HarvardX: PH125.9X - Data Science: Capstone
Movielens Rating Prediction

Wilson Tan

13 Nov 2020

Contents

1 Executive Summary

The objective of the project is to create a recommender system to predict movie ratings using the Movielens 10M dataset. The dataset is made up of 10 Million ratings, ranging from 0.5 to 5 stars, assigned by approximately 70,000 unique users across 11,000 unique movies.

90% of the dataset is set aside as "edx" to train the model while the remaining is used as "validation" to evaluate the proposed models. The Root Mean Square Error (RMSE) is used to evaluate the algorithm performance. RMSE measures the differences between predicted values and true values. This is regarded as a standard way to measure the model's accuracy. The RMSE of predicted values $\hat{y}$ versus true values y, for N observations (for movie i and user u) is given by:

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

Considering $RMSE = 0$ would indicate a perfect fit to the data, a lower RMSE is generally desired over a higher one. The best performing model has registered a RMSE of 0.794, representing a significant improvement from the RMSE of 1.060 based on the Naive Baseline Model. Despite this, there is still room for improvement by including effects like Release Year, Review Month and Genre to the model. Unfortunately, due to the limitations of the hardware (RAM), these models cannot be validated.

2 Preparation

This section elaborates on the steps taken from installing libraries through data processing to train-test split.

2.1 Prerequisites

The libraries required in this modeling are as follow:

```
# Load installed libraries
library(tidyverse)
library(caret)
library(data.table)
library(recosystem)
library(kableExtra)
```

The operating system used in this modelling are as follow:

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## system         x86_64, mingw32
## status
## major          4
## minor          0.2
## year           2020
## month          06
## day            22
## svn rev        78730
```

