Data Science Capstone Project: Data named: MovieLens

Navarro H Daniel D

2024-Mar-09

This is one capstone project of the Data Science course by HarvardX, capstone consists in two projects, this one is a predefined task consisting in generating a Machine Learning algorithm for Movie recommendation on a given dataset of movies and ratings.

The practical experience is considered a teaching tool to show all the skills acquired through the course. The intended objectives of HarvardX that the student learns:

> How to apply the knowledge base and skills learned throughout the series to real-world problems
> How to independently work on a data analysis project

The second project consists in a work chosen by the student for which raw data is not provided nor the objective is given.

The entire course in Data Science contains all the related abilities and skills from data collection, wrangling, visualization, description, analysis, statistical inference, linear regression analysis and finally the creation of Machine Learning algorithms for prediction purposes.

Prediction is an important ability for today's service and consume markets, that companies need to master in order to foresee trends and optimize their offerings.

Data used was provided as part of the course material, popular dataset MovieLens, which have become a commonly used data set for machine learning practice around platforms due to the success of the web-based recommendation system MovieLens that uses 3 different algorithms to recommend movies to the users based on previous ratings.

You can find more about it in Wikipedia: https://en.wikipedia.org/wiki/MovieLens

Goal

To model a movie recommendation algorithm with high level of accuracy.

A machine learning algorithm needs to be trained in order to predict movie ratings based on the user and the movie genre, which can be used afterwards for fine tune recommendation systems, also called click optimization.

The process is to be performed on the data provided by the course staff, which contains movies recommendations and users data for around 10 million registers.

Raw data for the project contains 10 million lines of ratings provided by 69878 users and 10677 rated movies listed and already classified by genre as well.

The accuracy of the model will be measured by achieving a goal, set for the evaluation of the course, of RMSE lower than 0.86490.

Raw data is prepared, splitted and classified into training and test sets with the code provided by the course.

This project resumes the kit of abilities for Data Science as tough by the Data Science curriculum of HarvardX. However data scraping and wrangling was not needed as the provided data is already tidy and ready for analysis.

Contents

1 Method

1.1 Data preparation

Data was split with the provided code into a training set of roughly 9 000 000 lines and 1 000 000 lines was reserved for the test of the model. Of the resulting training set we have also split in 80% for training purpose and 20% for testing purpose.

We double checked the data in search of invalid or empty values to ensure its quality and avoid mistakes that can easily pass inadverted in such a large data set, no invalid or empty values were found.

1.2 Using linear regression

After an initial exploration of the data and visualization we will perform a Linear Regression algorithm modeling in search of a low Root-Mean-Square error (RMSE).

A low RMSE means a low deviation between obtained prediction and real value in the data, ideally the lower the RMSE the best prediction power of the algorithm that allows to perform better movies recommendation based on previous ratings given by the users.

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

2 Initial exploration

After the data have been loaded -data and loading code was provided as part of the project- we perform some initial exploration to get to be familiar with the data.

The parameter to be predicted is the rating of the movies, that is a discrete variable given as stars rating in a range from 0 till 5 with half star steps, namely: 0; 0.5; 1.0; 1.5; 2.0; 2.5; 3.0; 3.5; 4.0; 4.5; 5.0 stars.

The factors to explain the rating are user rating, movie rating, genre rating and timestamp.

The mean rating found in the data set is 3.512464, median of 4 with a standard deviation of 1.0603927.

2.1 Checking the quality of the data

Although the data is a known data set and the quality was not in doubt, it is worth performing a quick check for invalid values to ensure the quality of the results and avoid silly mistakes that can happen with such a big data size.

```r
sum(!complete.cases(edx))
```

```
## [1] 0
```

```r
colSums(is.na(edx))
```

```
##    userId   movieId    rating timestamp     title    genres
##         0         0         0         0         0         0
```

```
sum(is.na(edx))
```

```
## [1] 0
```

We ensured there are not empty or invalid values present in any column of the data set.

2.2 Counting number of rows and columns of dataset "edx"

A dataset called "edx" was generated by combining the ratings and list of movies indexed by the movie ID.

Counting the dimension of the data set we find we have 9000061 rows with 6 columns, columns were titled: "userId", "movieId", "rating", "timestamp" for the ratings and "movieId", "title", "genres" for the movies, common index factor is the movieId.

