MovielensProject

REHAM ALSHEHRI

1/20/2022

Contents

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.6     v purrr   0.3.4
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## Warning: package 'tibble' was built under R version 4.0.5
## Warning: package ’tidyr’ was built under R version 4.0.5

## Warning: package ’readr’ was built under R version 4.0.5

## Warning: package ’purrr’ was built under R version 4.0.5

## Warning: package ’stringr’ was built under R version 4.0.5

## Warning: package ’forcats’ was built under R version 4.0.5

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::collapse()          masks nlme::collapse()
## x mosaic::count()            masks dplyr::count()
## x purrr::cross()             masks mosaic::cross()
## x mosaic::do()               masks dplyr::do()
## x tidyr::expand()            masks Matrix::expand()
## x dplyr::filter()            masks stats::filter()
## x ggstance::geom_errorbarh() masks ggplot2::geom_errorbarh()
## x dplyr::lag()               masks stats::lag()
## x tidyr::pack()              masks Matrix::pack()
## x mosaic::stat()             masks ggplot2::stat()
## x mosaic::tally()            masks dplyr::tally()
## x tidyr::unpack()            masks Matrix::unpack()

## Warning: package ’caret’ was built under R version 4.0.5

##
## Attaching package: ’caret’

## The following object is masked from ’package:purrr’:
##
##     lift

## The following object is masked from ’package:mosaic’:
##
##     dotPlot

## Warning: package ’data.table’ was built under R version 4.0.5

##
## Attaching package: ’data.table’

## The following object is masked from ’package:purrr’:
##
##     transpose

## The following objects are masked from ’package:dplyr’:
##
##     between, first, last

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

1 Introduction

In e-commerce or internet streaming sites like Netflix, Facebook, and eBay, recommendation algorithms are critical (Lu, Wu, Mao, Wang, & Zhang, 2015). Providing the appropriate suggestion for another product, song, or film boosts customer retention and happiness, which results in increased sales and profits. Businesses striving for customer satisfaction investment in platforms that collect and process users' preferences and then recommend items or services that are more likely to be purchased (Gomez-Uribe & Hunt, 2015). The economic consequence of this kind of business-customer connection is obvious: Amazon is the world's biggest online retailer by revenue, and a major advantages of their strategy is due to its recommender system and direct advertising based on the customer interests (Smith & Linden, 2017). Typically, recommender system use a grading range of one to 5 categories or stars ratings, with one representing the lowest level of satisfaction and five indicating the greatest degree of satisfaction. Additionally, other factors such as opinions expressed on previously used products; clip, songs, or URL sharing with mates; proportion of movies watched or songs started listening; web sites went to visit and hours invested on every website; product group; or any connection with the firm's web application might be utilized as predictors (Jain, Grover, Thakur, & Choudhary, 2015). The basic objective of recommender system is to assist users in locating desired items tastes and preferences and prior interactions, as well as to forecast the ratings of a specific product. Here, we develop a movie recommender systems in this analysis by utilizing the 'MovieLens' dataset and implementing the solution to get lowest error score for the model (grouplens, 2009).

2 Project Objectives

The main objective of this analysis is to build the model which have the lowest RMSE score from 0.86490 and compare the various models on the validation dataset.

2.1 Dataset

Movielens dataset is utilized here to conduct the analysis. Here, we only considered it 10M subset of the whole dataset to make a recommender system on the dataset (grouplens, 2009).

3 Methods

3.1 Data Exploration with Descriptive Analysis

Data exploration with descriptive analysis is very useful step which helps to understand the dataset dimensions, structure and relationship with each other. This step is performed to get the familiarization of the dataset and find the hidden insights from ii. For this purpose, we utilized the various types of the charts, tables, histograms and summary statistics on the dataset. Moreover, this phase also utilized to draw an attractive, catchy and meaningful visualization using the various types of the features in the dataset. There is total '6' variables and '9000055' objects in the dataset. There are some integer variables and some of them are character. The data type of for each variable is shown here.

```
## Classes 'data.table' and 'data.frame':   9000061 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 231 292 316 329 355 356 362 364 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983392 838983421 838983392 838983392 838984474 838983653 83
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Dumb & Dumber (1994)" "Outbreak (1995)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Comedy" "Action|Drama|Sci-Fi|Thriller"
##  - attr(*, ".internal.selfref")=<externalptr>
```

Moreover, the dim() command is utilized as well to find the dimension of the dataset and its found that there are 9000055 and 6 objects and columns in the EDX dataset respectively.

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

.The summary of the numerical variable in the dataset is also shown below that have their min, max, 1st, 2nd, 3rd quartiles and mean values using the summary() command.

```
##      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 top '5' and bottom '5' rows from the EDX dataset is shown below. There are total columns in the dataset and their names are presented as well. Moreover, here is the target variable is 'rating' and all other are features. The movies information with against each 'userid' is given for each row that can be scene easily.

