MOVIELENS RECOMMENDER SYSTEM

CAPSTONE PROJECT - 1

Submitted by : LEEBA ANN VARGHESE

TABLE OF CONTENTS

1. Introduction

Recommendation systems are one of the major applications of Machine learning in Data science. They are widely used in large companies such as Netflix, YouTube, Amazon, Reddit etc. to predict user preferences, and return it to the users. This project is a part of the course -'Professional Certificate in Data science' launched by Harvard University and Edx. In this project, we use the Movielens dataset - the 10M Movielens dataset which was generated by the Grouplens research lab to develop the best possible algorithm that predicts user ratings for movies, using the entities available in the Movielens dataset and the knowledge acquired from the entire course. The whole dataset will be split into two- the Training set and Validation set .Different models will be tested on the training set .Finally, the approach with the best accuracy will be applied on the validation set.

2. Project Objective

The objective of this project is to build a movie recommendation system using the Movielens dataset. The accuracy is measured by the metrics RMSE-Root Mean Squared Error. RMSE measures how far the predicted values deviate from the true values. Our aim is to develop a model with the least possible RMSE.

3. Data Cleanup - Preparation of Data

We will first extract the data from the sources and do some data wrangling techniques to prepare the data to work on.

3.1 Sources

To prepare the data, Movielens 10M dataset has been extracted from the below mentioned web resources.

i. Movie ratings

https://grouplens.org/datasets/movielens/10m/

We got the Userid, Movie-id, Rating and Timestamp from the above source.

ii. Movies list

http://files.grouplens.org/datasets/movielens/ml-10m.zip

We got the Movie-id, Movie-title and the genre from the above source.

Using the above two sources, we have prepared the data set for Movies and Movie Ratings.

3.2 Preparing the Movielens dataset.

Using the join function in the dplyr library, we have created the Movielens dataset by combining the data from Movie ratings and Movies list dataset.

Structure of Movielens dataset:

- MovieId : Unique id of the movie
- Title: The title of the movie
- Year: The year of movie release
- Genres: The genre of the movie
- UserId: Unique user id
- Rating: Movie rating provided by the user (0.5 to 5 stars)
- Timestamp: Represents the time in seconds since midnight UTC.

```
> str(movielens)
'data.frame':   100004 obs. of  7 variables:
 $ movieId  : int  31 1029 1061 1129 1172 1263 1287 1293 1339 1343 ...
 $ title    : chr  "Dangerous Minds" "Dumbo" "Sleepers" "Escape from New York"
 ...
 $ year     : int  1995 1941 1996 1981 1989 1978 1959 1982 1992 1991 ...
 $ genres   : Factor w/ 901 levels "(no genres listed)",..: 762 510 899 120 762
 836 81 762 844 899 ...
 $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ rating   : num  2.5 3 3 2 4 2 2 2 3.5 2 ...
 $ timestamp: int  1260759144 1260759179 1260759182 1260759185 1260759205 12607
59151 1260759187 1260759148 1260759125 1260759131 ...
>
```

4. Data Modeling - Splitting the data

4.1 Preparing the Validation set.

The validation set (10M dataset) is split into two sets- edx and validation using the createDataPartition () function in the caret package. Validation set will be 10% of the Movielens dataset and will be reserved to do the final hold out test.

4.2 Splitting the edx dataset

The edx set is further split into two – the test set and training set. Our algorithm is built on the edx dataset.

```
> str(edx)
Classes 'data.table' and '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 838983707 838984596
 ...
 $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
 $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|Adventure|Sci-Fi"
```

After splitting,below are the dimensions of test set and training.

```
> dim(edx)
[1] 9000055       6
> dim(test_set)
[1] 899990      6
> dim(train_set)
[1] 8100065       6
```

5. Data Analysis and Data Visualization

The edx dataset has 9,000,055 observations and 6 columns.

Below screenshot shows the first 6 rows of this dataset. This helps us to get a better picture of the data.

