# MovieLens Capstone Project:
# HarvardX PH125.9x Data Science Capstone

Adam J. E. Blanchard

May 2, 2021

## Contents

# 1 Overview

Based roughly on the Netflix challenge (https://www.netflixprize.com), this project aimed to develop a machine learning algorithm to predict movie ratings using the publicly available MovieLens dataset collected by GroupLens Research (https://movielens.org). The dataset contains 10 million user ratings of over 10,000 different movies. The aim of the project was to develop a predictive algorithm that would result in the lowest root mean squared error (RMSE) in a partitioned validation dataset. In particular, this report details the installation, cleaning, and exploration of the MovieLens data, as well as the development and evaluation of several predictive models. The code complete code for this project is available in the supplemental materials.

## 1.1 Introduction

Recommendation systems utilize user generated ratings of items in order to provide specific recommendations for other items. Many companies use recommendation systems in their pursuits. Typically, companies that sell products or services to a large volume of customers and also allow customers to rate these items are then able to amass extremely large datasets of user ratings. These datasets can subsequently be used to develop algorithms used to make predictions of specific user ratings for given items. The use of these algorithms allows the companies to make specific item recommendations to specific users that the users is likely to rate highly. Recommendation systems are used in a variety of areas including movies, books, articles, and social media by many well-known companies including Amazon, Netflix, Spotify and Twitter.

Recommendation systems are a common machine learning application. For instance, Netflix uses an advanced machine learning algorithm to provide media recommendations. The Netflix prize was an open competition announced in 2006 to the data science community. The goal was to find the best filtering algorithm to predict how much a user is going to enjoy a show or movie based on prior movie ratings; the company aimed to find an algorithm that would improve their current system by at least 10%. The $1 million prize was awarded in 2009 to the team "BellKor's Pragmatic Chaos" using an advance ensemble of machine learning techniques (https://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf).

## 1.2 Aim of the Project

This project aimed to accomplish a similar feat to the Netflix challenge. Specifically, the aim was to develop an efficient movie recommendation algorithm using machine learning techniques on the MovieLens dataset. The utility of the algorithm was based on the resultant loss function in the validation dataset.

## 1.3 Specific Requirements of the Project

The 10M version of the MovieLens data was used in order to develop a predictive model. The dataset was first separated into a the edx dataset, used to develop the predictive model, and a validation dataset (i.e., the final hold-out validation set), used to test the final model. The final predictive model was evaluated based on its performance in the final hold-out validation set.

The loss function used to evaluate the predictive algorithm was the root mean square error (RMSE), which is a common metric of distance between predicted and observed values. Values of RMSE were used to evaluate the accuracy of the predictive models during the training phase and to determine the overall accuracy of the final model using the validation dataset. As a measure of distance between predicted and observed values, lower values of RMSE indicate more accurate predictions. The specific RMSE formula used in this project is provided here:

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

## 1.4 Dataset

MovieLens is an online service that provides movie recommendations to its users, as well as conducts online experiments related to recommendation systems and interfaces. For this project the 10M MovieLens dataset was used (https://grouplens.org/datasets/movielens/10m/). This dataset contains over 10 million movies ratings for 10,681 unique movies by 71,567 users of the online service. Users were selected at random to be included in the dataset provided that they had rated at least 20 movies.

Information from MovieLens regarding the dataset reveals the following features:

- No user information is provided in the dataset; all users have been anonymized and assigned a unique UserId code.
- Ratings are made on a 5-star system, which includes half-star increments (i.e., the scale extends from 0.5 to 5.0 with increments of 0.5).
- The timestamp of each rating is included and represents the number of seconds from midnight Universal Time (UTC) of January 1, 1970, to the time the rating was made.
- MovieId numbers are provided directly by MovieLens, while movie titles are entered manually in the format found on IMBD which includes the year of release in parentheses following the title.
- Genre information is stored in a pipe-separated list representing the relevant combination of the following 19 genre options: action, adventure, animation, children's, comedy, crime, documentary, drama, fantasy, film-noir, horror, musical, mystery, romance, sci-fi, thriller, war, and western.

# 2 Data Wrangling

The code to download and create the dataset was provided by the HarvardX PH125.9x Data Science Capstone course (see the complete code for this project in the supplemental code file). Code was also provided by the course to partition the MovieLens dataset into two subsets: the edx dataset (90%) and the final hold-out validation dataset (10%).The provided code also ensured that there were no users and/or movies in the validation dataset that do not appear in the edx dataset.

Notably, the edx dataset will be used for exploratory data analysis, testing different models, and developing the predictive algorithm; it will be split into training and test datsets for building the predictive model. In contrast, the validation dataset will only be used to test the accuracy of the final predictive model, and the overall accuracy of the model will be based on the resulting RMSE from the validation dataset.

## 2.1 Data Inspection

After wrangling the data using the provided code, it can be seen that the edx and validation datasets both contain the aforementioned six variables (i.e., userId, movieId, rating, timestamp, title, and genres). Notably, the outcome of interest that we are interested in predicting is the movie rating variable. The edx dataset contains 9,000,055 ratings, while the validation dataset contains 999,999 ratings. This is consistent with our partitioning of the MovieLens data (i.e., 90% to edx and 10% to validation).

```
## [1] "edx dataset"
```

```
## 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..
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "St"..
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci"..
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
## [1] "validation dataset"
```

```
## Classes 'data.table' and 'data.frame':   999999 obs. of  6 variables:
##  $ userId   : int  1 1 1 2 2 2 3 3 4 4 ...
##  $ movieId  : num  231 480 586 151 858 ...
##  $ rating   : num  5 5 5 3 2 3 3.5 4.5 5 3 ...
##  $ timestamp: int  838983392 838983653 838984068 868246450 868245645 868245920..
##  $ title    : chr  "Dumb & Dumber (1994)" "Jurassic Park (1993)" "Home Alone "..
##  $ genres   : chr  "Comedy" "Action|Adventure|Sci-Fi|Thriller" "Children|Come"..
##  - attr(*, ".internal.selfref")=<externalptr>
```

Looking at Table 1, it is apparent that each row represents a single rating from a single user for a single movie. As the validation dataset will only be used to test the final model, only the edx dataset will be examined in detail. Looking at the data, several important features are immediately apparent:

- The userId and movieId variables are stored as integer and numeric variables, respectively; these variables should be treated as factors or grouping variables in much of the analyses.
- As the timestamp represents the seconds since midnight UTC January 1, 1970, to the time of the rating and is stored as an integer variable; it will likely require some transformation or careful processing.
- As the movie titles are entered as a character string with the year of release in parentheses following the title, the year will need to be extracted from this variable in order to be used in any models.
- As the genre is stored in a pipe-separated list representing the relevant combination of 19 genre options, there are several manners in which this variable could be treated; for instance, the overall combinations could be considered separate categories or the individual genre options could be considered features of each applicable movie.

Table 1: Examination of the edx Data Structure

| 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 |
| 1 | 356 | 5 | 838983653 | Forrest Gump (1994) | Comedy\|Drama\|Romance\|War |
| 1 | 362 | 5 | 838984885 | Jungle Book, The (1994) | Adventure\|Children\|Romance |
| 1 | 364 | 5 | 838983707 | Lion King, The (1994) | Adventure\|Animation\|Children\|Drama\|Musical |
| 1 | 370 | 5 | 838984596 | Naked Gun 33 1/3: The Final Insult (1994) | Action\|Comedy |

## 2.2 Data Transformations

Based on the current state of the variables, several tranformations of the data were undertaken. First, the timestamp variable was transformed from the seconds since midnight UTC January 1, 1970, into a date variable. Converting this variable to represent the date of the rating will aid in the examination and interpretation of any possible effect of timing of the ratings.

Second, the date of the movie release was extracted from the title variable. In the current form, with the date included in the title (a character string variable), the variable does not facilitate the examination of any possible effect of the age of the movie. As a result, the dates were extracted into a seperate variable that represents the year the movie was first released.

Finally, the difference between the year of the rating and the year of the movie release were calculated based on the above two variables. Thus, this variable represents the number of years between the movie release and user rating of the movie. The code to complete these transformations is available in the supplemental file. After running this code, the data was checked for invalid and missing dates, as well as other errors.

In addition, the dataset could be converted from its current format in which a given row represents a single rating from a single user for a single movie. The dataset could be presented as a large matrix with users represented in the rows and movies represented in the columns. However, this dataset would be quite sparse, containing many missing values, as each user has not rated every single movie (i.e., 69,878 users have rated 10,677 movies). Moreover, due to the size of the edx dataset, reformatting the dataset into this format would be considerably time consuming and require considerable memory. As such, the dataset was not converted into this matrix format (see the "Limitations" section).

# 3 Exploratory Data Analysis

Table 2 presents the number of unique users and unique movies in the edx dataset. Although, the dataset contains nearly 70,000 user rating of over 10,000 movies, these values are slightly lower than the total number of unique users and movies in the MovieLens dataset described above, which is consistent with the data being partitioning into the edx (90%) and valdiation (10%) datasets.

