MovieLensProject

Abdallah Hazem

5/10/2020

#1- Introduction

You will find here a document that related to the MovieLens Project of the HervardX: PH125.9x Data Science: Capstone course. The present report start with a general idea of the project and by representing its objectif. The purpose for this project is creating a recommender system using MovieLens dataset.Recommender systems are a class of statistical learning systems that analyse individual past choices and/or preferences to propose relevant information to make future choices.This document contains an exploratory analysis section in which some characteristics of the data set are shown. This section will also explain the process, techniques and methods that were used to handle the data and to create the predicive model.

#1-1 Aim of the project

The aim of this Project is to create a recommendations system by machine learning algorithm that predicts user ratings (from 0.5 to 5 stars) using the inputs of a provided subset (edx dataset provided by the staff) to predict movie ratings in a provided validation set.

#2- Explore Dataset #2-1 Data componant The MovieLens dataset is automatically downloaded The data provided is a list of ratings made by anonymised users of a number of movies. The entire training dataset is a table of 9000055 rows and 6 variables. Note that the dataset is extermely sparse:if each user had rated each movie, the dataset should contain 54000330 ratings, i.e. 85 times more.

The 6 varibales are :

userId: Unique identification number given to each user. numeric variable • movieId: Unique identification number given to each movie. numeric variable. • timestamp: Code that contains date and time in what the rating was given by the user to the specific movie. integer variable. • title: Title of the movie. character variable. • genres: Motion-picture category associated to the film. character variable. • rating: Rating given by the user to the movie. From 0 to 5 stars in steps of 0.5. numeric variable.

userId	movieId	rating	timestamp	title	genres
1	122	5	1996-08-02 14:24:06	Boomerang (1992)	Comedy|Romance
1	185	5	1996-08-02 13:58:45	Net, The (1995)	Action|Crime|Thriller
1	231	5	1996-08-02 13:56:32	Dumb & Dumber (1994)	Comedy
1	292	5	1996-08-02 13:57:01	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
1	316	5	1996-08-02 13:56:32	Stargate (1994)	Action|Adventure|Sci-Fi
1	329	5	1996-08-02 13:56:32	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi

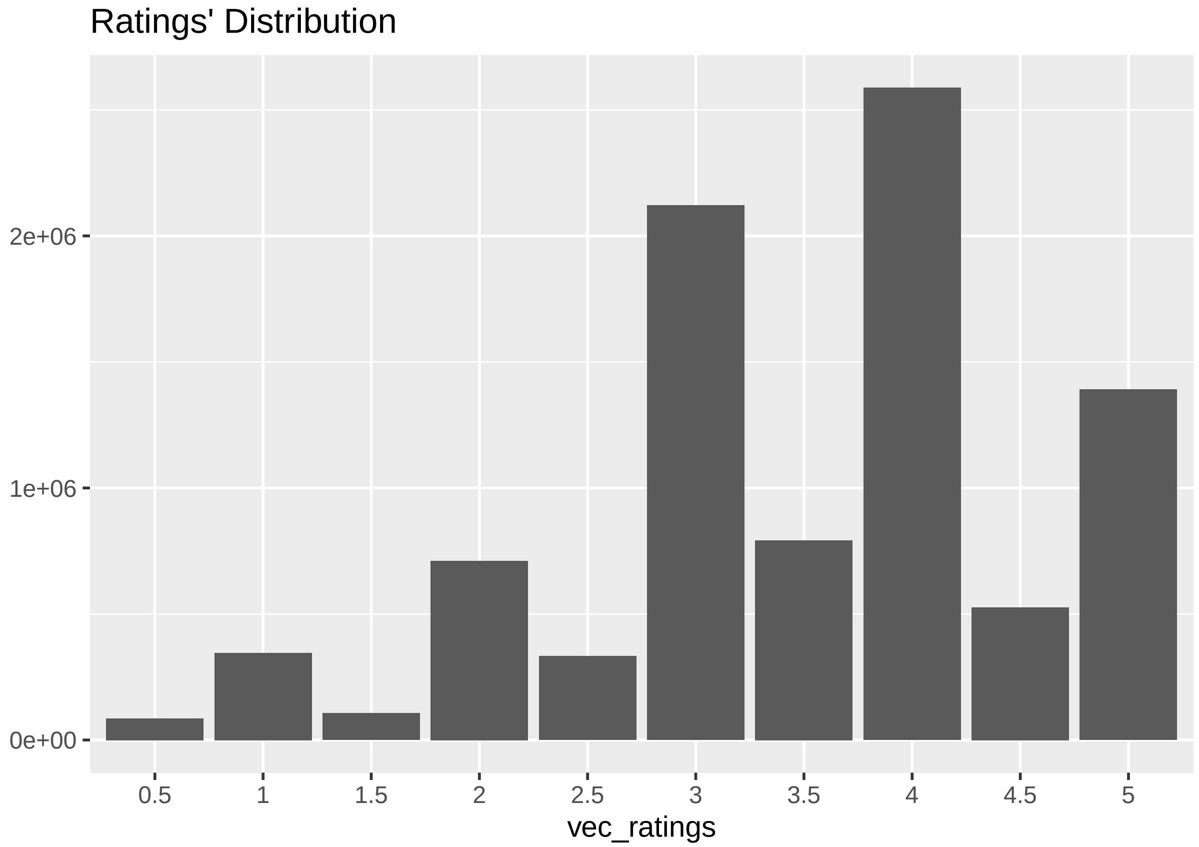
#2-2 Dataset Pre-Processing and Feature Engineering

