Movie Lens Project

Renato Festa

2023-03-01

Overview

The Movie Lens Project is a part of the HarvardX Data Science Certificate Program. The goal is to build a recommendation system using the knowledge acquired from the course. We will use the MovieLens dataset provided by the course to run this project, which contains 10 million ratings from different users for different movies. The code to obtain the data has already been split into edx (train set) and final_holdout_test (test set).

Method

To calculate the accuracy of the prediction, we will compute the **RMSE** of the final code. RMSE stands for Root Mean Square Error, and it is one of the most commonly used methods for calculating differences between values. When computing the RMSE, we should aim to obtain a lower value. This means that at each step of the code, we will try to improve the results with the goal of achieving a lower RMSE.

```
    RMSE = √((1/N) *∑u,i(y^u,i−yu,i)2
```

Analyzing and understanding the MovieLens dataset

In order to make a more accurate prediction, all of the analysis will be performed only on the training set, which is the edx dataset. First, let's understand the size of this dataset:

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

It is a large dataset, and as a result, some methods of analysis may not be feasible due to the computational effort required to compute predictions.

Now, let's understand how many columns there are in the dataset and what class of information is contained in each column.

```
## 'data.frame':    9000055 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : int  122 185 292 316 329 355 356 362 364 370 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983421 838983392 838983392
838984474 838983653 838984885 838983707 838984596 ...
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)"
```

```
"Stargate (1994)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller"
"Action|Drama|Sci-Fi|Thriller" "Action|Adventure|Sci-Fi" ...
```

Here, we can see that the dataset is a data.frame with 6 columns, which contain information about a specific rating, such as who rated which movie, the genre of the movie, and the rating that the user gave.

For a better visualization, let's take a look at the head of the 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)
## 7      1     355      5 838984474       Flintstones, The (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
## 7       Children|Comedy|Fantasy
```

So, the edx dataset has more than 9 million rows. But, how many movies were rated, and how many users rated those movies?

```
##   Users movies
## 1 69878  10677
```

Movies specifics

Lets see which movie has the greatest rating score

```
##         userId movieId rating  timestamp                          title
## 6025330  42966   53355      5 1226808362 Sun Alley (Sonnenallee) (1999)
##                 genres
## 6025330 Comedy|Romance
```

This is the movie with the highest rating from this list. However, it had only 1 rating from users. How often do movies get to the top of the ranking because they have few ratings?

```
## # A tibble: 6 × 3
##   title                                    rates mean_rating
##   <chr>                                    <int>       <dbl>
## 1 Blue Light, The (Das Blaue Licht) (1932)     1           5
## 2 Fighting Elegy (Kenka erejii) (1966)         1           5
## 3 Hellhounds on My Trail (1999)                1           5
## 4 Satan's Tango (Sátántangó) (1994)            2           5
## 5 Shadows of Forgotten Ancestors (1964)        1           5
## 6 Sun Alley (Sonnenallee) (1999)               1           5
```

Here, we can observe that the movies with the highest ratings are the ones with less than 2 ratings. To get a more accurate understanding, we should remove the movies with few ratings. Let's only calculate the ratings for the movies with 100 or more ratings and see which one has the highest rating.

```
## # A tibble: 6 × 3
##   title                            rates mean_rating
##   <chr>                            <int>       <dbl>
## 1 Shawshank Redemption, The (1994) 28015        4.46
## 2 Godfather, The (1972)            17747        4.42
## 3 Usual Suspects, The (1995)       21648        4.37
## 4 Schindler's List (1993)          23193        4.36
## 5 Casablanca (1942)                11232        4.32
## 6 Rear Window (1954)                7935        4.32
```

Now it makes more sense. The best-rated movie is The Shawshank Redemption, which has 28,015 ratings. Looking at this list, we can also notice that the six highest-rated movies were all released before the year 2000. We will investigate this further later.

Despite having the highest mean rating, does The Shawshank Redemption have the most ratings overall?

