Movie Recommendation Project

Jake Hogan

11/13/2020

Executive Summary

Recommendation systems are a common application of machine learning throughout the tech industry from recommending products on Amazon to recommending movies on Netflix. The goal of this project was to create a movie recommendation system based on the MovieLens dataset that minimizes the residual mean squared error (RMSE). The dataset used for this analysis is a smaller subset of the one generated by the GroupLens research lab and contains 10 million movie ratings of 10,000 movies by 72,000 users. Data analysis and visualization was used to determine the effects that the movie, user, genre, and age of the movie have on the rating. Then a linear regression model was created to account for each of these effects and predict movie ratings. The recommendation model was tested once on a validation dataset with a final RMSE of 0.8623.

Data Analysis

The 10 million rating MovieLens dataset can be found on the GroupLens Website (grouplens.org). The data comes in two files, a ratings file and a movies file. The entries in each file are delimited by "::" which is used to define a new column in the file. The ratings dataset is assigned the column names userId, movieId, rating, and timestamp. The movies dataset is assigned the column names movieId, title, and genres. A single movielens data frame is created by joining the ratings and movies datasets by the movieId. From there the MovieLens data is partitioned into two data frames. The first is the validation set which is 10% of the MovieLens data and the rest goes into the edx set. The validation set will only be used once to validate the model that is created using only the edx set.

Looking into the edx data set shows that it has 9000055 rows and 6 columns. This breaks down to 10677 different movies and 69878 different users. Below is a list of the 10 most rated movies.

Table 1: 10 Most Rated Movies

title	n
Pulp Fiction (1994)	31362
Forrest Gump (1994)	31079
Silence of the Lambs, The (1991)	30382
Jurassic Park (1993)	29360
Shawshank Redemption, The (1994)	28015
Braveheart (1995)	26212
Fugitive, The (1993)	25998
Terminator 2: Judgment Day (1991)	25984
Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)	25672
Apollo 13 (1995)	24284

The distributions below show how often movies are rated and how many ratings a user provides, respectively. Most movies have less than 1,000 reviews and most users review less than 100 movies.

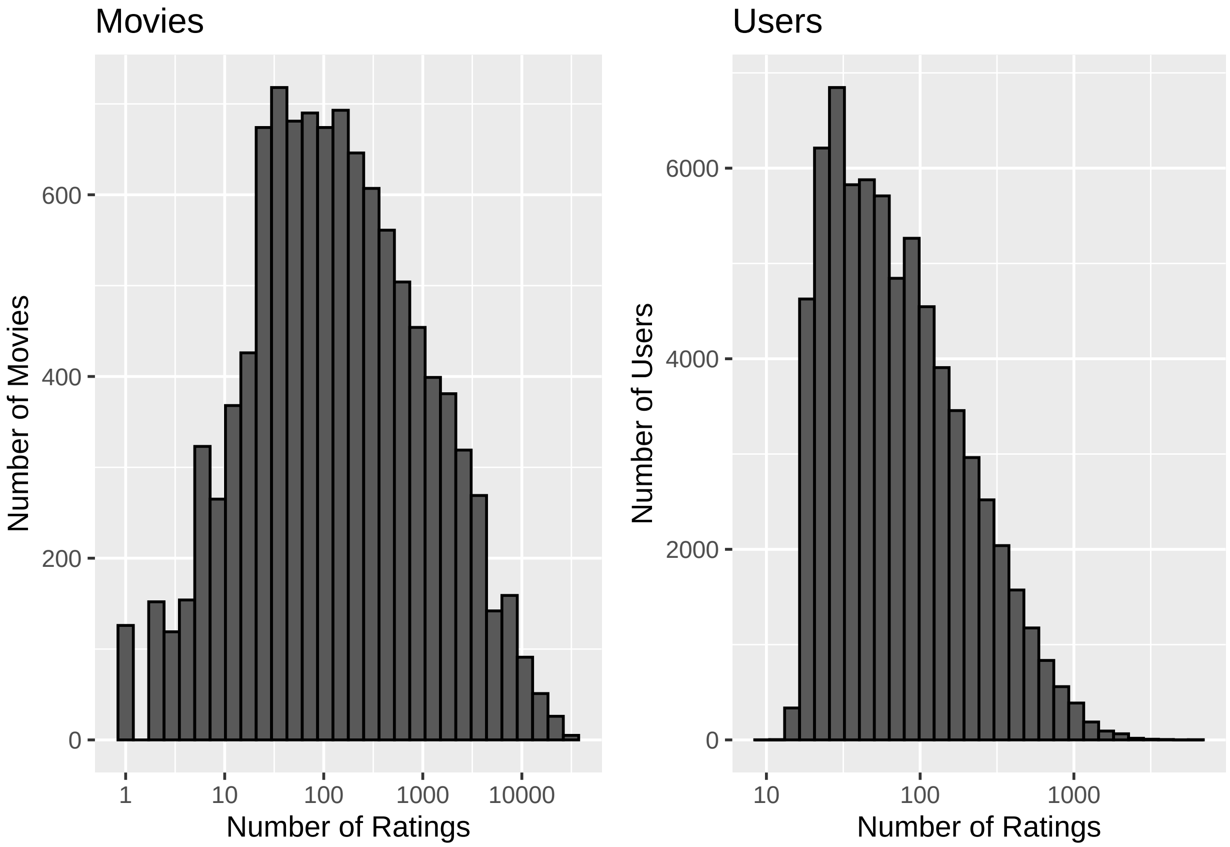

Figure 1: Distribution of Movie and User Ratings

Looking at the distribution of average moving rating by user shows that some users don't like many movies and others really tend to enjoy all movies.

