PH125.9x Data Science: Capstone MovieLens

Krithika Ganeshkumar

2023-12-13

Contents

1 Executive Summary

A movie recommendation system is a tool to suggest movies to users based on the ratings provided by the user. The tool employs various machine learning techniques to process and analyze data including user provided data. It generates personalized movie recommendations based on the user ratings. Based on a user rating, a one star rating means it is not a good movie whereas a five star rating means it is a great movie. Based on the ratings by the user, the recommendation system such as one used by Netflix, predicts how many stars a user will give to a movie. The famous Netflix challenge back in 2006, offered $1M USD for the best model to predict user ratings based on previous ratings of the user.

The goal of this project is to create a movie recommendation system using the MovieLens dataset containing 10 million movie ratings and by employing various tools learned throughout the courses in the PH125.x data science series. The MovieLens data and the code to generate the datasets are already provided to begin with.The objective is to determine an model that predicts ratings with a RMSE (Root Mean Square Error) less than 0.86490 versus the actual ratings in the final hold out set.

In this report, several key steps are followed to achieve the objective by exploratory analysis of data including data visualization, and train and test the model using the datasets provided.The pre-processing techniques are applied to both edx and final holdout sets. The model is trained on the train set (edx_train_set) which is 90% of the edx dataset. The test set (edx_test_set) is derived from the 10% of the edx dataset. The final hold out set (final_holdout_test) is derived 10% from the original MovieLens data. The RMSE results are analyzed at each step and eventually the final model is validated by determining the RMSE on the final hold out set.Please note that the final model is derived by training and testing on the edx dataset which consists of 90% of the original MovieLens data. The final hold out set consists of 10% of the original MovieLens data and is used **only** for the final validation against the final model to determine the RMSE and not for training or regularization.

2 Analysis and Methods

2.1 Dataset analysis

The provided edx dataset consists of 9000055 rows and 6 columns with ratings ranging from 0.5 to 5 with 0.5 increments. There are 10677 unique movies, 69878 unique users and 797 unique genres.

Table 1: Top 6 rows of edx raw dataset

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

```
str(edx)
```

```
## 'data.frame':    9000055 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : int  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
```

It is evident, the dataset warrants some cleaning prior to exploratory analysis. The title column includes both the title and the year a movie was released. Similarly, most of the movies appear to belong to multiple genres. Prior to further data exploration, let's check for NA values, if any, present in the dataset. It is apparent that there are no NA values since the output of anyNA on edx dataset yields FALSE

2.2 Dataset cleaning

In this section, we are going to "clean" the dataset. After a glimpse of the edx dataset, it is evident that the title column includes both the title and the year a movie was released. Also, the timestamp column should be converted into a human readable format. The converted timestamp serves as the year a user rated the movie.

First, we extract the year of the movie release from the title column from the title column. Second we process the timestamp data and convert it into a human readable format. Please note, these pre-processing techniques should be applied to the final holdout set as well.

Tables below display the new edx dataset and final holdout set after cleaning and formatting.

