Edx - Data Science Capstone - Paradox of Choice and Machine Learning

Matthew Saayman

11/06/2021

Introduction

The Paradox of Choice, as described by psychologist Barry Schwartz, refers to the paralysis and anxiety induced by an over-abundance of choice. We can observe this phenomenon in the physical world, in the case of a store being stocked with countless brands, and in the digital world, in the realm of online dating or online movie streaming. The experience of some Netflix users struggling to decide the right movie to watch is hardly uncommon. This effect negatively impacts decision making with some users even eschewing a process or product altogether (https://tophotel.news/experts-voice-how-to-upsell-101-too-much-choice-can-lead-to-no-choice/). In that light, correctly predicting what a consumer will enjoy is important for human well-being.

Creating a movie recommendation system can be seen as one way to address this paradox and represents a useful study in machine learning. This paper uses a sample of the movielens dataset to build a linear regression model for movie recommendations and includes the following biases: weekday, hour of the day, year difference, genre, user, and movie.

The first section describes the dataset in greater detail and the technical challenges with working with the data. In addition, an exploration of the data is provided along with visualizations to demonstrate the existence of certain effects on ratings. The second section describes the model in greater detail and provides a justification for why the given method was used. The final section describes the results of the model on the validation dataset and concludes with a discussion on future uses for the model and its limitations.

Section 1. Methods and Analysis – Data Exploration

The movielens dataset is freely available dataset containing millions observations, with each row counting as a rating by one user of one movie. As part of the edX capstone project, a sample of one version of the Movielens dataset, containing 10 million rows, is created as “edx”, containing 9,000,055 observations including 69,878 unique users of 10,677 unique movies.

A training and dataset were then partitioned from the edx set, with 50% going to each set. The structure of the edX dataset appears in Figure 1 (below), with movieId representing the movie that was rated, userId representing the individual who rated the movie, rating (from 0.5 to 5 stars), the timestamp representing the date and time the rating was made, title of the movie, and genres of the movie.

In examining the first few rows of the edx dataset it becomes immediately clear that we will face technical challenges in building the model. Timestamp is stored as the number of seconds since 1970/01/01. Genres has multiple genres , with some movies having a single genre (ie// “Drama”) and others having multiple (“Drama | comedy”).

Figure 1

```
##    userId movieId rating timestamp                          title
## 1:      1     122      5 838985046              Boomerang (1992)
## 2:      1     185      5 838983525               Net, The (1995)
```

```
## 3:      1     292      5 838983421               Outbreak (1995)
## 4:      1     316      5 838983392               Stargate (1994)
## 5:      1     329      5 838983392 Star Trek: Generations (1994)
## 6:      1     355      5 838984474       Flintstones, The (1994)
##                          genres
## 1:                Comedy|Romance
## 2:          Action|Crime|Thriller
## 3:  Action|Drama|Sci-Fi|Thriller
## 4:       Action|Adventure|Sci-Fi
## 5: Action|Adventure|Drama|Sci-Fi
## 6:       Children|Comedy|Fantasy
```

1.2 Exploration of the time effect

Before building the model we can ask ourselves questions about the interaction of the columns in the dataset based on general knowledge of human behaviour. We can start by exploring the training data set to see evidence of "time effect", specifically looking at and manipulating the timestamp column. How are average ratings affected by:

-The time of the day?

-The day of the week?

-The current year?

Netflix's algorithm takes into account the time of day a user watches a movie (https://help.netflix.com/en/node/100639), and it seems plausible that the time of day (or week) would influence a user's mood and therefore the rating they offer a movie. In addition, as noted in the edx Machine Learning Course (6.2) assessment, the year of the movie release may play a role too. A newer movie will have a lower rating on average because fewer users have had the opportunity to rate it, compared to an older movie for which there has been additional time for users to rate it.

Accordingly, we can create new variables using the timestamp column and the title column.

-Year: extracted from the title column, this is the year the movie was released.

-Year of rating: this is the year the user made their rating.

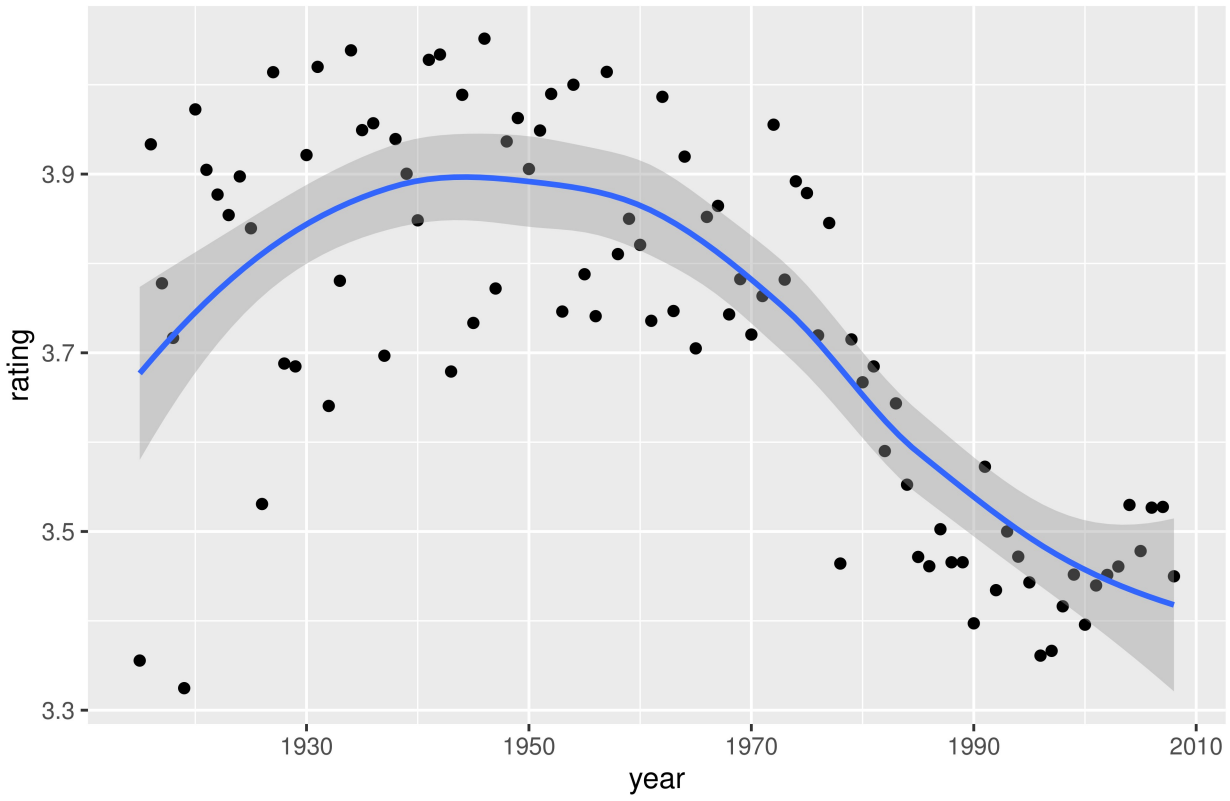
-YearDif: Year of the rating minus the year of the movie release.