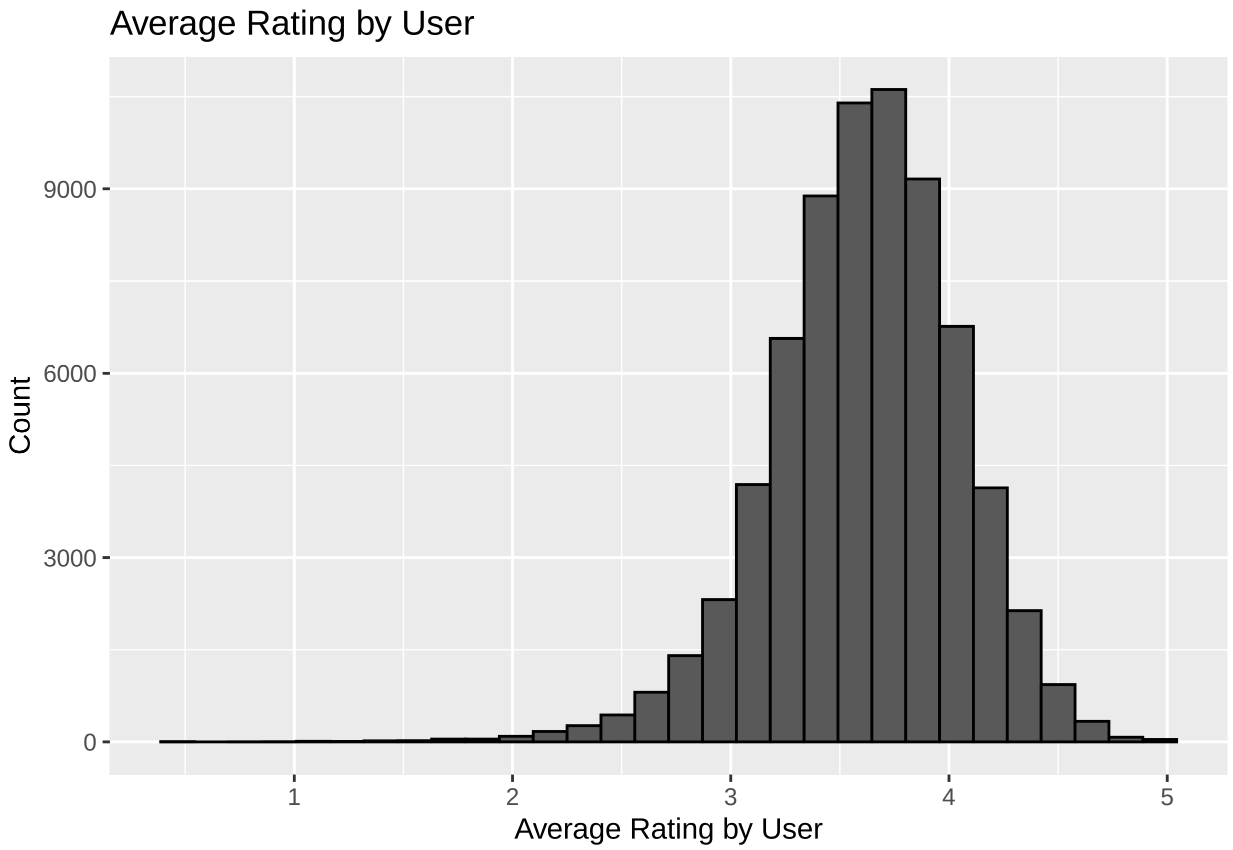

Figure 2: Average Movie Rating by User

Here is a look at the distribution of movie ratings. A rating of 4 is the most common while half point ratings (e.g. 3.5) are much less common than whole point ratings.

