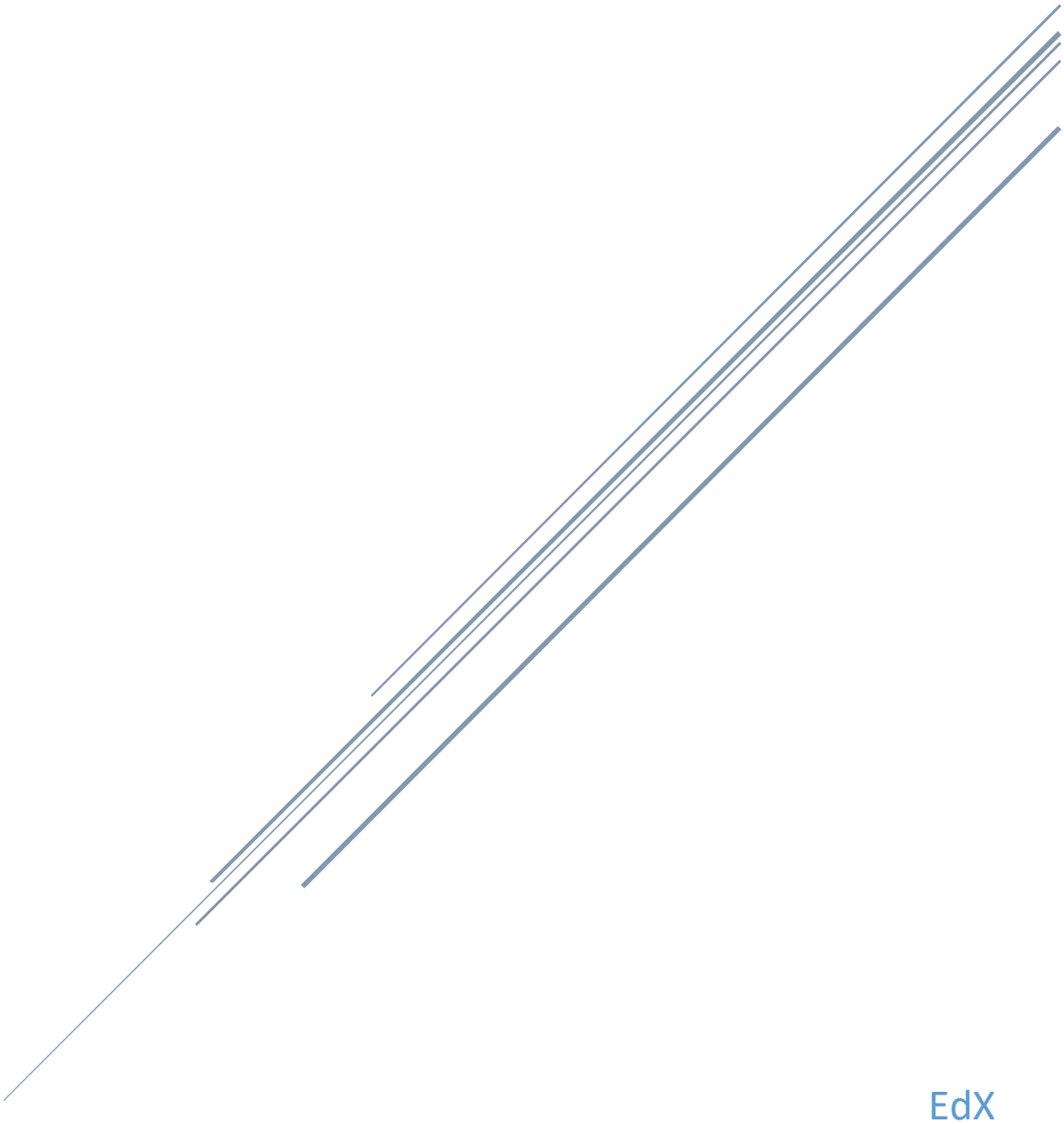
MovieLens Project Submission

Sideek Headlie



EdX

Data Science: Capstone

Contents

Introduction

This report aims to analyze the 'MovieLens 10M Dataset' through the application of data science to derive a greater understanding of the information within the data, as well as generate actionable insights. This dataset is a stable benchmark dataset provided by the GroupLens research lab in the Department of Computer Science and Engineering at the University of Minnesota, Twin Cities.

In the MovieLens 10M Dataset, users were selected randomly for inclusion in the 10M, with ratings taken from the full set of MovieLens data. All users selected had rated at least 20 movies. Users are only represented by an ID, with no other associated demographic information.

In effort to explore the dataset, firstly the data was downloaded and ran in accordance with the instructions provided in this assignment. In addition, several packages were loaded prior to data exploration, an iterative step as the data was integrated using different approaches. In addition, the dataset was explored from multiple and combined perspectives in effort to achieve the aim of the project.

The aim of this project was to predict the ratings for future movies. The dataset was divided into a 90/10 split of training and test datasets. A model was generated from the training dataset, and then applied to the test dataset. The success metric used was the Root Mean Square Estimate (RMSE). The goal was to predict movie ratings on a scale of 0.5 to 5, whereby RMSE was measured on the same scale.

N.B. RMSE of 0 means that we are always correct, which is unlikely. An RMSE of 1 means that predicted ratings are off by 1.