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

5.1 Distinct number of movies, users and genres.

Using the summarize function, we have summarized the distinct number of users, movies and genres in this dataset.

```
 n_users n_movies n_genres
   69878    10677       797
```

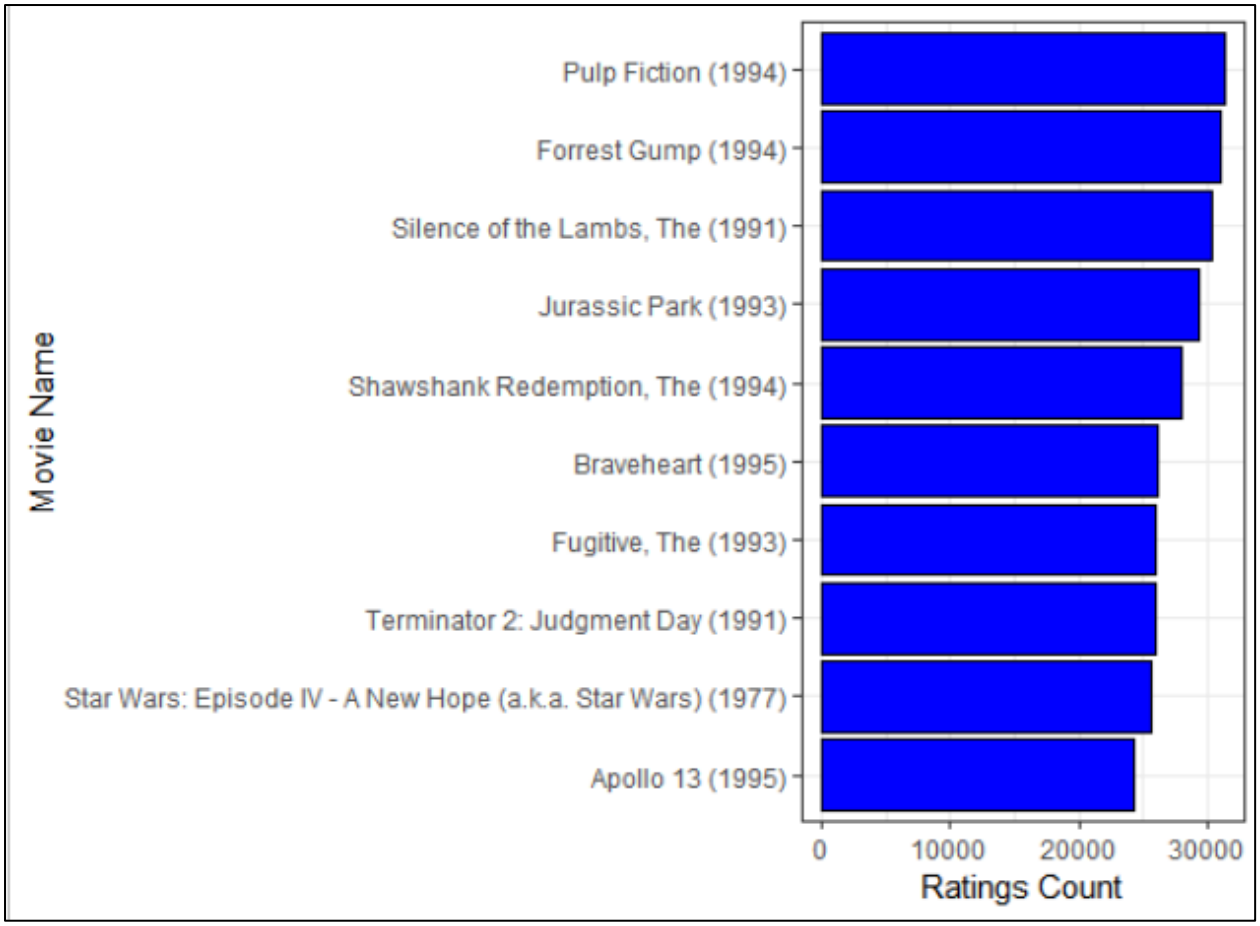
5.2 Number of movies in each genre

We have summarized the number of movies by genres as below. We see that there are 4 genres and the most number of movies are in the genre 'Drama'.

```
   Drama   Comedy Thriller  Romance
 3910127  3540930  2325899  1712100
```