```
## # A tibble: 6 × 3
##   title                            rates mean_rating
##   <chr>                            <int>       <dbl>
## 1 Pulp Fiction (1994)              31362        4.15
## 2 Forrest Gump (1994)              31079        4.01
## 3 Silence of the Lambs, The (1991) 30382        4.20
## 4 Jurassic Park (1993)             29360        3.66
## 5 Shawshank Redemption, The (1994) 28015        4.46
## 6 Braveheart (1995)                26212        4.08
```

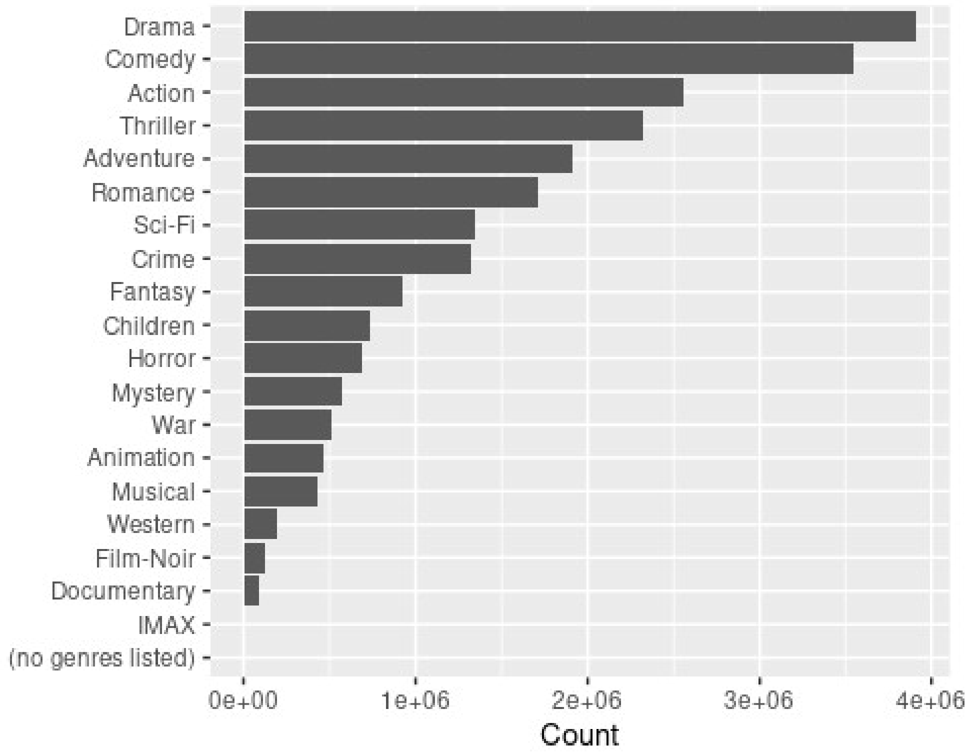
Pulp Fiction is the movie with the most ratings. However, it was not enough to put it in the top 6 movies. Now, let's take a look at the worst-rated movies on the list.