```
##    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)
##                           genres
## 1:                Comedy|Romance
## 2:         Action|Crime|Thriller
## 3:                        Comedy
## 4: Action|Drama|Sci-Fi|Thriller
## 5:       Action|Adventure|Sci-Fi
```

```
##    userId movieId rating  timestamp                        title
## 1:  59269   59680    3.0 1229014701     One Hour with You (1932)
## 2:  59269   64325    3.0 1229014646       Long Night, The (1947)
## 3:  59342   61768    0.5 1230070861    Accused (Anklaget) (2005)
## 4:  60713    4820    2.0 1119156754 Won't Anybody Listen? (2000)
## 5:  68986   61950    3.5 1223376391             Boot Camp (2007)
##                                     genres
## 1:                 Comedy|Musical|Romance
## 2: Crime|Drama|Film-Noir|Romance|Thriller
## 3:                                   Drama
## 4:                             Documentary
## 5:                                Thriller
```

Now, the detailed description for the each column is the dataset is being described using the tables and charts. Their details is given below as a reference.

3.1.1 Genres Feature

In this dataset, there is column with the name of ‘genres’ and have the information about the various movies genres. The summary of the genres features is computed using the summary function with ‘groupby’ and top ‘10’ genres are displayed below in a table. The highest genre for action/adventure category that is ‘68688’. The numbers for other categories are also given below. Moreover, the unique genres length in the dataset is 797.

```
## [1] 797
```

```
## # A tibble: 10 x 2
##    genres                                                  n
##    <chr>                                               <int>
##  1 (no genres listed)                                      6
##  2 Action                                              24575
##  3 Action|Adventure                                    68611
##  4 Action|Adventure|Animation|Children|Comedy           7438
##  5 Action|Adventure|Animation|Children|Comedy|Fantasy    191
##  6 Action|Adventure|Animation|Children|Comedy|IMAX        62
##  7 Action|Adventure|Animation|Children|Comedy|Sci-Fi     600
##  8 Action|Adventure|Animation|Children|Fantasy           743
##  9 Action|Adventure|Animation|Children|Sci-Fi             51
## 10 Action|Adventure|Animation|Comedy|Drama              1896
```

3.1.2 Timestamp (Date) Feature

The date feature is calculated from the ‘timestamp’ feature that have almost 14 years of the time period. This period is mutated and calculated as well.

```
## Warning: package ’lubridate’ was built under R version 4.0.5
```

```
##
## Attaching package: ’lubridate’
```

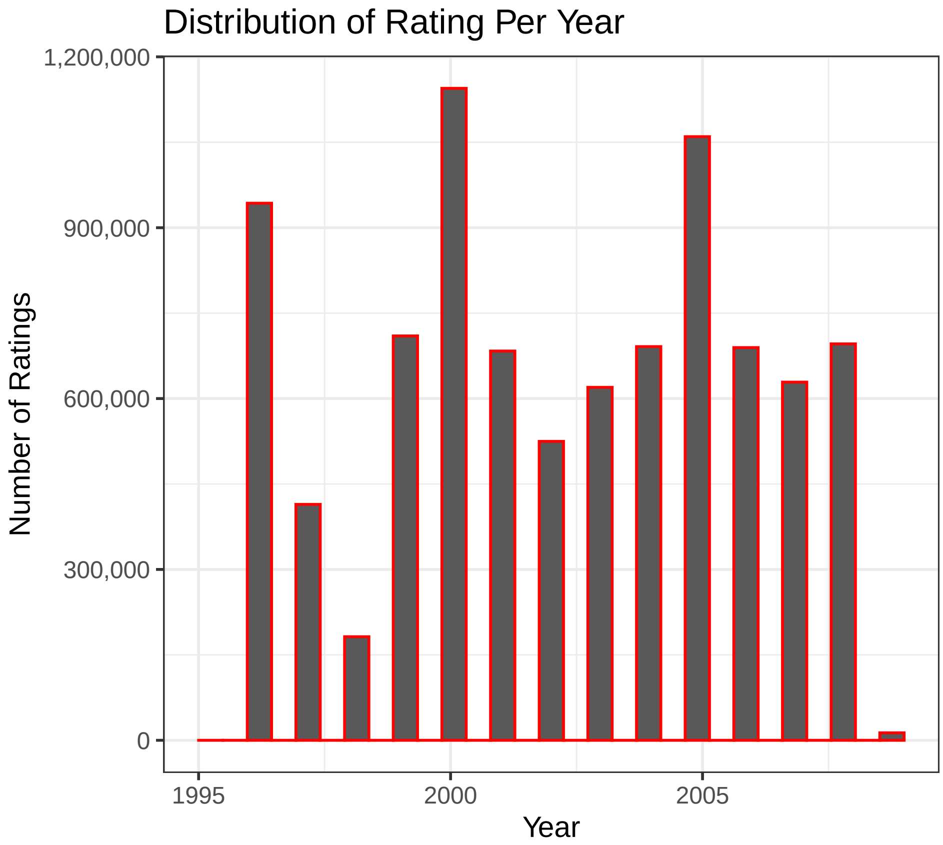
```
## The following objects are masked from ’package:data.table’:
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```
## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union
```

