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

Carlos Dominguez Monferrer

July 30th, 2020

Contents

1 Executive summary

The key idea of the project is to create a movie recommendation system, using all the tools that have been used in previous courses of the Data Science Professional Certificate.

Recommendation systems use ratings that users have given items to make specific recommendations. Companies like Netflix or HBO are able to collect massive datasets that can be used to predict what rating a particular user will give a specific item. Items for which a high rating is predicted for a given user are them recommended to that user.

Using this definition of recommendation systems as a basis, a 10M version of the MovieLens dataset will be used. With this dataset, another two sets will be created too: the edx set, to develop the algorithm, and the validation set, to check the algorithm. To evaluate how close our predictions are to the true values in the validation set, a RMSE (Root Mean Squared Error) will be used.

2 Methods/Analysis

Before creating and optimizing the algorithm, an analysis of MovieLens dataset is needed to know the type of data we will work with and the influence of the different variables on ratings.

In order to make the code easier to understand, the Analysis section has been divided in two parts:

Data exploration:

- Number of rows and columns
- Name of the variables
- Summary of edx and validations sets
- Number of different movies and users

Data cleaning and Influence of variables on ratings:

- Convert Timestamp variable to a date.
- Relation between Timestamp and Ratings.
- Extract the year of the movie that is in the Title column.
- Relation between Year and Ratings.
- Plot a histogram that represent the relation between Mean Rating and Number of users.
- Plot a histogram that represent the relation between Mean Rating and Number of movies.
- Check how many genre types are in the datasets.
- Due to may be more than one genre per movie, we have to separate these genres to facilitate the analysis.
- Relation between Genres and Ratings.

2.1 Data exploration

2.1.1 Number of rows & columns

- Edx dataset

Number of rows

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

Number of columns

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

- Validation dataset

Number of rows

```
## [1] 999999
```

Number of columns

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

2.1.2 Name of the variables

There are 6 different variables in both datasets:

```
## [1] "userId"    "movieId"   "rating"    "timestamp" "title"     "genres"
```

2.1.3 Summary stadistics

- Edx dataset

```
##      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
##
##
##
```

- Validation dataset

```
##      userId         movieId          rating        timestamp
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08
##  1st Qu.:18096   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.467e+08
##  Median :35768   Median : 1827   Median :4.000   Median :1.035e+09
##  Mean   :35870   Mean   : 4108   Mean   :3.512   Mean   :1.033e+09
##  3rd Qu.:53621   3rd Qu.: 3624   3rd Qu.:4.000   3rd Qu.:1.127e+09
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09
##     title              genres
##  Length:999999      Length:999999
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

2.1.4 How many different users and movies are in both datasets

- Edx dataset

Different users