Table 2: Top 6 rows of edx dataset after pre-processing

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

Table 3: Top 6 rows of final hold out set dataset after pre-processing

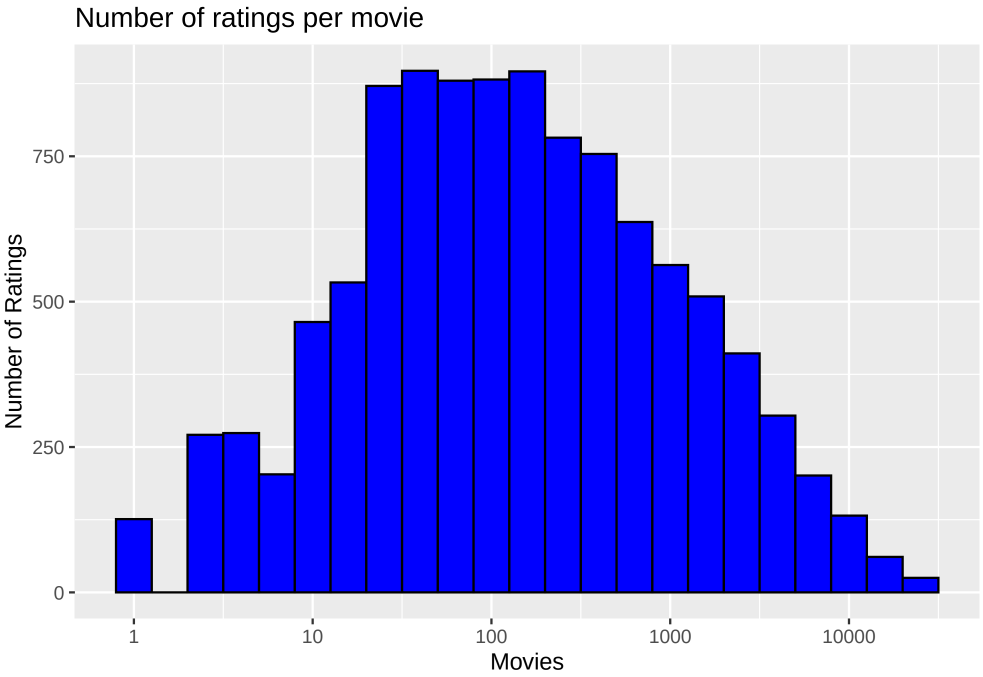
userId	movieId	rating	timestamp	title	genres	release__year	review__year
1	231	5	838983392	Dumb & Dumber (1994)	Comedy	1994	1996
1	480	5	838983653	Jurassic Park (1993)	Action|Adventure|Sci-Fi|Thriller	1993	1996
1	586	5	838984068	Home Alone (1990)	Children|Comedy	1990	1996
2	151	3	868246450	Rob Roy (1995)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

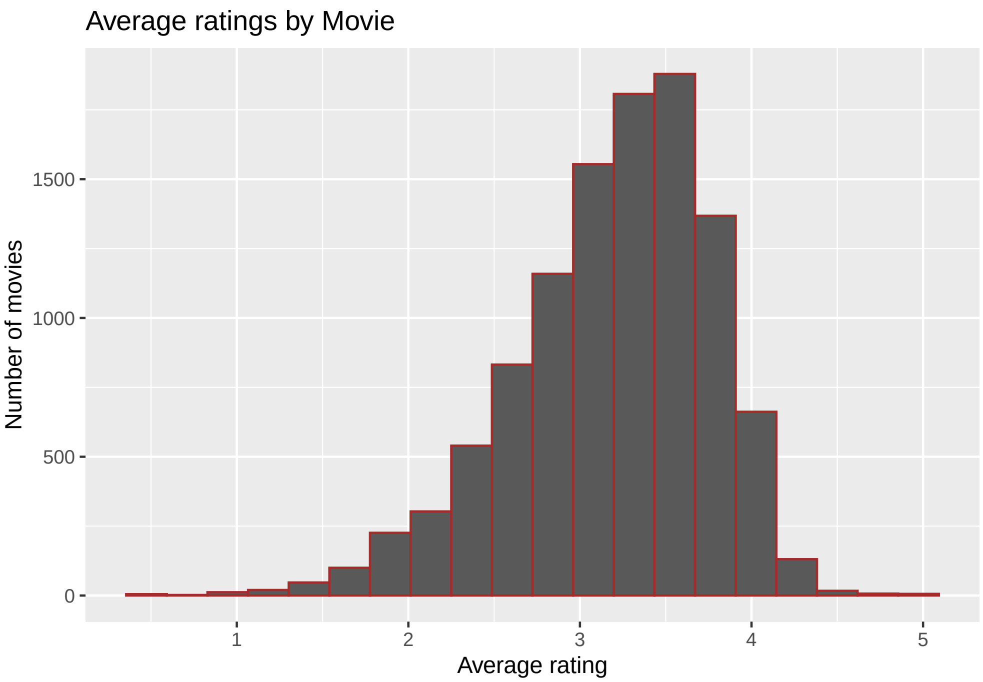
2.3 Dataset exploration

In this section, we will visually explore the effects of movie, users, genres, the year a movie was reviewed and the year a movie was released on the ratings. The overall average rating is 3.5124652 and it is also observed the half-star ratings are fewer than full-star ratings. 0.5 is the minimum rating and 5 is the maximum rating a movie received.