```
## # A tibble: 1 x 3
##   ‘Initial Date‘ ‘Final Date‘ Period
##   <date>         <date>       <Duration>
## 1 1995-01-09     2009-01-05   441479294s (~13.99 years)
```

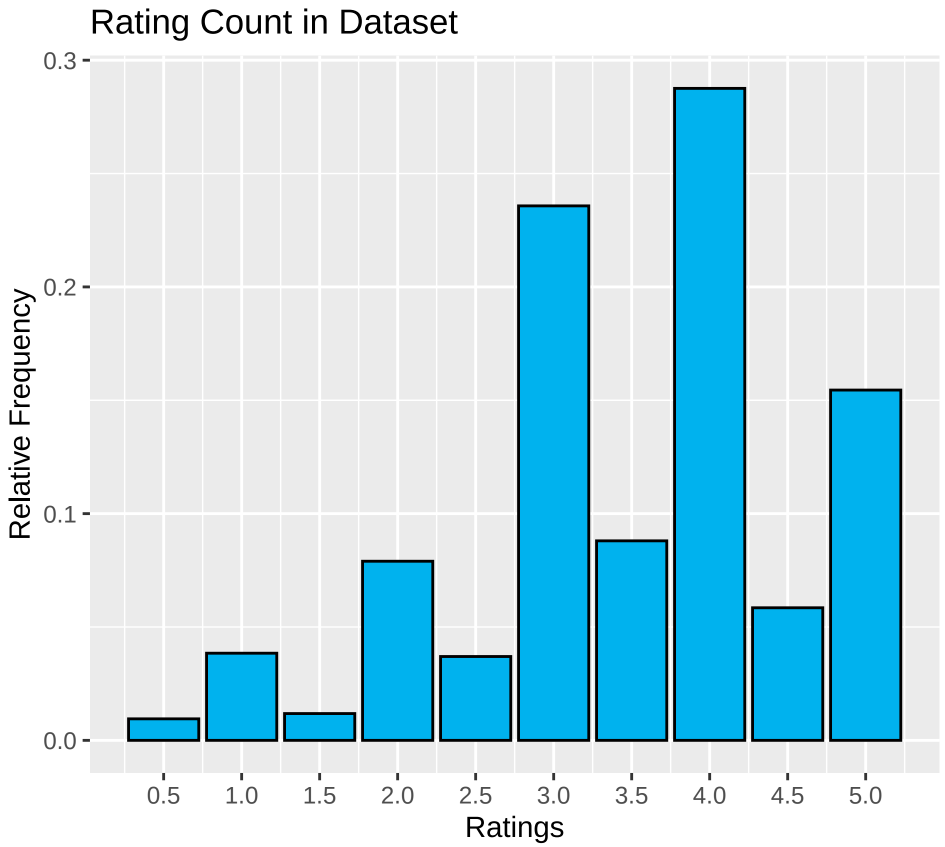
The ratings distribution for each year is also plotted calculating the years from ‘timestamp’ variable and outcome shows that the highest rating in 2000 year. On the other hand, the distribution for the other also given here and can be scene easily.

```
## Warning: package 'ggthemes' was built under R version 4.0.5

##
## Attaching package: 'ggthemes'

## The following object is masked from 'package:mosaic':
##
##     theme_map

## Warning: package 'scales' was built under R version 4.0.5

##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor

## The following object is masked from 'package:mosaic':
##
##     rescale

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



3.1.3 Ratings Target

The ratings of the movies are also visualized using the bar chart. The range of rating is also given below and it shows that the most movies rating are '4'. It means that the user have good experience while watching the movies in this data set. There are total '10' ratings and these are displayed below.

