MovieLens Capstone Project

Tarun Ch. Bordoloi

3/7/2020

1.0 Executive summary:

1.1 The Data set: Overview:

This project is a part of the HarvardX: PH125.9x: Data Science: Capstone. The primary objective of the project is to create a movie recommendation system using the MovieLens dataset. MovieLens itself is a research site run by GroupLens Research group at the University of Minnesota. The first automated recommender system was developed there in 1993. The full data set contains 26,000,000 ratings and 750,000 tag applications applied to 45,000 movies by 270,000 users. In this project ,however, we shall be using the 10M version of the MovieLens dataset . This particular data set contains 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users.

1.2 Summary goals:

The purpose of the recommender system being developed in this project is to predict user movie ratings based on other users' ratings. The data set has been split into 2 parts ,namely the 'edx' set and the 'validation' set. Algorithm has been developed using the 'edx' set. For a final test of the algorithm, movie ratings were predicted in the 'validation' set as if they were unknown. RMSE (Root Mean Square Error) has been used to evaluate how close the predictions are to the true values in the validation set.

1.3 Key steps performed :

- Downloaded the dataset $ Ensured that the required packages and libraries are installed $ Splitted the data set into 'edx' and 'validation' set

- Carried out exploration of the data and performed feature engineering $ Included data visualization tools as required $ Incorporated insights gained

- Models were developed and those were evaluated $ Results tabulated $ The performance of our final model was evaluated based on the 'Penalized Root Mean Squared Error' approach. This algorithm achieved a RMSE of 0.86482 while testing on the 'validation set' • Conclusion stated

2.2 Data exploration & Feature Egineering

```
## Observations: 9,000,055
## Variables: 6
## $ userId    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ movieId   <dbl> 122, 185, 292, 316, 329, 355, 356, 362, 364, 370, 377, 42...
## $ rating    <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ...
## $ timestamp <int> 838985046, 838983525, 838983421, 838983392, 838983392, 83...
## $ title     <chr> "Boomerang (1992)", "Net, The (1995)", "Outbreak (1995)",...
## $ genres    <chr> "Comedy|Romance", "Action|Crime|Thriller", "Action|Drama|...
```

It would appear that the'edx' data set has 9,000,055 observations and 6 variables

```
## Observations: 999,999
## Variables: 6
## $ userId    <int> 1, 1, 1, 2, 2, 2, 3, 3, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, ...
## $ movieId   <dbl> 231, 480, 586, 151, 858, 1544, 590, 4995, 34, 432, 434, 8...
## $ rating    <dbl> 5.0, 5.0, 5.0, 3.0, 2.0, 3.0, 3.5, 4.5, 5.0, 3.0, 3.0, 3....
## $ timestamp <int> 838983392, 838983653, 838984068, 868246450, 868245645, 86...
## $ title     <chr> "Dumb & Dumber (1994)", "Jurassic Park (1993)", "Home Alo...
## $ genres    <chr> "Comedy", "Action|Adventure|Sci-Fi|Thriller", "Children|C...
```

The ‘validation’ data set has 999,999 observations and same 6 variables.

These variables are :

$ userId `<integer>` which contains an unique identification number of each user $ movieId `<numeric>` which contains an unique identification number for each movie $ timestamp `<integer>` which contains timestamp for a specific rating provided by one user $ title `<character>` which contains the title of each movie with the year of the release $ genres `<character>` which contains the list of pipe-delimited genre of each movie $ rating `<numeric>` which contains raing for each movie by one user.Movies are rated in a 5 star scale in an increment of half star

```
##   Unique_Users Unique_Movies Unique_Genres
## 1        69878         10677           797
```

There are 69878 unique users, 10677 unique movies and 797 unique genres

```
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 3      1     292      5 838983421               Outbreak (1995)
## 4      1     316      5 838983392               Stargate (1994)
## 5      1     329      5 838983392 Star Trek: Generations (1994)
## 6      1     355      5 838984474       Flintstones, The (1994)
##                          genres year_rated
## 1               Comedy|Romance       1996
## 2        Action|Crime|Thriller       1996
## 3  Action|Drama|Sci-Fi|Thriller       1996
## 4      Action|Adventure|Sci-Fi       1996
## 5 Action|Adventure|Drama|Sci-Fi       1996
## 6      Children|Comedy|Fantasy       1996
```

```
##   userId movieId rating timestamp
## 1      1     231      5 838983392
## 2      1     480      5 838983653
## 3      1     586      5 838984068
## 4      2     151      3 868246450
## 5      2     858      2 868245645
## 6      2    1544      3 868245920
##                                                title
## 1                               Dumb & Dumber (1994)
## 2                               Jurassic Park (1993)
## 3                                  Home Alone (1990)
## 4                                     Rob Roy (1995)
## 5                              Godfather, The (1972)
## 6 Lost World: Jurassic Park, The (Jurassic Park 2) (1997)
##                                  genres year_rated
## 1                                Comedy       1996
## 2       Action|Adventure|Sci-Fi|Thriller       1996
```

```
## 3                           Children|Comedy       1996
## 4                  Action|Drama|Romance|War       1997
## 5                               Crime|Drama       1997
## 6 Action|Adventure|Horror|Sci-Fi|Thriller       1997
```

