HarvardX Data Science Capstone Project: Movielens

Matthew Joseph Diliberto

16 February 2021

Introduction and Objective

This project represents the final step in the 9-course Data Science Series offered by HarvardX.

By analyzing the Movielens dataset, which provides about 10M movie ratings, we will be building a Movie Recommendation System.

In order to facilitate the process, the EdX team has set up the data so that 90% of the entries (i.e. user ratings) are assigned to a Training Set whereas the remainder will be assigned to a Validation set. The quality of the Recommendation System will be evaluated according to the RMSE (formula provide here below). The RMSE should be lower than 0.87750 on the validation dataset.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$$

Outline of Approach

The approach followed througout the project will be based upon a sequential process, typical of most Data Science projects:

1. Data Collection
2. Data Preparation
3. Exploratory Data Analysis (EDA)
4. Model Design and Development
5. Model Evaluation and Optimization
6. Results Interpretation

We can essentially skip over the first two steps of the process given the fact that they have largely been carried out directly by the EdX Team when providing the data.

With regards to the Data Preparation step, the only actions that I executed in the project were linked to transforming the timestamp variable into a more interpretable format and creating an additional variable describing the year of release of the movie.

Exploratory Data Analysis and Visualization

During this phase, we will try to look more closely into the data and mature a better understanding of it. Through this closer inspection, we will start to develop some key intuitions that will be useful when designing the models.

Key Questions Addressed

Some of the key driving questions to be addressed in this phase can be summarized in the following list:

1. How is the data stored (i.e. which data formats are being used)?
2. How many unique entries in terms of movies, users and genres are present in the data?
3. How do users give ratings on average? What are the most common ratings assigned to movies?
4. How do these ratings vary across users on average?
5. How many ratings do users provide on average?
6. How do ratings change on average according to the year of release?
7. How does the year of release impact the average number of ratings for the movie?

