HarvardX Ph125.9x Data Science Capstone
MovieLens Project

Yaping Zhang

16.07.2020

Inhalt

1. Introduction

The project is to create a movie recommendation system using machine learning method based on the *MovieLens* 10M dataset. Through different machine learning approaches, a model will be trained and optimized to predict the rating of movies, based on which, the movies are recommended to users. The dataset is provided by GroupLens research lab containing numerous movies with ratings from various users.

In the report the dataset will be first described, furthermore, the goal and key steps will be determined. After that, we will introduce the method of the project, which are the detailed steps to reach the goal. Here we will first clean the dataset and then explore and visualise it to find out the insights for building up the model in next step. We look for the influence factors for ratings in the exploration. Based on the findings, machine learning algorithms with influencing factors(parameters) will be constructed. Based on the model we make the prediction of ratings on test dataset and then evaluate the result using the residual mean squared error (RMSE). Different models might be used until its RMSE reaches the goal. In the end of the report, a conclusion including summary, limitation and future work of the project will be explained.

1.1. Dataset description

The dataset is downloaded from the internet and wrangled into a data frame which then is separated randomly into a training set and a test set. We will explain it in detail:

1. The dataset is first pulled from the following link:

   http://files.grouplens.org/datasets/movielens/ml-10m.zip

2. Then it is cleaned and converted to a data frame containing 10000054 observations(ratings) of 6 variables(columns) as the following:

```
# A tibble: 10,000,054 x 6
   userId movieId rating timestamp title              genres
    <int>   <dbl>  <dbl>     <int> <chr>              <chr>
 1      1     122      5 838985046 Boomerang (1992)     Comedy|Romance
 2      1     185      5 838983525 Net, The (1995)      Action|Crime|Thriller
 3      1     231      5 838983392 Dumb & Dumber (1994) Comedy
 4      1     292      5 838983421 Outbreak (1995)      Action|Drama|Sci-Fi|Thrill~
 5      1     316      5 838983392 Stargate (1994)      Action|Adventure|Sci-Fi
 6      1     329      5 838983392 Star Trek: Generati~ Action|Adventure|Drama|Sci~
 7      1     355      5 838984474 Flintstones, The (1~ Children|Comedy|Fantasy
 8      1     356      5 838983653 Forrest Gump (1994)  Comedy|Drama|Romance|War
 9      1     362      5 838984885 Jungle Book, The (1~ Adventure|Children|Romance
10      1     364      5 838983707 Lion King, The (199~ Adventure|Animation|Childr~
# ... with 10,000,044 more rows
```

Here each row(observation) represents a rating given by the userId to the movieId. Important to notice, not every movie gets rating from every user. Therefore the number of ratings varies.

3. The dataset is then randomly separated into two sets called *edx* and *validation*. *Edx* consists of 90% of the *movieLens* dataset and is only used for training and tuning a model with tests. On the other side, *validation* dataset, possessing the remaining 10%, is only used to test the model with the final RMSE. Additional to that, the movies only existing in *validation* are filtered out and added into the *edx*, to make sure the users and movies included in the test set(*validation*) exist in the training set(*edx*). Here we see the dimension of the two datasets:

```
> dim(edx)
[1] 9000055       6
> dim(validation)
[1] 999999      6
```

1.2 Goal of the project and key steps

The goal of the project is to create an optimized machine learning model with **RMSE** value less than **0.8649**. The RMSE is defined as the following:

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

with: u, i: user, movie

$\hat{y}_{u,i}$, $y_{u,i}$: predicted rating, the true rating (from user u of movie i)

N: number of rating

The key steps of the procedure are the followings:

1. cleaning the entire data. From the chapter 1.1, we see some columns are not presenting the unique or clear information, eg. Title and coming out year of the movie are included in the same column. Also, the timestamp of the rating is still in the epoch time format.
2. Exploring the data. This is an essential step of the whole project. Through this step we find out the infulnce factors to support the next step.
3. Modelling. Based on the last step we will build up two different model only using *edx*. One will be more complex than the other.
4. Resulting. We will make the prediction with the models and evaluate it the RMSE value. In the end we will test the model with better RMSE value on the *validation* for final RMSE.

2. Methods

In this chapter the key steps mentioned above will be explained in detail. In the end of the chapter, the models will be built up and ready for the resulting procedure.

