MovieLens Project

Ellis Hodgdon

Date: Fri 08 Nov 2024

Overview

Problem Statement In the analysis of the scenario, we will be chasing the statistical quantity of Root Mean Square Error, or as mathematicians/statisticians prefer to call it, RMSE. It, and its close cousin, AME (Mean Absolute Error) have been embroiled in controversy for quite some time. Which one is better? Willmott and Matsuura give arguments favoring one metric over the other, but, in reality, neither metric is inherently better (Willmott 2005). RMSE is optimal for normal (Gaussian) errors while MAE is optimal of Laplacian errors. When the errors don't follow one of these patterns,there are other metrics out there that are better (Hodson 2022). For this specific analysis, the RMSE will be used.
A standard definition of RMSE is that it measures the average difference between a statistical model's predicted value and the actual values. Mathematically, it is the standard deviation of the residuals, which represent the distance between the regression line and the data points (Frost 2023). It can be calculated by the formula:

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

The calculation is not complicated and with a limited number of samples, can be done in Excel. However, when the number of samples gets large, as in this analysis, R has its RMSE function to do the calculations. Our goal in this effort is to develop an algorithm that will produce a RMSE of a final_holdout_test dataset of 0.86490 or lower.

Method

The download and data wrangling code is provided by the course and is designed to join the *ratings* dataset with the *movies* dataset, This code has been augmented with code that will process other datasets that are in the same general format. Functions are used within the R code to provide flexibility and the make the code easier to read.

Read external dataset

```
## [1] "Dataset ml-10m will be reloaded from disk"
```

Data Sets The MovieLens dataset (Harper (2015)) contains 10,000,054 rows and 6 columns to which two columns will be added during data wrangling. There are technically three separate datasets which are all created from the combined dataset, *movielens.*These three datasets are identified as the training set *(edx)* that contains 9,000,055 rows, the testing set *(testing)* which is used to test the training, and the final dataset *(final_holdout_test)*, which is the validation dataset. There is no duplication of rows in any of the datasets and the final is not used for any training but only to validate the algorithm on the analysis. The size of the testing and final datasets will be 10% of the total size of *(movielens).* This can be changed, if necessary, by modifying the parameters in the YAML header. Just in case there was a problem with generating the combined dataset, a quick check verifies that no rows from the *(movies)* dataset were omitted from the join

with the *(ratings)* dataset.
Two features were added to the training dataset: year of release and rating lag time. The year of release and subsequently the lag time are derived from the title feature. When these features are added, a check is made and no titles that do not fit this pattern were found .

List of features After the movies and ratings set have been combined, any "dirty" rows removed, and any other data wrangling performed, the first eight rows of the dataset that will be used for training loos like:

	userId	movieId	rating	timestamp	title	genres
1	1	122	5	838985046	Boomerang (1992)	Comedy|Romance
2	1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
4	1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
5	1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
6	1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi
7	1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy
8	1	356	5	838983653	Forrest Gump (1994)	Comedy|Drama|Romance|War
9	1	362	5	838984885	Jungle Book, The (1994)	Adventure|Children|Romance

Table 1: Description of Features

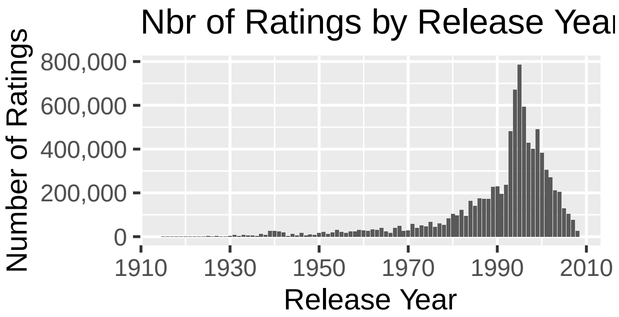
feature	class	description
userId	integer	unique ID for each movie
movieId	numeric	unique ID for each user
rating	numeric	rating (0.5-5) that a user gave a movie
timestamp	integer	timestamp as to when the user rated the movie
title	character	title of the movie and year of release
genres	character	the genre or genres assigned to the movie
ratingLag	numeric	time between the release date and the date the movie was rated by the user
releaseYear	date	the year of release