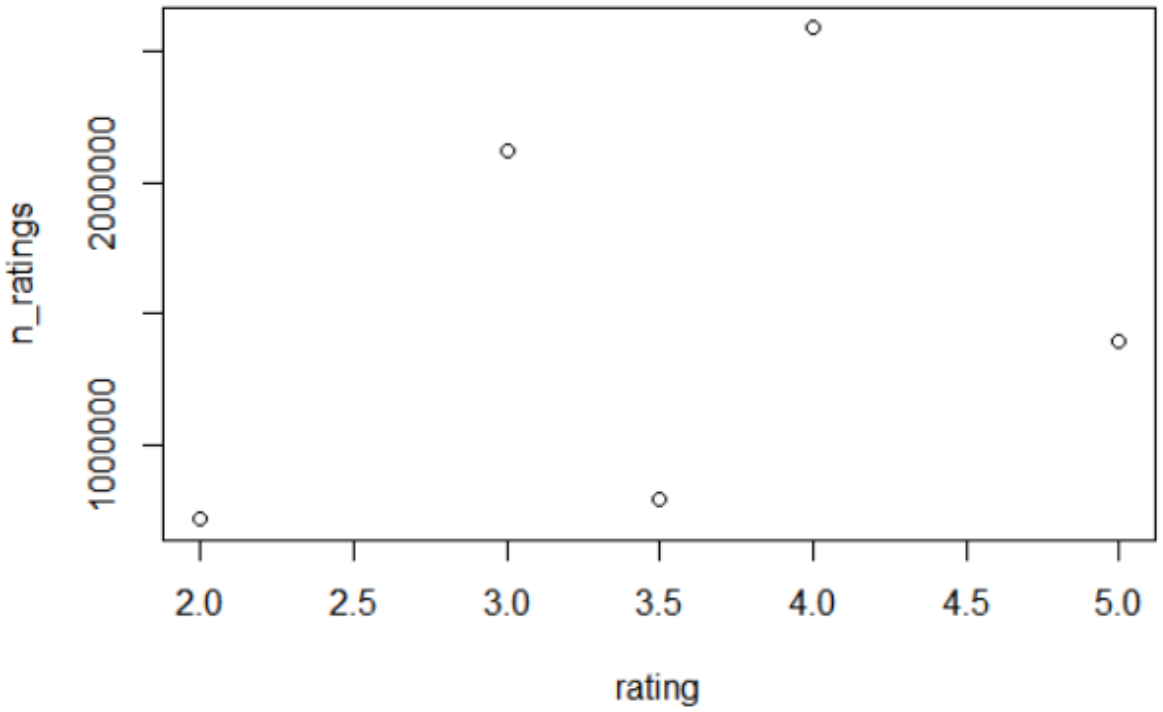
5.3 Top 10 most rated movies in the dataset.

We see that the most popular movies have received approximately 20,000 to 30,000 reviews.