-Weekday: The day of the week the rating was made (values "Sunday", "Monday", etc.).

-Hour: The time of day the rating was made (values are 0 to 23 with 0 representing midnight).

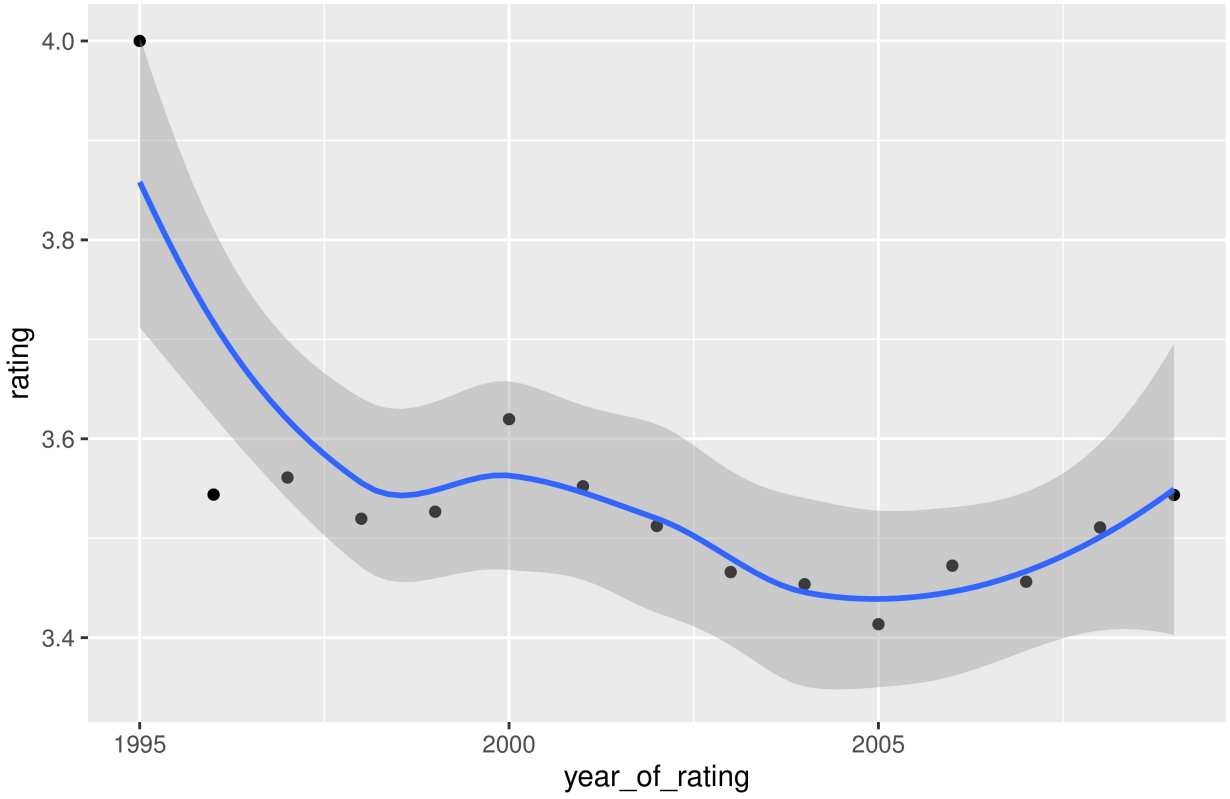
The following plot (Figure 2) shows how the average rating varies according to movie release year

Figure 2 – Average movie rating by each movie–release year



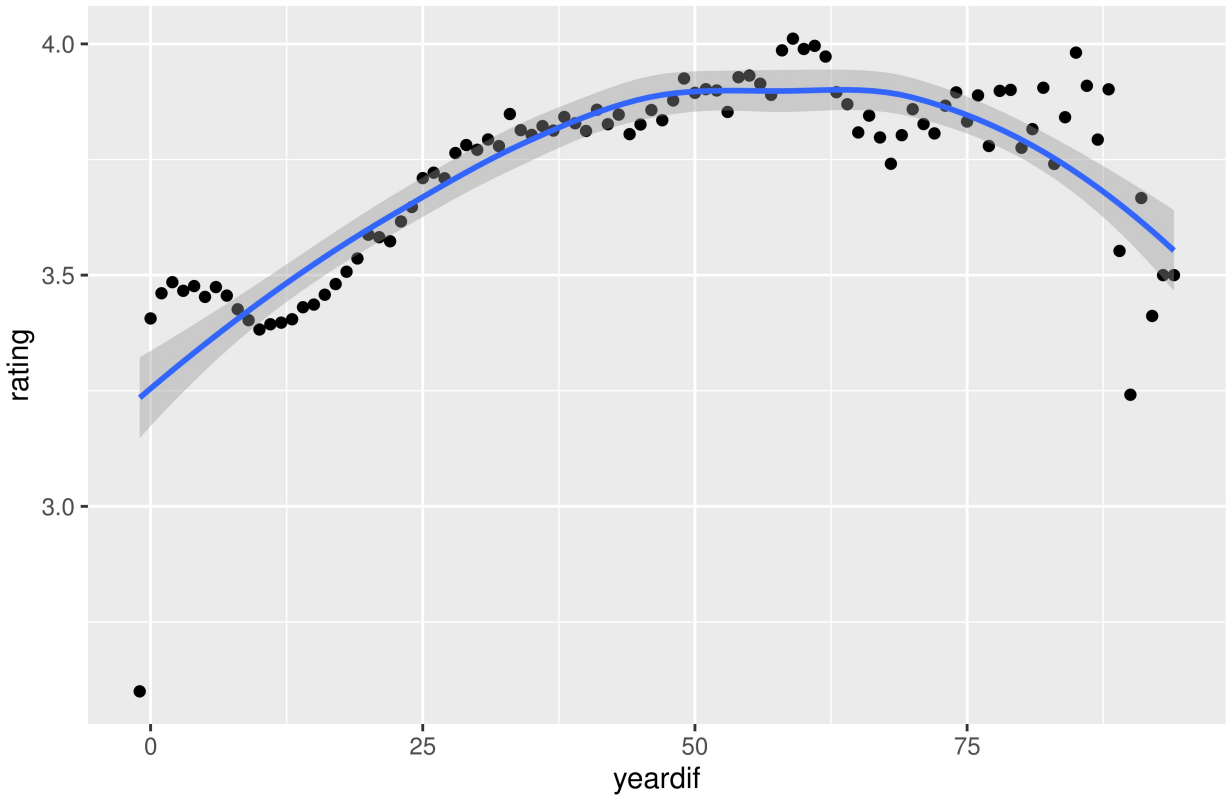
Here, each data point represents a given movie-release year and the average rating of all movie-ratings in that year, the line represents the general trend when the data is grouped by movie-release year, and the shaded area around the line represents the confidence intervals for each data point. Although the general pattern presented here suggests newer movies do indeed get lower ratings, on average, it is also clear there is much variation with so many data points outside the bounds of the confidence intervals.

