MovieLense Project

John Estrada

1/2/2020

TABLE OF CONTENT

INTRODUCTION

The main objective of this project is to develop an algorithm using the "edx" dataset to predict movie ratings in the "validation" set as if they were unknown. The RMSE will be used as the measuring criteria to determine how close are the results obtained from the herein algorithm to the actual validation dataset. In order to avoid training the set using the validation set, an additional partition (train_set, test_set) has been created for training purposes. Different models are evaluated on the later partition in order to determine which one minimizes the RMSE. Then, that trained model is implemented to retrain the "edx" dataset and evaluate the final RMSE against the "validation" data set.

In this report you will navigate through the different steps taken in consideration for data analyisis. This steps include Data cleaning, Data exploration and visualization, Analyzis from the data and Models Aproach. The results and analysis are presented. Finally, some conslusions and future work are listed.

NOTE TO THE GRADER: The code to elaborate this report is hidden. Only the code for determining the best fit model from the train_set and test_set, and the application of that model to the edx and validation data is displaed on this report. If you decide to take a look at the code, please refer to the .Rmd file or .R code. Thank you for your comments and feedback.

METHODS

The methodology followed to minimize the RMSE is described as follows.

Data Exploration and Analysis

Data exploration and visualization: The first step of the data exploration is to determine the dimensions of the datasets. The edx (training) contains 9000055 rows and 6 columns, the validation (test) contains 999999 rows and 6 columns. This confirms that the two sets have been roughly partitioned in a 9/1 ratio. The potential predictors are userID, movieID associated with the movie title, the timestamp and genre.

```r
head(edx)
```

```
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 3      1     231      5 838983392          Dumb & Dumber (1994)
## 4      1     292      5 838983421               Outbreak (1995)
## 5      1     316      5 838983392               Stargate (1994)
## 6      1     329      5 838983392 Star Trek: Generations (1994)
##                           genres
## 1                 Comedy|Romance
## 2          Action|Crime|Thriller
## 3                         Comedy
## 4  Action|Drama|Sci-Fi|Thriller
## 5        Action|Adventure|Sci-Fi
## 6 Action|Adventure|Drama|Sci-Fi
```

```r
dim(validation)
```

```
## [1] 999993      6
```

```r
dim(edx)
```

```
## [1] 9000061       6
```

Defining RMSE

Root Mean Square Error (RMSE) is defined as the standard deviation of the residuals (prediction errors). In our case the Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. For this project (and for other Data Analyis Projects), the RMSE has been defined to quantify the models herein presented. The smaller the RMSE the better the fit. The RMSE can be thought as the typical error made at estimating the rating of a movie (i) based on the user (u), with N number of user/movie combinations.

Fit Models on train_set and test_set

In order to determine the model that minimizes the RMSE, a subset of train_set and test_set from the "edx" dataset has been created. The models are evaluated using the train and test data sets. Once the best model is encountered, it will be retrain on the edx data and finally tested on the "validation" data set to estimate the final RMSE for the project.

Naive Model

A first approach is to evaluate the simplest model that considers the estimated movie rating to the average movie rating independent of movie or user with all the differences explain by the random variability (E). This model can be represented as follows:

Yu,i = mu + Eu,i

Where (Y) represents the expected rating of the movie (i) from user (u), (mu) as the average movie rating and (E) the random variability of the ratings.

```
## [1] 3.51238
```

```
## [1] 1.059643
```

