Report on MovieLens Project -RMSE

Alma Bytyqi

January 29, 2019

Contents

1 Introduction

The goal of this project is to learn how to apply the knowledge base and skills learned throughout the series to real-world problems and how to independently work on a data analysis project. For this project, we will create a movie recommendation system using the MovieLens data set. The version of movielens included in the dslabs package is just a small subset of a much larger data set with millions of ratings.
We need to train a machine learning algorithm using the inputs in one subset to predict movie ratings in the validation set. Develop our algorithm using the edx set and finally test out prediction for movie ratings in the validation set as if they were unknown. RMSE will be used to evaluate how close our predictions are to the true values in the validation set.

2 Dataset description and Analysis

For this project, We used the 10M version of the MovieLens data set to make the computation a little easier. We downloaded the MovieLens data and ran code provided to generate our data sets.

First, we created edx set, validation set, and submission file with code already provided to us, using the tidyverse and caret packages. Those data sets where created in several steps:

1. downloaded file from a URL
2. using read.table, we read a file in table format and created a data frame from it named“ratings”, with cases corresponding to lines and variables to fields in the file.
3. with str_split, the strings were split up into pieces creating a character matrix with n columns and creating data set “movies” to which column names where added, converted into a data frame.
4. data sets movies and ratings where joined into one new data set named Movielens.
5. next step was to create the partitions with test and train sets where validation set is 10% of MovieLens data.
6. Add rows removed from validation set back into edx set
7. removed all extra files that will not be needed for the analysis thus freeing up system memory.

2.1 Data Structure

Using the Tidyverse Package, we can easily analyse the structure of the datasets. In order to better understand the challenge of this project, we need to see the general properties of the data. Data set edx, which is the test dataset has :

```
## [1] 9000055       6
```

lines and columns.

With following internal structure:

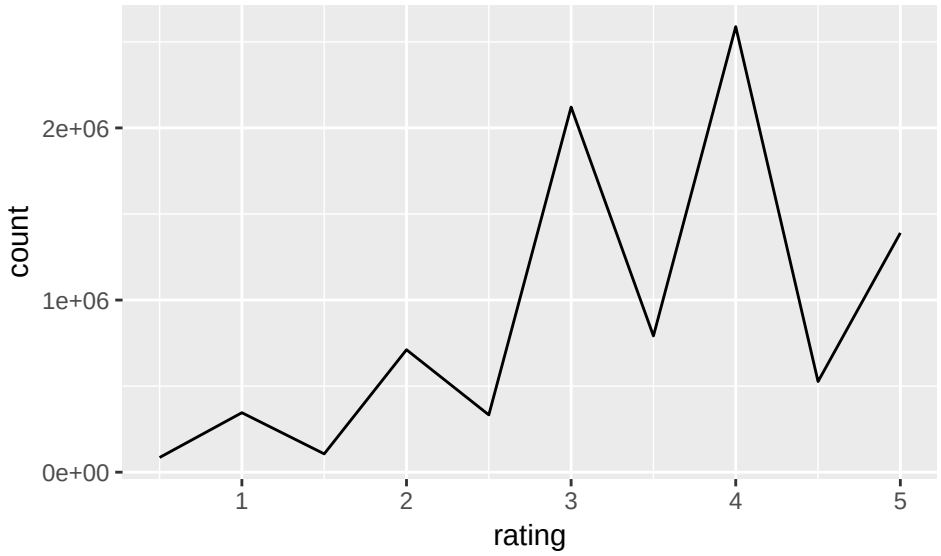
```
## 'data.frame':    9000055 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 292 316 329 355 356 362 364 370 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983421 838983392 838983392 838984474 838983653 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|A
```

The rating distribution table is the following:

```
## # A tibble: 10 x 2
##    rating   count
##     <dbl>   <int>
##  1    0.5   85374
##  2    1    345679
##  3    1.5  106426
```

```
##  4    2    711422
##  5    2.5  333010
##  6    3   2121240
##  7    3.5  791624
##  8    4   2588430
##  9    4.5  526736
## 10    5   1390114
```

The table tells us that ratings are not round numbers, but contain also half points such as 0.5 or 1.5.



From the plot, we see that most (62%) of the ratings are between 3 and 4.