```
## [1] 69878
```

Different movies

```
## [1] 10677
```

- Validation dataset

Different users

```
## [1] 68534
```

Different movies

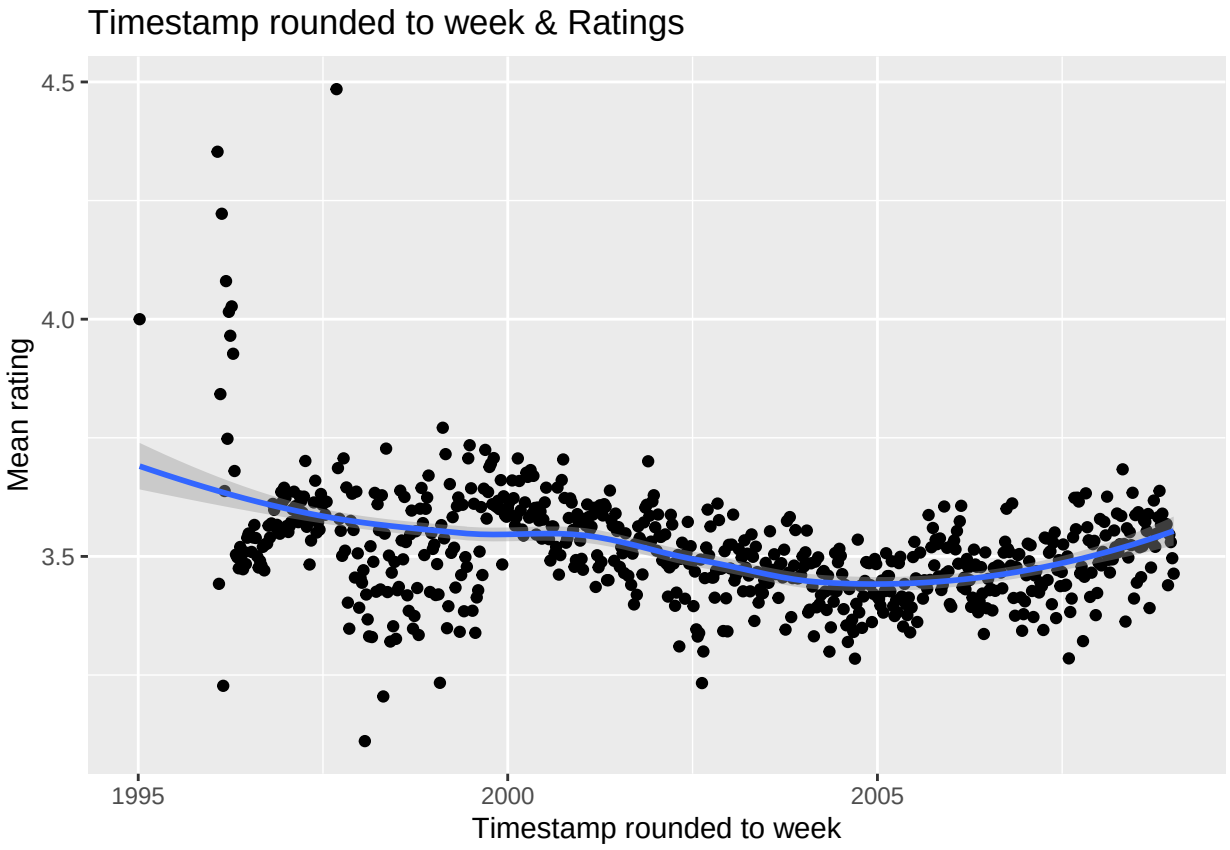
```
## [1] 9809
```

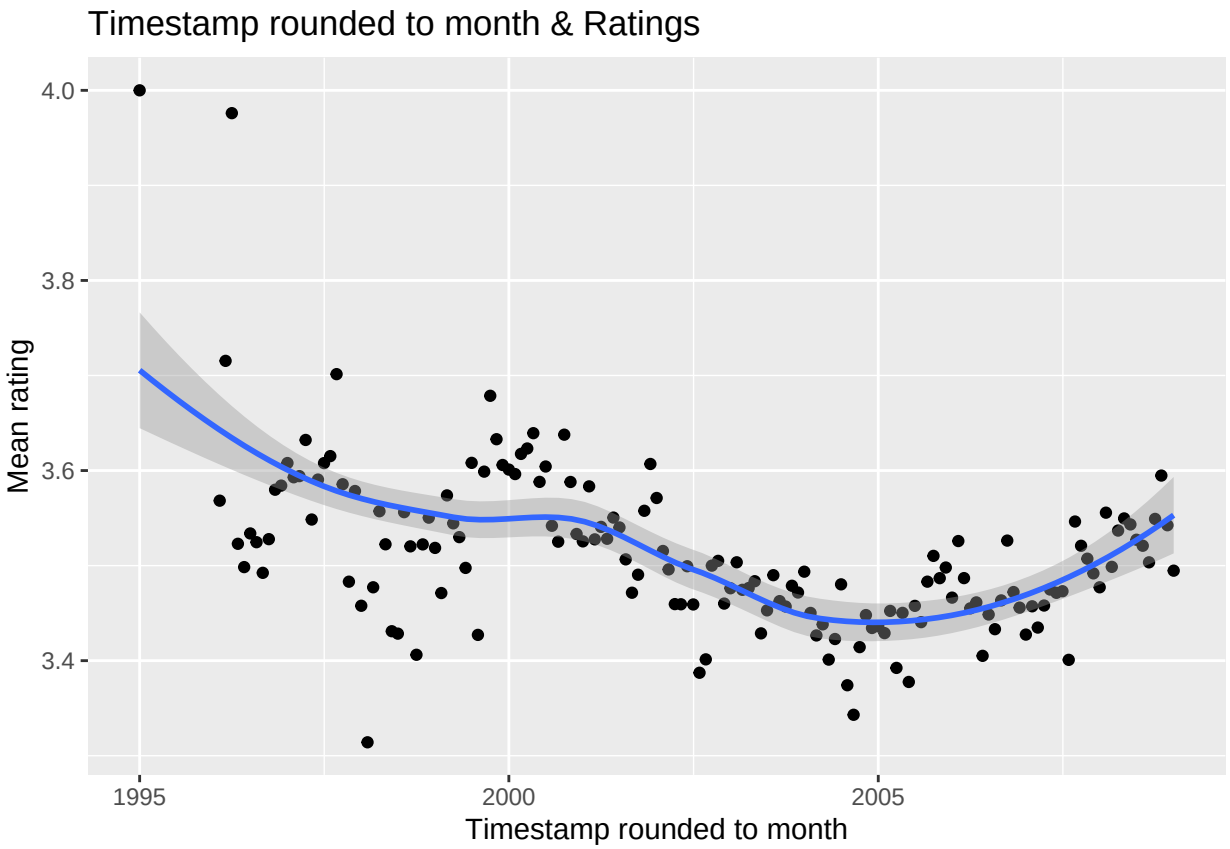
2.2 Data cleaning and Influence of variables on rating

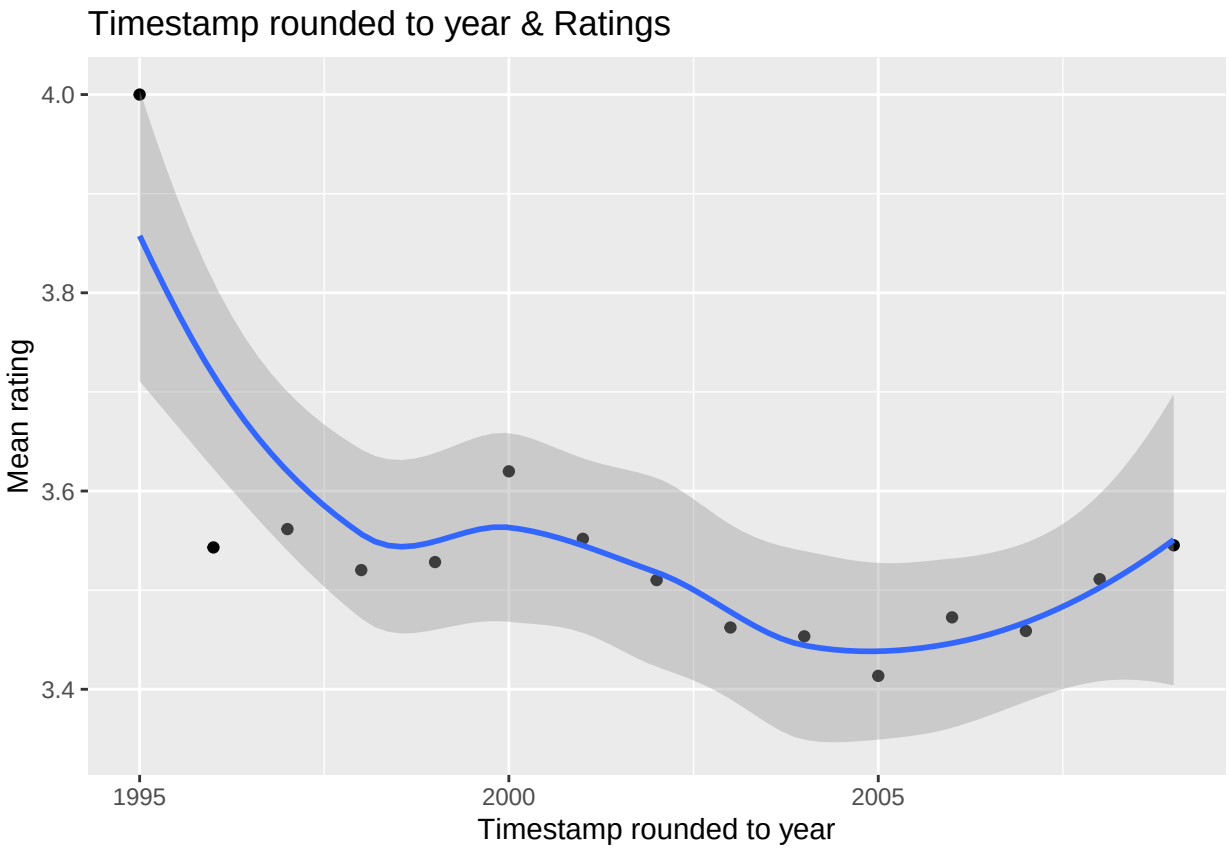
2.2.1 Timestamp & Ratings