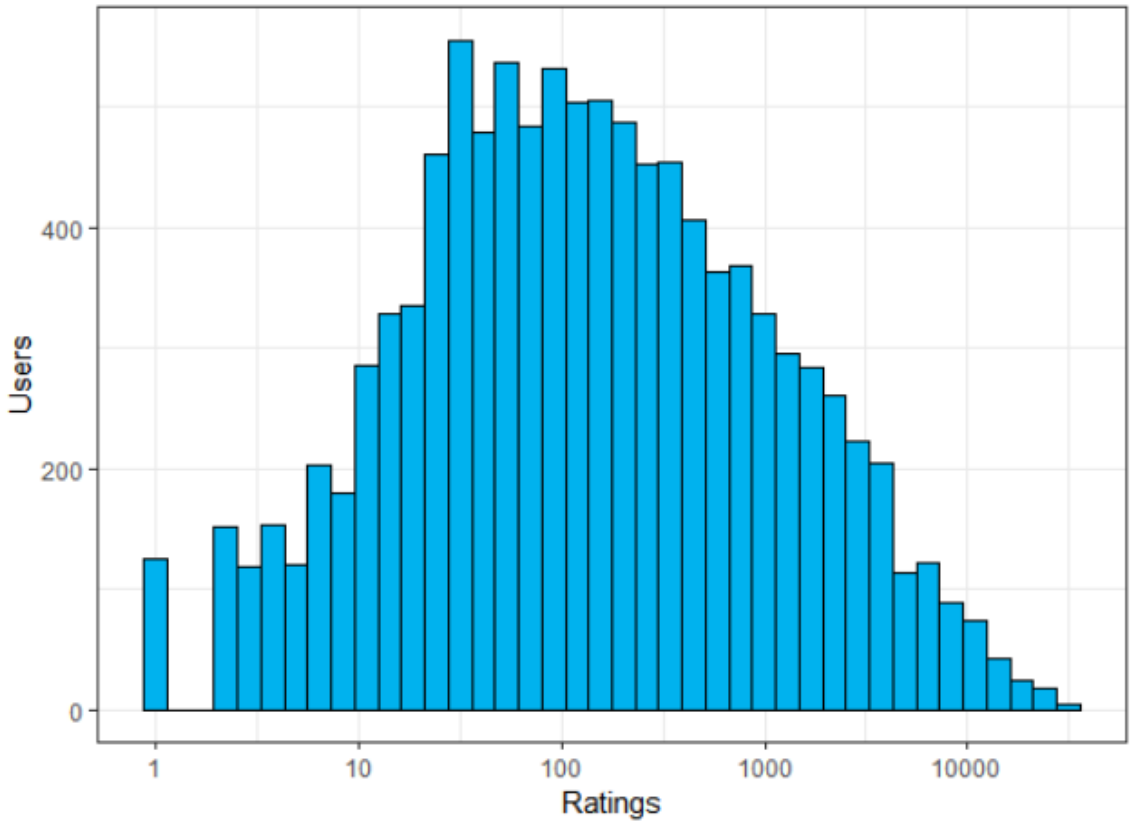
5.4 Plot of the most common ratings

Below is a plot of the most common ratings. We see that the most commonly used rating is 4 .



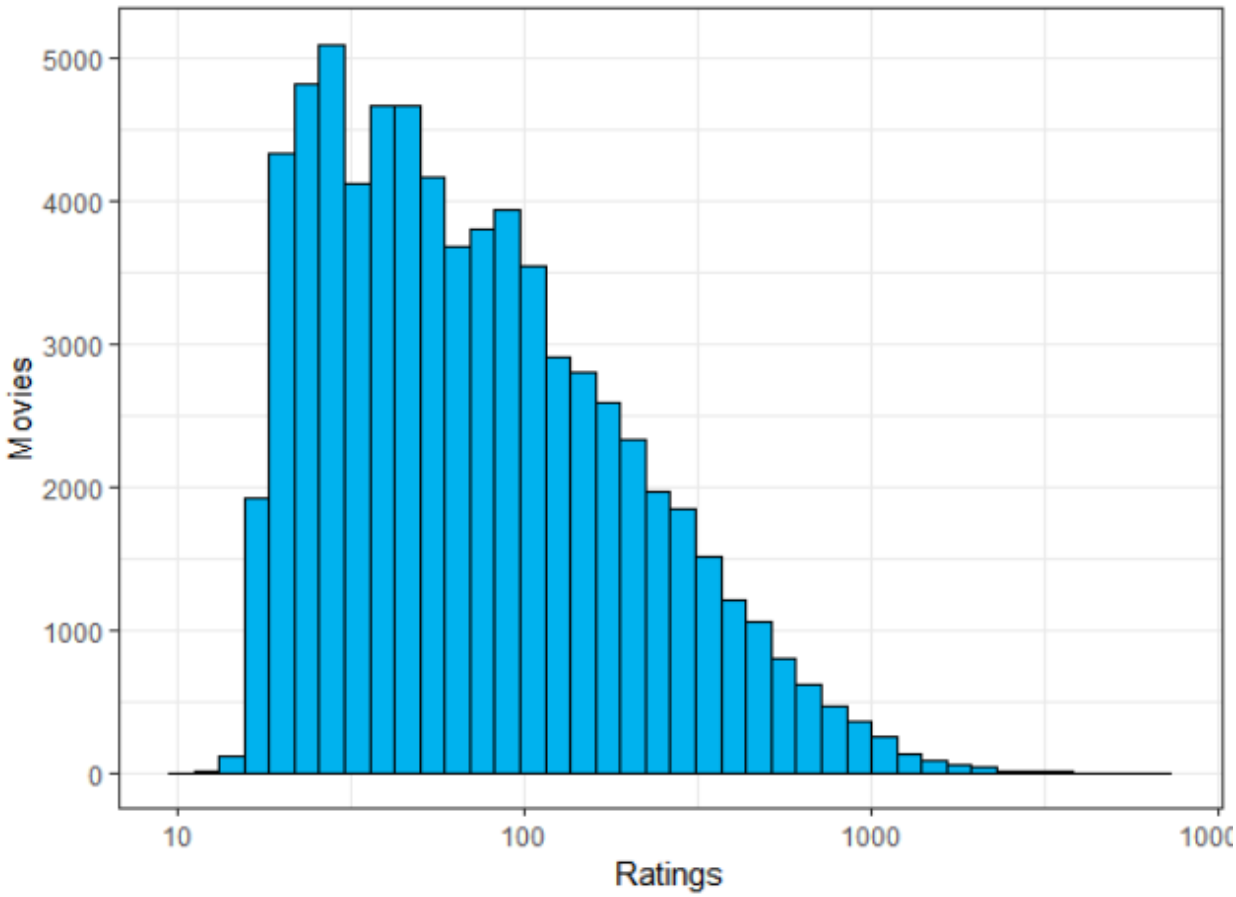
5.5 Number of rating versus movies.