Extracting the year of release of each movie and creating 'year' column .As would be observed release date of each movie is included with the "title'

```
edx <- edx %>% mutate(year = as.numeric(str_sub(title,-5,-2)))
head(edx)
```

```
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 3      1     292      5 838983421               Outbreak (1995)
## 4      1     316      5 838983392               Stargate (1994)
## 5      1     329      5 838983392 Star Trek: Generations (1994)
## 6      1     355      5 838984474       Flintstones, The (1994)
##                           genres year_rated year
## 1                 Comedy|Romance       1996 1992
## 2          Action|Crime|Thriller       1996 1995
## 3   Action|Drama|Sci-Fi|Thriller       1996 1995
## 4        Action|Adventure|Sci-Fi       1996 1994
## 5 Action|Adventure|Drama|Sci-Fi       1996 1994
## 6        Children|Comedy|Fantasy       1996 1994
```

	x
userId	0
movieId	0
rating	0
timestamp	0
title	0
genres	0
year_rated	0
year	0

It appears there is no missing value in any column

```
##   userId movieId rating                        genres year_rated year
## 1      1     122      5                Comedy|Romance       1996 1992
## 2      1     185      5         Action|Crime|Thriller       1996 1995
## 3      1     292      5  Action|Drama|Sci-Fi|Thriller       1996 1995
## 4      1     316      5       Action|Adventure|Sci-Fi       1996 1994
## 5      1     329      5 Action|Adventure|Drama|Sci-Fi       1996 1994
## 6      1     355      5       Children|Comedy|Fantasy       1996 1994
```

```
##   userId movieId rating                                  genres year_rated
## 1      1     231      5                                  Comedy       1996
## 2      1     480      5        Action|Adventure|Sci-Fi|Thriller       1996
## 3      1     586      5                         Children|Comedy       1996
## 4      2     151      3                Action|Drama|Romance|War       1997
## 5      2     858      2                             Crime|Drama       1997
## 6      2    1544      3 Action|Adventure|Horror|Sci-Fi|Thriller       1997
```

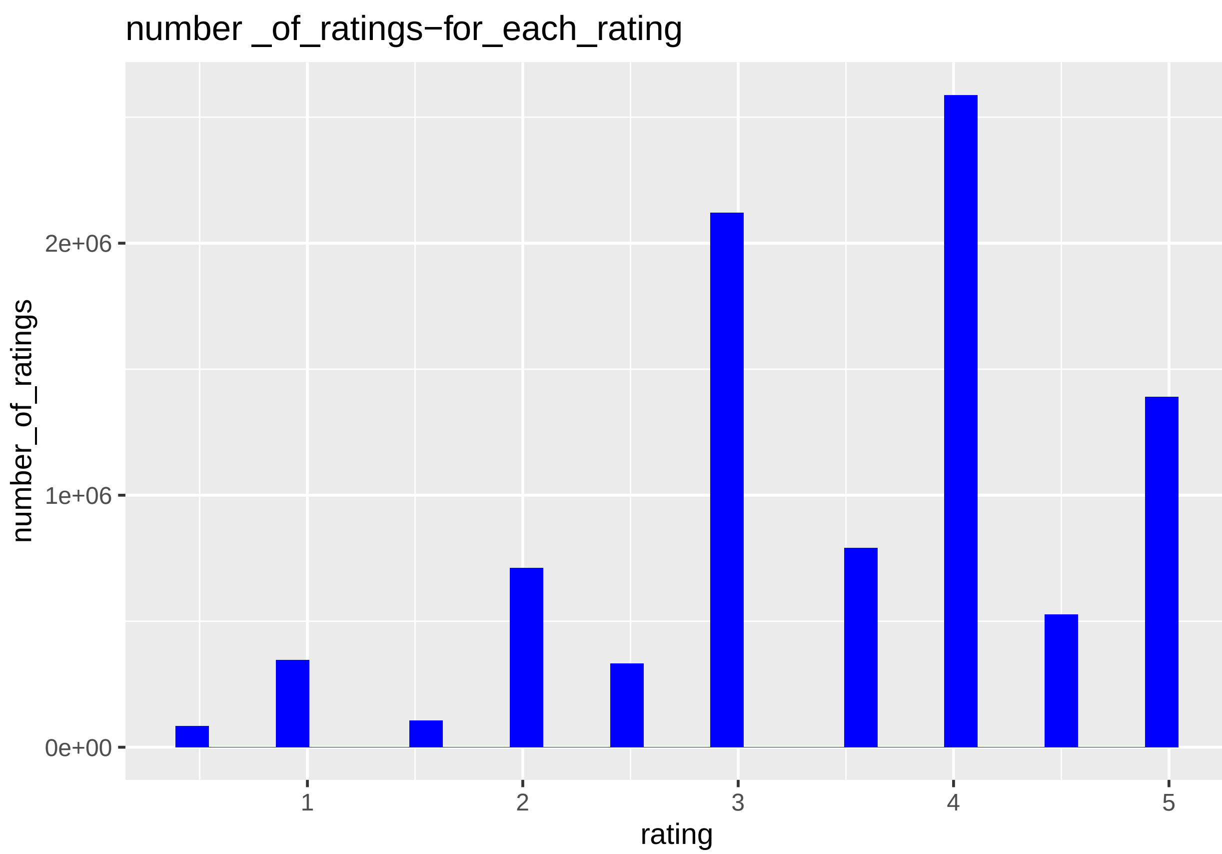
```
##      userId         movieId          rating          genres
##  Min.   :    1   Min.   :    1   Min.   :0.500   Length:9000055
##  1st Qu.:18124   1st Qu.:  648   1st Qu.:3.000   Class :character
##  Median :35738   Median : 1834   Median :4.000   Mode  :character
##  Mean   :35870   Mean   : 4122   Mean   :3.512
##  3rd Qu.:53607   3rd Qu.: 3626   3rd Qu.:4.000
##  Max.   :71567   Max.   :65133   Max.   :5.000
##    year_rated        year
##  Min.   :1995   Min.   :1915
##  1st Qu.:2000   1st Qu.:1987
##  Median :2002   Median :1994
##  Mean   :2002   Mean   :1990
##  3rd Qu.:2005   3rd Qu.:1998
##  Max.   :2009   Max.   :2008
```

Let us create a data frame 'rating_distribution' with half star and whole star rating from the 'edx' data set

```
##   edx.rating      group
## 1          5 whole_star
## 2          5 whole_star
## 3          5 whole_star
## 4          5 whole_star
## 5          5 whole_star
## 6          5 whole_star
```