```
## language       R
## version.string R version 4.0.2 (2020-06-22)
## nickname       Taking Off Again
```

2.2 Access to Data

The MovieLens dataset can be accessed via:

- [MovieLens 10M dataset] https://grouplens.org/datasets/movielens/10m/
- [MovieLens 10M dataset - zip file] http://files.grouplens.org/datasets/movielens/ml-10m.zip

2.3 Edx-Validation Split

In order to evaluate the performance of the model, the Movielens 10M dataset is split into 2 subsets, "edx" and "validation". Algorithm development will be carried out on the "edx" subset while "validation" subset will be used to test the final algorithm.

2.4 Data Processing

A quick overview of the table indicates that additional information could be extracted from the existing features for a more consistent evaluation.

Unprocesssed edx 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

From the table, it is notable that some of the features can be further processed. These include:

1. Extract the `release` year from the `title` feature;
2. Convert `timestamp` of the review to a readable date format before extracting the `month` and `year`.
3. Separate `genre` from the pipe-separated value in the `genres` feature. This is expected to increase the size of the dataset. To avoid the dataset having duplicate ratings, this will be stored as the edx_genre dataset for analysis purpose;

After processing the dataset, the number of columns in both the edx and validation datasets should increase from 7 to 10.

Processed edx dataset

userId	movieId	rating	timestamp	title	release	genres	date	year	month
1	122	5	838985046	Boomerang	1992	Comedy|Romance	1996-08-02 19:24:06	1996	08
1	185	5	838983525	Net, The	1995	Action|Crime|Thriller	1996-08-02 18:58:45	1996	08
1	292	5	838983421	Outbreak	1995	Action|Drama|Sci-Fi|Thriller	1996-08-02 18:57:01	1996	08
1	316	5	838983392	Stargate	1994	Action|Adventure|Sci-Fi	1996-08-02 18:56:32	1996	08
1	329	5	838983392	Star Trek: Generations	1994	Action|Adventure|Drama|Sci-Fi	1996-08-02 18:56:32	1996	08
1	355	5	838984474	Flintstones, The	1994	Children|Comedy|Fantasy	1996-08-02 19:14:34	1996	08

As the separation of genres could result in duplicate ratings, it is preferable to store the results of the separation of multiple genres under a different dataset.

Processed edx_genre dataset

userId	movieId	rating	timestamp	title	release	genres	date	year	month	genre
1	122	5	838985046	Boomerang	1992	Comedy|Romance	1996-08-02 19:24:06	1996	08	Comedy
1	122	5	838985046	Boomerang	1992	Comedy|Romance	1996-08-02 19:24:06	1996	08	Romance
1	185	5	838983525	Net, The	1995	Action|Crime|Thriller	1996-08-02 18:58:45	1996	08	Action
1	185	5	838983525	Net, The	1995	Action|Crime|Thriller	1996-08-02 18:58:45	1996	08	Crime
1	185	5	838983525	Net, The	1995	Action|Crime|Thriller	1996-08-02 18:58:45	1996	08	Thriller
