Report on Capstone/MovieLens-project 02/2022

alex-stocker

04.02.2022

Contents

Executive Summary

This report describes my work in the Capstone MovieLens project in order to develop a recommender system for movies using the 10M MovieLens dataset. In my project report I elaborate on data exploration and preprocessing. Further, I introduce all the different models that I created and try to give a rationale for them. I test their effectiveness on a test set. I finally decide on the model with the highest RMSE value and then apply this model to the validation dataset.

Introduction and motivation

The application of machine learning algorithms is one of the biggest challenges in data science. Recommender systems are one of the most important use cases for machine learning.

This is the report on the Capstone "Movielens" project, which is part of the HarvardX PH125.9x Data Science course. The Movielens datasets are provided by [GroupLens Research] (https://grouplens.org/datasets/movielens/) in various sizes. These datasets were collected as part of Netflix's movie recommendation competition, the [Netxflix Prize] (https://www.netflixprize.com/). To win the Netflix challenge, the participants had to improve the algorithm of Netflix by at least 10% for $1 million.Therefore, a root mean squared error (**RMSE**) score of of about **0.857** had to be achieved. GroupLens Research has launched its own movie recommendation service and created its own datasets to train its recommendation system.

The goal of this capstone project "Movielens" was to **develop a recommendation system for movies** using the smaller [10M version of the MovieLens dataset] (http://grouplens.org/datasets/movielens/10m/) to simplify the calculations. This dataset contains 10000054 ratings and 95580 tags for 10681 movies assigned by 71567 different users of the MovieLens online movie recommendation service. Here, users give a 1-5 star rating to different films depending on how they found those films. This is a more complicated machine learning challenge because each outcome has a different set of predictors. For example, different users rate a different number of movies and they also rate different movies. There are many types of biases in the movie reviews that have to be tackled.

Methods/analysis

Data Ingestion

The code provided in the Capstone module downloads the MovieLens data and generates two datasets, (1) the **edx set** that is used to train the algorithm, and (2) **the validation set** (the final hold-out test set) to evaluate how close the predictions are to the true values in the validation set at the end of the project.

As the validation set may not be used to test the RSME of multiple models (as stated in the introduction of the course) the edx data has to be split into separate **training** and **test datasets** to first design and then test the models.

Hence there will be three different datasets in my project, (1) a training dataset for model training, (2) a test dataset for model validation through a 80:20 split of the edx dataset and (3) a validation dataset for the final validation of the best model

Data Exploration

As a first step, I complete a series of exercises to better understand what is in the dataset as well as the characteristics of the data to analyse. These includes some general analysis such as looking at the structure of the dataset, counting rows and columns, looking at user ratings, counting different users and films, and looking at film ratings for specific film genres.

The table below shows whats inside the **MovieLens dataset**.

userId	movieId	rating	timestamp	title	genres
1	122	5	838985046	Boomerang (1992)	Comedy|Romance
1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi

<table>
<tr><th>userId</th><th>movieId</th><th>rating</th><th>timestamp</th><th>title</th><th>genres</th></tr>
<tr><td>1</td><td>355</td><td>5</td><td>838984474</td><td>Flintstones, The (1994)</td><td>Children|Comedy|Fantasy</td></tr>
</table>

The table below shows the structure of the MovieLens dataset.

```
## Classes 'data.table' and 'data.frame':   9000055 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  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 83
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|Ac
##  - attr(*, ".internal.selfref")=<externalptr>
```

|| || || ||

The table below shows a summary of the MovieLens dataset.

	userId	movieId	rating	timestamp	title	genres
	Min. : 1	Min. : 1	Min. :0.500	Min. :7.897e+08	Length:9000055	Length:9000055
	1st Qu.:18124	1st Qu.: 648	1st Qu.:3.000	1st Qu.:9.468e+08	Class :character	Class :character
	Median :35738	Median : 1834	Median :4.000	Median :1.035e+09	Mode :character	Mode :character
	Mean :35870	Mean : 4122	Mean :3.512	Mean :1.033e+09	NA	NA
	3rd Qu.:53607	3rd Qu.: 3626	3rd Qu.:4.000	3rd Qu.:1.127e+09	NA	NA
	Max. :71567	Max. :65133	Max. :5.000	Max. :1.231e+09	NA	NA

The next part of my data exploration was part of the **R quiz** to complete.