Table 2: Number of Users and Movies

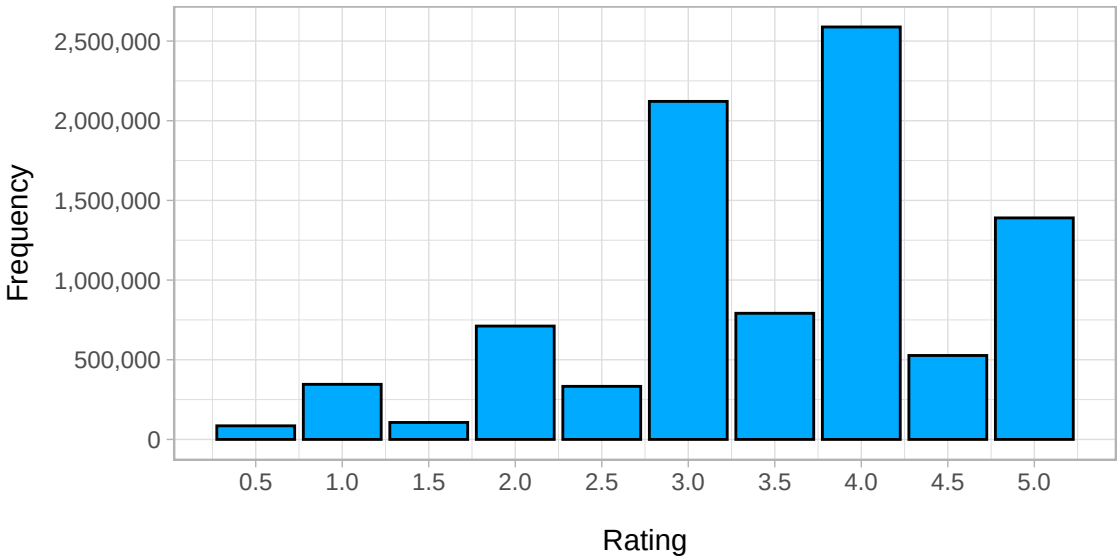
| Users | Movies |
|---|---|
| 69878 | 10677 |

Table 3 shows the frequency of each rating in descending order. From this table, we see that the minimum rating is 0.5 and the maximum is 5.0, with 4.0 being the most common rating. As well, Figure 1 displays the distrubtion of ratings. As seen, users tend to rate movies rather favourably. Generally, whole-star ratings are more common than half-star ratings. Generally, the most common ratings (in descending order) are 4.0, 3.0, and 5.0, while ratings of 0.5 and 1.5 are relatively rare. In particular, the mean rating is 3.512, the median is 4.000, and the standard deviation is 1.060.

Table 3: Frequency of Ratings

| rating | number |
|---|---|
| 4.0 | 2588430 |
| 3.0 | 2121240 |
| 5.0 | 1390114 |
| 3.5 | 791624 |
| 2.0 | 711422 |
| 4.5 | 526736 |
| 1.0 | 345679 |
| 2.5 | 333010 |
| 1.5 | 106426 |
| 0.5 | 85374 |

Figure 1: Frequency Distribution of Ratings



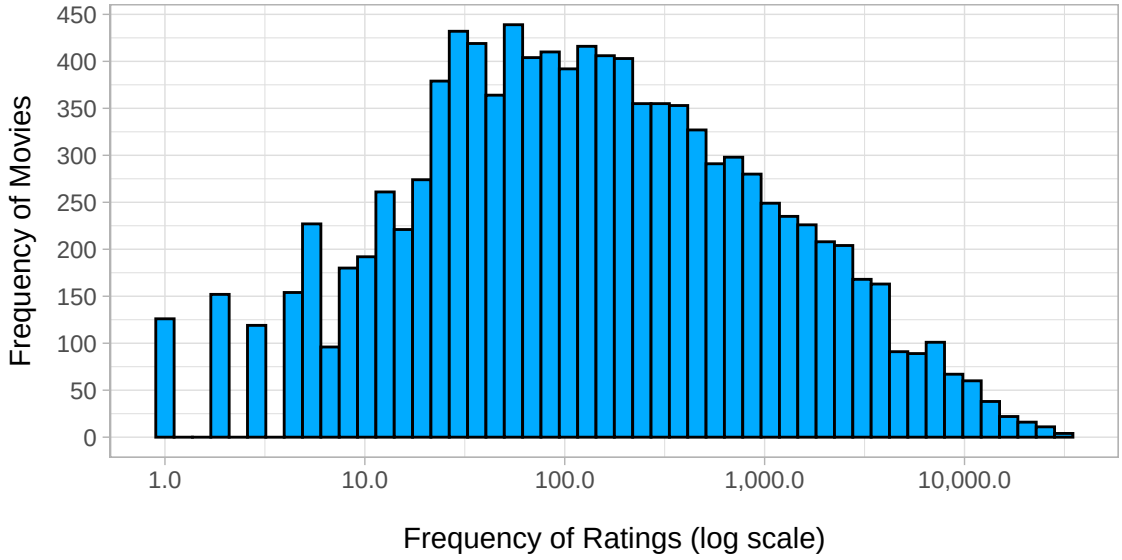
Overall, the dataset contains a number of variables that might be useful in predicting ratings:

- Movies may have an impact on ratings
- Users may have an impact on ratings
- Year of release may impact ratings
- Date of the rating may impact ratings
- Movie genre may impact ratings

## 3.1 Movie Effects

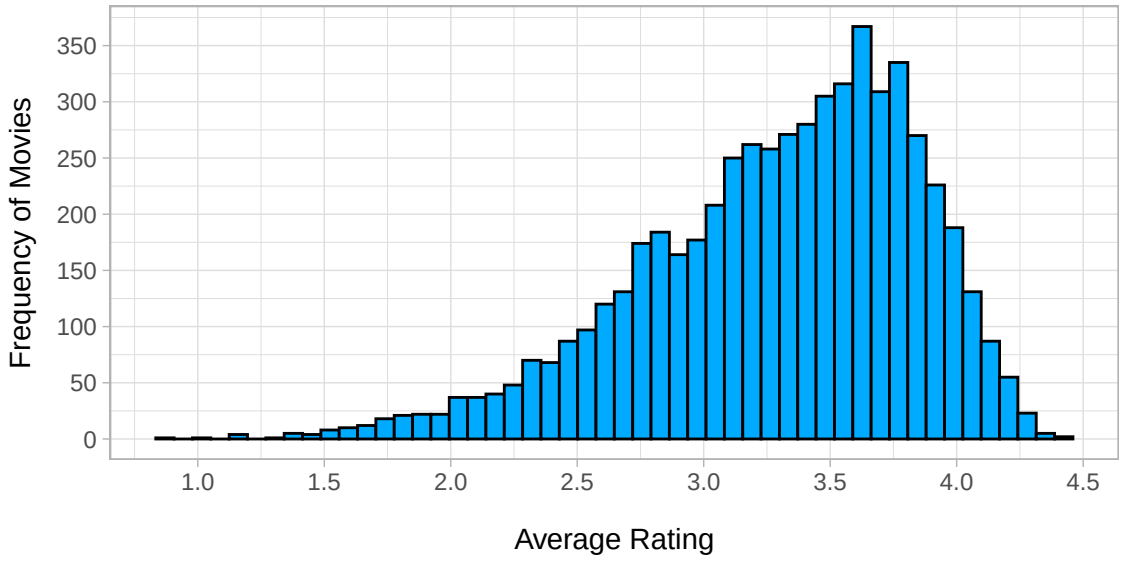
Figure 2 presents the frequency at which each movie was rated, specifically showing the frequency of ratings by frequency of movies. The graph shows ratings in log scale to highlight the pattern. Based on this histogram, we can see that some movies are rated much more frequently than others, with some movies receiving only a single rating and other receiving over 30,000 ratings. In general, the majority of movies have received somewhere between 10 and 1000 ratings.

Figure 2: Frequency of Movie Ratings (log scale)




As well, Figure 3 presents the average movie rating (for movies that have been rated at least 100 times). In general, most movies have an average rating between 2.5 and 4.0; however, as seen in the graph, there is considerable varition in the average movie rating. That is, movies differ substantially in their average ratings. Some movies tend to receive much lower (or higher) ratings than others. Due to the variability in average movie rating, the predictive models may benefit from including a term to represent a movie effect or bias.

Figure 3: Frequency of Average Movie Rating




Although average ratings vary across movies, the frequency upon which each of these averages is based also varies considerably. As seen in Figure 2, some movies have received relatively few ratings compared to others. Notably, 125 movies have only been rated a single time. The variability in the frequency at which each movie has been rated is exemplified in Tables 4 and 5, which present the most and least rated movies. The variability in the number of ratings per movie may have an important impact in the development of our predictive model, as estimates of the average movie ratings will be based on substantially different numbers of ratings. The averages based on low numbers of ratings provide poorer estimates of the true means compared to those based on a large number of ratings.