The following table presents the total number of rated movies and users that rated all those movies:

analyse	total
Distinct Titles	10676
Distinct movie	10677
Distinct User	69878

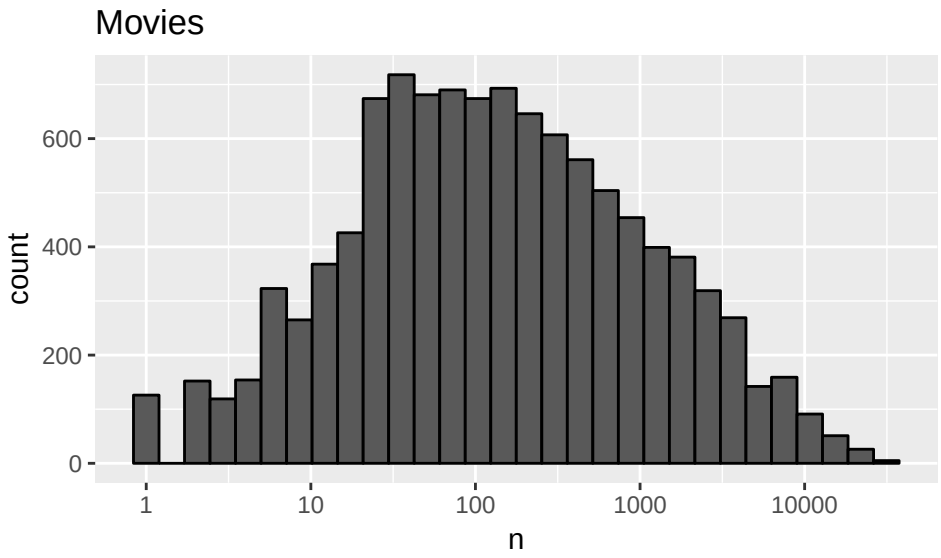
Below is the list of all different genres:

```
## # A tibble: 20 x 2
##    genres               count
##    <chr>                <int>
##  1 Drama              3910127
##  2 Comedy             3540930
##  3 Action             2560545
##  4 Thriller           2325899
##  5 Adventure          1908892
##  6 Romance            1712100
##  7 Sci-Fi             1341183
##  8 Crime              1327715
##  9 Fantasy             925637
## 10 Children            737994
## 11 Horror              691485
## 12 Mystery             568332
```

```
## 13 War                 511147
## 14 Animation           467168
## 15 Musical             433080
## 16 Western             189394
## 17 Film-Noir           118541
## 18 Documentary          93066
## 19 IMAX                  8181
## 20 (no genres listed)       7
```

We see that there are 20 different genres.

Next, we should check if some movies get rated more than others. Here is the distribution:



So, indeed there are movies that are rated less then 10 times and those that are rated more than 10000 times.

Whereas below is the list of 10 top movies with most ratings and 10 top movies with least ratings:

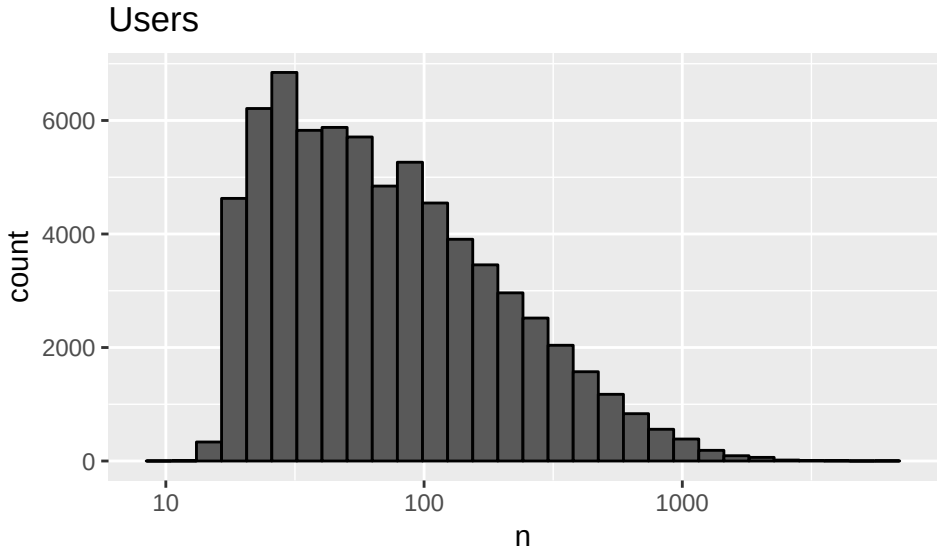
```
## # A tibble: 10,677 x 3
## # Groups:   movieId [10,677]
##    movieId title                                                     count
##      <dbl> <chr>                                                     <int>
##  1     296 Pulp Fiction (1994)                                       31362
##  2     356 Forrest Gump (1994)                                       31079
##  3     593 Silence of the Lambs, The (1991)                          30382
##  4     480 Jurassic Park (1993)                                      29360
##  5     318 Shawshank Redemption, The (1994)                          28015
##  6     110 Braveheart (1995)                                         26212
##  7     457 Fugitive, The (1993)                                      25998
##  8     589 Terminator 2: Judgment Day (1991)                         25984
##  9     260 Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (19~ 25672
## 10     150 Apollo 13 (1995)                                          24284
## # ... with 10,667 more rows
```

```
## # A tibble: 10,677 x 3
## # Groups:   movieId [10,677]
##    movieId title                                   count
##      <dbl> <chr>                                   <int>
##  1    3191 Quarry, The (1998)                          1
##  2    3226 Hellhounds on My Trail (1999)               1
```

```
##  3    3234 Train Ride to Hollywood (1978)                              1
##  4    3356 Condo Painting (2000)                                       1
##  5    3383 Big Fella (1937)                                            1
##  6    3561 Stacy's Knights (1982)                                      1
##  7    3583 Black Tights (1-2-3-4 ou Les Collants noirs) (1960)          1
##  8    4071 Dog Run (1996)                                              1
##  9    4075 Monkey's Tale, A (Les ChÃ¢teau des singes) (1999)            1
## 10    4820 Won't Anybody Listen? (2000)                                1
## # ... with 10,667 more rows
```

As the table show, there are obscure movies only once rated. All this means that while predicting the ratings, we should be very careful in the cases when the results are offset.

Our next observation is that some users are more active than others at rating movies:



3 Creating the recommendation system

First we need to create the Loss function, the residual mean squared error (RMSE) on a test set. the interpretation of which is if this number is larger than 1, it means our typical error is larger than one star, which is not good.

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

3.1 the simple model

The first model is to build the simplest possible recommendation system: same rating for all movies regardless of user. and we will call the simple model RMSE, the naive RMSE. Where mu_hat is equal to:

```
## [1] 3.512465
```

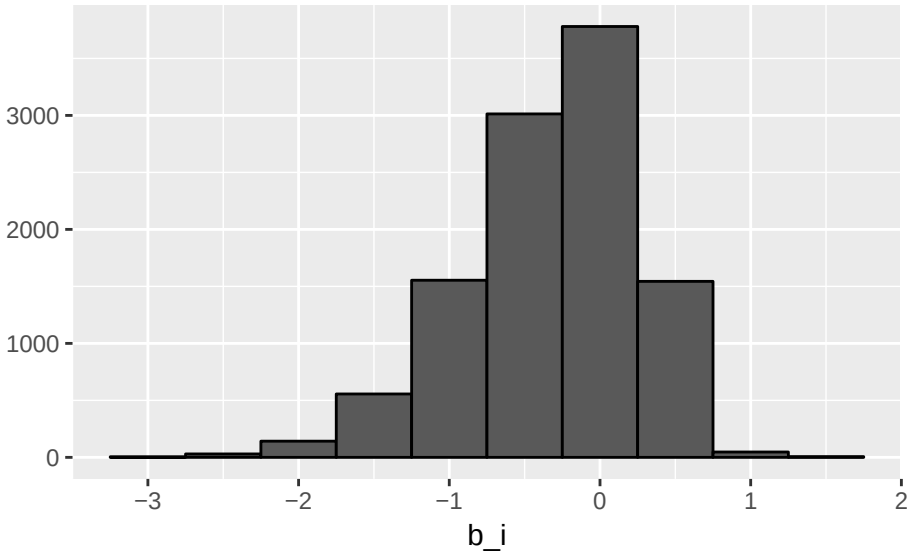
Thus, value of naive RMSE is equal to:

```
## [1] 1.061202
```

method	RMSE

method	RMSE
Just the average	1.061202

This shows us that RMSE > 1 meaning that the prediction will have low accuracy. Next step is to introduce the movie effect model for predicting the ratings:



method	RMSE
Just the average	1.0612018
Movie Effect Model	0.9439087

The table shows a slight improvement of the prediction using the movie effect, as RMSE is equal to 0.9439.