```
## # A tibble: 6 × 3
##   title                                                 rates mean_rating
##   <chr>                                                 <int>       <dbl>
## 1 From Justin to Kelly (2003)                             199       0.902
## 2 Pokémon Heroes (2003)                                   137       1.03
## 3 Glitter (2001)                                          339       1.18
## 4 Pokemon 4 Ever (a.k.a. Pokémon 4: The Movie) (2002)   202       1.18
## 5 Barney's Great Adventure (1998)                         208       1.19
## 6 Gigli (2003)                                            313       1.19
```

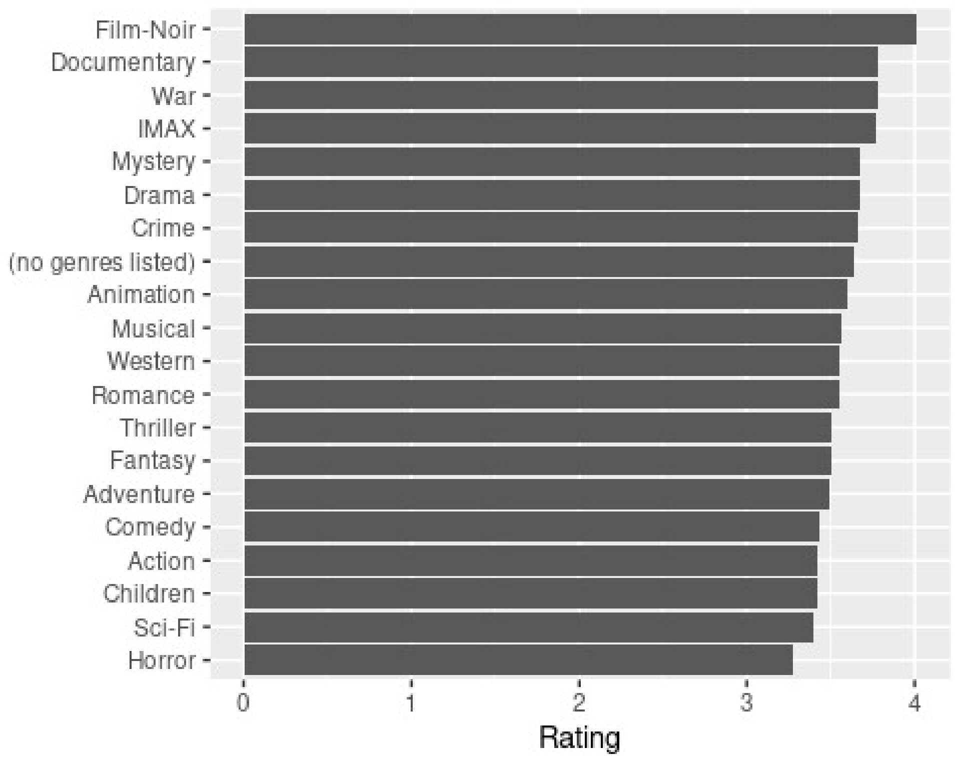
The worst-rated movies also have very few ratings. This highlights the importance of recommendations in guiding users towards better content.

Genres specifics

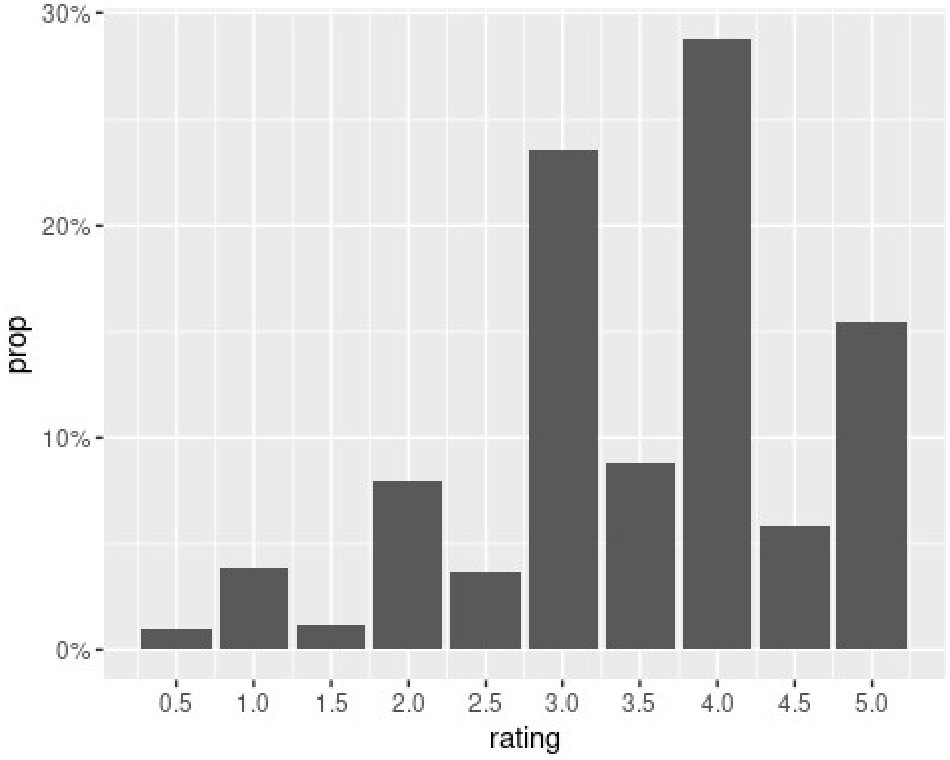
