HarvardX Data Science Capstone Project: Predicting Movie Ratings in the MovieLens 10M Dataset

Venkata Earanti

July 17, 2021

Contents

Summary

As part of HarvardX Data Science Capstone Project, I worked on creating a movie recommendation system using the Movie Ratings in the MovieLens 10M Dataset.

Recommendation systems allow the customer to rate, gather the user ratings and business analyze the data to predict user behavior.

The goal of this project is to train a machine learning algorithm that predicts user movie ratings using the inputs of a supplied subset to predict film ratings in a supplied validation set. The term used for evaluating efficiency of the algorithm is the Root Mean Square Error, or RMSE. RMSE is one of the most widely used measurement of the differences between the values predicted by a model and the observed values. RMSE is a measure of accuracy, to compare forecasting errors of different models for a particular dataset, An RMSE of 0 means we are always correct, not a possibility. An RMSE of 1 means the predicted ratings are off by 1 star, a lower RMSE is better than a higher one. The effect of increasing error on RMSE is proportional to the size of the squared error; hence, larger errors impact RMSE disproportionately.

Data will be divided into two data sets, edx for training and validation for the data validation purpose. After the data analysis, models will be developed and compared to arrive to a conclusion.

Data Analysis

Data preparation

```
knitr::opts_chunk$set(error = TRUE)
dl <- tempfile()
download.file("http://files.grouplens.org/datasets/movielens/ml-10m.zip", dl)
ratings <- fread(text = gsub("::", "\t", readLines(unzip(dl, "ml-10M100K/ratings.dat"))),
                 col.names = c("userId", "movieId", "rating", "timestamp"))

movies <- str_split_fixed(readLines(unzip(dl, "ml-10M100K/movies.dat")), "\\::", 3)
colnames(movies) <- c("movieId", "title", "genres")

movies <- as.data.frame(movies) %>% mutate(movieId = as.numeric(movieId),
                                           title = as.character(title),
                                           genres = as.character(genres))
movielens <- left_join(ratings, movies, by = "movieId")

# Validation set will be 10% of MovieLens data
set.seed(1) # if using R 3.5 or earlier, use `set.seed(1)`
test_index <- createDataPartition(y = movielens$rating, times = 1, p = 0.1, list = FALSE)
edx <- movielens[-test_index,]
temp <- movielens[test_index,]

# Make sure userId and movieId in validation set are also in edx set
validation <- temp %>%
     semi_join(edx, by = "movieId") %>%
     semi_join(edx, by = "userId")
# Add rows removed from validation set back into edx set
removed <- anti_join(temp, validation)
```

```
## Joining, by = c("userId", "movieId", "rating", "timestamp", "title", "genres")
```

```
edx <- rbind(edx, removed)
rm(dl, ratings, movies, test_index, temp, movielens, removed)

#Algorithm will be developed on Edx data set and validation data set will be used
#for testing the final algorithm.
```

Exploratory Data Analysis

Exploration/Validation 1

Get first few rows of the edx and get familiarize with 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	231	5	838983392	Dumb & Dumber (1994)	Comedy
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

- edx data set contain the six variables 'userID', 'movieID', 'rating', 'timestamp','title', and 'genres'.
- Each row represent a single rating of a user for a single movie.
- Rating is the target variable,the value we are tryng to predict