The target RMSE was identified to be a value less than 0.86490 (i.e. RMSE < 0.86490)

The RMSE generated from the predictive model developed for this project was 0.8644266; which is below the target RMSE of 0.86490.

The report reviews the steps undertaken to arrive at this RMSE provided as well as provide an overview of the rationale.

Key data science activities included;

- Data Cleansing
- Data Extraction
- Data Visualization
- Data Prediction

Methods and Analysis

Loading the data followed the instructions provided in the Data Science Capstone Project – MovieLens Project Submission.

In an effort to explore the dataset, firstly the data was downloaded and ran in accordance with the instructions provided in this assignment. Several packages were loaded prior to data exploration, an iterative step as the data was integrated using different approaches. In addition, the dataset was explored from multiple and combined perspectives in effort to achieve the aim of the project.

The following code was provided for data loading and splitting into training and validation sets, and additional packages were installed and loaded for further analysis.

```
############################################################
# Create edx set, validation set (final hold-out test set)
############################################################

# Note: this process could take a couple of minutes

if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(lubridate)) install.packages("lubridate", repos = "http://cran.us.r-project.org")
if(!require(Metrics)) install.packages("Metrics", repos = "http://cran.us.r-project.org")
if(!require(corrplot)) install.packages("corrplot", repos = "http://cran.us.r-project.org")
if(!require(RColorBrewer)) install.packages("RColorBrewer", repos = "http://cran.us.r-project.org")


library(tidyverse)
library(caret)
library(data.table)
library(lubridate)
library(Metrics)
library(corrplot)
library(RColorBrewer)




# MovieLens 10M dataset:
# https://grouplens.org/datasets/movielens/10m/
# http://files.grouplens.org/datasets/movielens/ml-10m.zip
```

```
dl <- tempfile()
download.file("http://files.grouplens.org/datasets/movielens/ml-10m.zip", dl)

ratings <- fread(text = gsub("::", "\t", readLines(unzip(dl, "ml-10M100K/ratings.dat"))),
                 col.names = c("userId", "movieId", "rating", "timestamp"))

movies <- str_split_fixed(readLines(unzip(dl, "ml-10M100K/movies.dat")), "\\::", 3)
colnames(movies) <- c("movieId", "title", "genres")

# if using R 4.0 or later:
movies <- as.data.frame(movies) %>% mutate(movieId = as.numeric(movieId),
                                           title = as.character(title),
                                           genres = as.character(genres))

movielens <- left_join(ratings, movies, by = "movieId")


# Validation set will be 10% of MovieLens data
set.seed(1, sample.kind="Rounding") # if using R 3.5 or earlier, use `set.seed(1)`
test_index <- createDataPartition(y = movielens$rating, times = 1, p = 0.1, list = FALSE)
edx <- movielens[-test_index,]
temp <- movielens[test_index,]


# Make sure userId and movieId in validation set are also in edx set
validation <- temp %>%
  semi_join(edx, by = "movieId") %>%
  semi_join(edx, by = "userId")


# Add rows removed from validation set back into edx set
removed <- anti_join(temp, validation)
edx <- rbind(edx, removed)

rm(dl, ratings, movies, test_index, temp, movielens, removed)
```

For data cleaning and, data exploration and visualization

A quick overview of the data loaded shows the features present in the MovieLens dataset. It is observed that users who rate movies are represented with a userId, and each movie has a movieId. The rating score that each user gives a movie is also provided. The timestamp that marks the point in time when the movie was rated is also given, along with the movie title and genres of the movie. Note that the movie title contains the year of release in parentheses. This information will be extracted and stored as another feature. Note also that a movie can have multiple different genres.

A data.table: 6 × 6

userId	movieId	rating	timestamp	title	genres
<int>	<dbl>	<dbl>	<int>	<chr>	<chr>
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

Figure 1: Overview of Original Training Dataset

The structure of the data shows that userId is an integer, movieId and rating are floats, timestamp is an integer, and title and genres are characters.

```
Rows: 9,000,055
Columns: 6
$ userId    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, …
$ movieId   <dbl> 122, 185, 292, 316, 329, 355, 356, 362, 364, 370, 377, 420, …
$ rating    <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, …
$ timestamp <int> 838985046, 838983525, 838983421, 838983392, 838983392, 83898…
$ title     <chr> "Boomerang (1992)", "Net, The (1995)", "Outbreak (1995)", "S…
$ genres    <chr> "Comedy|Romance", "Action|Crime|Thriller", "Action|Drama|Sci…
```

Figure 2: Structure of Original Training Dataset

The number of distinct movies, users are shown in the table below:

Attribute	Number of Distinct Elements
Movie ID	10,677
User ID	69,878

Table 1: Distinct counts of movie IDs and User IDs

The dataset has 9,00,055 rows. It is possible for the same user to rate multiple movies. It is also possible for multiple users to rate the same movie. However, there may be bias that arises from users rating a movie, as well as, bias arising from one movie being rated more frequently by users. For clarification, suppose one user rates hundreds of movies while the average user rates only a few movies. Then the user that rates an abnormal number of movies will introduce bias into the dataset. On the other hand, suppose a movie is very popular and so attracts a large number of different users that rate this particular movie. Then this will also skew the results of the analysis if the average movie has a lower number of users rating it. These biases will be addressed further later on in the report.