Figure 3 – Average movie rating in each year of movie rating



When plotting the year of the rating, there is an alogether different pattern (Figure 3). As the dataset is for ratings from 1995 onwards, we can see that after the year 1995 the average sharply decreases each subsequent year, likely a reflection of an increased number of users in the dataset and ratings in the system. By the year 2000, the line is relatively stable which makes sense as we should hardly expect that users are more or less critical (on average) year over year.

Figure 4 – Average movie rating by year difference



When plotting the "Year Diff" column (the difference between the year of a rating and the year the movie was released), a clear pattern emerges (Figure 4). When the year difference is < 1 years between the rating year and the movie release year, the average rating is lower. But the average rating increases for each additional year, before decreasing at around 75 years. What this plot essentially demonstrates is that, just as in Figure 2, average ratings for newer movies will generally be lower than older movies, holding all else equal.There is of course variation within this plot but the data appears to more closely gather around the line than in Figure 2.

We should therefore consider adding this YearDiff variable to the model.

What about examining the time of day? It is not clear from the dataset if all users were in the same timezone or even the same geographic area when they submitted the rating. Nevertheless, a clear pattern emerges when we look at the average rating according to hour of the day.

Figure 5 – Number of ratings by hour of the day

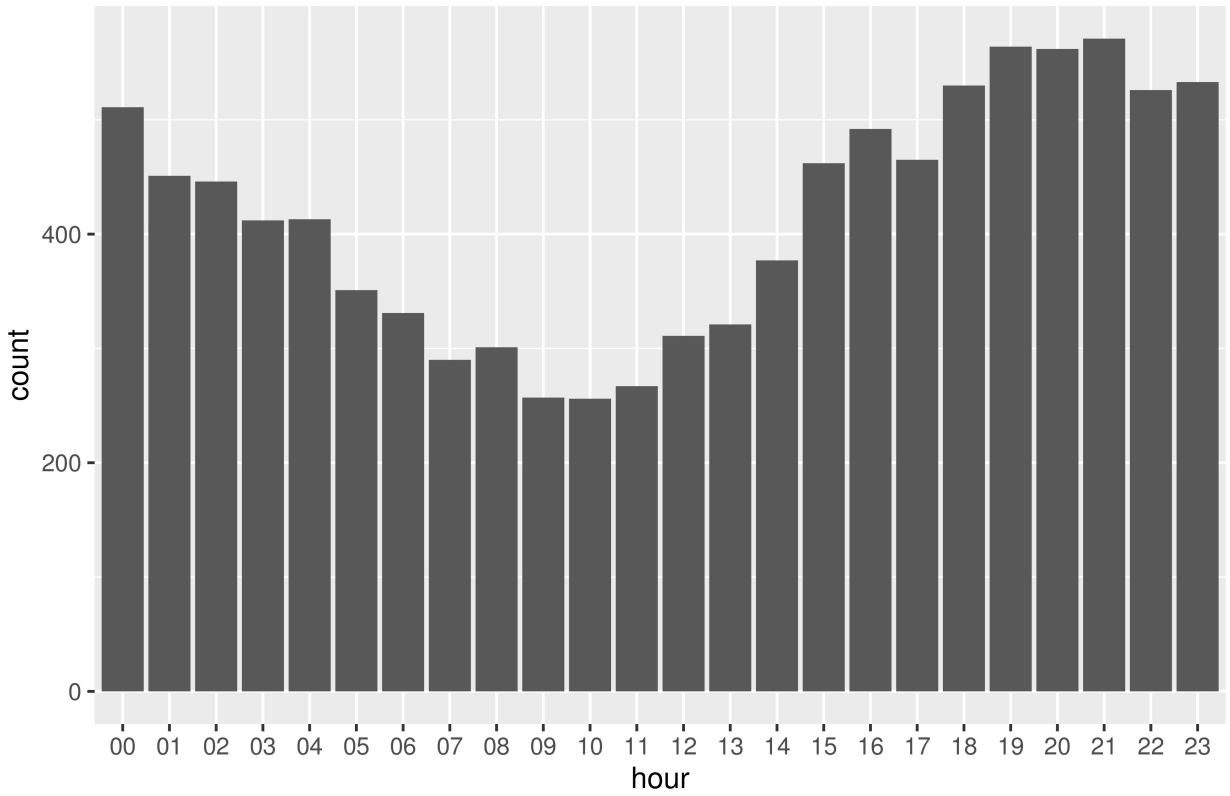
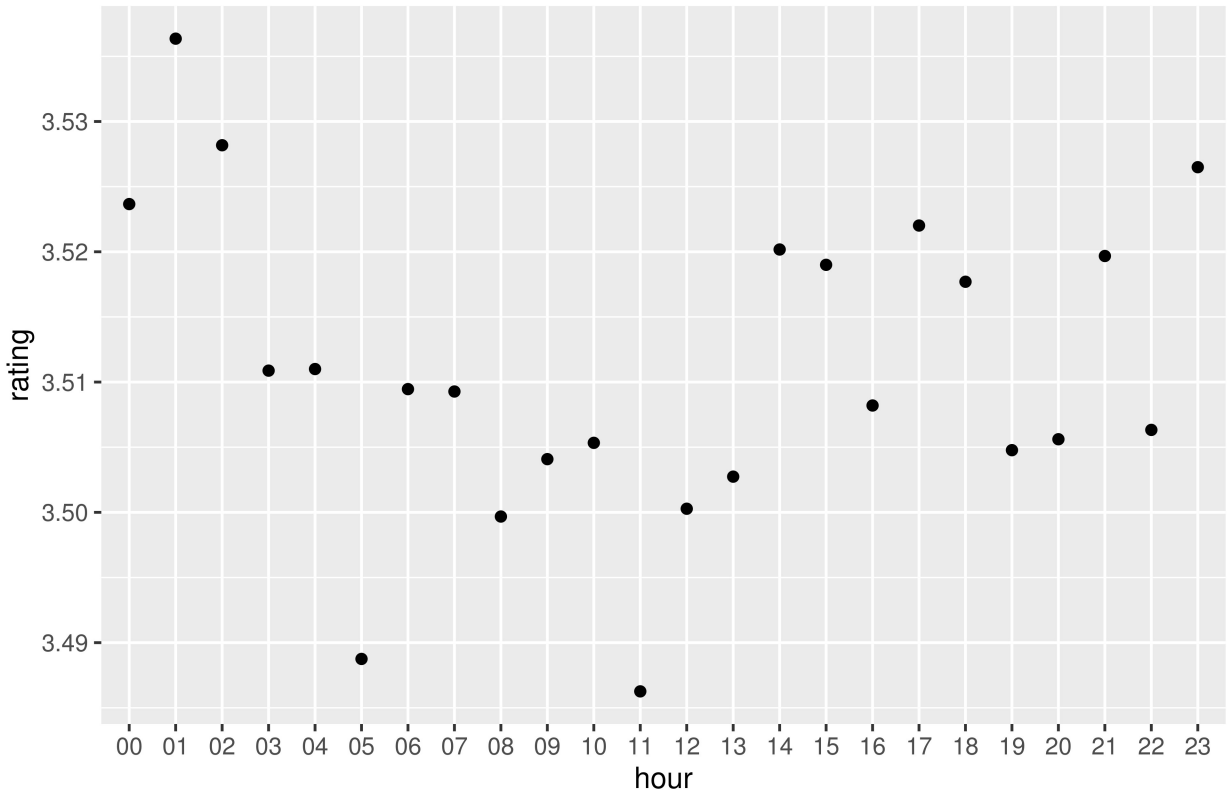