We see that most movies have below 1000 reviews, while some have more than 10000 reviews.



5.6 Number of rating versus users

We see that most users wrote less than 100 reviews, while some wrote more than 1000.It shows a right skew in the distribution.



5.7 Top 10 most rated genres

Below are the most popular genres and we see that they are mostly rated above 4.

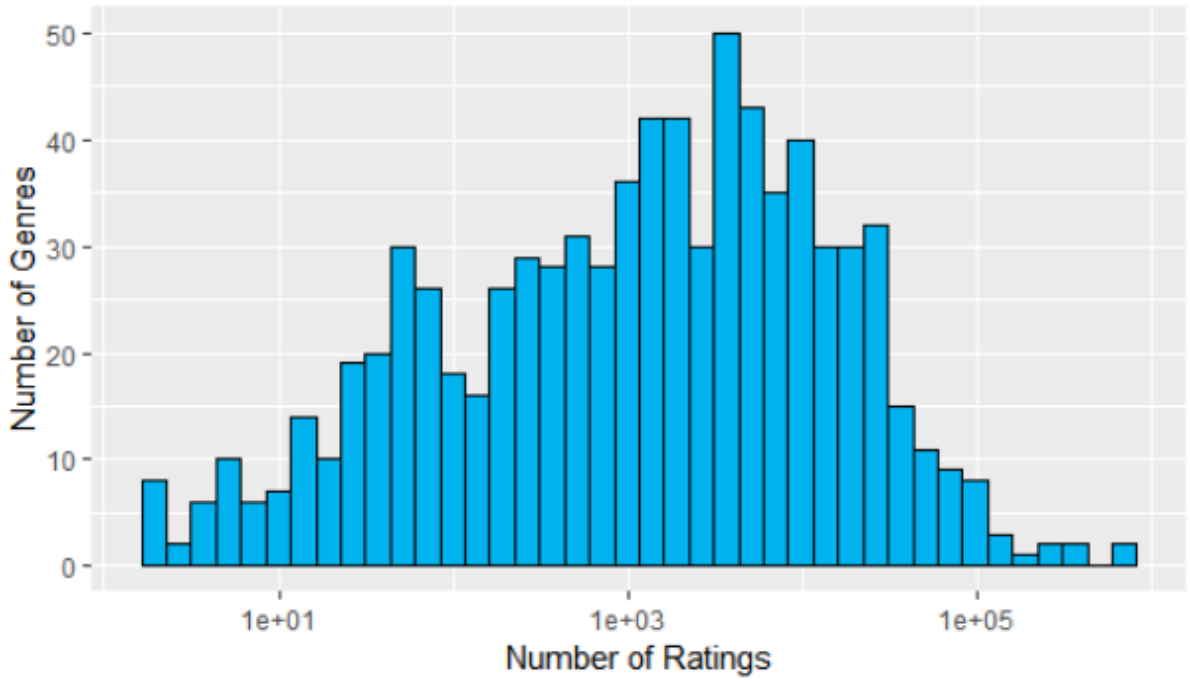
	genres	avg_rating	num_reviews
	<chr>	<dbl>	<int>
1	Animation|IMAX|Sci-Fi	4.71	7
2	Drama|Film-Noir|Romance	4.30	2989
3	Action|Crime|Drama|IMAX	4.30	2353
4	Animation|Children|Comedy|Crime	4.28	7167
5	Film-Noir|Mystery	4.24	5988
6	Crime|Film-Noir|Mystery	4.22	4029
7	Film-Noir|Romance|Thriller	4.22	2453
8	Crime|Film-Noir|Thriller	4.21	4844
9	Crime|Mystery|Thriller	4.20	26892
10	Action|Adventure|Comedy|Fantasy|Romance	4.20	14809