Movie effects on the ratings

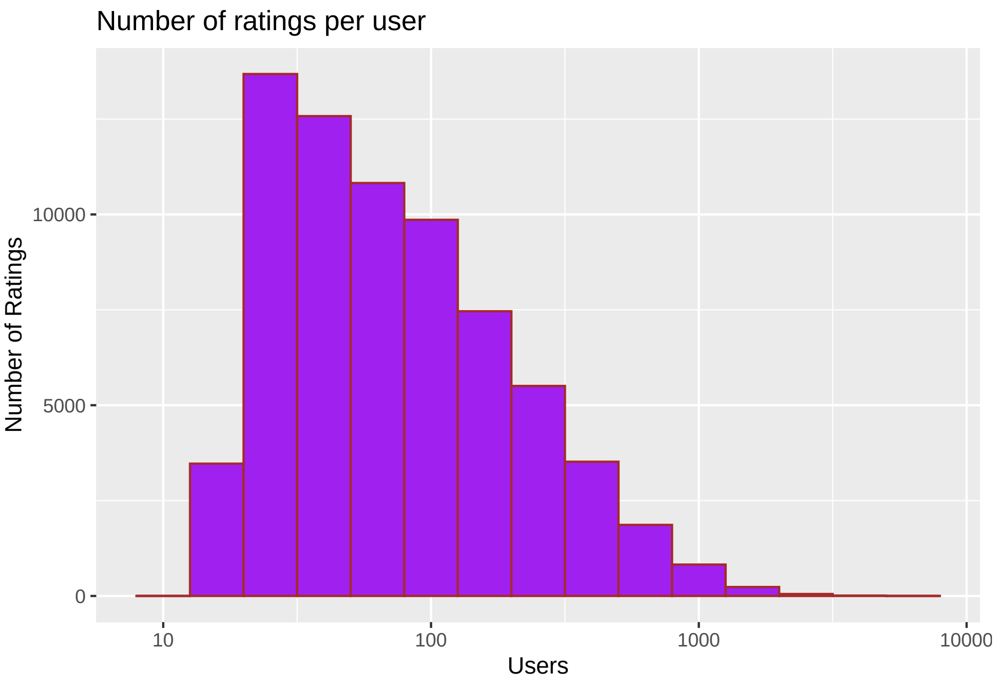
Let's explore the number of ratings by movie. Some movies received more ratings than the others (see figures below). Evidently popular movies were frequently rated than their least popular counterparts. This implies there could be a potential bias that could potentially impact the movie recommendation system.