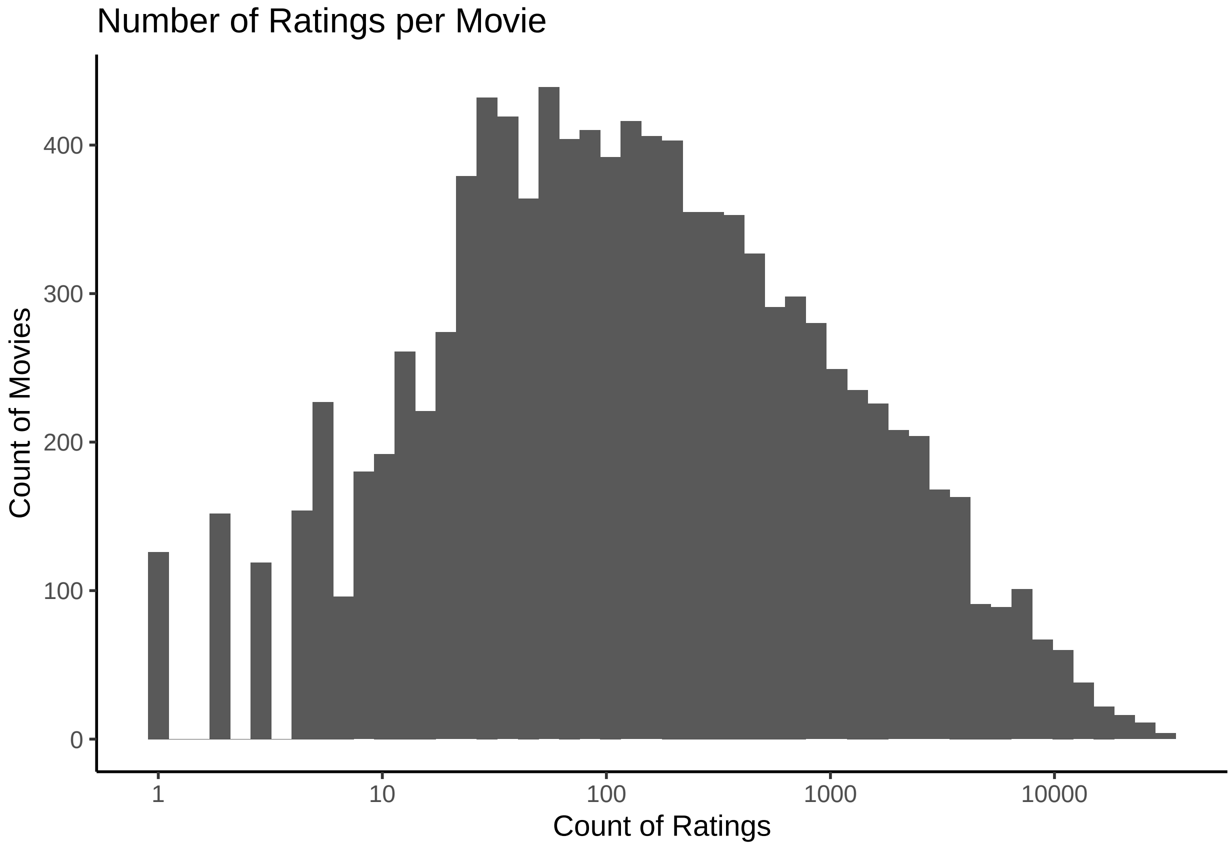
```
## Rows: 9,000,055
## Columns: 8
## $ userId       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ movieId      <dbl> 122, 185, 292, 316, 329, 355, 356, 362, 364, 370, 377,...
## $ rating       <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ...
## $ timestamp    <int> 838985046, 838983525, 838983421, 838983392, 838983392,...
## $ title        <chr> "Boomerang (1992)", "Net, The (1995)", "Outbreak (1995...
## $ genres       <chr> "Comedy|Romance", "Action|Crime|Thriller", "Action|Dra...
## $ date_review  <dttm> 1996-08-04, 1996-08-04, 1996-08-04, 1996-08-04, 1996-...
## $ year_release <dbl> 1992, 1995, 1995, 1994, 1994, 1994, 1994, 1994, 1994, ...
```

As we can see, the structure of the dataset is based on the fact that each row represents a rating provided by a user. Each user can potentially provide a single rating for multiple movies. This basic understanding will be important when taking into consideration the various groupings and aggregations that will be implemented.

```
##   n_unique_movies n_unique_users n_unique_genres
## 1           10677          69878             797
```

