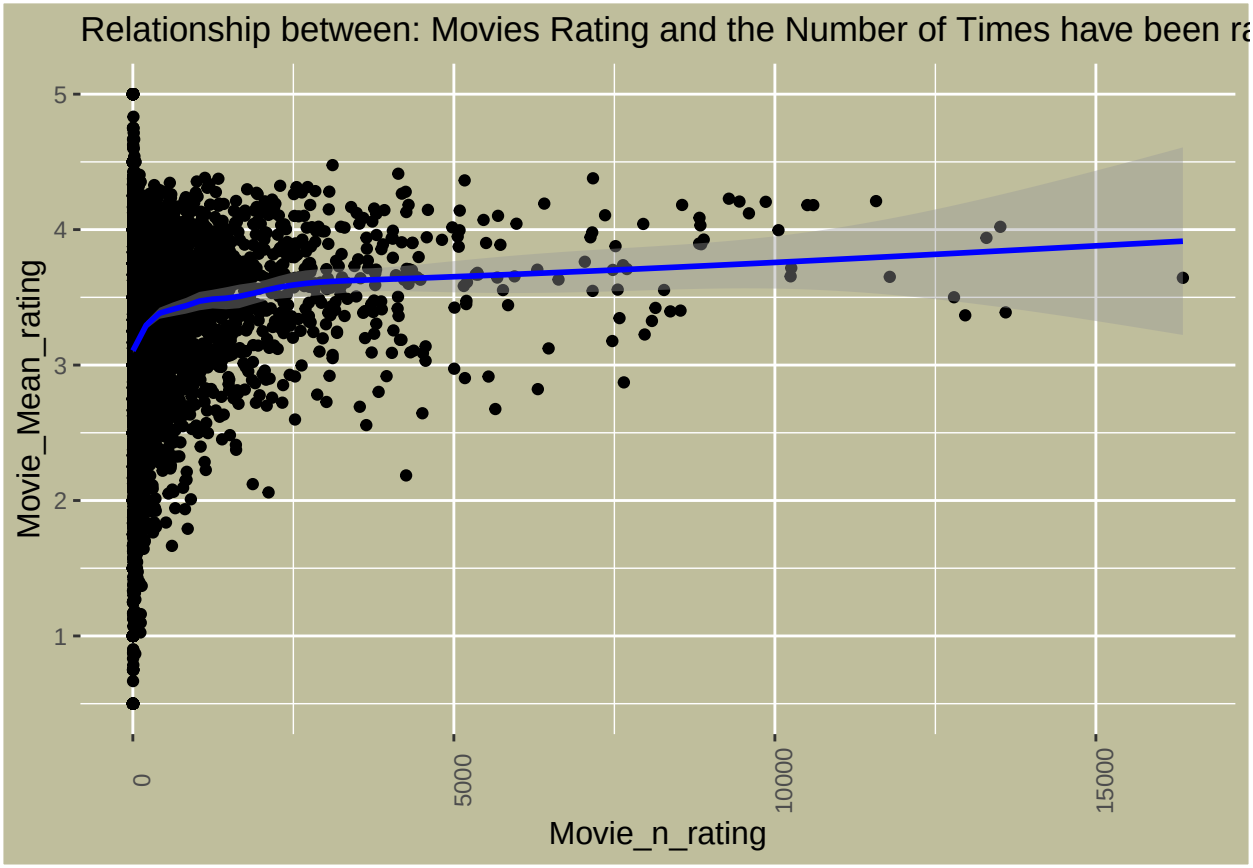
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -1.623   3.115   3.569   3.522   3.979   6.586
```

5 Regularization

Good movies are attracted by users, these will watch and rate each movies, movies with few viewers generate variable results.

This plot shows that the rating mean and number of movie per quantity of n_rating quantity, this is the relationship between the movies rating and the number of times have been rated.

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



5.1 Residuals or Prediction errors

Create a residuals data frame that is residuals, that is the rating minus mu + b_i + b_u + b_g, with this difference shows how concentrated the data is around the best fit or how far the data points are in this case:

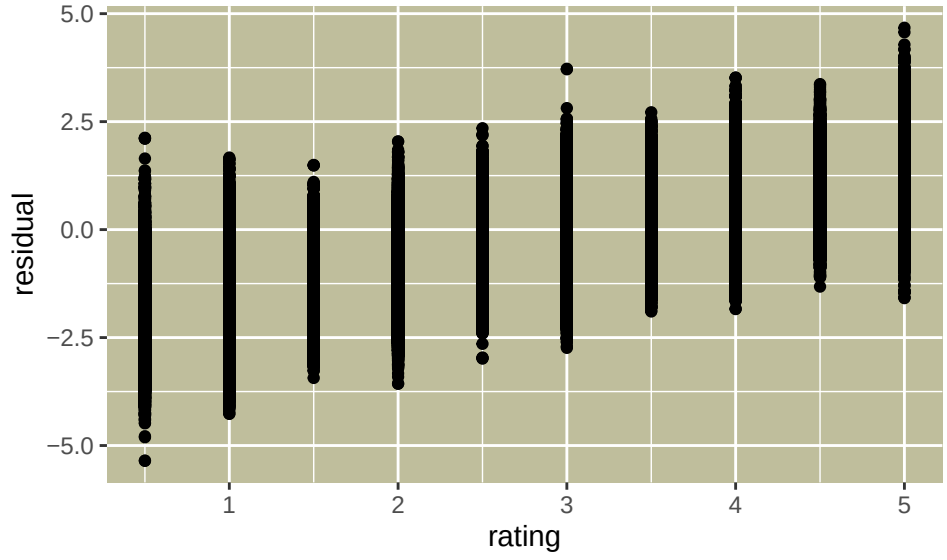
$$error = actual - predicted$$

This is the same as:

$$residual = rating - predicted(mu + b_i + b_u + b_g)$$

