MovieLens

Abinav Bhagam

2024-08-22

Contents

1 Introduction

This project - MovieLens - is the the first of the two projects required to pass the *HarvardX - PH125.9x - Data Science: Capstone* course, the finale of the *Data Science Professional Certificate* program.

The objective of this project is the development of a movie recommendation system using the Movie Lens data set. Recommendation systems operate by analyzing previous choices/preferences in order to *recommend* new suggestions.

The movielens dataset provided for this report contains approximately 10 million movie observations, bifurcated into a training set (`edx`) and a validation set (`final_holdout_test`) with 9 million and 1 million observations respectively. The code required to construct the sets has been provided below:

Following a brief analysis of the dataset, we will aim to construct a recommendation system with a Root Mean Square Error (RSME) below 0.86490 for the estimated ratings of the user-movie pairs.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$$

2 Analysis of the Dataset

2.1 Description of the Data

The `edx` and `final_holdout_test` sets have a respective 9000055 and 999999 observations, with each having the same 6 variables.

Let's take a look at the `edx` dataset.

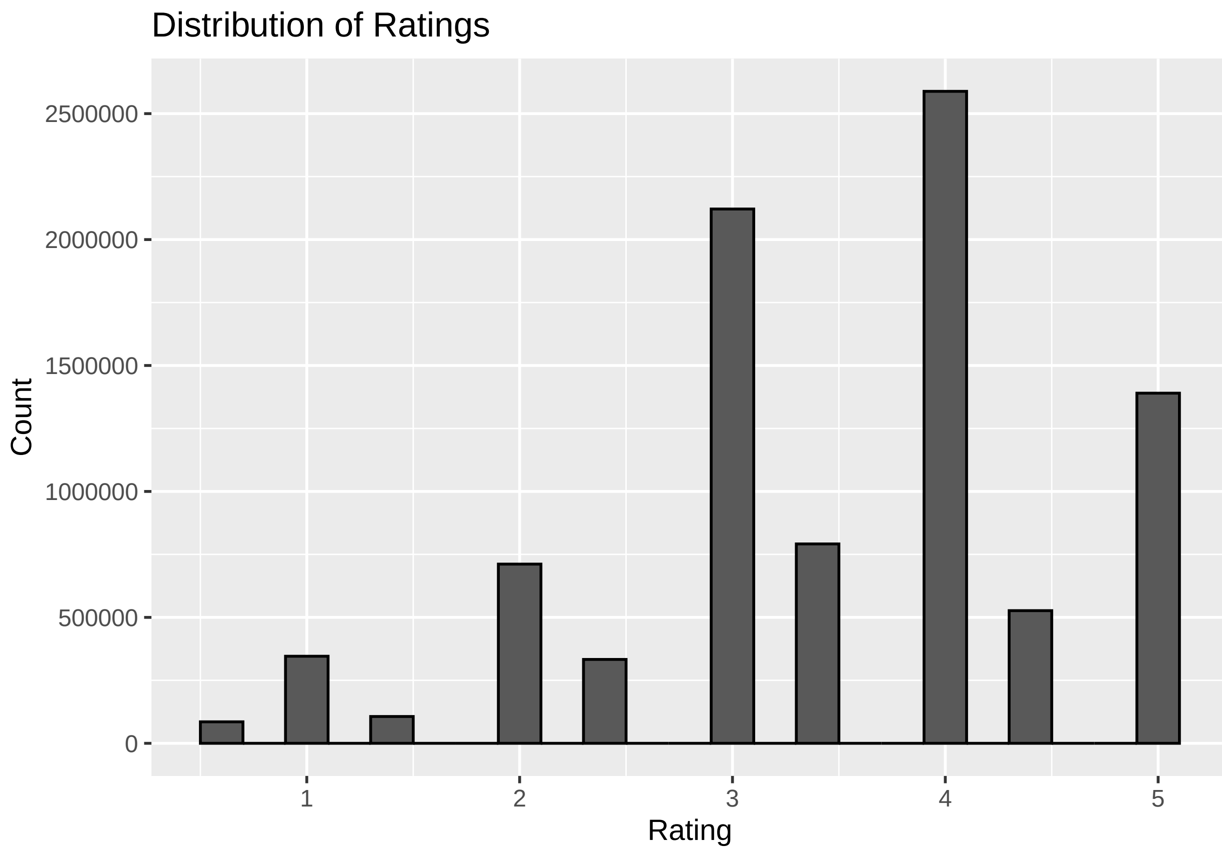
```
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 4      1     292      5 838983421               Outbreak (1995)
## 5      1     316      5 838983392               Stargate (1994)
## 6      1     329      5 838983392 Star Trek: Generations (1994)
##                           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
```