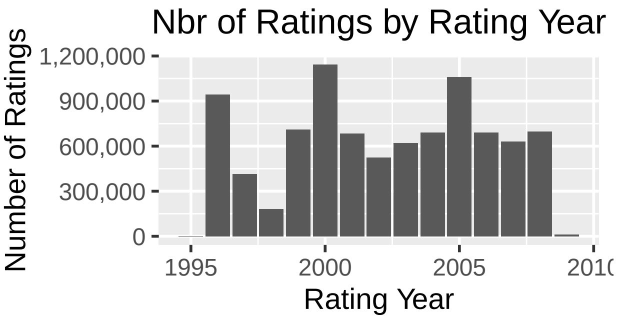
```
## Warning in geom_histogram(stat = "count", bins = 30, binwidth = 0.3, color =
## "black"): Ignoring unknown parameters: `binwidth`, `bins`, and `pad`
```

Exploratory Data Analysis

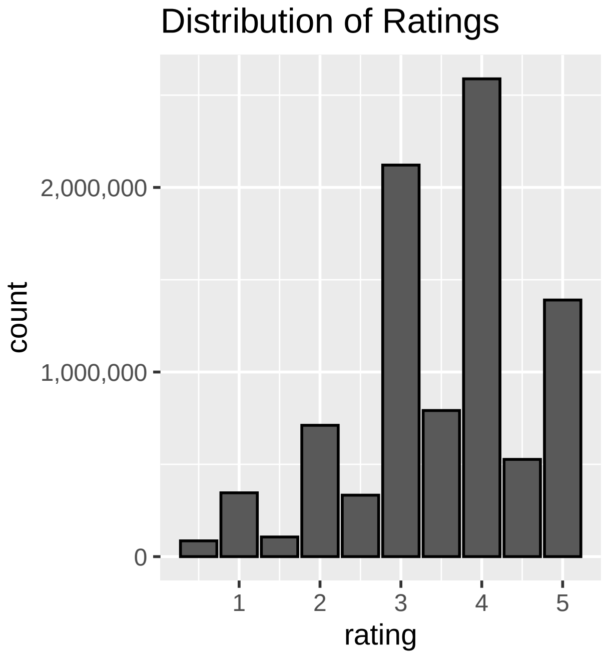
Exploratory Data Analysis (EDA) is not a formal process with a strict set rules but rather a state of mind where the analyst should investigate every idea that pops up. Some will work out; some will be duds, but it is important part of any data analysis because you always need to investigate the quality of the data (Wickham, Centinkaya-Rundel, and Grolemund 2023). For this analysis,there are three components of the investigation: the movies, the users, and the ratings.
We start visualizing our EDA by first looking at the raters, a.k.a. users, of this data. The users were selected at random and there is no demographic information about a rater other than the user ID, so we are left with the user ID and their actions. In this dataset, there are 69,878 unique raters that have made 10,000,054 ratings in the period of 1995 to 2009. The following graphs show the number of raters the participated by year and the length of time that a user remains active in the rating process from a minimum of 0 days (24 raters have this duration) to a maximum of 4,122.22 days(approximately 11 years).

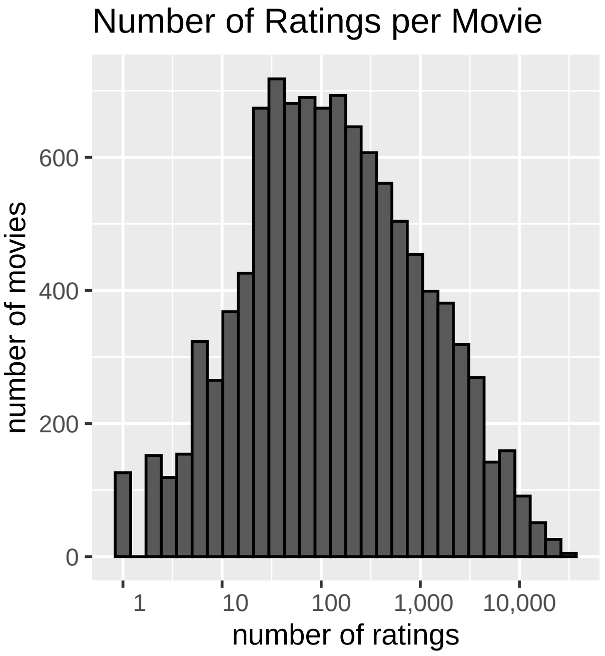
We next look at how these ratings were spread out over the years. The two graphs that follow show the number ratings by the year the movie was released and the number of ratings by calendar year.