This data can be obtained with the following code.

```
dim(edx)
```

```
## [1] 9000061       6
```

and the result is 9000061 rows and 6 columns as below:

```
## [1] 9000061       6
```

We will use the same structure when providing answer to the rest of the initial exploratory analysis.

2.3 Counting ratings

We find that, for example, there is no movie rated 0 and 2121638 movies rated 3.

```
sum(edx$rating == 0)
```

```
## [1] 0
```

```
sum(edx$rating == 3)
```

```
## [1] 2121638
```

2.4 Counting number of users and movies included in the data

By counting unique appearances we can see there are 10677 movies and 69878 users.

```
length(unique(edx$movieId))
```

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

```
length(unique(edx$userId))
```

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

2.5 Counting movies by genre

By genre we find to have 3 909 401 drama movies, 3 541 284 comedies, 2 325 349 thrillers and 1 712 232 romance movies.

```
## $Drama.n
## [1] 3909401
##
## $Comedy.n
## [1] 3541284
##
## $Thriller.n
## [1] 2325349
##
## $Romance.n
## [1] 1712232
```

2.6 Finding the most rated film

The most rated film in the dataset is Pulp Fiction with 31 336 ratings followed by Forrest Gump.

```
## # A tibble: 5 x 3
##   movieId numRating movieTitle
##     <int>     <int> <chr>
## 1     296     31336 Pulp Fiction (1994)
## 2     356     31076 Forrest Gump (1994)
## 3     593     30280 Silence of the Lambs, The (1991)
## 4     480     29291 Jurassic Park (1993)
## 5     318     27988 Shawshank Redemption, The (1994)
```

2.7 Finding the top most given ratings

The top ratings are 4 stars with a count of 2 588 021 and 3 stars with 2 121 638.

```
## # A tibble: 5 x 2
##   rating   count
##    <dbl>   <int>
## 1    4   2588021
## 2    3   2121638
## 3    5   1390541
## 4    3.5  792037
## 5    2    710998
```

2.8 Brief statistical analysis of the data

It was already check that the median rating found in the data set is 4, with a mean rating of 3.512464 and a standard deviation of 1.0603927.

We will see also later on, that the distribution of the data is approximately normal, this fact joined with the size of the data allows to perform a analysis of the probability of every rating for any given movie randomly selected.

Following the empirical rule Empirical Rule - Wikipedia we know that 68% of the cases lie at a distance of one standard deviation from the mean, so the ratings are taking a value of 3.51 +/- 1.06, as the given stars is a discrete value from 0 to 5 with 0.5 size steps, we can see that the most given stars are 2.5, 3.0, 3.5, 4.0 and 4.5.

Although this is not the goal of the project, and because of it we won't deep into the concept, we can build a confidence interval of 97.5% of the ratings given to movies.

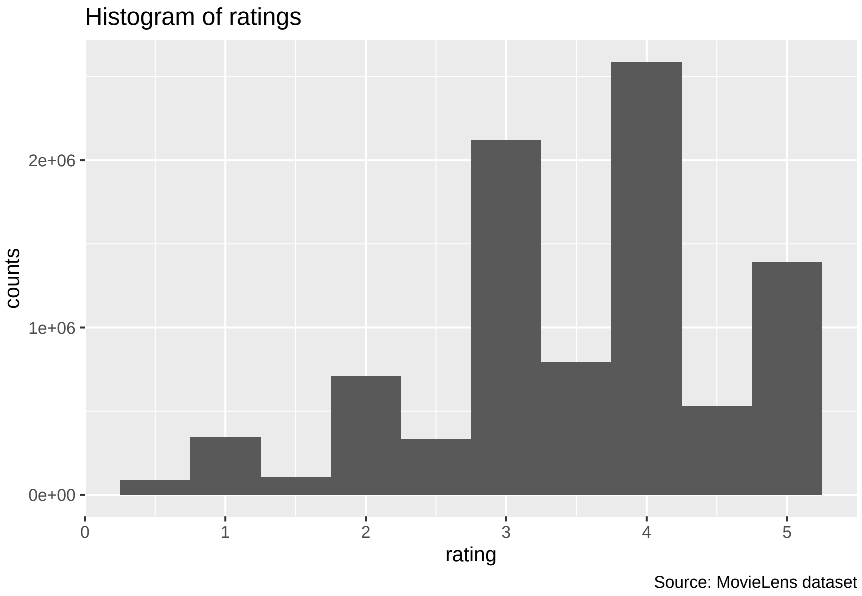
n = 9000061 xbar = 3.512464 sd = 1.0603927

With this data we have calculated a 90% confidence interval for the rating lies between 2 and 5 like this:

$$interval = xbar + / - Z * sd$$

3 Data visualization

Using a somewhat similar process as the one used during the machine learning course with the movielens data set of "dslabs" package, we can have a full sense of the distribution and content of the data.