In order to do a complete analysis of the influence of Timestamp on Ratings, 3 graphs are plotted:

- Timestamp rounded to week.
- Timestamp rounded to month.
- Timestamp rounded to year.

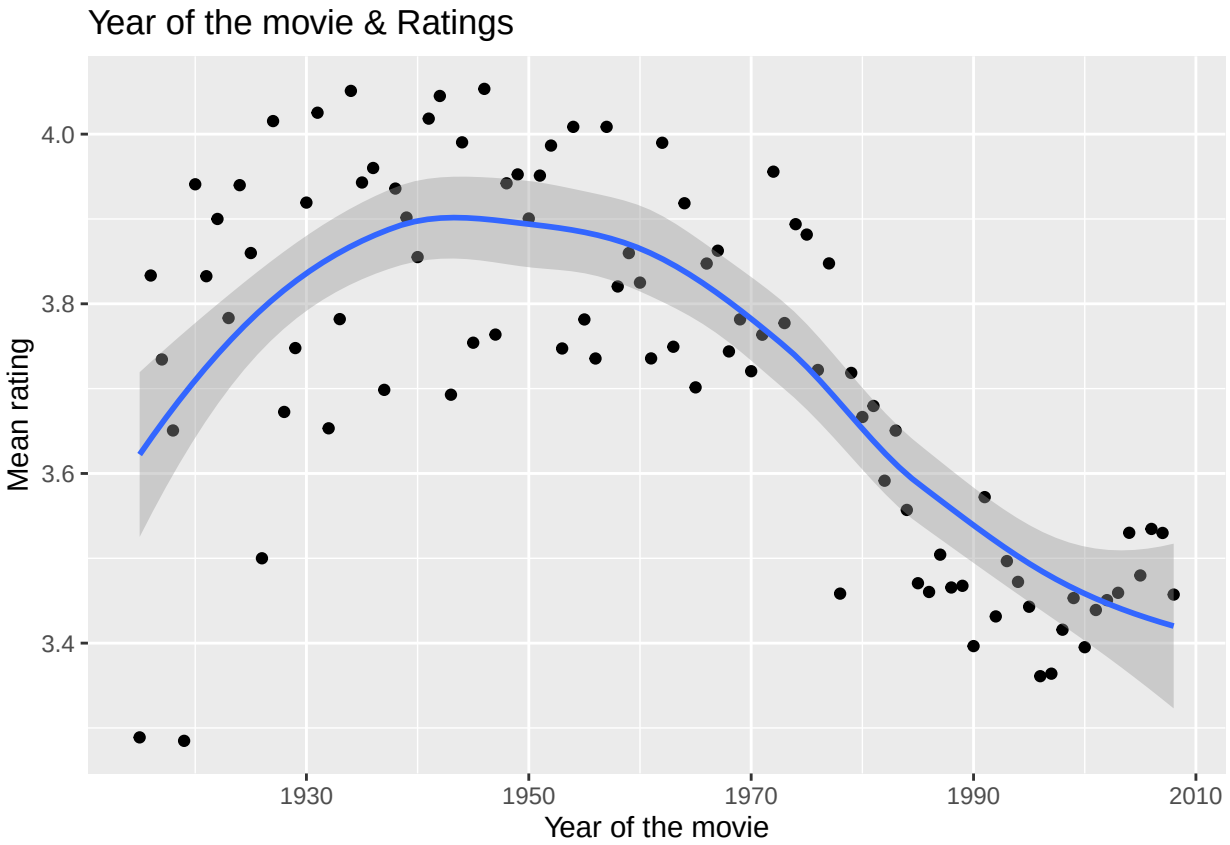



Timestamp rounded to month & Ratings
4.0
3.8
3.6
3.4
Mean rating
1995
2000
2005
Timestamp rounded to month



Conclusion 1.-: There is some evidence of a time effect on average rating.

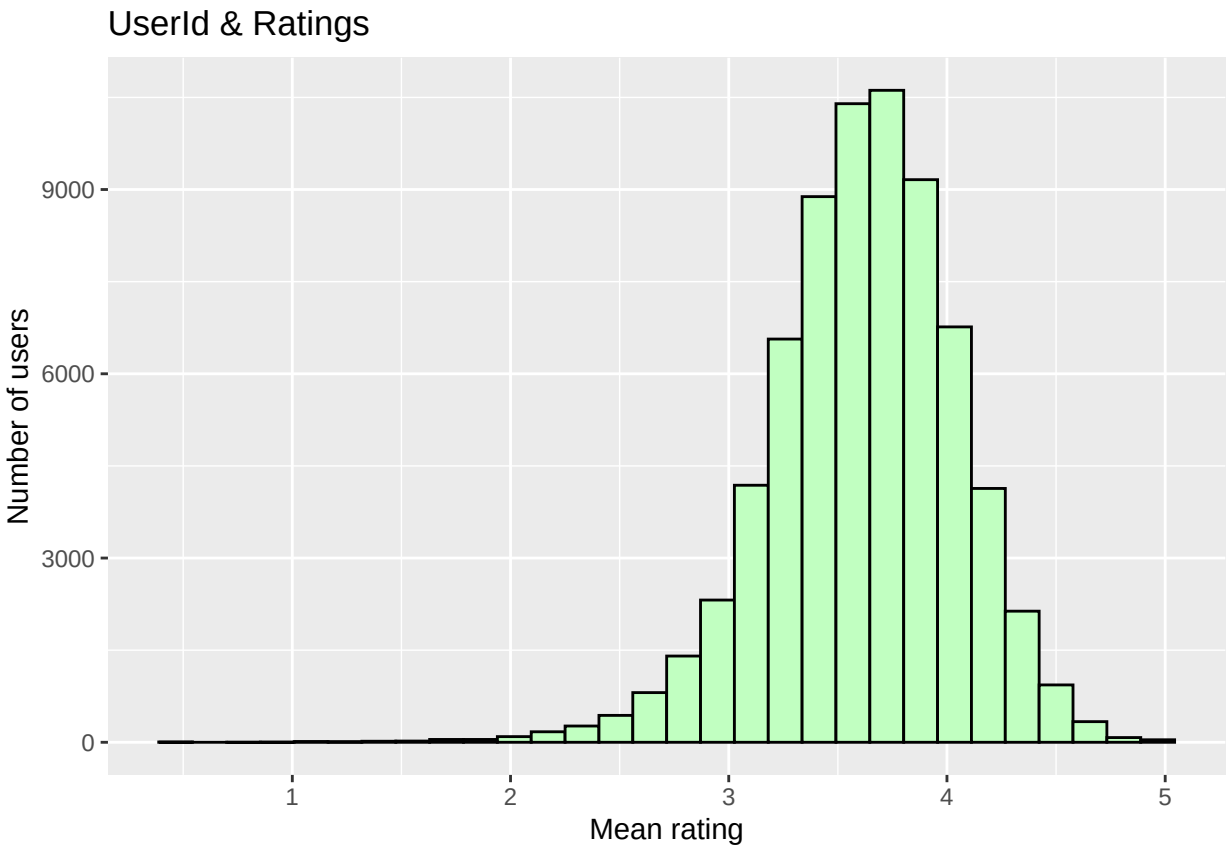