Let's take a look at what exactly these variables denote.

- `userId`: Unique identifier that marks the user who made the rating
- `movieId`: Unique identifier that marks the movie was rated
- `rating`: A number ranging from 0 to 5 - with 0.5 increments - that represents the quality of the movie
- `timestamp`: The date/time of the rating
- `title`: The title of the movie and the year it came out
- `genres`: The genre - or genres - the movie belongs to

Let's examine each of these variables, one by one.

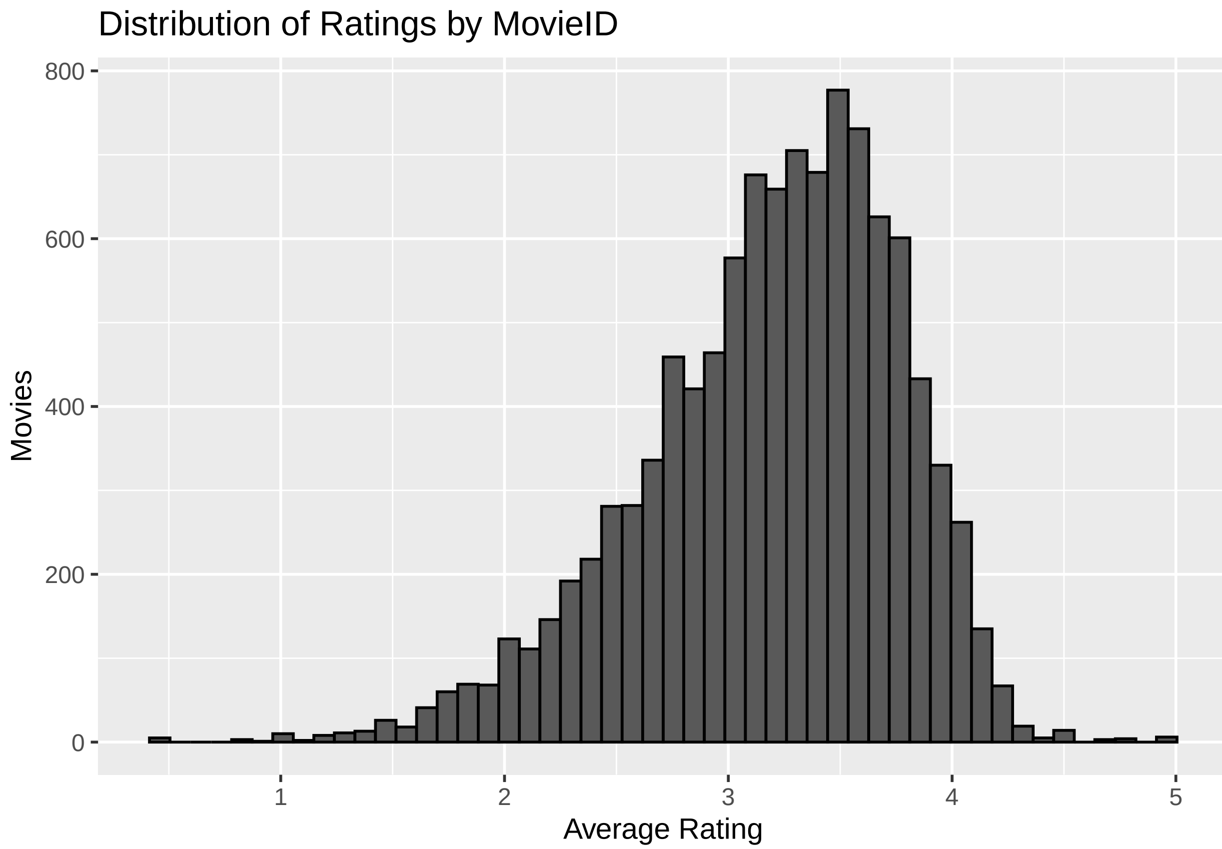
2.2 Rating



One thing that immediately pops out from looking at the histogram, is that users seem to gravitate to rating whole numbers. 1 is more frequent than 1.5, 2 is more frequent that 2,5 and so on.