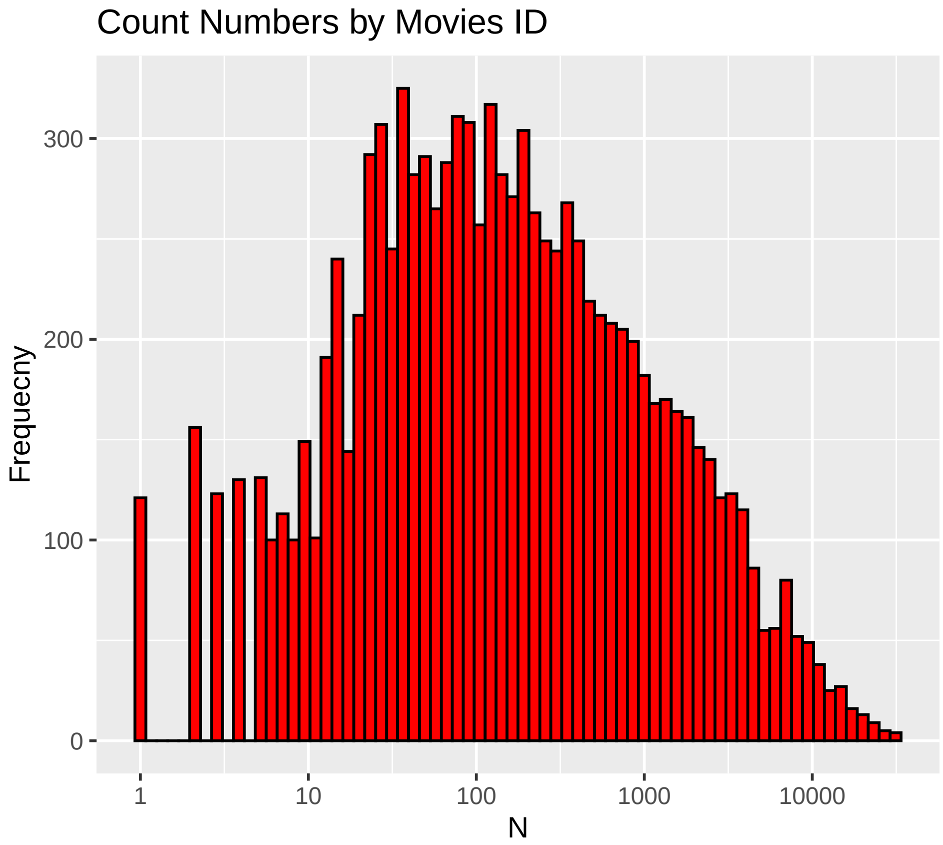


3.1.4 Movies Feature

The movies unique length in the EDX dataset is '10677' which means that these number are unique in dataset and their length is also calculated.

```
## [1] 10677
```

The count numbers of MoviesID is also determined and displayed their frequent against each number using the histogram. Outcomes shows that the most of the movies have the range from 50-500. This range frequency is higher than to other one.

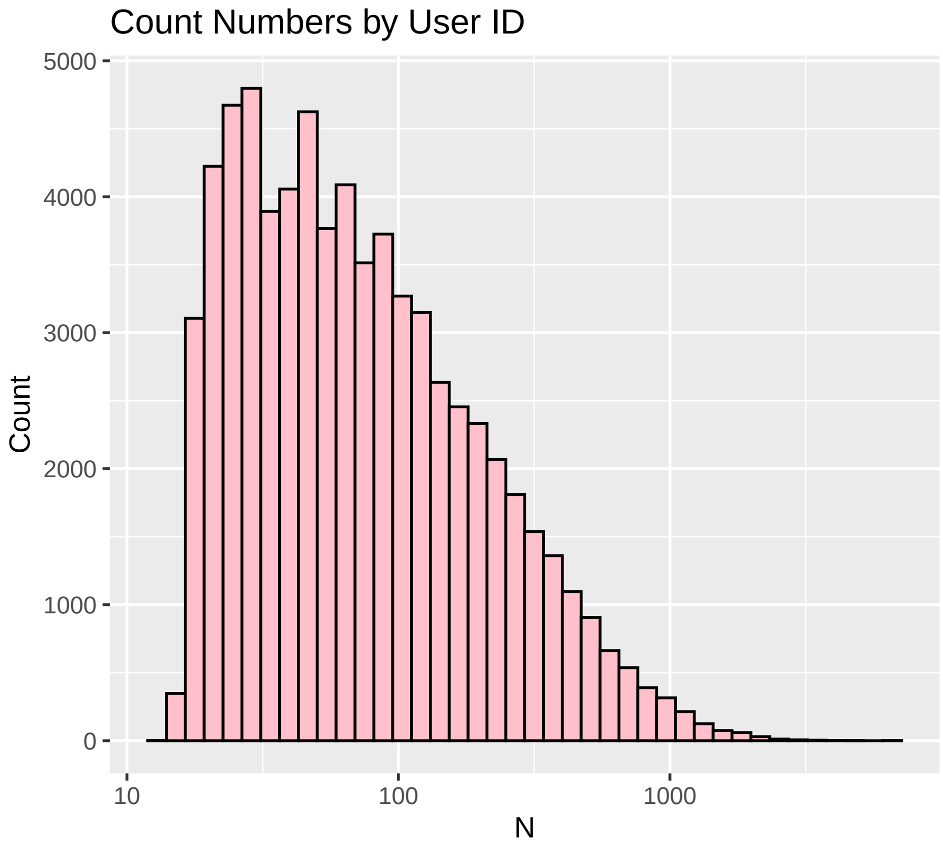


3.1.5 Users Feature

The unique length of the Users with respect to their ID's is '69878' and there are total '9000055' rows in the dataset.

```
## [1] 69878
```