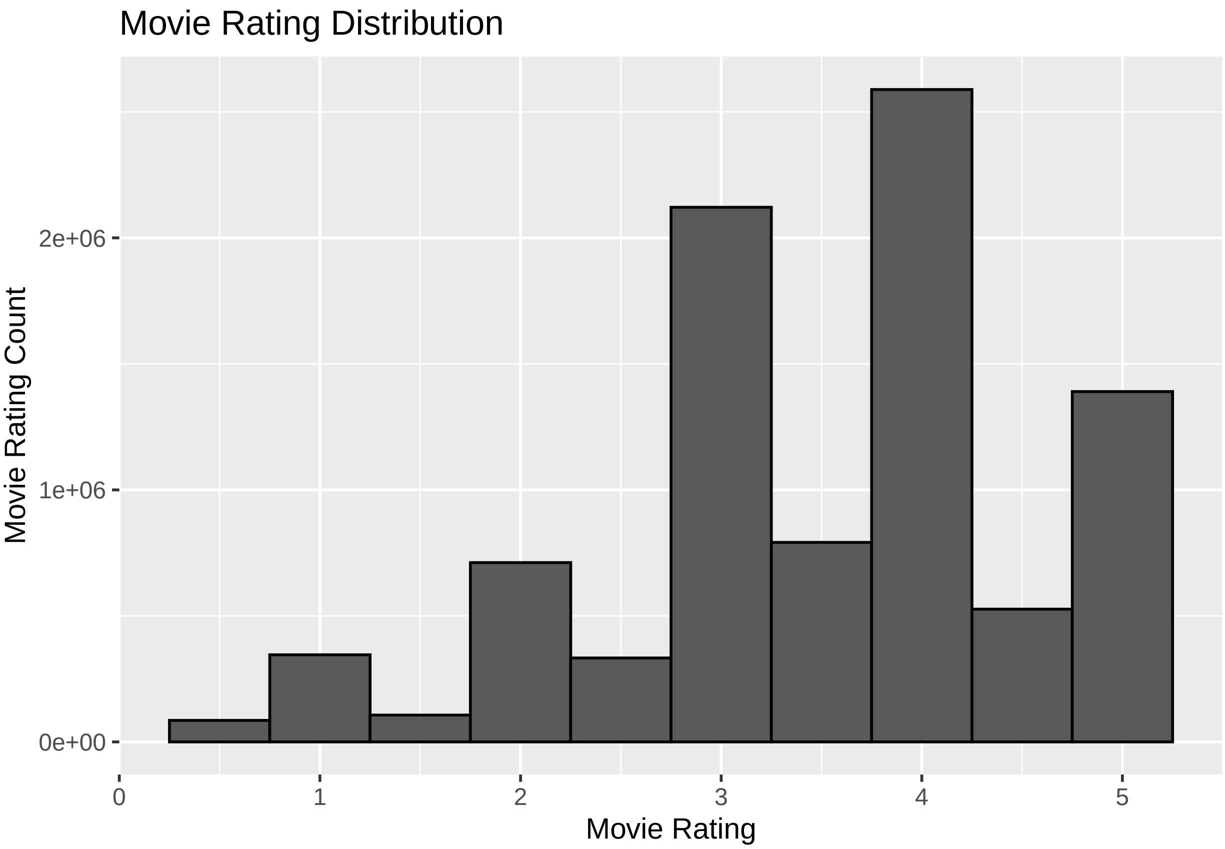

Figure 3: Distribution of Movie Ratings

Table 2: Movie Ratings

rating	count
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

Looking at the number of movie ratings by title shows that the movies with the most ratings are typically blockbusters or action movies.

Table 3: 10 Most Rated Movies

title	n
Pulp Fiction (1994)	31362
Forrest Gump (1994)	31079
Silence of the Lambs, The (1991)	30382
Jurassic Park (1993)	29360
Shawshank Redemption, The (1994)	28015
Braveheart (1995)	26212
Fugitive, The (1993)	25998
Terminator 2: Judgment Day (1991)	25984
Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)	25672
Apollo 13 (1995)	24284

Movies with the highest average moving rating are typically small, independent movies with fewer total ratings.

Table 4: 10 Highest Rated Movies

title	n	avg_rating
Blue Light, The (Das Blaue Licht) (1932)	1	5.00
Fighting Elegy (Kenka erejii) (1966)	1	5.00
Hellhounds on My Trail (1999)	1	5.00
Satan's Tango (Sátántangó) (1994)	2	5.00
Shadows of Forgotten Ancestors (1964)	1	5.00
Sun Alley (Sonnenallee) (1999)	1	5.00
Constantine's Sword (2007)	2	4.75
Human Condition II, The (Ningen no joken II) (1959)	4	4.75
Human Condition III, The (Ningen no joken III) (1961)	4	4.75
Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1980)	4	4.75

Our own intuition tells us that the age of a movie as well as the genre could have an impact on its rating. To look at specific genres we first need to separate out each genre listed for each movie.

```
edx_genres <- edx %>% separate_rows(genres, sep = "\\|")
```

Then we can look at the average rating by genre. Clearly the Film-Noir genre has a higher avg rating than all other genres.

