Movielens Capstone Project

Agnes Munee

1/20/2022

This project is based on the data Movielens, which aims to make movie recommendatuons to viewers/users based on the movies they have already watched and the rating that they gave to those movies.The features or variables under consideration are the UserID(which is the user identification), the movieId (which is the movie Identification), the title of the movie and the genre of each movie.The "rating" is the target variable that we will be predicting.First, we analysed the data, and derived the key relationships between the variables and the target variable. This was visualized in diagramatic plots. The strength of the mentioned relationships was also measured using the correlation function. Second, we tried a couple of models, which we have descrbed in the Modelling & Results section of this report. Finally, we tested the optimal model on the validation data to determine the final RMSE.

Method and Analysis

```
# Create edx set, validation set (final hold-out test set)- (provided by the edx team)
```

```
## Loading required package: tidyverse
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift

## Loading required package: data.table

##
## Attaching package: 'data.table'

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

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

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

#Check if the data has been loaded correctly

```
## [1] 9000061       6

## [1] 999993      6
```

#Data Exploration #Split the edx data into train and test data:

#Explore the train data set

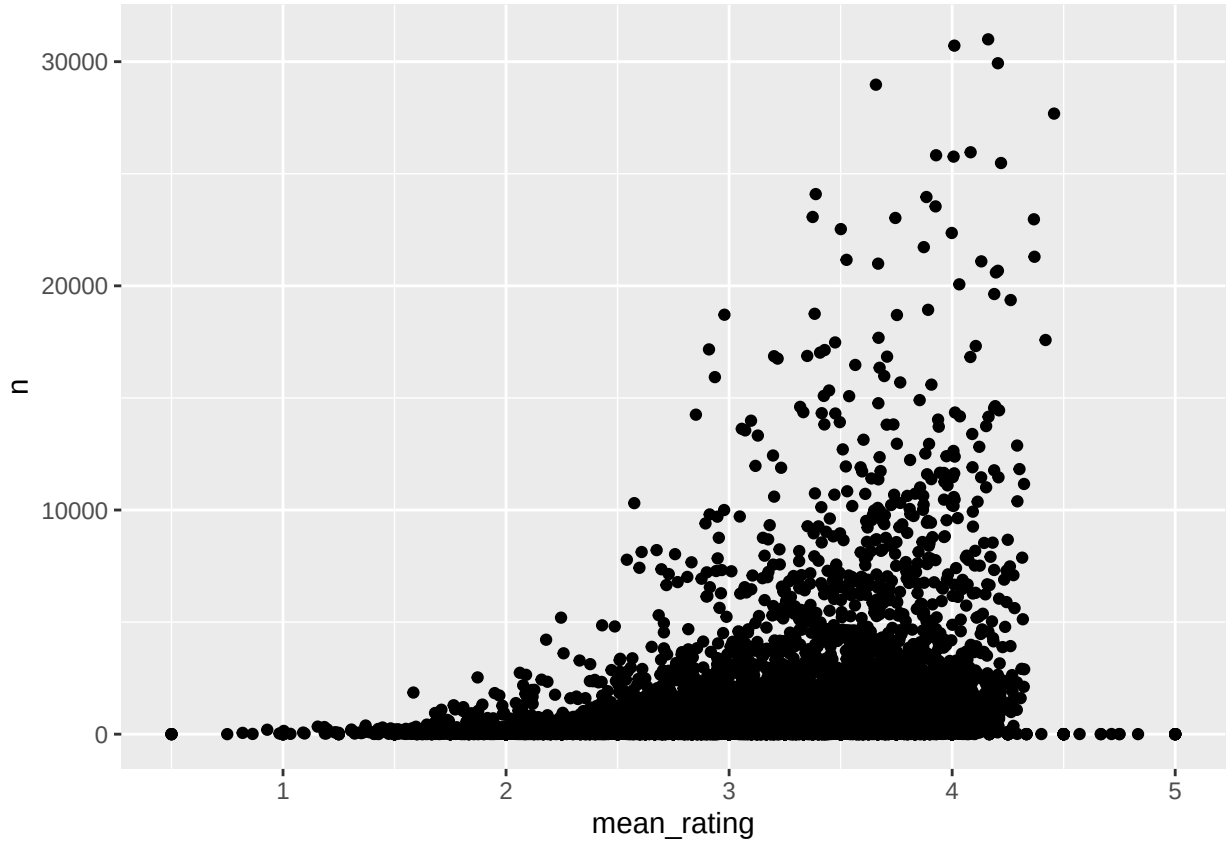
```
## Classes 'data.table' and 'data.frame':   8900060 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 292 316 329 355 356 362 364 370 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983421 838983392 838983392 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|Ad
##  - attr(*, ".internal.selfref")=<externalptr>