Exploration/Validation 2

Check if there are any missing values

```
##      userId         movieId          rating        timestamp
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08
##  1st Qu.:18122   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08
##  Median :35743   Median : 1834   Median :4.000   Median :1.035e+09
##  Mean   :35869   Mean   : 4120   Mean   :3.512   Mean   :1.033e+09
##  3rd Qu.:53602   3rd Qu.: 3624   3rd Qu.:4.000   3rd Qu.:1.127e+09
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09
##     title              genres
##  Length:9000061     Length:9000061
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

- The summary of the edx confirms that there are no missing values.

Exploration/Validation 3

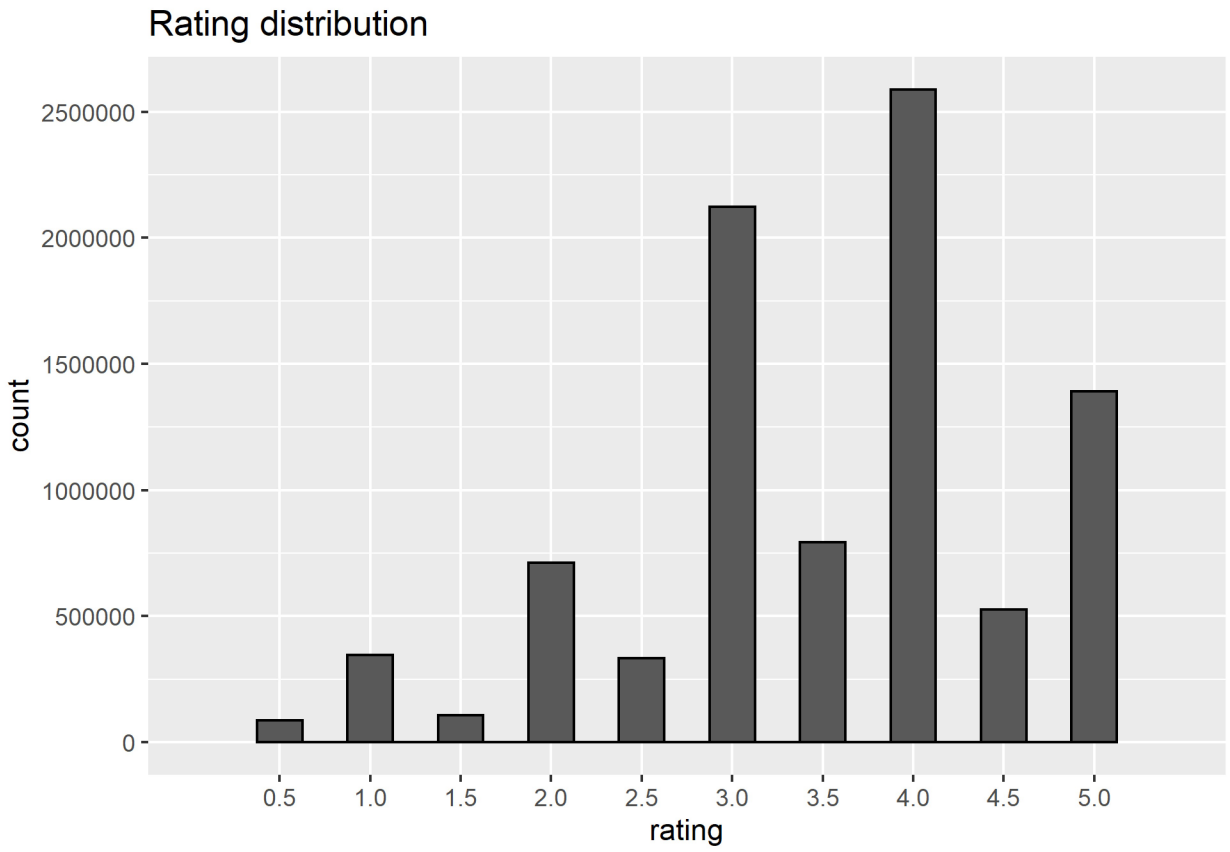
Check total of unique movies and users