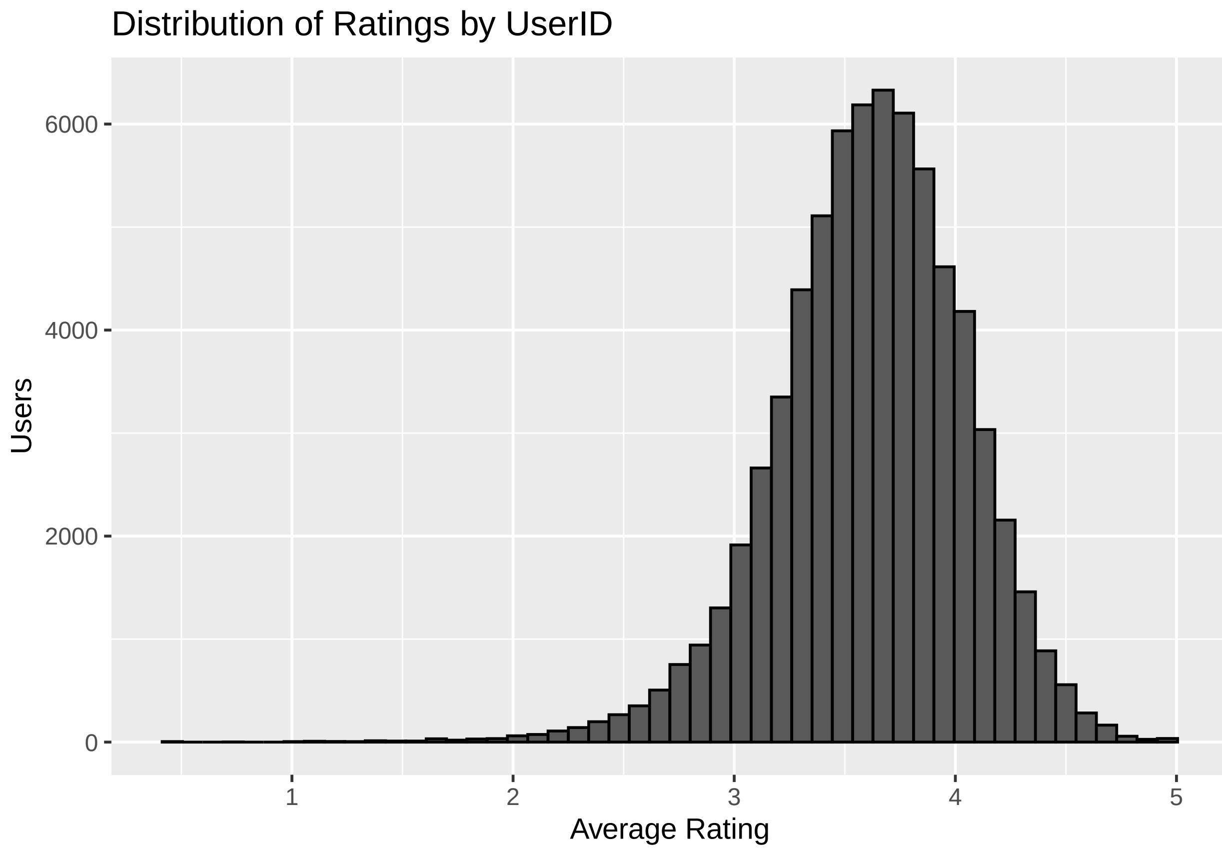
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.500   3.000   4.000   3.512   4.000   5.000
```

2.3 MovieID



The histogram is self-evident. Not all movie are valued the same. Some are rated higher or lower than others. Considering that the mean rating is 3.51, it seems that even the most average movie is viewed generously.

2.4 UserID



UserID seems to follow a similar pattern to MovieID, the average user is rather generous. Perhaps this generosity is inadvertently causing the recommendation system to develop biases. That is something that should be adjusted.

2.5 Genres

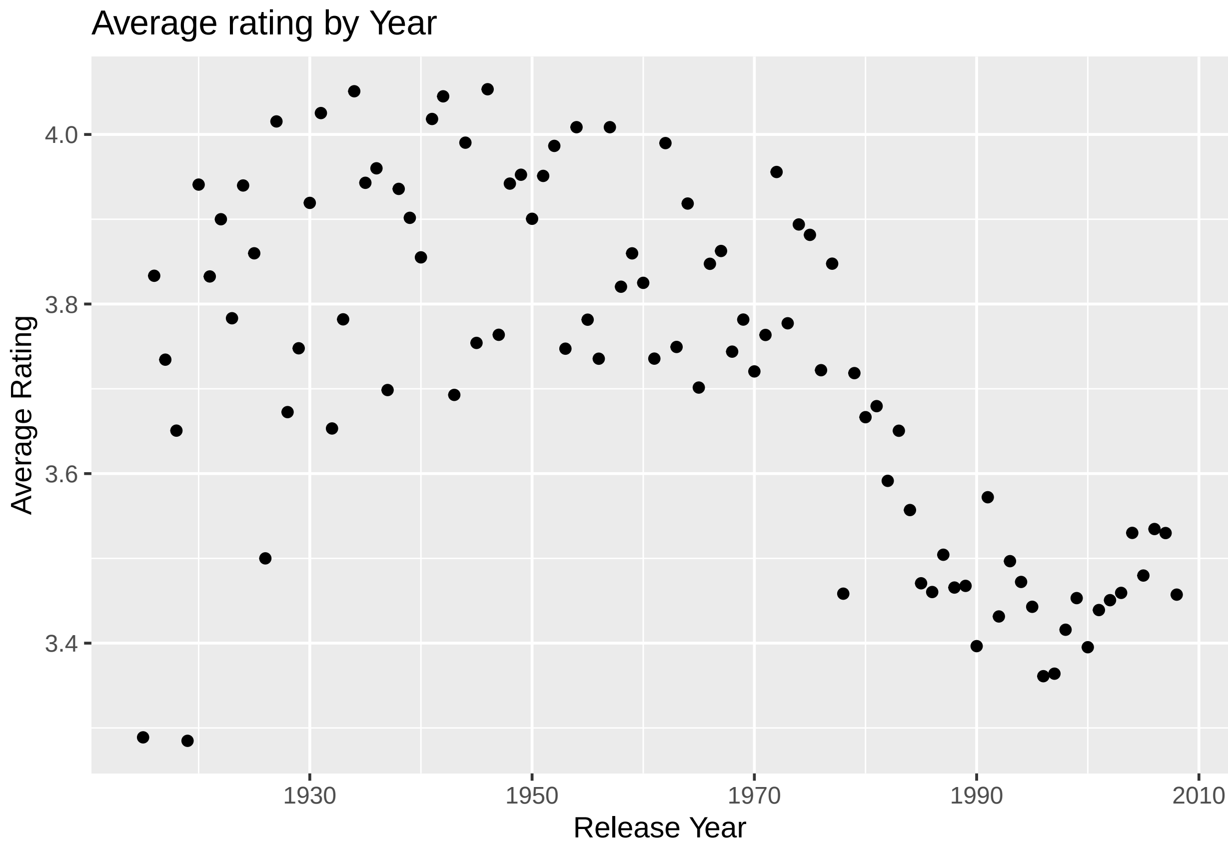
```
## # A tibble: 20 x 3
##    genres             Number_of_Movies Rating
##    <chr>                         <int>  <dbl>
##  1 Drama                       3910127   3.67
##  2 Comedy                      3540930   3.44
##  3 Action                      2560545   3.42
##  4 Thriller                    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
```

The `genres` variable in the `edx` dataset has multiple genres attached to a single movie, but in the table above, I've stratified them and have ranked them by the number of ratings received. I've also appended the average rating each genre posses to see if genres with a higher number of movies, tend to have a higher rating. But nothing conclusive can be drawn from the data above.

2.6 Title

The `title` variable has two major components, the first is the name of the movie and the second is the year of the movie's release. The former is largely irrelevant to our analysis, but the latter can be extracted and analyzed to determine if year of release plays a role in the rating of a movie.

Once we've done that, we can chart a graph to see if there is a relationship between year and average rating.



There does seem to be a considerable uptick in average rating ther further you go back in time, but I'd chalk that up to survivorship bias. Nobody wants to dig through history to watch mediocre movies.

2.7 Timestamp

I'll be frank, I'm not entirely sure how the timestamp variable could assist with constructing the recommendation system. Perhaps there is some bias I'm overlooking. Tat will have to be examined at a further time.

3 Creating the Model

Now that we have performed an analysis of all the relevant variables, it is time to move on to actually constructing the recommendation system model.

3.1 Baseline Model

Our first model, the baseline, takes the average ratings of the '`edx` set and uses it to estimate the average rating of the `final_holdout_set`. While this may seem immaterial, it help with setting a baseline.