2.2.2 Year & Ratings



Conclusion 2.-: There is strong evidence of a Year effect on average rating. We can see that the films broadcast between 1930 and 1970 have much better score than the current ones.

2.2.3 UserId & Ratings

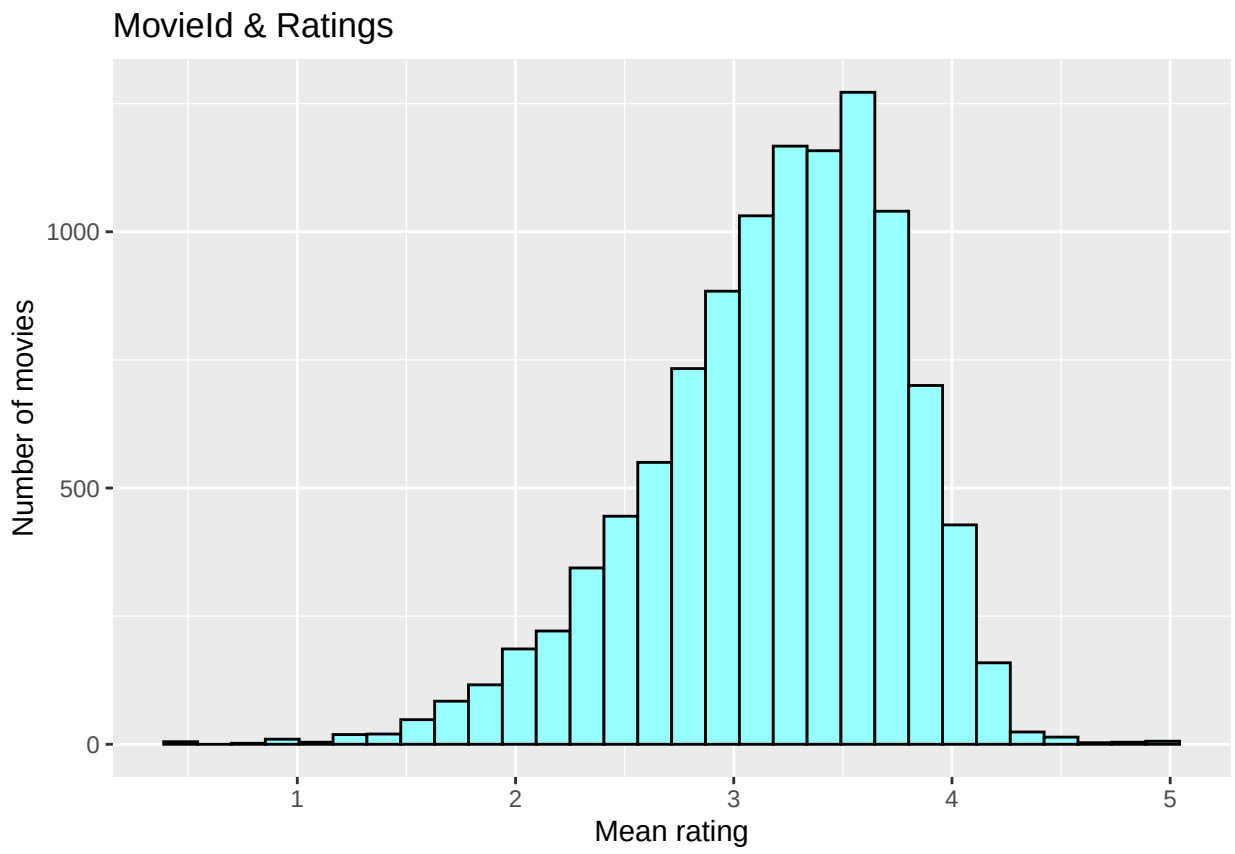
A filter is applied to select the users that have voted, at least, 50 times.



Conclusion 3.-: There is strong evidence of a UserId effect on average rating. Most users rate movies with a score of around 3.5.