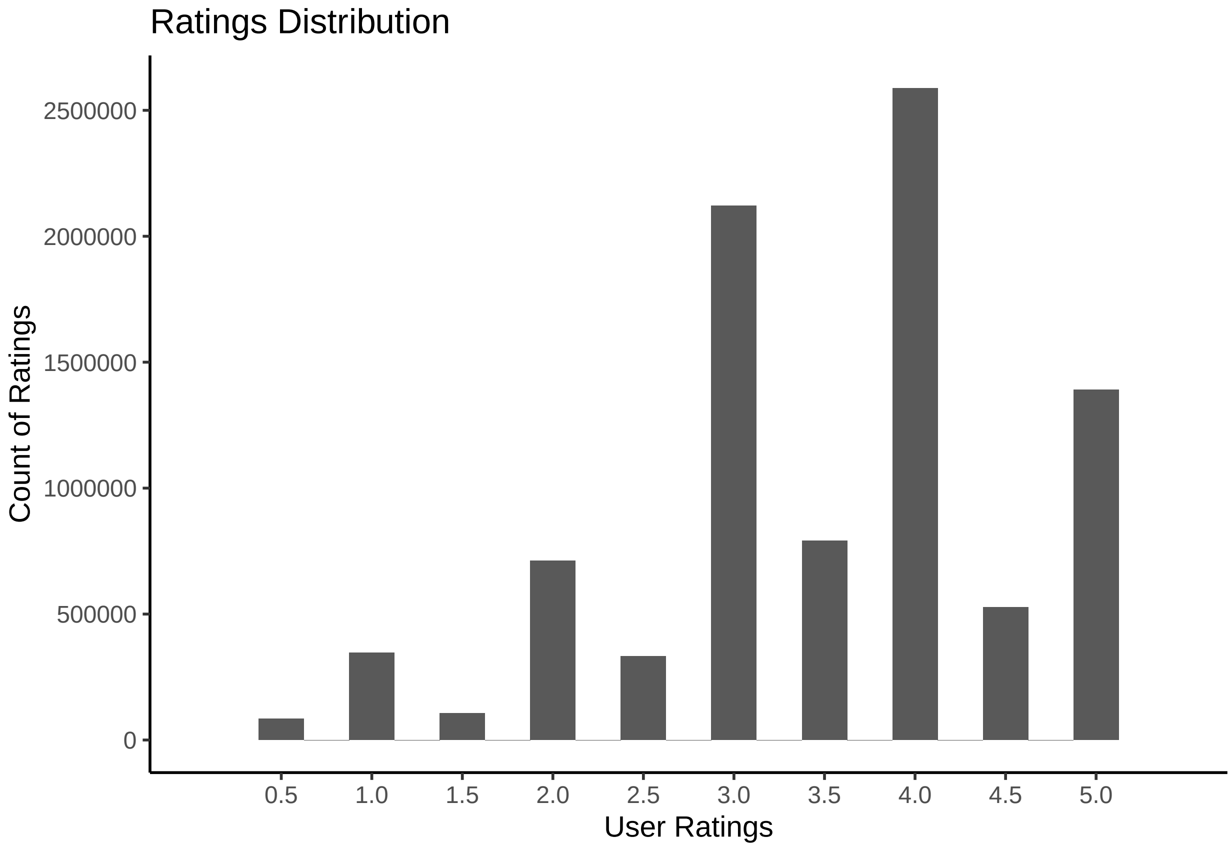
Given the fact that there are about 70k unique users, 10.7k movies and about 10M ratings, it will be interesting to understand how these ratings are distributed across the movies (i.e. how many ratings are provided on average for each movie).

For this purpose, the following visualization will prove useful.