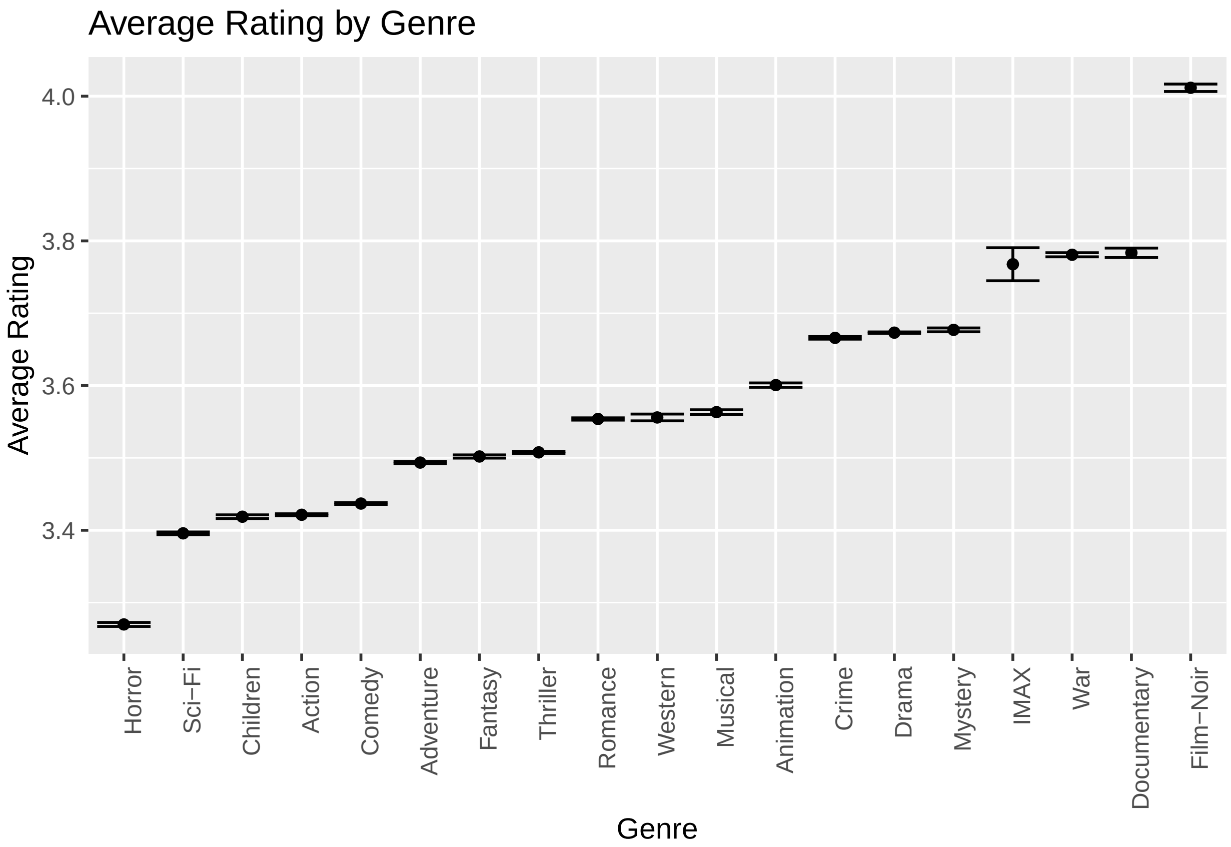

Figure 4: Average Movie Rating by Genre

Based on intuition, Film-Noir doesn't seem to be a popular genre so let's take a look at how many times each genre is rated. There are over 30 times more Drama ratings than Film-Noir ratings.

Table 5: Number of Ratings by Genre

genres	n
Drama	3910127
Comedy	3540930
Action	2560545
Thriller	2325899
Adventure	1908892
Romance	1712100
Sci-Fi	1341183
Crime	1327715
Fantasy	925637
Children	737994
Horror	691485
Mystery	568332
War	511147
Animation	467168
Musical	433080
Western	189394
Film-Noir	118541
Documentary	93066
IMAX	8181

The age of a movie can have an impact on the rating. Contemporary society views change and older movies can reflect views of different generations. Let's see if release year has an effect on ratings. To do this we need to pull the year the movie was released from the movie title.

```
library(stringr)
edx <- edx %>% mutate(release_year = as.numeric(str_extract_all(title,
               "(?<=\\()\\d{4}(?=\\))")))
```

Then we can look at the average movie ratings by movie release year. This shows that how old a movie is having an effect on the rating.