Table 4: Most Rated Movies

| **movieId** | **title** | **number** |
|---|---|---|
| 296 | Pulp Fiction (1994) | 31362 |
| 356 | Forrest Gump (1994) | 31079 |
| 593 | Silence of the Lambs, The (1991) | 30382 |
| 480 | Jurassic Park (1993) | 29360 |
| 318 | Shawshank Redemption, The (1994) | 28015 |
| 110 | Braveheart (1995) | 26212 |
| 457 | Fugitive, The (1993) | 25998 |
| 589 | Terminator 2: Judgment Day (1991) | 25984 |
| 260 | Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977) | 25672 |
| 150 | Apollo 13 (1995) | 24284 |

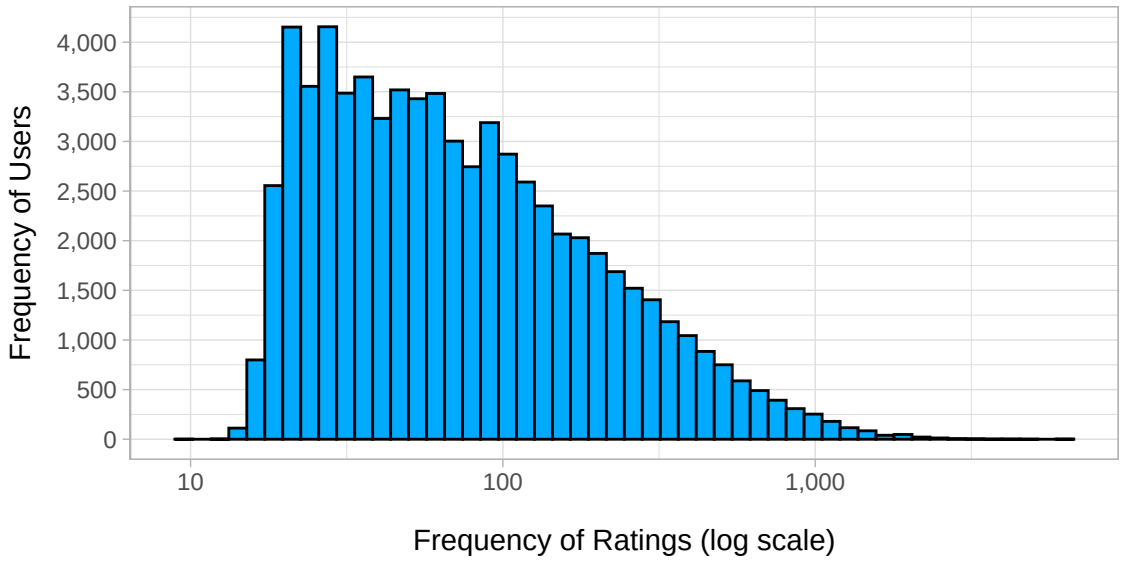
## 3.2 User Effects

Now looking at the effect of users on the ratings, Figure 4 presents the frequency at which each user provided ratings (i.e., frequency of ratings by frequency of users). The graph shows ratings in log scale to highlight the pattern. Based on this histogram, we see the same pattern for users that is present for movies: large variability in the number of ratings per user.

Table 5: Least Rated Movies

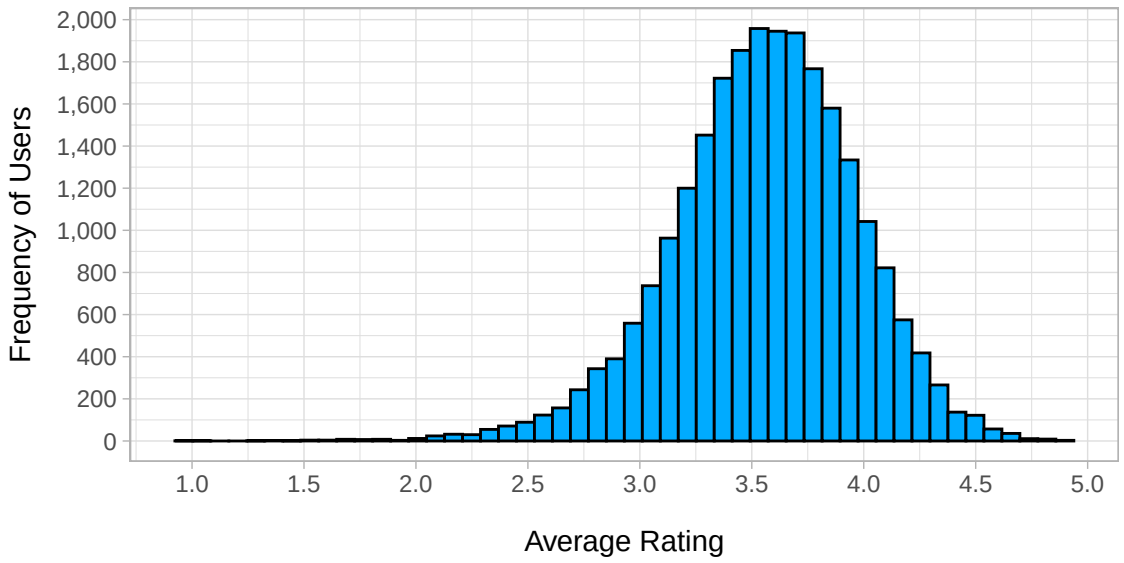
| movieId | title | number |
|---|---|---|
| 3191 | Quarry, The (1998) | 1 |
| 3226 | Hellhounds on My Trail (1999) | 1 |
| 3234 | Train Ride to Hollywood (1978) | 1 |
| 3356 | Condo Painting (2000) | 1 |
| 3383 | Big Fella (1937) | 1 |
| 3561 | Stacy's Knights (1982) | 1 |
| 3583 | Black Tights (1-2-3-4 ou Les Collants noirs) (1960) | 1 |
| 4071 | Dog Run (1996) | 1 |
| 4075 | Monkey's Tale, A (Les Château des singes) (1999) | 1 |
| 4820 | Won't Anybody Listen? (2000) | 1 |

Figure 4: Frequency of User Ratings (log scale)




Additionally, Figure 5 presents the average user rating (for users that have rated at least 100 movies). As seen in the graph, most users have an average rating between 3.00 and 4.25. Once again, it is evident that users differ substantially in how critical they are in providing their ratings. Some users tend to provide much lower (or higher) ratings on average compared to others. As a result, a user effect accounting for these differences in average user rating may be useful in the predictive models.

Figure 5: Frequency of Average User Rating




Similar to the case with movies, there is variability in average user rating, but some of these averages are based on relatively few observations. In general, the majority of users have provided between 30 and 500 ratings. Notably, four users have provided less than 14 ratings and 116 users have provided less than 16 ratings, whereas five users have provided more than 4000 ratings. This variability is particularly noticeable in Tables 6 and 7 which present the users with the most and least ratings, respectively.

Table 6: Most Active Users

| **userId** | **number** |
|---|---|
| 59269 | 6616 |
| 67385 | 6360 |
| 14463 | 4648 |
| 68259 | 4036 |
| 27468 | 4023 |
| 19635 | 3771 |
| 3817 | 3733 |
| 63134 | 3371 |
| 58357 | 3361 |
| 27584 | 3142 |

As is the case with the ratings per movie, the variability in the number of ratings per user may have an important impact in the development of our predictive model. The confidence in each of the average user ratings will vary considerably due to the substantially different numbers of ratings per user, with some estimates being based on relatively low numbers of ratings.

Due to the large variability in the number of ratings per movie and per user, the predictive models will likely benefit from the inclusion of regularization. Regularization involves applying a weight to the main effect terms (e.g., the movie and user effects) in order to control for differences in the number of ratings used to calculate the user and movie averages. In particular, averages based on low numbers of ratings are penalized more due to the relative inaccuracy of these estimates, while the weight or penalty term has little influence on the estimates based on large numbers of observations.

Table 7: Least Active Users

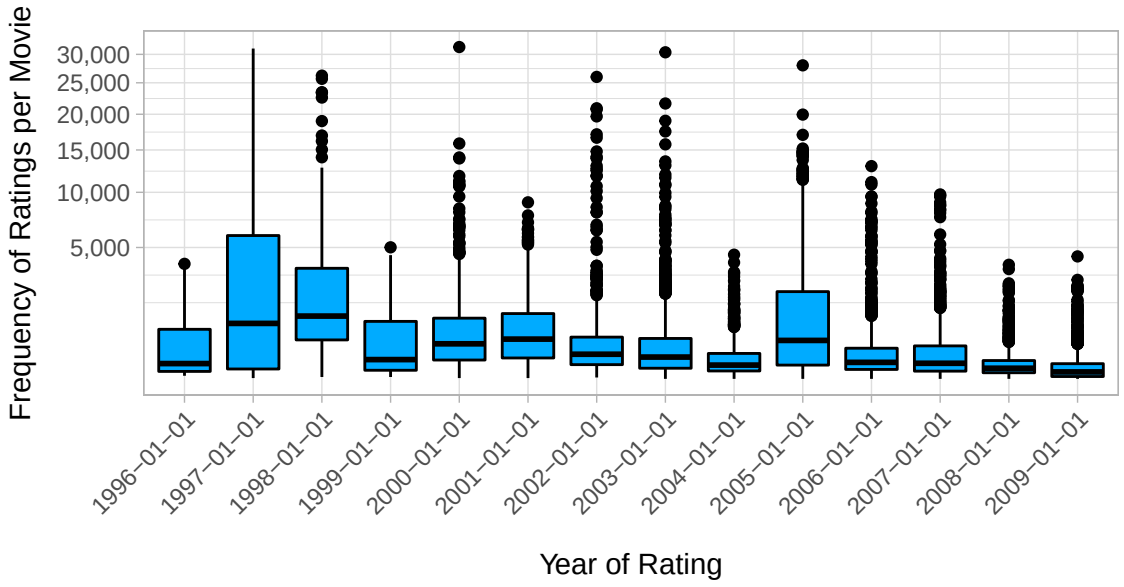
| userId | number |
|---|---|
| 62516 | 10 |
| 22170 | 12 |
| 15719 | 13 |
| 50608 | 13 |
| 901 | 14 |
| 1833 | 14 |
| 2476 | 14 |
| 5214 | 14 |
| 9689 | 14 |
| 10364 | 14 |

### 3.3 Time Effects

The impact of time on ratings can be examined in a number of ways, including the age of the rating, the age of the movie, or the difference in time between the release of the movie and the rating.