```
## # A tibble: 6 x 12
##   userId movieId rating timestamp title   genres date                 year   b_u
##    <int>   <dbl>  <dbl>     <int> <chr>   <chr>  <dttm>              <dbl> <dbl>
## 1      1     231      5 838983392 Dumb &~ Comedy 1996-08-02 10:56:32  1994  1.58
## 2      1     480      5 838983653 Jurass~ Action 1996-08-02 11:00:53  1993  1.58
## 3      1     480      5 838983653 Jurass~ Adven~ 1996-08-02 11:00:53  1993  1.58
## 4      1     480      5 838983653 Jurass~ SciFi  1996-08-02 11:00:53  1993  1.58
## 5      1     480      5 838983653 Jurass~ SciFi  1996-08-02 11:00:53  1993  1.58
## 6      1     480      5 838983653 Jurass~ Thril~ 1996-08-02 11:00:53  1993  1.58
## # ... with 3 more variables: b_i <dbl>, b_g <dbl>, residual <dbl>
```

```
residuals %>% arrange(desc(abs(residual))) %>%
    ggplot(aes(rating, residual))+
    geom_point(aes(rating, residual))+
  theme(panel.background = element_rect(fill = "#bfbe9c"))
```



5.2 Basic RMSE

To obtain a Basic RMSE, take the rating and the mean of rating, both in validation data set.

The basic RMSE it´s 1.054861

```
basic_rmse
```

```
## [1] 1.054861
```

```
rmse_res
```

```
## # A tibble: 1 x 2
##   method                RMSE
##   <chr>                <dbl>
## 1 Just the mean result  1.05
```

5.3 Improvement

Thanks to all the vectors that were created before, now it´s possible to obtain the final RMSE, taking all the average that were obtained:

- The Rating Average of Each Movie: **b_i_rating_avg**
- The User Average of each userId: **b_u_user_avg**
- The Rating less mu less b_i and less b_u : **b_g_genres_avg**

Now the results are:

Predicted Ratings Summary

```
summary(predicted_ratings)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.500   3.240   3.596   3.522   3.892   5.000
```

```
  final_model_rmse
```

```
## [1] 0.9377159
```

The single or Basic RMSE result it´s 1.05, but with The movie effect Model, the result it´s getting better:

```
  rmse_res
```

```
## # A tibble: 2 x 2
##   method                   RMSE
##   <chr>                   <dbl>
## 1 Just the mean result   1.05
## 2 The movie effect Model 0.938
```

6 Final RMSE

0.81: Now taking Movie, User and Genre Effects Model, the RMSE it´s better obtaining a 0.81 result.

```
rmse_res
```

```
## # A tibble: 3 x 2
##   method                              RMSE
##   <chr>                              <dbl>
## 1 Just the mean result               1.05
## 2 The movie effect Model             0.938
## 3 Movie + User + Genre Effects Model 0.818
```

7 Conclusion

Based on The Sample Mean Square Error or RMSE taking the averages of Movie, User and the Genre Effects, makes a Prediction Model Performance Using data analysis packages, obtaining a good RMSE result or approximation.

The limitations in the RMSE result, even the same as the random forest, the RMSE needs more variables or columns to obtain a better result, more numerical variables to obtain a better RMSE. Other limitations was the time to execute the data code, but this was necessary the large data to obtain a good RMSE.

In the future, to obtain a better result, it´s necessary to obtain more numerical variables or columns to obtain more averages. It´s important to have a larger data, all these to doing a matrix factorization, that can improve the model and the result.