Figure 5 shows that the busiest hours for submitting ratings were between 7 and 10 (19 and 22 on the chart, respectively), with the least busy hours between 8 and 11 am. Figure 6 shows that the average rating varies somewhat according to the hour of the day.

Figure 6 – Average rating by hour of the day



This effect may not appear as strong as the "year difference" effect but we could still consider adding it to the model as well.

Finally, what about the day of the week? We should expect more users to submit more ratings on weekends, potentially affecting the rating.

Figure 7 shows that Monday and Tuesday are the busiest days for movie ratings. Figure 8 shows that the average rating differs by day as well, with the average rating highest for Saturdays and lowest for Thursdays.

Figure 7 – Number of ratings by weekday

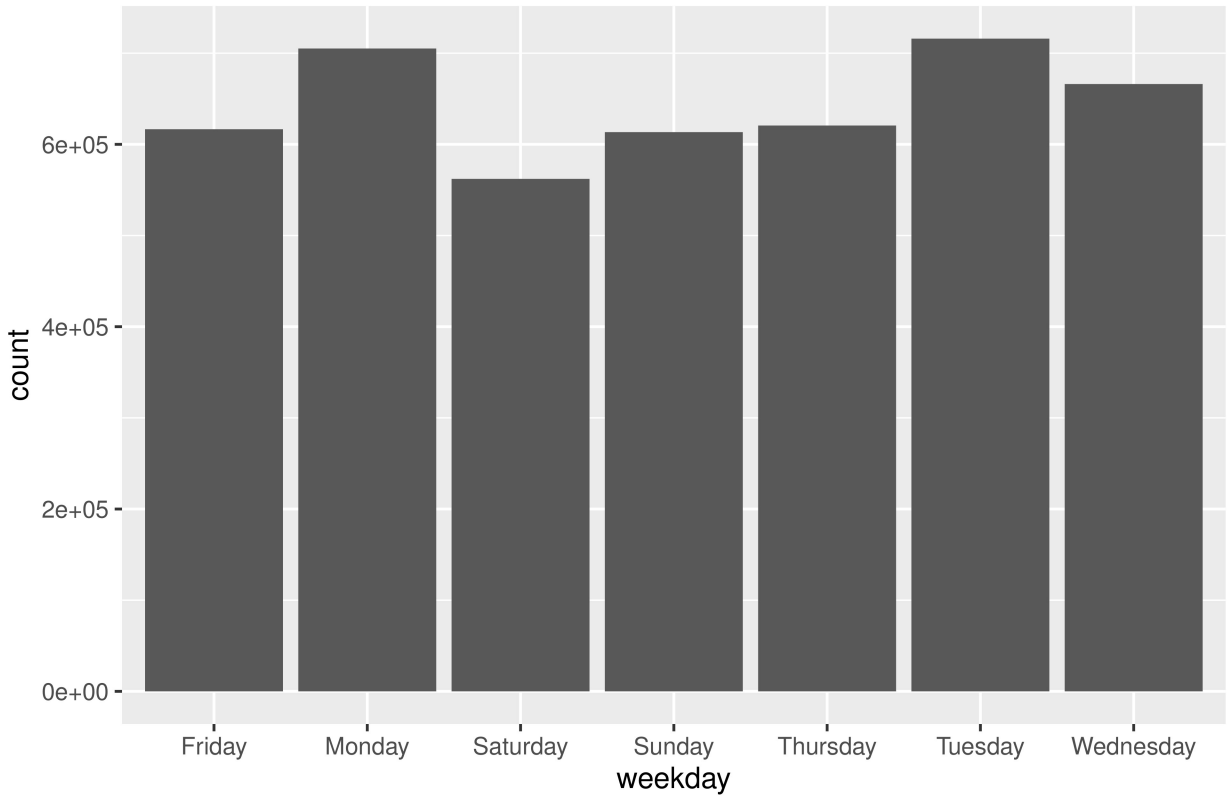
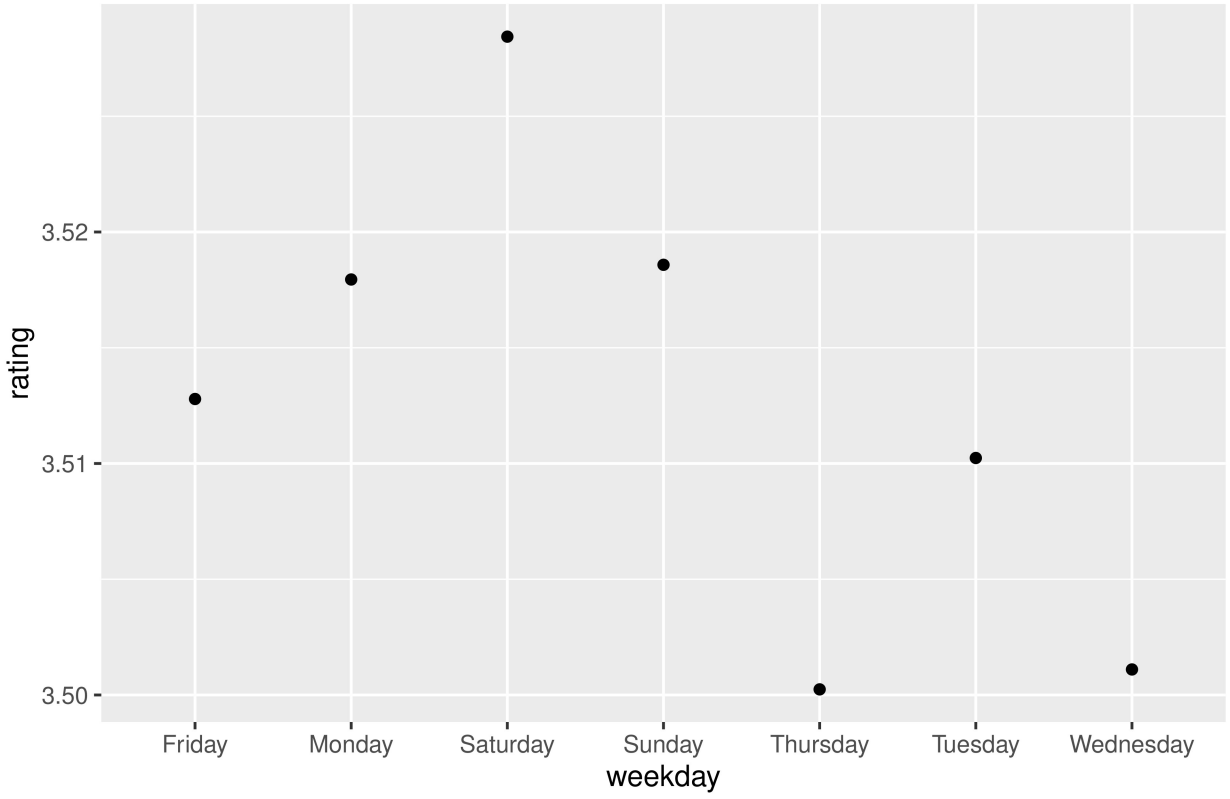


Figure 8 – Average ratings for each week day



1.3 - Exploration of the Genre effect

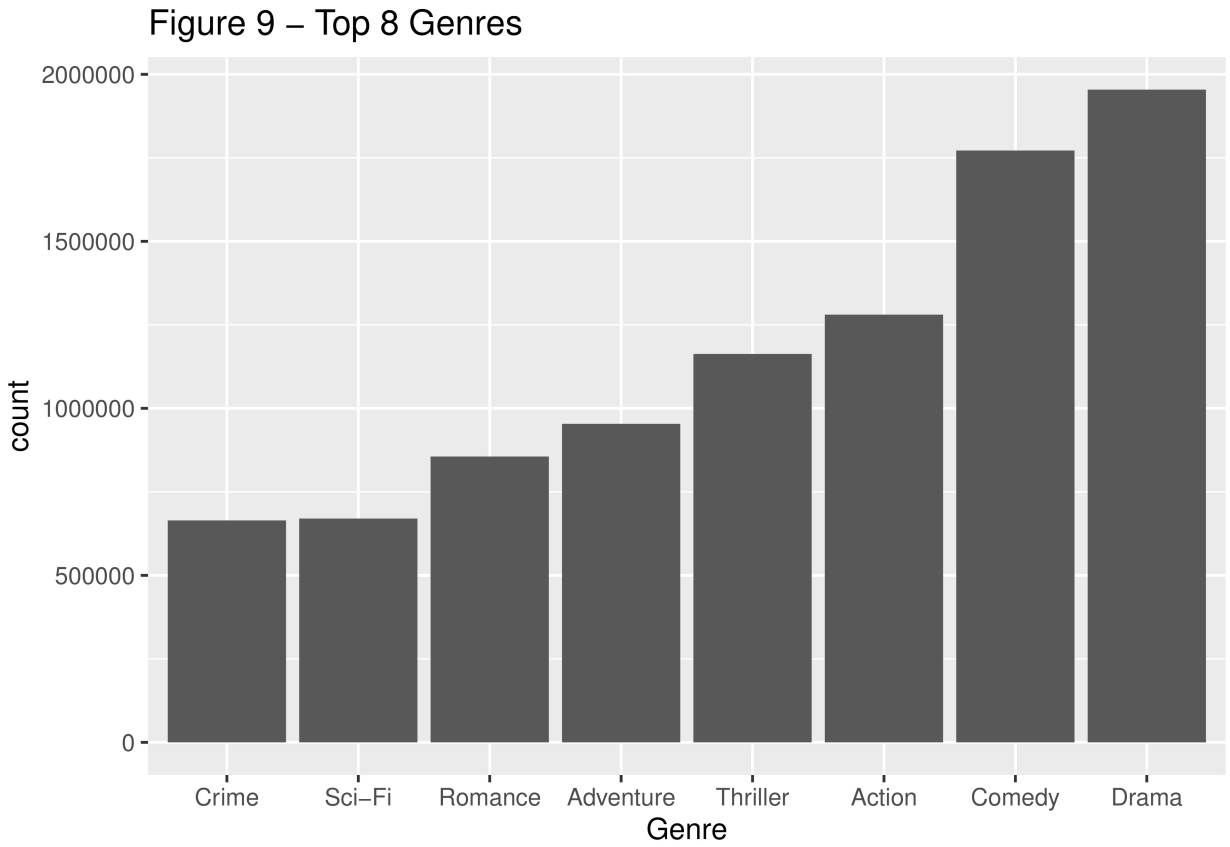
We have so far examined the effect timing may have on the rating, specifically year, hour and weekday. What about genre?