##      userId         movieId          rating        timestamp
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08
##  1st Qu.:18702   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08
##  Median :36148   Median : 1834   Median :4.000   Median :1.036e+09
##  Mean   :36225   Mean   : 4123   Mean   :3.512   Mean   :1.033e+09
##  3rd Qu.:53779   3rd Qu.: 3629   3rd Qu.:4.000   3rd Qu.:1.127e+09
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09
##     title              genres
##  Length:8900060     Length:8900060
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

#movieId

```
## # A tibble: 20 x 3
##    movieId     n mean_rating
##      <dbl> <int>       <dbl>
##  1    3226     1        5
##  2   33264     2        5
##  3   42783     1        5
##  4   51209     1        5
##  5   53355     1        5
##  6   64275     1        5
##  7   65001     1        5
##  8   26048     3        4.83
##  9    5194     4        4.75
## 10   26073     4        4.75
## 11    4454     7        4.71
## 12    5849     3        4.67
## 13   63808     3        4.67
## 14   32657     7        4.57
## 15    7452     1        4.5
## 16    7823     1        4.5
## 17   25975     3        4.5
## 18   26049     2        4.5
## 19   50477     1        4.5
## 20   53883     4        4.5
```

```
## # A tibble: 20 x 3
##    movieId     n mean_rating
##      <dbl> <int>       <dbl>
##  1    5805     2       0.5
##  2    8394     1       0.5
##  3    8707     1       0.5
##  4   61768     1       0.5
##  5   64999     2       0.75
##  6    8859    58       0.819
##  7    7282    11       0.864
##  8    6483   198       0.929
##  9   61348    31       0.984
## 10    3561     1       1
## 11    4071     1       1
## 12    4075     1       1
## 13    6189     1       1
## 14    6766     2       1
## 15   33160     1       1
## 16   55324     1       1
## 17    6371   145       1.00
## 18    8856    15       1.03
## 19    3574    72       1.09
## 20    4051    31       1.10
```



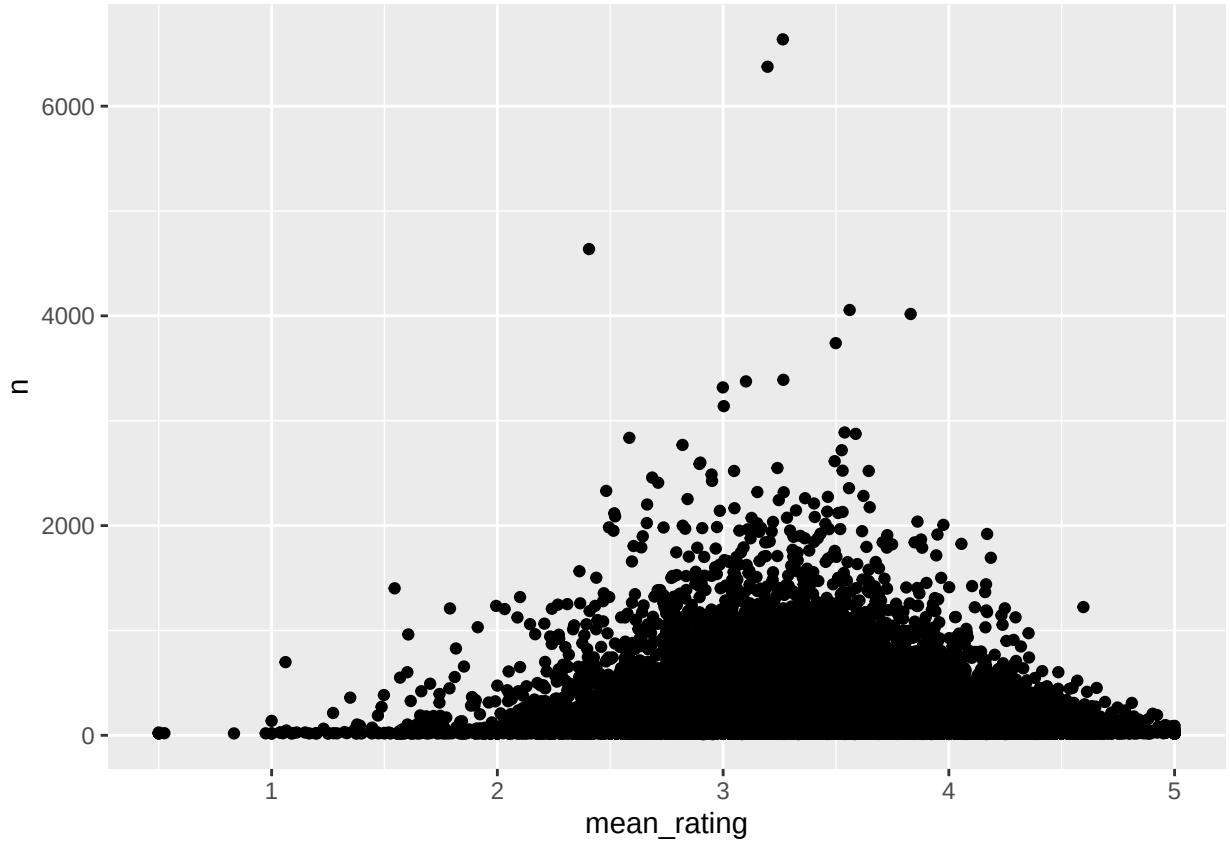
The scatter plot above shows that the movies with the highest and lowest mean rating have been rated very few times. Thus, it would be wise to penalize the rating with the lowest number of ratings.This will be referred to as regularization.