#2-2-3 The pre-processing phase is composed by this steps:

1. Convert `timestamp` to a human readable date format;
2. Extract the month and the year from the date;
3. Extract the release year for each movie from the title;

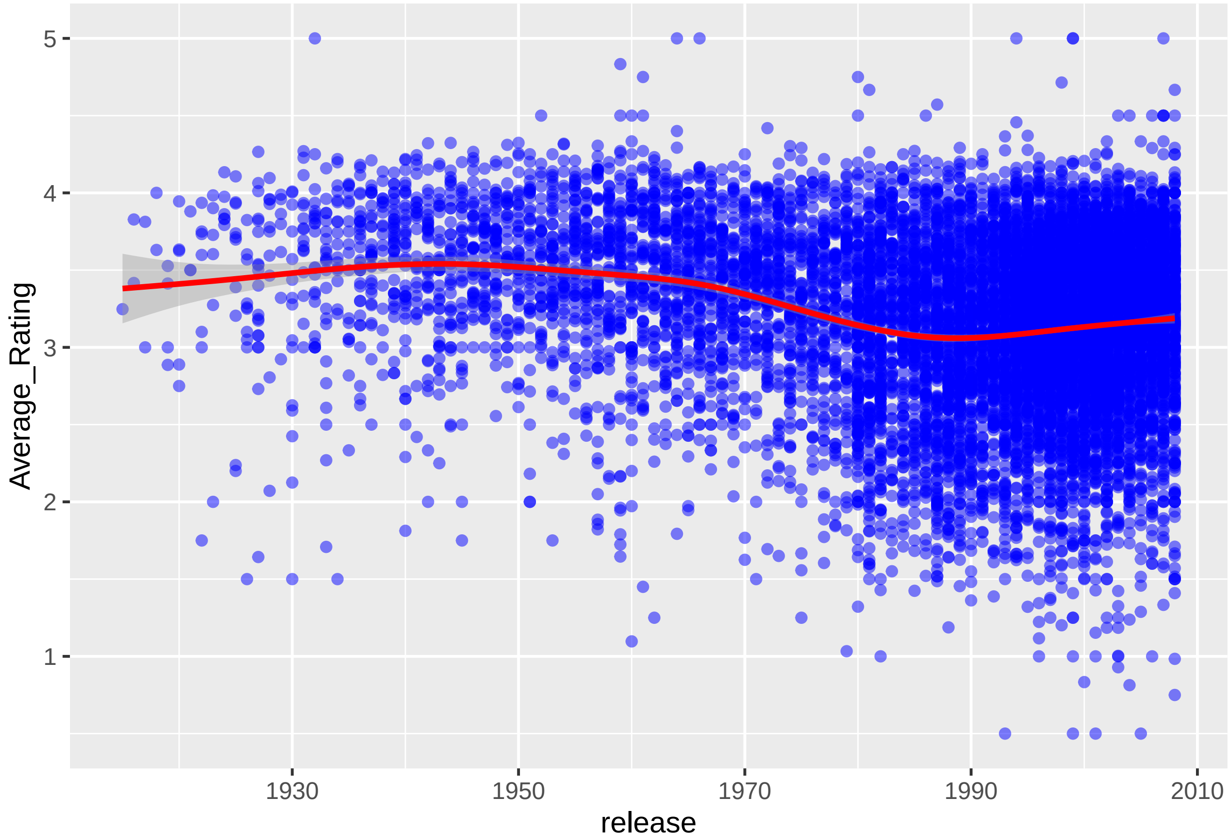
userId	movieId	rating	timestamp	title	release	genres
1	122	5	1996-08-02 14:24:06	Boomerang	1992	Comedy|Romance
1	185	5	1996-08-02 13:58:45	Net, The	1995	Action|Crime|Thriller
1	231	5	1996-08-02 13:56:32	Dumb & Dumber	1994	Comedy
1	292	5	1996-08-02 13:57:01	Outbreak	1995	Action|Drama|Sci-Fi|Thriller
1	316	5	1996-08-02 13:56:32	Stargate	1994	Action|Adventure|Sci-Fi
1	329	5	1996-08-02 13:56:32	Star Trek: Generations	1994	Action|Adventure|Drama|Sci-Fi