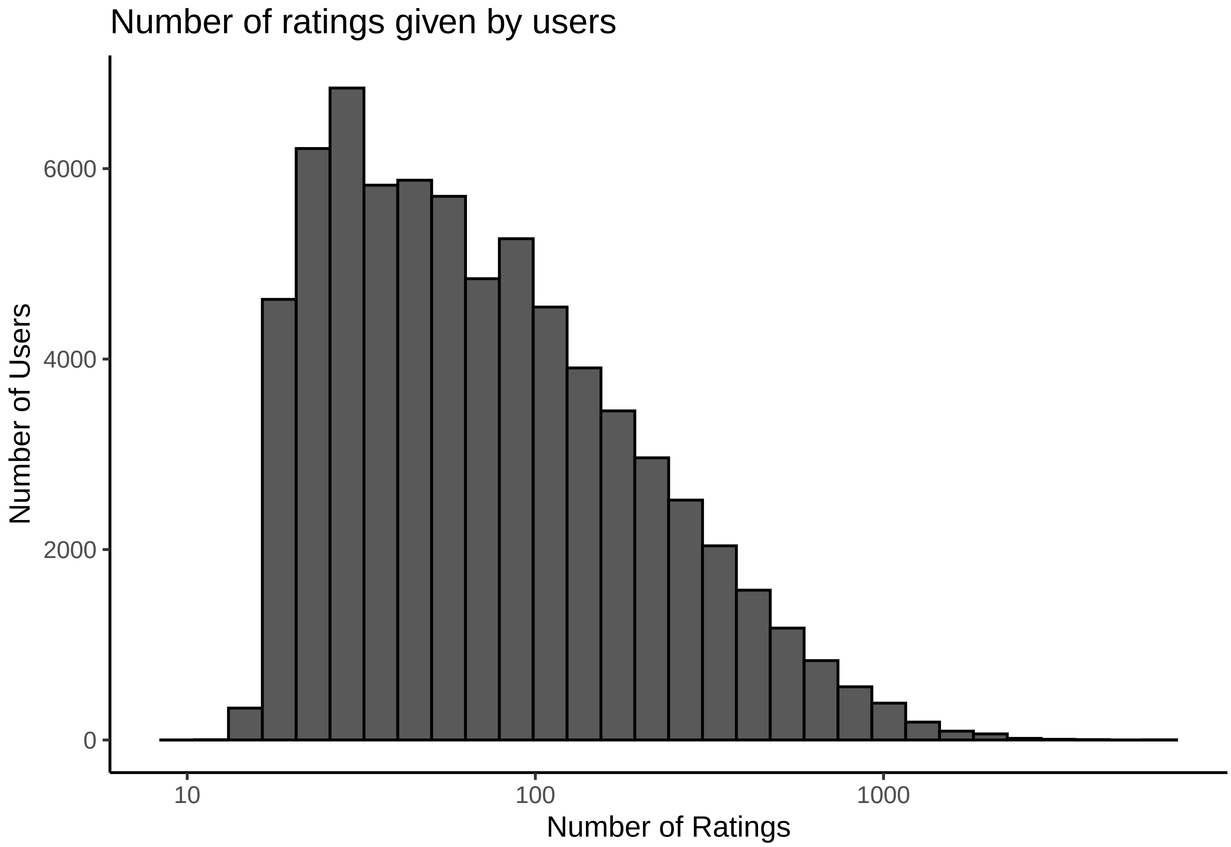
As we can see, there is a rather large variance in the dataset (note: the ‘Count of Ratings’ axis is expressed on log-scale). We immediately notice that there are many movies receving a single rating. On the other hand, there are several movies that have received a number of ratings above 10k.

Another key question is related to understanding how users tend to assign ratings. What are the most common ratings assigned by users?