Another aspect of the movies is their genres. Certain genres may have more views and better acceptance than others.



Drama and Comedy are the most rated movies on our list. Now lets see if they are the ones with best ratings:



The highest-rated genres are Film-Noir and Documentary, but these movies also have a low number of ratings. Additionally, we can see that only one genre has an average rating higher than 4, and all genres have ratings higher than 3. To better understand the distribution of ratings throughout the dataset, let’s plot the frequency of each rating.



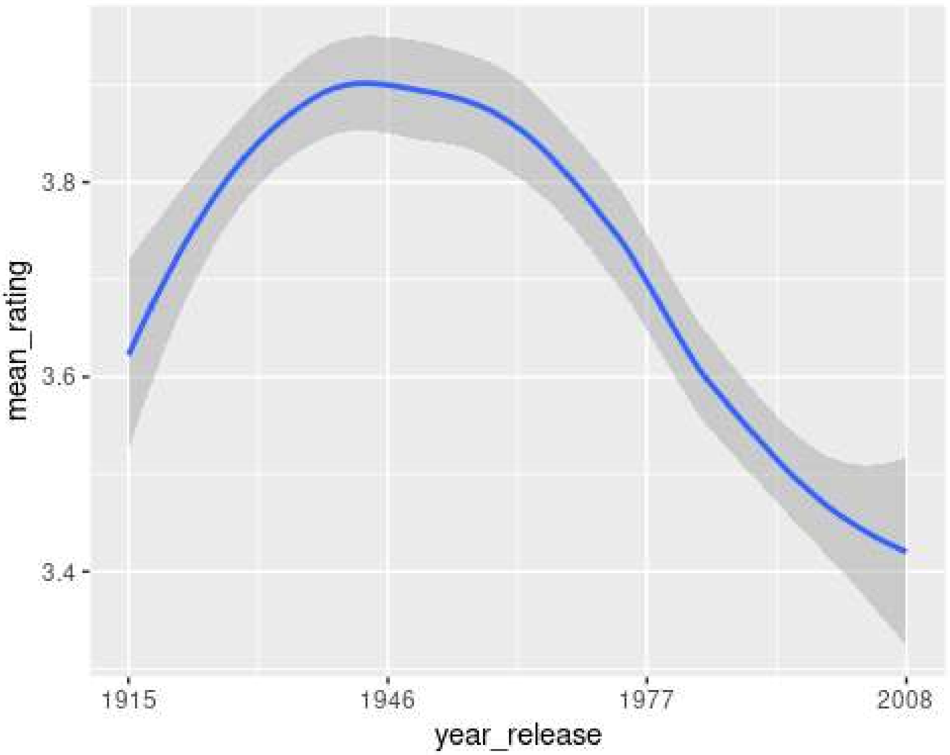
More than half of the ratings fall between 3.0 and 4.0. We can also observe that most of the ratings were whole numbers, such as 3.0, 4.0, and 5.0.

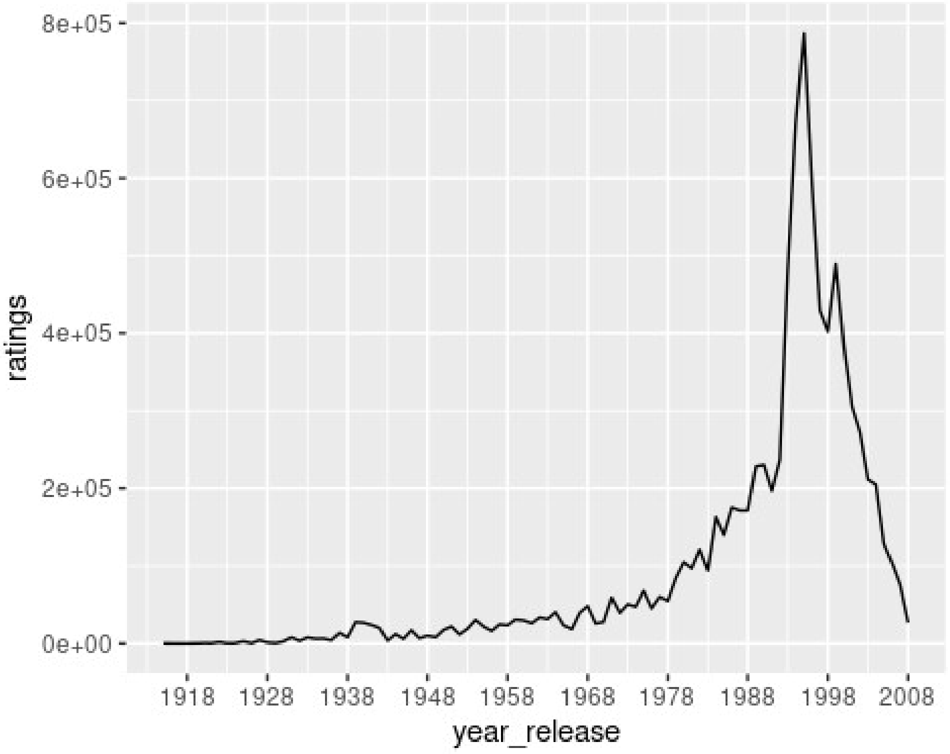
Year specifics

In the next step, we will analyze the effect of time on movies. To do that, we need to create another column containing the release year of each movie.