#**2-3 Ratings distribution**



The above rating distribution shows that the users have a general tendency to rate movies between 3 and 4. This is a very general conclusion. We should further explore the effect of different features to make a good predictive model.

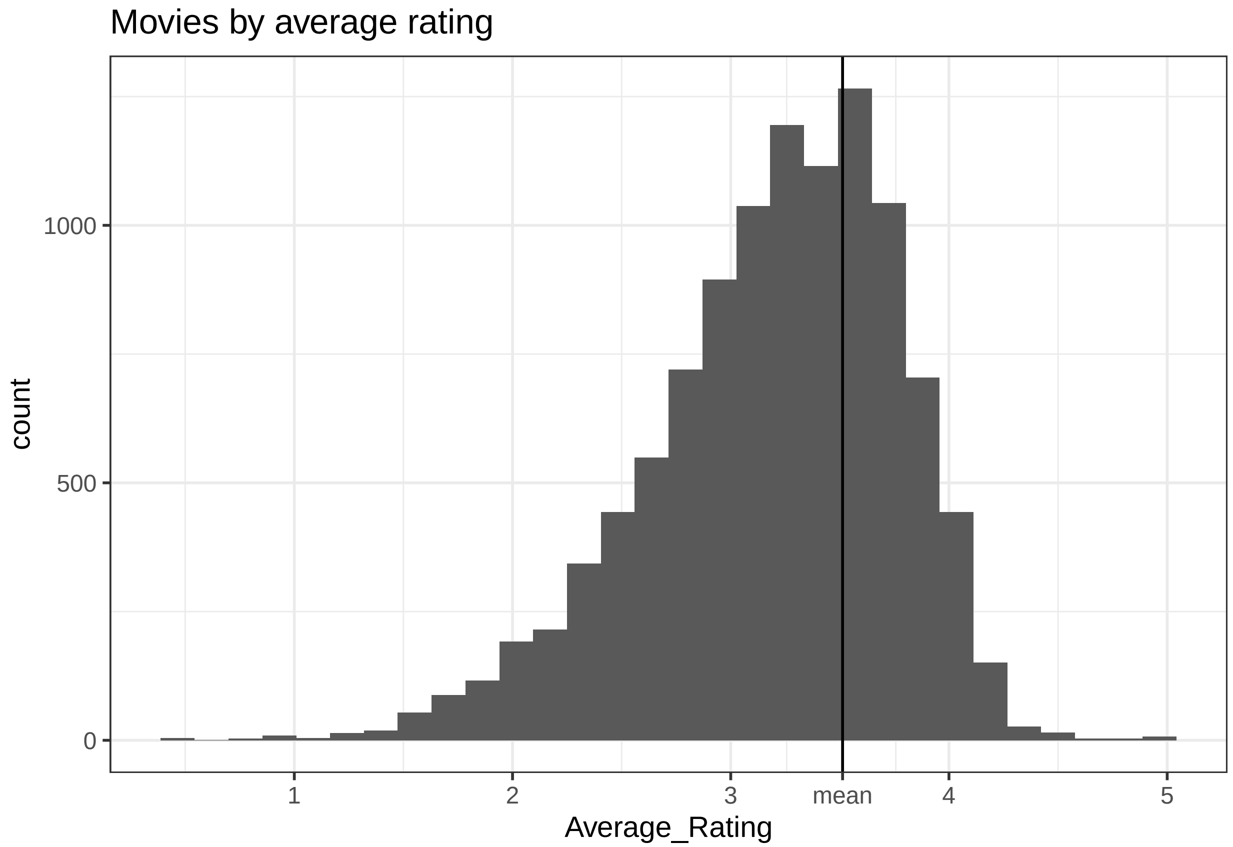
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