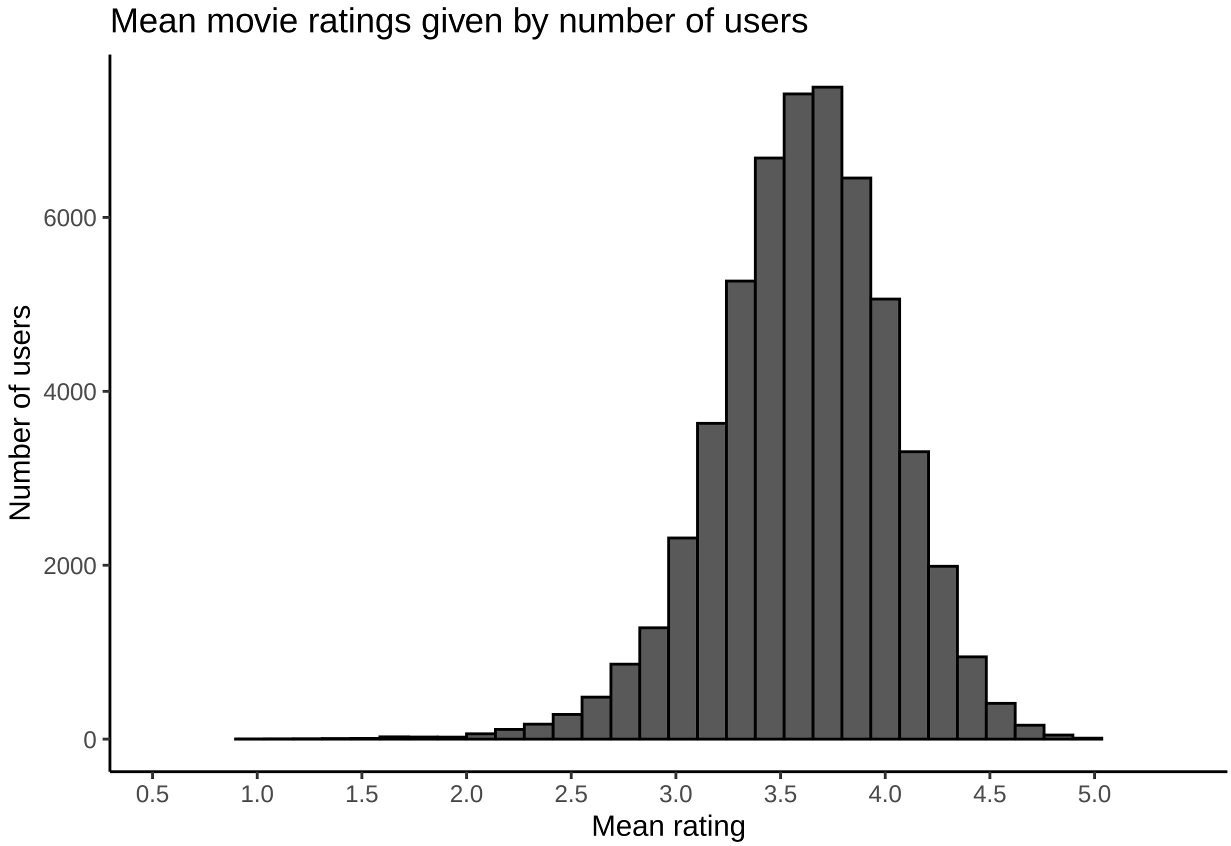
The most common ratings appear to be 4.0, 3.0, 5.0 and 3.5, with the majority of the ratings falling between 3.0 and 4.0. Generally speaking, we can notice how 'half-star ratings' are less common compared to ratings will full integers.

We can move on to understand how the number of ratings per user varies across the userbase. Can we identify some users that are more active than others?



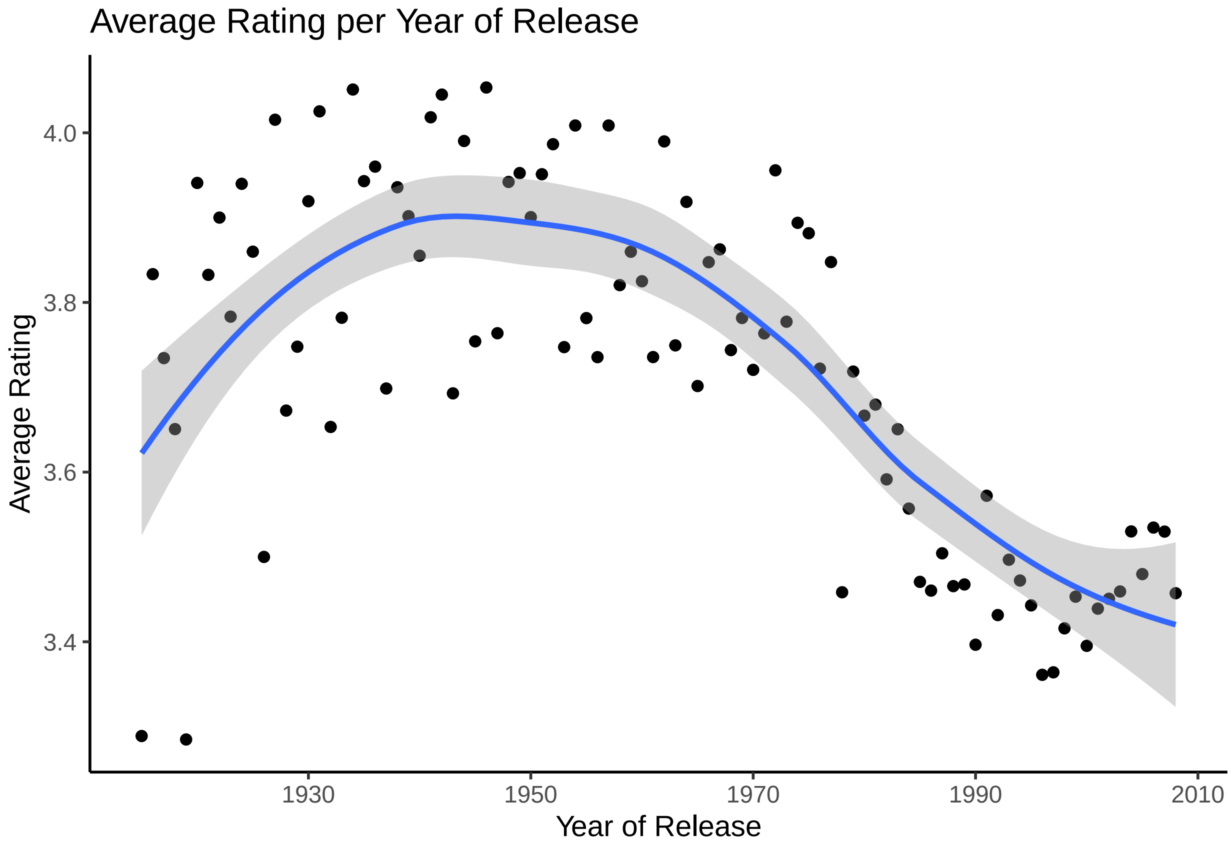
In this case as well, there appears to be a significant degree of variance across the user base, with some users providing more than a 1000 ratings!