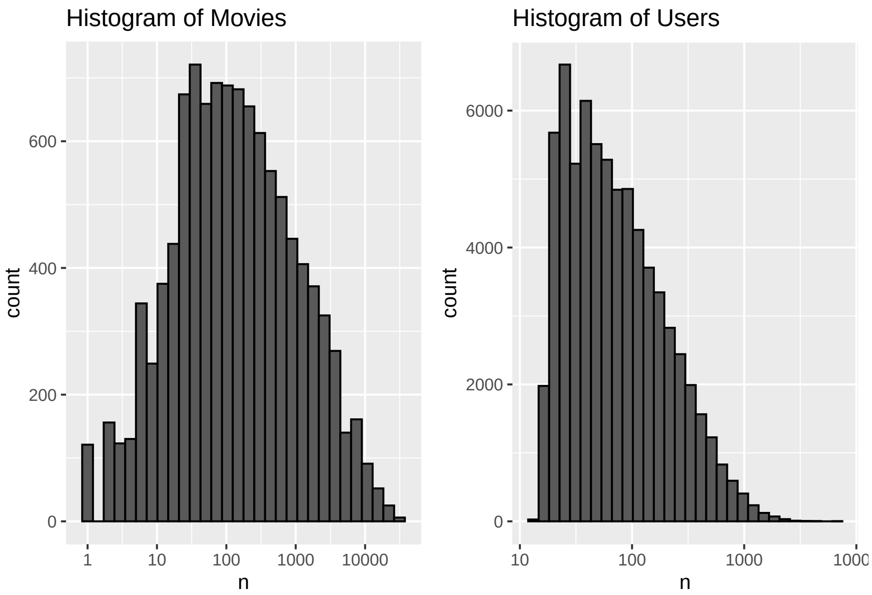


Source: MovieLens dataset

Figure 1: Ratings distribution

Data is left skewed with a median between three and a half and 4 stars rating, most of ratings are for full stars over half stars.

Also by plotting the number of ratings by movie and by user we see that some movies are more rated and some users tend to submit more ratings than others.



4 Modeling the algorithm

4.1 Building a first model

Simple Linear Regression

We start by building a first model with lineal regression assuming the ratings are due to random variation.

$\hat{y}\ u,i\ = \mu + \epsilon\ u,i$

```
## # A tibble: 1 x 2
##   method         RMSE
##   <chr>         <dbl>
## 1 Simple Model  1.06
```

So the RMSE is a simple average of all ratings contained in the data set.

We obtained a RMSE of 1.06 when considering only the average ratings, and we can still tune further with the movie effects still way larger than the given goal of 0.86490.