5.8 Least 10 rated genres

Below is the list of least rated genres.

	genres	avg_rating	num_reviews
	<chr>	<dbl>	<int>
1	Documentary|Horror	1.45	619
2	Action|Animation|Comedy|Horror	1.5	2
3	Action|Horror|Mystery|Thriller	1.61	327
4	Comedy|Film-Noir|Thriller	1.64	21
5	Action|Drama|Horror|Sci-Fi	1.75	4
6	Adventure|Drama|Horror|Sci-Fi|Thriller	1.75	217
7	Action|Adventure|Drama|Fantasy|Sci-Fi	1.90	57
8	Action|Children|Comedy	1.91	518
9	Action|Adventure|Children	1.92	824
10	Adventure|Animation|Children|Fantasy|Sci-Fi	1.92	691

5.9 Number of ratings versus genre



From the above histogram, we see that some genres have received more ratings and some are rated less as we saw in the figures earlier.

6. Evaluation Metric

The evaluation metric used in this project will be the RMSE- Root mean square error. Several models will be used and the assessment will be based on the least value of RMSE obtained. It measures the error of a model in predicting quantitative data by measuring the difference between actual and predicted values. RMSE is defined as follows:

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

Where:

N- number of observations, here it is the number of user-movie combinations.

$\hat{y}_{u,i}$ is the predicted value, here it is the predicted rating of movie i by user u

$y_{u,i}$ – is the actual rating of movie i by user u

7. Data Modeling Approaches- Building the Models

Now that we have analyzed and visualized the data, we will try different models and see the accuracy by calculating RMSE.All the models are applied on the training set. Once we reach a minimal RMSE, we shall apply the model on the validation set to get the final RMSE.

7.1 Model 1: Using the Mean of ratings

In this model, we have predicted ratings as the mean of ratings in the training dataset.

```
> result_model1
[1] 1.060054
```

The RMSE obtained using this model is 1.060054.The value is greater than 1 and it lacks accuracy.

7.2 Model 2: Using the Median of ratings

In this model, we have predicted ratings as the median of ratings in the training dataset.

```
> result_model2
[1] 1.166756
```

The RMSE obtained using this model is 1.166756,which is slightly higher than while using the mean. Hence, we will use mean of ratings in our further models.

7.3 Model 3: Using Mean and Movie bias

In this model, along with the mean we have also considered the average rating of each individual movie, as we have seen that more popular movies have received higher ratings.

$Y_{u,i}=\mu+mb_i+\epsilon_{u,i}$

```
> result_model3
[1] 0.9429615
```

The RMSE obtained using this model is 0.9429615, which is much better compared to the previous model.

7.4 Model 4: Using Mean, Movie bias and User bias

In this model, along with the mean and average movie rating, we have also considered the average user rating. We see the RMSE has reduced to 0.8646843.