Let us know see how the ratings vary on average compared to the user base.



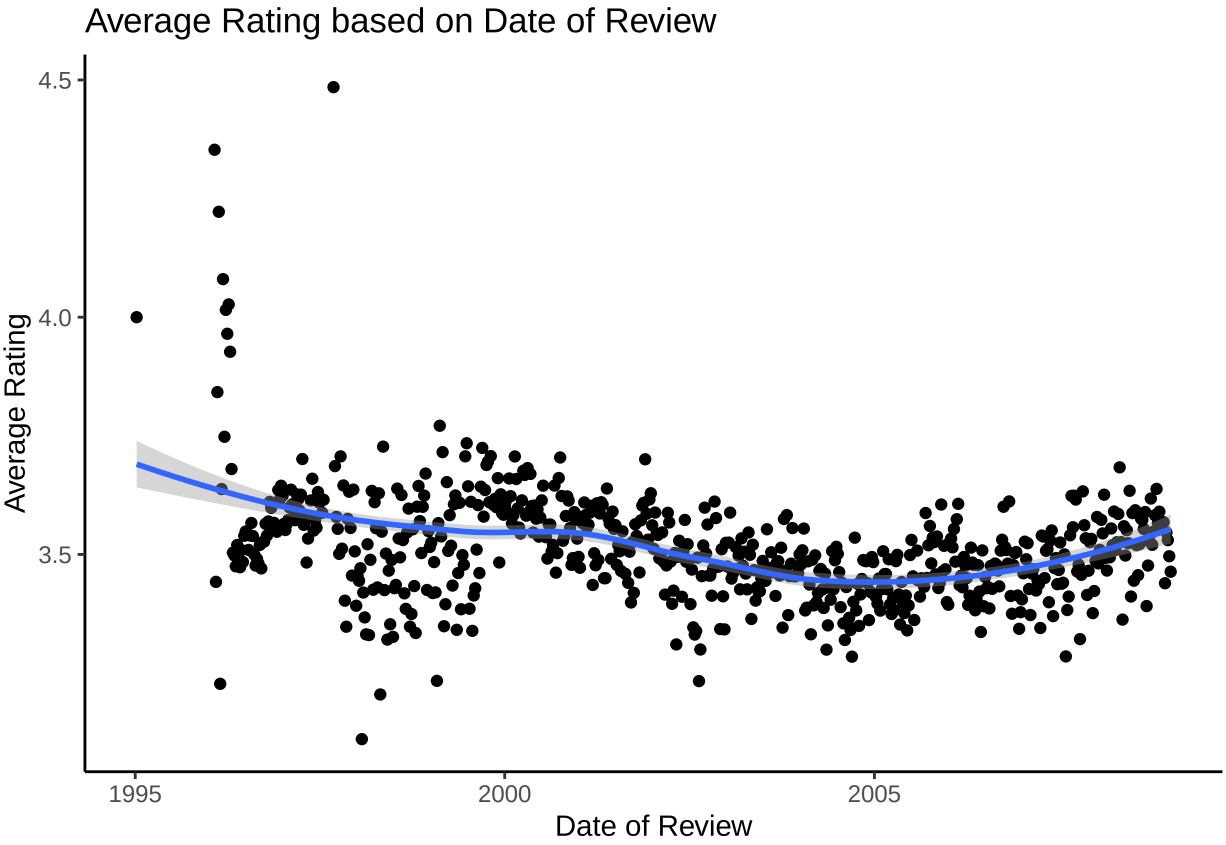
The distribution seems to reach a peak around a mean rating of 3.5. This will be useful when designing our models. We will be able to leverage this aspect when making a prediction with regards to user ratings.