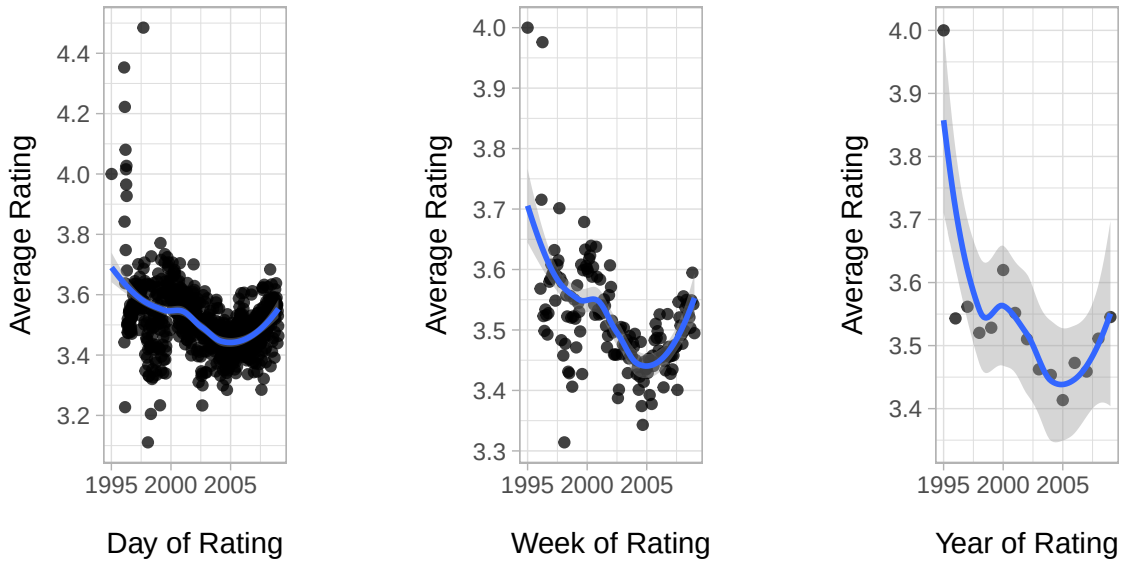
First, the date or age of the rating was explored. As seen in Figure 6, which shows the frequency of ratings per movie per year, the number of ratings per movie per year varies considerably. As such, the average rating per year is based on relatively different frequencies of ratings. The averages based on the larger numbers of observations provide better estimates of the true means relative to those based on relatively few observations.

Figure 6: Frequency of Ratings per Movie by Year of Rating




The plots in Figure 7 present the average rating across time, averaging across different time periods (i.e., weeks, months, and years). From these graphs, it is evident that the average rating varies slightly over time. For instance, looking at the pattern in the bottom graph with the date of rating rounded to the year, it can be seen that the average movie rated dropped from approximately 3.85 in 1995 to a low of approximately 3.45 in 2005 before rising again slightly to 3.55 in 2010. As such, the predictive models may benefit from the inclusion of term accounting for the date of the rating.

Figure 7: Average Rating Across Time




Next, the year of release was examined. Figure 8 presents the frequency of ratings by year of movie release, while Figure 9 presents the average rating by year of release. Based on Figure 8, it can be seen that the frequency of ratings varies across the release year of the movie. Movies released in the 1980s and 1990s have received the most ratings, while older movies tend to receive fewer and fewer ratings (with the exception of some movies released in the 1940s). Although movies in the 1980s and 1990s received the greatest number of ratings and older movies received relatively few ratings, the average movie rating presented in Figure 9 shows a different pattern. The average movie appears to depict a non-linear relationship that rises from the 1920s peaking in the 1940s before falling back down beginning in the 1960s.

Figure 8: Frequency of Ratings by Release Year

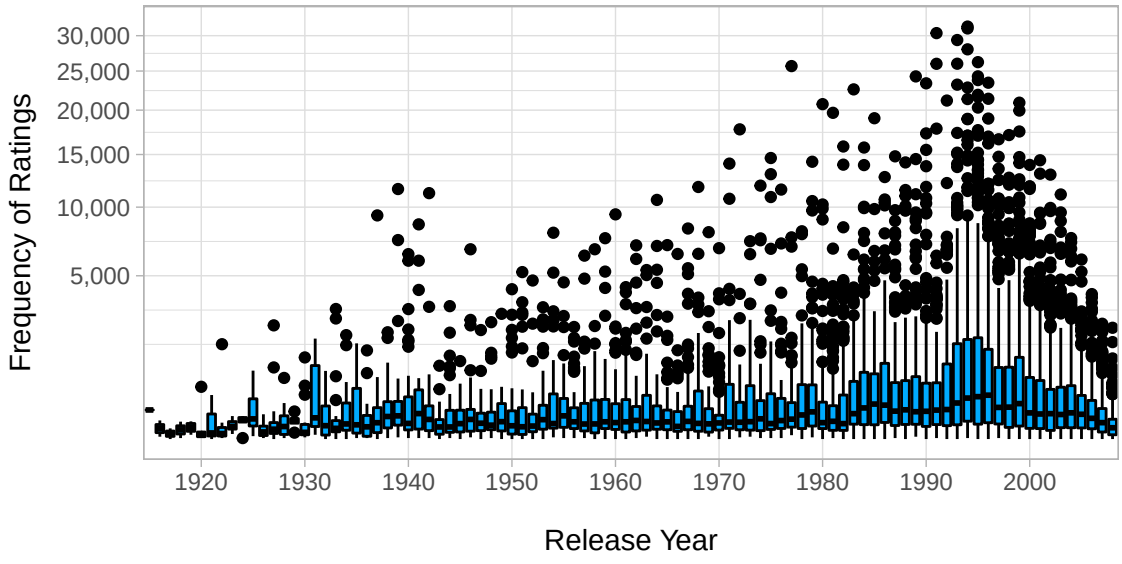
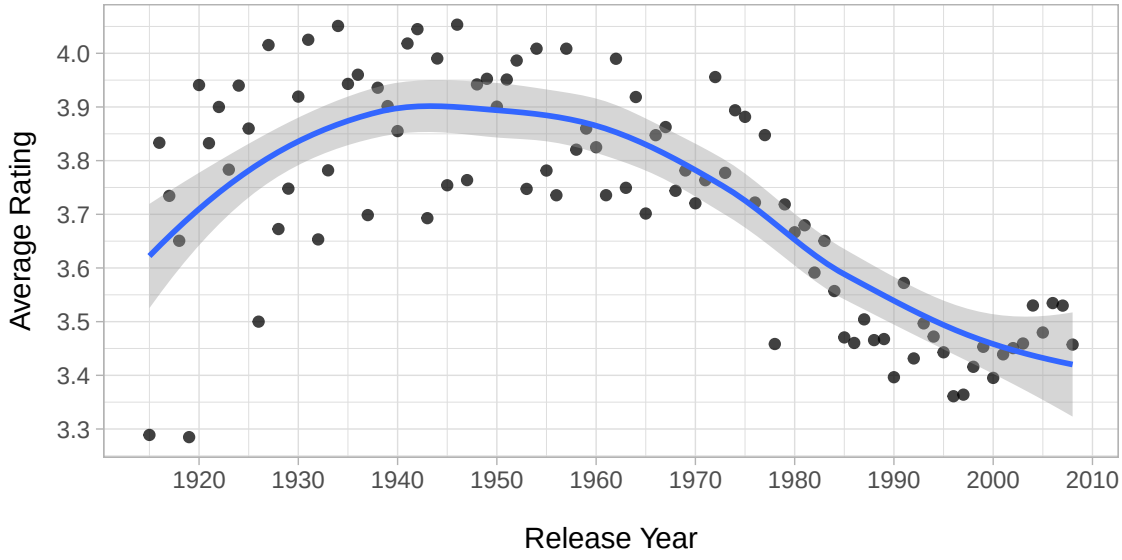
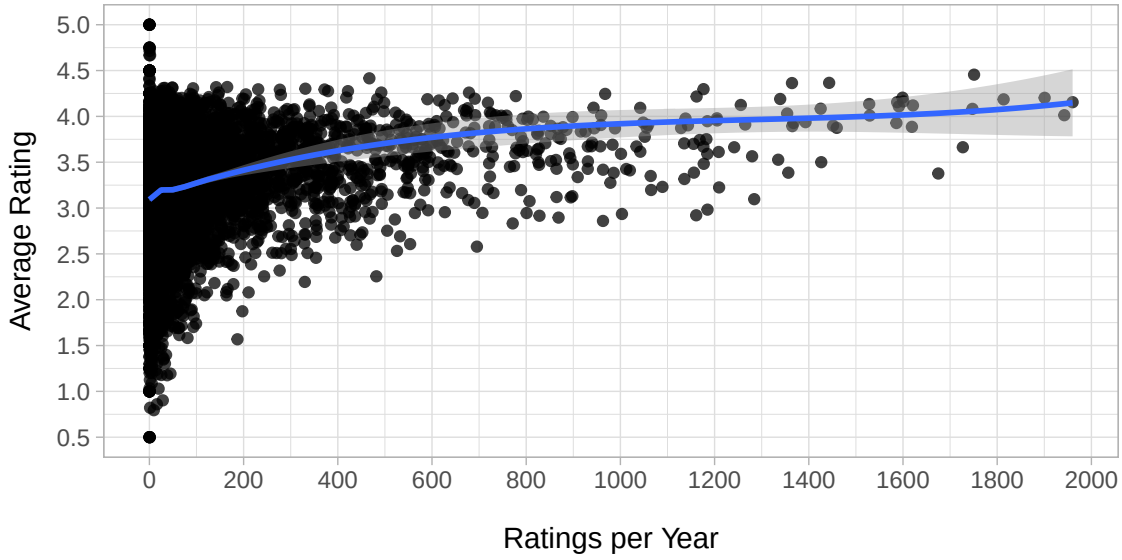


Figure 9: Average Rating by Release Year




Additionally, the impact of the time was examined in Figure 10. This figure presents the average movie rating by the number of ratings per year. Generally, as the number of ratings per year increases, so does the average rating. That is, movies with the fewest ratings per year tend to have an avergae rating just over 3.0, while the movies with the most ratings per year tend to have an average rating over 4.0.