In order to determine if age of the movie is a factor for predicting rating, I extracted the premiere date of the movie, and then calculated the age of the movie from this information. I also converted the timestamp to a premiere date and dropped the timestamp variable as it was no longer needed. In addition, I identified the primary genre of the movie by isolating the first genre that appears in the genre column. My hypothesis is that the primary genre is the most essential genre of the movie and may have an impact on user preference. Finally, I counted the number of genres that were assigned to each movie. This feature may be useful because a movie with a lot of genres may appeal to a large array of different audiences, or may perhaps affect the critical value of a movie when compared to a more focused movie with few genres.

A data.table: 6 × 11

userId	movieId	rating	title	genres	year_rated	premiere_date	age_of_movie	rating_date_range	primary_genre	genre_count
<int>	<dbl>	<dbl>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<int>
1	122	5	Boomerang (1992)	Comedy|Romance	1996	1992	26	4	Comedy	2
1	185	5	Net, The (1995)	Action|Crime|Thriller	1996	1995	23	1	Action	3
1	292	5	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller	1996	1995	23	1	Action	4
1	316	5	Stargate (1994)	Action|Adventure|Sci-Fi	1996	1994	24	2	Action	3
1	329	5	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi	1996	1994	24	2	Action	4
1	355	5	Flintstones, The (1994)	Children|Comedy|Fantasy	1996	1994	24	2	Children	3

Figure 3: Age of movie and genre details

The skewness in the distribution of movie ratings as well as user ratings are shown clearly in Figure 4 and Figure 5. Note that the x-axis is on a logarithmic scale so the distributions are not symmetric.

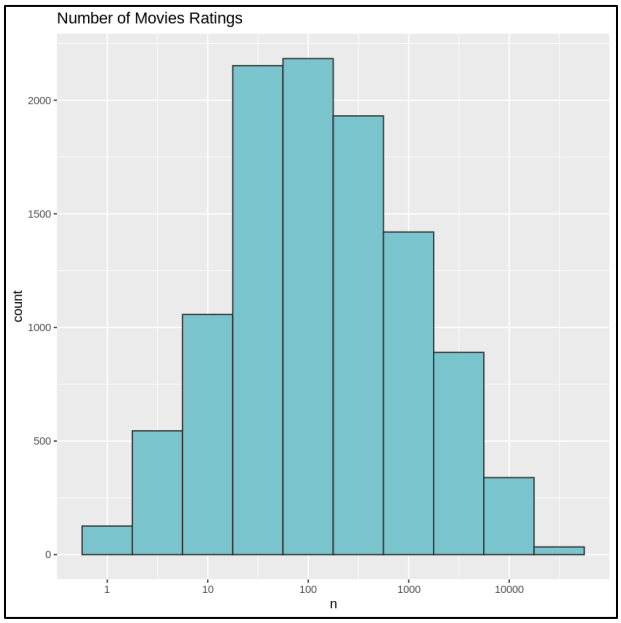


Figure 4: Distribution of Movie Ratings

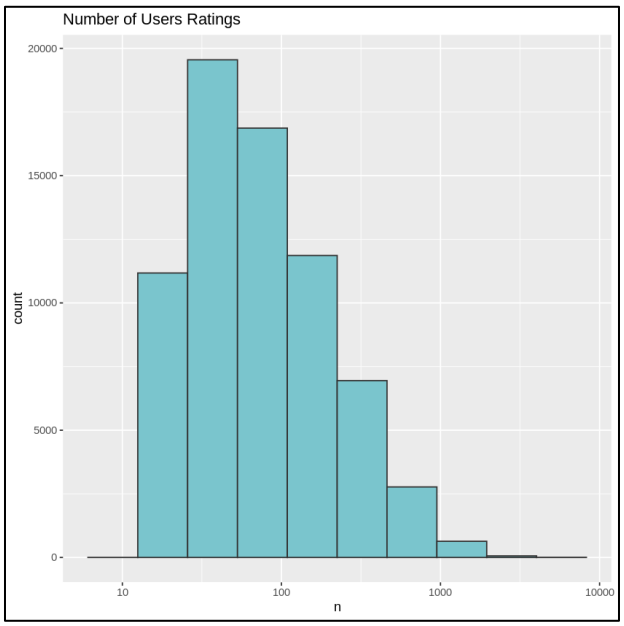


Figure 5: Distribution of User Ratings

An overview of the average rating by genre count shows that the average ratings are relatively low when there are only a few genres, and also when there are many genres. The average rating appears to be highest when the number of genres in a movie is around 3 or 4. There also seems to be a high rating when the number of genres is 7, and a very low rating when the genre count is 6.

genre_count	avg_rating_by_genre_count
1	3.46
2	3.48
3	3.54
4	3.55
5	3.59
6	3.33
7	3.52
8	3.45

Table 2: Movie rating averages by genre count

When exploring the impact of movie age on rating, it is seen that in general, newer movies tend to be rated more harshly than older movies. The trend is not linear since the average rating fluctuates a bit, suggesting that there might be seasonality in user's rating behavior depending on certain time periods.