The histogram for the Users Id is determined that is the left skewed. The count ratio of users id have maximum values from 30-70 range. The distribution of the dataset can be scene easily.



3.2 Data Partitioned for Training and Testing

The EDX dataset is partitioned for training and testing with the ratio of 90% and 10% respectively.

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

3.3 Data Cleaning Process

As stated previously, various characteristics may be utilized to forecast a user's rating based on his or her behavior. In this analysis, nevertheless, the predicted ratings are based just on info from the movie and from the users, since a large number of predictors enhances the computational burden of model and demands more computational power.

3.4 Modeling Process

The Modeling is very useful process which helps to build the on the choose variables. So, here various types of the Models aree being developed on the dataset and their MSE, MAE, MSE are calculated and compared the. Our main goal is to achieve the low RMSE score as compared to the baseline that is 0.864900.

3.4.1 Random Predictions Modeling

A very basic approach is to simply forecast the rating's randomly based on the distribution's of probability obtained throughout data analysis. For instance, if it is known that the likelihood of complete users assigning a film a ratings of '3' is '10 %', it may assume that '10%' to ratings will be 3. Thus a forecast establishes the worst-case scenario, implying that any other modeling must provide a more accurate outcome.

3.4.2 Linear Modeling

The easiest model forecast that all users would rate all movies equally and that the discrepancy between movies is due to arbitrarily distributed errors. Even though the projected rating may be whatever number, statistical theory dictates that the averaging minimizes the root mean square error (RMSE), and hence the first forecast is simply the mean of all recorded values (ratings).

$$\hat{b}_u = \frac{1}{N}\sum_{i=1}^{N}(y_{u,i} - \hat{b}_i - \hat{\mu})$$

3.4.3 Regularization Modeling

While the linear modeling offers an accurate prediction of the ratings, it ignores the fact that several movies get a small number of ratings and that certain people rate relatively few movies. It indicates that now the sample group for such movies & users is rather limited. This results in a substantial estimated mistake in terms of statistics. The approximate amount may be enhanced by having a component that penalizes small samples but has little or no effect on the estimated worth elsewhere. Therefore, the following formulae may be used to compute projected movie and user impacts.

$$\hat{b}_u = \frac{1}{n_u + \lambda}\sum_{i=1}^{n_u}(y_{u,i} - \hat{b}_i - \hat{\mu})$$

3.4.4 Matrix Factorization Method

It is a frequently used data mining technique for rating prediction in recommender system. This technique gained widespread attention even during Netflix prize competition (NetFlix). The recosystem package in R is utilized to implement the matrix factorization on the dataset (CRAN).

4 Results

The comparison and implementation of the various techniques is compared and find the most optimal solution for the problem.

4.1 Metrics for Models Assessment

The evaluation metrics are evaluated and assed the performance of each model, Their names are RMSE, MAE and MSE.

4.2 Random modeling Prediction

The random modeling prediction is being applied here on the dataset.

The RMSE values is very huge.

```
## # A tibble: 2 x 4
##   Method                    RMSE   MSE   MAE
##   <chr>                    <dbl> <dbl> <dbl>
## 1 Baseline (Project Goal) 0.865 NA    NA
## 2 Random prediction       1.50   2.25  1.17
```

4.3 Linear Modeling

It is implemented and results are shown below:

```
## # A tibble: 3 x 4
##   Method                    RMSE   MSE    MAE
##   <chr>                    <dbl> <dbl>  <dbl>
## 1 Baseline (Project Goal) 0.865 NA    NA
## 2 Random prediction       1.50   2.25  1.17
## 3 Mean                    1.06   1.12  0.855
```

4.3.1 Include Movie Effect (bi)

```
## # A tibble: 6 x 2
##   movieId    b_i
##     <dbl>  <dbl>
## 1       1  0.412
## 2       2 -0.312
## 3       3 -0.361
## 4       4 -0.626
## 5       5 -0.437
## 6       6  0.297
```

```
## # A tibble: 4 x 4
##   Method                    RMSE    MSE    MAE
##   <chr>                    <dbl>  <dbl>  <dbl>
## 1 Baseline (Project Goal) 0.865 NA     NA
## 2 Random prediction       1.50   2.25   1.17
## 3 Mean                    1.06   1.12   0.855
## 4 Mean + bi               0.943  0.889  0.737
```