2.1 Data cleaning

Before starting to explore the data, data cleaning can be done to optimize the visualization and analysis.

First of all, we convert the “timestamp” to “date” using function **as.date()**. After that, extract coming out year information from the variable “title” and add it as a new column “year”. For this step I tried with several method. There is function such as str_split_fixed(), but it takes quite longer operating time. As a result I used functions **str_extract()** and **str_replace().** The advantage of them is you can be sure of extraction of the correct information and it takes much less time. The dataset looks like the following after the cleaning:

```
> as.tibble(edx)
# A tibble: 9,000,055 x 7
   userId movieId rating genres                              date       year  title
    <int>   <dbl>  <dbl> <chr>                               <date>     <chr> <chr>
 1      1     122      5 Comedy|Romance                      1996-08-02 1992  "Boomerang "

 2      1     185      5 Action|Crime|Thriller               1996-08-02 1995  "Net, The "
 3      1     292      5 Action|Drama|Sci-Fi|Thriller        1996-08-02 1995  "Outbreak "

 4      1     316      5 Action|Adventure|Sci-Fi             1996-08-02 1994  "Stargate "
 5      1     329      5 Action|Adventure|Drama|Sci-Fi       1996-08-02 1994  "Star Trek: Generations
"
 6      1     355      5 Children|Comedy|Fantasy             1996-08-02 1994  "Flintstones, The "

 7      1     356      5 Comedy|Drama|Romance|War            1996-08-02 1994  "Forrest Gump "

 8      1     362      5 Adventure|Children|Romance          1996-08-02 1994  "Jungle Book, The "

 9      1     364      5 Adventure|Animation|Children|Drama|Musical 1996-08-02 1994  "Lion King, The "
```

```
10      1    370      5 Action|Comedy                    1996-08-02 1994  "Naked Gun 33 1/3: The Fina
l Insult "
# ... with 9,000,045 more rows
```

The same is processed with dataset *validation*. Now the data is suitable to be explored and visualized.

2.2 Data exploration and visualization

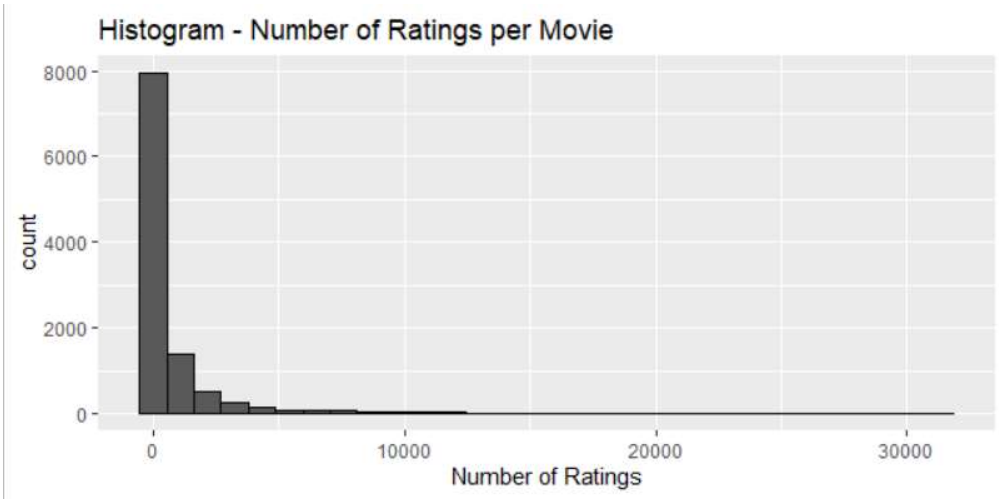
In this procedure, the main features of the dataset will be explored, in order to find out patterns for next step to build up the model. First, we will have a look at the distribution of rating. Furthermore, we inspect the movie, user and genres effects regarding number of ratings.

From the **summary of "rating",** we can find out the mean value and range of the ratings:

```
> summary(edx$rating)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.500   3.000   4.000   3.512   4.000   5.000
```

Here we see the mean rating **μ is 3.512**, while the rating range is between 0.5 and 5.

Next, we check how many ratings a movie normally gets to see if the movies are getting enough number of ratings to have a rating with certainty or less error.