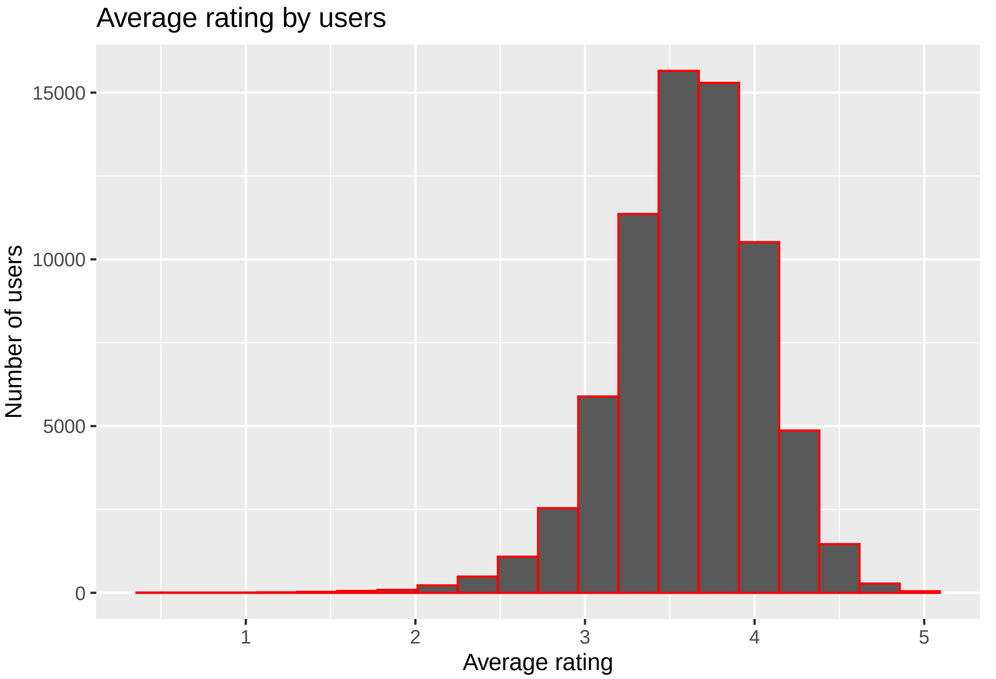




User effects on the ratings

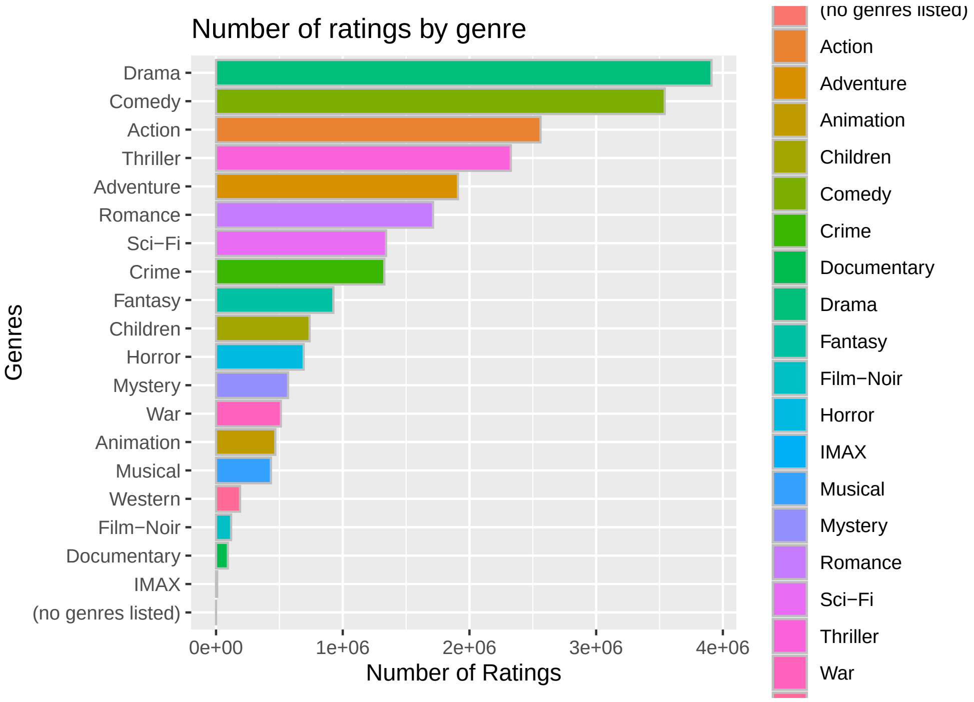
Let's explore the number of ratings by user. Not all users rated a movie. This is evident from the following chart.This implies user effects on ratings poses a potential bias that could potentially impact the movie recommendation system.