Now, we should add also the user effect into the model as usually different users will rate differently:

method	RMSE
Just the average	1.0612018
Movie Effect Model	0.9439087
Movie + User Effects Model	0.8292477

In this case the prediction have highly improved since the RMSE is equal to 0.8292.

However, the analysis does not stop here since we can add also the genre effect to the model as this factor impacts also the rating values:

method	RMSE
Just the average	1.0612018
Movie Effect Model	0.9439087
Movie + User Effects Model	0.8292477
Movie + User +genres Effects Model	0.8285157

method	RMSE

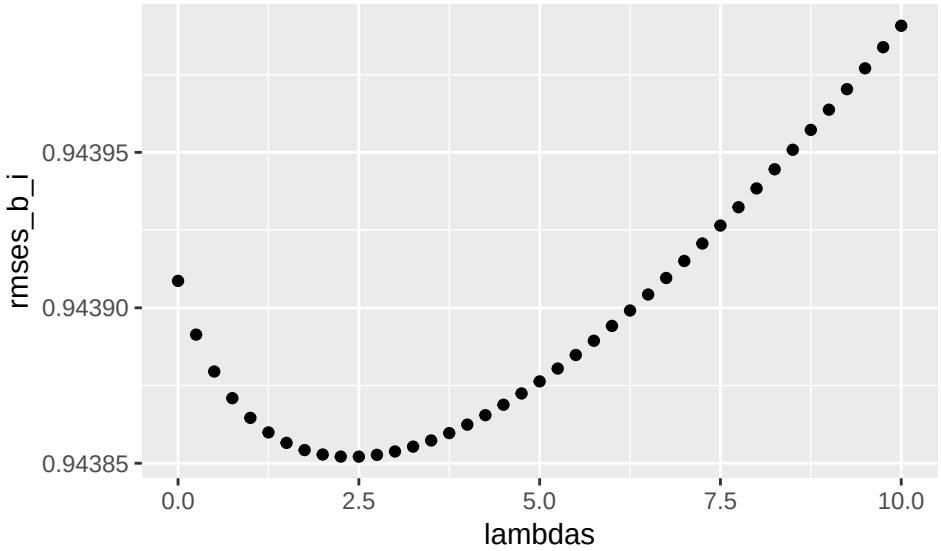
Here, we notice that result are slightly improved with RMSE equaling to 0.8285.

3.2 Regularization

Because data set analysis showed us that some movies are rarely rated and some users rarely rate, we should add a regularization effect to the prediction. This is done by introducing the Penalized Least squares with Lambda a penalty factor. First, we create a sequence of Lambdas which will be applied to a new function for determining the best lambda fit:

```
##  [1]  0.00  0.25  0.50  0.75  1.00  1.25  1.50  1.75  2.00  2.25  2.50
## [12]  2.75  3.00  3.25  3.50  3.75  4.00  4.25  4.50  4.75  5.00  5.25
## [23]  5.50  5.75  6.00  6.25  6.50  6.75  7.00  7.25  7.50  7.75  8.00
## [34]  8.25  8.50  8.75  9.00  9.25  9.50  9.75 10.00
```

Next, we create the function and plot the evaluation of the lambdas against RMSE with bi (movie effect model):



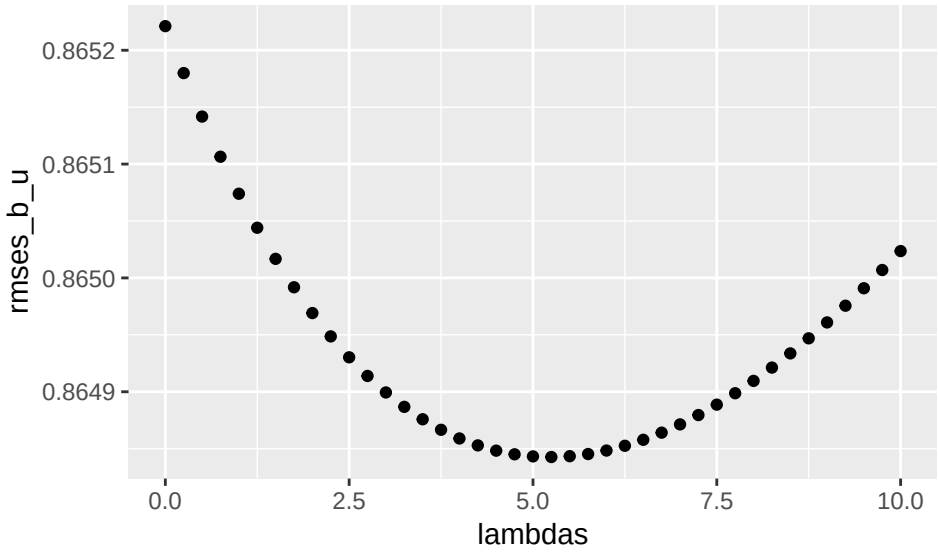
Thus optimal lambda with movie effect model is:

```
## [1] 2.5
```