Q1: There are 9000055 rows and 6 columns are in the edx dataset

Q2: There are 0 zeros and 2121240 threes given as ratings in the edx dataset.

Q3: There are 10677 different movies in the edx dataset.

Q4: There are 69878 different users in the edx dataset.

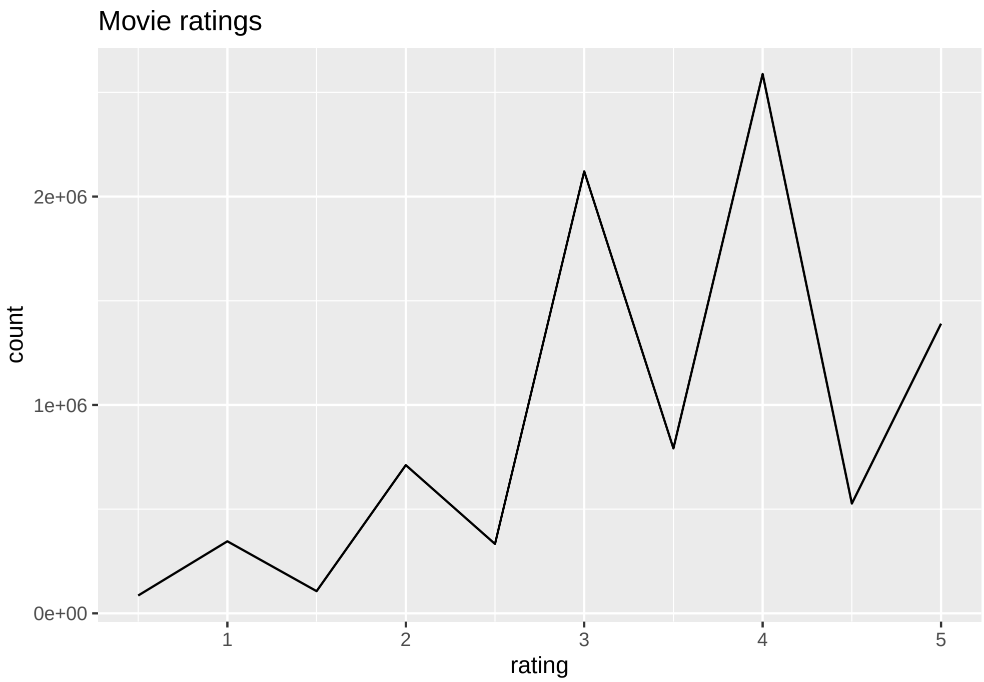
Q5: There are 3910127 movie ratings in the genre "Drama", 3540930 movie ratings in the genre "Comedy", 2325899 movie ratings in the genre "Thriller", and 1712100 movie ratings in the genre "Romance".

Q6: The movie "Pulp Fiction (1994)" has the greatest number of ratings.