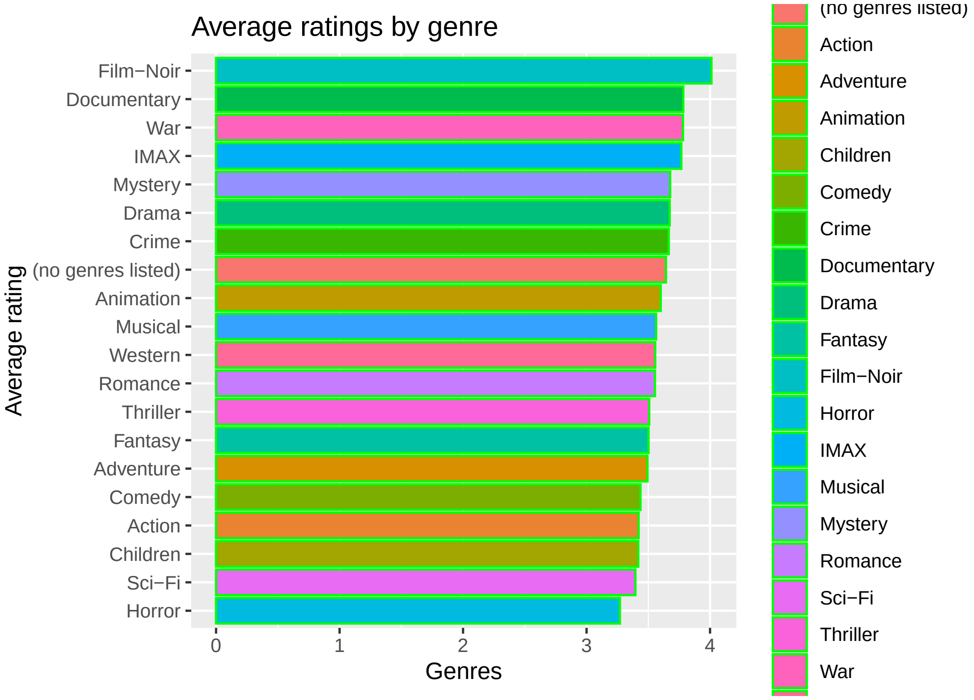

Number of ratings per user
Number of Ratings
10000
5000
0
10
100
1000
10000
Users


Average rating by users
Number of users
15000
10000
5000
0
1
2
3
4
5
Average rating

Genre effects on the ratings

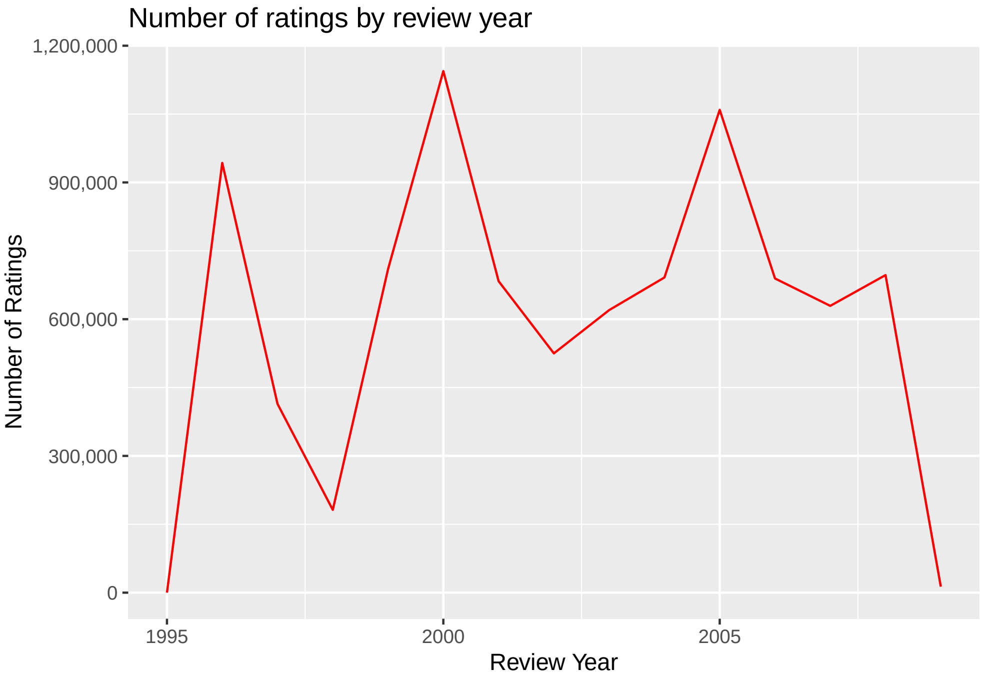
Let's explore the number of ratings by genre. Some genres are more popular than the others. First, let's categorize the genres for each movie since each movie belong to more than one genre.As shown in the chart and table below, the ratings vary vastly depending on the genre.This leads to a potential bias that needs to be considered when training the dataset.