Unique users - 69878
Unique movies - 10677

n_users	n_movies
69,878	10,677

Exploration/Validation 4

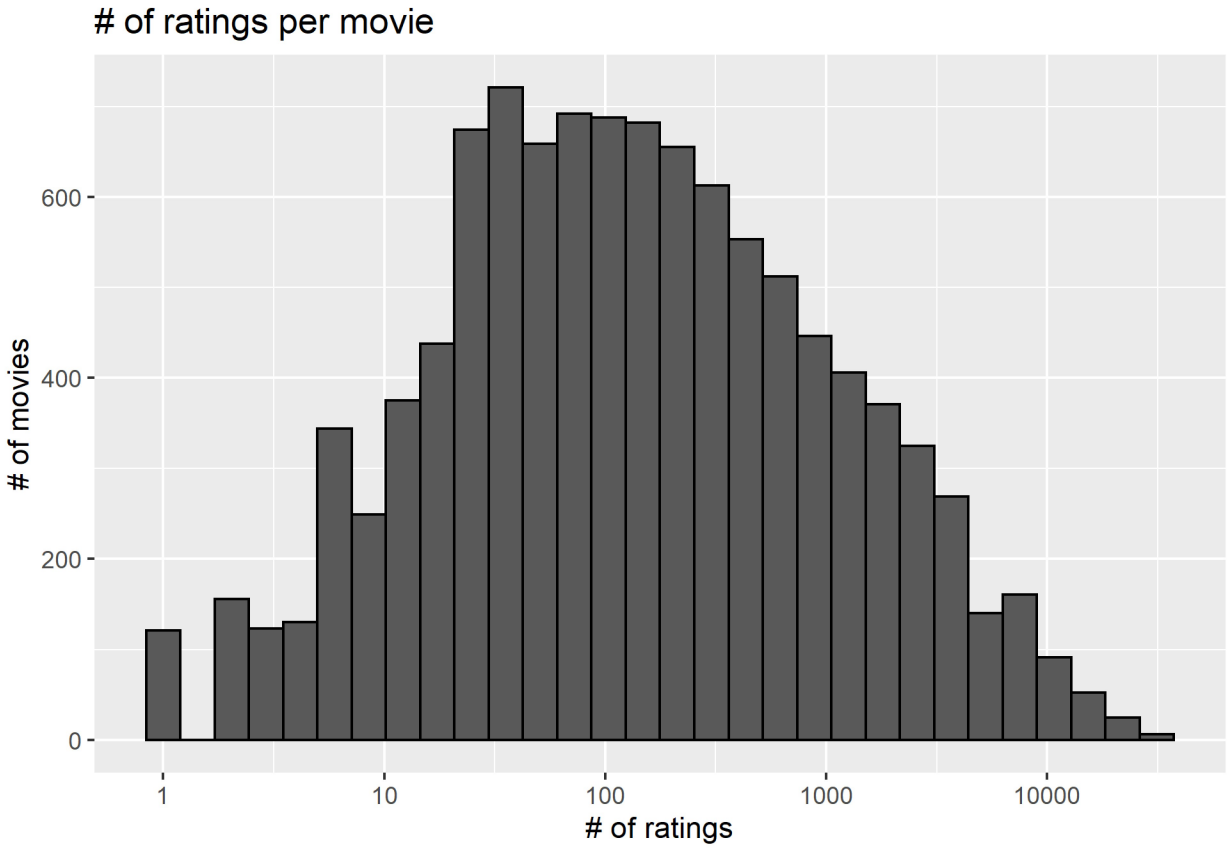
Review ratings distribution



- Based on the Rating Distribution Diagram, most users rated 4. 0.5 is the least rated.
- More Users tends to give "Full-Star" rating, few users gave "Half-Star" rating
- If we consider anything less than 3 is negative, there are small number of negative ratings

Exploration/Validation 5

Review number of ratings per movie



- Some movies (approx 125) were rated only one time (very low)
- Some movies were rated more often than others
- Low rating number could impact the quality of the prediction

Exploration/Validation 6

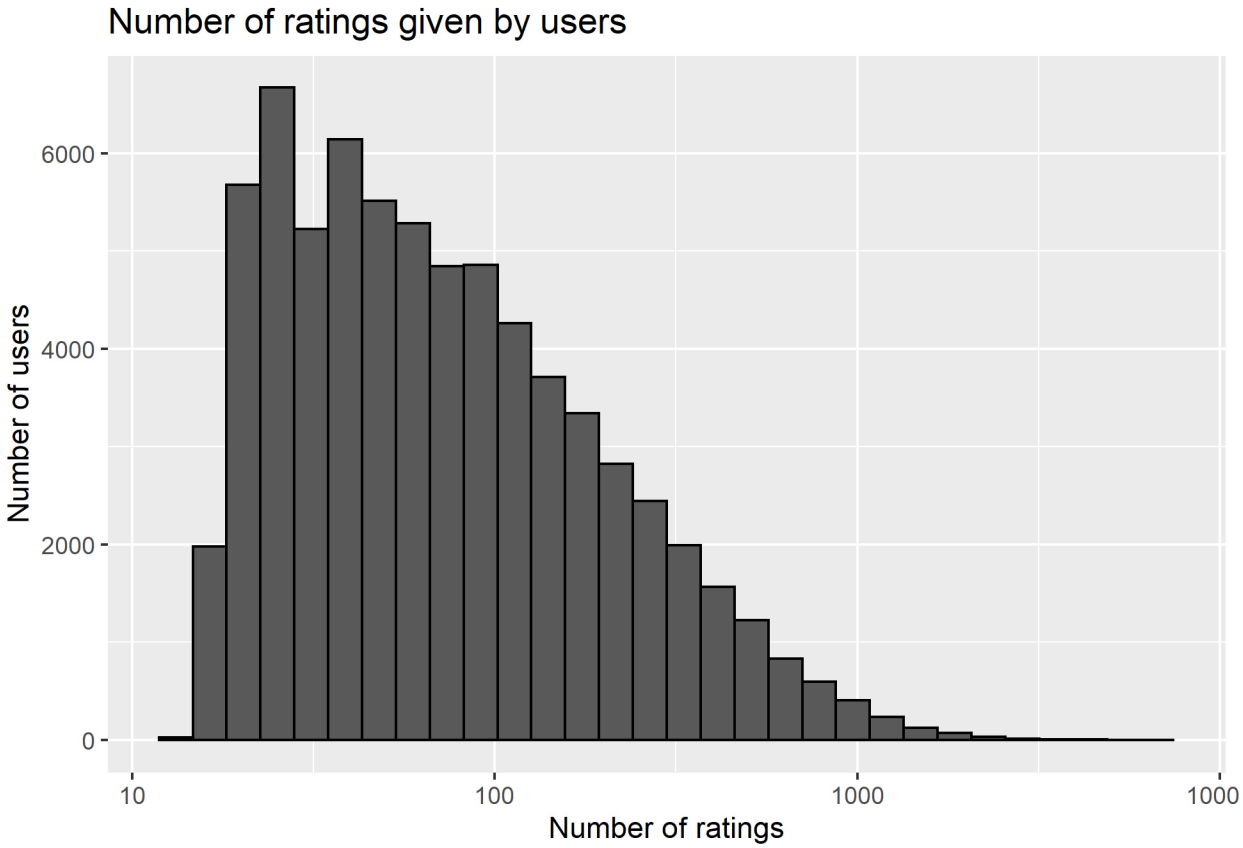
Movies rated only once

title	rating	n_rating
100 Feet (2008)	2.0	1
4 (2005)	2.5	1