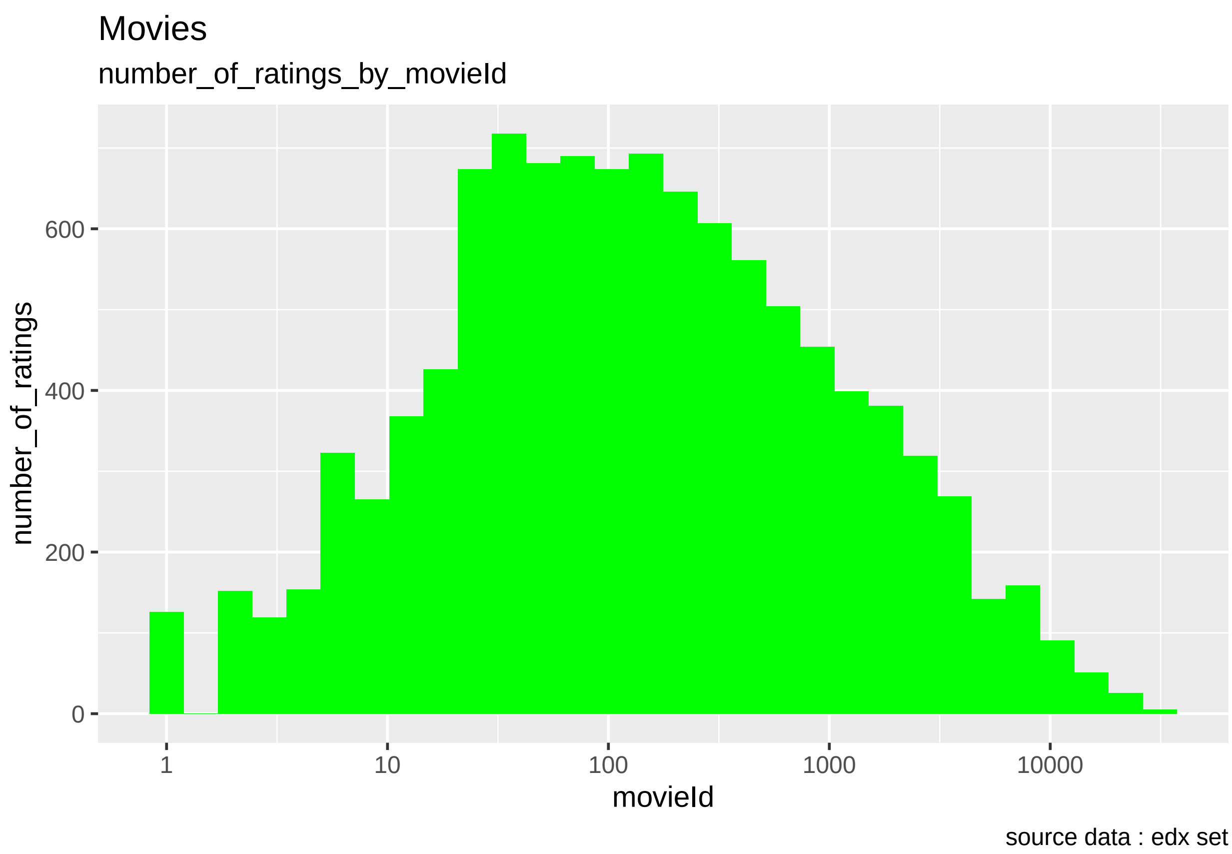
Histogram of ratings

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

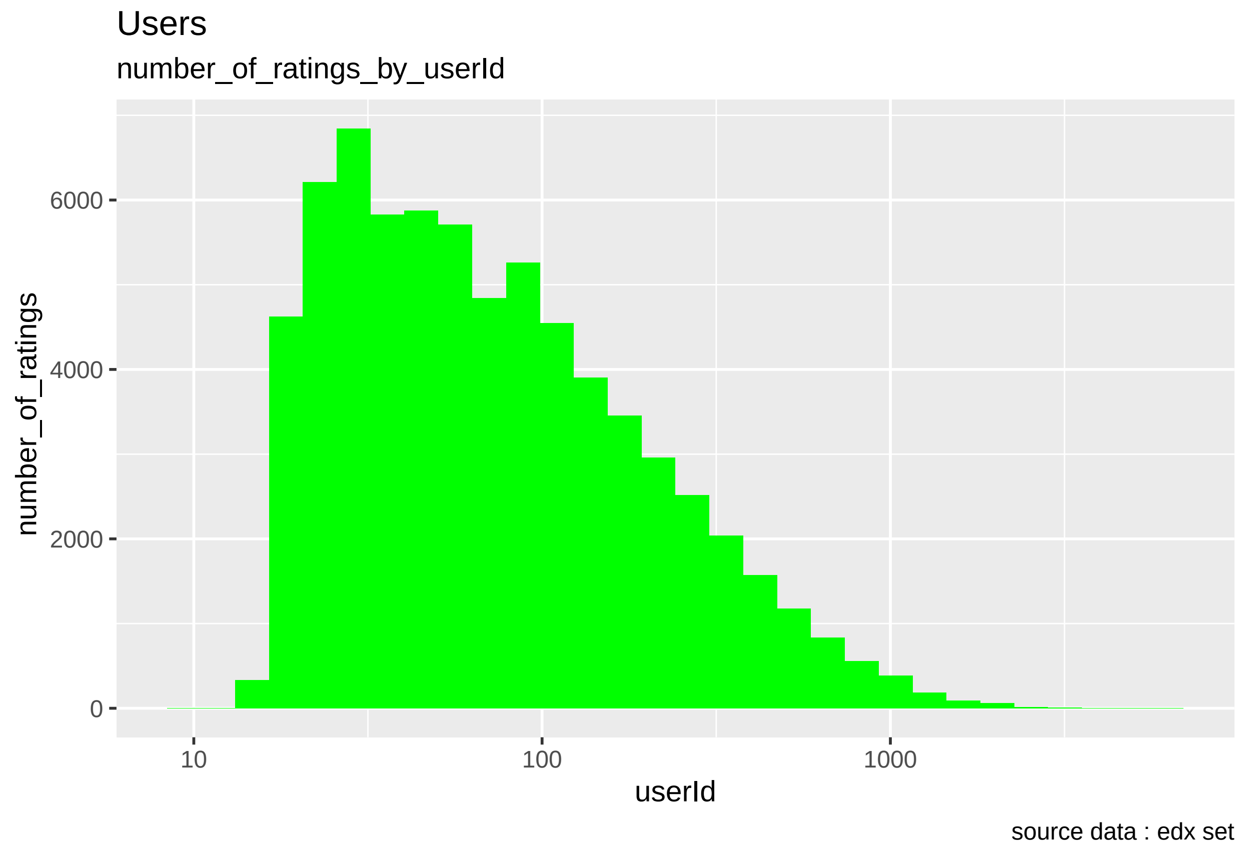


It is observed that : $ No user gives a 0 rating $ The distribution of 'rating' is left skewed $ Number of ratings are in the descending order are 4,3,5,3.5and 2 $ Higher ratings are more than the lower ratings $ Half star ratings are less common It is likely that an user habitually recommends a movie only if he/she likes it somewhat strongly. This is just a possibility though

```
## Warning: Ignoring unknown parameters: bin

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

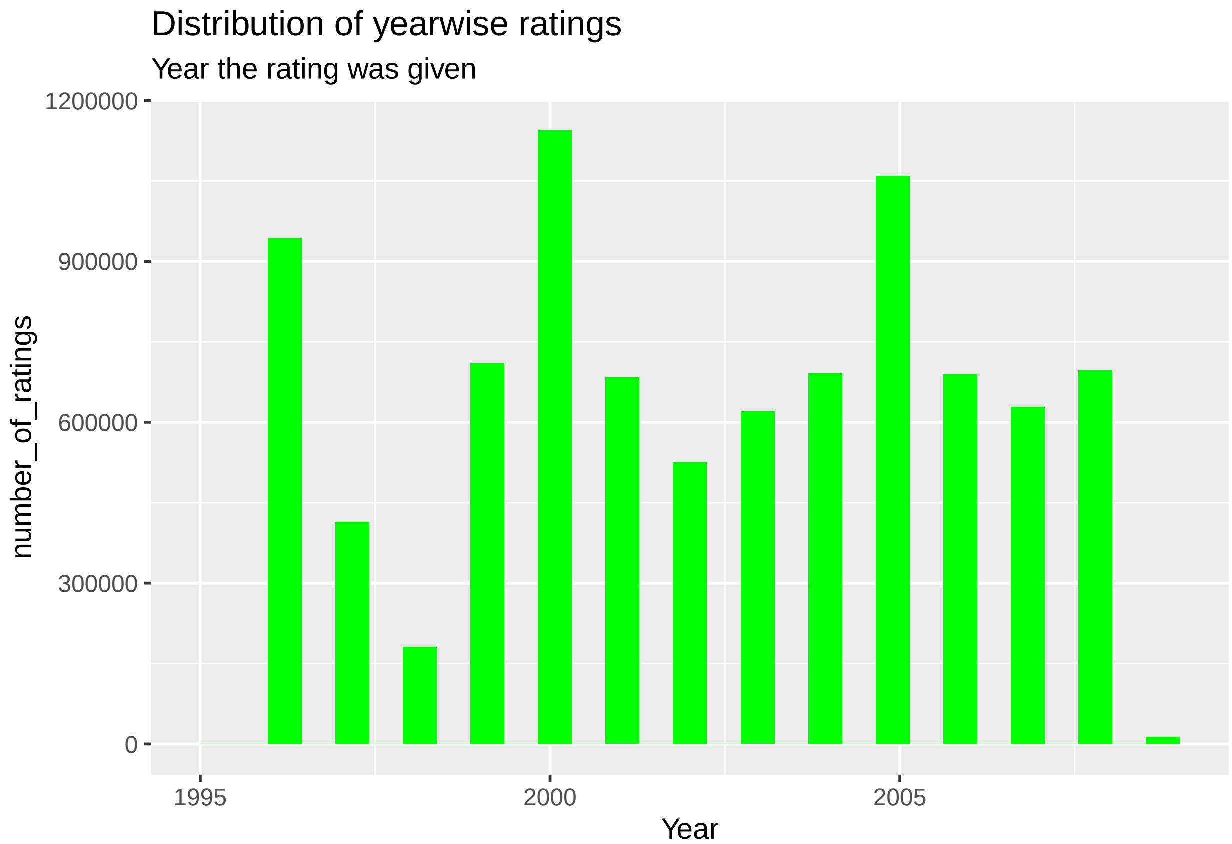


Movies
number_of_ratings_by_movieId
number_of_ratings
0
200
400
600
1
10
100
1000
10000
movieId
source data : edx set



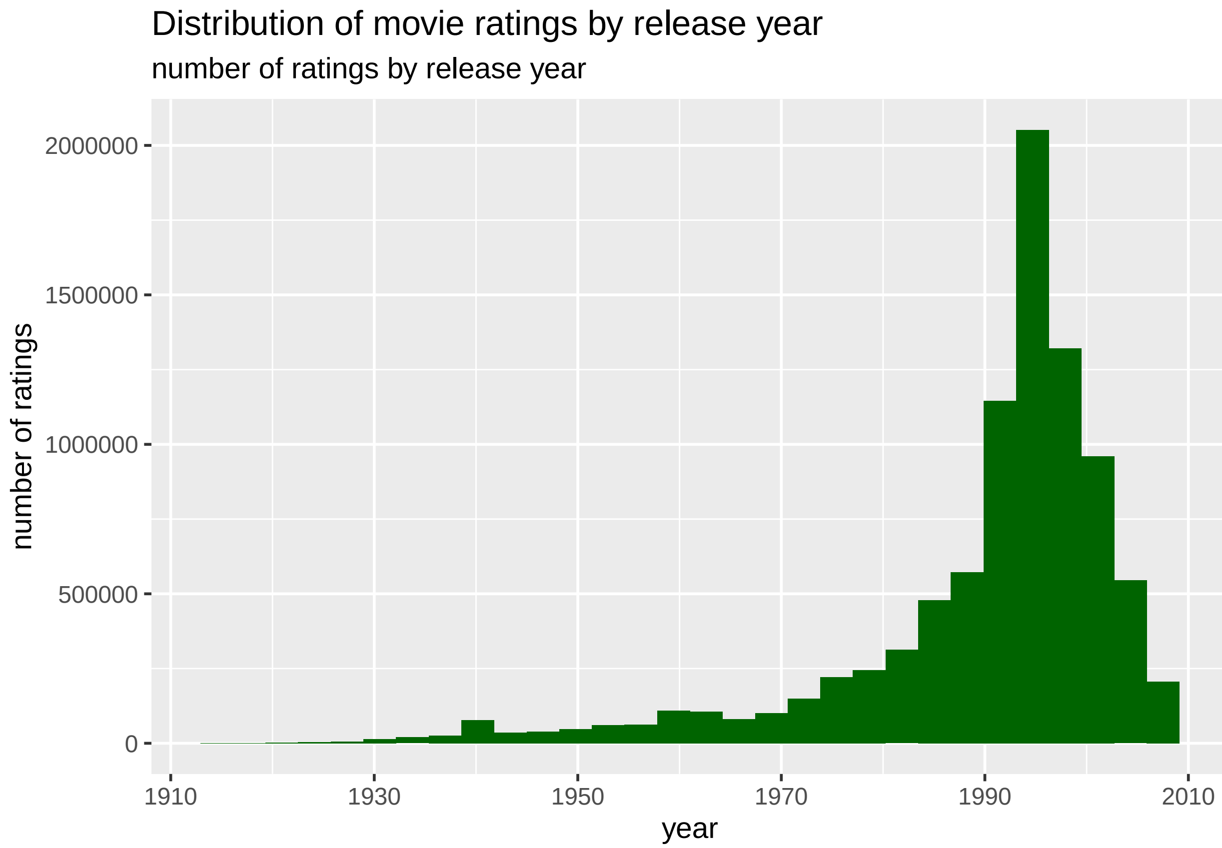
From the above analysis it appears that some movies are rated significantly more than the others while some users are more active in rating movies.These phenomena likely to suggest presence of strong movie effect and user effect on the ratings

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



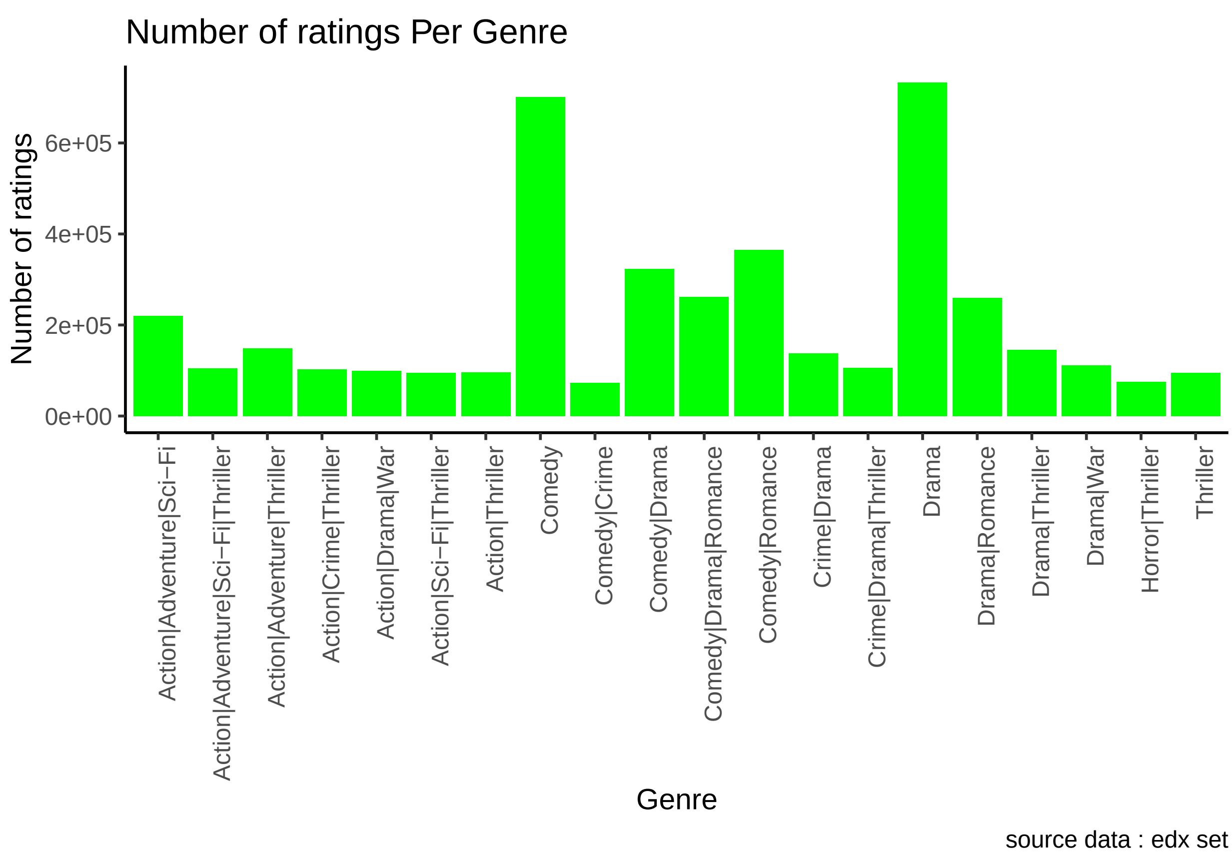
$ Year wise ratings appear to be irregular $ 1998 and 2002 have fewer ratings Having observed such behaviour I would not consider the feature ‘year_rated’ of the data to be a reliable predictor.