and we get RMSE result of 0.9438:

method	RMSE
Just the average	1.0612018
Movie Effect Model	0.9439087
Movie + User Effects Model	0.8292477
Movie + User +genres Effects Model	0.8285157
Regularized Movie Effect Model	0.9438521

Next step is to add the user effect regularized model:



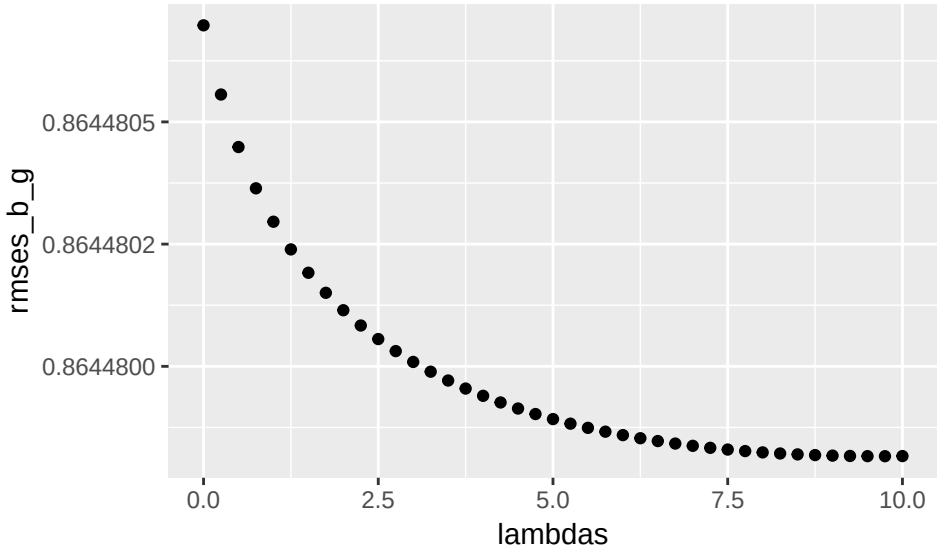
The plot shows us that the optimal lambda for user effect model is:

```
## [1] 5.25
```

And RMSE result are improved as shown on table below where RMSE equals 0.8648:

method	RMSE
Just the average	1.0612018
Movie Effect Model	0.9439087
Movie + User Effects Model	0.8292477
Movie + User +genres Effects Model	0.8285157
Regularized Movie Effect Model	0.9438521
Regularized Movie + User Effect Model	0.8648427

Last effect to be introduced into the model is the genre effect:



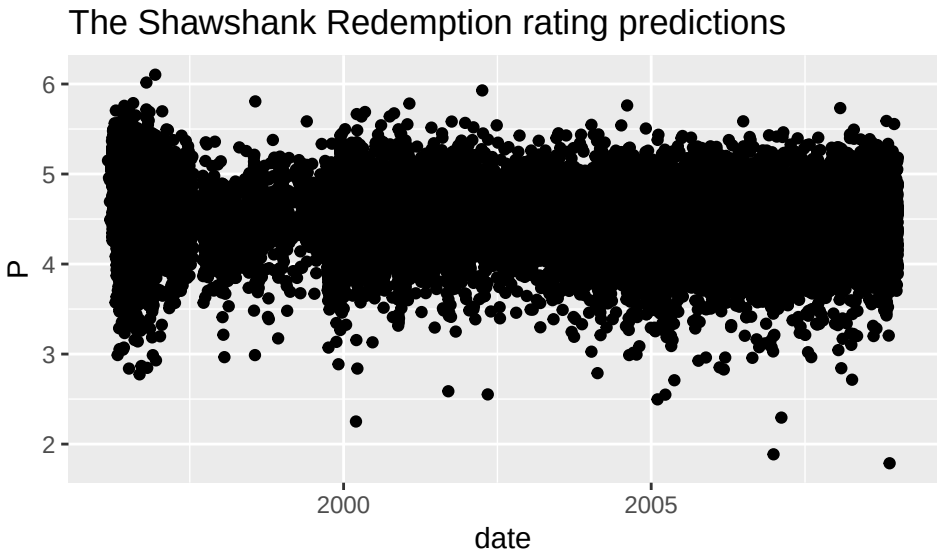
The plot shows us that optimal lambda for genre effect model is:

```
## [1] 9.75
```