$Y_{u,i}=\mu+mb_i+ ub_i +\epsilon_{u,i}$

```
> result_model4
[1] 0.8646843
```

7.5 Model 5 : Using Mean ,Movie bias , User bias and Movie Genre

In this model, along with the mean, average of movie rating and user rating, I have also considered the effect of movie genre.

$Y_{u,i}=\mu+mb_i+ ub_i + gb_i +\epsilon_{u,i}$

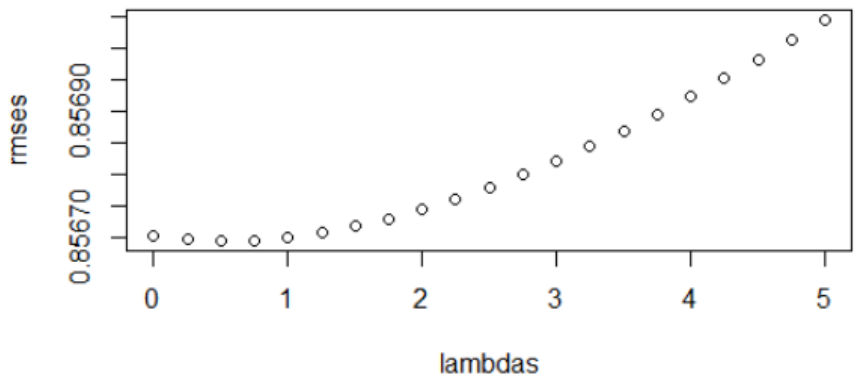
```
> result_model5
[1] 0.8643241
```

We see that there is not much of a considerable difference in the RMSE.

7.6 Model 6: Regularization by calculating the least lambda.

In this model, we will try to implement the concept of regularization into the previous model.

We have calculated and arrived at the least regularization factor- min_lambda 0.5.The plot of lambda and the arrived accuracy is shown below.



The minimum value of lambda is 0.5.

```
> min_lambda
[1] 0.5
```

The RMSE with this lambda is 0.8566952

```
> min_rmse
[1] 0.8566952
```

The RMSE is much lower than the previous model.

7.7 Approach using Matrix Factorization

In this approach, we process our data as a large and sparse matrix and then decompose into two smaller dimensional matrices with less sparsity. We will utilize the recosystem package in this model.

Below steps are followed in this method:

1. We created a model object r by calling the Reco() function.
2. We used the tune() method to select the best tuning parameter.
3. Using the train() method, we trained the model with parameter as the result from step 2.
4. Using predict() method, we have calculated the predicted values.

The RMSE obtained with this model is 0.784071 which is so far the best. Hence we will use apply this model on the validation set and see the result.

```
> result_model7
[1] 0.784071
```

8. Application to Validation Set

We will apply the Model 7 - Approach using Matrix Factorization with recosystem to the validation set and see the accuracy.

The final RMSE when applied to the validation set is shown below:

```
> final_rmse
[1] 0.7805994
```

9. Conclusion

We have tried several models taught in this course series to provide a best movie rating prediction with the best accuracy and least RMSE. Below is the summary of all the models we have tried and the RMSEs. We see that the best approach is Matrix Factorization using recosystem, although it is time consuming and utilization of RAM is more compared to other models.

Summary of models used and the RMSE:

#	Model	RMSE
1	Using Mean of ratings	1.060054
2	Using Median of ratings	1.66756
3	Using Mean and Movie bias	0.9429615
4	Using Mean , Movie bias and User bias	0.8646843
5	Using Mean ,Movie bias , User bias and Movie Genre	0.8643241
6	Approach using the least regularization factor(lambda)	0.8566952
7	Approach using Matrix Factorization with recosystem	0.784791
Final	Applied on Validation set	0.7805994

10. References

https://rpubs.com/faisalcep/dsMachineLearning

https://cran.r-project.org/web/packages/recosystem/vignettes/introduction.html
Simple guide to confusion matrix terminology (dataschool.io)

Confusion Matrix in R | A Complete Guide | DigitalOcean

http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/153-penalized-regression-essentials-ridge-lasso-elastic-net/

x----------x