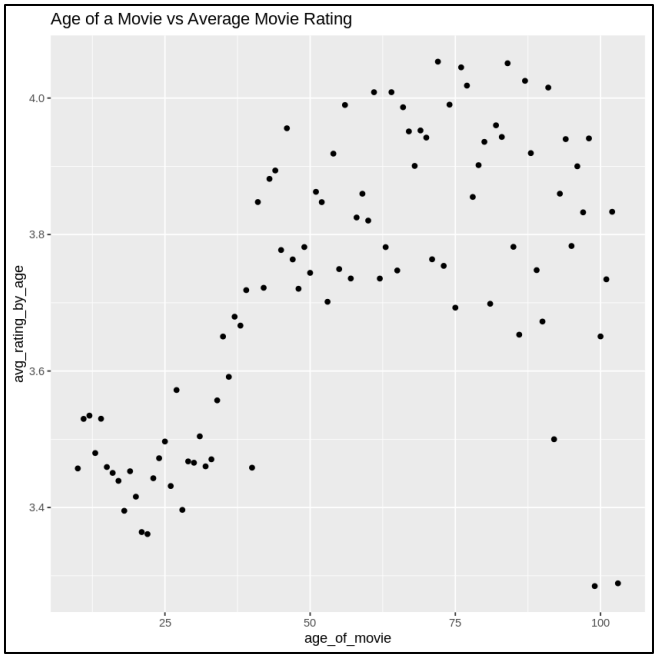


Figure 6: Relationship between the Age of Movie and Average Movie Rating

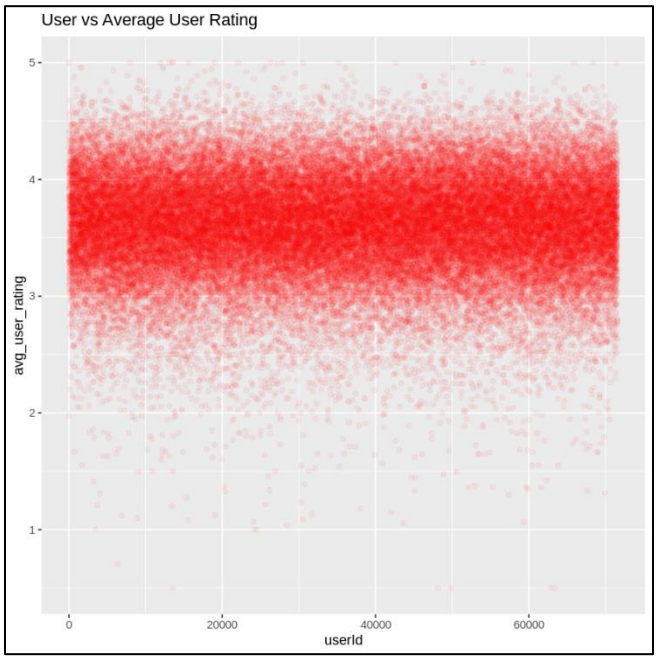


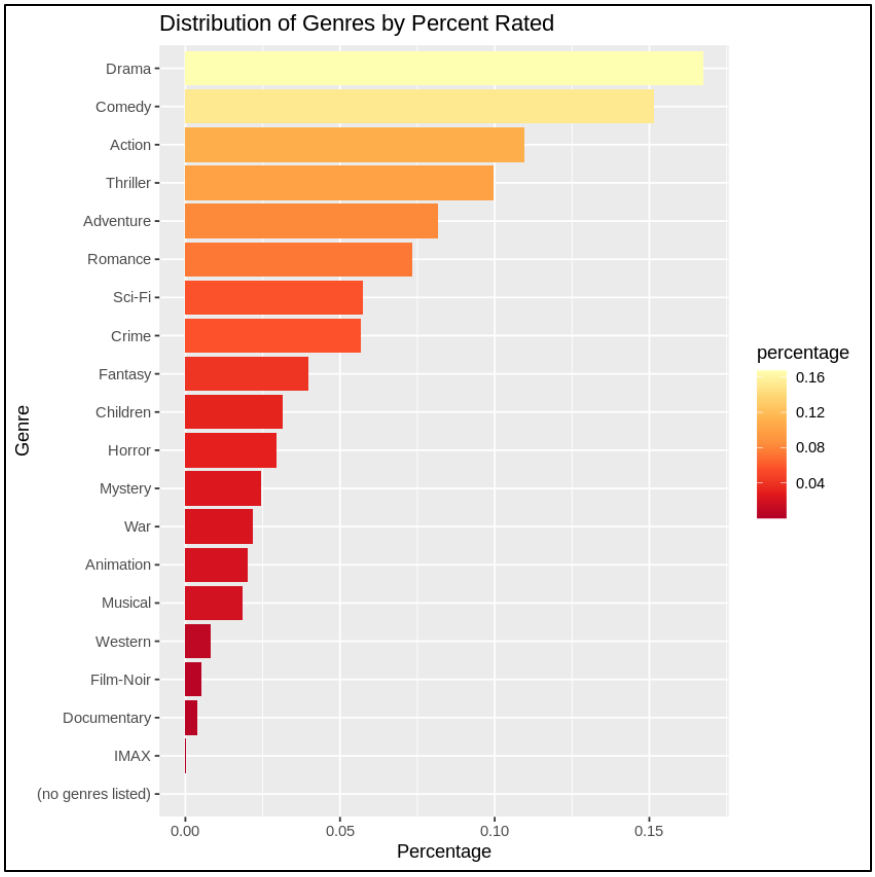
Figure 7: UserId vs Average Movie Rating

In Figure 7, we see that most user ratings are approximately around 3.5 with a relatively symmetric distribution where most ratings are between 2.5 and 4.5. This suggests that user ratings are well-behaved and normal, with few outliers.