```
##           model     RMSE
## 1 Baseline Model 1.061202
```

1.061202 is our RSME. A far cry from the targeted 0.86490, but I suppose we need to start somewhere

3.2 Movie Model

The movie model, our first evolution to the baseline, acknowledges that not all movies are the same. The average of the `edx` data set may be ~ 3.51, not every movie is average. Some are rated worse, while others are rated higher.

```
##           model      RMSE
## 1 Baseline Model 1.0612018
## 2    Movie Model 0.9439087
```

And so, we've made a significant improvement to our RSME, still a way's off from our target, but we're getting there.

3.3 Movie and User Model

The next evolution acknowledges that users are not homogeneous, they have different tastes and will therefore give different scores to the same movie.

```
##                  model      RMSE
## 1       Baseline Model 1.0612018
## 2          Movie Model 0.9439087
## 3 Movie and User Model 0.8653488
```

And we're almost there! I think we just need one more improvement before we meet our metric!

3.4 Regularized Model

Regularization is a method to reduce errors in the recommendation system that arise from outlier ratings that skew the Root Square Mean Error.

```
##                  model      RMSE
## 1       Baseline Model 1.0612018
## 2          Movie Model 0.9439087
## 3 Movie and User Model 0.8653488
## 4    Regularized Model 0.8648177
```

And there we go! The RMSE for the regularized model is 0.8648177, below our target of 0.86490!

4 Results

```
##                  model      RMSE
## 1       Baseline Model 1.0612018
## 2          Movie Model 0.9439087
## 3 Movie and User Model 0.8653488
## 4    Regularized Model 0.8648177
```

The lowest RMSE predicted value is 0.8648177 This value was obtained by applying regularization onto a Movie and User Model

5 Conclusion

A machine learning algorithm was used to constructed in order to predict the ratings from the Movie Lens dataset. The aim of the algorithm was to reach an RMSE of 0.86490 or below, This was achieved by a Regularized Movie and User model.

However this model can be further improved - and the RMSE brought down even further - by accounting for biases present in the genres and timestamp variables.

6 Appendix

https://github.com/AlessandroCorradini/Harvard-Data-Science-Professional/blob/master/09%20-%20PH125.9x%20-%20Capstone/MovieLens%20Recommender%20System%20Project/MovieLens%20Project%20Report.Rmd

https://github.com/bnwicks/Capstone/blob/master/MovieLens.Rmd

https://github.com/ujjawalmadan/Movielens-Capstone-Project/blob/master/Movielens%20Capstone%20Project.Rmd

https://www.rpubs.com/Airborne737/movielens

https://rpubs.com/christianakiramckinnon/MovieLens