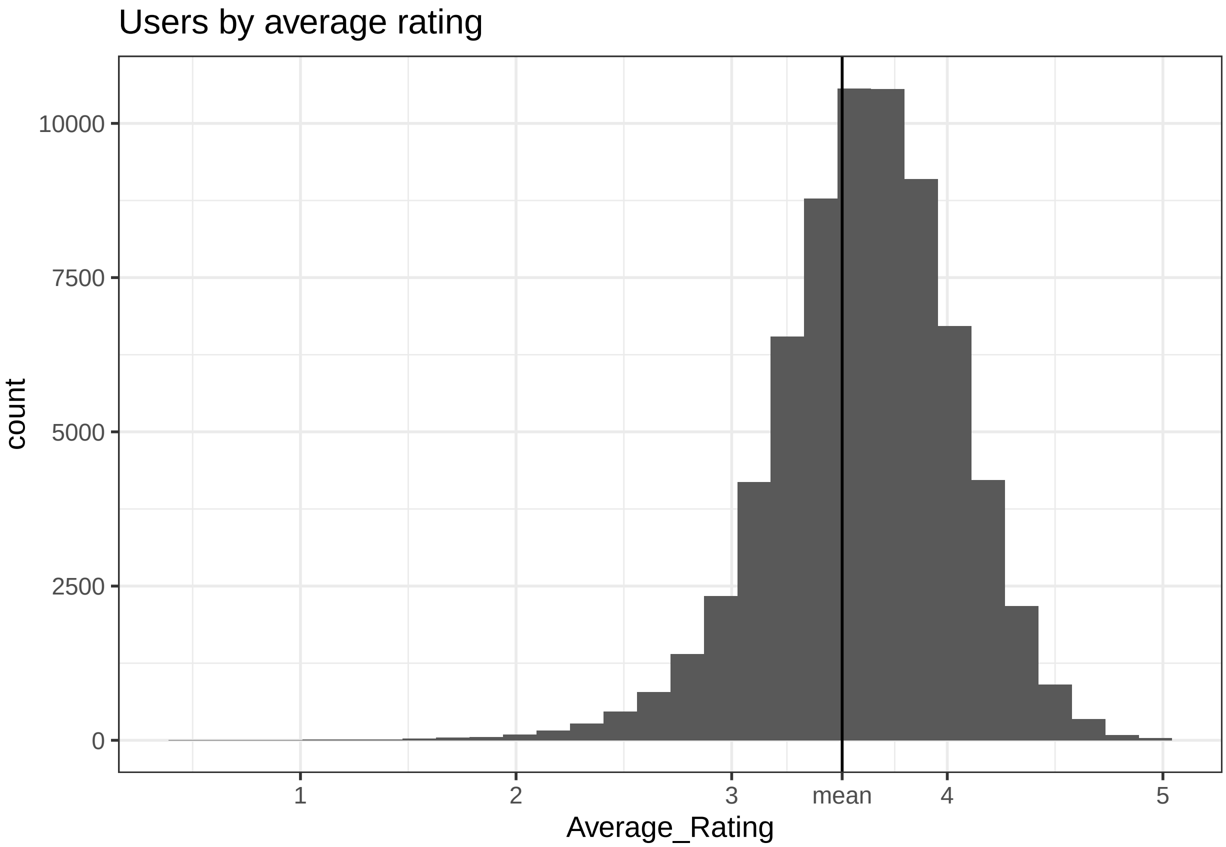
• There is clearly an effect where the average rating goes down. More striking is that recent movies are more likely to receive a bad rating, where the variance of ratings for movies before the early seventies is MUch lower.