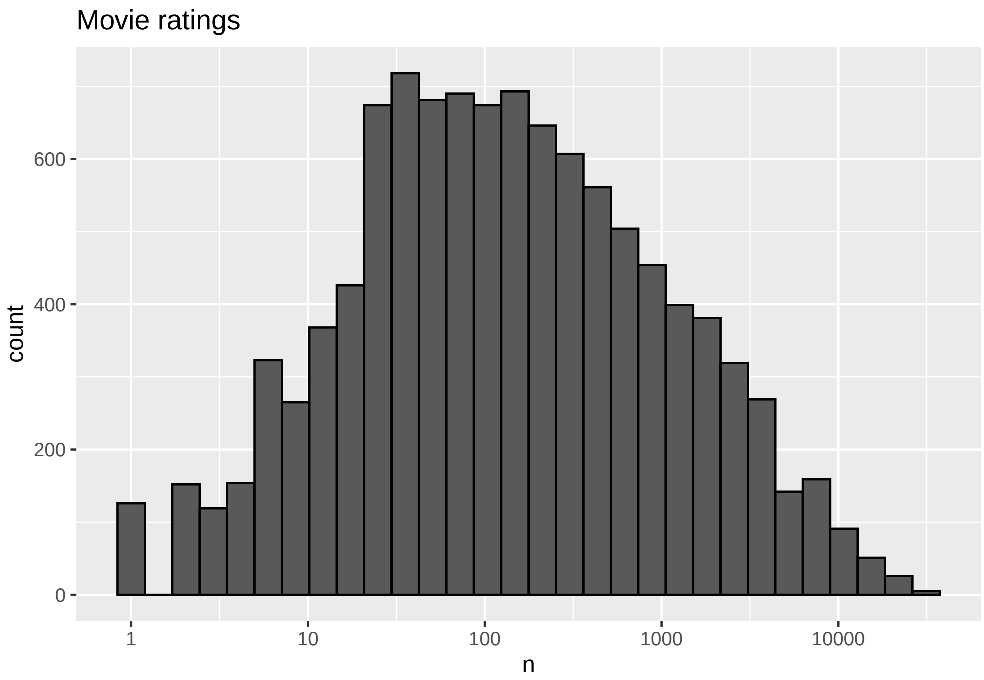
Q7: The following table shows the five most given movie ratings from most to least.

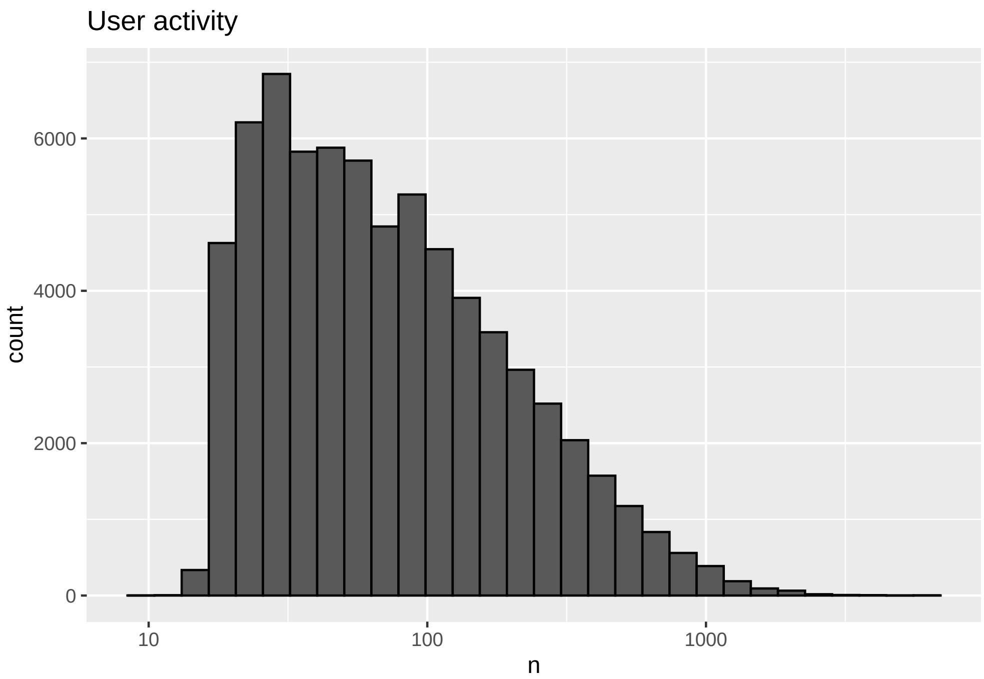
rating	number
4.0	2588430
3.0	2121240
5.0	1390114
3.5	791624
2.0	711422

Q8: In general, half star ratings are less common than whole star ratings. This is shown in the subsequent plot.