Let us know explore how the average year may vary based on the year of release of the movie (derived from the title variable).



There appears to be an interesting trend where the average rating increases in the year of release range between 1940 and 1960, to then decrease onwards.

Let's look into the how the date of review could impact the average rating.



Aside from the first few years of ratings, it seems that the average rating has remained approximately stable throughout the period.

Model Development and Implementation

In the following section, we will build various models, gradually increasing the complexity and decreasing the RMSE.

Simple Average

The first model will be structured around making a prediction simply based on the average rating provided in the training data set.

```
## [1] 3.512465
```

The average of the training data is around 3.51. Using this for making a prediction in the validation set, we obtain the following results.

Model	RMSE
MODEL A: Simple Average	1.061202

As we can see, the result for this first model does not satisfy the threshold of 0.87750 RMSE.

Movie Effect Model

We shall now explore how the model will change leveraing information regarding the movie itself. More specifically, we shall refine the model by incorporating a "Movie Effect", which captures the average rating for the movie itself.

Model	RMSE
MODEL A: Simple Average	1.0612018
MODEL B: Movie Effect	0.9439087

The results have improved to an RMSE of 0.944, suggesting that this is an important element.

User Effect Model

We shall build on this model taking into consideration the 'User Effect', which captures the average rating of the user in order to further refine the predicted recommendation.

Model	RMSE
MODEL A: Simple Average	1.0612018
MODEL B: Movie Effect	0.9439087
MODEL C: User & Movie Effect	0.8653488