• Very early years: very few ratings (very pale colour) possibly since fewer people decide to watch older movies. • Early years: Strong effect where many ratings are made when the movie is first screen, then very quiet period. • Recent years: More or less constant colour.

###2-3-1 Movies distribution by Average Rating



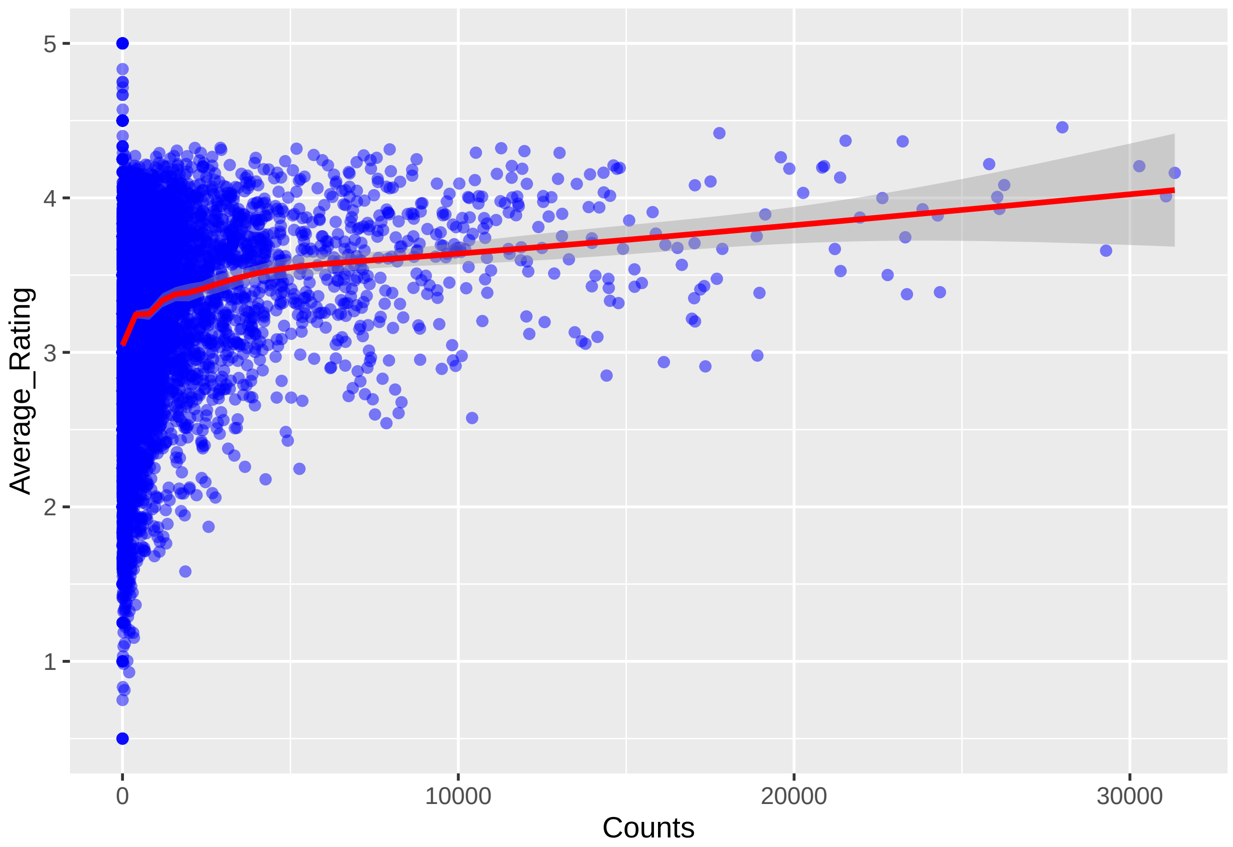
###2-3-2 Users distribution by Average Rating