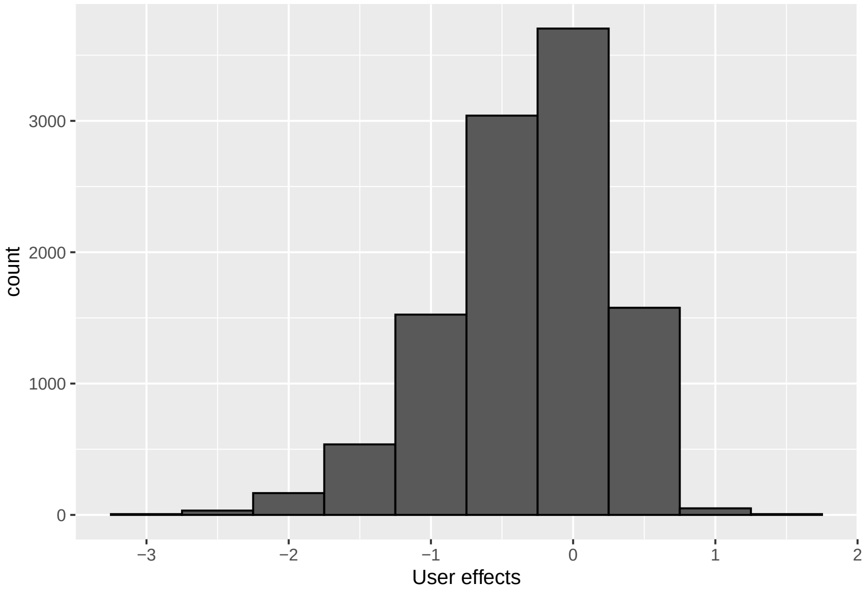
Modeling movies effect

We add now the effect of the movies in the model as some movies have different ratings and its influence can impact the predicted rating.

By adding the movie effect to the model our formula becomes:

$Y\ u,i\ = \mu + b\ i\ + \epsilon\ u,i\ \ \hat{b}\ i\ = mean(\hat{Y}\ u,i\ - \hat{\mu} - \hat{b}\ i\ )$

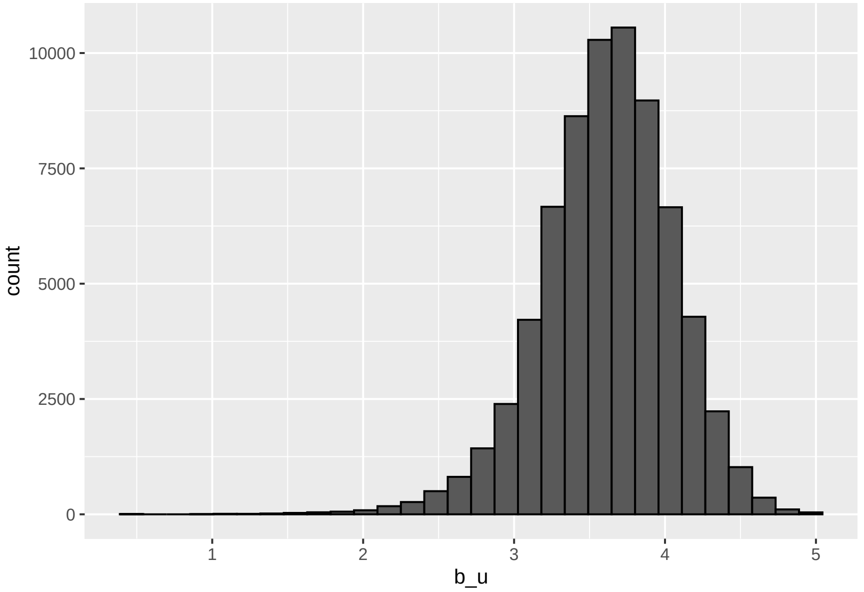
The following plots the movies effect:



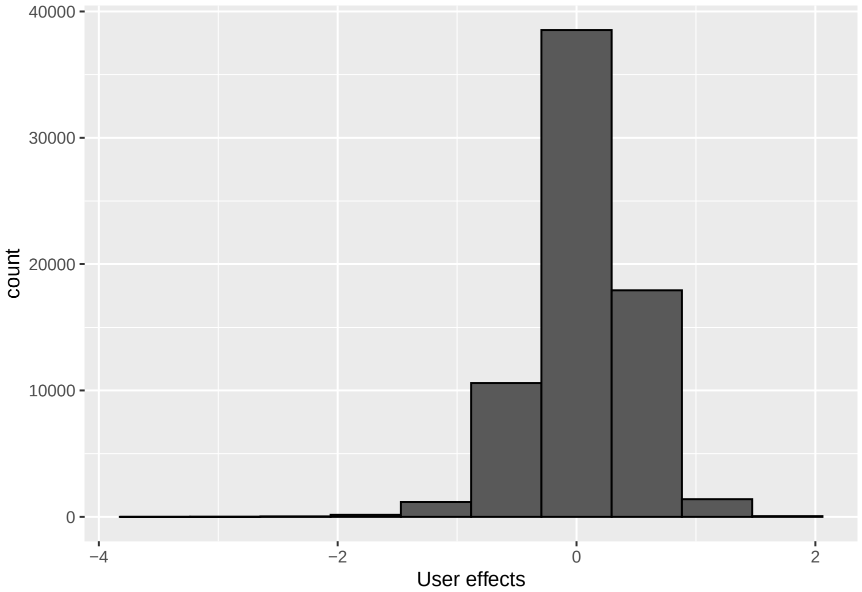
method	RMSE
Simple Model	1.0605787
Model with Movie Effect	0.9423688

After considering the movie effect, because movies also have some effect on how users tend to rate them on average, the RMSE of the model reduces to 0.94. Way to go but still need improvement for the set goal.