Figure 10: Average Rating by Ratings per Year




The difference in average rating between the movies with the most ratings per year compared to the movies with the least ratings per year can be seen in Tables 8 and 9. These tables present the movies with the most and least ratings per year, respectively. Notice the considerable differences in the number of ratings (n),

average rating (rating), and ratings per year (rate) between Table 8, with the highest ratings per year, and Table 9, with the lowest.

Table 8: Movies with the Most Ratings per Year

| movieId | n | years | title | rating | rate |
|---|---|---|---|---|---|
| 296 | 31362 | 16 | Pulp Fiction (1994) | 4.15479 | 1960.12 |
| 356 | 31079 | 16 | Forrest Gump (1994) | 4.01282 | 1942.44 |
| 2571 | 20908 | 11 | Matrix, The (1999) | 4.20258 | 1900.73 |
| 2858 | 19950 | 11 | American Beauty (1999) | 4.18566 | 1813.64 |
| 318 | 28015 | 16 | Shawshank Redemption, The (1994) | 4.45513 | 1750.94 |
| 110 | 26212 | 15 | Braveheart (1995) | 4.08185 | 1747.47 |
| 480 | 29360 | 17 | Jurassic Park (1993) | 3.66352 | 1727.06 |
| 780 | 23449 | 14 | Independence Day (a.k.a. ID4) (1996) | 3.37690 | 1674.93 |
| 5952 | 12969 | 8 | Lord of the Rings: The Two Towers, The (2002) | 4.11940 | 1621.12 |
| 150 | 24284 | 15 | Apollo 13 (1995) | 3.88579 | 1618.93 |

Table 9: Movies with the Least Ratings per Year

| movieId | n | years | title | rating | rate |
|---|---|---|---|---|---|
| 48374 | 1 | 93 | Father Sergius (Otets Sergiy) (1917) | 3.0 | 0.010753 |
| 45707 | 1 | 89 | Ace of Hearts, The (1921) | 3.5 | 0.011236 |
| 63688 | 1 | 83 | Gaucho, The (1927) | 3.5 | 0.012048 |
| 64897 | 1 | 83 | Mr. Wu (1927) | 3.0 | 0.012048 |
| 59680 | 1 | 78 | One Hour with You (1932) | 3.0 | 0.012821 |
| 64275 | 1 | 78 | Blue Light, The (Das Blaue Licht) (1932) | 5.0 | 0.012821 |
| 65027 | 1 | 77 | Death Kiss, The (1933) | 2.5 | 0.012987 |
| 48649 | 1 | 76 | Chapayev (1934) | 1.5 | 0.013158 |
| 3383 | 1 | 73 | Big Fella (1937) | 3.0 | 0.013699 |
| 64903 | 1 | 67 | Nazis Strike, The (Why We Fight, 2) (1943) | 3.5 | 0.014925 |
| 64926 | 1 | 67 | Battle of Russia, The (Why We Fight, 5) (1943) | 3.5 | 0.014925 |

Finally, the impact of time on ratings was investigated by examining the relative age of the ratings (i.e., the number of years between the date of the movie release rating and the date of the rating, calculated by substracting the year of release from the year of the rating). Figure 11 presents the frequency of ratings per movie by the average relative age of the rating, while Figure 12 presents the average rating by average relative age of rating. As seen in these graphs, most ratings were provided relatively closely to the release of the movie (i.e., with an average relative age of rating less than 20 years). As well, it appears there is a relationship between the average rating and the relative age of the ratings. Generally, relatively older ratings (i.e., ratings provided longer after the release) have a slightly higher mean than younger ratings.

Figure 11: Frequency of Ratings by Average Relative Age of Rating

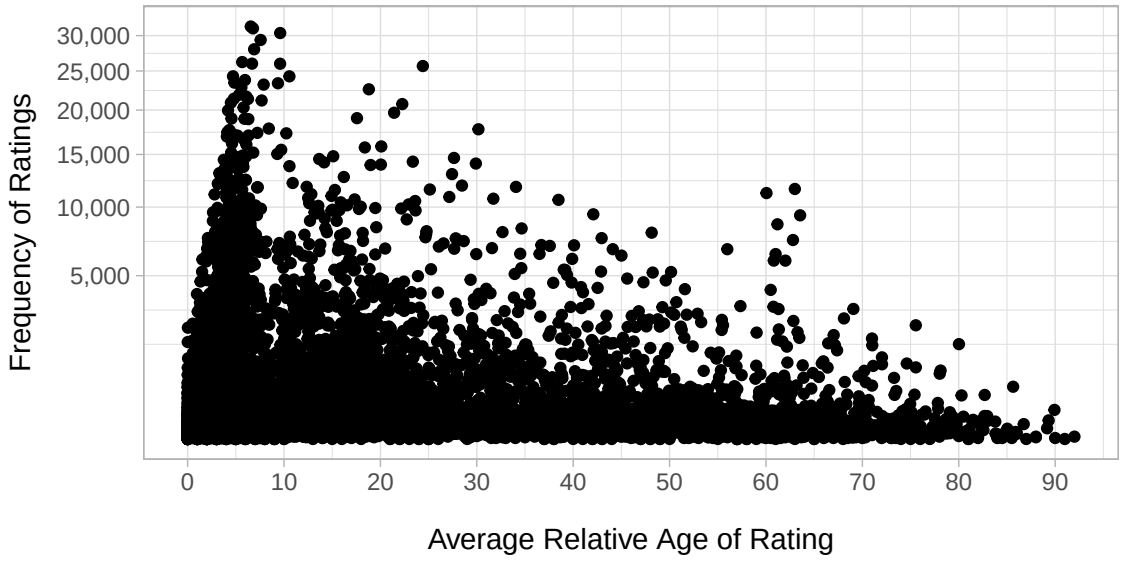
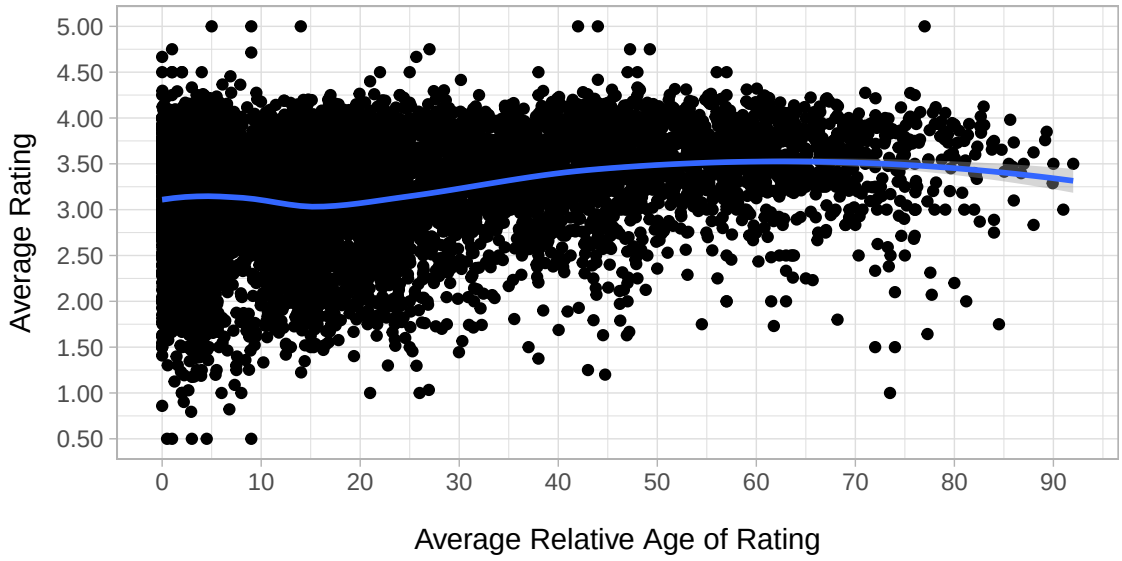



Figure 12: Average Rating by Average Relative Age of Rating




### 3.4 Genre Effects

As described above, the genre information consists of a pipe separated string variable that references all of the relevant genres for a given movie. Under this system, some movies receive only a single genre tag, while others receive multiple genre tags. This can be seen in Table 10, which presents the number and average ratings for the ten most common genre combinations. In contrast, Table 11 presents the number and average ratings for each of the ten most common individual genre tags. Due to the manner in which the genre infromation is stored, two main possibilities present themselves: treat the overall genre combinations as separate categories, or treat each individual genre tag as a separate attribute of each movie.