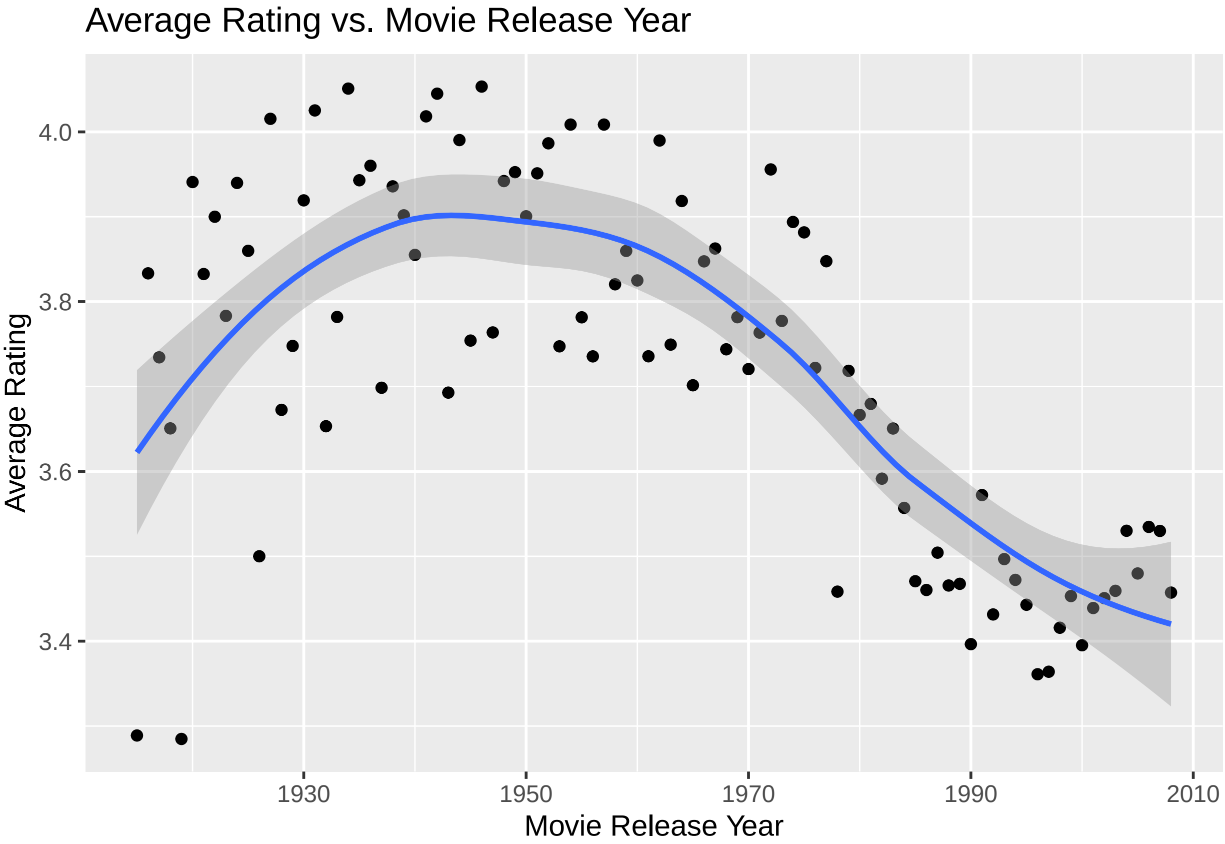

Figure 5: Average Movie Rating by Release Year

Taking four of the more common genres (Action, Comedy, Drama, Romance) and plotting average rating by release year shows genre popularity over time as well.

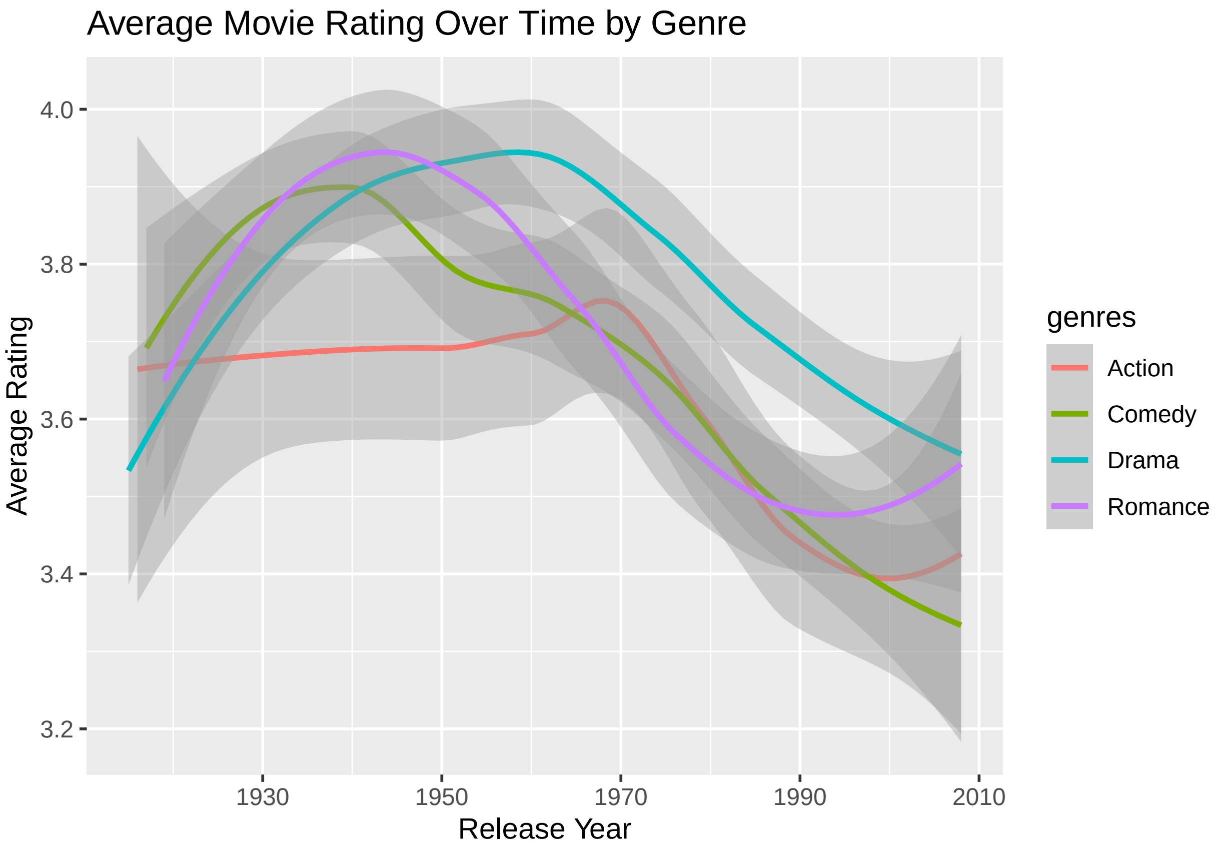

Figure 6: Average Movie Rating by Genre and Release Year

Model Development Methods

The performance of the recommendation system will be judged on the residual mean squared error (RMSE) on a validation set. RMSE is defined as[1]:

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

Where $y_{u,i}$ is the rating for movie i by user u, the prediction is $\hat{y}_{u,i}$, N is the number of user/movie combinations with the sum occurring over all combinations.

```
#define the RMSE function for testing losses
RMSE <- function(actual_ratings, pred_ratings){
  sqrt(mean((actual_ratings - pred_ratings)^2))
}
```

Since the validation set will only be used once to test our final model we'll need to partition the edx dataset into train and test sets. A standard partition of 80% for training and 20% for testing was selected since the edx set is still large after the validation split.

Average Rating Model

A simple, initial model to make movie recommendations is to use the average movie rating as the basis for the recommendation. In other words, assume the same rating for all movies. The linear model would look like this[1]:

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

where $\varepsilon_{i,u}$ is independent errors sampled from the same distribution and μ is the average rating for all movies. The average rating is 3.512 with an RMSE of 1.06.