5 Centimeters per Second (ByÃ´soku 5 senchimÃªtoru) (2007)	3.5	1
Accused (Anklaget) (2005)	0.5	1
Ace of Hearts (2008)	2.0	1
Ace of Hearts, The (1921)	3.5	1
Adios, Sabata (Indio Black, sai che ti dico: Sei un gran figlio di...) (1971)	1.5	1
Africa addio (1966)	3.0	1
Archangel (1990)	2.5	1
Bad Blood (Mauvais sang) (1986)	4.5	1
Battle of Russia, The (Why We Fight, 5) (1943)	3.5	1
Bell Boy, The (1918)	4.0	1
Black Tights (1-2-3-4 ou Les Collants noirs) (1960)	3.0	1
Blind Shaft (Mang jing) (2003)	2.5	1
Blue Light, The (Das Blaue Licht) (1932)	5.0	1
Borderline (1950)	3.0	1
Boys Life 4: Four Play (2003)	3.0	1
Brothers of the Head (2005)	2.5	1
CaÃ³tica Ana (2007)	4.5	1
Chapayev (1934)	1.5	1

- some movies are rated only once,predictions of future ratings for them will be difficult.

Exploration/Validation 7

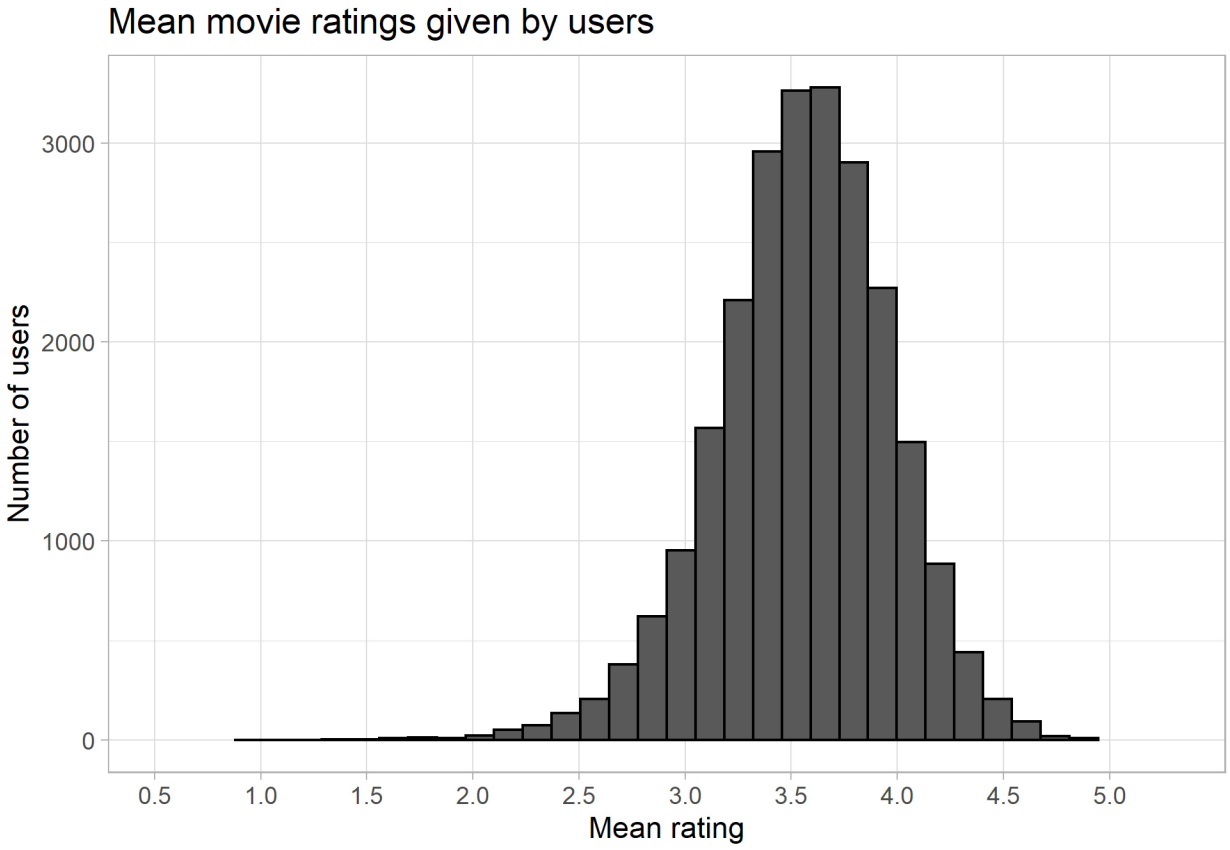
Review number of ratings given by users



- Majority of the users rated 30-100 movies

Exploration/Validation 8

Review mean movie ratings by users



- Looking at users who rated at least 100 movies, some users gave much lower rating and some gave much higher than average.
- Users differe on how critical they are.

Approach

RMSE is defined as follows

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

The RMSE is our measure of model accuracy

1. Average rating (Naive Baseline) Model

The first basic model to predict the same rating for all movies is using dataset's mean rating.

A model based approach assumes the same rating for all movie with all differences explained by random variation :

$$Y_{u,i} = \mu + \epsilon_{u,i}$$

with $\epsilon_{u,i}$ independent error sample from the same distribution