First, the overall genre combinations were examined (i.e., treating each specific combination of genre tags

Table 10: Ratings by Overall Genre Combination

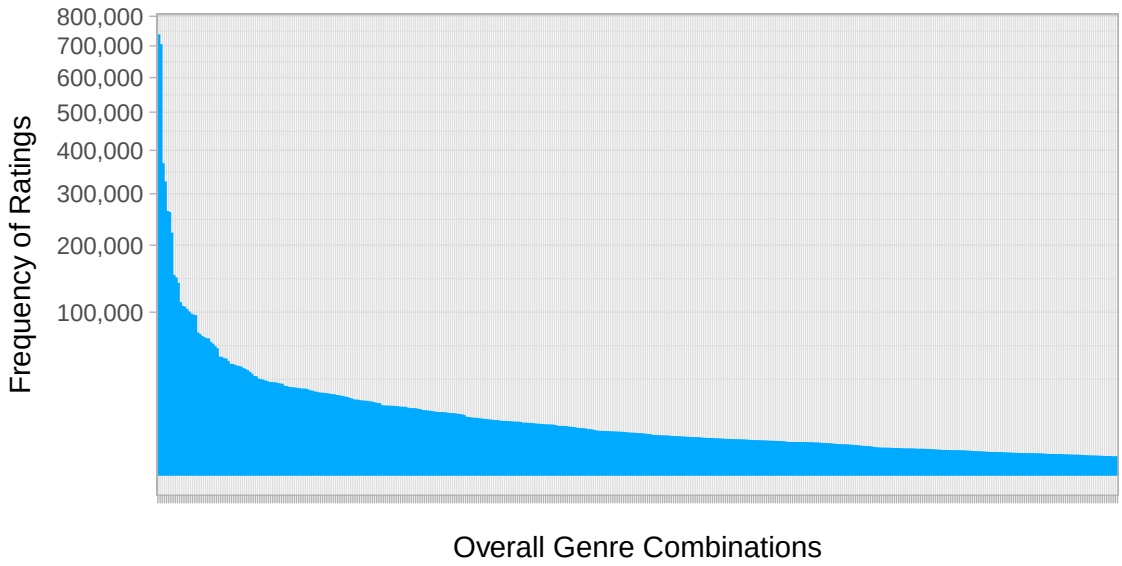
| genres | number | average |
|---|---|---|
| Drama | 733296 | 3.71236 |
| Comedy | 700889 | 3.23786 |
| Comedy\|Romance | 365468 | 3.41449 |
| Comedy\|Drama | 323637 | 3.59896 |
| Comedy\|Drama\|Romance | 261425 | 3.64582 |
| Drama\|Romance | 259355 | 3.60547 |
| Action\|Adventure\|Sci-Fi | 219938 | 3.50741 |
| Action\|Adventure\|Thriller | 149091 | 3.43410 |
| Drama\|Thriller | 145373 | 3.44634 |
| Crime\|Drama | 137387 | 3.94713 |

Table 11: Ratings by Individual Genre Tags

| genres | number | average |
|---|---|---|
| Drama | 3910127 | 3.67313 |
| Comedy | 3540930 | 3.43691 |
| Action | 2560545 | 3.42141 |
| Thriller | 2325899 | 3.50768 |
| Adventure | 1908892 | 3.49354 |
| Romance | 1712100 | 3.55381 |
| Sci-Fi | 1341183 | 3.39574 |
| Crime | 1327715 | 3.66593 |
| Fantasy | 925637 | 3.50195 |
| Children | 737994 | 3.41872 |
| Horror | 691485 | 3.26981 |
| Mystery | 568332 | 3.67700 |
| War | 511147 | 3.78081 |
| Animation | 467168 | 3.60064 |
| Musical | 433080 | 3.56331 |
| Western | 189394 | 3.55592 |
| Film-Noir | 118541 | 4.01163 |
| Documentary | 93066 | 3.78349 |
| IMAX | 8181 | 3.76769 |
| (no genres listed) | 7 | 3.64286 |

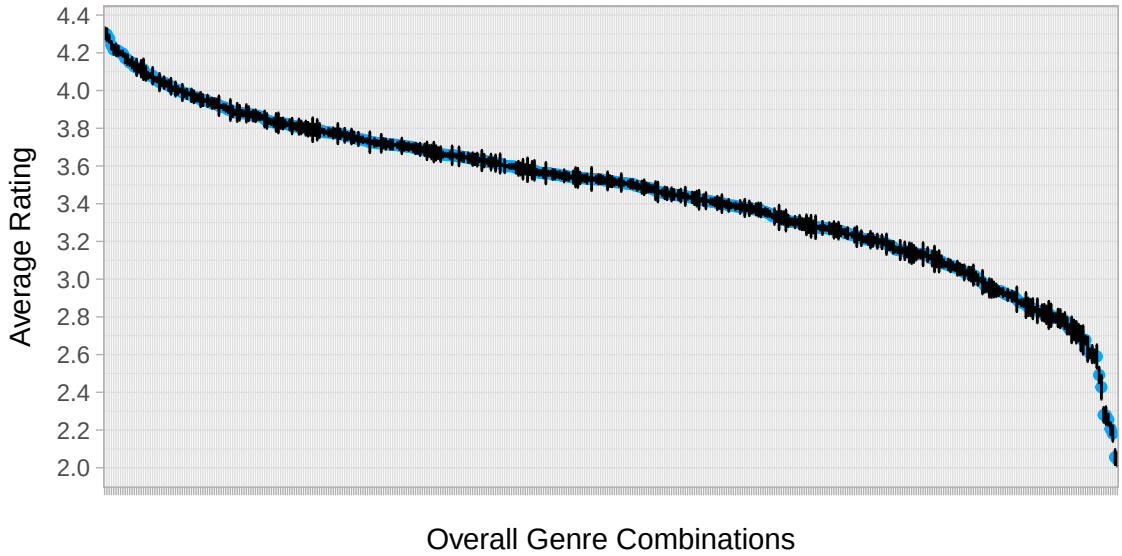
as a separate category). Considered in this manner, there are 797 unique genre combinations. Figure 13 presents the frequency of ratings for all of the genre combinations with at least 1000 ratings in descending order. As seen in this figure and Table 10, two genre combinations have received nearly twice as many ratings as all of the other combinations (i.e., “Drama” and “Comedy”). The top six categories that have received the majority of ratings are also all closely related in genre combinations.

Figure 13: Frequency of Ratings by Overall Genre Combination




As well, Figure 14 presents the average rating for all of the genre combinations (for combinations with at least 1000 ratings in descending order), with error bars around the means set to plus or minus two standard errors. The average rating for some genre combinations peaks at a high of just over 4.3 compared to the lowest values of under 2.1. Therefore, overall genre combination may be a useful variable to include in the predictive models.

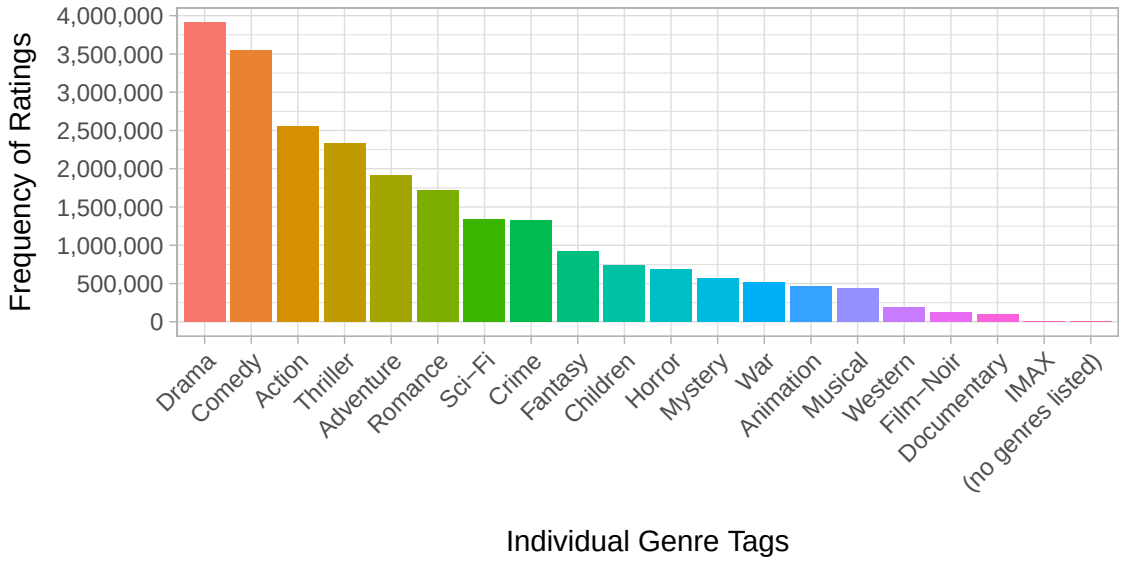
Figure 14: Average Rating by Overall Genre Combination




Next, the individual genre tags were examined (i.e., separating each overall genre combination into the individual genre tags and treating the individual tags as separate characteristics). Considered in this manner, there are 19 unique genre tags and a no genre listed category (see Table 11). Once again, two genre tags (i.e., "Drama" and "Comedy") have received far more ratings than the others, with some categories receiving

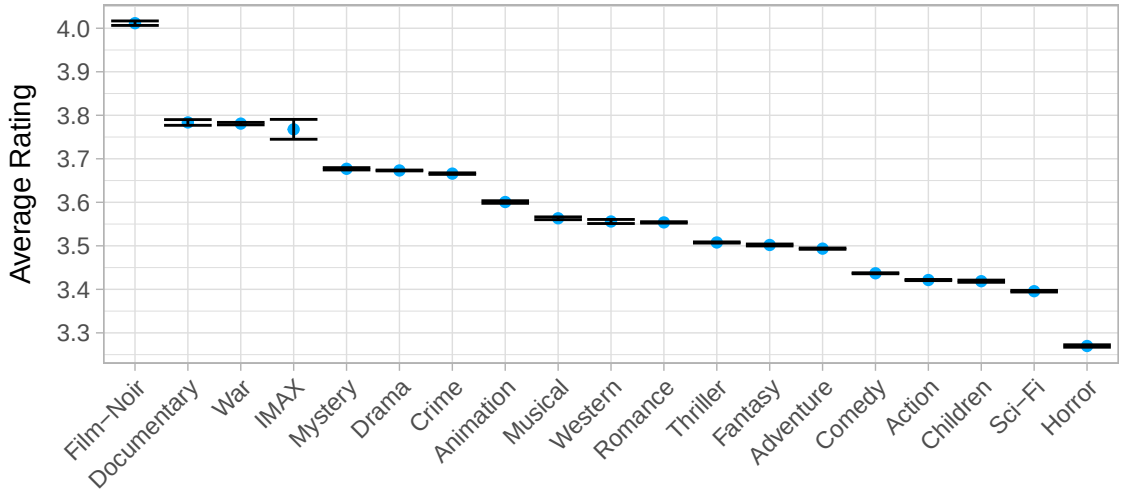
relatively few ratings (i.e., “Documentary” and “IMAX”). Figure 15 presents the frequency of ratings for each of the individual genre tags in descending order. The same pattern is apparent as before; some genre tags receiving susbstantially more ratings than others.