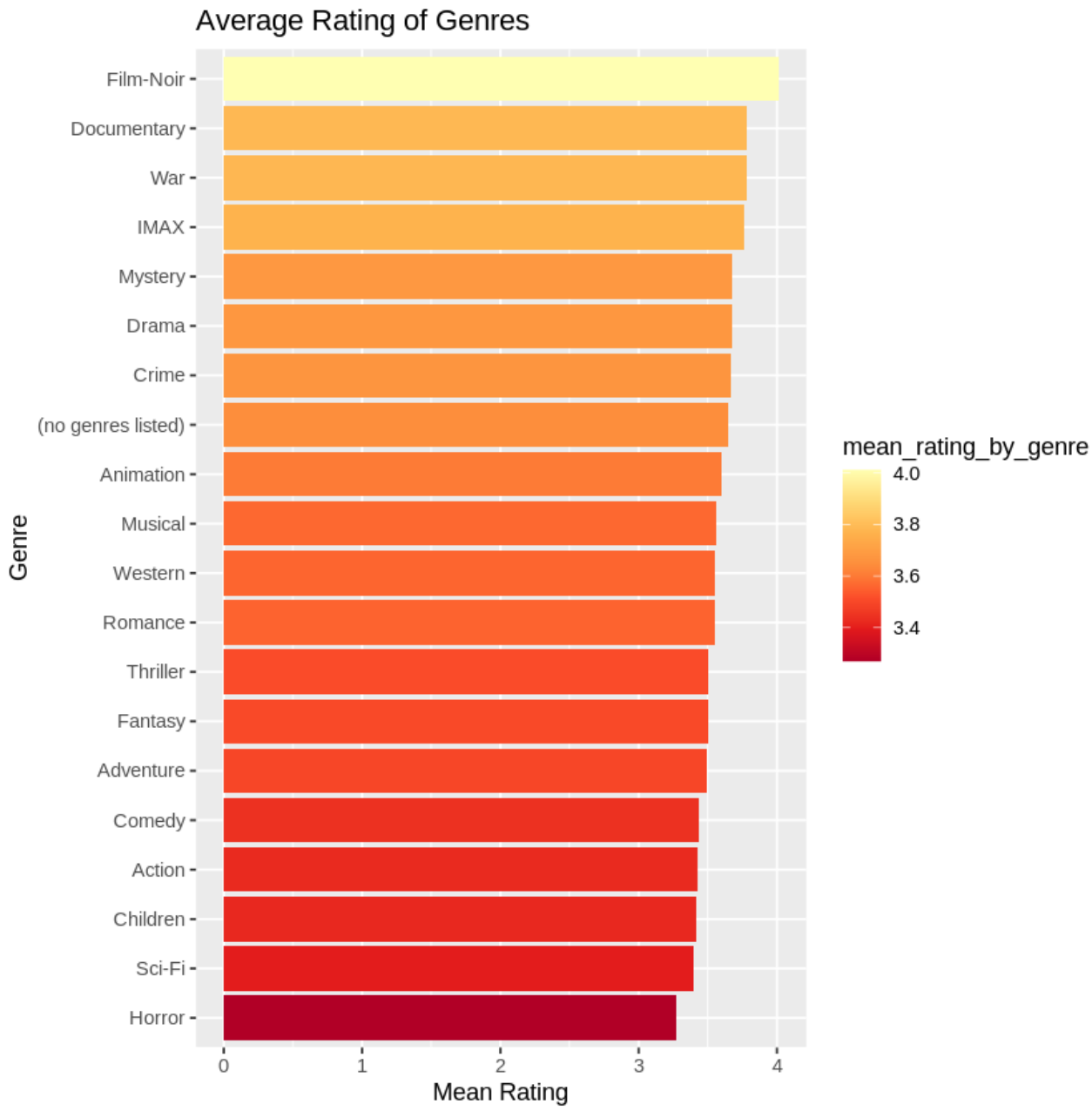
Genre	Number of Movies with Genre
No Genre Listed	7
Action	2560545
Adventure	1908892
Animation	467168
Children	737994
Comedy	3540930
Crime	1327715
Documentary	93066
Drama	3910127
Fantasy	925637
Film-Noir	118541
Horror	691485
IMAX	8181
Musical	433080
Mystery	568332
Romance	1712100
Sci-Fi	1341183
Thriller	2325899
War	511147
Western	189394

Figure 8: Distribution of Movies by Genre

From Figure 8, it is observed that some genres such as Drama, Comedy, Romance, Thriller, Crime, Action and Adventure are most dominant in the dataset.