Table 6: RMSE Results

Method	RMSE
Average Only Model	1.060704

The standard deviation of the movie ratings is 1.0602373 which nearly matches the RMSE so this is not a good method for making predictions but it can be built on by accounting for effects that were apparent from data exploration.

Movie Effects

Data exploration showed that different movies tend to get different ratings (e.g. blockbusters vs. independent films). The movie effect, or the average rating for each movie, is represented as b_i and can be added to the linear model[1]:

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

The least squares estimate can be used to calculate this effect. However, in this case, the least squares estimate of b_i is also the average of the difference between the rating of each movie and the overall average rating:

```r
movie_avgs <- edx_train_set %>%
  group_by(movieId) %>%
  summarize(b_i = mean(rating - mean_rating)) #calculate the LSE b_i
```

The distribution shows that b_i estimates have quite a bit of variation.

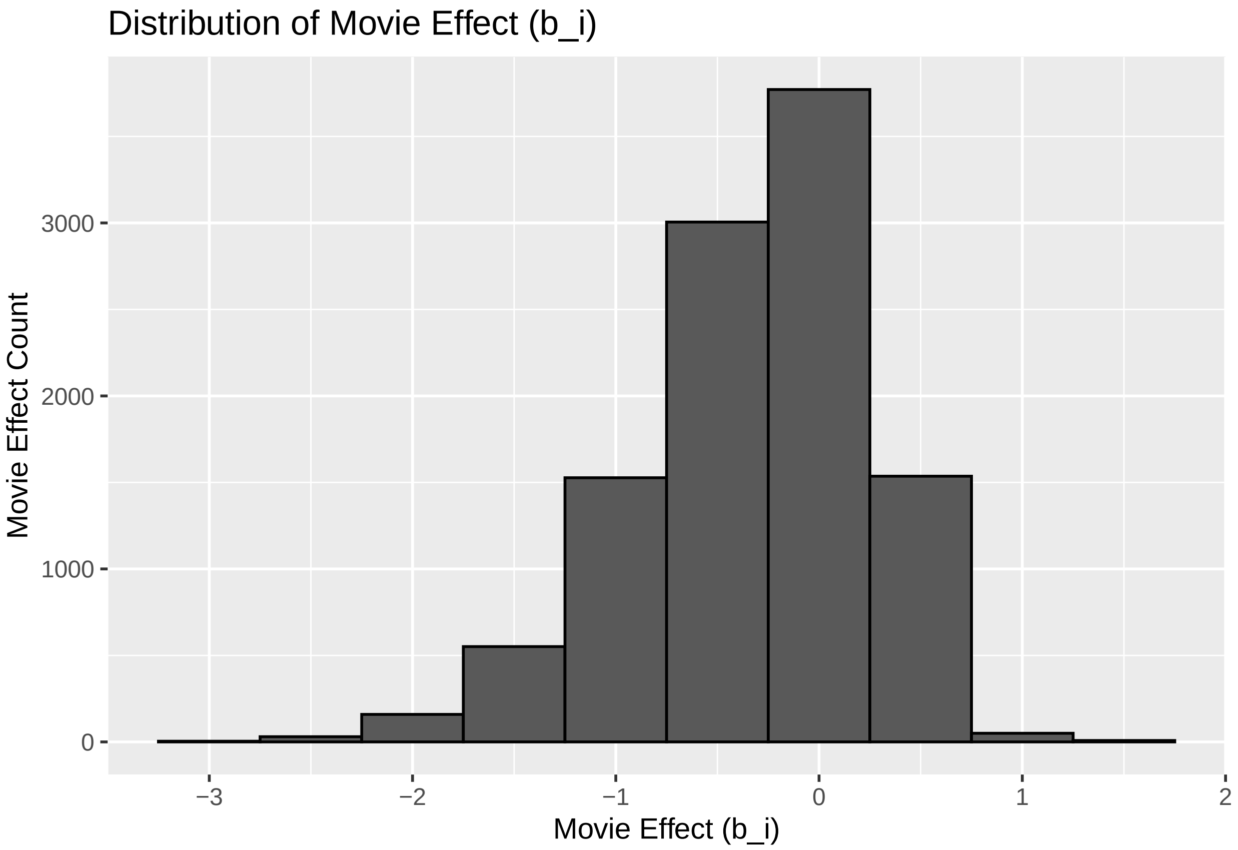

Figure 7: Distrbution of Movie Effect Estimates

Plugging in the b_i estimates into the prediction yields a better RMSE:

Table 7: RMSE Results

Method	RMSE
Average Only Model	1.0607045
Movie Effect Model	0.9437144

With just the movie effect the model has improved quite a bit. This can still be improved by adding more effects.