2.2.4 MovieId & Ratings

A filter is applied to select the movies that have voted, at least, 15 times.



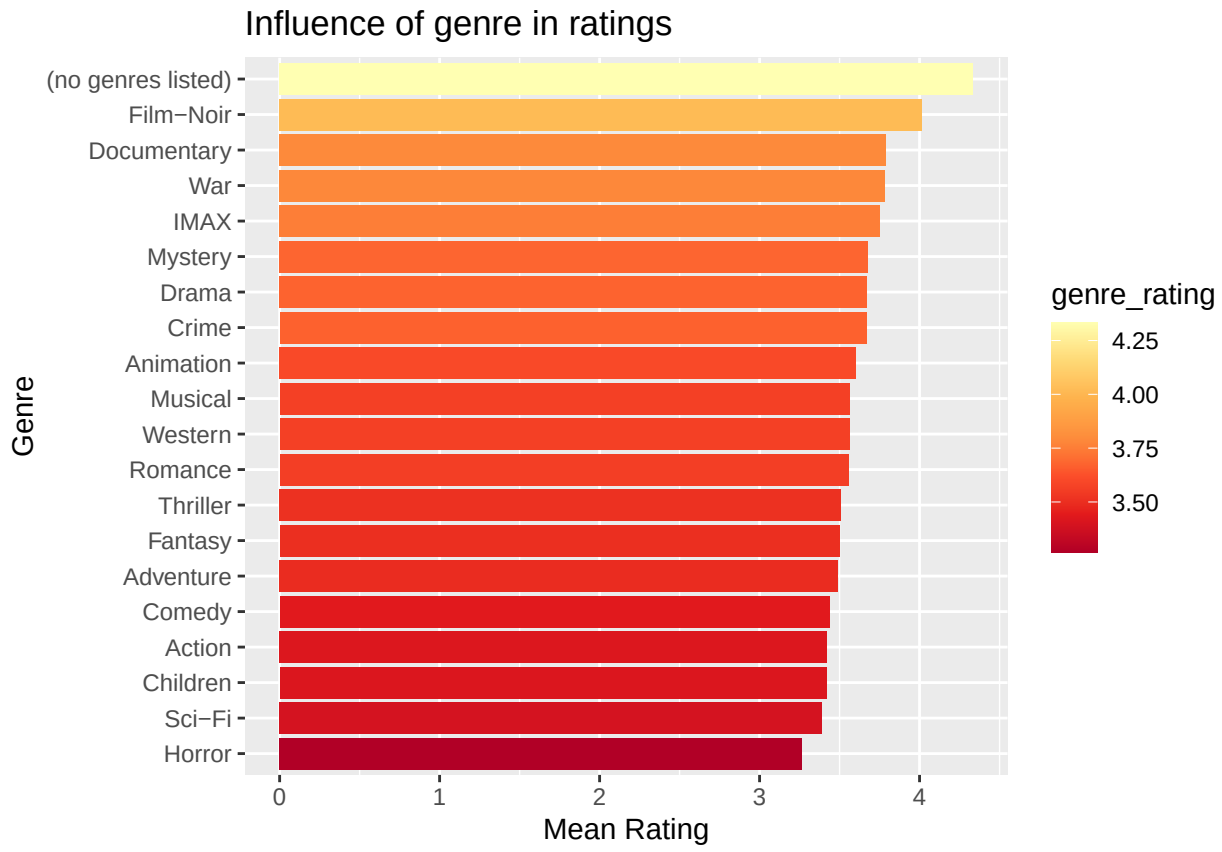
Conclusion 4.-: There is strong evidence of a MovieId effect on average rating. Most films have been rated with a score between 3 and 4.

2.2.5 Genre & Ratings

There are 20 different genres in both datasets:

```
##  [1] "Comedy"             "Romance"            "Action"
##  [4] "Crime"              "Thriller"           "Drama"
##  [7] "Sci-Fi"             "Adventure"          "Children"
## [10] "Fantasy"            "War"                "Animation"
## [13] "Musical"            "Western"            "Mystery"
## [16] "Film-Noir"          "Horror"             "Documentary"
## [19] "IMAX"               "(no genres listed)"
```

Due to we have datasets with more than 9M of observations, createDataPartition function is needed to make the analysis easier.



Conclusion 5.-: There is strong evidence of a Genre effect on average rating.

3 Results

3.1 Training process

To train our algotithm, we will calculate first RMSE without regularization technique.

3.1.1 Just the average

```
## # A tibble: 1 x 2
##   Model              RMSE
##   <chr>             <dbl>
## 1 Just the average 1.0609
```

3.1.2 Date effect

```
## # A tibble: 2 x 2
##   Model               RMSE
##   <chr>              <dbl>
## 1 Just the average  1.0609
## 2 Date Effect Model 1.0583
```

3.1.3 Year effect

```
## # A tibble: 3 x 2
##   Model               RMSE
##   <chr>              <dbl>
## 1 Just the average  1.0609
## 2 Date Effect Model 1.0583
## 3 Year Effect Model 1.0500
```

3.1.4 Movie effect

```
## # A tibble: 4 x 2
##   Model                 RMSE
##   <chr>                <dbl>
## 1 Just the average   1.0609
## 2 Date Effect Model  1.0583
## 3 Year Effect Model  1.0500
## 4 Movie Effect Model 0.94403
```

3.1.5 User effect

```
## # A tibble: 5 x 2
##   Model                 RMSE
##   <chr>                <dbl>
## 1 Just the average   1.0609
## 2 Date Effect Model  1.0583
## 3 Year Effect Model  1.0500
## 4 Movie Effect Model 0.94403
## 5 User Effect Model  0.97879
```

Due to User and Movie variables got a RSME < 1, we will combine them in order to check if we can reduce the Root Mean Squared Error.

3.1.6 User + Movie effect

```
## # A tibble: 6 x 2
```

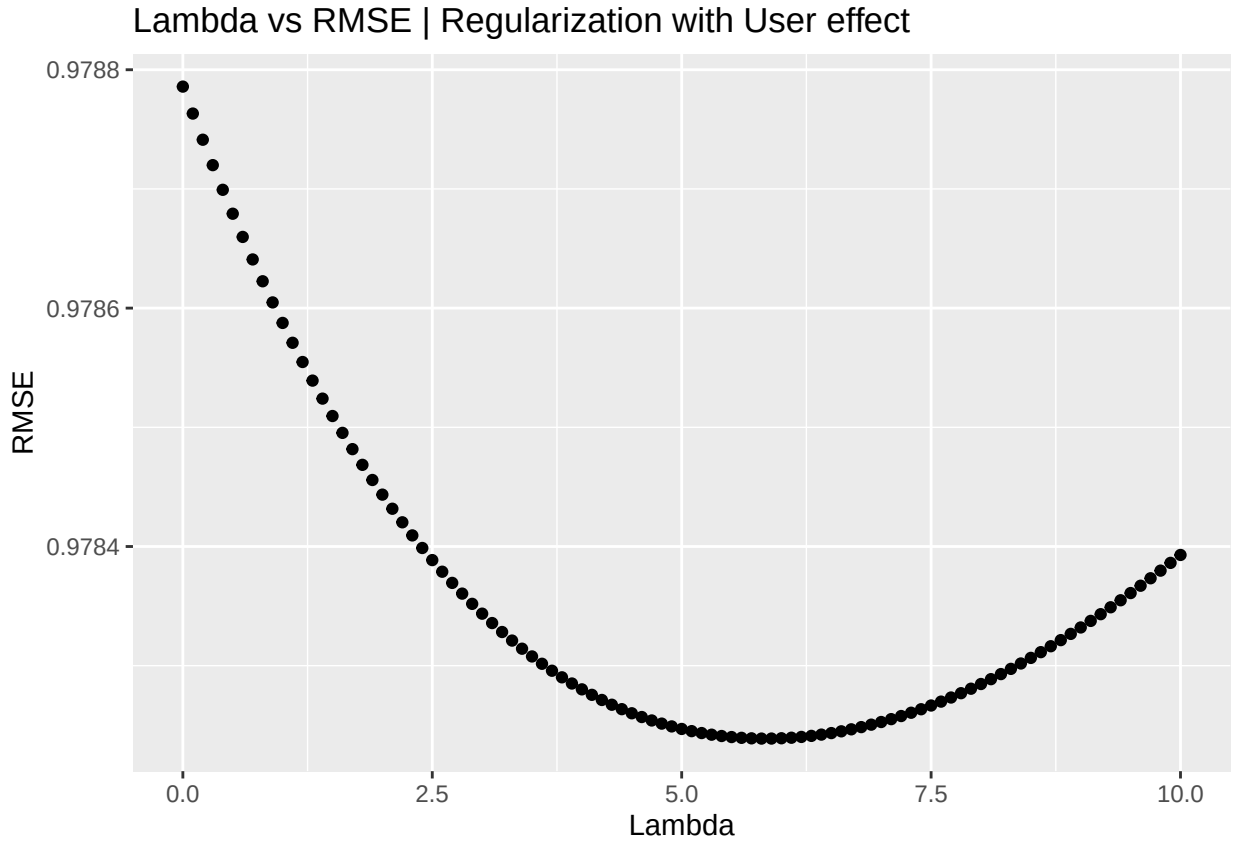
```
##   Model                          RMSE
##   <chr>                         <dbl>
## 1 Just the average           1.0609
## 2 Date Effect Model          1.0583
## 3 Year Effect Model          1.0500
## 4 Movie Effect Model         0.94403
## 5 User Effect Model          0.97879
## 6 User + Movie Effects Model 0.86571
```

Know, we will calculate RMSE with regularization technique.

3.1.7 Regularization with User effect

Lambda value:

```
## [1] 5.8
```

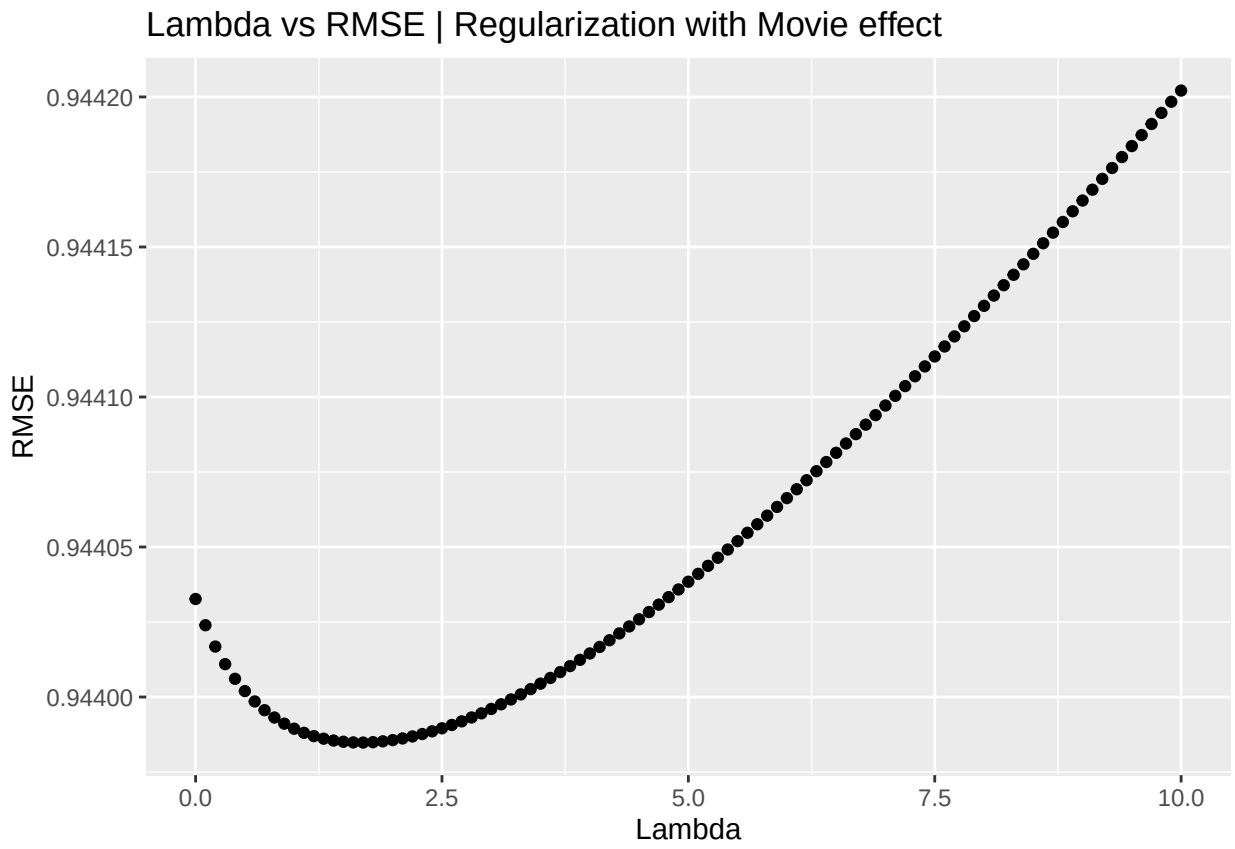



```
## # A tibble: 7 x 2
##   Model                            RMSE
##   <chr>                           <dbl>
## 1 Just the average              1.0609
## 2 Date Effect Model             1.0583
## 3 Year Effect Model             1.0500
## 4 Movie Effect Model            0.94403
## 5 User Effect Model             0.97879
## 6 User + Movie Effects Model    0.86571
## 7 Regularized User Effect Model 0.97824
```

3.1.8 Regularization with Movie effect

Lambda value:

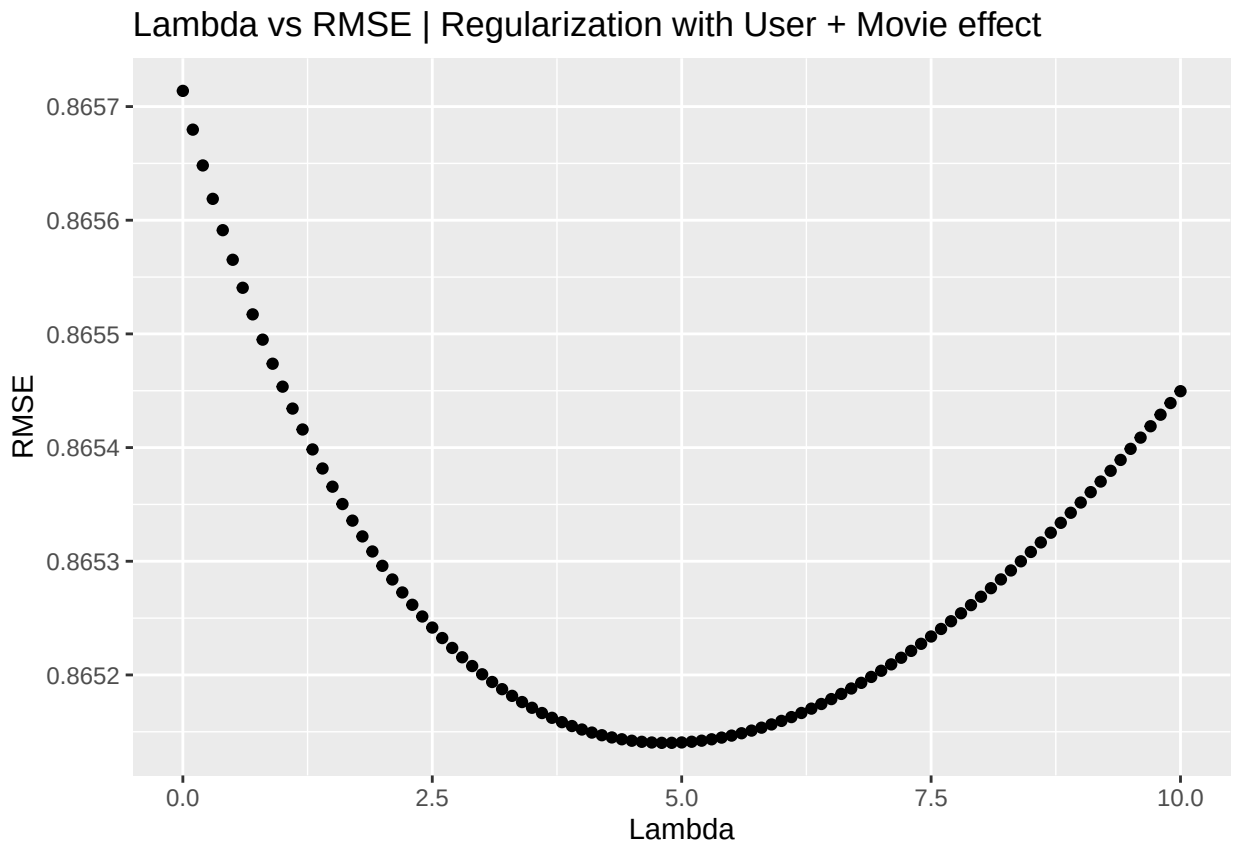
```
## [1] 1.7
```




```
## # A tibble: 8 x 2
##   Model                          RMSE
##   <chr>                         <dbl>
## 1 Just the average              1.0609
## 2 Date Effect Model             1.0583
## 3 Year Effect Model             1.0500
## 4 Movie Effect Model            0.94403
## 5 User Effect Model             0.97879
## 6 User + Movie Effects Model    0.86571
## 7 Regularized User Effect Model  0.97824
## 8 Regularized Movie Effect Model 0.94398
```

Due to User and Movie variables got a RSME < 1, we will combine them in order to check if we can reduce the Root Mean Squared Error with regularization technique.

3.1.9 Regularization with User + Movie effect




```
## # A tibble: 9 x 2
##   Model                                  RMSE
##   <chr>                                 <dbl>
## 1 Just the average                      1.0609
## 2 Date Effect Model                     1.0583
## 3 Year Effect Model                     1.0500
## 4 Movie Effect Model                    0.94403
## 5 User Effect Model                     0.97879
## 6 User + Movie Effects Model            0.86571
## 7 Regularized User Effect Model         0.97824
## 8 Regularized Movie Effect Model        0.94398
## 9 Regularized User + Movie Effects Model 0.86514
```

Analyzing the results, we notice that Regularization with User + Movie effect model give us the smallest RSME. We will use the lambda value of this model to check the RMSE in the Validation set.

Lambda value:

```
## [1] 4.8
```

3.2 Validations process

```
## # A tibble: 10 x 2
##    Model                                  RMSE
##    <chr>                                 <dbl>
##  1 Just the average                      1.0609
```