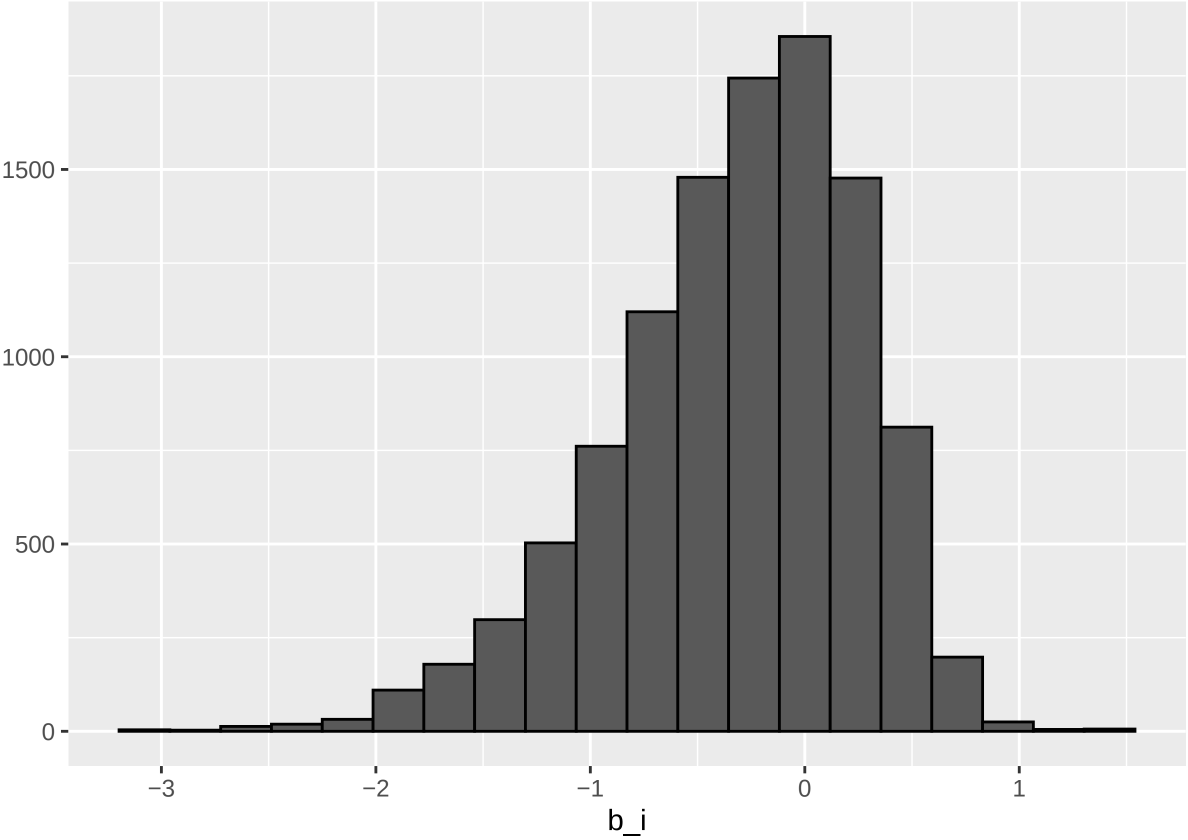
```
## # A tibble: 1 x 2
##   method                              RMSE
##   <chr>                           <dbl>
## 1 Just the rating average (Naive)  1.06
```

Thus, the movie average (mu) is 3.51238 and the estimated RMSE for this simple model is 1.06.

Movie Bias Effect Model

The previous model in escence fails to include the movie bias effect. Not all movies are good, and not all are bad. Therefore, some movies may have higher rating than others. We can add to the previous model the movie bias effect (b) that stands for the average rating of the movie (i) regardless of the user.

As we can observe from the following plot, whereas most of the movie ratings are concentrated towards the average movie rating centered to zero, there are other movies that substancially deviated from the average. This deviation motivates the inclusion of a movie effect bias parameter to the model.



This model can be represented as follows:

Yu,i = mu + bi + Eu,i

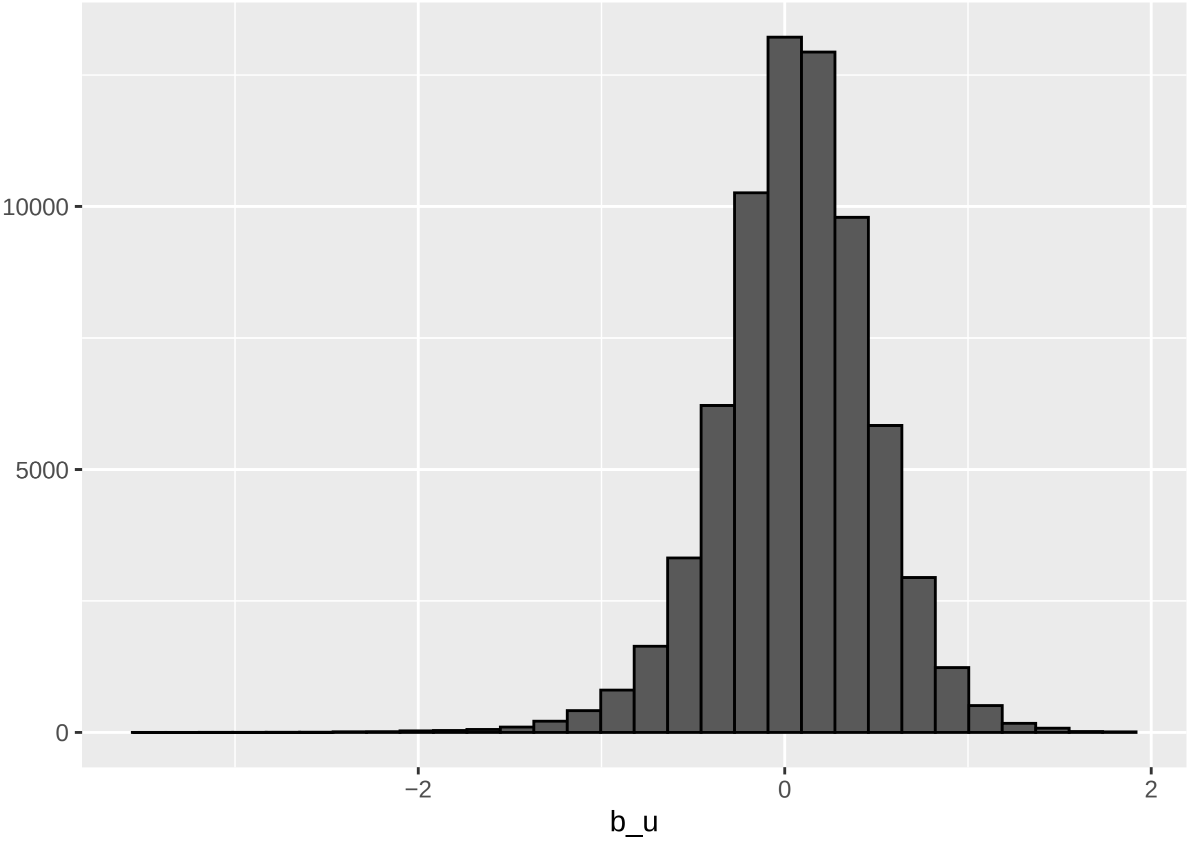
To develop the code, the least square estimate bi is determined as the average of Yu,i - mu for each movie i.

```
## # A tibble: 2 x 2
##   method                           RMSE
##   <chr>                           <dbl>
## 1 Just the rating average (Naive) 1.06
## 2 Movie Bias Effect Model         0.943
```

The inclusion of the movie bias effect improved the model considerably to 0.943. However, we can check if we can improve the model even more.

User Specific Effect

Not all users provide the same rating to a movie they like. For example, a kindly user can rate a bad movie with a 3 instead of a 1. In that sense, a user specific effect adjusts that effect to corrently predict the rating that this user gives to a bad, normal, or great movie. This effect can be represented in the following chart.



In that sense, a model that includes both the movie effect and user effect can be establish as follows:

Yu,i = mu + bi + bu Eu,i

To develop the code, the least square estimate (bu) is determined as the average of Yu,i - mu - bi for each movie (i) and user (u).

```
## [1] 0.8655154
```

```
## # A tibble: 3 x 2
##   method                          RMSE
##   <chr>                          <dbl>
## 1 Just the rating average (Naive) 1.06
## 2 Movie Bias Effect Model         0.943
## 3 Movie + User Effects Model      0.866
```

Notably, the new model yielded a very good RMSE of 0.8655. However, we can see whether the model can be improved with regularization.

Regularized Movie Effect

A common mistake is to give a very high or very low estimate (bi) to a movie that has not been rated several times since it will affect the overall prediction. If this is the case, we want to penalized those estimates of (bi) with low amount of ratings. We can confirm is we regularization can have a good impact.

First, we can evaluate the best and worst estimates of (bi) and check how many times that movie was rated.

The best estimates appear as follow:

```
## Joining, by = "movieId"
```

```
## # A tibble: 10 x 3
##    title                                                               b_i     n
##    <chr>                                                             <dbl> <int>
##  1 Shanghai Express (1932)                                            1.49     1
##  2 Satan's Tango (Sátántangó) (1994)                                  1.49     1
##  3 Fighting Elegy (Kenka erejii) (1966)                               1.49     1
##  4 Sun Alley (Sonnenallee) (1999)                                     1.49     1
##  5 Constantine's Sword (2007)                                         1.49     1
##  6 Human Condition II, The (Ningen no joken II) (1959)                1.32     3
##  7 Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to t...  1.24     4
##  8 Life of Oharu, The (Saikaku ichidai onna) (1952)                   1.24     2
##  9 I'm Starting From Three (Ricomincio da Tre) (1981)                 1.15     3
## 10 Human Condition III, The (Ningen no joken III) (1961)              1.15     3
```

The best estimates appear to be movies that are not well known. Additionally, the number of times that these movies were rated are very low.