#userId

```
## # A tibble: 20 x 3
##    userId     n mean_rating
##     <int> <int>       <dbl>
##  1      1    20           5
##  2   1686    19           5
##  3   7984    15           5
##  4  11884    19           5
##  5  13027    31           5
##  6  13513    19           5
##  7  13524    19           5
##  8  15575    28           5
##  9  18965    50           5
## 10  22045    22           5
## 11  26308    14           5
## 12  27831    19           5
## 13  30519    19           5
## 14  35184    22           5
## 15  42649    20           5
## 16  52674    18           5
## 17  52749    90           5
## 18  54009    26           5
```

```
## 19  65873    19           5
## 20  68379    57           5
```

```
## # A tibble: 20 x 3
##    userId     n mean_rating
##     <int> <int>       <dbl>
##  1  13496    19       0.5
##  2  48146    27       0.5
##  3  49862    18       0.5
##  4  62815    19       0.5
##  5  63381    20       0.525
##  6   6322    18       0.833
##  7   8920    19       0.974
##  8   3457    16       1
##  9  24176   138       1
## 10  24490    19       1
## 11  15515    27       1.04
## 12  28416    26       1.04
## 13  43628    20       1.05
## 14  59342   698       1.06
## 15  24101    46       1.07
## 16  31710    17       1.09
## 17  19059    21       1.10
## 18  30585    27       1.11
## 19  42019    27       1.15
## 20  52056    18       1.17
```