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



$ Ratings by release year distribution is clearly left skewed $ Movies were rated most which were released during the period 1990 and 2006 which seemed to have tapered down there after.This could possibly be a reflection of the innovation of technology and upswing in its use by the consumers(users in this context) and tapering down phenomenon may well reflect the looming economic crisis of 2008. Due to such inconsistency this feature, 'year_release', may not be a reliable feature of the data set to be considered as a predictor.

genres	count
Drama	733296
Comedy	700889
Comedy|Romance	365468
Comedy|Drama	323637
Comedy|Drama|Romance	261425
Drama|Romance	259355
Action|Adventure|Sci-Fi	219938
Action|Adventure|Thriller	149091
Drama|Thriller	145373
Crime|Drama	137387
Drama|War	111029
Crime|Drama|Thriller	106101
Action|Adventure|Sci-Fi|Thriller	105144
Action|Crime|Thriller	102259
Action|Drama|War	99183
Action|Thriller	96535
Action|Sci-Fi|Thriller	95280
Thriller	94662
Horror|Thriller	75000
Comedy|Crime	73286



$ Most rated category appears to be 'Drama' , 'Comedy' and 'Comedy|Romance' meriting the 2nd and 3rd place.However,the difference between the 2nd and 3rd appears to be fairly large.This could also mean 1st and 2nd are the most watched categories in that order. $ We ,however, need to note that data provided is not distinctly seperated category wise(e.g.Action|Drama|Sci-Fi|Thriller) . Even their seperation does not give individually reliable data category wise. I hsten to add that I did have checked this although have not included the exercise here. $ Considering the category wise rating pattern we , perhaps, would do well to consider this feature as another influencing predictor.

3.0 Building and evaluating model

3.1 Loss function

The performance of our final model will be evaluated based on the Residual Mean Squared Error(RMSE). Simply defined , if $y_{u,i}$ as the rating given to a movie 'i'by user 'u' and denote our prediction with $\hat{y}_{u,i}$, RMSE is then defined by the formula.

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

with N being the number of user/movie combinations and the sum occurring over all these combinations. Let's write a function that computes the RMSE for vectors of ratings and their corresponding predictors:

3.2 Train and Test sets

First task is to split the 'edx' set to 'train' and 'test' sets We are taking 'test' set as 10% of the 'edx'

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

Checking the sets