Nbr of Ratings by Release Year

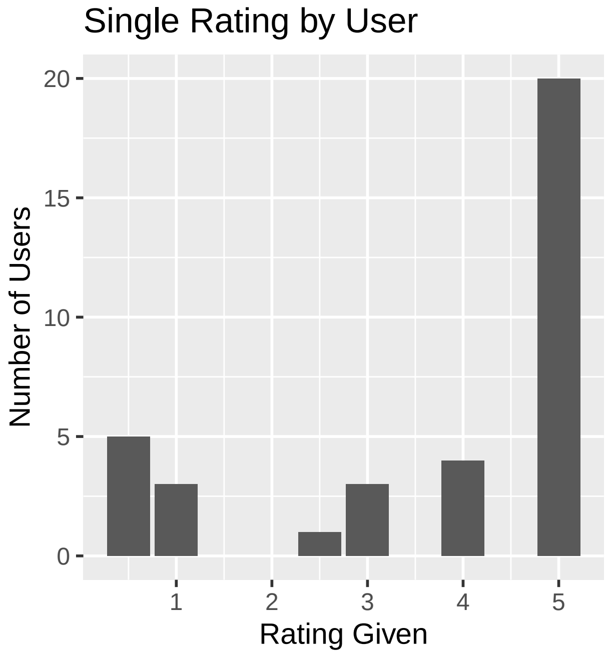

Nbr of Ratings by Rating Year

Now we examine how these ratings are applied to the various movies. The movies are rated on a scale between 0 and 5 in 0.5 increments. The first graph shows the distribution of the rating values. The distribution is not normal by any means and the graph reveals that using a ½ rating was not a prevalent as a full integer rating. Not all movies received the same number of ratings, but the number of movies that did receive a rating as shown in the second graph indicates that there is sufficient data to proceed with the analysis.

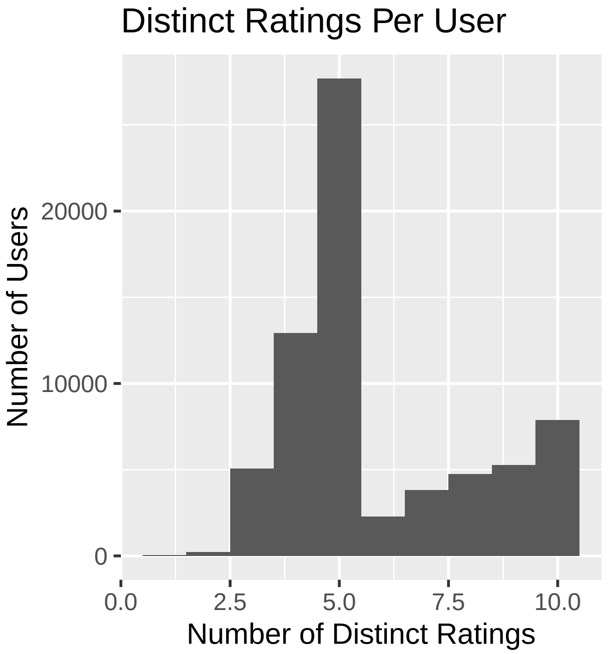

Distribution of Ratings


Number of Ratings per Movie

The next two graphs show how a particular rater would rate a movie. When a rater gives only a single rating, the first graph shows the distribution of that rating. This could affect the objectivity of the rating. To look at this somewhat closer we find that there are a total of 36 which represents 0.0004%, which is a rather insignificant component and will be ignored. When a user gave different ratings, of the ten possible ratings, how many were used. The second graph shows this distribution.