1	292	5	838983421	Outbreak	1995	Action|Drama|Sci-Fi|Thriller	1996-08-02 18:57:01	1996	08	Action

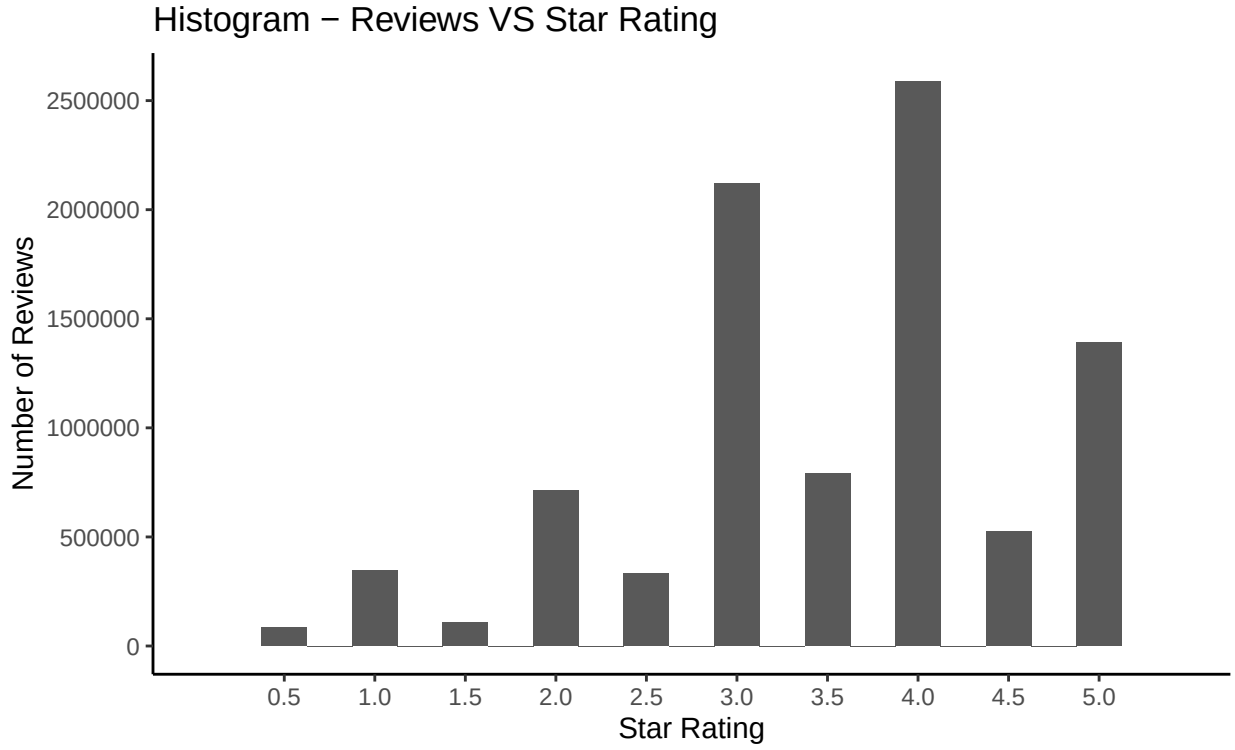
2.5 Train-Test Split

With such a large dataset, the modeling process may be too demanding for the available RAM in some machines. To reduce the time spend on testing the different models prior to the final algorithm, the "edx" dataset is split further, with 10% of the data randomly allocated to the "test" set and the remaining 90% allocated to the "train" set.

3 Data Analysis

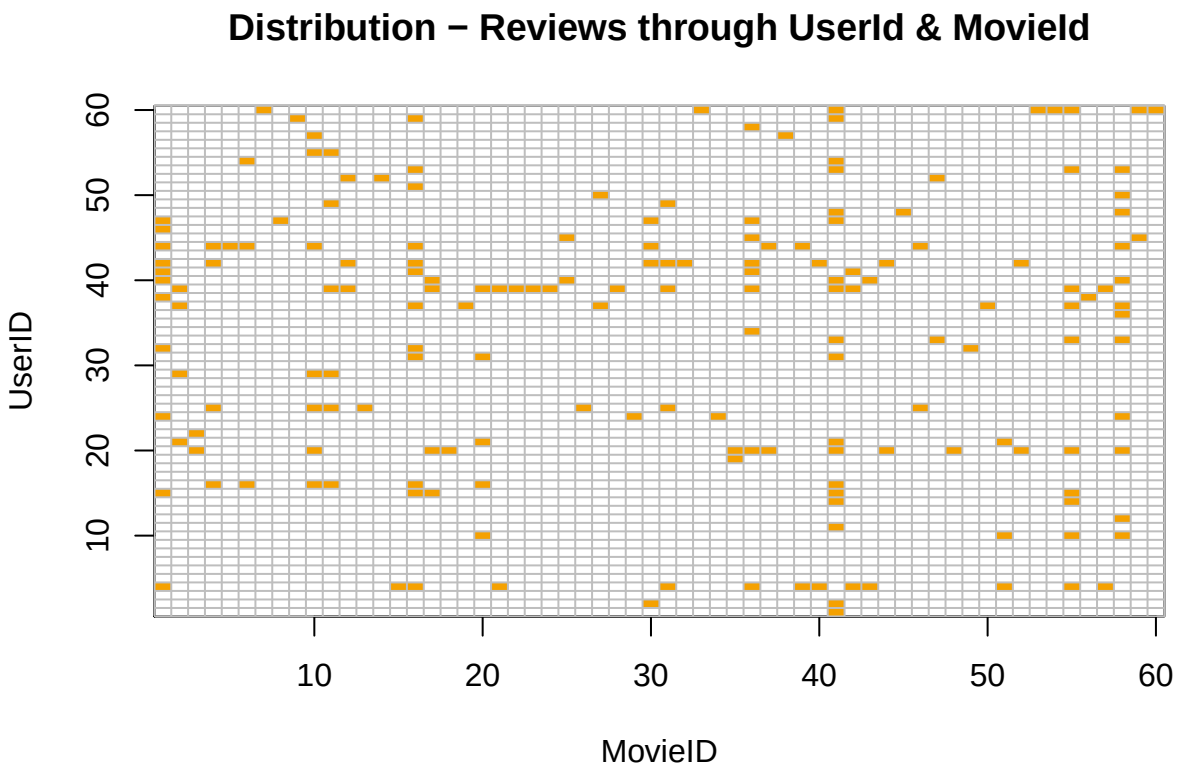
This section elaborates on the steps taken to identify notable trends and correlations between the rating and the features.

3.1 Number of Reviews on Rating Score

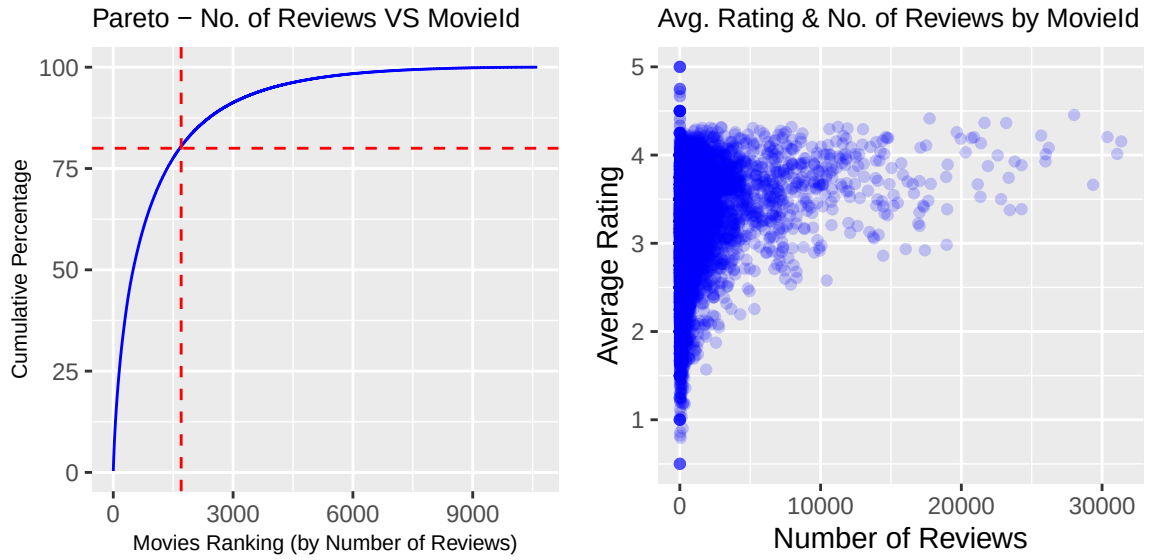


By plotting the number of reviews against rating score, it is apparent that half star ratings are less common than full star ratings.

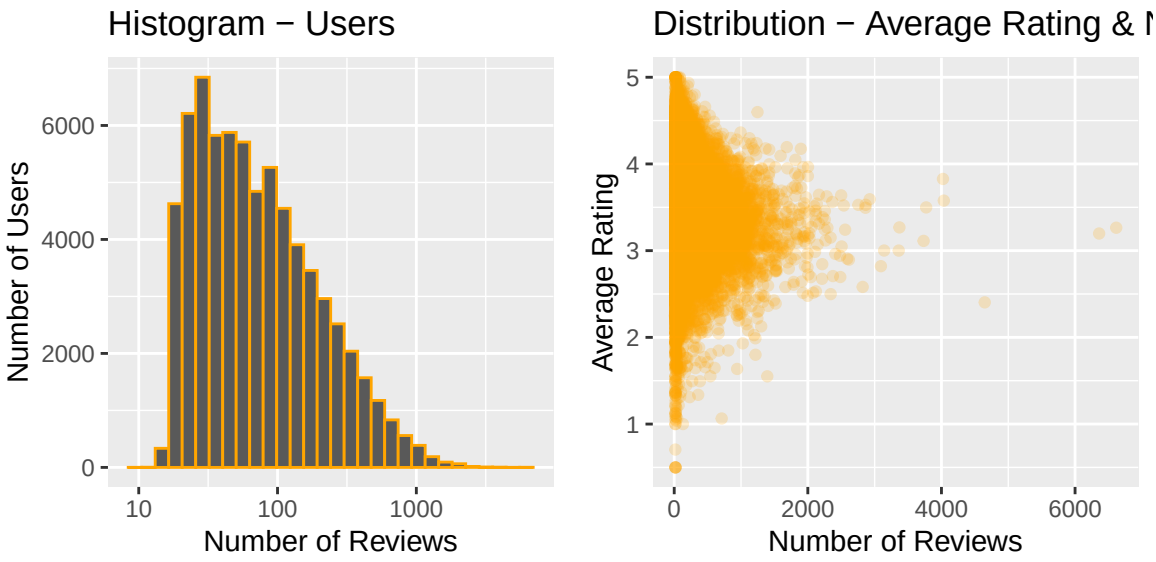
3.2 UserId & MovieId



Clearly, there are movies with more reviews than the others. At the same time, the frequency of review carried out by some users is higher than others.

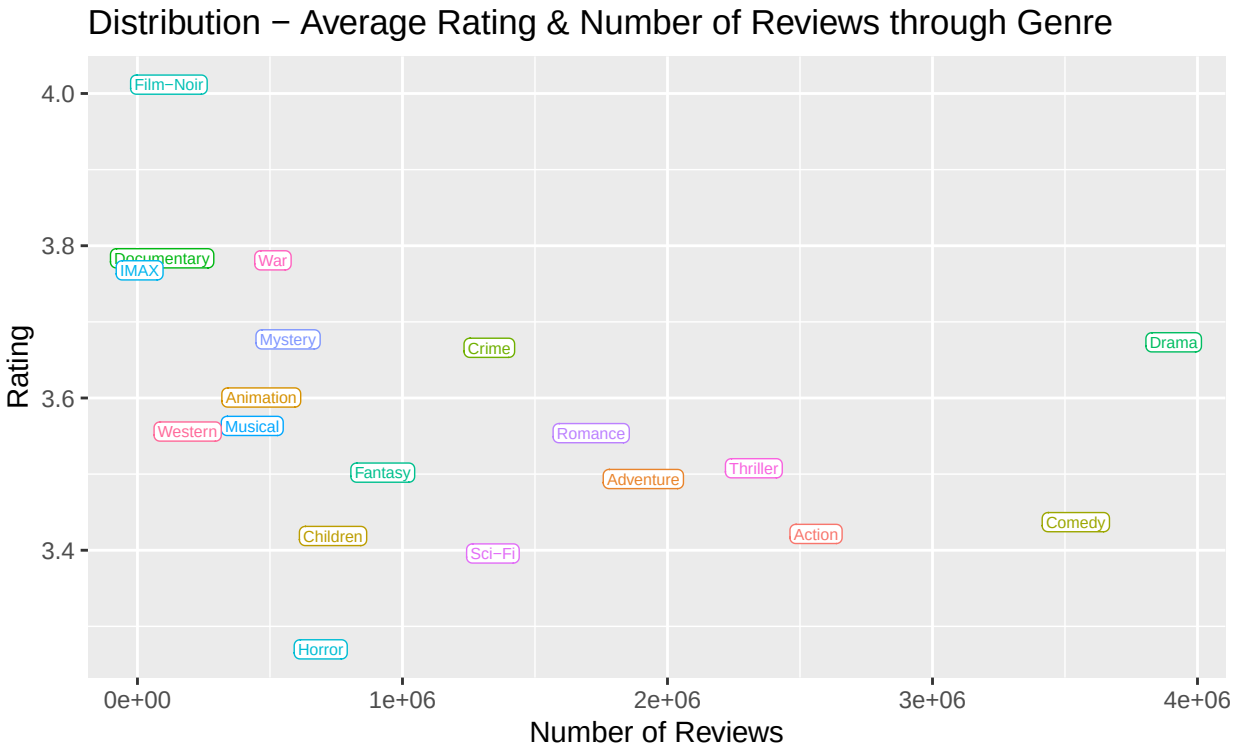


The Pareto Chart offers a deeper insight into the distribution of the reviews by movie. In fact, roughly 1,700 (<20%) movies made up 80% of the total number of reviews. It is also noteworthy to point out from the distribution on the right that the more critic movies (with more reviews) tend to have higher average rating.