```
## Observations: 8,100,048
## Variables: 6
## $ userId     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,...
## $ movieId    <dbl> 122, 292, 316, 329, 355, 356, 362, 364, 370, 377, 420, 4...
## $ rating     <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,...
## $ genres     <chr> "Comedy|Romance", "Action|Drama|Sci-Fi|Thriller", "Actio...
## $ year_rated <dbl> 1996, 1996, 1996, 1996, 1996, 1996, 1996, 1996, 1996, 19...
## $ year       <dbl> 1992, 1995, 1994, 1994, 1994, 1994, 1994, 1994, 1994, 19...
```

```
## Observations: 900,007
## Variables: 6
## $ userId     <int> 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5,...
## $ movieId    <dbl> 185, 260, 590, 1049, 1210, 1148, 1552, 3684, 6539, 435, ...
## $ rating     <dbl> 5.0, 5.0, 5.0, 3.0, 4.0, 4.0, 2.0, 4.5, 5.0, 3.0, 3.0, 5...
## $ genres     <chr> "Action|Crime|Thriller", "Action|Adventure|Sci-Fi", "Adv...
## $ year_rated <dbl> 1996, 1997, 1997, 1997, 1997, 2005, 2005, 2006, 2005, 19...
## $ year       <dbl> 1995, 1977, 1990, 1996, 1983, 1993, 1997, 1989, 2003, 19...
```

3,3 Baseline model

In its simplest form this model is generated by considering the same rating for all the movies irrespective of the 'userId' and the 'movieId'.All the differences explained by random variation. The formula would look like this: $Y_{u,i} = \hat{\mu} + \varepsilon_{u,i}$ With $\hat{\mu}$ is the mean and $\varepsilon_{u,i}$ is the independent errors sampled from the same distribution centered at 0.

```
## [1] 3.512457
```

If we predict all the unknown ratings with $\hat{\mu}$ our RMSE will be as follows

```
## [1] 1.060056
```

Let us now prepare a data frame to record all the RMSEs here after for all our evaluations

method	RMSE
Baseline approach	1.060056

3.4 Movie effect model

The 'base_rmse' of 1.06 as seen above is by no means acceptable. Hence , we need to attempt to improve this 'RMSE' and as a first step we are trying to achieve this by accounting for the movie effect. The intuition that different movies are rated differently are confirmed by the data.The movie effect can be taken into account by taking the difference from mean rating as shown below.This effect is termed as bias and we will be calling this b_i. We shall now augment the previous model as shown in the following formula :

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

With $\hat{\mu}$ is the mean and $\varepsilon_{u,i}$ is the independent errors sampled from the same distribution centered at 0.The b_i is a measure for the user's bias for the movie i.

```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

```
## Warning in bind_rows_(x, .id): binding factor and character vector, coercing
## into character vector
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector, coercing
## into character vector
```

```
##              method      RMSE
## 1 Baseline approach 1.0600561
## 2     Movie Effect  0.9429666
```

method	RMSE
Baseline approach	1.0600561
Movie Effect	0.9429666

We have achieved some improvement of RMSE at 0.9429 over that of 'Baseline model'. But it is still from the target RMSE of 0.8649

3.5 User and Movie effect model

We shall now be trying to improve further our earlier RMSE by 'Movie effect model' incorporating the 'User + Movie' effect, which will be in the following form :

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

With $\hat{\mu}$ is the mean and $\varepsilon_{u,i}$ is the independent errors sampled from the same distribution centered at 0. The b_i is a measure for the user's bias for the movie i. The b_u is a measure for the user's rating behaviour u.

```
## [1] 0.8646859
```

```
##                method      RMSE
## 1   Baseline approach 1.0600561
## 2        Movie Effect  0.9429666
## 3 User & Movie effect 0.8646859
```

method	RMSE
Baseline approach	1.0600561
Movie Effect	0.9429666
User & Movie effect	0.8646859

It is encouraging that we have succeeded in improving the RMSE further to 0.8646 which is already better than the target of 0.8649.But,we are not there yet unitl we test it on the validation set.

However , we intend trying further to see if we can achieve further improvement with our tests on the test set.