```
##  2 Date Effect Model                       1.0583
##  3 Year Effect Model                       1.0500
##  4 Movie Effect Model                      0.94403
##  5 User Effect Model                       0.97879
##  6 User + Movie Effects Model              0.86571
##  7 Regularized User Effect Model           0.97824
##  8 Regularized Movie Effect Model          0.94398
##  9 Regularized User + Movie Effects Model 0.86514
## 10 Validation Model                        0.86482
```

The RMSE value in the Validation set is less than 0.86490 so we have achieved our objetive.

Notice that we dont have use techniques like lm, loess, glm, randomforests or knn because there are thousands of different and unique data so these fuctions will be very slow here. Evenmore, there is not enought space in our computers to compute them.

4 Conclusion

The Methods/Analysis section has been necessary to know the type of data we were going to work with. Due to the dataset contained about 10 million data, we have ruled out using techniques such as linear regression because the computation time would be very high.

With the analysis of the influence of variables on ratings, we have seen that the UserId and the MovieId were the most important variables. However, other variables such as the year of the movie or the genre were also important.

In the beginning, RMSEs with basic models, like Just the Average, have been obtained. Due to RSME values were greater than 1, it has been sought to reduce error by using regularization.

It has been concluded that the Regularization with User + Movie effect model has been optimal for the lower RSME value. However, other methods such as matrix factorization could have been used to get lower RMSE.

5 Appendix - Enviroment

```
## [1] "Operating System:"
```

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## system         x86_64, mingw32
## status
## major          3
## minor          6.3
## year           2020
## month          02
## day            29
## svn rev        77875
## language       R
## version.string R version 3.6.3 (2020-02-29)
## nickname       Holding the Windsock
```