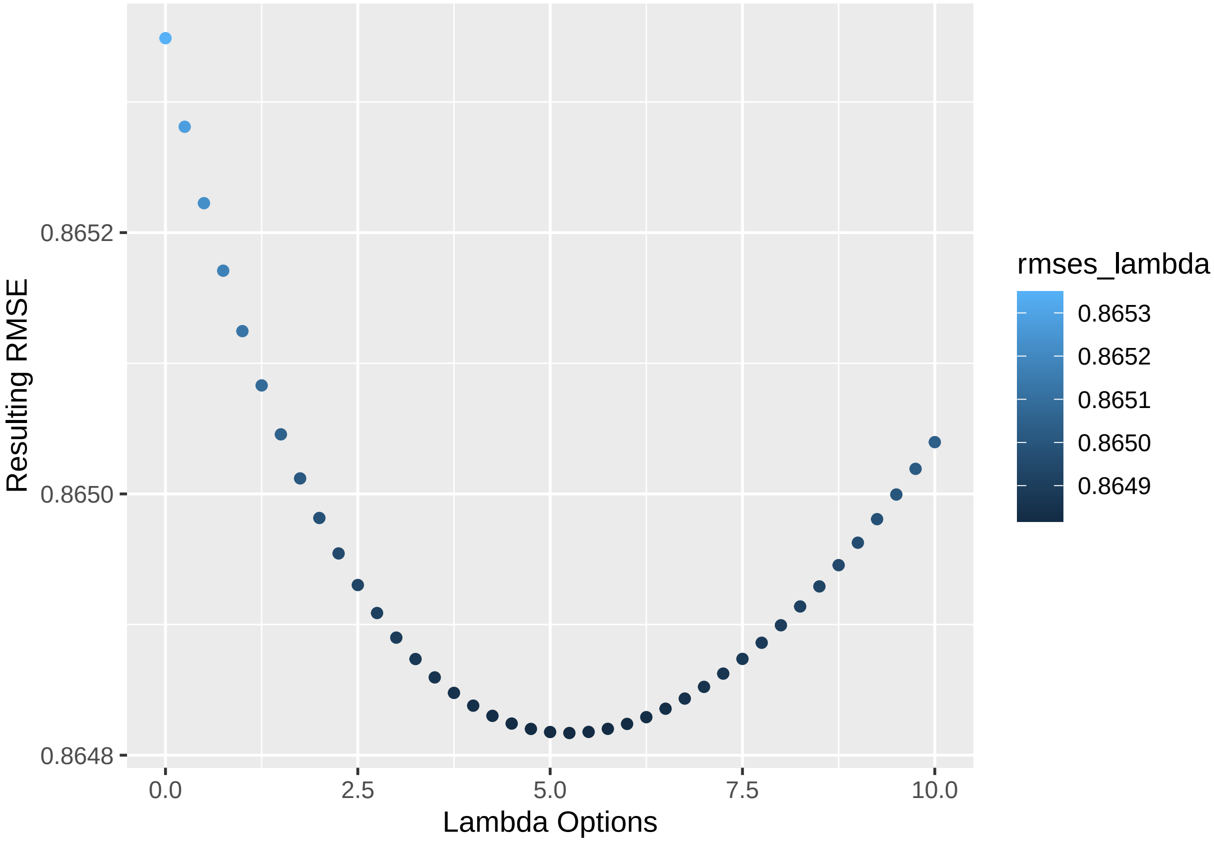
Once again, the model has improved significantly. This model satisfies the threshold of 0.87750 RMSE.

Regularized Model

To further improve on this, we shall apply regularization to these effects. In order to do so, we must find the optimal lambda to be utilized.

Model	RMSE
MODEL A: Simple Average	1.0612
MODEL B: Movie Effect	0.9439
MODEL C: User & Movie Effect	0.8653
MODEL D: Regularized User & Movie Effect	0.8648

The optimal lambda to be utilized is equal to 5.25.



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

This result represents a further (slight) improvement compared to the previous model with an RMSE of 0.864817, thus being slightly below the requirement for achievement the maximum grade in the project evaluation.

Final Results and Interpretation

As we can see, the best results have been provided through a regularized version of the model. Regularization ensures that the model rating prediction is not excessively affected by user or movie effects estimated with small sample sizes.

In order to further improve the model, we could evaluate the following strategies:

1. Importing additional data sources. For example, one could leverage the imbd database to link movies to directors, actors and other factors that may be relevant.
2. Identify whether movies are part of a 'series' (i.e. sequels or prequels). The intuition (that should be tested) is that people who enjoy a movie typically will also appreciate the prequel or sequel in the series.
3. Utilize more advanced techniques mentioned during the course such as Matrix Factorization.