Capstone Project Movielens

Michael Hovey

1/22/2022

Contents

0.1 Introduction

Recommendation systems are one of the most used models in machine learning algorithms. Recommendation systems use ratings that users have given to items to make specific recommendations. Companies such as Amazon, Barnes and Noble , and Netflex allow their customers to rate their various products and are able to collect massive datasets that can be used to predict what rating a particular user will give to a specific item. Items that have the highest ratings are predicted for a given user and then offered as recommendations.

For this project I will create a movie recommendation system that recommends movies based on a rating scale.

I will train a machine learning algorithm that predicts user ratings (from 0.5 to 5 stars) using the inputs of a provided subset of data to predict movie ratings in a provided validation set.

The value used to evaluate algorithm performance is the Root Mean Square Error, or RMSE. RMSE is one of the most used measure of the differences between values predicted by a model and the values that are observed. RMSE is a measure of accuracy by comparing forecasting errors of different models for a particular dataset, a lower RMSE is better than a higher one. The effect of each error on RMSE is proportional to the size of the squared error; thus larger errors have a disproportionately large effect on RMSE. Consequently, RMSE is sensitive to outliers. the models that will be developed will be compared using their resulting RMSE in order to assess their quality. The evaluation criteria for this algorithm is a RMSE expected to be lower than 0.8775.

The model with the best results will be used to predict the movie ratings.

##Data set This project uses the MovieLens Data set collected by the GroupLens Research and can be found on the MovieLens web site (http://movielens.org).

The MovieLens dataset will be splitted into 2 subsets incluuding an edx data set , a training subset to train the algorithm, and a validation data set to test the movie ratings.

All development will be performed on the edx data set only, as validation subset will be used to test the final algorithm.

1 Methods and Analysis

1.1 Data Analysis

It is a good habit to familiarize yourself with the dataset, below we will find the first rows of edx data set. The data set contains the six variables: "userID", "movieID", "rating", "timestamp", "title", and "genres". Each row represent a single rating from a user for a single movie.

```
##   userId movieId rating timestamp title genres
## 1      1     122      5 838985046  <NA>   <NA>
## 2      1     185      5 838983525  <NA>   <NA>
## 3      1     231      5 838983392  <NA>   <NA>
## 4      1     292      5 838983421  <NA>   <NA>
## 5      1     316      5 838983392  <NA>   <NA>
## 6      1     329      5 838983392  <NA>   <NA>
```