User Effects

Data exploration showed that different users tend to rate movies differently (e.g. some love every movie and some don't like any movies). The user effect, or the average rating for each user, is represented as b_u and can be added to the linear model[1]:

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

Similar to the movie effect (b_i) the least squares estimate of b_u is also the average of the difference between the rating of each movie, the movie effect, and the overall average rating:

```r
#b_u calculated similarly to b_i
user_avgs <- edx_train_set %>%
  left_join(movie_avgs, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating - mean_rating - b_i))
```

Here is the new RMSE with the user effect added:

Table 8: RMSE Results

Method	RMSE
Average Only Model	1.0607045
Movie Effect Model	0.9437144
Movie + User Effect Model	0.8661625

Release Year Effect

Data exploration showed that the release year appeared to have an effect on movie ratings (e.g. more recent movies had a lower rating on average than movies from the 20th century). The release year effect, or the average rating for each release year, is represented as b_y and can be added to the linear model:

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

The least squares estimate of b_y can be determined in the same manner that the movie and user effects were calculated:

```r
#b_y calculated similarly to b_i
rel_year_avgs <- edx_train_set %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by = 'userId') %>%
  group_by(release_year) %>%
  summarize(b_y = mean(rating - mean_rating - b_i - b_u))
```

Here is the newest, slightly improved, RMSE:

Table 9: RMSE Results

Method	RMSE
Average Only Model	1.0607045
Movie Effect Model	0.9437144
Movie + User Effect Model	0.8661625
Movie + User + Year Effect Model	0.8658322

Genre Effect

Finally, movie genre also appeared to have an effect on movie ratings. The genre effect, or the average rating for each genre, is represented as b_g and can be added to the linear model:

$$Y_{u,i} = \mu + b_i + b_u + b_y + b_g + \varepsilon_{u,i}$$

Recall that each movie can list multiple genres so each listed genre is separated out for each movie. By doing this genre effect becomes the sum of the effects of each genre listed for each movie. The least squares estimate of b_g is calculated similarly to all the previously discussed effects:

```r
#b_g calculated similarly to b_i
genre_avgs <- edx_train_genres %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by = 'userId') %>%
  left_join(rel_year_avgs, by = "release_year") %>%
  group_by(genres) %>%
  summarize(b_g = mean(rating - mean_rating - b_i - b_u - b_y))
```

Using b_g to make new predictions the newest RMSE is:

Table 10: RMSE Results

Method	RMSE
Average Only Model	1.0607045
Movie Effect Model	0.9437144
Movie + User Effect Model	0.8661625
Movie + User + Year Effect Model	0.8658322
Movie + User + Year + Genre Effect Model	0.8640294

It seems that the release year effect and the genre effect didn't have as great an impact on the RMSE as the movie and user effects.

Regularization

The recommendation model has improved dramatically over the initial method of assuming the average rating for all movies. However, there may still be room for improvement. Let's take a look at how each effect was wrong. One way to look at this is the residuals for each effect. The residual is the difference between the actual rating and the estimate. Calculating the movie effect (b_i) residuals and looking at the largest 10:

Table 11: Highest Movie Effect Residuals

title	resid
From Justin to Kelly (2003)	4.139752
Shawshank Redemption, The (1994)	-3.957980
Pokémon Heroes (2003)	3.914414
Godfather, The (1972)	-3.914337
Schindler's List (1993)	-3.868707
Usual Suspects, The (1995)	-3.866020
Pokemon 4 Ever (a.k.a. Pokémon 4: The Movie) (2002)	3.828402
Casablanca (1942)	-3.824928
Double Indemnity (1944)	-3.822300
Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)	-3.821537

Some of these movies are pretty unfamiliar and are way off. Looking closer at b_i here are the top 10 best and worst movies based on the movie effect:

Table 12: 10 Highest Movie Effects

Title	Number of Ratings
Hellhounds on My Trail (1999)	1
Shanghai Express (1932)	1
Satan’s Tango (Sátántangó) (1994)	2
Fighting Elegy (Kenka erejii) (1966)	1
Sun Alley (Sonnenallee) (1999)	1
Bullfighter and the Lady (1951)	1
Blue Light, The (Das Blaue Licht) (1932)	1
Human Condition II, The (Ningen no joken II) (1959)	3
Who’s Singin’ Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1980)	4
Life of Oharu, The (Saikaku ichidai onna) (1952)	2

Table 13: 10 Lowest Movie Effects

Title	Number of Ratings
Besotted (2001)	2
Grief (1993)	1
Confessions of a Superhero (2007)	1
War of the Worlds 2: The Next Wave (2008)	1
Disaster Movie (2008)	30
SuperBabies: Baby Geniuses 2 (2004)	40
From Justin to Kelly (2003)	161
Hip Hop Witch, Da (2000)	10
Criminals (1996)	2
Mountain Eagle, The (1926)	1

In both tables these movies are again pretty unknown. When you look at the number of times each movie was rated you can see it’s very few. Fewer ratings can lead to greater uncertainty and increase the RMSE.

Repeating the process but looking at user effect (b_u) shows a similar relationship between the number of ratings and the user effect.

Table 14: 10 Highest User Effects

UserId	Number of Ratings
18591	14
13524	15
46484	22
45895	15
54009	19
36022	80
36896	124
52749	74
7999	33
22650	28

Table 15: 10 Lowest User Effects

UserId	Number of Ratings
13496	17
48146	17
49862	14
63381	15
62815	18
6322	14
42019	25
8920	12
15515	24
44684	32

We already know from intuition that number of movies released per year has increased significantly over time. Also, the analysis of each genre showed Film-Noir had the highest average rating but was also a relatively uncommon genre. Based on these findings, regularization, which penalizes large estimates formed by small sample sizes, could provide some additional benefit in the model. Each effect is then estimated by minimizing the least squares estimate with an additional penalty term (λ) added. For example, the movie effect (b_i) estimate becomes[1]:

$$\hat{b}_i(\lambda) = \frac{1}{\lambda + n_i} \sum_{u=1}^{n_i} (Y_{u,i} - \hat{\mu})$$

where n_i is the number of ratings for movie i. The effect of this equation is to reduce the estimate of b_i when the number of ratings is small but, when n_i is large, λ is essentially ignored and the equation becomes an average. Lambda can be optimized for the lowest RMSE. The chart below shows RMSE over a range of λ values.