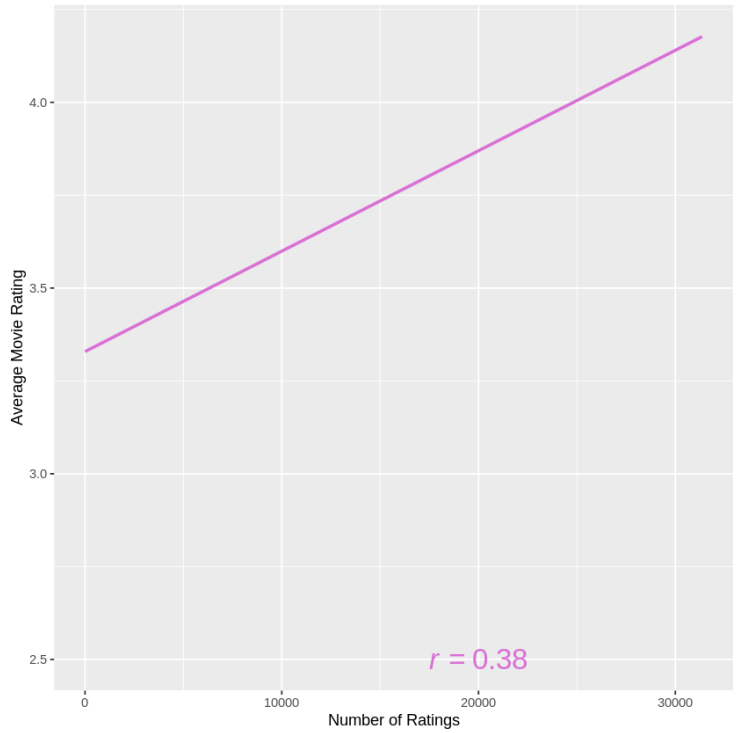


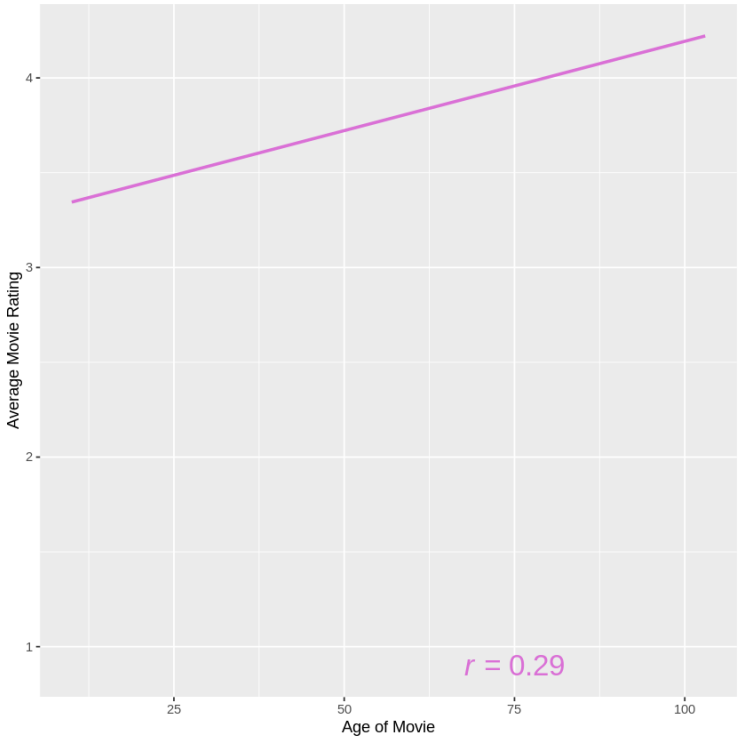
Distribution of Ratings per Genre


Average Rating of Genres
Film-Noir
Documentary
War
IMAX
Mystery
Drama
Crime
(no genres listed)
Animation
Musical
Western
Romance
Thriller
Fantasy
Adventure
Comedy
Action
Children
Sci-Fi
Horror
Genre
0
1
2
3
4
Mean Rating
mean_rating_by_genre
4.0
3.8
3.6
3.4

To determine if there a relationship between number of ratings and the average rating