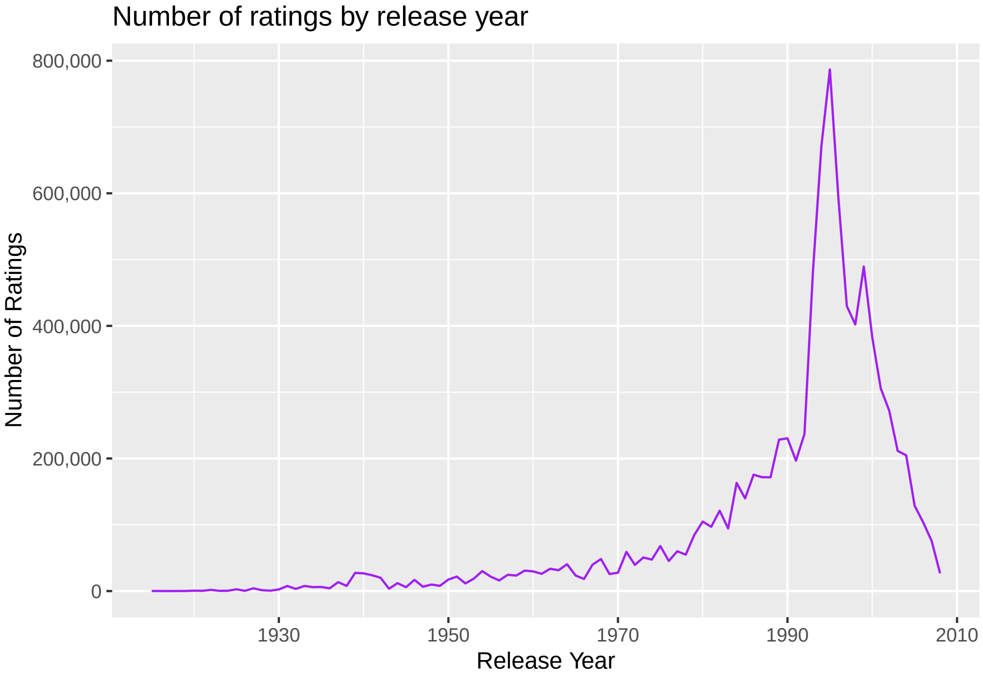
Review date effects on the ratings

Let's explore the number of ratings by the year a movie was rated. As shown in the chart below, rating count and frequency varies by the year.



Release date effects on the ratings

Let's explore the number of ratings by the year a movie was released. As displayed in the chart below, the year a movie was released also impacts the rating. Movies released in 1990s received more ratings than rest of the decades included in the dataset. Nevertheless, we will consider this factor as well and will quantify the impact.



2.4 Methods

In this section, we will split the MovieLens dataset into edx for training and testing purposes. The code to split MovieLens dataset to edx and final hold out set has already been provided. The edx dataset is further split into train and test sets as mentioned in project instructions.