Figure 15: Frequency of Ratings per Individual Genres




In addition, Figure 16 shows the average rating for each of the indvidual genre tags, with error bars around the means set to plus or minus two standard errors. The average rating across the individual genre tags varies from over 4.0 for “Film-Noir” to under 3.3 for “Horror” movies. Consistent with the above examination, genre infromation appears to have some backing for inclusion in a predictive model.

Figure 16: Average Rating per Individual Genres



# 4 Modeling Approaches

Prior to building the predictive models, the edx dataset was partitioned into a training (90%) and a test dataset (10%), in a similar manner that was done previously to partition the MovieLens data into the edx and validation datasets. In this instance, the training dataset will be used to develop the predictive models and the test dataset will be used to evaluate the overall performance of the models during the training stage. As mentioned, the validation dataset (i.e., final hold-out validation dataset) will only be used to evaluate the performance of the final model during the evaluation stage. Reminder, the code used to partition the edx dataset, as well as all code for this project, is available in the supplemental code file.

Due to the size of the dataset and limitations of the operating system used, many machine learning functions were not possible. As described above, the MovieLens dataset contains over 10 million ratings (9,000,055 in the edx dataset) with over 70,000 unique users (69,878 in the edx datasets) and over 10,000 unique movies (10,677 in the edx dataset). As a result of the sheer volume of data, it was not possible to run many functions in R, including any training functions from the caret package or attempts at matrix factorization (i.e., R would crash or produce a system related error). Therefore, a linear approach using the least squares estimate was undertaken using the possible predictors identified in the exploratory analyses.

Based on the exploratory analyses described above, it appears that each of the examined variables may prove useful in trying to build the predictive algorithm for the recommendation system. Accordingly, the approach taken here will entail the additional of each variable in turn and examination of the performance of the predictive model at each stage using the test dataset.

Additionally, as described above, the RMSE will be used to evaluate the performance of the models, both during development and evaluation. In this context, the RMSE is a measure of deviation, similar to the standard deviation. It defines the error present in the predictive models. An RMSE of one means that on average the predicted values are off by one star. The aim is to develop a predictive model that produces the lowest RMSE. The code used for the RMSE function is available in the supplemental materials.

## 4.1 Model #1 - Baseline

The first model consisted of the simplest recommendation system. In order to get a baseline RMSE for comparison with other models, the first model included no predictor variables. The mean value across all ratings in the training dataset was used to predict all ratings. That is, this model predicted the average rating for all of the movies ignoring any possible effects or biases.

This model can be represented with the following formula, with

- $\mu$ equal to the overall mean of all ratings
- $\epsilon_{u,i}$ equal to the residual errors that are assumed to be independent across all ratings

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

Table 12 displays the results of the baseline model. From Table 12, we see the baseline model predicts with an average error of roughly 1.0604 stars in the test dataset. Using only the mean value for all predictions, the model was off on average by just over one star. This value was used to examine whether the inclusion of other predictors improves the predictive algorithm.

Table 12: Results in the Test Dataset

| Method | RMSE |
|---|---|
| Model #1 - Average | 1.06041 |

## 4.2 Model #2 - Movie Effects

From the exploratory analyses, it was apparent that there was considerable variability in the average movie ratings (see Figure 3). Thus, a term was included in the second model to account for these differences in average movie rating. Specifically, the average deviation of each movie from the overall mean rating was included to account for the movie effect or bias.

This model can be represented with the following formula, with

- $\mu$ and $\epsilon_{u,i}$ the same as above
- $b_i$ equal to the average deviation from $\mu$ for movie i

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

Table 13 shows the results of the second model. This model now takes into account that some movies are generally rated higher or lower than the overall average rating. The second model including a movie effect term increased the accuracy from 1.06041 to 0.94329 in the test dataset. With the addition of other predictor variables, we should be able to improve upon this model.

Table 13: Results in the Test Dataset

| Method | RMSE |
|---|---|
| Model #1 - Average | 1.060408 |
| Model #2 - Movie Effects | 0.943293 |

## 4.3 Model #3 - User Effects

The exploratory analyses also revealed that some users give higher and lower ratings than others. That is, there was variability in the average user rating (see Figure 5). In the third model, a term was included to account for these differences in average user rating. Specifically, the average deviation of each user from the overall mean and movie mean was included to account for the user effect or bias.

This model can be represented with the following formula, with

- $\mu$, $\epsilon_{u,i}$, and $b_i$ the same as above
- $b_u$ equal to the average deviation from $\mu$ minus $b_i$ for user u

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

Table 14 shows the results of the third model. This model now takes into account the variability in average movies and average user ratings. With these two predictors, the model has improved the accuracy substantially to 0.86540 in the test dataset. However, as discussed above, there is variability in the frequency of movie and user rating, with some movies and users having realtively few ratings compared to the others. As these average movie and user ratings are then based on few observations, these averages are less trustworthy than those based on relatively larger numbers of observations.

Table 14: Results in the Test Dataset

| Method | RMSE |
|---|---|
| Model #1 - Average | 1.060408 |
| Model #2 - Movie Effects | 0.943293 |
| Model #3 - Movie & User Effects | 0.865404 |

## 4.4 Model #4 - Regularized Movie and User Effects

As seen in Figure 2 and Figure 4, the number of ratings per movie and user varies considerably. Some of the averages are based on relatively small numbers of observations. Table 15 shows the movies with the largest movie effect and the number of times they were rated, while Table 16 shows the movies with the lowest movie effects. Most of these movies have been rated a single time or only a few times.

Table 15: Best 10 Movie Effects and Frequency of Ratings

| title | b_i | n |
|---|---|---|
| Hellhounds on My Trail (1999) | 1.48756 | 1 |
| Satan's Tango (Sátántangó) (1994) | 1.48756 | 2 |
| Shadows of Forgotten Ancestors (1964) | 1.48756 | 1 |
| Fighting Elegy (Kenka erejii) (1966) | 1.48756 | 1 |
| Sun Alley (Sonnenallee) (1999) | 1.48756 | 1 |
| Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1980) | 1.23756 | 4 |
| Human Condition II, The (Ningen no joken II) (1959) | 1.23756 | 4 |
| Human Condition III, The (Ningen no joken III) (1961) | 1.23756 | 4 |
| Constantine's Sword (2007) | 1.23756 | 2 |
| More (1998) | 1.15422 | 6 |

Table 16: Worst 10 Movie Effects and Frequency of Ratings

| title | b_i | n |
|---|---|---|
| Besotted (2001) | -3.01244 | 1 |
| Hi-Line, The (1999) | -3.01244 | 1 |
| Under the Lighthouse Dancing (1997) | -3.01244 | 1 |
| Accused (Anklaget) (2005) | -3.01244 | 1 |
| Confessions of a Superhero (2007) | -3.01244 | 1 |
| War of the Worlds 2: The Next Wave (2008) | -3.01244 | 2 |
| Disaster Movie (2008) | -2.72673 | 28 |
| SuperBabies: Baby Geniuses 2 (2004) | -2.68244 | 50 |
| Hip Hop Witch, Da (2000) | -2.66629 | 13 |
| From Justin to Kelly (2003) | -2.59082 | 185 |

The next model attempted to account for the differences in the number of observations across movies and users. Regularization involves the application of a weight to the movie and user effects to control for differences in the number of ratings per movie and per user. Movie and user averages based on a small number of ratings are penalized more than averages based on a large number of ratings in order to account for differences in the confidence in these estimates. Accordingly, this model included regularized movie and regularized user effects.

This model can be represented with the following formula, with

- $\mu$, $\epsilon_{u,i}$, $b_i$, and $b_u$ the same as above
- $\lambda_i$ and $\lambda_u$ equal to the penalized least square estimate weights for the movie and user effects

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

Lambda is a tuning parameter that must be selected. The optimal lambda was selected based on the resulting RMSE in the test set. Figure 17 shows a plot of possible lambda coefficients against the resulting RMSE. Based on these results, lambda is set to 5.00.

Figure 17: Selection of Optimal Lambda

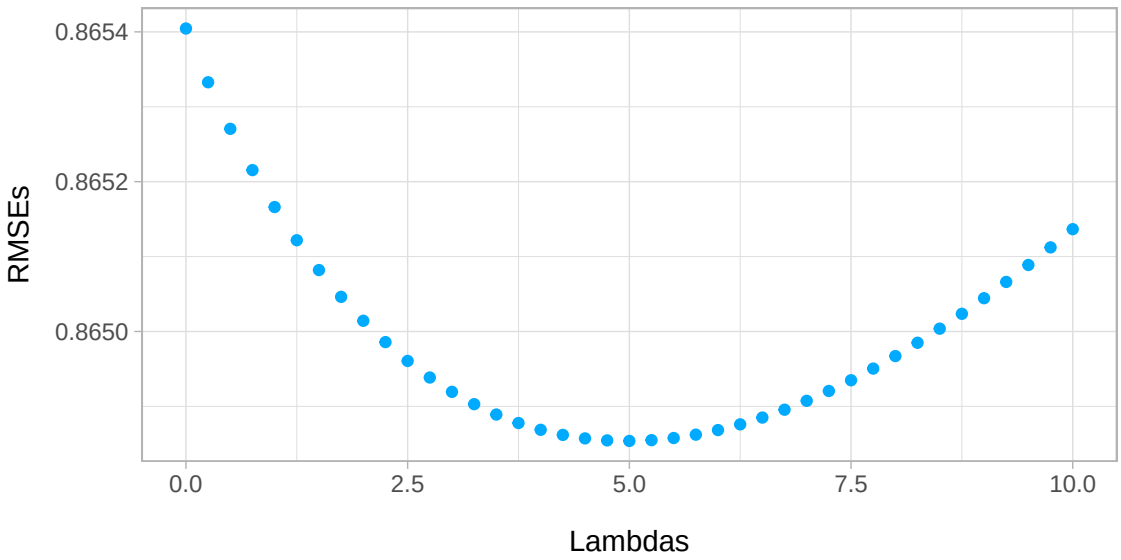



Table 17 displays the results of this model. The fourth model now takes into account the variability in average movie and average user ratings, as well as the variability in the frequencies upon which these averages are based. With these two predictors and their regularized weighting, the accuracy has improved slightly to 0.86485 in the test dataset. Nevertheless, the addition of other predictors may improve the model.