4.3.2 Include User Effect (bu)

Predict the rating with mean + bi + bu

```
## # A tibble: 5 x 4
##   Method                    RMSE    MSE    MAE
##   <chr>                    <dbl>  <dbl>  <dbl>
## 1 Baseline (Project Goal) 0.865 NA     NA
## 2 Random prediction       1.50   2.25   1.17
## 3 Mean                    1.06   1.12   0.855
## 4 Mean + bi               0.943  0.889  0.737
## 5 Mean + bi + bu          0.865  0.748  0.668
```

4.3.3 Evaluating the model result

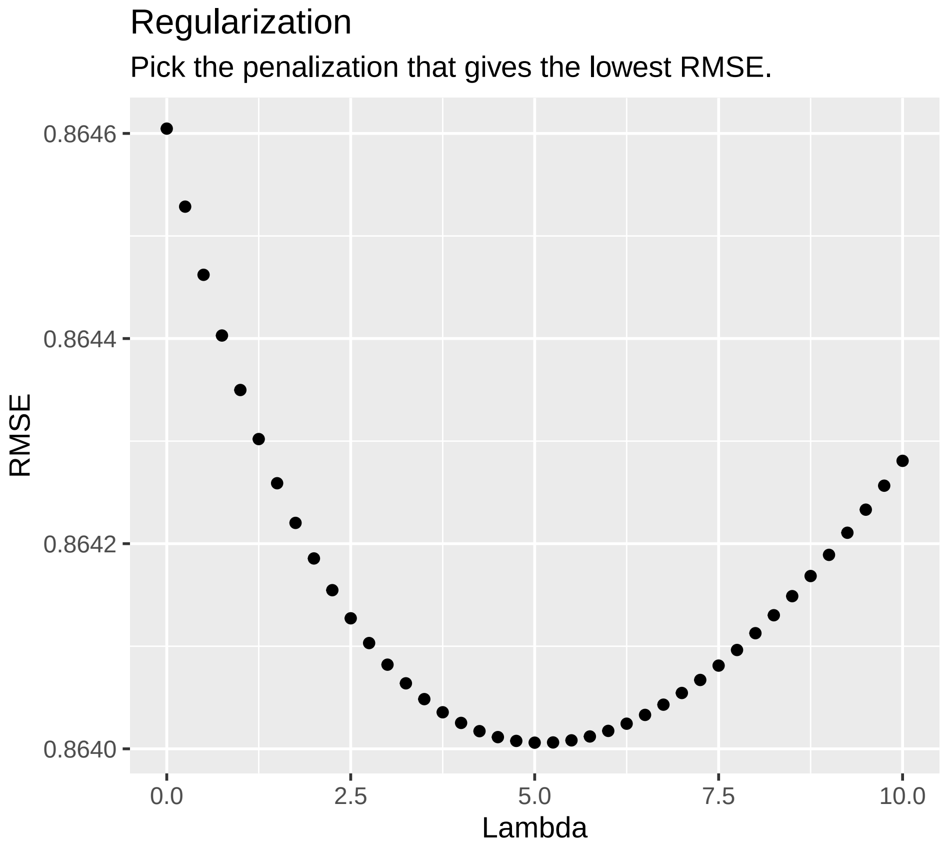
```
##     userId movieId rating                           title        b_i  residual
##  1:  26423    6483    5.0      From Justin to Kelly (2003) -2.5942749  4.081921
##  2:   5279    6371    5.0           PokÃ©mon Heroes (2003) -2.5399130  4.027559
##  3:  56965    6371    5.0           PokÃ©mon Heroes (2003) -2.5399130  4.027559
##  4:  57863    6371    5.0           PokÃ©mon Heroes (2003) -2.5399130  4.027559
##  5:   2507     318    0.5 Shawshank Redemption, The (1994)  0.9471604 -3.959514
##  6:   9214     318    0.5 Shawshank Redemption, The (1994)  0.9471604 -3.959514
##  7:   9568     318    0.5 Shawshank Redemption, The (1994)  0.9471604 -3.959514
##  8:   9975     318    0.5 Shawshank Redemption, The (1994)  0.9471604 -3.959514
##  9:  10680     318    0.5 Shawshank Redemption, The (1994)  0.9471604 -3.959514
## 10:  10749     318    0.5 Shawshank Redemption, The (1994)  0.9471604 -3.959514
```

Top 10 best movies (ranked by bi).
These are unknown movies

```
## # A tibble: 10 x 1
##    title
##    <chr>
##  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 Hospital (1970)
##  8 Constantine's Sword (2007)
##  9 Human Condition II, The (Ningen no joken II) (1959)
## 10 Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1~
```

4.4 Regularization Modeling

It is applied here.