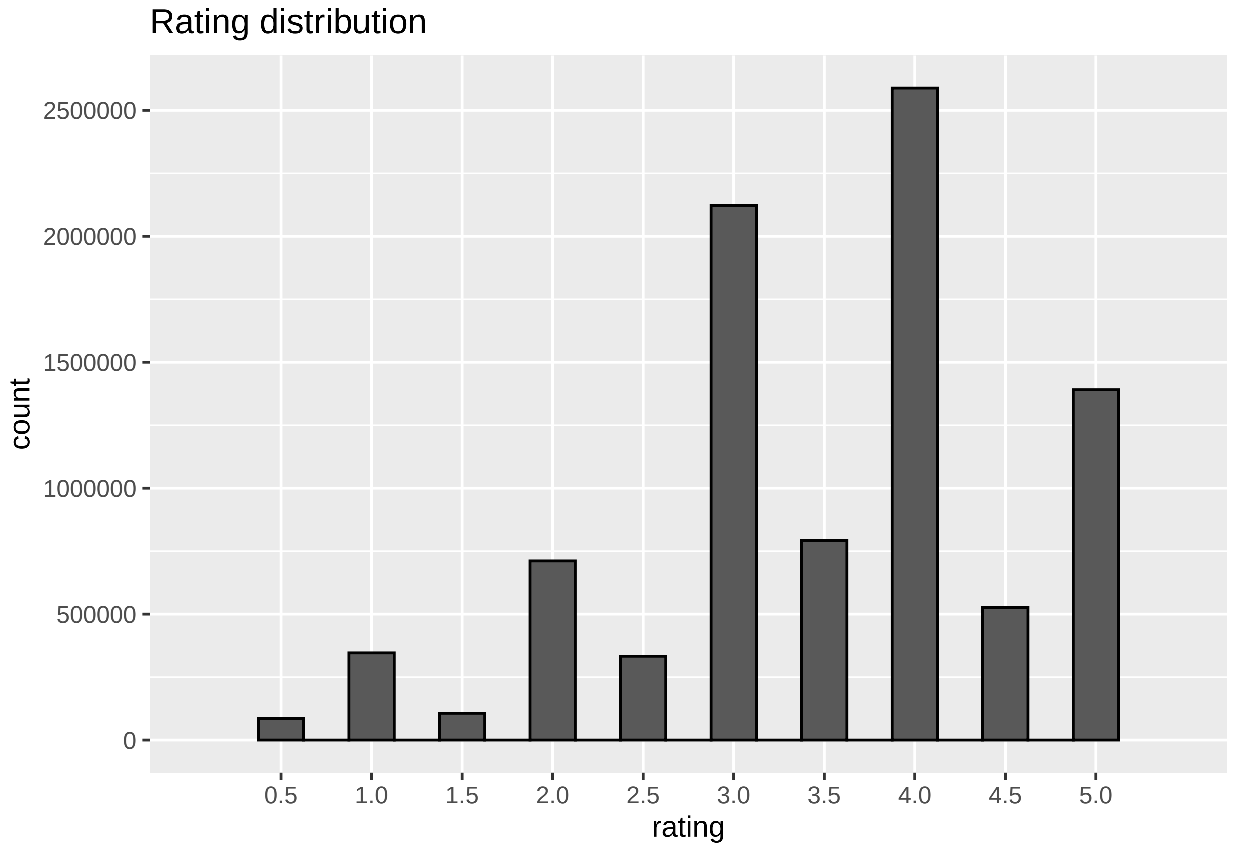
A summary of the data set confirms that there are no missing values.

```
##      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 total of unique movies and users in the edx data set is 69,878 unique users and 10,677 different movies:

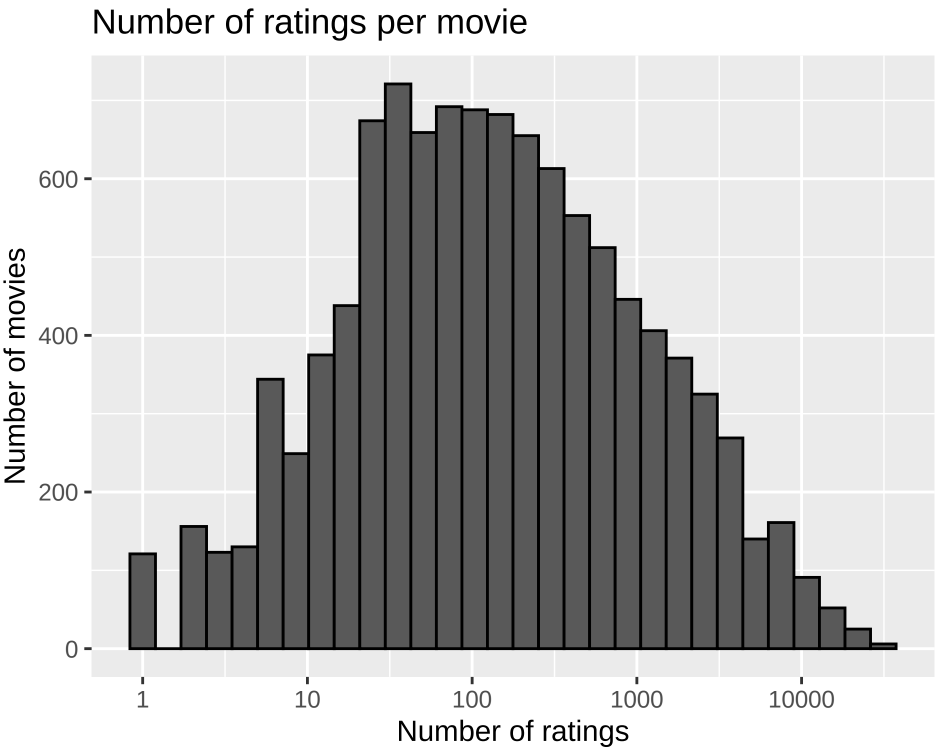
n_users	n_movies
69878	10677

Users tend to rate movies higher than lower as shown by the distribution of movie ratings below. A rating of four is the most common rating, followed by 3 and 5. The lease common rating is 0.5.

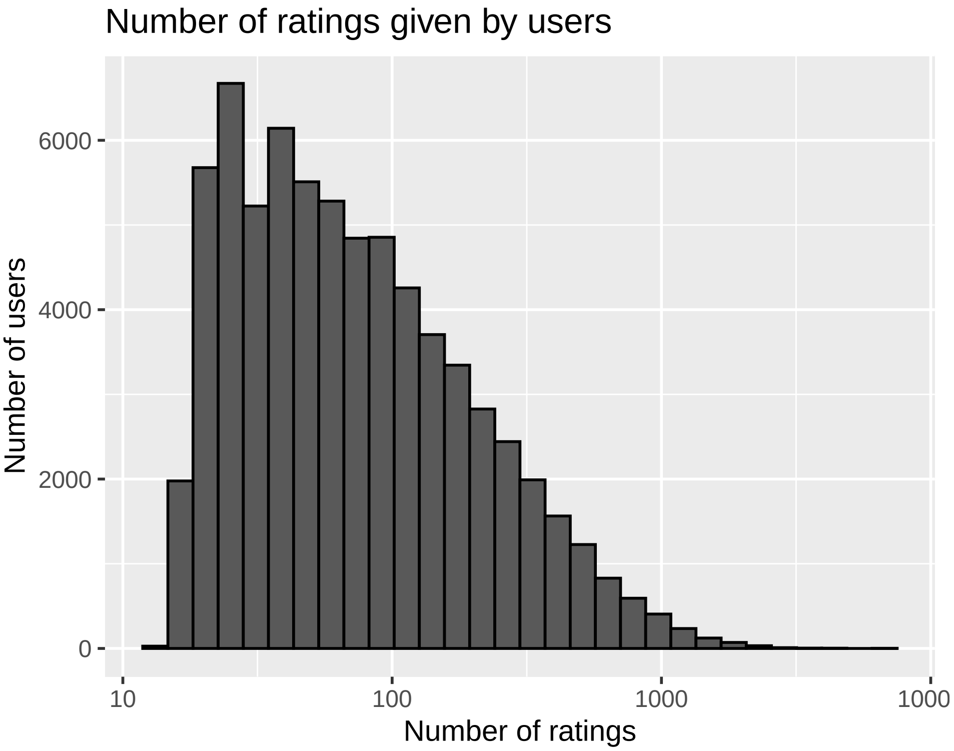


Certain movies are rated more ofetn that others. Movies with low rating numbers can result in untrustworthy estimates for our predictions.

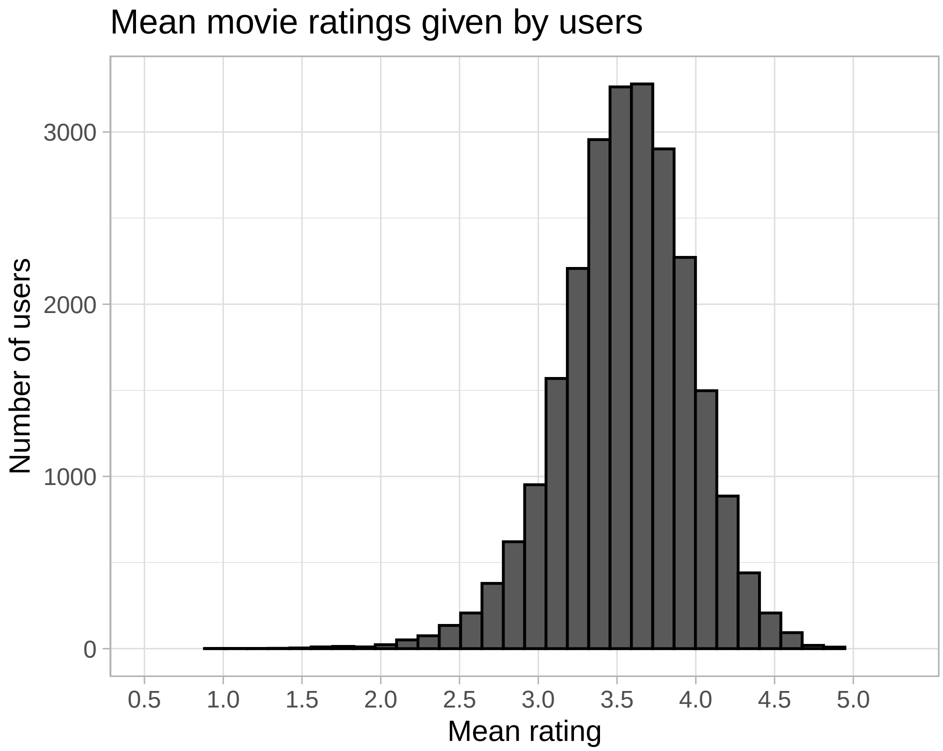
Regularisation and penalty terms can be applied to the data models to prevent this. Regularizations are techniques used to reduce the error by applying a function on the given training set and avoid overfitting (the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably). Regularization is a