```
## [1] 3.512464
```

```
## [1] 1.060651
```

The mean is 3.512465

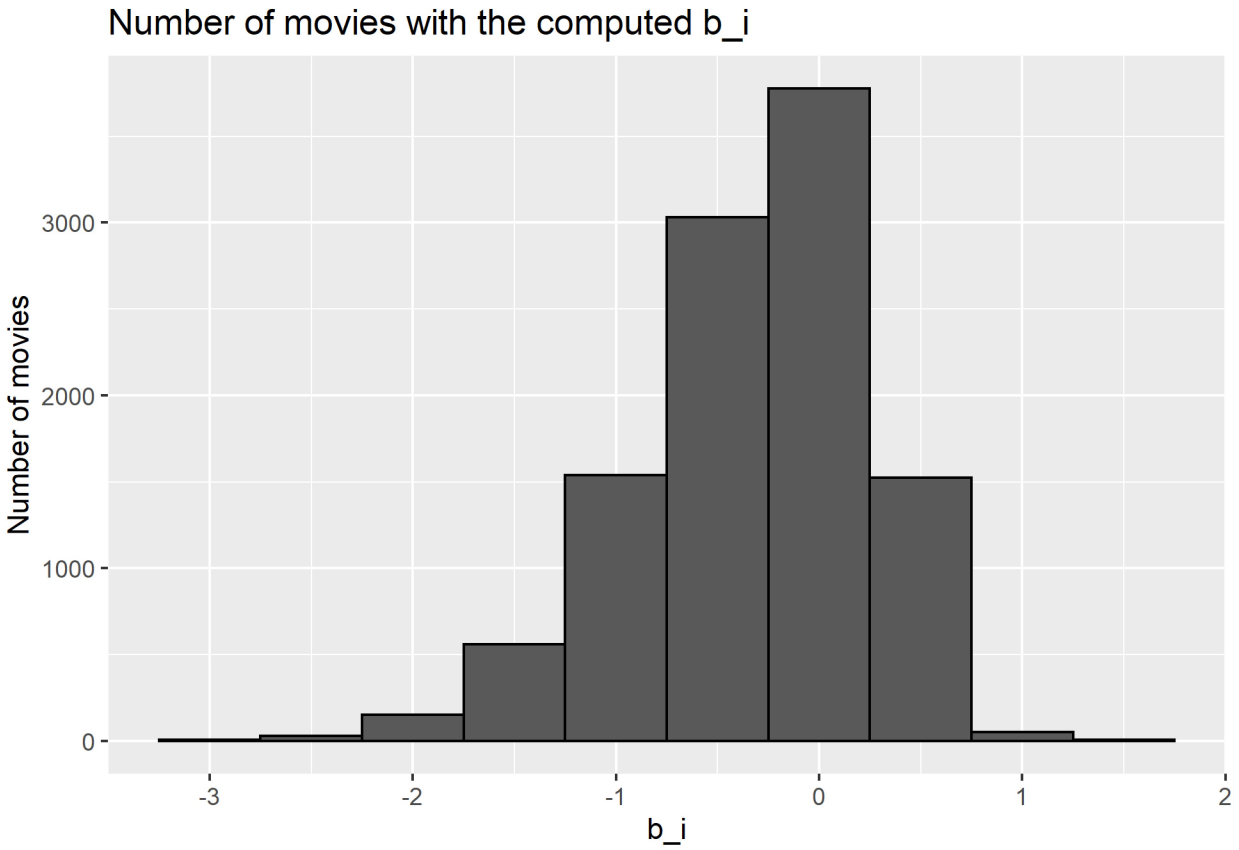
Naive RMSE is 1.061202.

It is very far for the target RMSE (below 0.87) and that indicates poor performance for the model.

method	RMSE
Average movie rating model	1.060651

2.Movie effect model

Based on out data analysis, some movies are just generally rated higher than others. Higher ratings are mostly linked to popular movies among users and the opposite is true for unpopular movies. We compute the estimated deviation of each movies' mean rating from