Now, the worst estimates:

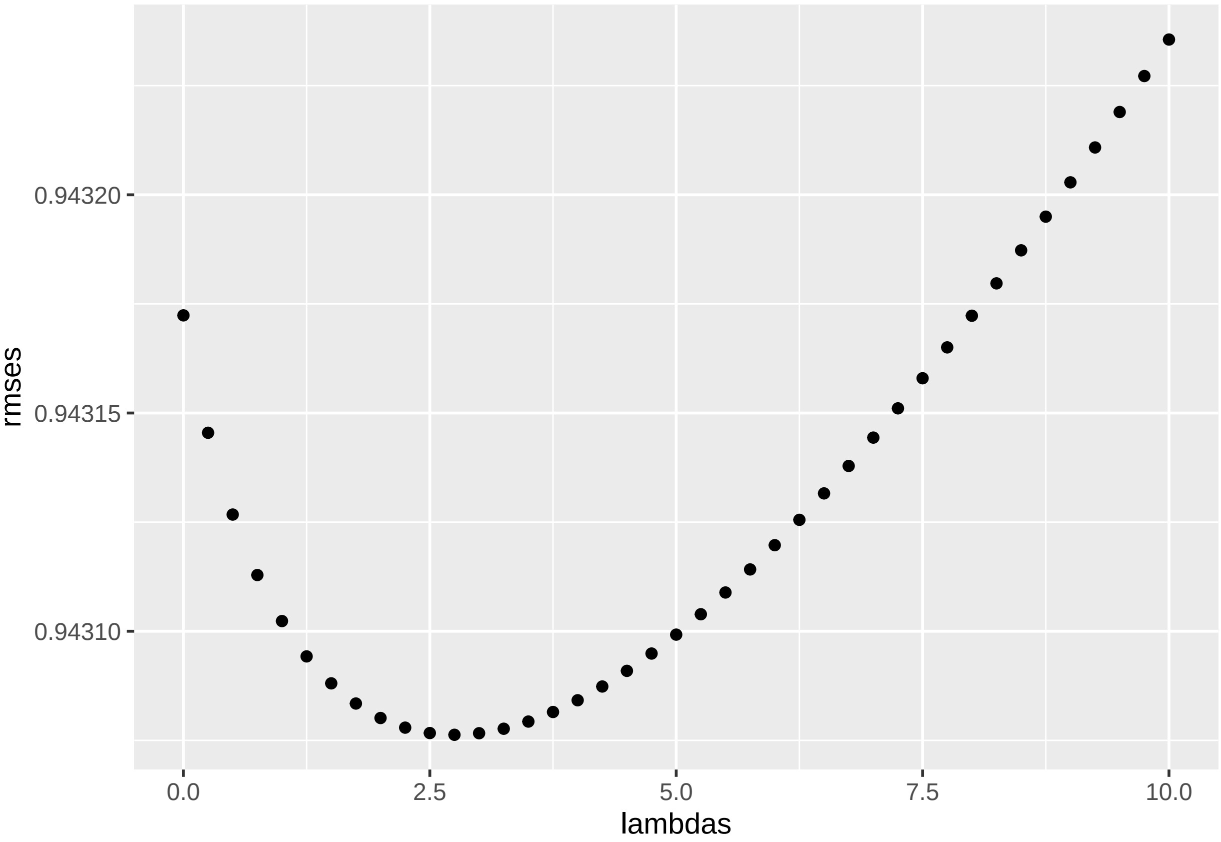
```
## Joining, by = "movieId"
```

```
## # A tibble: 10 x 3
##    title                                 b_i     n
##    <chr>                               <dbl> <int>
##  1 Besotted (2001)                     -3.01     2
##  2 Hi-Line, The (1999)                 -3.01     1
##  3 Uncle Nino (2003)                   -3.01     1
##  4 Accused (Anklaget) (2005)           -3.01     1
##  5 Hip Hop Witch, Da (2000)            -2.95     8
##  6 Karla (2006)                        -2.85     3
##  7 SuperBabies: Baby Geniuses 2 (2004) -2.74    48
##  8 From Justin to Kelly (2003)         -2.61   161
##  9 Pokémon Heroes (2003)               -2.59   105
## 10 Criminals (1996)                    -2.51     2
```

Here, however, there are movies that were poorly rated by several users. Therefore, the effect of regularization may not have a big impact here. However, we can confirm this by regularizing the variability of the size effects. A penalty factor (called lambda) is introduced to the model. As the sample size increases, the penalty effect decreases.