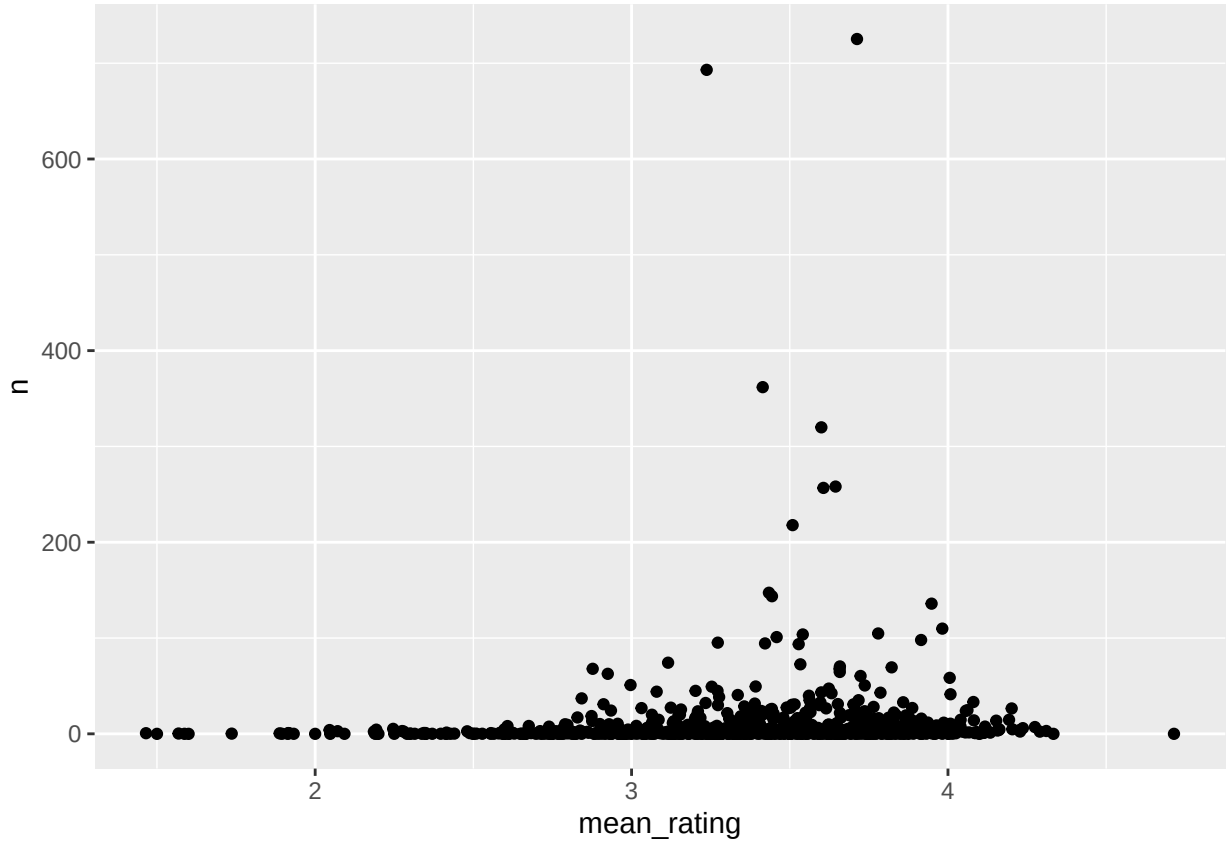


From the above scatter graph, the distribution of the mean rating of movies per each user is normal.

#Genres

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

```
## # A tibble: 797 x 3
##    genres                                        n mean_rating
##    <fct>                                     <int>       <dbl>
##  1 Animation|IMAX|Sci-Fi                         7        4.71
##  2 Adventure|Fantasy|Film-Noir|Mystery|Sci-Fi    3        4.33
##  3 Drama|Film-Noir|Romance                    2961        4.31
##  4 Action|Crime|Drama|IMAX                    2313        4.29
##  5 Animation|Children|Comedy|Crime            7094        4.27
##  6 Film-Noir|Mystery                          5933        4.24
##  7 Film-Noir|Romance|Thriller                 2394        4.23
##  8 Crime|Film-Noir|Mystery                    3940        4.23
##  9 Crime|Film-Noir|Thriller                   4764        4.20
## 10 Crime|Mystery|Thriller                    26487        4.20
## # ... with 787 more rows
```



There are 797 unique genres. The most watched genre is drama. There seems to be a genres bias, as the genre with the highest mean_rating(Animation) has only 7 movies. Thus, regularization will be needed to remove the bias.