Movies and users distribution by Average Rating look “Normal”. We can see that there is movie and user effects/bias, as “good” movies tempt to be rated higher than others, as well as some users are more easygoing than others.

##2-4 Ratings distribution by count

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



If a movie is very good, many people will watch it and rate it. In other words, we should see some correlation between ratings and numbers of ratings. Again, some sort of rescaling of time, logarithmic or other, need considering. The effect of good movies attracting many spectators is noticeable. It is also very clear that movies with few spectators generate extremely variable results.

#3- The Model

#3-1 Crating Train and Test set

First we wust create the train set and test set.

```
edx <- edx %>% select(userId, movieId, rating)

test_index <- createDataPartition(edx$rating, times = 1, p = .2, list = F)
    # Create the index

  train <- edx[-test_index, ] # Create Train set
  test <- edx[test_index, ] # Create Test set
  test <- test %>% # The same movieId and usersId appears in both set. (Not the same cases)
    semi_join(train, by = "movieId") %>%
    semi_join(train, by = "userId")
```

#3-2 Creating Baseline

We will generate basic model which consider the most common rating from the train set to be predicted into the test set. This is the baseline model.

Now, we have the RMSE to be *beaten* by our model.

Method	RMSE
Baseline	1.060006

We can observe that the RMSE of the most basic model is 1.060006. It's bigger than 1! In this context, this is a very bad model.

#3-3 User and Movie effect Model

We are trying to get a better RMSE by considering the user effect and the movie effect as predictors.

$$\hat{y}_i = UI + MI + \varepsilon$$

Method	RMSE
Baseline	1.0600060
User & Movie Effect	0.8423125

We've got obtained a better RMSE so we can make predictions on Validation data.