the total mean of all movies μ. The resulting variable is called "b" (as bias) for each movie "i" b_i, that represents average ranking for movie i: $\$Y_{u,i} = \mu + b_i + \epsilon_{u,i}$



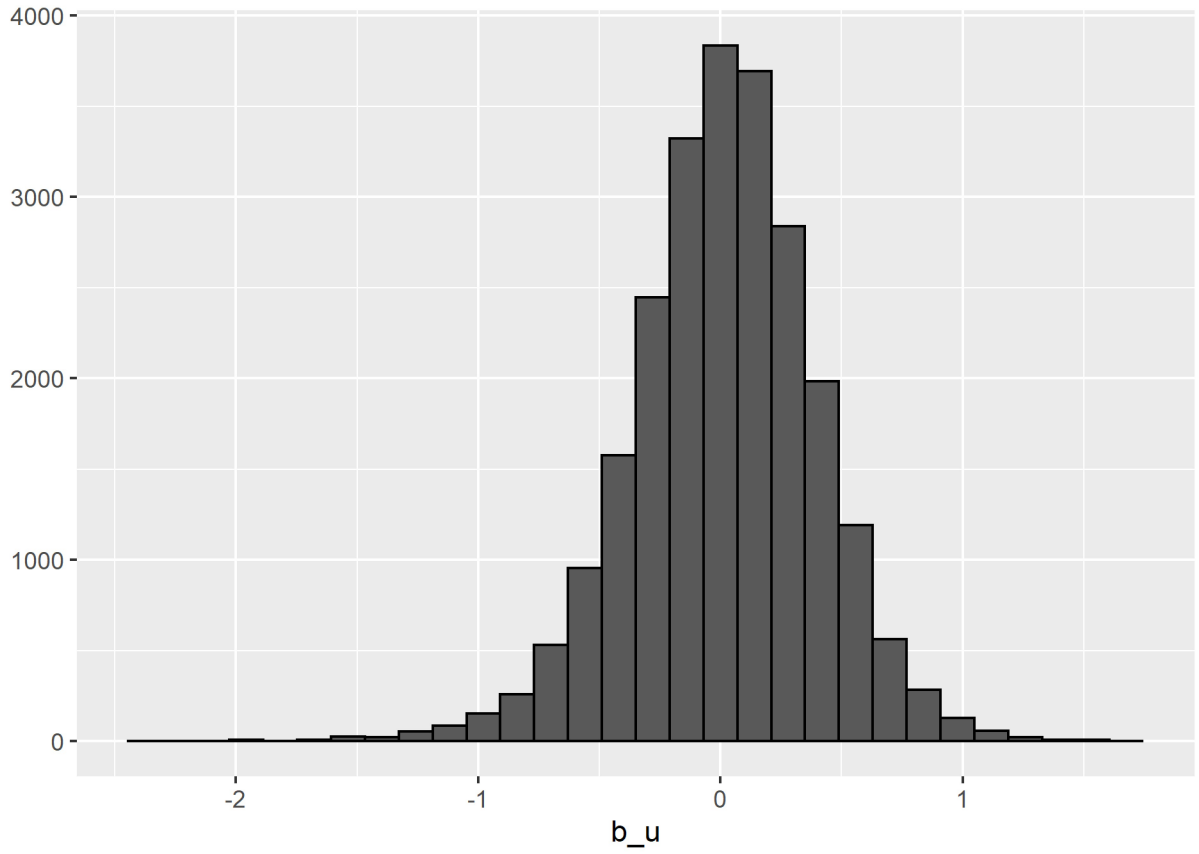
Movie effect model RMSE is 0.9439087

method	RMSE
Average movie rating model	1.0606506
Movie effect model	0.9437046

3.Movie and user effect model

The above model do not consider individual user rating effect, that was found in exploration/validation 8. Compute the average rating for user μ, for those that have rated over 100 movies.