technique used for tuning the function by adding an additional penalty term in the error function. The additional term controls the excessively fluctuating function such that the coefficients don't take extreme values.


Number of ratings per movie

The majority of users have rated between 30 and 100 movies. So, a user penalty term needs to be included later in our data models.


Number of ratings given by users

Users differ vastly in how critical they are with their ratings. Some users tend to give much lower star ratings and some users tend to give higher star ratings than average. We can include only users that have rated at least one hundread movies to make estimates more accurate.


Mean movie ratings given by users
Number of users
3000
2000
1000
0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Mean rating

1.2 Modelling Approach

Creation of the loss-function, that computes the RMSE, is defined as:

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

with N being the number of user/movie combinations and the sum occurring over all of these combinations. The RMSE is the measure of model accuracy. By interpretting the RMSE to a standard deviation: the typical error made when predicting a movie rating. If its result is larger than 1, it means that the typical error is larger than one star, which is not a good result. The written function to compute the RMSE for vectors of ratings and their corresponding predictions is as follows:

1.2.1 I. The average movie rating model

The first basic model predicts the same rating for all movies, so we compute the dataset's mean rating. The expected rating of the underlying data set is between 3 and 4. We start by building a simple recommendation system by predicting the same rating for all movies regardless of the user who gave it. A model based approach assumes the same rating for all movie with all differences explained by random variation :

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

By predicting all unknown ratings with μ or mu, we obtain the first naive RMSE:

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

The results table with RMSE:

```
## Warning: ‘data_frame()‘ was deprecated in tibble 1.1.0.
## Please use ‘tibble()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was generated.
```

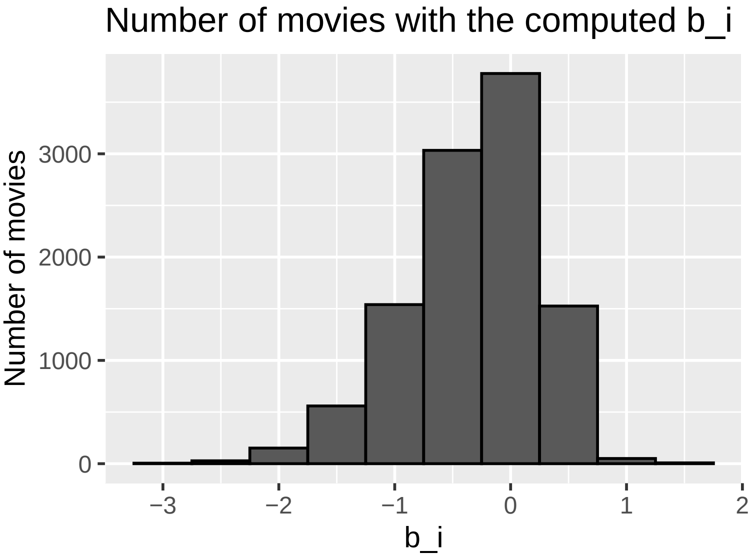
method	RMSE
Average movie rating model	1.060651

This give us our baseline RMSE to compare with next modelling approaches.

1.2.2 II. Movie effect model

To improve the first model we focus on the fact that, from experience, we know that 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:

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



By observing that the histogram is skewed we can imply that more movies have negative effects. This is called the penalty term movie effect.

We can improve our prediction by using this model.

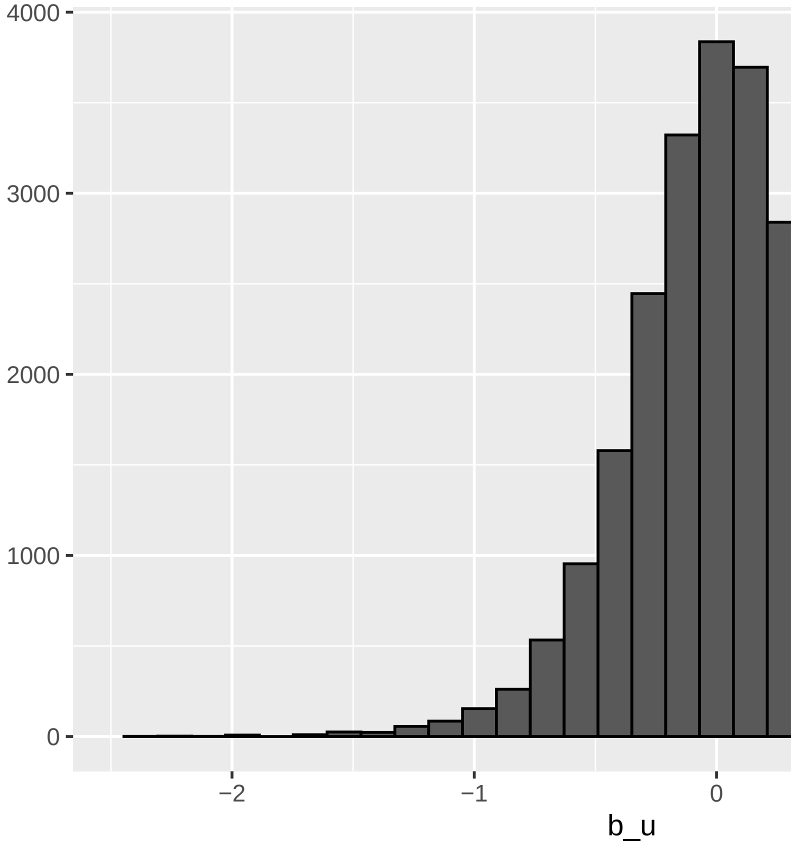
method	RMSE
Average movie rating model	1.0606506
Movie effect model	0.9437046

So we have predicted movie rating based on the fact that movies are rated differently by adding the computed b_i to μ. If an individual movie is on average rated worse that the average rating of all movies μ , we predict that it will rated lower that μ by b_i, the difference of the individual movie average from the total average.

This model represents an improvement but this model does not consider the individual user rating effect.

1.2.3 III. The movie and user effect model

By cuting the average rating for user μ, for those that have rated over 100 movies, said penalty term user effect. Users affect the ratings positively or negatively.



There is substantial variability across users as well as some users are very opinionated and others love every movie. We can further improve this model by

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

where b_u is a user-specific effect. If a cranky user (negative b_u rates a great movie (positive b_i), the effects counter each other and we may be able to correctly predict that this user gave this great movie a 3 rather than a 5.