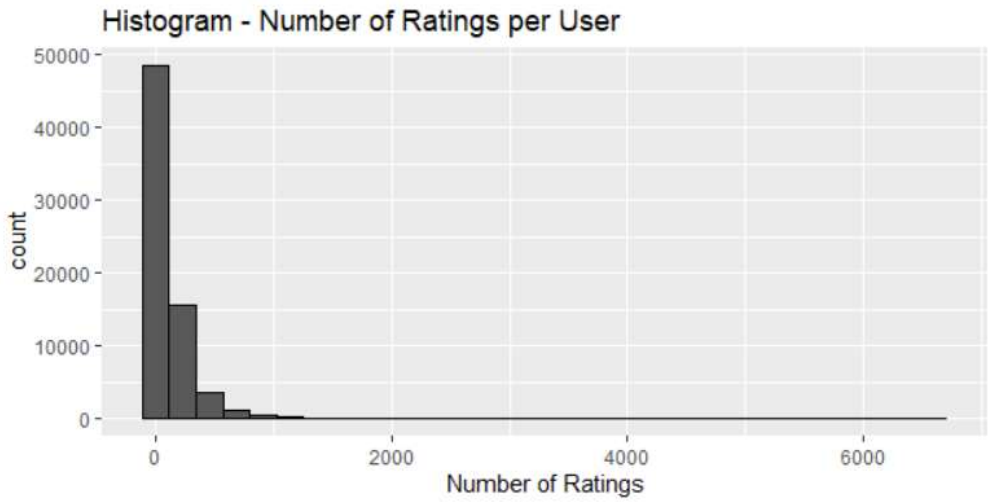


From the visualization we see, a large portion of movies has less than 100 ratings. Let's look into more details:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0    30.0   122.0   842.9   565.0 31362.0
```

The median of number of ratings is only 122 while the maximum is 31362. These show that the number of ratings is an influence factor of the rating parameter per movie.

In the same way let us look at the number of ratings per user and if there is the same pattern.



Most users give less than 100 ratings. It concludes that, the number of rating penalty term for users and movies are necessary to be introduced into the algorithms. In the same way we can find out the similar pattern on number of ratings on genres.

2.3 Insights and modelling approach

From the last chapter we see the model should include the effects from the mean rating, the movie and user effects with respectively the number of rating penalty.

2.3.1 General strategy

We will first generate a train (90%) and test (10%) subset from *edx*. The models will be trained and tuned on these two subsets to get our final model. With this final model, we make prediction on *validation* dataset and get the final RMSE value.

In general, we can see a rating is the sum of average rating, movie effects, user effects, genres effect and residual error. Following is the equation of predicated rating $Y_{u,I}$ with the first model:

$$Y_{u,i} = \mu + b_i + b_u + b_g + \varepsilon_{u,i},$$

With: bi: the movie effect,

b_u: the user effect,

b_g: the genres effect,

$\varepsilon_{u,i}$: the residual error.

With this model, we can check once if the RMSE reaches the targeted value. In case not, we will need to introduce the number of rating penalty λ for the two effects, as they would have bigger errors, if the number of ratings was too less. The equation will be changed to the following:

$$Y_{u,i} = \mu + \frac{1}{ni+\lambda}\sum_{n=0}^{ni} b_i + \frac{1}{nu+\lambda}\sum_{n=0}^{nu} b_u + \frac{1}{ng+\lambda}\sum_{n=0}^{ng} b_g + \varepsilon_{u,i},$$

With: λ: the penalty,

ni: the number of rating of each movie,

nu: the number of rating of each user.

ng: the number of rating of each genre.

For the second model, different λ values can be used to make the prediction with the test dataset and calculate the RMSEs. Here we make the RMSE equation to a function in R:

```
> RMSE <- function(true_ratings, pred){
+   sqrt(mean((true_ratings - pred)^2))
+ }
```

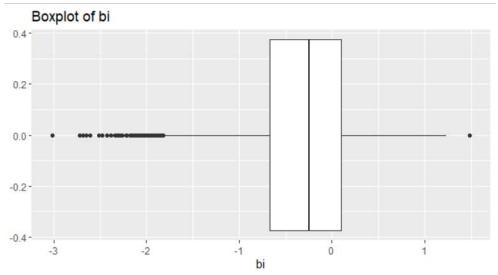
In next chapter, we explain in detail how to calculate the parameters.

2.3.2 Parameters in the model

We have the value of μ already calculated from the last chapter. Let us further calculate the b_i. It is generally the average rating of a movie and extracted μ out of it. Then we visualize it in a box plot to see the distribution (here we only plot the bi and bu):

$$bi = \frac{1}{ni}\sum_{n=0}^{n} (\text{yi} - \mu)$$

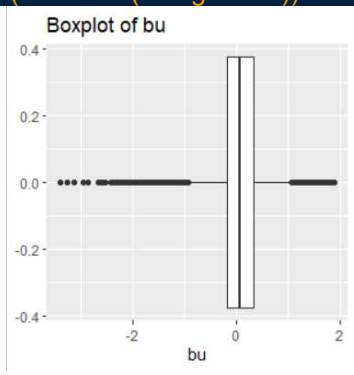
```
bi<-edxtrain%>%group_by(movieId)%>%summarise(bi=mean(rating-mu))
```



The next parameter to calculate is b_u. The formula and box plot are the following:

$$bu = \frac{1}{nu}\sum_{n=0}^{n} (\text{yi} - \mu - bi)$$

```
bu<-edxtrain%>%left_join(bi,by="movieId")%>%
+   group_by(userId)%>%summarise(bu=mean(rating-bi-mu))
```



The same way to calculate bg. For the more complex model with λ we give new names for the regularized two effects: bil, bul and bgl

$bil = \frac{1}{\text{ni}+\lambda}\sum_{n=0}^{ni} b_i,$

$bul = \frac{1}{\text{nu}+\lambda}\sum_{n=0}^{nu} b_u,$

$bgl = \frac{1}{\text{ng}+\lambda}\sum_{n=0}^{ng} b_g,$

In the following code, l represents λ.

```
+   bil<-edxtrain%>%
+     group_by(movieId)%>%summarise(bil=sum(rating-mu)/(n()+l))
+
+   bul<-edxtrain%>%left_join(bil,by="movieId")%>%
+     group_by(userId)%>%summarise(bul=sum(rating-bil-mu)/(n()+l))
+
+   bg<-edxtrain%>%left_join(bi,by="movieId")%>%left_join(bu,by="userId
")%>%group_by(genres)%>%
```

```
+      summarise(bgl=sum(rating-bi-bu-mu)/(n()+l))
```

λ can be any values, however there will be a best value, which has the lowest RMSE, that we finally use. We use the RMSE from the best λ as the result of the complex model. Now we start to make the prediction with the two models.

3. Results

This chapter including two steps. Based on the two model above. We will calculate the RMSE values with tuning. After we conclude the final model, we use the validation dataset to predict to get the final RMSE.

3.1 Comparing RMSE

Let us make the prediction and check the RMSE with the first model, to see if the goal of the project is already reached. After joining the columns bi and bu to *edx test subset*, the predicted rating is calculated by the following formula. Then we apply the RMSE function to evaluate the prediction:

$$\hat{y}u,i = \mu + bi + bu + bg$$

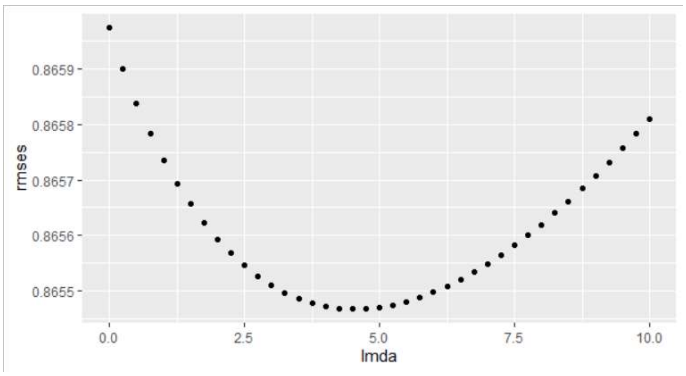
```
> pred1<-edxtest%>%left_join(bi,by="movieId")%>%left_join(bg,by="genres
")%>%
+    left_join(bu,by="userId")%>%mutate(pred=mu+bi+bu+bg)%>%pull(pred)

> rmse1<-RMSE(pred1,edxtest$rating)
> rmse1
[1] 0.8656019
```

This result is better than just predicting using the mean value but not yet reaches the goal. Let us use the complex model with penalty λ. Here we use the sapply to calculate RMSEs using different λs and find out the lowest RMSE and respectively the λ:

$$\hat{y}u,i = \mu + bil + bul + bgl$$