#3-4 User and Movie effect Model on validation data

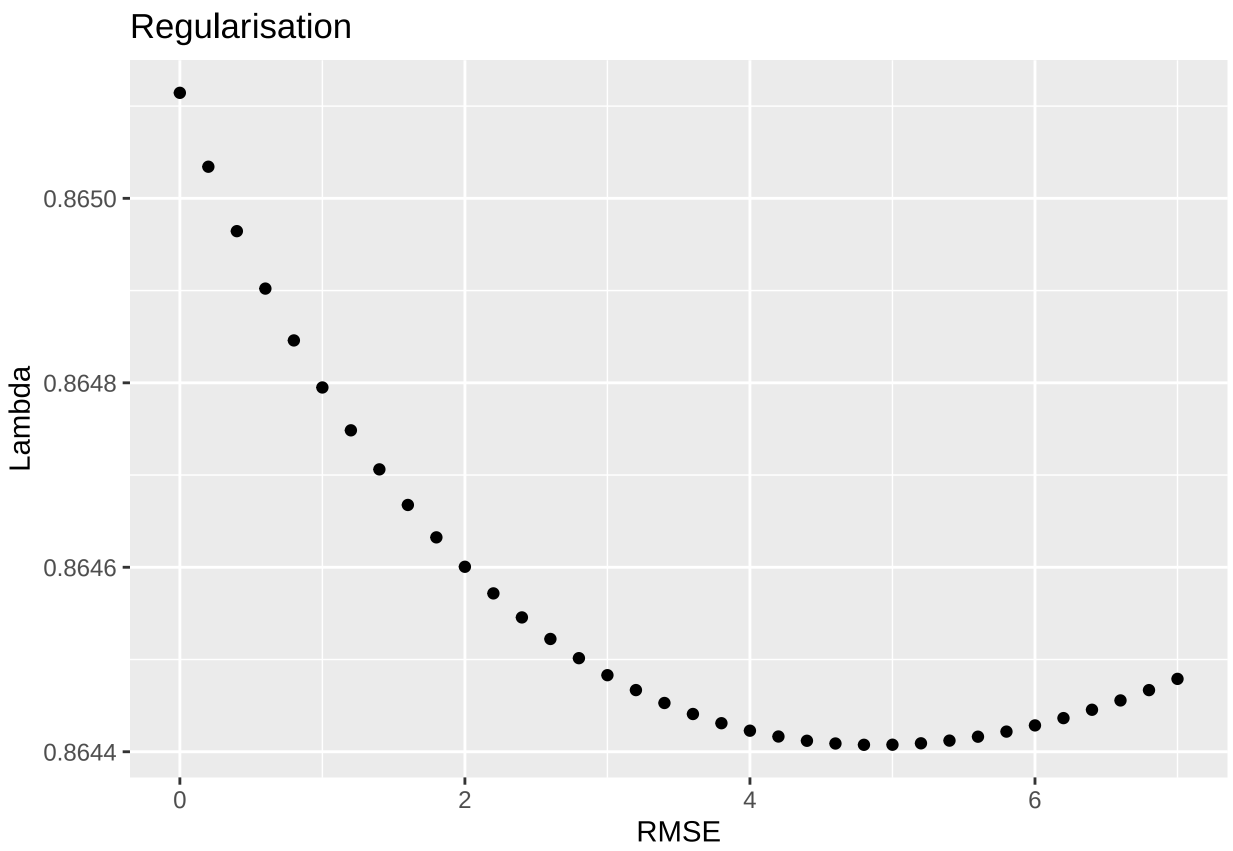
Validation data set needs to be handled the same as the train data set was handled after that we will be apple to make predictions.

Method	RMSE
User & Movie Effect on validation	0.8821831

We can see above that this RMSE is higher than the RMSE on the test set. This is highly probable, given that this is unseeing data. The good thing is that the difference is just 0.0398706. Now, let's see if *regularisation* give us better results.

#3-5 Getting Regularisation

Regularisation will evaluate the various values for lambda to give us the corresponding RMSE.



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

Method	RMSE
Baseline	1.0600060
User & Movie Effect	0.8423125
User & Movie Effect Regularisation	0.8644075