3.6 User , Movie and Genre effect model

We shall now be trying to improve our earlier RMSE by 'User + Movie' effect incorporating the 'User + Movie + Genre' effect, which will be in the following form :

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

With $\hat{\mu}$ is the mean and $\varepsilon_{u,i}$ is the independent errors sampled from the same distribution centered at 0. The b_i is a measure for the user's bias for the movie i. The b_u is a measure for the user's rating behaviour u. The $b_{u,g}$ is a measure for the bias of an user u for the genre g.

```
##                      method      RMSE
## 1         Baseline approach 1.0600561
## 2              Movie Effect  0.9429666
## 3       User & Movie effect 0.8646859
## 4 Movie+User+Genre effectl 0.8643257
```

method	RMSE
Baseline approach	1.0600561
Movie Effect	0.9429666
User & Movie effect	0.8646859
Movie+User+Genre effectl	0.8643257

We have achieved some further improvement at 0.8643

Aithough this is our best achievment interms of RMSE so far we shall now be testing both our models 'user & movie effect' and the 'Movie+User+Genre effectl' on the validation set.

3.7 Models - testing on validation set.

```
## [1] 0.8658556
```

```
##                                    method      RMSE
## 1                       Baseline approach 1.0600561
## 2                            Movie Effect  0.9429666
## 3                     User & Movie effect 0.8646859
## 4                Movie+User+Genre effectl 0.8643257
## 5 User & Movie effect on validation set 0.8658556
```

method	RMSE
Baseline approach	1.0600561
Movie Effect	0.9429666
User & Movie effect	0.8646859
Movie+User+Genre effectl	0.8643257
User & Movie effect on validation set	0.8658556

It would seem the RMSE has declined to 0.8654 while testing on the unseen validation data.At this stage all we can assume that it is possible and we need to prod along further.

We shall now be testing our 'user_movie_genre_model' on the validation set

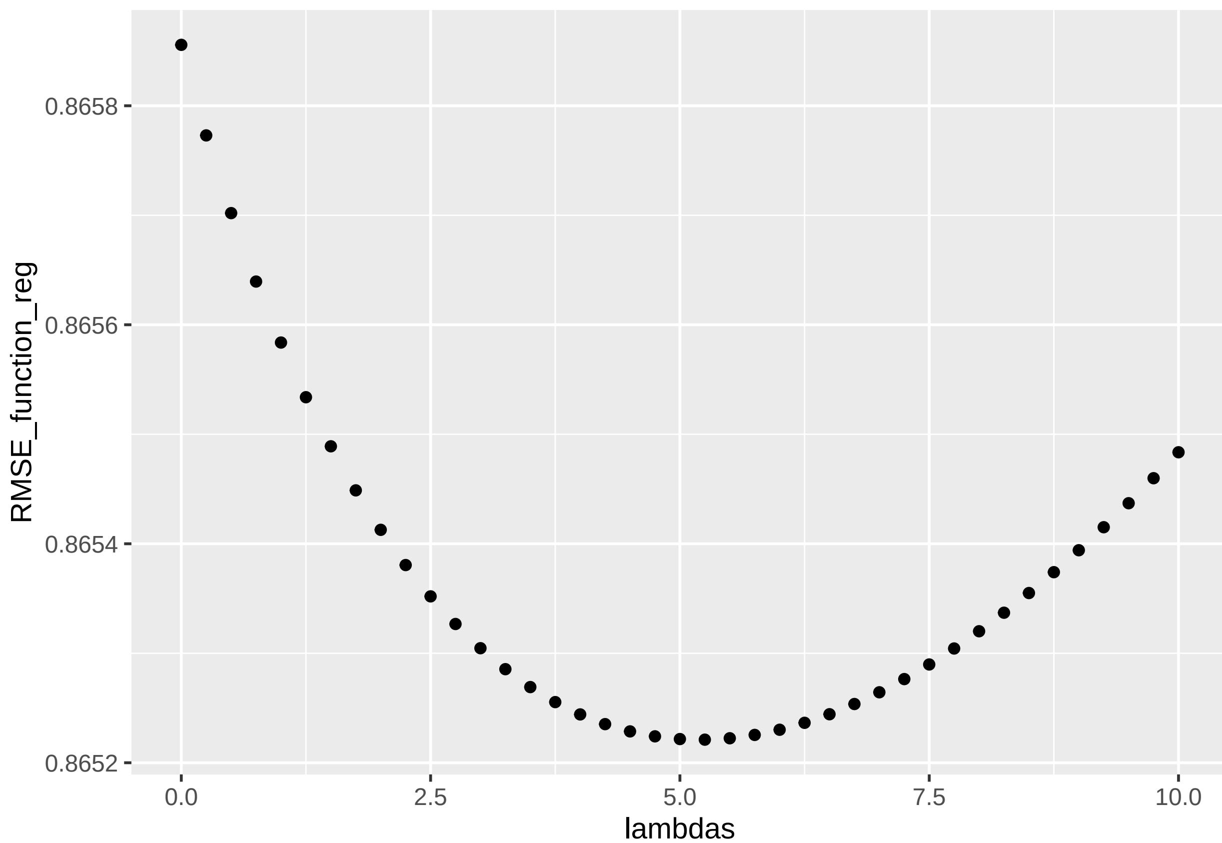
```
## [1] 0.8654518

##                                            method      RMSE
## 1                               Baseline approach 1.0600561
## 2                                    Movie Effect  0.9429666
## 3                             User & Movie effect 0.8646859
## 4                       Movie+User+Genre effectl 0.8643257
## 5            User & Movie effect on validation set 0.8658556
## 6 User & Movie & genre effect on validation set 0.8654518
```

method	RMSE
Baseline approach	1.0600561
Movie Effect	0.9429666
User & Movie effect	0.8646859
Movie+User+Genre effectl	0.8643257
User & Movie effect on validation set	0.8658556
User & Movie & genre effect on validation set	0.8654518

It would seem the RMSE has declined to 0.8658 while testing on the unseen validation data. As mentioned earlier at this stage all we can assume that it is possible and we need to prod along further.

3.8 Regularization based approach(Penalized RMSE)

It has come to light during our data exploration above, that some users have more actively participated in movie reviewing. At the same time there are some who have rated very few movies . Again there are instances where some movies are rated very few times . These are basically misleading noisy estimates . Further, RMSEs are sensitive to large errors. Large errors can increase our RMSE. So such issues necessitate putting a penalty term to give less importance to such effect.The regularisation method allows us to add a penalty λ to penalise movies with large estimates from small sample size.Let us call it Penalized RMSE approach Although we have accomplished a significant improvement over the 'Baseline model','Movie effect model' through the 'User and Movie effect model'and 'Movie+User+Genre effect model while testing these models on the test set RMSEs of these models showed a decine while testing on the unseen validation set. However We shall now be dealing with these models with the regularisation(Penalized) apprach.