We compute an approximation by computing μ and b_i, and estimating b_u, as the average of

$$Y_{u,i} - \mu - b_i$$

By constructing predictors we can see determine if our RMSE improves:

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

/pagebreak My rating predictions reduced the RMSE. Some of the best and worst movies happened to be rated only by a few users, in a lot of cases just one user. These movies happened to be mostly obscure ones.

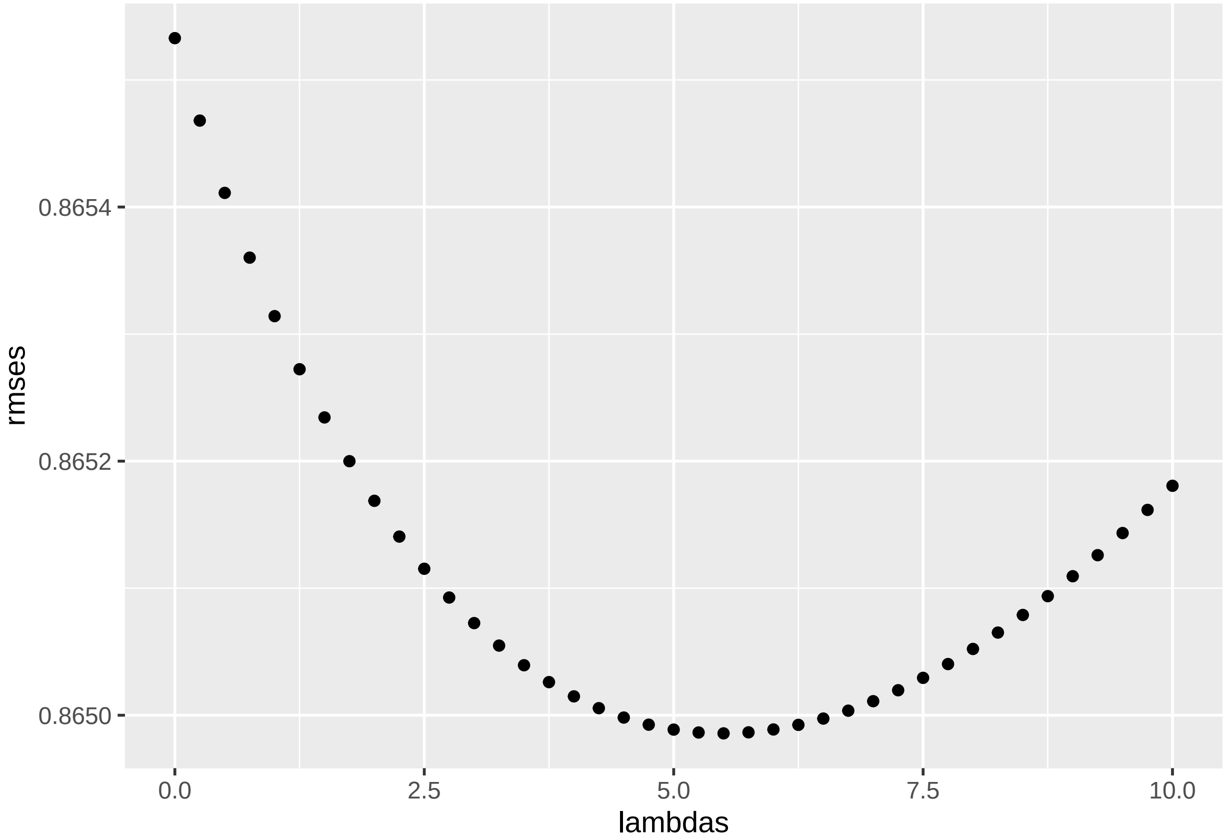
This is because by using only a few users, we create more uncertainty. Therefore larger estimates, negative or positive, are more likely. Large errors can increase our RMSE.

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

1.2.4 IV. Regularized movie and user effect model

So estimates of b_i and b_u are caused by movies with very few ratings and that some users only rated a small number of movies. This can strongly influence the prediction. The use of the regularization permits to penalize these aspects. By using a turning parameter such as lambda, we can find the value that will minimize the RMSE.

Plotting the RMSE vs lambdas to select the optimal lambda



The optimal lambda is:

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

The optimal lambda is: 5.25

The new results are:

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

2 Final Results

The RMSE values of all the represented models are the following:

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

We have found the lowest value of RMSE that is 0.8648170.

3 Conclusion

The regularized model including the effect of user is characterized by the lower RMSE value and is hence the optimal model to use for the present project. The optimal model characterised by the lowest RMSE value (0.8648170) lower than the initial evaluation criteria (0.8775) given by the goal of the present project. We could also affirm that improvements in the RMSE could be achieved by adding other effect (genre, year, age,etc).