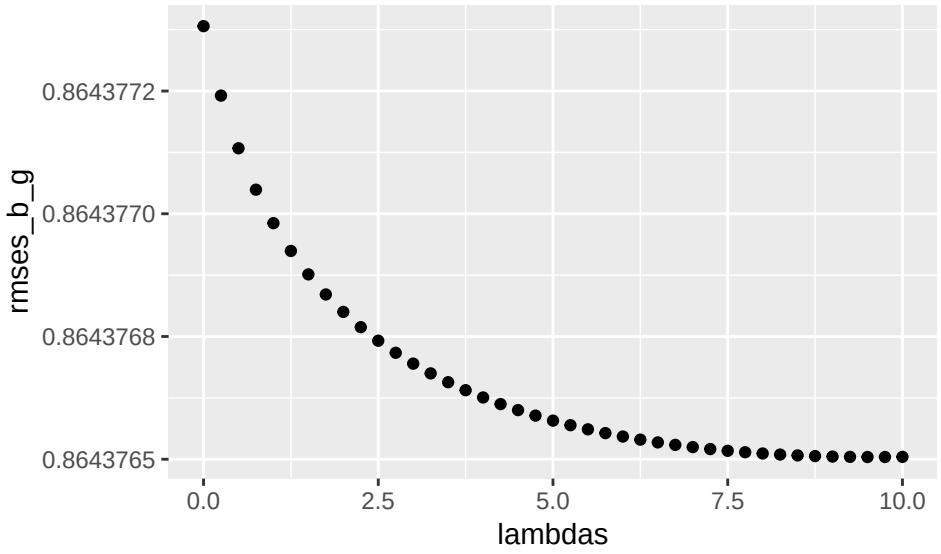
Whilst we see that the RMSE for all 3 effects models improves the prediction with RMSE =0.8645 as shown on the table below.

method	RMSE
Just the average	1.0612018
Movie Effect Model	0.9439087
Movie + User Effects Model	0.8292477
Movie + User +genres Effects Model	0.8285157
Regularized Movie Effect Model	0.9438521
Regularized Movie + User Effect Model	0.8648427
Regularized Movie + User + Genres Effect Model	0.8644798

Performing the verification of the prediction model, we can see that if extracting the prediction for "The Shawshank Redemption" movie, we see that many predicted ratings are higher then 5.



Thus, I have looked into improving the RMSE and the prediction model by adding a cap to the predicted values where instead of using the genre effect model, we introduce the capped Genre effect model where we limit the predicted rating to max of 5 since the ratings go from 0 to 5.



where the value for lambda with capped genre effect model is:

```
## [1] 9.5
```

Thus, the RMSE is slightly improved to 0.8644, as shown on table below:

method	RMSE
Just the average	1.0612018
Movie Effect Model	0.9439087
Movie + User Effects Model	0.8292477
Movie + User +genres Effects Model	0.8285157
Regularized Movie Effect Model	0.9438521
Regularized Movie + User Effect Model	0.8648427
Regularized Movie + User + Genres Effect Model	0.8644798
Regularized Movie + User + Genres Effect Model capped	0.8643765