Further data exploration included to take a look at the distribution of movie ratings delivering two results: Some movies get many more ratings than other movies and some users are much more active then other users.





Data preparation

Training and test datasets

Before starting to work on my machine learning algorithm using the inputs in one subset to predict the movie ratings in the final validation set, I will first split the edx data into separate **training** and **test sets** to design and test my algorithm as often practiced during the machine learning course.

This is important as the **validation data** (i.e. the final hold-out test set) should NOT be used for training, developing, or selecting my algorithm and **ONLY be used for evaluating the RMSE of my final algorithm**.

I have to make sure to not include movies in my test set that dot appear in my training set and will remove those entries using the semi_join command.

Computing RMSE scores

I write a function that computes the RMSE for vectors of ratings and their corresponding predictors. I will use this function to benchmark the models I further introduce in the report.

```
RMSE <- function(true_ratings, predicted_ratings){
  sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

Model training

In the following section, I first present the developed models and second the results of their validation with the test set.

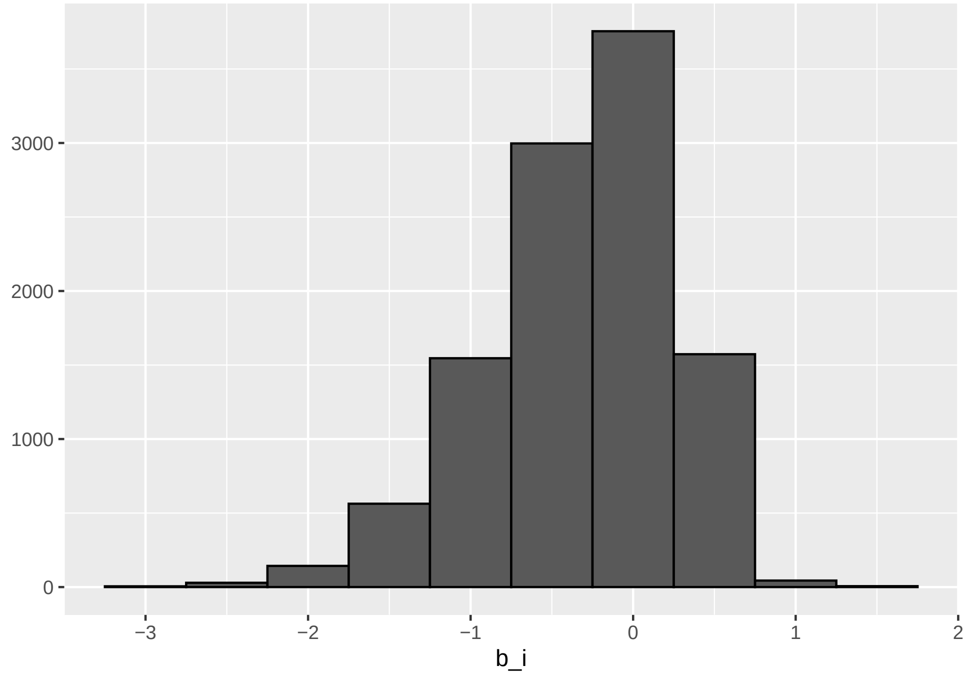
Mean only rating model

The simplest recommendation system is to predict the same rating for all movies, i.e. the mean of all ratings.

The **mean of all ratings** is 3.5124568.

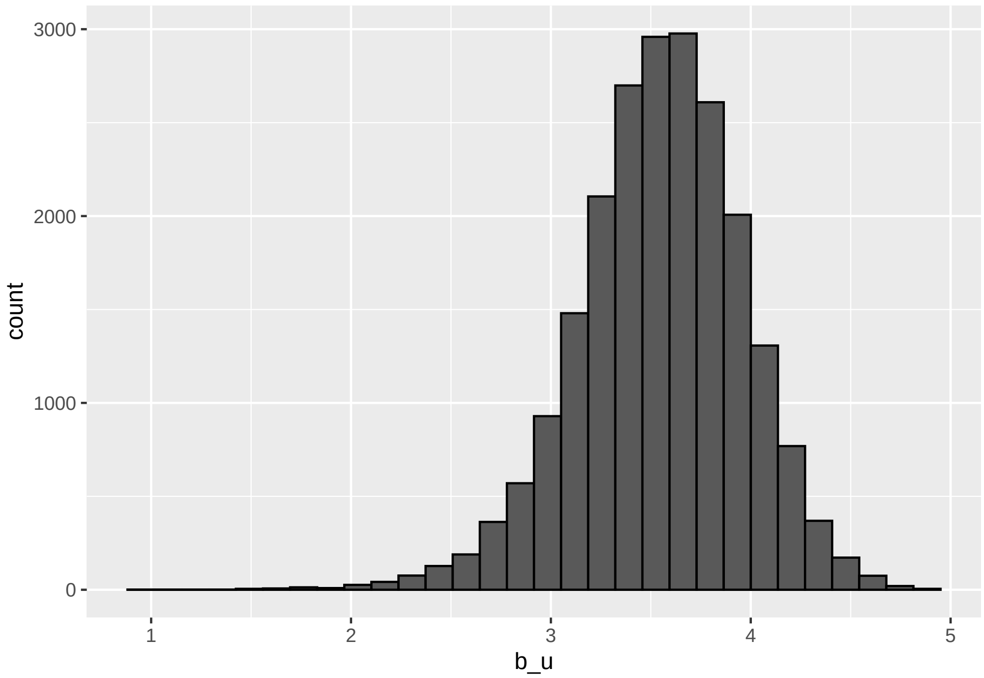
Movie Effect Model

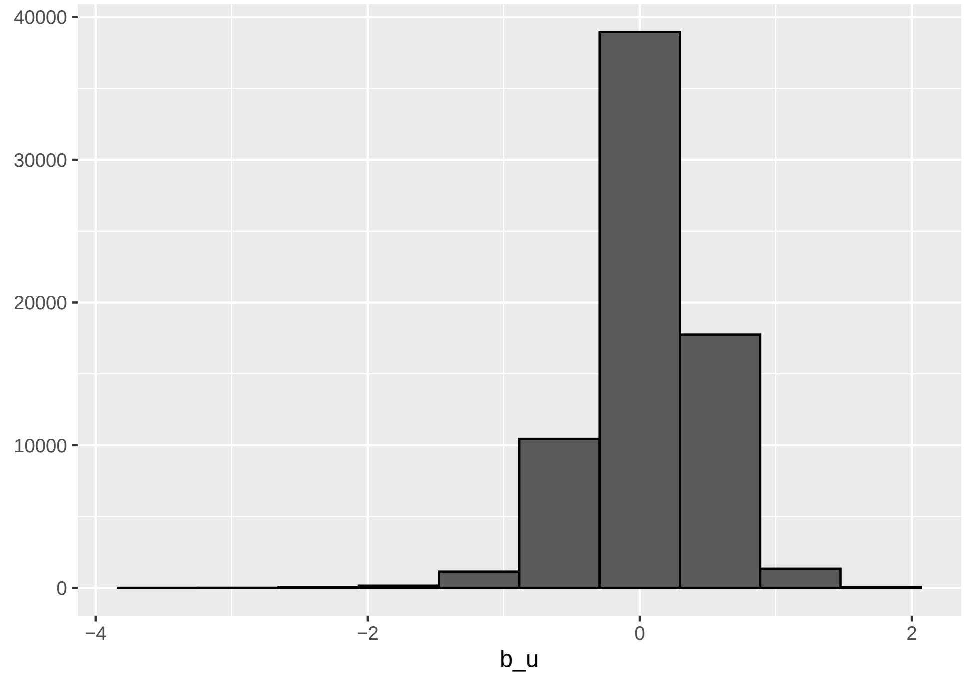
The second recommendation system considers movie effects: Some movies are in general rated higher than others. Hence, movie ratings are subtracted by the mean for each rating that the movie received. "b_i" is introduced as penalty term for the movie effect.



Movie + User Effect Model

The second recommendation system considers movie and user effects: Some users rate movies generally higher than others do. "b_u" is introduced as penalty term for movie effect.





A further exploration of the data delivered interesting results: The supposed “best” and “worst” movies were rated by very few users, in most cases just 1. These noisy estimates should not be trusted.