Number of ratings vs avg movie ratings

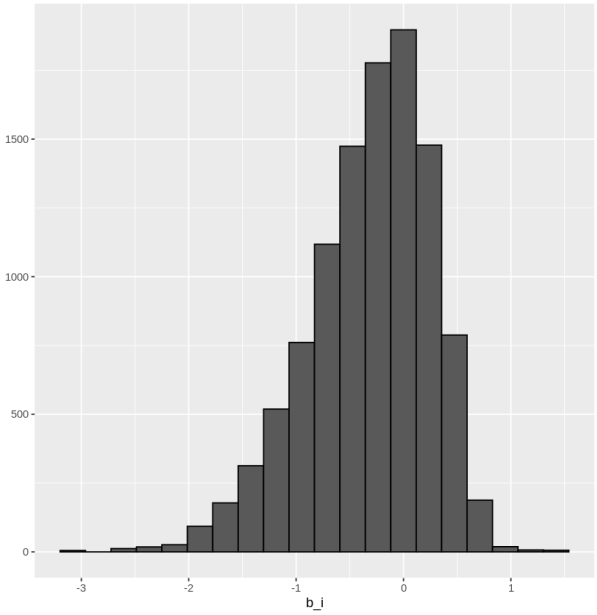




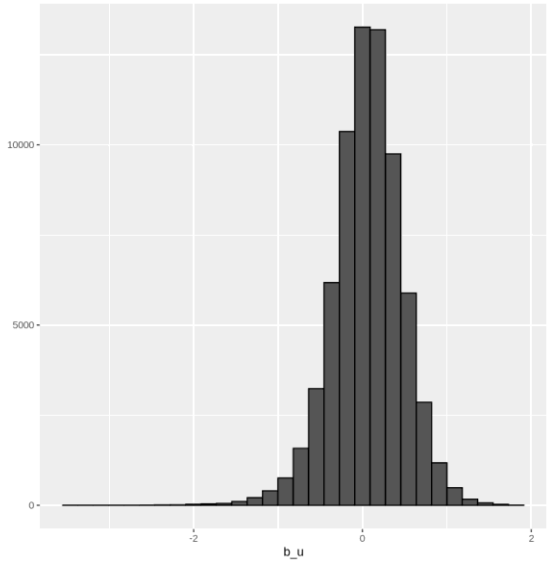
To derive the simplest possible model, the dataset's mean rating is used to predict the same rating for all movies, regardless of the user and movie

3.51246520160155

Movie Average - Norm



User Average – Norm



Baseline Model: just the mean
Test results based on simple prediction

1.06120181029262

Check results

A tibble: 1 × 2

method	RMSE
<chr>	<dbl>
Using mean only	1.061202

Movie effects only

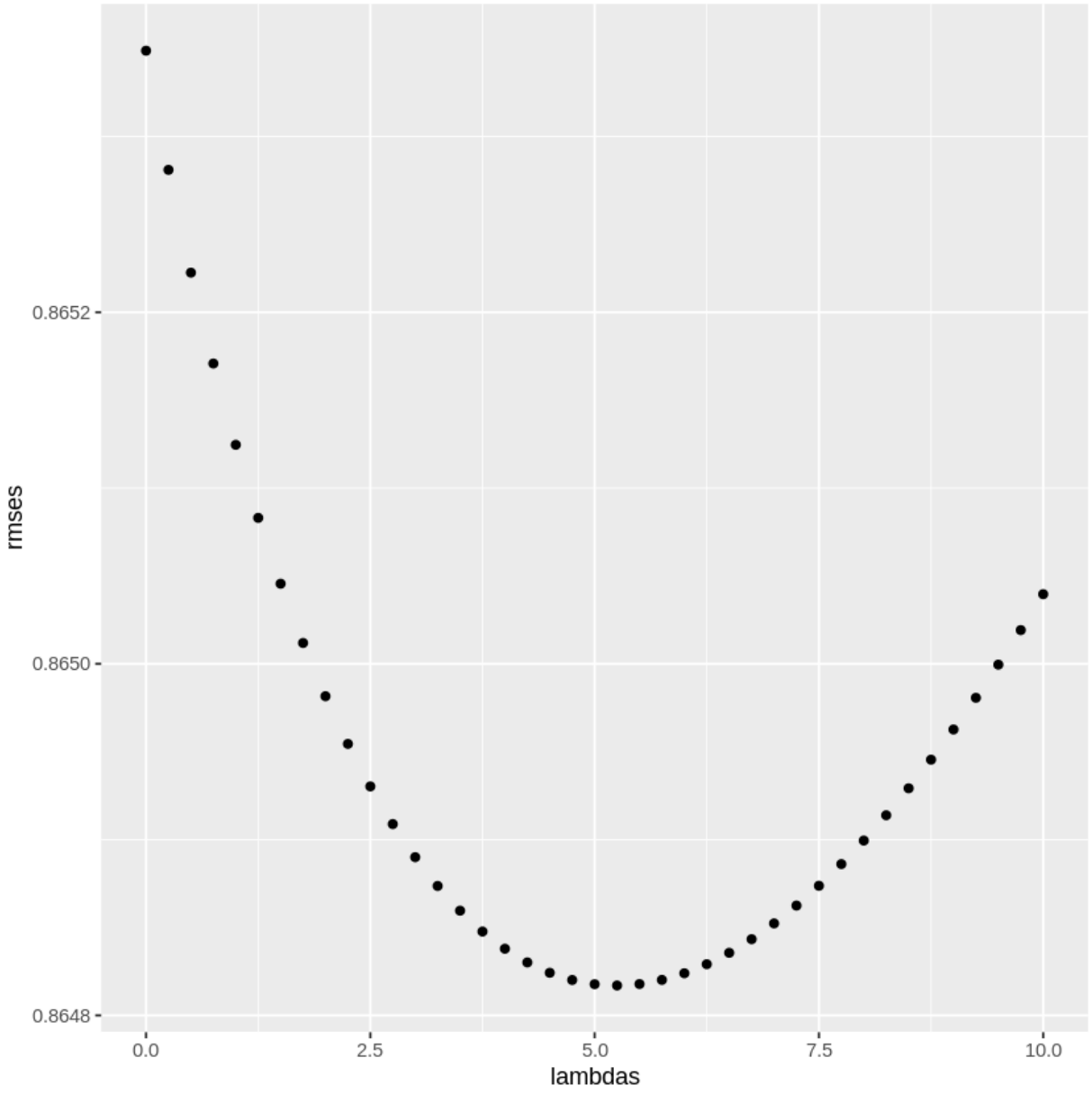
method	RMSE
Using mean only	1.0612018
Movie Effect Model	0.9439087