4 Results and Prediction table

Now, that we have the final model with RMSE of less then 0.865, we will continue to produce the whole prediction table, However, since it will be impossible to show the whole table, here is the predicted ratings for each movies rated by user with userId=74:

```
##    userId                                    title rating
## 1      74                     Nine Months (1995)    4.0
## 2      74          Miracle on 34th Street (1994)    4.0
## 3      74                 Specialist, The (1994)    4.0
## 4      74                Muriel's Wedding (1994)    4.0
## 5      74 Naked Gun 33 1/3: The Final Insult (1994)    4.0
## 6      74               Courage Under Fire (1996)    4.0
## 7      74                   Godfather, The (1972)    3.5
## 8      74                   Basic Instinct (1992)    3.5
## 9      74                           Grease (1978)    4.0
## 10     74                      Liar Liar (1997)    3.0
## 11     74               Conspiracy Theory (1997)    4.0
## 12     74                           Game, The (1997)    2.5
```

```
## 13     74                Wedding Singer, The (1998)    3.5
## 14     74                    Lethal Weapon 2 (1989)    3.5
## 15     74           Honey, I Shrunk the Kids (1989)    2.5
## 16     74                Romancing the Stone (1984)    3.0
## 17     74                             Cocoon (1985)    3.5
## 18     74                   Sixth Sense, The (1999)    3.5
## 19     74                   Charlie's Angels (2000)    4.5
## 20     74              M*A*S*H (a.k.a. MASH) (1970)    4.0
##    predicted_ratings
## 1           3.062440
## 2           3.688086
## 3           3.068516
## 4           3.724532
## 5           3.148447
## 6           3.773774
## 7           4.622508
## 8           3.554515
## 9           3.509981
## 10          3.403046
## 11          3.513034
## 12          4.007510
## 13          3.599226
## 14          3.503074
## 15          2.871259
## 16          3.666836
## 17          3.488270
## 18          4.301151
## 19          3.023699
## 20          4.124697
```

And the aggregated ratings by movie for top 20 movies :

```
## # A tibble: 20 x 4
##    title                                     n avg_ratings pred_ratings
##    <fct>                                 <int>       <dbl>        <dbl>
##  1 Pulp Fiction (1994)                   34864        4.16         4.18
##  2 Forrest Gump (1994)                   34457        4.01         4.03
##  3 Silence of the Lambs, The (1991)      33668        4.20         4.21
##  4 Jurassic Park (1993)                  32631        3.66         3.69
##  5 Shawshank Redemption, The (1994)      31126        4.46         4.49
##  6 Braveheart (1995)                     29154        4.08         4.12
##  7 Fugitive, The (1993)                  28951        4.01         4.04
##  8 Terminator 2: Judgment Day (1991)     28948        3.93         3.94
##  9 Star Wars: Episode IV - A New Hope (a.k.~ 28566        4.22         4.23
## 10 Apollo 13 (1995)                      27035        3.89         3.92
## 11 Batman (1989)                         26996        3.39         3.39
## 12 Toy Story (1995)                      26449        3.93         3.95
## 13 Independence Day (a.k.a. ID4) (1996)  26042        3.38         3.38
## 14 Dances with Wolves (1990)             25912        3.74         3.74
## 15 Schindler's List (1993)               25777        4.36         4.40
## 16 True Lies (1994)                      25381        3.50         3.51
## 17 Star Wars: Episode VI - Return of the Je~ 25098        4.00         4.02
## 18 12 Monkeys (Twelve Monkeys) (1995)    24397        3.88         3.89
## 19 Usual Suspects, The (1995)            24037        4.37         4.37
## 20 Fargo (1996)                          23794        4.13         4.14
```

5 Conclusion

The goal of this project was to learn how to apply the knowledge base and skills learned throughout the series to real-world problems and how to independently work on a data analysis project. For this project, we needed to create a movie recommendation system using the MovieLens data set, train a machine learning algorithm using the inputs in one subset to predict movie ratings in the validation set. Develop our algorithm using the edx set and predict movie ratings in the validation set as if they were unknown. RMSE was used to evaluate how close our predictions in the validation set. For the analysis and creating the Recommendation system, I started to perform the naive process where we assumed that all ratings are the for each movie, which provided a RMSE>1, hence are results would have been very inaccurate with this first initial algorithm. I continued by adding the movie effect to the model and the user effect. however an additional factor could be added which is the genre effect. I ended with a recommendation system using all three effects : movie, user and genres rendering a good RMSE. Despite the good results with those 3 effect the solution was still naive as it would have been offset in cases of movies rated rarely or user rating rarely. therefore, I added to the algorithm also the regularization effect by using the Penalty least square function and finding best fitted lambdas where lambdas where the penalty factors, but also I ended by capping the final predicted values since some would predict ratings of higher then 5 which was offsetting in our case. Hence, I ended with a Recommendation system with a final RMSE = 0.8644 after applying the effects of movies, users and genres capping the prediction to a maximum value of 5 as the ratings are all from 0 to 5.

Note: the algorithms and coding are all in the attachment of this project with extension R and Rmd files.