```
## `summarise()` ungrouping output (override with `.groups` argument)
```




```
## [1] 0.8655329
```

Movie and user effect model RMSE is 0.8653488

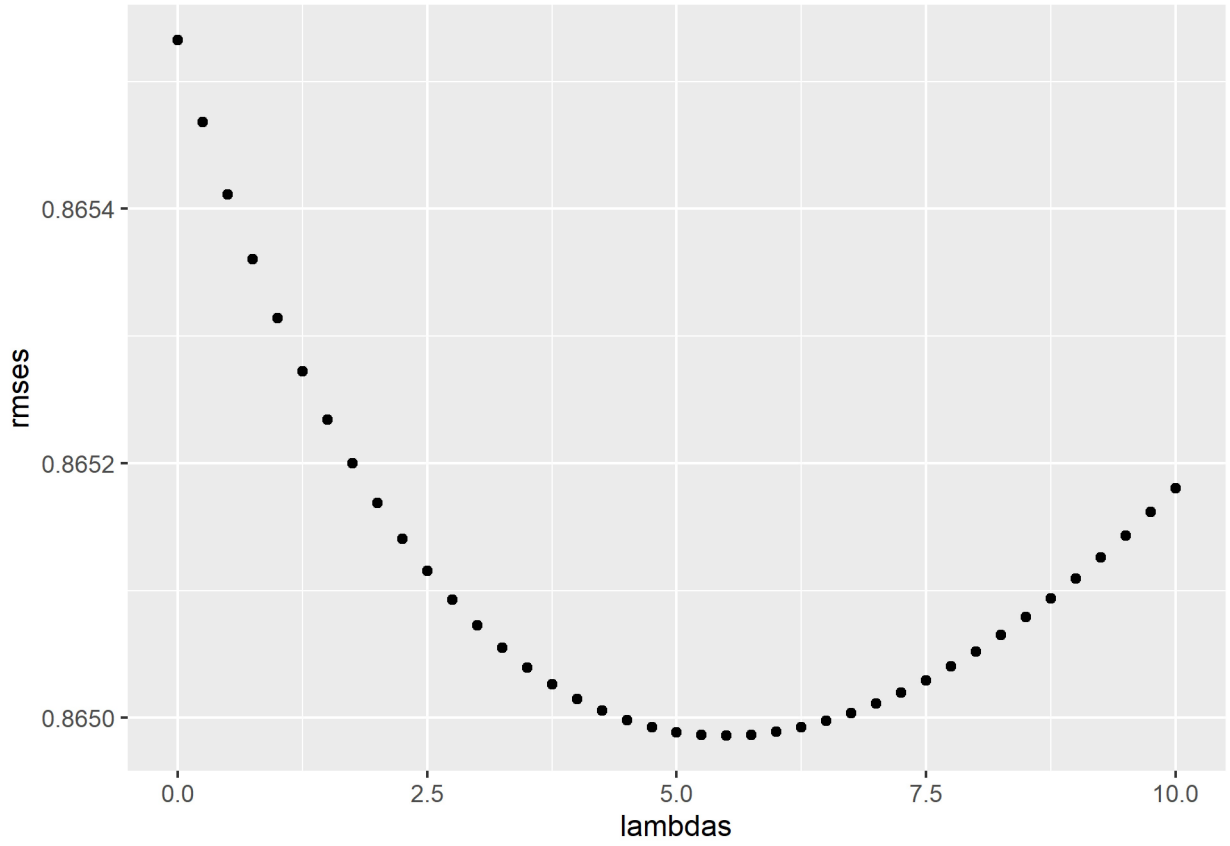
method	RMSE
Average movie rating model	1.0606506
Movie effect model	0.9437046
Movie and user effect model	0.8655329

4.Regularized movie and user effect model

In the earlier models we computed standard error and constructed confidence intervals to account for different levels of uncertainty. However, when making predictions, we need one number, one prediction, not an interval.

We introduce the concept of regularization, that permits to penalize large estimates that come from small sample sizes. The idea is to add a penalty for large values of b_i to the sum of squares equation that we minimize. So having many large b_i, make it harder to minimize. Regularization is a method used to reduce the effect of overfitting.

Estimates of b_i and b_u are caused by movies with very few ratings and in some users that only rated a very small number of movies. Hence this can strongly influence the prediction. The use of the regularization permits to penalize these aspects. We should find the value of lambda (that is a tuning parameter) that will minimize the RMSE. This shrinks the b_i and b_u in case of small number of ratings.




```
## [1] 5.5
```

method	RMSE
Average movie rating model	1.0606506
Movie effect model	0.9437046
Movie and user effect model	0.8655329
Regularized movie and user effect model	0.8649857

Regularized movie and user effect model - RMSE - 0.8648170

Optimal Lambda - 5.25

Predictions

We are going to choose the "Regularized movie and user effect model" calculation since it has the lowest RMSE

This gives a sample of rating and predicted rating

userId	movieId	rating	predicted_ratings	timestamp
1	588	5.0	5.037407	838983339
2	1210	4.0	3.700301	868245644
2	1544	3.0	2.645599	868245920
3	151	4.5	3.798682	1133571026
3	1288	3.0	4.332867	1133571035
3	5299	3.0	3.769643	1164885617