Intuitively we know some genres are more popular than others, so how does the genre impact the rating of a movie?

The genre column in the dataset presents us with a technical challenge as there are multiple genres for some movies. How can we account for the effects of different genres on the rating? Or should each combination of genre (Science Fiction | Drama) be considered as a genre in of itself? One movie, The Host (2008) has 8 associated genres, highlighting the challenge of reorganizing this data:

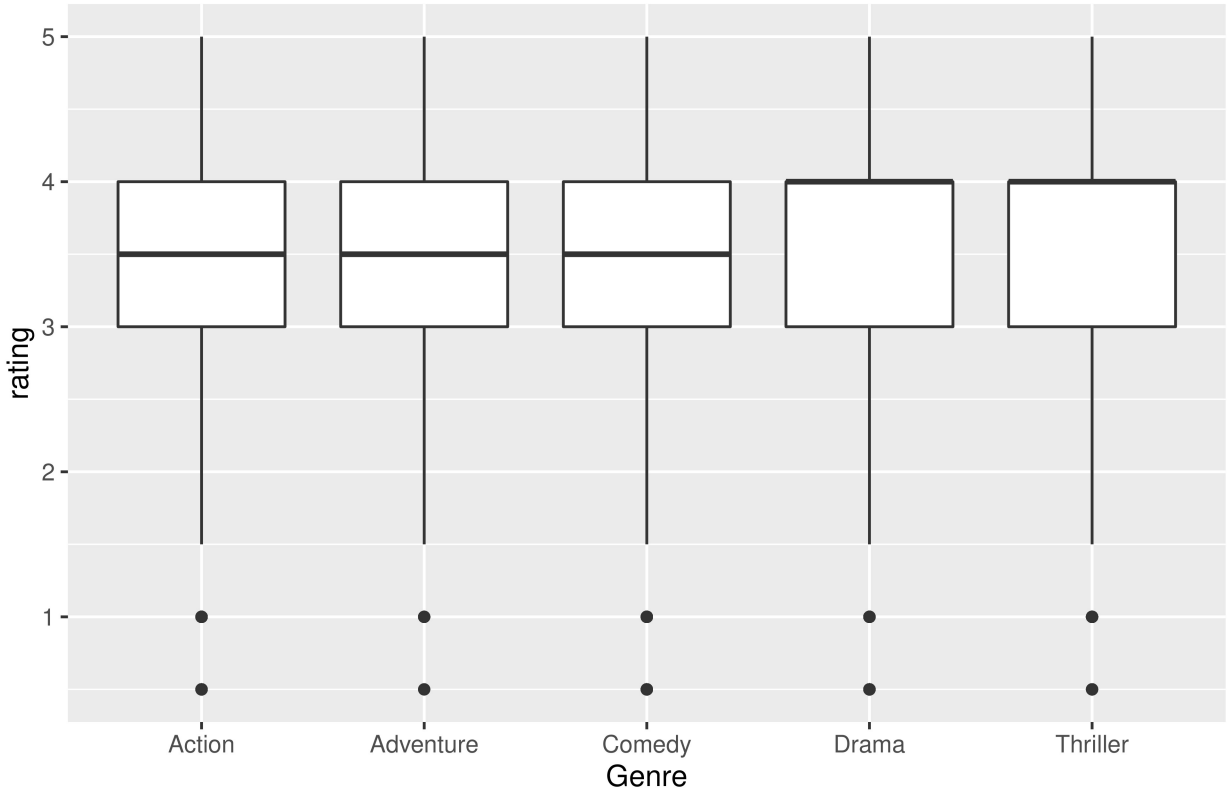
Action|Adventure|Comedy|Drama|Fantasy|Horror|Sci-Fi|Thriller

We can begin by breaking up the genres column to see what are the unique values and how often they occur. The following plot shows the 20 unique values in the genres column and their occurrences throughout the training set.



The most common 8 genres are Dramas, Comedies, Action, Thriller, Adventure, Romance, Science Fiction, and Crime. What patterns can be observed looking at individual genres? For expediency, we look at the top 5 only (Figure 10).

Figure 10 – Median Ratings by Top 5 Genres



The limitation of the above approach is that in the case where movies have multiple genres, they are being 'coerced' into one genre. A "Drama | Comedy" is coded here as a "Drama" and a "Comedy | Thriller" coded as a Comedy, and so on. Yet, a pattern emerges here with Dramas having a median rating of 4 and action adventure, and comedy having medians of 3.5.