Getting the lambda value that minimises the RMSE

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

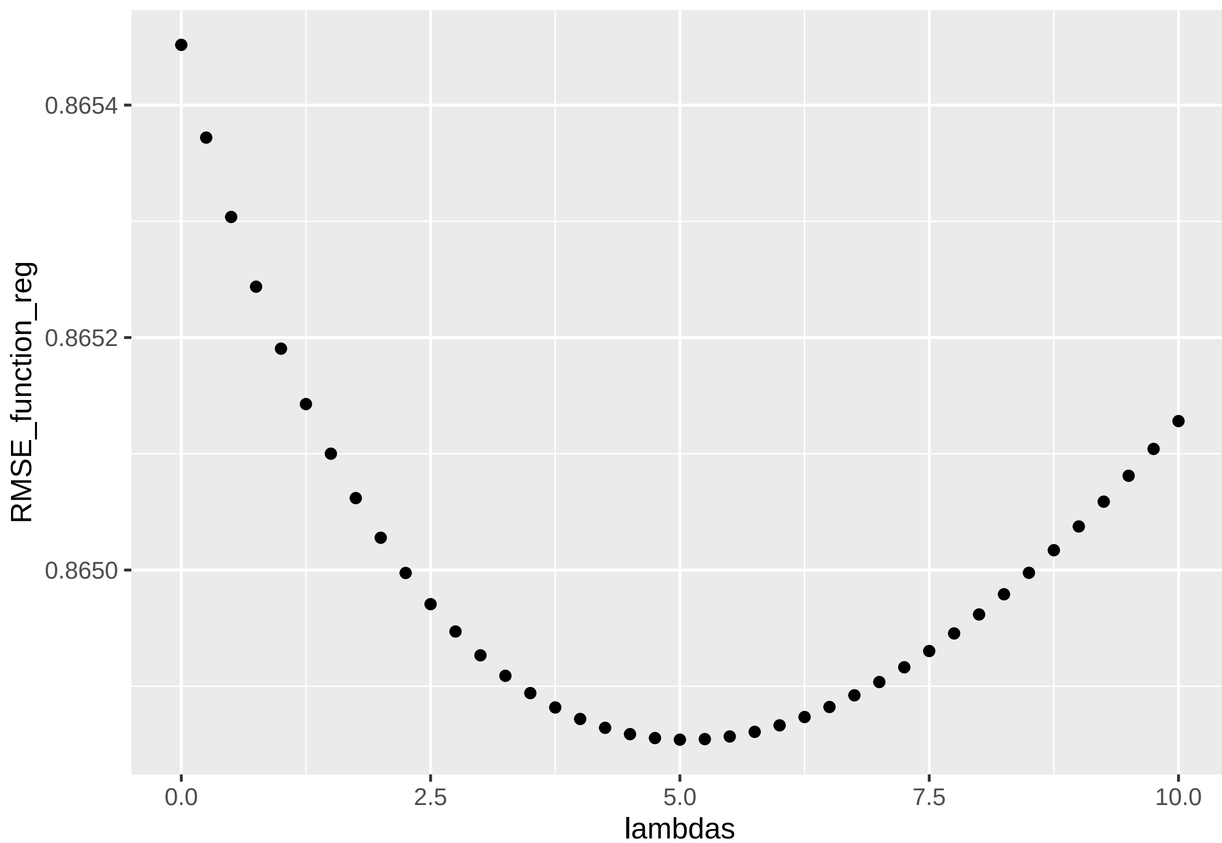
Now we shall be predicting the RMSE on the validation set with this mininised lambda value

```
## [1] 0.8652211
```

method	RMSE
Baseline approach	1.0600561
Movie Effect	0.9429666
User & Movie effect	0.8646859
Movie+User+Genre effectl	0.8643257
User & Movie effect on validation set	0.8658556
User & Movie & genre effect on validation set	0.8654518
Regularised User & Movie effect model on validation set	0.8652211

Although there has been some improvement from 0.8658 to 0.86522 this is still some distance away from the target RMSE of 0.8649

Now we shall be trying the regularised 'user_movie_genre model' on the validation set



Getting the lambda value that minimises the RMSE

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

Now we shall be predicting the RMSE on the validation set with this mininised lambda value

```
## [1] 0.8648541
```

method	RMSE
Baseline approach	1.0600561
Movie Effect	0.9429666
User & Movie effect	0.8646859
Movie+User+Genre effectl	0.8643257
User & Movie effect on validation set	0.8658556
User & Movie & genre effect on validation set	0.8654518
Regularised User & Movie effect model on validation set	0.8652211
Regularised User & Movie & genre effect model on validation set	0.8648541

We seem to have finally achieved an RMSE of 0.86485 .

4.0 Conclusion

The primary objective of this project was to predict user movie ratings based on the other user's ratings using a 10M version of the MovieLens dataset. The exploration of the dataset and the key revelations of their visualization has lead us to believe that the features strongly suggesting an influence on prediction would be the movie(movieId),the user(userId) and the genre of the movie(genres).Accordingly algorithms with different combinations of these features were trained and tested to evaluate the accuracy of the RMSE(prediction).

Results and performance of each of those models have been individually tabulated and discussed under the relevant sections of the detailed report. Finally, we have achieved the highest RMSE accuracy of 0.86482 with a lambda of 5 on the validation set with the Regularised(Penalized) Root Mean Square Error approach. Talking of the future work, we have good scope of utilizing Matrix factorization in the context of this movie recommendation system.Our final model leaves out an important source of variation related to the fact that groups of movies have similar rating patterns and groups of users have similar rating patterns as well. We could also train different models, recommender engines and ensemble methods in our endeavour to better our accuracy .Considering (in my perception though) that, limited scope of this project does not call for further exhaustive work than what has been done here.