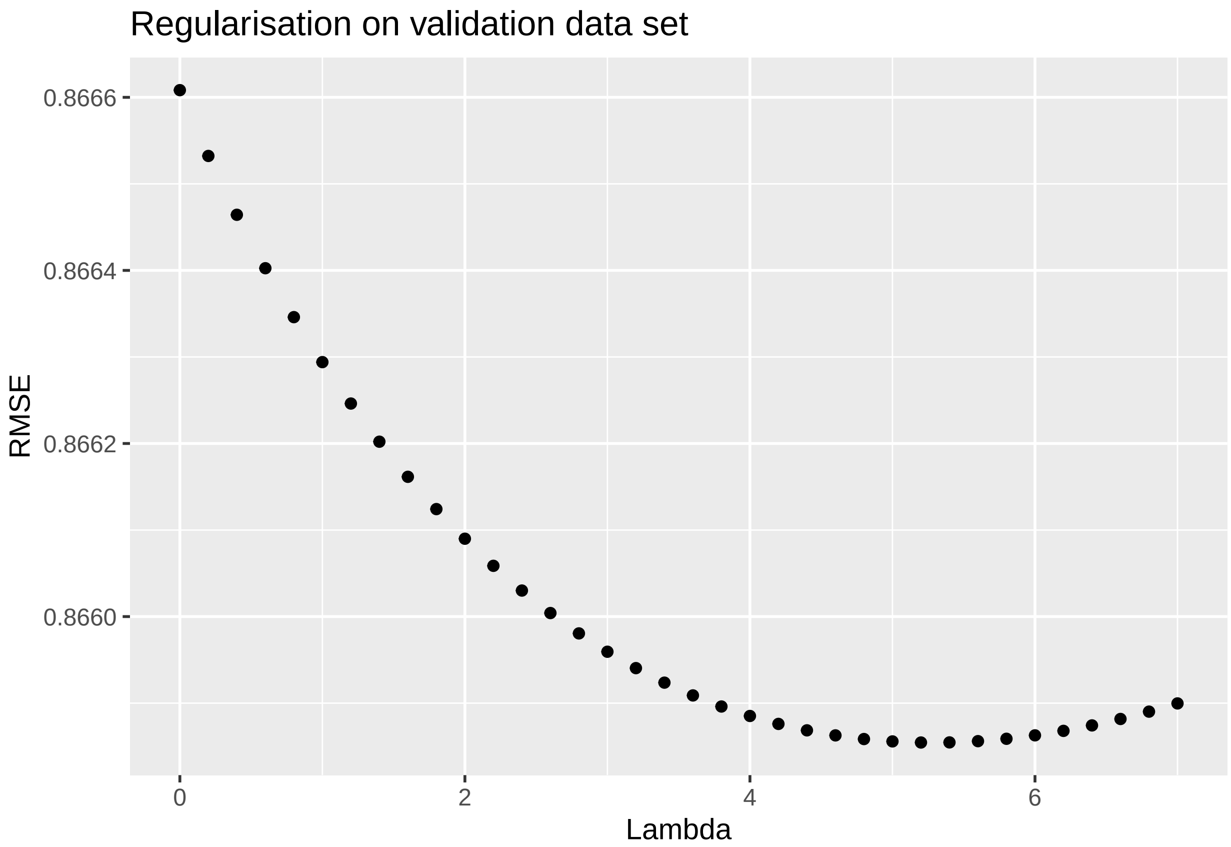
The regularisation give as a higher RMSE than the first "User & Movie Effect" model. This is unexpected

#**3-6 Getting Regularisation on validation data set**

let's see what is the result and performance of the regularisation on the validation data .

Table 1: RMSEs Summary

Method	RMSE
Baseline	1.0600060
User & Movie Effect	0.8423125
User & Movie Effect Regularisation	0.8644075
User & Movie Effect on validation	0.8821831
User & Movie Effect Reg. on validation	0.8658545



```
## [1] 5.2
```

```
## [1] 0.8658545
```

RMSE IS

Method	RMSE
User & Movie Effect on validation	0.8821831
User & Movie Effect Reg. on validation	0.8658545

#4- Results

We can observe that the better RMSE is obtained from the User & Movie Effect model. However, this RMSE only obtained on the test set. When we move to the validation data set, we obtain the worse RMSE (ignoring the baseline).

Considering that we MUst trust more in the performance of the model when we predict from unseeing data, we can say that the RMSE that results from the *User & Movie Effect with Regularisation on validation* (the last line in the table above) is our definitive model. This RMSE is obtained when $\lambda = 5$ which permit us to achieve **RMSE equal to 0.8658545.**

#**5- Conclusion** The variables userId and movieId have sufficient predictive power to permit us to predict how a user will rate a movie. This tell us that we could make better recommendations about movie to specific users of the streaming service. Therefore, the user could decide to spend more time using the service. The RMSE is pretty acceptable considering that we have few predictors, but both User and Movie effects are power enough to predict the rating that will be given to a movie, by a specific user.