Next, we apply the best 'lambda' to the linear model.

```
## # A tibble: 6 x 4
##   Method                    RMSE    MSE    MAE
##   <chr>                    <dbl>  <dbl>  <dbl>
## 1 Baseline (Project Goal) 0.865 NA     NA    
## 2 Random prediction       1.50   2.25   1.17 
## 3 Mean                    1.06   1.12   0.855
## 4 Mean + bi               0.943  0.889  0.737
## 5 Mean + bi + bu          0.865  0.748  0.668
## 6 Regularized bi and bu   0.864  0.747  0.669
```

4.5 Matrix Factorization Modeling

It took much time so recosystem package is applied on the dataset and compute the results for better RMSE, MSE and MAE score.

```
## Warning: package 'recosystem' was built under R version 4.0.5
```

```
## iter      tr_rmse          obj
##    0       0.9826   1.1072e+07
##    1       0.8763   9.0016e+06
##    2       0.8439   8.3618e+06
##    3       0.8218   7.9727e+06
##    4       0.8043   7.6977e+06
##    5       0.7906   7.4911e+06
##    6       0.7798   7.3418e+06
##    7       0.7710   7.2227e+06
##    8       0.7638   7.1283e+06
##    9       0.7577   7.0498e+06
##   10       0.7523   6.9859e+06
##   11       0.7475   6.9285e+06
##   12       0.7433   6.8839e+06
##   13       0.7394   6.8416e+06
##   14       0.7357   6.8038e+06
##   15       0.7324   6.7699e+06
##   16       0.7294   6.7416e+06
##   17       0.7265   6.7109e+06
##   18       0.7239   6.6871e+06
##   19       0.7215   6.6647e+06
```

```
##  [1] 5.120416 4.077981 3.699822 4.006856 3.629455 3.811159 4.298897 4.395237
##  [9] 4.066725 2.091906
```

It's a very robust model and increase the results potentially.

```
## # A tibble: 7 x 4
##   Method                              RMSE    MSE    MAE
##   <chr>                              <dbl>  <dbl>  <dbl>
## 1 Baseline (Project Goal)            0.865 NA     NA
## 2 Random prediction                  1.50   2.25   1.17
## 3 Mean                               1.06   1.12   0.855
## 4 Mean + bi                          0.943  0.889  0.737
## 5 Mean + bi + bu                     0.865  0.748  0.668
## 6 Regularized bi and bu              0.864  0.747  0.669
## 7 Matrix Factorization - recosystem 0.786  0.618  0.605
```

4.6 Final Validation Preocess

As seen in the result table, regularisation modeling and matrix factorization modeling both reduced the RMSE to the desired value. Ultimately, we trained both models on the whole 'edx' set and compute the root mean square error (RMSE) on the 'validation' subset. The project's objective is met if the root mean square error (RMSE) remains below the target value.

4.6.1 Linear Modeling With Regularization modeling

As during train and test data, the linear modeling with regularisation came within a narrow margin of the goal RMSE. With the 'validation' set, we perform the final validation.

```
## # A tibble: 8 x 4
##   Method                                     RMSE    MSE    MAE
##   <chr>                                     <dbl>  <dbl>  <dbl>
## 1 Baseline (Project Goal)                   0.865 NA     NA
## 2 Random prediction                         1.50   2.25   1.17
## 3 Mean                                      1.06   1.12   0.855
## 4 Mean + bi                                 0.943  0.889  0.737
## 5 Mean + bi + bu                          0.865  0.748  0.668
## 6 Regularized bi and bu                   0.864  0.747  0.669
## 7 Matrix Factorization - recosystem       0.786  0.618  0.605
## 8 Final Regularization (edx vs validation) 0.865  0.748  0.670
```

As expected, the root mean square error computed on the 'validation' set is less than the goal of 0.8649 and somewhat more than the root mean square error obtained on the test set.