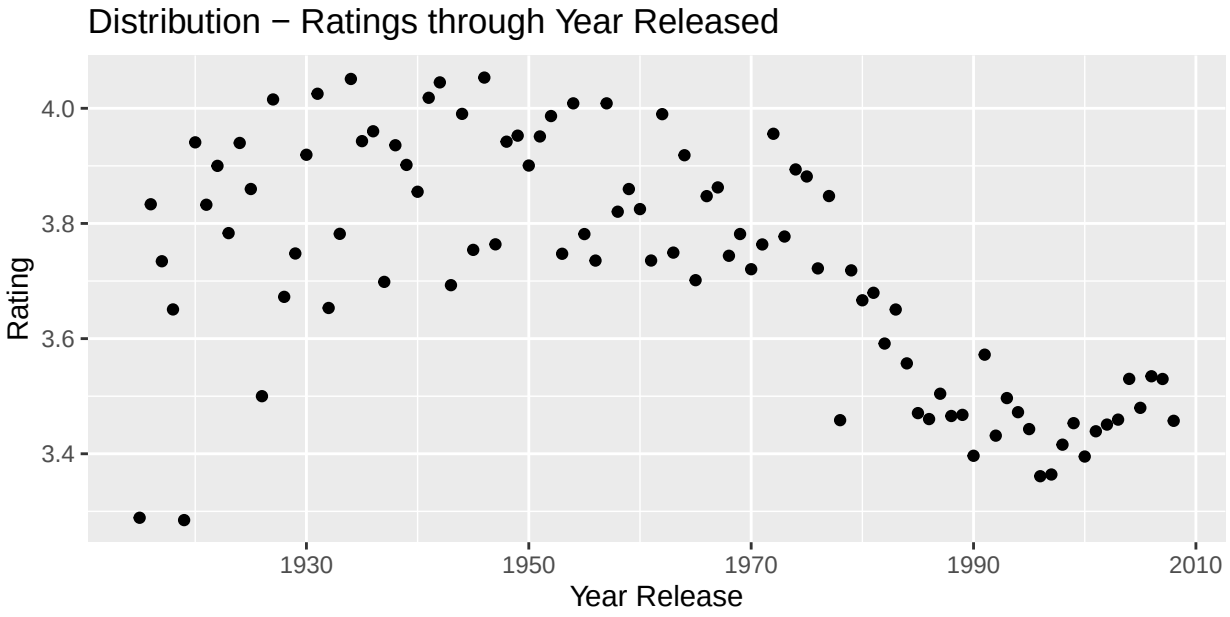
The histogram has further reinforced the point that certain users review more often than the others. As for the distribution, it can be observed that for the users with more reviews, the average rating tends towards the overall mean at 3.5125.

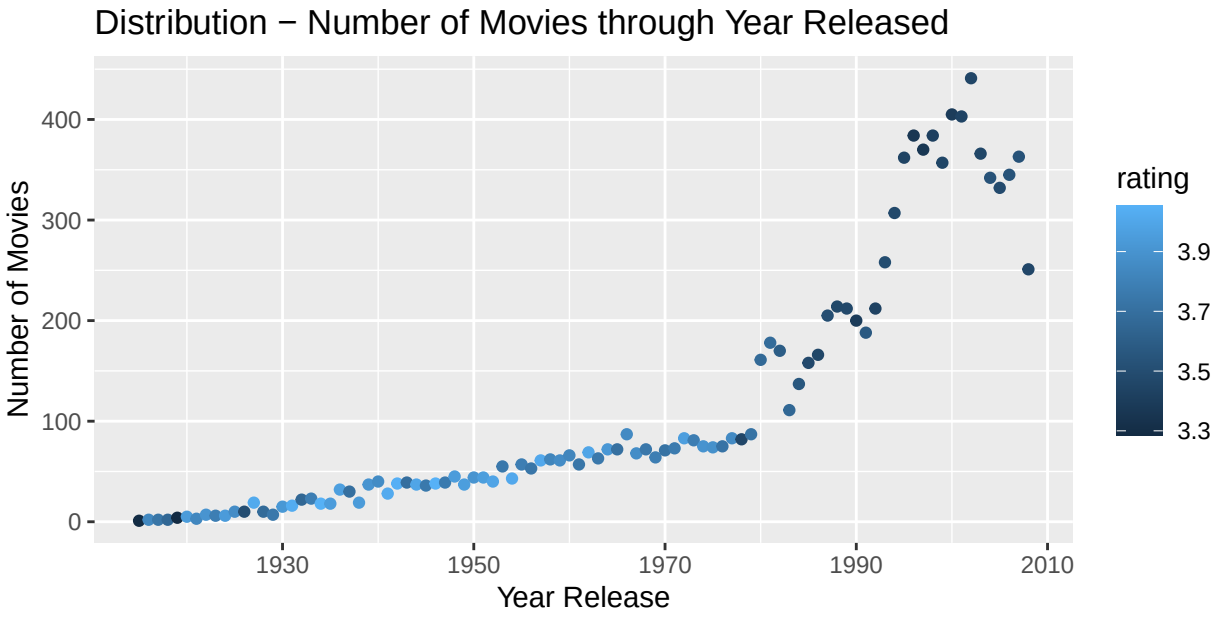
3.3 Genre



From the distribution, it can be pointed out that genres with lower number of reviews tend to have better average rating.

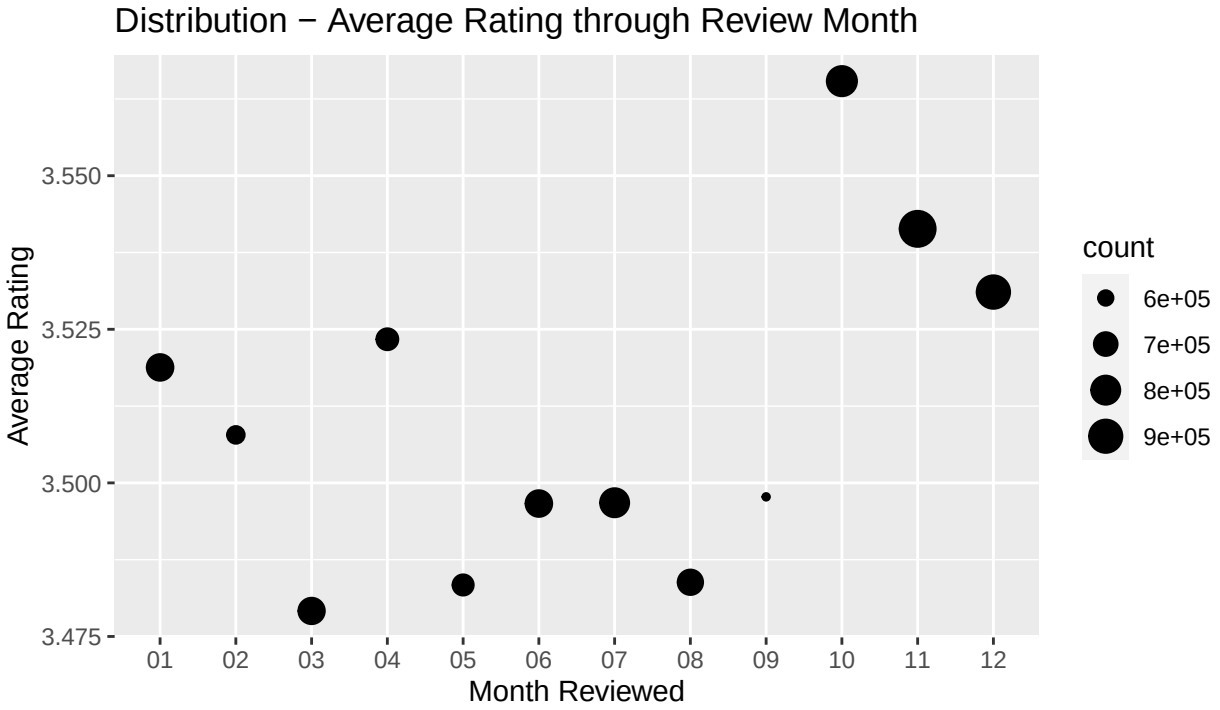
3.4 Release Year

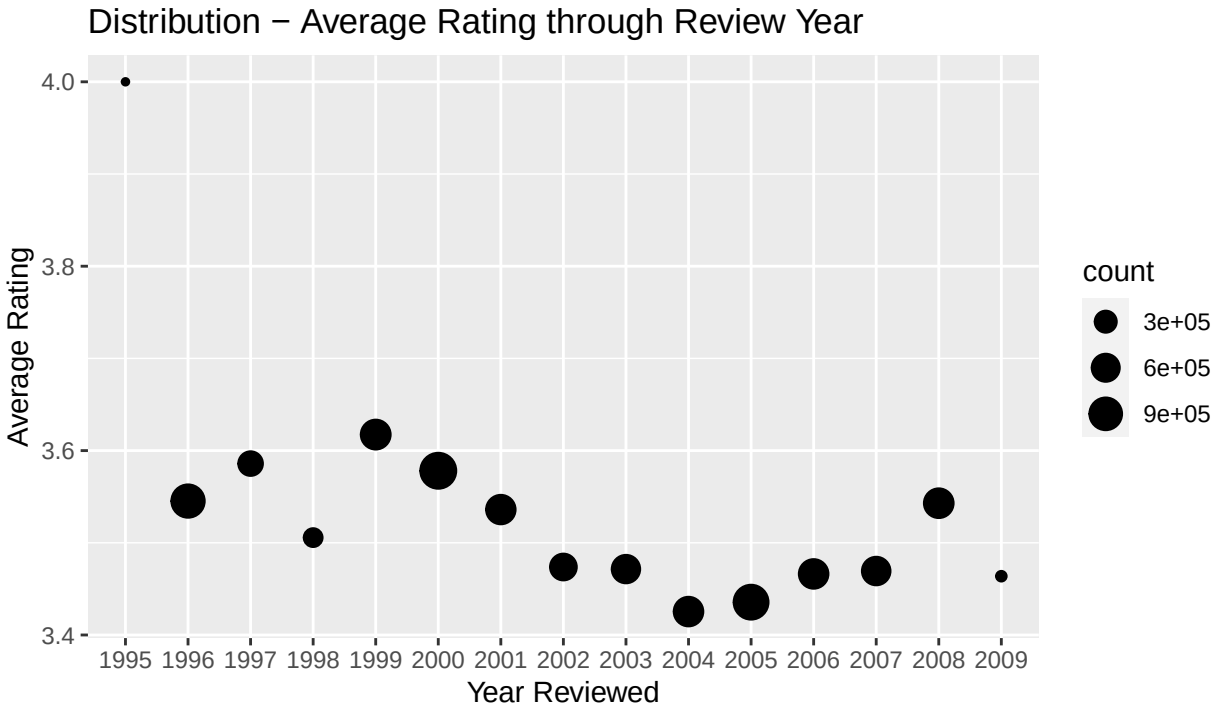




The first chart implies that movies released in earlier years are more highly rated than movies released in recent years. A reason could be that the audience tends to re-watch classics and are likely to rate these favorably. A further point to note is that the incorporation of reviews into online platforms only started in 1990s. (The age group of the reviewers may explain the sudden drop in rating between 1980s to 1990s)

3.5 Review Timestamp





As reflected in the first chart, both the average rating and number of reviews in Oct, Nov and Dec tend to be higher than the other months. This may be due to blockbuster (highly anticipated) movies release scheduled for the winter holidays. As for the review year, the significance remains inconclusive.

4 Modeling Approach

4.1 Naive-Baseline Model