Modeling users effect



We can see the distribution of the users effect in the following plot.



Now the obtained RMSE is:

method	RMSE
Simple Model	1.0605787
Model with Movie Effect	0.9423688
Movies + users effect	0.8558259

$Y\ u,i\ = \mu + b\ i\ + b\ u\ + \epsilon\ u,i$

It becomes evident, the more factors considered in the model the smaller the error and the algorithm can better explain the movie ratings.

We have improved from approximately 1.0604 regression mean standard error to approximately 0.8558. With this the goal of the course of obtaining a RMSE lower than 0.86490.

With this the model has fulfilled the course goal, but: “Can we do better?”

Modeling genre effect

We tried the effect of the movies genre on the algorithm.

$Y\ u,i\ = \mu + b\ i\ + b\ u\ + b\ g\ + \epsilon\ u,i,g$

method	RMSE
Simple Model	1.0605787
Model with Movie Effect	0.9423688
Movies + users effect	0.8558259
Model + Movies + genre effect	0.8554850

However we see that evaluating the genre effect is not adding benefit to the model as the gain in terms of RMSE is marginal, for which further tests are omitted.

5 Results

Due to hardware limitations further tests were omitted, no kNN modelling could be performed with Caret and it was not considered necessary due to the fact that the initial objective RMSE was obtained.

By performing a linear regression model of the data it was possible to reach a RMSE of ‘0.855485 for predicting movie rating based on users previous ratings, including movies and genre effect.

The final tuned model came to be:

$\hat{y}\ u,i,g\ \ = \mu + b\ u\ \ + b\ i\ \ + b\ g\ \ + \epsilon\ u,i,g$

A simple linear regression analysis performed on the average rating was not accurate enough, but after completing the analysis with other factors like user’s previous ratings and the movies average ratings the accuracy of the model improved to a RMSE under the given goal of 0.86490.

Worth mentioning that only the movie effect did not result in enough accuracy, only after combining it with the users effect we had a convincing accuracy level.

Further refinement of the model by adding the genre effect provided such a small improvement in accuracy that might not be worth the extra effort.

This part of the code gave the author the more complication and essay and error tries, such a small improvement was at the beginning unadverted, looked like it was the very same RMSE.

```r
rmse_results |> knitr::kable()
```

method	RMSE
Simple Model	1.0605787
Model with Movie Effect	0.9423688
Movies + users effect	0.8558259
Model + Movies + genre effect	0.8554850

The author did not consider necessary to perform an analysis with the timestamp nor further analysis because of the hardware limitations and because the initial goal was achieved.

6 Conclusion

The whole process of data science was followed during the capstone project, although data wrangling was not needed, the final goal of the machine learning project was to achieve a specific goal Root Mean Square Error of 0.86405, that was met with a standard linear regression model after adding to the model the movie and user effect on the predicted ratings.

We move one step forward towards the genre effect with a marginal gain, thus considering not necessary further tests and checks.

The author tried kNN analysis with caret but hardware limitations made it impossible.

Summary

With the following model $\hat{y}\ u,i,g\ = \mu + b\ u\ + b\ i\ + b\ g\ + \epsilon\ u,i,g$ we obtained a RMSE of 0.8558259 of prediction of movies rating.

Limitations

Due to hardware limitations, Lenovo T450 i5 with 16Gb RAM, some analysis and plots were no possible, either the program reported that the data size could not be handled, plots did not appear or both Rstudio or the computer just restarted.

More tests, essay and error, insights are worth getting by testing and visualizing the effect of more advance algorithms in the future.

The experience leave more questions after a first contact with such a large data set in terms of other possibilities and what the model's performance can be with a different data set.

It must be noticed that the quality of the model is limited at the maximum by the quality of the raw data used during the training. Furthermore the author is conscious of the responsibility in Machine Learning work, trying to be as ethic as it is possible.

Future work

Future work on the same data set and with the same goal would be focus in using more advance tools from caret package to find new relations and test the limits of the predictions that can be obtained.