```
##  [1] "From Justin to Kelly (2003)"      "PokÃ©mon Heroes (2003)"
##  [3] "PokÃ©mon Heroes (2003)"           "Shawshank Redemption, The (1994)"
##  [5] "Shawshank Redemption, The (1994)" "Shawshank Redemption, The (1994)"
##  [7] "Shawshank Redemption, The (1994)" "Shawshank Redemption, The (1994)"
##  [9] "Shawshank Redemption, The (1994)" "Shawshank Redemption, The (1994)"
## [11] "Shawshank Redemption, The (1994)" "Shawshank Redemption, The (1994)"
## [13] "Shawshank Redemption, The (1994)" "Shawshank Redemption, The (1994)"
## [15] "Shawshank Redemption, The (1994)"
```

This are the *ten best rated* movies.

```
##  [1] "Hellhounds on My Trail (1999)"
##  [2] "Satan's Tango (SÃ¡tÃ¡ntangÃ³) (1994)"
##  [3] "Shadows of Forgotten Ancestors (1964)"
##  [4] "Fighting Elegy (Kenka erejii) (1966)"
##  [5] "Sun Alley (Sonnenallee) (1999)"
##  [6] "Blue Light, The (Das Blaue Licht) (1932)"
##  [7] "Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1980)"
##  [8] "Life of Oharu, The (Saikaku ichidai onna) (1952)"
##  [9] "Human Condition II, The (Ningen no joken II) (1959)"
## [10] "Human Condition III, The (Ningen no joken III) (1961)"
```

This are the *ten worst rated* movies.

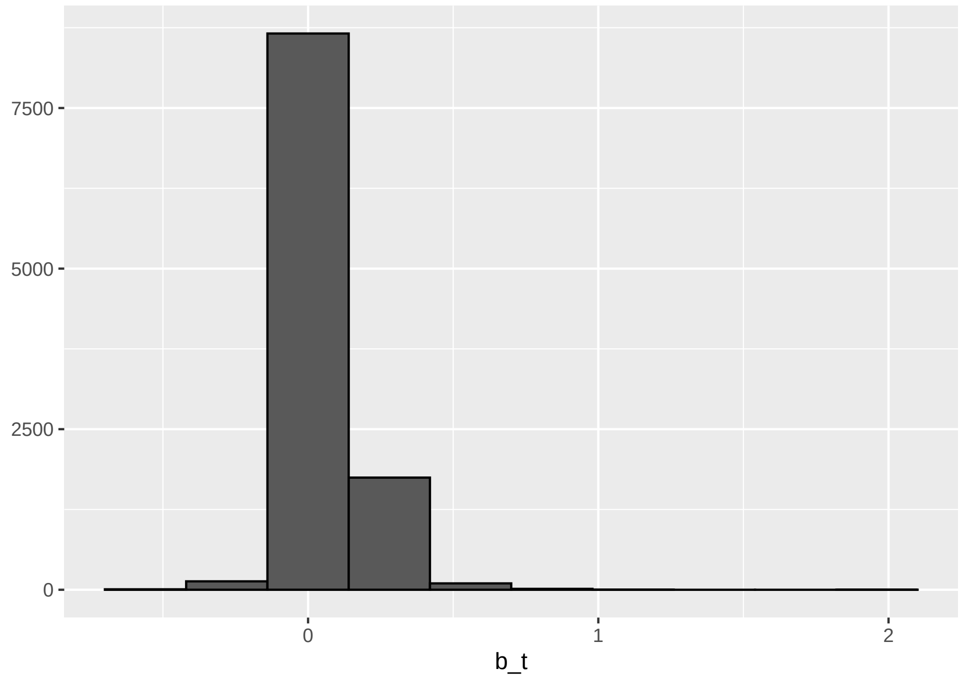
```
##  [1] "Besotted (2001)"
##  [2] "Hi-Line, The (1999)"
##  [3] "Confessions of a Superhero (2007)"
##  [4] "War of the Worlds 2: The Next Wave (2008)"
##  [5] "SuperBabies: Baby Geniuses 2 (2004)"
##  [6] "Disaster Movie (2008)"
##  [7] "From Justin to Kelly (2003)"
##  [8] "Hip Hop Witch, Da (2000)"
##  [9] "Criminals (1996)"
## [10] "Mountain Eagle, The (1926)"
```

The supposed "best" and "worst" movies were rated by very few users, only, in most cases just one. These noisy estimates should not be trusted. The result below shows how often the movies are rated.