As the name suggests, the Naive-Baseline (Simple Average) Model will be used as the reference model for measuring the performance of subsequent models. The Naive-Baseline Model assumes that ratings across all movies and users will be the same. In this model, the mean value, which is approximated to be ~3.5125, will be used as the predicted rating for all reviews, regardless of movie or user. The formula of this Naive-Baseline Model can be represented by:

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

where $\hat{\mu}$ refers to the mean and $\varepsilon_{i,u}$ refers to the independent error sampled from the same distribution centered at 0.

The RMSE of the Naive-Baseline Model on the `test` dataset is 1.060.

4.2 Movie Effect

As pointed out in the Data Analysis section, certain movies are rated higher than the others. Therefore, it would be sensible to include the movie effect, b_i, to enhance the model. The resulting formula that represents the Movie Effect Model is given by:

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

where b_i refers to the movie effect or bias for movie i.

The RMSE of the Movie Effect Model on the `test` dataset is 0.9430.

4.3 Movie & User Effect

Another observation from the Data Analysis is that certain users review more often than others. Certain users also tend to award higher rating than the others. In this model, the user effect, b_u, is introduced to incorporate user bias into the model. The resulting formula that represents the Movie & User Effect Model is given by:

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

where b_u refers to the user effect or bias for user u.

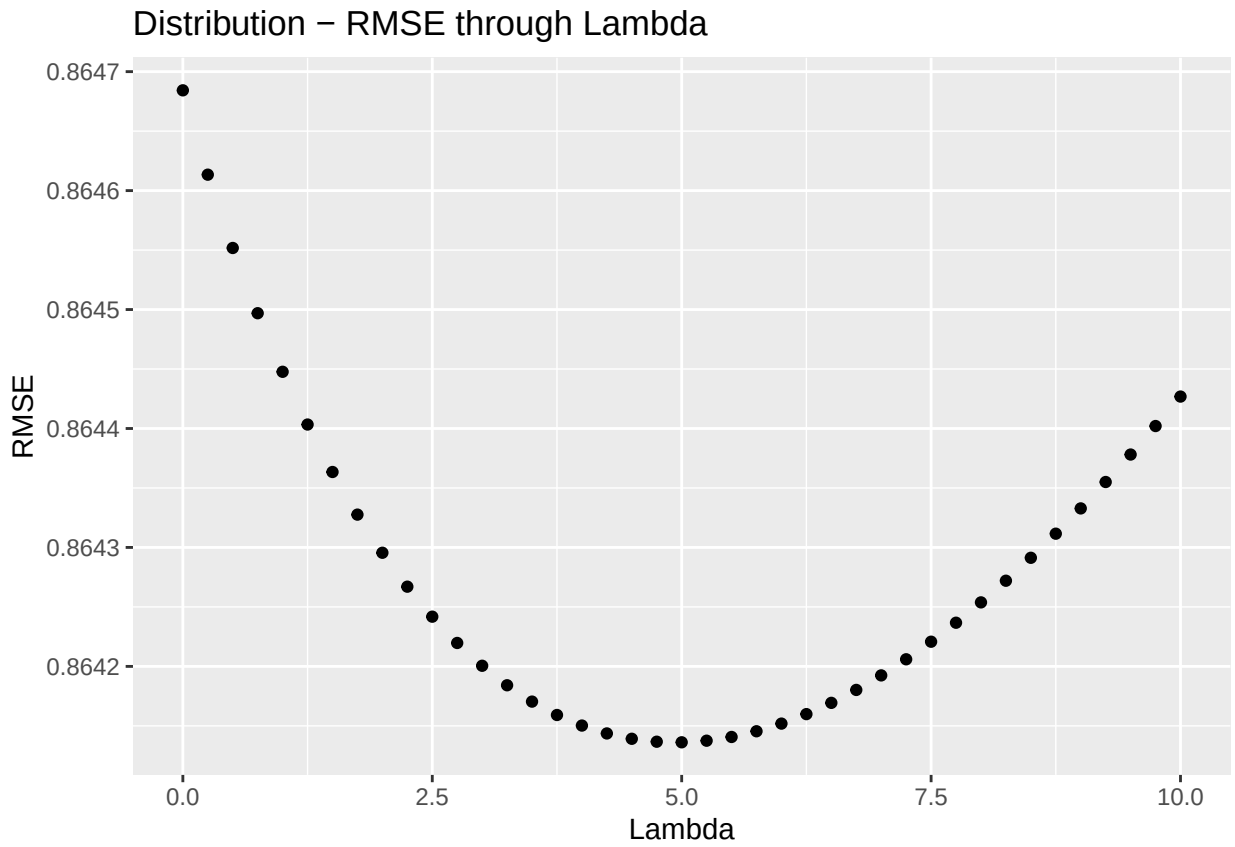
The RMSE of the Movie & User Effect on the `test` dataset is 0.865.

4.4 Movie & User Effect + Regularization

To mitigate the risk of overfitting, regularization is applied to the model. The use of regularization penalizes on movies with very few ratings or users who only rated a very small number of movies. The formula that represents the Movie & User Effect + Regularization Model is:

$$\frac{1}{N}\sum_{u,i}(y_{u,i} - \mu - b_i - b_u)^2 + \lambda(\sum_i b_i^2 + \sum_u b_u^2)$$

where λ is the tuning parameter applied to the movie and user effect.



From the plot, it can observed that the lambda value that corresponds to lowest RMSE of 0.864 in the train set is 0.5. Applying this lambda value onto the the test set, the resulting RMSE (of the Movie & User Effect + Regularization Model) is 0.864. This only represents a slight improvement from the Movie & User Effect Model.

4.5 Movie & User Effect + Matrix Factorization

Matrix Factorization can be applied to further improve the accuracy of the prediction model. The general idea behind the model is to approximate the matrix $R_{m,n}$ by the dot product of two matrices $P_{k,m}$ and $Q_{k,n}$. More information related to Matrix Factorization is available here: https://www.youtube.com/watch?v=ZspR5PZemcs

The resulting RMSE for the Movie & User Effect + Matrix Factorization Model on the `test` dataset is 0.797. As compared to the Movie & User Effect + Regularization Model, this represents a significant improvement from the Movie & User Effect Model. As such, the Movie & User Effect + Matrix Factorization Model will be selected as the final algorithm to be applied on the `validation` dataset.

RMSE Results for All Models

Model	RMSE
[Test] Naive Baseline (Mean) Model	1.0600537
[Test] Movie Effect Model	0.9429615
[Test] Movie & User Effect Model	0.8646843
[Test] Movie & User Effect + Regularization Model	0.8641362
[Test] Movie & User Effect + Matrix Factorization Model	0.7965118
[Validation] Movie & User Effect + Matrix Factorization Model	0.7939559

As reflected in the table, the RMSE for the final Movie & User Effect + Matrix Factorization Model is 0.794.

Conclusion

Characterized by the lowest RMSE value, The Movie & User Effect + Matrix Factorization Model is regarded as the optimal model for predicting movie ratings. From the earlier data analysis, it is evident that there are other features like release year, review month and genre that could be applied to further improve on the model's prediction accuracy. However, due to the limitations of the hardware (RAM), these models cannot be validated.