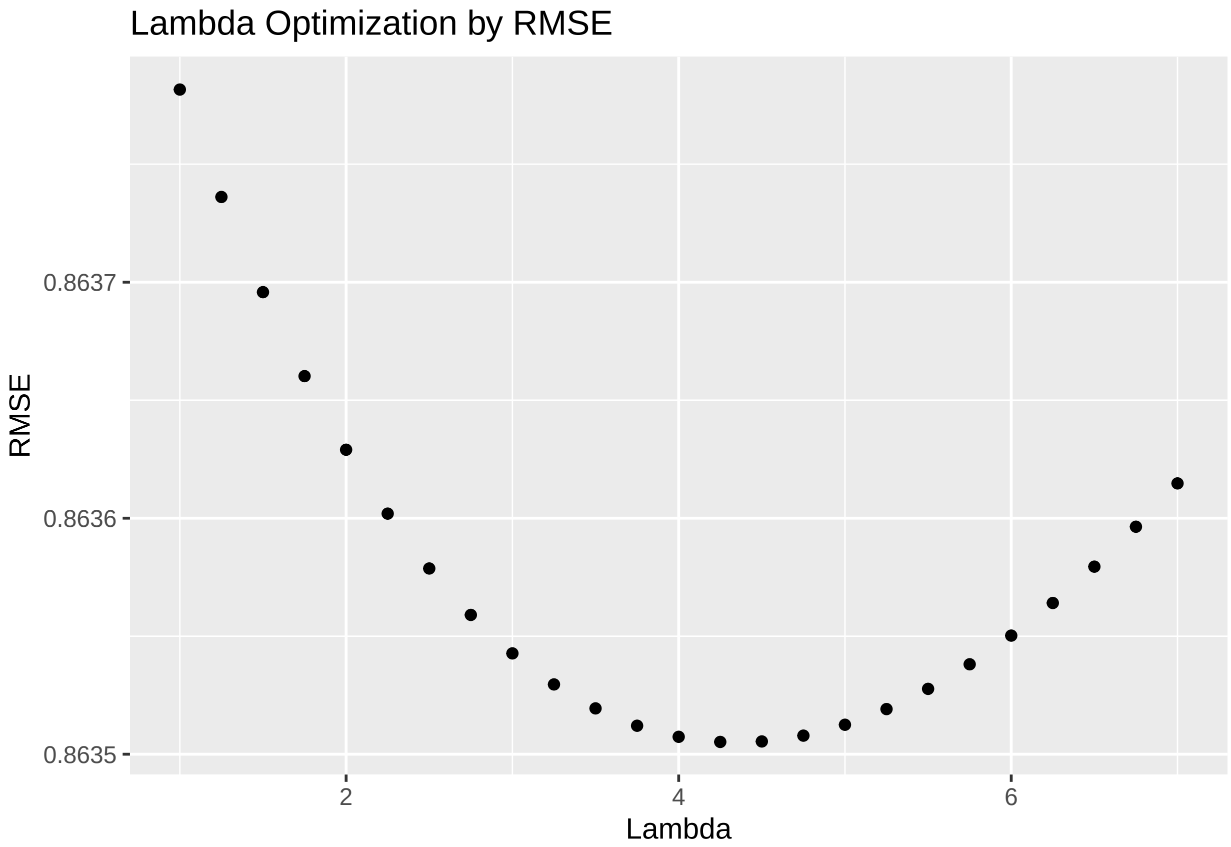

Figure 8: Lambda Effect on RMSE

From the chart we can see that the optimized lambda is 4.25. Comparing the regularized RMSE to the previous models shows that we have improved the model even further.

Table 16: RMSE Results

Method	RMSE
Average Only Model	1.0607045
Movie Effect Model	0.9437144
Movie + User Effect Model	0.8661625
Movie + User + Year Effect Model	0.8658322
Movie + User + Year + Genre Effect Model	0.8640294
Regularized Movie + User + Year + Genre Effect Model	0.8635052

Results

As a final step, recalculate each regularized effect on the entire edx set using the optimized lambda from the training set. Then use the model to make predictions on the validation set created initially and so far unused for training or testing. In order to use genre and release year, the validation set is separated by genre and a release year column is added in the same manner that the edx data set was modified. Finally the RMSE of the predictions on the validation set is calculated.

Table 17: Validation RMSE

Method	RMSE
Regularized Movie + User + Year + Genre Effect Model	0.8623469

Conclusion

Data analysis and visualization showed the effects that the movie, user, genre, and age of the movie have on the rating. A linear regression model was created to account for each of these effects. The residual mean squared error (RMSE) was used to validate the model with a final RMSE of 0.8623. One modeling method that could further improve the prediction model could be exploring any effect that the timestamp of the movie rating had on the rating. Another would be to look into to breaking down release year effect by genre. Singular value decomposition and principal component analysis could potentially derive underlying rating effects that aren't initially obvious from the data visualization performed here.

References

1) Irizarry, Rafael A. *Introduction to Data Science: Data Analysis and Prediction Algorithms with R*. 2020.