Use test set,join movie averages & user averages
Prediction equals the mean with user effect b_u & movie effect b_i
Test and save rmse results

method	RMSE
Using mean only	1.0612018
Movie Effect Model	0.9439087
Movie and User Effect Model	0.8653488

lambda is a tuning parameter
Use cross-validation to choose it.
For each lambda,find b_i & b_u, followed by rating prediction & testing

Plot rmses vs lambdas to select the optimal lambda

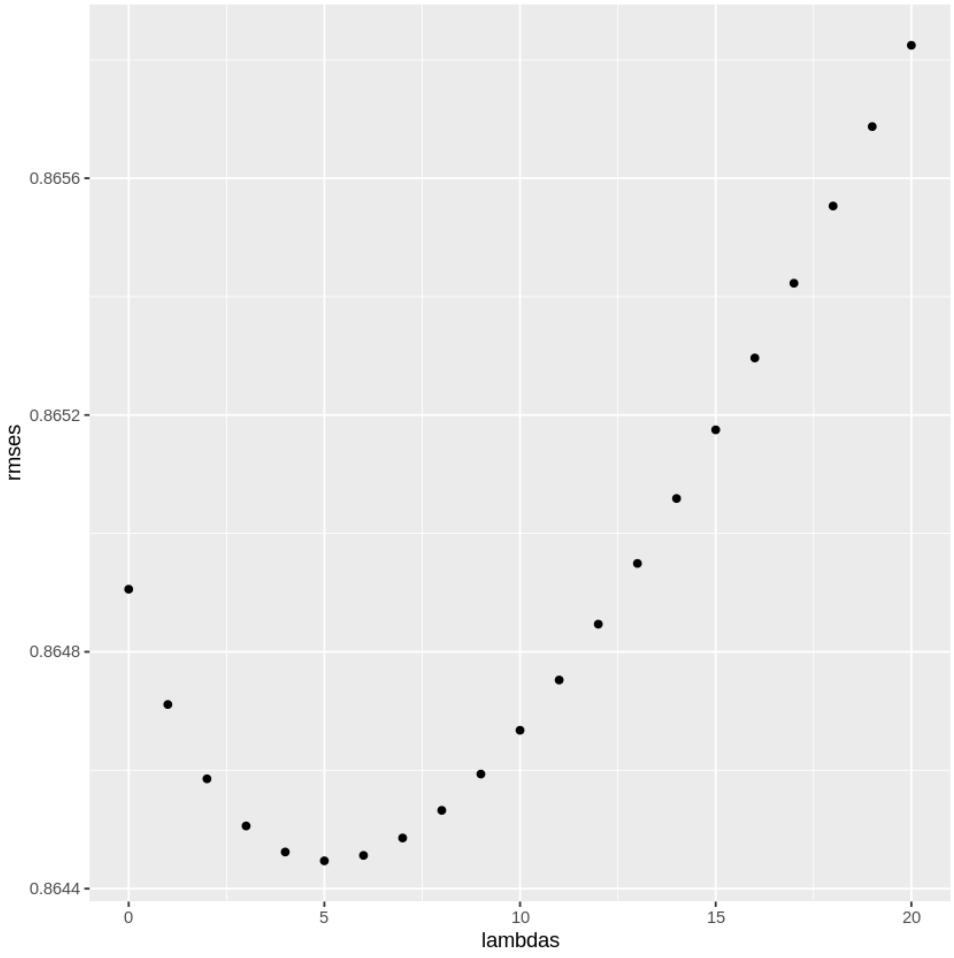


Lambda
5.25

Compute regularized estimates of b_i using lambda
Compute regularized estimates of b_u using lambda
Predict ratings
Test and save results

method	RMSE
Using mean only	1.0612018
Movie Effect Model	0.9439087
Movie and User Effect Model	0.8653488
Regularized Movie and User Effect Model	0.8648170

b_y and b_g represent the year & genre effects, respectively
Compute new predictions using the optimal lambda
Test and save results



lambda_2
5

method	RMSE
Using mean only	1.0612018
Movie Effect Model	0.9439087
Movie and User Effect Model	0.8653488
Regularized Movie and User Effect Model	0.8648170
Reg Movie, User, Year, and Genre Effect Model	0.8644304

Model Results

method	RMSE
Using mean only	1.0612018
Movie Effect Model	0.9439087
Movie and User Effect Model	0.8653488
Regularized Movie and User Effect Model	0.8648170
Reg Movie, User, Year, and Genre Effect Model	0.8644304
Reg Movie, User, Year, Genre and Genre Count Effect Model	0.8644266

The RMSE generated from the predictive model developed for this project was **0.8644266**; which is below the target RMSE of 0.86490.

This means the models performs well when recommending movies to users.
The model was effective due to regularization of certain effects which reduced the bias in the data.
There are several ways to improve the model. More feature engineering could be done and splitting the data into separate genres.

Conclusion

In this project I developed a recommender system to recommend movies to users. The accuracy of how well a movie would be rated by the user was measured using RMSE. Biases were accounted for through the capture of effects and regularization. The RMSE of the final model was proven to be robust. This RMSE was found to be **0.8644266.**

Future Applications of this include recommending telecommunications products to customers and recommending products on Amazon.