Top 10 best movies

```
## # A tibble: 10 x 1
## # Groups:   title [9]
##    title
##    <chr>
##  1 Shawshank Redemption, The (1994)
##  2 Godfather, The (1972)
##  3 Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)
##  4 Goodfellas (1990)
##  5 Godfather, The (1972)
##  6 Pulp Fiction (1994)
##  7 Blade Runner (1982)
##  8 Annie Hall (1977)
##  9 Schindler's List (1993)
## 10 Usual Suspects, The (1995)
```

Top 10 worst movies

```
## # A tibble: 10 x 1
## # Groups:   title [9]
##    title
##    <chr>
##  1 Police Academy 5: Assignment: Miami Beach (1988)
##  2 RoboCop 3 (1993)
##  3 Iron Eagle IV (1995)
##  4 Eye of the Beholder (1999)
##  5 Pokemon 4 Ever (a.k.a. PokÃ©mon 4: The Movie) (2002)
##  6 Turbo: A Power Rangers Movie (1997)
##  7 Prom Night IV: Deliver Us From Evil (1992)
##  8 Slaughterhouse 2 (1988)
##  9 Pokemon 4 Ever (a.k.a. PokÃ©mon 4: The Movie) (2002)
## 10 Glitter (2001)
```

4.6.2 Matrix Factorization

The first test demonstrates that matrix factorization produces the lowest root mean square error. Validate now using the whole 'edx' and 'validation' sets.

```
## iter      tr_rmse          obj
##    0       0.9720   1.2008e+07
##    1       0.8725   9.8772e+06
##    2       0.8385   9.1712e+06
##    3       0.8163   8.7475e+06
##    4       0.8010   8.4706e+06
##    5       0.7892   8.2695e+06
##    6       0.7795   8.1236e+06
##    7       0.7715   8.0012e+06
##    8       0.7646   7.9038e+06
##    9       0.7587   7.8249e+06
##   10       0.7536   7.7560e+06
##   11       0.7490   7.7015e+06
##   12       0.7448   7.6484e+06
##   13       0.7410   7.6070e+06
##   14       0.7375   7.5663e+06
##   15       0.7343   7.5310e+06
##   16       0.7314   7.5002e+06
##   17       0.7287   7.4722e+06
##   18       0.7262   7.4466e+06
##   19       0.7240   7.4242e+06
```

The final RMSE with matrix factorization is better than the linear model with regularization (Final Matrix Factorization - RMSE 0.7834771)

```
## # A tibble: 9 x 4
##   Method                                    RMSE    MSE    MAE
##   <chr>                                    <dbl>  <dbl>  <dbl>
## 1 Baseline (Project Goal)                  0.865 NA     NA
## 2 Random prediction                        1.50   2.25   1.17
## 3 Mean                                     1.06   1.12   0.855
## 4 Mean + bi                                0.943  0.889  0.737
## 5 Mean + bi + bu                           0.865  0.748  0.668
## 6 Regularized bi and bu                    0.864  0.747  0.669
## 7 Matrix Factorization - recosystem        0.786  0.618  0.605
## 8 Final Regularization (edx vs validation) 0.865  0.748  0.670
## 9 Final Matrix Factorization - recosystem  0.783  0.613  0.604
```

Now, let's check the best and worst movies predicted with matrix factorization.

Top 10 best movies:

```
## # A tibble: 10 x 1
## # Groups:   title [8]
##    title
##    <chr>
##  1 Becket (1964)
##  2 Cats Don't Dance (1997)
##  3 Lord of the Rings: The Return of the King, The (2003)
##  4 Star Wars: Episode V - The Empire Strikes Back (1980)
##  5 Shawshank Redemption, The (1994)
##  6 Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)
##  7 Shawshank Redemption, The (1994)
##  8 Pulp Fiction (1994)
##  9 Lord of the Rings: The Fellowship of the Ring, The (2001)
## 10 Pulp Fiction (1994)
```

Top 10 worst movies:

```
## # A tibble: 10 x 1
## # Groups:   title [10]
##    title
##    <chr>
##  1 Beast of Yucca Flats, The (1961)
##  2 Siegfried & Roy: The Magic Box (1999)
##  3 Pearl Harbor (2001)
##  4 Zombie Lake (Le Lac des morts vivants) (1981)
##  5 Separation, The (La SÃ©paration) (1994)
##  6 Switchblade Sisters (1975)
##  7 Murder on a Sunday Morning (Un coupable idÃ©al) (2001)
##  8 Santa Clause 3: The Escape Clause, The (2006)
##  9 Texas Chainsaw Massacre, The (1974)
## 10 Message in a Bottle (1999)
```

5 Conclusion

We began by gathering and prepping the data for evaluation, and then analyzing the it in search of findings that may aid in the modelling process. Secondly, using the probabilistic model of every ratings, we constructed a randomized modeling that predict the ratings. The said model produces the most inaccurate outcome. Afterward, we began the linear modeling by constructing a very basic model consisting just from the average of the observed data. From and where it, we incorporated movie and users' effects to simulate user activity and distribution of movies. We introduced a penalties value to regularisation for films and people with a small number of rating. The linear modeling produced a root mean square error of 0.8648, above the objective of 0.8649. Lastly, we assessed the recosystem library, which implements the LIBMF method, and determined that the root mean square error (RMSE) was 0.7826 that is the most lowest one.