An alternative approach is to split out the genres column into eight new columns (eight being the highest number of genres for a movie), with a new column representing the first genre identified for each movie. Here a clear limitation is that each column "Gen1", "Gen2", etc. will have different values within it.

```
##    userId movieId rating timestamp                        title
## 1:      1     185      5 838983525              Net, The (1995)
## 2:      1     292      5 838983421              Outbreak (1995)
## 3:      1     316      5 838983392              Stargate (1994)
## 4:      1     355      5 838984474      Flintstones, The (1994)
## 5:      1     356      5 838983653          Forrest Gump (1994)
## 6:      1     420      5 838983834 Beverly Hills Cop III (1994)
##                          genres     Gen1      gen2     gen3     gen4 gen5 gen6
## 1:        Action|Crime|Thriller   Action     Crime Thriller     <NA> <NA> <NA>
## 2: Action|Drama|Sci-Fi|Thriller   Action     Drama   Sci-Fi Thriller <NA> <NA>
## 3:      Action|Adventure|Sci-Fi   Action Adventure   Sci-Fi     <NA> <NA> <NA>
## 4:      Children|Comedy|Fantasy Children    Comedy  Fantasy     <NA> <NA> <NA>
## 5:     Comedy|Drama|Romance|War   Comedy     Drama  Romance      War <NA> <NA>
## 6: Action|Comedy|Crime|Thriller   Action    Comedy    Crime Thriller <NA> <NA>
##    gen7 gen8 year_of_rating year yeardif weekday weekdayorder hour
## 1: <NA> <NA>           1997 1995       2  Friday            5   11
## 2: <NA> <NA>           1997 1995       2  Friday            5   11
## 3: <NA> <NA>           1997 1994       3  Friday            5   11
```

```
## 4: <NA> <NA>           1997 1994       3  Friday           5   11
## 5: <NA> <NA>           1997 1994       3  Friday           5   11
## 6: <NA> <NA>           1997 1994       3  Friday           5   11
```

In light of the limitations mentioned in wrangling the genre column, our approach will be to use the genre column 'as is' and treat each genre combination as a genre in of itself.

1.3 - Exploration of User and Movie effects

Finally, we can consider adding in the previously examined user and movie effects from the Machine Learning course, section 6.2

We know from that module that there is variability in terms of the number of ratings that users submit and in their ratings.

Some users may be overcritical while others may be the complete opposite and give every movie a 5-star rating.

Similarly, some movies will receive higher ratings and others, lower ratings.

Figure 11 – Average rating per users who had 50+ reviews

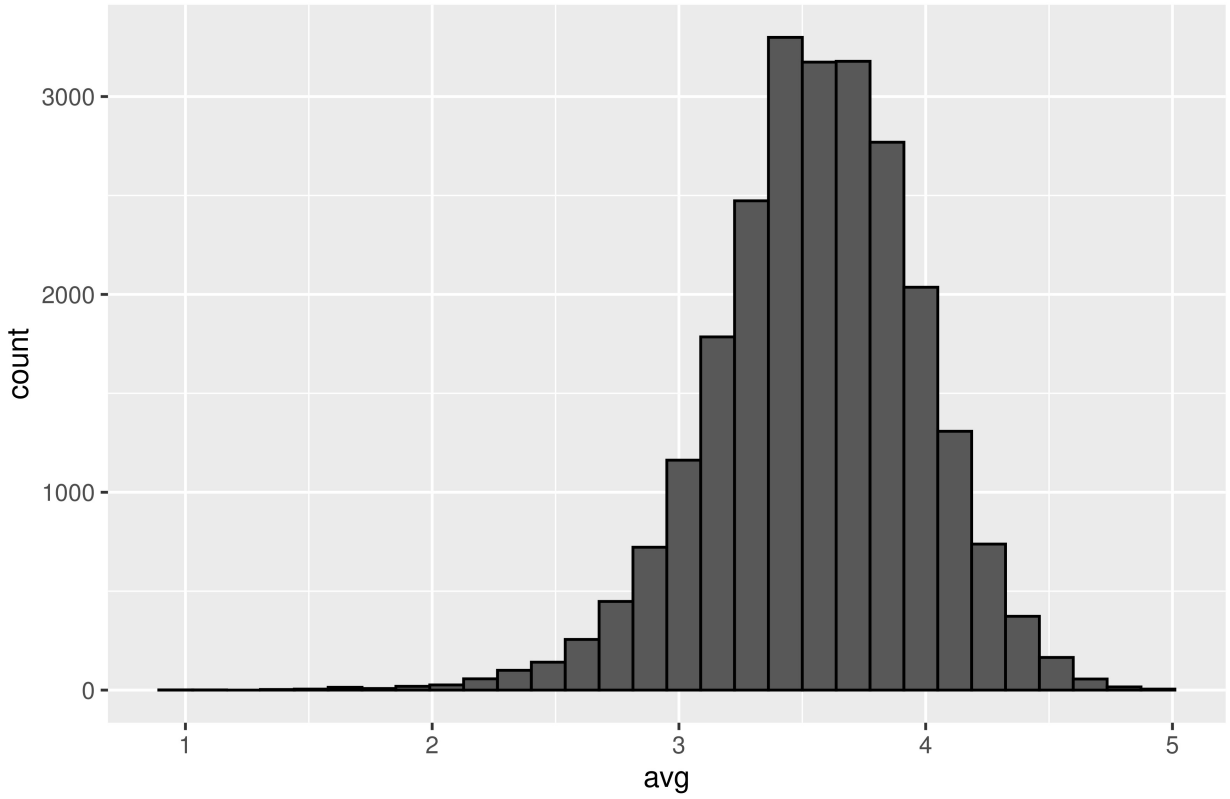
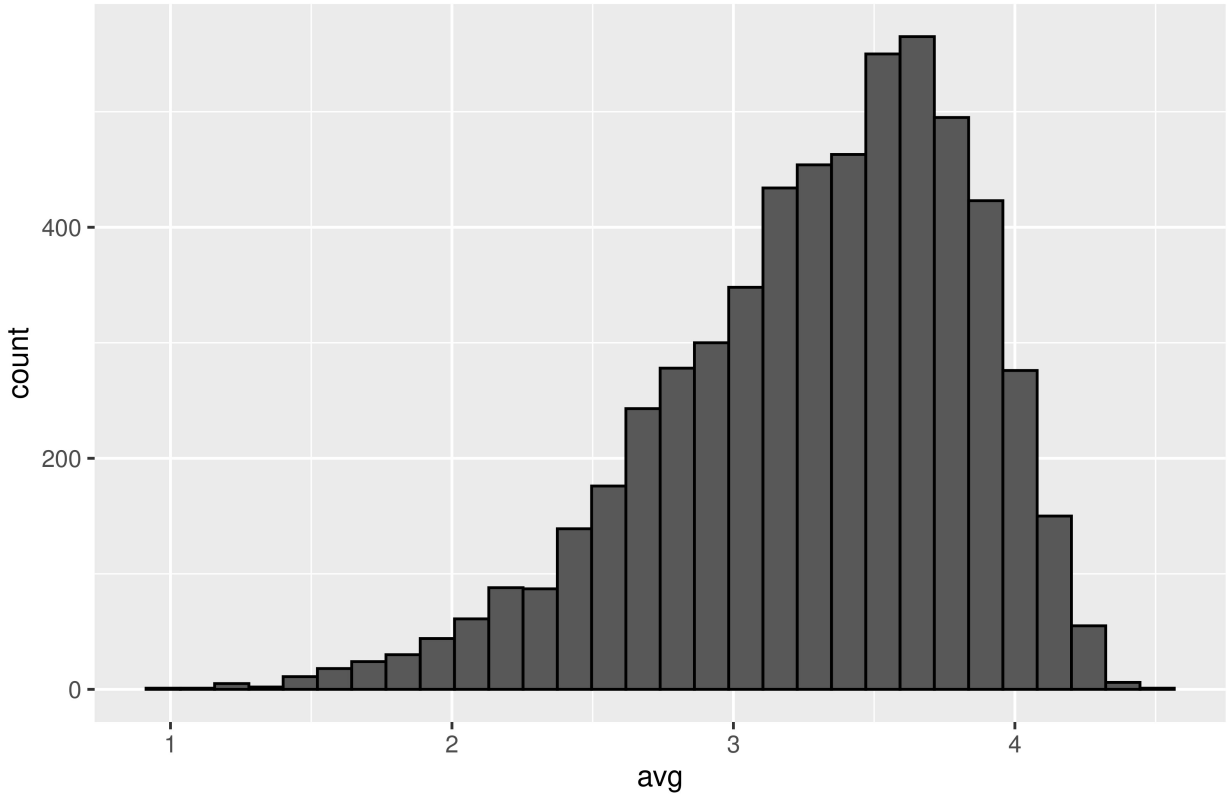