The lambda is a tuning parameter and we can use cross-validation to estimate the lamda that minimizes the RMSE for the model.

The following plot represents the behavior of the RMSE as the lamda changes.



```
## [1] 2.75
```

From the plot, the lambda that minimizes the RMSE is 2.75. The RMSE value of the model with lambda is included as follows:

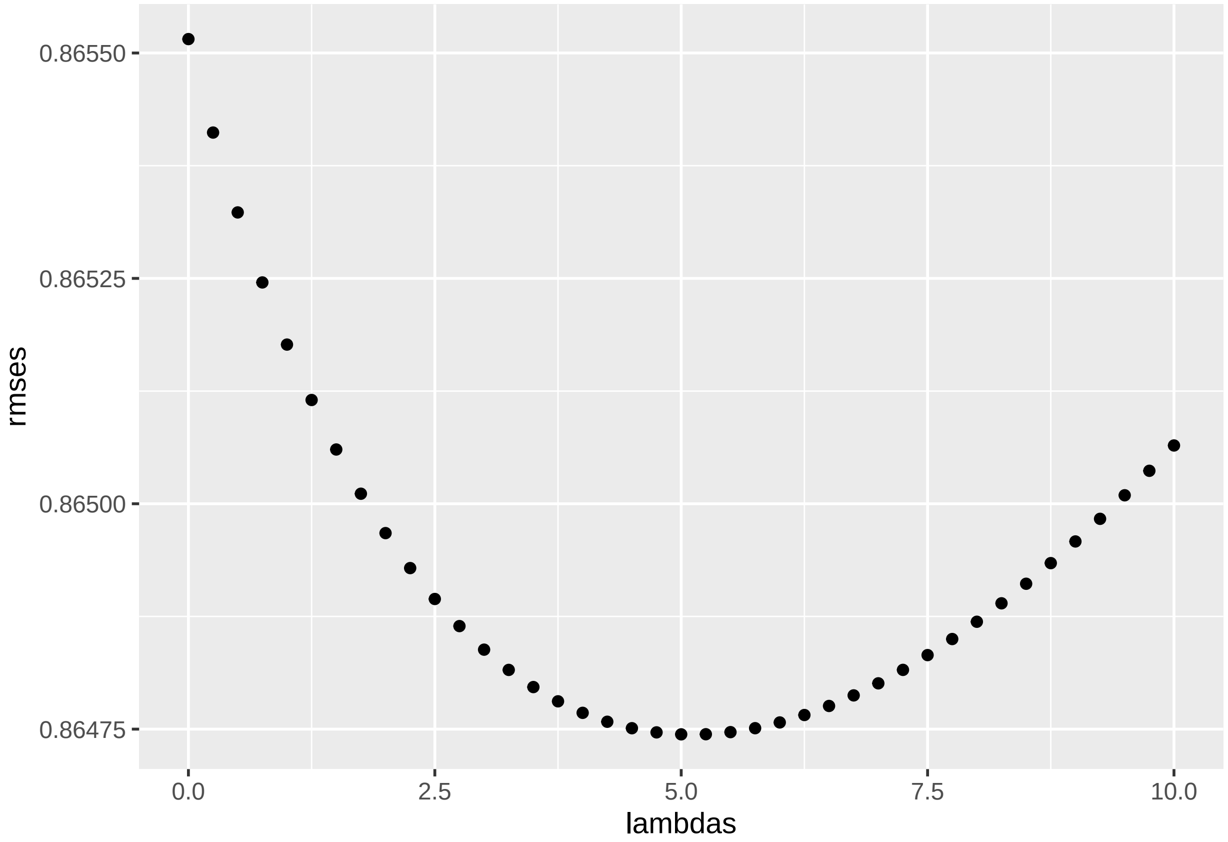
```
## # A tibble: 4 x 2
##   method                             RMSE
##   <chr>                             <dbl>
## 1 Just the rating average (Naive) 1.06
## 2 Movie Bias Effect Model         0.943
## 3 Movie + User Effects Model      0.866
## 4 Regularized Movie Effect Model  0.943
```

As expected, the RMSE did not decrease considerably with regularization as some poorly rated movies had a sample size of user highly enough.