Table 17: Results in the Test Dataset

| Method | RMSE |
|---|---|
| Model #1 - Average | 1.060408 |
| Model #2 - Movie Effects | 0.943293 |
| Model #3 - Movie & User Effects | 0.865404 |
| Model #4 - Regularized Movie & User Effects | 0.864854 |

## 4.5 Model #5 - Adding Age of Movie Effects

Based on Figure 9, the model may benefit from including the year of release. There was some variability in the average rating based on the year of release. In this model a term was added to represent the age of the movie. As seen in the formula below, this term represents the average deviation for each year of release from the overall mean minus the other variables already in the model.

This model can be represented with the following formula, with

- $\mu$, $\epsilon_{u,i}$, $b_i$, and $b_u$ the same as above
- $f(y_i)$ equal to the average deviation from the overall mean minus the other effects
- Note that the regularized movie and user effects were used in the model

$$Y_{u,i} = \mu + b_i + b_u + f(y_i) + \epsilon_{u,i}$$

Table 18 presents the results of this model. With the addition of this term to account for difference in average rating across year of release, as well as the regularized movie and user effects, the model improved slightly to achieve a RMSE of 0.86456 in the test dataset.

Table 18: Results in the Test Dataset

| Method | RMSE |
|---|---|
| Model #1 - Average | 1.060408 |
| Model #2 - Movie Effects | 0.943293 |
| Model #3 - Movie & User Effects | 0.865404 |
| Model #4 - Regularized Movie & User Effects | 0.864854 |
| Model #5 - Adding Age of Movie Effects | 0.864562 |

## 4.6 Model #6 - Adding Age of Rating Effects

Based on Figure 6, the model may also benefit from including the date the rating was made. That is, there was some variability in the average rating based on the date of the rating. As a result, a term was added in this model to represent the month of the rating. This term represents the average deviation for each month of rating from the overall mean minus the other variables already in the model.

This model can be represented with the following formula, with

- $\mu$, $\epsilon_{u,i}$, $b_i$, $b_u$, and $f(y_i)$ the same as above
- $f(m_{u,i})$ equal to the average deviation from the overall mean minus the other effects
- Note that the regularized movie and user effects were used in the model

$$Y_{u,i} = \mu + b_i + b_u + f(y_i) + f(m_{u,i}) + \epsilon_{u,i}$$

Table 19 presents the results of this model. With the addition of this term to account for difference in average rating across year of rating, as well as all of the previous effects or biases, the model was able to achieve a RMSE of 0.86441 in the test dataset.

Table 19: Results in the Test Dataset

| Method | RMSE |
|---|---|
| Model #1 - Average | 1.060408 |
| Model #2 - Movie Effects | 0.943293 |
| Model #3 - Movie & User Effects | 0.865404 |
| Model #4 - Regularized Movie & User Effects | 0.864854 |
| Model #5 - Adding Age of Movie Effects | 0.864562 |
| Model #6 - Adding Age of Rating Effects | 0.864410 |

## 4.7 Model #7 - Adding Genre Effects

Finally, as seen in Figures 14 and 16, the movie genre also appeared to impact ratings. In particular, as the mean rating varied considerably across the overall genre combinations (see Figure 14), a term was added in this model to represent the overall combination of genre tags. This term represents the average deviation for the genre from the overall mean minus the other variables already in the model.

This model can be represented with the following formula, with

- $\mu$, $\epsilon_{u,i}$, $b_i$, $b_u$, $f(y_i)$, and $f(m_{u,i})$ the same as above
- $b_g$ equal to the average deviation from the overall mean minus the other effects
- Note that the regularized movie and user effects were used in the model

$$Y_{u,i} = \mu + b_i + b_u + f(y_i) + f(m_{u,i}) + b_g + \epsilon_{u,i}$$

Table 20 presents the results of all the models. With the addition of this term to account for difference in average rating across genres, as well as all of the previous effects or biases, the model was able to achieve a RMSE of 0.86413 in the test dataset. With this value of RMSE in the test dataset, this final model was subsequently evaluated in the validation dataset.

Table 20: Results in the Test Dataset

| **Method** | **RMSE** |
|---|---|
| Model #1 - Average | 1.060408 |
| Model #2 - Movie Effects | 0.943293 |
| Model #3 - Movie & User Effects | 0.865404 |
| Model #4 - Regularized Movie & User Effects | 0.864854 |
| Model #5 - Adding Age of Movie Effects | 0.864562 |
| Model #6 - Adding Age of Rating Effects | 0.864410 |
| Model #7 - Adding Genre Effects | 0.864127 |

# 5 Results on Validation Dataset

Next, the final model was evaluated in the validation dataset (i.e., the final hold-out validation dataset). Note that prior to testing the final model in the validation dataset, the transformation conducted in the edx dataset must be run in the validation dataset (see the "Data Transformations" section). Table 21 presents the results of the final predictive model in the validation dataset, compared to its performance in the test dataset. The final predictive model performed nearly as well in the validation dataset as it did in the test dataset. Specifically, the model was able to achieve a RMSE of 0.86441 in the validation dataset.

Table 21: Final Results in the Validation Dataset

| **Method** | **RMSE** |
|---|---|
| Model #1 - Average | 1.060408 |
| Model #2 - Movie Effects | 0.943293 |
| Model #3 - Movie & User Effects | 0.865404 |
| Model #4 - Regularized Movie & User Effects | 0.864854 |
| Model #5 - Adding Age of Movie Effects | 0.864562 |
| Model #6 - Adding Age of Rating Effects | 0.864410 |
| Model #7 - Adding Genre Effects | 0.864127 |
| **Final Model - Validation Results** | **0.864414** |

# 6 Discussion

The aim of the project was to develop a recommendation system using the MovieLens dataset, in a similar vein to the Netflix challenge. Specifically, code was provided to download the 10M version of the MovieLens data and separate it into an edx dataset, used for exploration and development of the predictive model, and a validation dataset, used only for evaluating the final model. The predictive algorithms were evaluated based on the RMSE. Exploratory data analysis revealed considerable variability across each of the available variables. The edx dataset was then split into a training and a test dataset and several models were built in sequence adding a new term in each model.

The final predictive model attempted to account for difference across movies, users, genres, year of release, and month of rating, as well as account for the fact that there were some users and movies that had relatively few ratings compared to others. Using these five predictors and adjusting for the low number of observations for some users and movies, this predictive model was able to perform well with a RMSE of 0.86441 in the validation dataset.

The final model can therefore be represented with the following formula, with

- $\mu$ representing the overall mean
- $b_i$ representing the average movie effect
- $\lambda_i$ representing the penalized weight applied to movies
- $b_u$ representing the average user effect
- $\lambda_u$ representing the penalized weight applied to users
- $f(y_i)$ representing the average year of release effect
- $f(m_{u,i})$ representing the average month of rating effect
- $b_g$ representing the average genre effect

$$Y_{u,i} = \mu + \lambda_i b_i + \lambda_u b_u + f(y_i) + f(m_{u,i}) + b_g + \epsilon_{u,i}$$

## 6.1 Limitations

The size of the dataset and its relative sparsity when considered as a matrix of users in the rows and movies in the columns presented certain challenges. The magnitude of the dataset prevented the use of many functions (e.g., errors on allocating vectors). As a result, this project focused on a linear approach using least squares estimates. Due to the limitations of the current operating system combined with the volume of the data, alternative approaches were not feasible. This limited the available options for developing the predictive models.

## 6.2 Future Directions

A number of avenues could be pursued in order to improve upon the final model presented here. The model may be improved by reconsideration of the manner in which some of the variables were treated. For instance, the overall combinations of genre tags were treated as separate categories in the current project, whereas they could have been separated into the individual genre tags. The time effects could have also been handled differently, such as including a term to represent the relative time from movie release to rating. Regularization could have been used separately on each variable. The current project only used regularization for the movie and user effects and required the lambda coefficients to be equal for these variables. Additionally, alternative approaches could be investigated. Machine learning techniques such as k-means clustering or random forest could be used if the operating system permits. The use of matrix factorization, factor analysis, or principal component analysis would also likely improve upon the model.

# 7 Appendix

This project and code were completed using the following specifications:

```
##                _
## platform       x86_64-apple-darwin17.0
## arch           x86_64
## os             darwin17.0
## system         x86_64, darwin17.0
## status
## major          4
## minor          0.3
## year           2020
## month          10
## day            10
## svn rev        79318
## language       R
## version.string R version 4.0.3 (2020-10-10)
## nickname       Bunny-Wunnies Freak Out
```