```
##                            title year_release
## 1              Boomerang (1992)         1992
## 2               Net, The (1995)         1995
## 4               Outbreak (1995)         1995
## 5               Stargate (1994)         1994
## 6 Star Trek: Generations (1994)         1994
## 7       Flintstones, The (1994)         1994
```

Now with the new column we can understand better the year effect:





The movies from the 90s and beyond have the highest number of ratings, but they also have the lowest average rating.

Model of recommendation

Now we will build a recommendation model and work on it until we achieve the lowest RMSE value.

Calculating the mean

Let’s calculate the RMSE using only the mean rating of the movies in the dataset. As we learned during the analysis, to make better predictions, let’s filter the dataset to only include movies with 100 or more ratings to achieve a better RMSE value.

```
## [1] "Mean rating:"     "3.52001869095358"

## [1] "RMSE:"            "1.06123177204691"
```

The average rating for movies with more than 100 ratings is 3.52. If we use only this value to recommend movies, the resulting RMSE will be 1.061232.

Using the user effect

As some users are more optimistic than others, we will use this information to personalize the recommendations for each of them. We will assign higher ratings to users who typically give higher ratings and lower ratings to the more pessimistic ones

```
## [1] "RMSE:"             "0.978335971005418"
```

With this, we can improve our results, which means we are getting better predictions of the ratings that users would give to certain movies.

Using the movie effect

We can observe from the plots that some movies have higher ratings than others. By implementing a ranking system to compute predicted ratings, such that movies with better ratings receive higher predicted ratings, we should be able to achieve a lower RMSE. Let’s utilize this information to calculate an even more accurate recommendation.

```
## [1] "RMSE:"             "0.881609571352724"
```

Results

Our final RMSE is 0.8822445. Despite trying different approaches, I couldn’t lower it further. One of the reasons is that the dataset is quite large and requires significant computational effort.

Approach	RMSE
Mean rating puls user and movie effect	0.8816096

Conclusion

Although this model is simple, it could still work as a recommendation system. It takes into account whether a user tends to rate movies more positively or negatively, as well as the overall rating of a movie among users. Therefore, if a user usually rates movies poorly, the recommendation system will predict a lower rating for them, and if a movie has a low rating among users, the system will predict an even lower rating. The opposite occurs for users who tend to be more optimistic in their ratings; they will receive recommendations with higher predicted ratings.

The major problem with this model is that it does not account for genre bias, which limits its predictive accuracy. If the dataset were smaller, I could use a K-nearest neighbor model to make better predictions. This would allow me to use all the collected data to calculate a predicted rating for the final holdout test.