#Timestamp

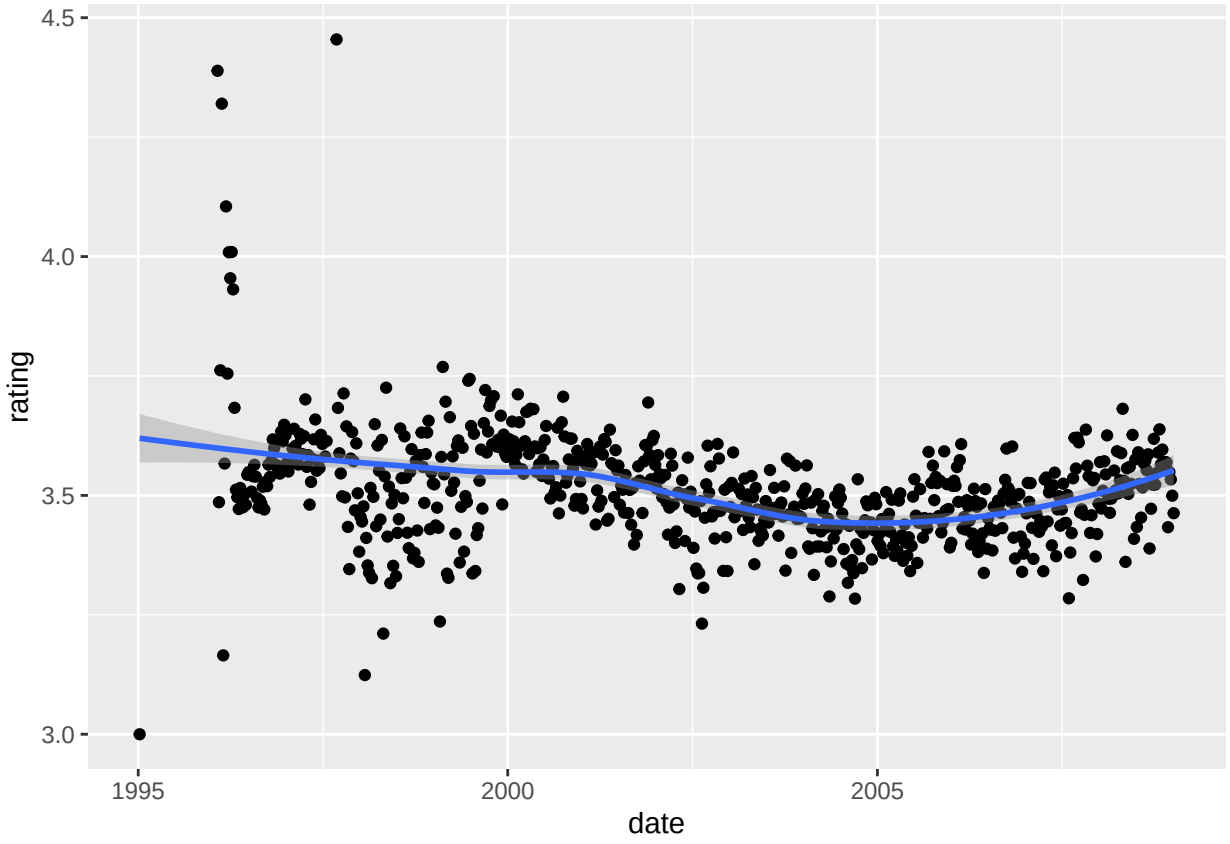
Split the timestamp to date

```
##
## 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

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



From the graph, there is some effect of time on the movie rating, however very small.Thus I will not use "date" as a variable in my model.

Modelling Approach. The variables that have a significant effect on the movie rating are UserId, movieId and genres.

Modelling and Results

```
# First Model: Just the mean
```

```
##           Model     RMSE
## 1 Just the Mean 1.059537
```

#Second Model:The movie Effect

```
##           Model      RMSE
## 1 Just the Mean 1.0595372
## 2  Movie Effect 0.9424554
```

#The Third model: The movie + user Effect

```
##                  Model      RMSE
## 1       Just the Mean 1.0595372
## 2        Movie Effect 0.9424554
## 3 Movie + User Effect 0.8656137
```

#The Forth model: The movie + user + genre Effect

```
##                          Model      RMSE
## 1                Just the Mean 1.0595372
## 2                 Movie Effect 0.9424554
## 3          Movie + User Effect 0.8656137
## 4 Movie + User + Genre Effect 0.8652964
```

The fifth Model: Regularization of the variables

From the above Results, the RMSE keeps decreasing. From the Analysis we did, we noticed that there is bias in the movieId, userId and genres, due to the few number of ratings in each class. Thus, the next step will be to regularize the variables so as to remove the bias. The more a movie is rated,the more a user rates movies and the more a certain type of genre is rated, the better we can recommend a certain type of movie in a specific genre to a specific user.

Regularization involves penalizing the movies that have the least number of ratings, the users who have rated the least number of movies and the genres that have been rated the least number of times.

We will need to identify the threshold number that will produce the lowest RMSE.

```
## [1] 0.8647226
```

```
##                          Model      RMSE
## 1                Just the Mean 1.0595372
## 2                 Movie Effect 0.9424554
## 3          Movie + User Effect 0.8656137
## 4 Movie + User + Genre Effect 0.8652964
## 5       Regularization Effect 0.8647226
```

Regularization has greatly reduced the RMSE. The final model will be that which give the minimum RMSE, which is the Fifth model,i.e, the After the regularization of the variables. The value of lambda that produces the minimum RMSE is 5

#Test the final model on the Validation data

```
## [1] 0.8647194
```

#Conclusion An RMSE of 0.8647194 has been attained on the Validation data. The model, based on 3 variables can be improved by increasing the number of variables that have an effect on the predicted rating of a movie recommended. This model is only limited to the movies that have an existing rating and those that are in the movie site. The more a movie is rated the better the predictions, and thus as more users watch and rate the movies, the better the model will become.

Thus, the model has potential to reduce the RMSE and improve predictions in future as more users rate more movies, which will reduce the user and movie bias.