```
> rmses<-sapply(lmda,function(l){
+    mu<-mean(edxtrain$rating)
+
+    bil<-edxtrain%>%
+      group_by(movieId)%>%summarise(bil=sum(rating-mu)/(n()+l))
+
+    bul<-edxtrain%>%left_join(bil,by="movieId")%>%
+      group_by(userId)%>%summarise(bul=sum(rating-bil-mu)/(n()+l))
+
+    bgl<-edxtrain%>%left_join(bi,by="movieId")%>%left_join(bu,by="userId
")%>%group_by(genres)%>%
+      summarise(bgl=sum(rating-bi-bu-mu)/(n()+l))
+
+    pred2<-edxtest%>%
+      left_join(bil,by="movieId")%>%left_join(bul,by="userId")%>%left_join
(bgl,by="genres")%>%
+      mutate(pred=mu+bil+bul)%>%pull(pred)
+    return(RMSE(pred2,edxtest$rating))
+ })
```

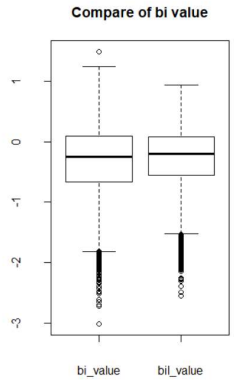
From the following plot we find out the best λ= 4.5 with the lowest RMSE:



```
> min(rmses)
[1] 0.8654673
```

This RMSE value is slightly bigger than 0.86490 but better than without regularization. Furthermore, when we use the entire edx dataset to train and test on validation, it is possible to get a better result. Therefore, we could already stop here and construct out final model.

Before we come to the conclusion of the final model. Let us have a look at the parameters before and after introducing the best λ, to see why we get a better result with the regularization:



We can see the bil value has smaller variety but has the similar median value compare to bi.

3.2 Final Result

Now our final model can be written in the following way:

$$Y_{u,i} = \mu + \frac{1}{ni+4.5}\sum_{n=0}^{ni} b_i + \frac{1}{ng+4.5}\sum_{n=0}^{ng} b_g + \frac{1}{nu+4.5}\sum_{n=0}^{nu} b_u$$

Now the very last step is to calculate RMSE on the *validation*:

```
> mu<-mean(edx$rating)

> bil<-edx%>%
+   group_by(movieId)%>%summarise(bil=sum(rating-mu)/(n()+4.5))
> bul<-edx%>%left_join(bil,by="movieId")%>%
+   group_by(userId)%>%summarise(bul=sum(rating-bil-mu)/(n()+4.5))
> bgl<-edx%>%
+   left_join(bil,by="movieId")%>%left_join(bul,by="userId")%>%group_by(gen
res)%>%
+   summarise(bgl=sum(rating-bil-bul-mu)/(n()+4.5))
> pred3<-validation%>%
+   left_join(bil,by="movieId")%>%left_join(bul,by="userId")%>%left_join(bg
l,by="genres")%>%
+   mutate(pred=mu+bil+bul+bgl)%>%pull(pred)
> RMSE(pred3,validation$rating)
[1] 0.8644543
```

With the final model we get the RMSE value smaller than the 0.86490. Therefore, we reached the goal of our project.

4. Conclusion

In this chapter we will summarise the project and inspect the limitation from different aspects and explain how to improve the future work.

In this project, we got the required dataset and were given the task of building up a prediction model with a goal RMSE value. The first and important thing to do is always cleaning the data and explore the data to get the insights. After that we built up different equations of models based on the patterns we found out from last step. In the end we tested the different models and chose the best out it. We tested the RMSE values from the final model. The targeted goal value is reached. There can be some improvement in the future work.

The limitation during project is, two processes take longer operation time and even sometimes crash the computer due to limitation of the RAM: data cleaning with str_ function and the RMSE calculation using different λ with sapply function. Additional, in the preparation of the project, the cross validation on the edx to train a model was also tested, but the Rstudio was crashed, therefore, the sub train and test set was used to construct the model.

Furthermore, the algorithms can be further improved to reach a lower RMSE. In this model, the movie general rating effect, user effect and genres effect with their number of rating penalty are introduced into the algorithms. With those parameters, the RMSE value does reach the target of the project. However, the year effect can also be considered into the algorithms in case of reaching lower RMSE value. Addition to that, the cross validation method could be researched and compare to the data participation method which is used in this project.