Regularized Movie and User Effect

Using a similar approach for regularization, the movie and user effect can be introduced. The lambda that minimizes the RMSE and the RMSE are calculated as follows:

```
####REGULARIZED MOVIE AND USER EFFECTS
lambdas <- seq(0, 10, 0.25)
rmses <- sapply(lambdas, function(l){
  mu <- mean(train_set$rating)
  b_i <- train_set %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu)/(n()+l))
  b_u <- train_set %>%
    left_join(b_i, by="movieId") %>%
    group_by(userId) %>%
    summarize(b_u = sum(rating - b_i - mu)/(n()+l))
  predicted_ratings <-
    test_set %>%
    left_join(b_i, by = "movieId") %>%
    left_join(b_u, by = "userId") %>%
    mutate(pred = mu + b_i + b_u) %>%
    pull(pred)
  return(RMSE(predicted_ratings, test_set$rating))
})
qplot(lambdas, rmses)
```



```
lambda <- lambdas[which.min(rmses)]
lambda
```

```
## [1] 5
```

```
rmse_results <- bind_rows(rmse_results,
                          tibble(method="Regularized Movie + User Effect Model",
                                 RMSE = min(rmses)))
rmse_results
```

```
## # A tibble: 5 x 2
##   method                                 RMSE
##   <chr>                                 <dbl>
## 1 Just the rating average (Naive)       1.06
## 2 Movie Bias Effect Model               0.943
## 3 Movie + User Effects Model            0.866
## 4 Regularized Movie Effect Model         0.943
## 5 Regularized Movie + User Effect Model 0.865
```

Thus, the model that minimized the RMSE is the regularized movie and user effect with a RMSE of 0.865. In consequence, we can apply this model to the edx and validation data.

RESULTS AND ANALYSIS

From the trained models the one that produces the lowest RMSE is the regularized model that considers the movie bias effect and user effect. The lambda that minimizes the RMSE was calculated and equal to 5.

With this trained and optimized model, we can implement it on the edx dataset and compare it to the validation dataset as follows:

```
################################
# FIT MODELS ON THE VALIDATION DATASET
################################

#APPLY REGULARIZED MOVIE AND USER EFFECT MODEL TO TRAIN THE EDX DATA
#AND TO TEST IT ON THE VALIDATE USING THE LAMBDA THAT MINIMIZED THE
#RMSE ON THE TRAINING DATA

l <- lambda
mu <- mean(edx$rating)
b_i <- edx %>%
  group_by(movieId) %>%
  summarize(b_i = sum(rating - mu)/(n()+l))
b_u <- edx %>%
  left_join(b_i, by="movieId") %>%
  group_by(userId) %>%
  summarize(b_u = sum(rating - b_i - mu)/(n()+l))
predicted_ratings <-
  validation %>%
  left_join(b_i, by = "movieId") %>%
  left_join(b_u, by = "userId") %>%
  mutate(pred = mu + b_i + b_u) %>%
  pull(pred)

RMSE <- RMSE(predicted_ratings, validation$rating)
RMSE
```

```
## [1] 0.8649887
```

The final model on the original dataset yields a RMSE of 0.8649887. The RMSE evaluated on the validation dataset is very similar to the one yielded with the training and test sets. I was expecting slightly higher RMSE since validation and edx dataset are larger. One of the reasons that I can think of is that train model did not have over-fitting, which I consider a positive aspect. Therefore, since it was properly regularized the model could estimate values from other datasets.

CONCLUSIONS AND FUTURE WORK

The objective of minimizing the error of predicting the rating that a particular user can give to a movie was accomplished using regularization to the model that included the user specific effect and movie bias. When tested on the validation set the RMSE was very similar to the one obtained from the training dataset. The interpretation is that the training model did not experience over-fitting because the model was regularized.

Regularization produced by the movie bias effect did not produced a considerable decrease of the RMSE since some of the movies that were poorly rated, were effectively poorly rated by a high considerable number of users. On the other hand, when regularization was applied to both the movie bias effect and user effect the

However, although the final model yields a RMSE of 0.8649887 some other effects could be included to the model such as the movie genre effect as some genres may be more popular than others. Also, a movie release year effect could be evaluated as some users may like older movies and dislike newer movies.