```
##  [1] 1 1 1 1 1 1 4 2 4 4
```

Movie + User + Title Effect Model

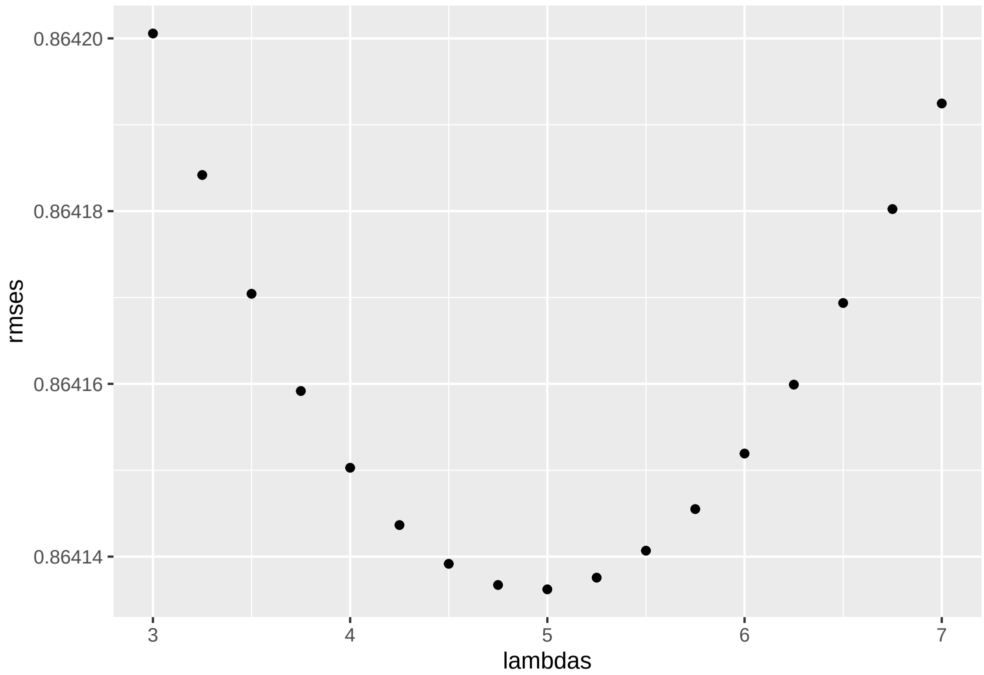
The subsequent analysis takes the title effect into account: Thereby, "b_t" is introduced as penalty term for the title(-rating) effect.



Regularized Movie + User Effect Model

I should not trust noisy estimates. Hence, I apply the concept of regularization to further improve on our RMSE score. Through regularization I can penalize large estimates formed using small sample sizes. In the following model, I use penalized least squares and add penalties. This allows me to control the total variability of the effects. I further improve the model by regularizing not only the movie, but also the user effect. This should also further improve the RMSE score. I introduce lambda as a tuning parameter and use cross validation to choose the ideal lambda. The RMSE score is shown below.