Dataset	Split
MovieLens	N/A
edx	90% of MovieLens
final_hold_out	10% of MovieLens
train_set	90% of edx
test_set	10% of edx

We employ various methods to train our models to determine the final model that would be validated against the final hold out set provided earlier. As described in the course book, the general approach to determine the ideal model is to define a loss function. The loss function is defined as $(\hat{y} - y)^2$. However, Root Mean Squared Error (RMSE) is often used to report since it is easier to perform mathematical computations.

RMSE is defined as,

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

Programatically, the function RMSE is defined as,

The goal of this project is to identify the model that would yield a **RMSE less than 0.86490**. We will train and test effects of movie, user, genre, review year and release year on the ratings to compute the RMSE. Additionally, if target RMSE is not achieved with those factors, we will employ regularization techniques to achieve the target RMSE.

2.4.1 Naive RMSE

This is the most simplest form of RMSE. It predicts the mean. Naive RMSE is,

```
## [1] 1.061135
```

2.4.2 Movie effects on RMSE

Let's determine the effect of movies on RMSE. Since movies get rated differently, the movie bias should contribute significantly in computing the RMSE. The formula is,

$$y_{u,i} = \mu + bm + \epsilon_{u,i}$$

where y_{u,i} is the predicted rating, μ is the average rating, bm is the movie bias and $\epsilon_{u,i}$ is independent errors.

```
## [1] 0.9441568
```

Movie effects model yielded a RMSE of 0.9441568. This is a significant improvement over Naive model which yielded 1.061135. However, it still does not meet our target RMSE.

2.4.3 User effects on RMSE

Let's determine the effect of user on RMSE. Since each user rates a movie differently and not all users rate a movie, the user bias contributes to compute the RMSE.

```
## [1] 0.9795916
```

User effects model yielded a RMSE of 0.9795916. This is lower than Naive RMSE of 1.061135 however it is higher than movie RMSE of 0.9441568. Nevertheless, it still does not meet our target RMSE.

2.4.4 Genre effects on RMSE

Similar to independent movie and user effects on RMSE, let's determine genre bias on RMSE.

```
## [1] 0.9441568
```

Genre effects model yielded a RMSE of 0.9441568. This is a significant improvement over previous models including naive and user models. However, it still does not meet our target RMSE of < 0.86490.

In the following section, we will combine the biases to determine a model which yields less than target RMSE.

2.4.5 Adding user effects to Movie RMSE

Let's determine the effects of user bias on RMSE in addition to the movie bias. The technique is similar to movie bias as shown below.

```
## [1] 0.8659736
```

Adding user bias to the existing movie bias yielded RMSE of 0.8659736 which is an improvement from earlier RMSE values computed.

2.4.6 Adding Genre effects to Movie and User RMSE

Let's determine the effects of genre bias on the existing movie and user model.

```
## [1] 0.8656018
```

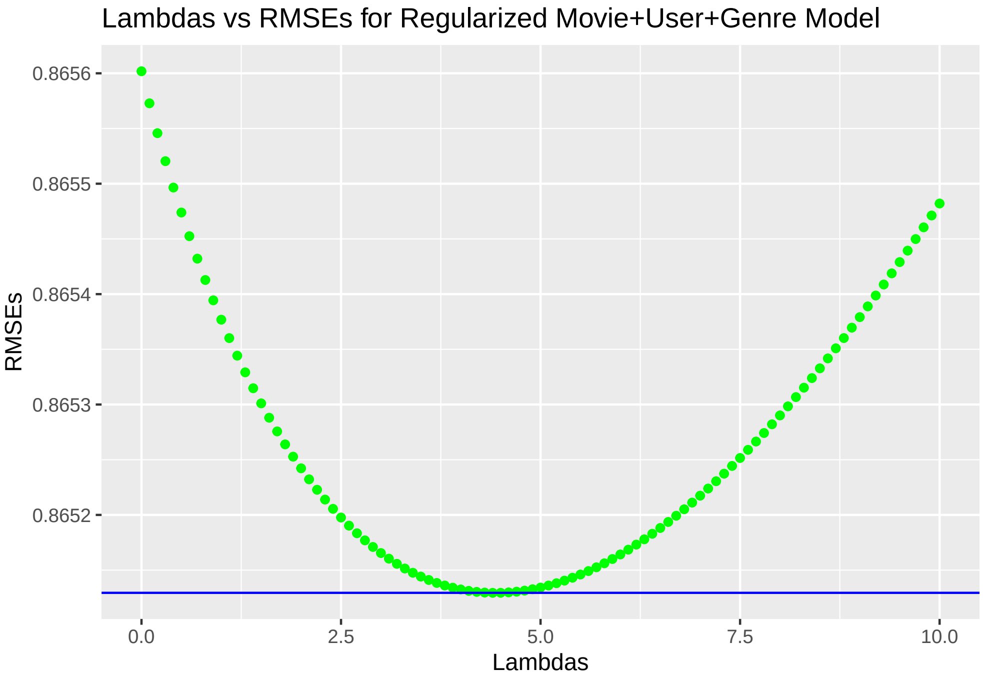
The movie+user+genre based model yielded RMSE of 0.8656018 which meets the target RMSE of < 0.86490. However, we could improvise on this by adding regularization method as described in the following section.