Figure 12 – Average rating per movie for movies with 50+ reviews



Our model must take all of these effects into account.

Section 2. Methods and Analysis – Building the model

So far we have described the Edx dataset in detail and demonstrated how recoding certain variables or creating new ones can reveal patterns about movie ratings. How can this information be added to a machine learning model?

Ultimately, our aim is to predict how a user will rate a movie, between 0.5 stars and 5 stars, based on their past ratings. The model we build will use our training/test sets and eventually be tested on the validation 'final hold-out test-set' to predict the real ratings as if they are unknown.When tested on the validation set, the Residual Mean Square Error (RMSE) should be less than 0.86490.

Since the aim is prediction and the output is a number, it seems appropriate that we classify this as a regression problem (https://towardsdatascience.com/do-you-know-how-to-choose-the-right-machine-learning-algorithm-among-7-different-types-295d0b0c7f60). While more advance modelling techniques could be used, such as those submitted as part of the competition for the 'Netflix Prize' (http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/), when this author attempted to process a random forest involving 1% of the edx set they were unable to due to the technical limitations of their computer. A regression model may not be the most advanced method but it may still allow us to accomplish our goal of obtaining an RMSE of <.86490.

Similar to the algorithm outlined in section 6.2 of the machine learning course, we proceed here to test the effect of one bias and then add in additional biases one at a time, observing how the RMSE changes in each model.

We begin by identifying the mean of ratings in the training set and then add in each bias to improve the predictive power of the model.

mu = the mean of all ratings.

B_wd = the weekday bias. This is the average rating for each weekday. We obtain it as follows:

-Group the data by weekday.

-Mean(Each individual rating – mu).

B_yd = the year difference bias . This is the average rating for each year difference (between the year of rating and the year of release). We obtain it as follows:

-Group by yeardiff.

-Mean(each individual rating – mu – b_wd).

B_h = the hour bias . This is the average rating for each hour. We obtain it as follows:

-Group by hour.

-Mean(each individual rating - movie rating – mu – b_wd – b_yd).

b_g = the genre bias. This is the average rating for each genre combination. We obtain it as follows:

-Group by genres.

-Mean(each individual rating – mu – b_wd – b_yd – b_h).

b_i = the movie bias. This is the average rating for every movie. We obtain it as follows:

-Group by movie_id.

-Mean(each individual rating – mu – b_wd – b_yd – b_h – b_g).

b_u = the user bias. The average rating made by every user. We obtain it as follows:

-Group by userId.

-Mean(each individual rating – mu – b_wd – b_yd – b_h – b_g - b_i).

Model 1: mu + b_wd.

RMSE is 1.060155.

Model 2: Model 1 + b_yd. RMSE is 1.051497.

Model 3: Model 2 + b_h.

RMSE is 1.051458.

Model 4: Model 3 + b_g.

RMSE is 1.010949.

Model5: Model 4 + b_i.

RMSE is 0.9443406.

Model 6: mu + b_wd + b_yd + b_h + b_g + b_i + bu.

RMSE is 0.8700048.

Section 3 – Results

Our final model used (Model 6) followed an identical methodology to the model developed in section 6.2 of the machine learning course. The model in that section used the user effect and the movie effect to produce an RMSE of 0.905 (without regularization)

When Model 6 was tested on the edx set, the RMSE was 0.8567675. When Model 6 was tested on the validation 'final hold-out' set, the RMSE was 0.8252246. We have successfully met our goal of obtaining an RMSE of $< .86490$.

Section 4. Discussion

Methodologically, we have followed a straightforward approach while still offering a model that has predictive power. The model did not come without its limitations. For example, it would be interesting to gain access a new version of the dataset (or a different one altogether) that includes columns such as gender and age. There are countless other biases that could have been added to the model. Another limitation is our use of the genre column. More advanced wrangling techniques might be able to properly isolate the individual genre. Finally, advanced equipment could permit experimenting with other modelling techniques (random forests).

Conclusion

This paper has presented a linear regression model for predicting how a given user will rate a given movie by making use of a sample of the Movielens dataset. During an initial data exploration, we computed new columns and discovered new patterns related to:

-The year in which the movie was made

-The year the rating was made, and

-The difference between them

-The week day on which the rating was made

-The hour of the day in which the rating was made

We saw how such time-related factors influence the average ratings for a movie.

We also attempted to examine the effects of genres on average ratings.

Several biases were created and added one at a time to the training model which were applied to the test set. A final model involving six biases was tested on the edx set and on the validation set to produce an RMSE of 0.8567675 and 0.8252246 respectively.