```
## [1] 0.8641362
```

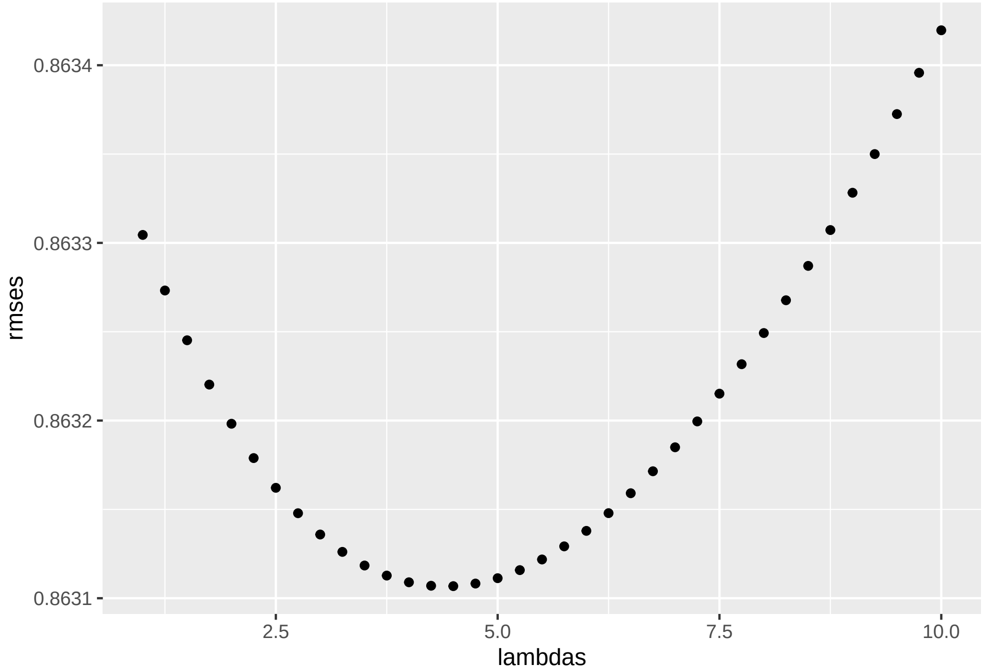


The optimal lambda for this model is **5**.

Regularized Movie + User + Title Effect Model

Finally, I further improve the model by also regularizing the title effect which should again improve the RMSE score shown below.

```
## [1] 0.8631068
```



The optimal lambda for this model is **4.5**.

Model validation

In this subsection, I show the results of the validation of the different models with the test data in order to identify the model with the highest RSME score and then finally validate the winning model with the validation data.

Model validation with testset

The table shows the **RSME scores** for all **models** developed. Only the model with the highest RSME score will be used with the validation dataset.

The subsequent table shows the RSME score for the test dataset.

Method	RMSE
Mean	1.0600537
+ Movie Effect	0.9429615
+ User Effect	0.8646844
+ Title Effect	0.8634655
Regularized Movie + User Effect	0.8641362
Regularized Movie + User + Title Effect	0.8631068

The model **Regularized Movie + User + Title Effect** gives the best RMSE score of **0.8631068** on the test dataset.

Model validation with validation dataset

The best model **Regularized Movie + User + Title Effect** is finally validated using the **validation dataset**. There are in total 999999 rows and 6 columns in this dataset.

The final RSME score using the best model **Regularized Movie + User + Title Effect** is **0.8645849**

Conclusion

This report presents my work in the Capstone MovieLens project related to the development of a recommender system for the MovieLens dataset. First, I examined the dataset to better understand its structure and what it contains. I performed some general analyses and computed some graphs. Furthermore, I started developing models for recommender systems, starting with the simplest model, the mean score model, using the training set. I validated all the models using the test set.

Finally, I validated the model with the best performance on the validation set. My final RMSE result for the model **Regularized Movie + User + Title Effect** against the validation dataset is **0.8645849**. This score is **<** **0.86490**, the value to beat, hence I successfully completed the Capstone MovieLens challenge.