2.4.7 Regularization

Regularization allows us to include a penalty, lambda (λ) which is used to penalize movies with large estimates that are formed by smaller sample sizes. We implement cross validation to select λ. We will apply regularization method on the movie, user and genre combination model and determine the RMSE.

2.4.7.1 Identify the lambda yields minimum RMSE value The lambda that yielded minimum value of RMSE, 0.8651294 is 4.4. We will utilize this lambda to determine the RMSE.

2.4.7.2 Plot of lambdas vs rmses from the regularized model



2.4.7.3 Finding the RMSE based on the lambda that yielded minimum RMSE Since we now know that the lambda that yielded the minimum RMSE of 0.8651294 is 4.4 for the movie+user+genre combination model, we will employ this lambda value in our regularized combination model to determine the RMSE.

```
## [1] 0.8651294
```

2.4.8 Final hold out set validation

We validate the final hold out set using the lambda from the previous section that generated a minimum RMSE and test it against the hold out set also known as validation set. This will be the only time this hold out set will be validated. This set was not used in any training or testing purposes. The RMSE from the final hold out set is,

```
## [1] 0.864456
```

3 Results

Let's build a table to display the RMSE results as we train and test each of the models.

Models	RMSEs
Naive	1.0611350
Movie Effects	0.9441568
User Effects	0.9795916
Genre Effects	0.9441568
Movie+User Effects	0.8659736
Movie+User+Genre Effects	0.8656018
Regularized Movie+User+Genre	0.8651294
Final Holdout Set	0.8644560

3.1 Model performance

As noted in the table above listing models and their corresponding RMSE, it is evident that the regularized movie, user and genre combination yielded the lowest RMSE. This is determined as the "best" model and the lambda from this model is used to validate and evaluate the final holdout set. The RMSE of the final holdout set meets the target requirement. Target requirement is < 0.86490 and final holdout set RMSE is 0.864456.

4 Conclusion

In summary, the goal of this project is to develop a recommendation system that generates a RMSE less than 0.86490 by using the MovieLens dataset and by leveraging the knowledge gained through PH125 course series. After training and testing various models including individual effects on movie, user, genre as well as a combination of movie,user, genre and after applying regularization to the combination model, RMSE of 0.8651294 was achieved. The RMSE of the final holdout set is 0.864456 which meets the project requirements of < 0.86490.

Even though the combination of movie, user and genre model with regularization meets the requirements, there are still room for improvement. For example, factors such as the year a movie released and the year a movie was rated and the combination of these effects could lead to even more reduced RMSE. Furthermore, various techniques could be employed such as Gradient Boosting Machine, KNN, Random Forest and matrix factorization.This warrants a more exhaustive computing which currently is lacking in my setup. For future work, I would focus on the advanced algorithms and techniques to explore other metrics to develop an ideal model.