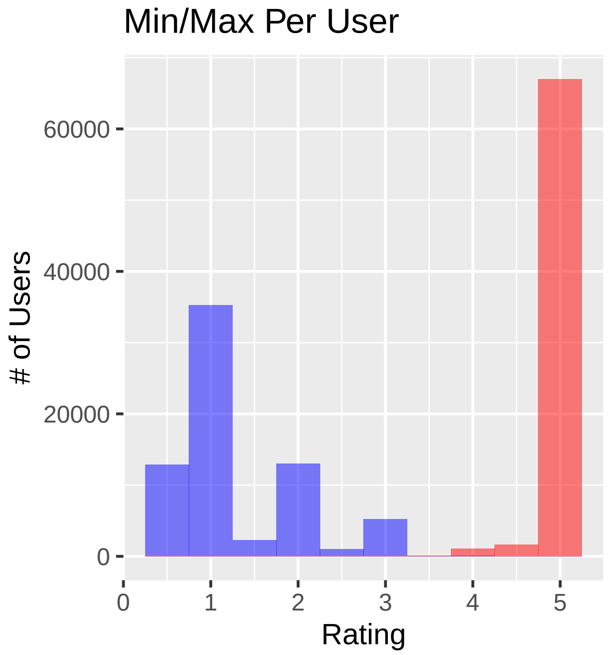


Single Rating by User

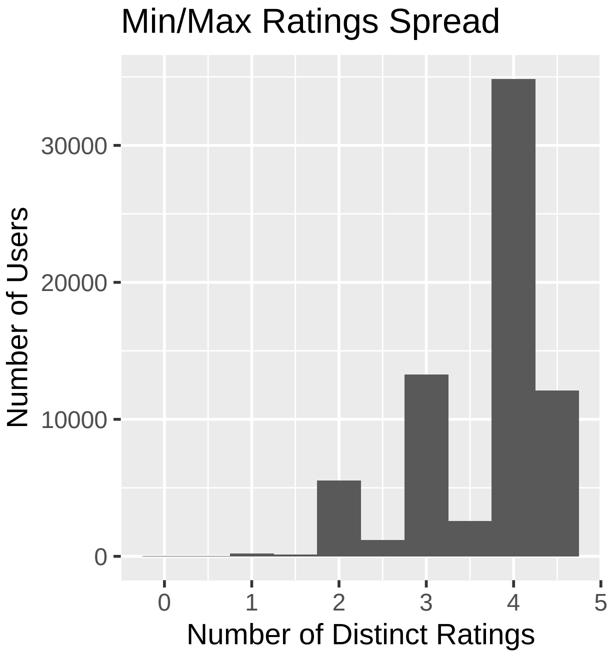


Distinct Ratings Per User

The first plot below shows how a particular rater spread the ratings that were given. The second graph shows for those users that did not give a single rating how many of the ten (0 - 5) ratings were used.

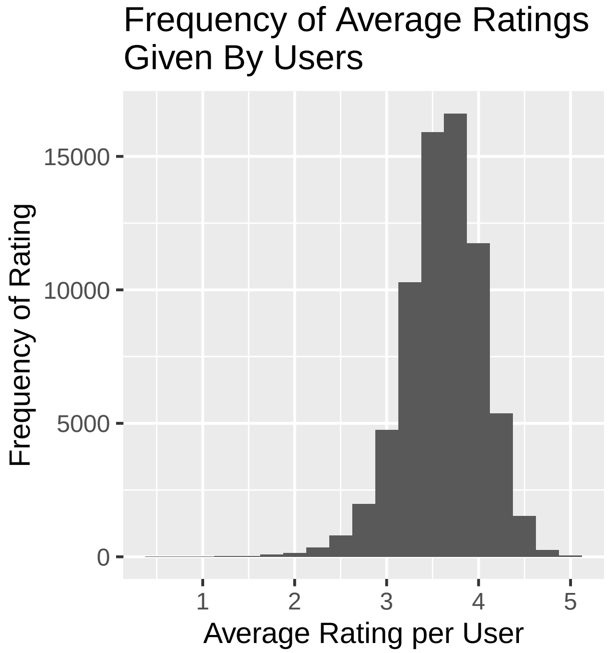


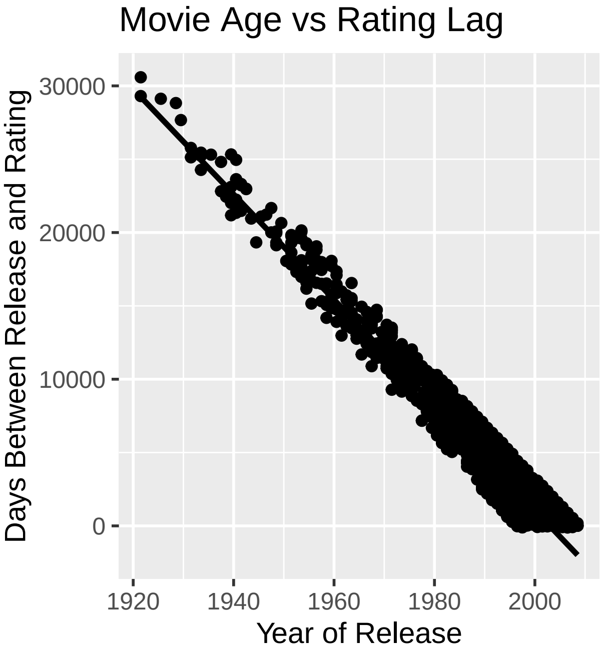
Min/Max Per User



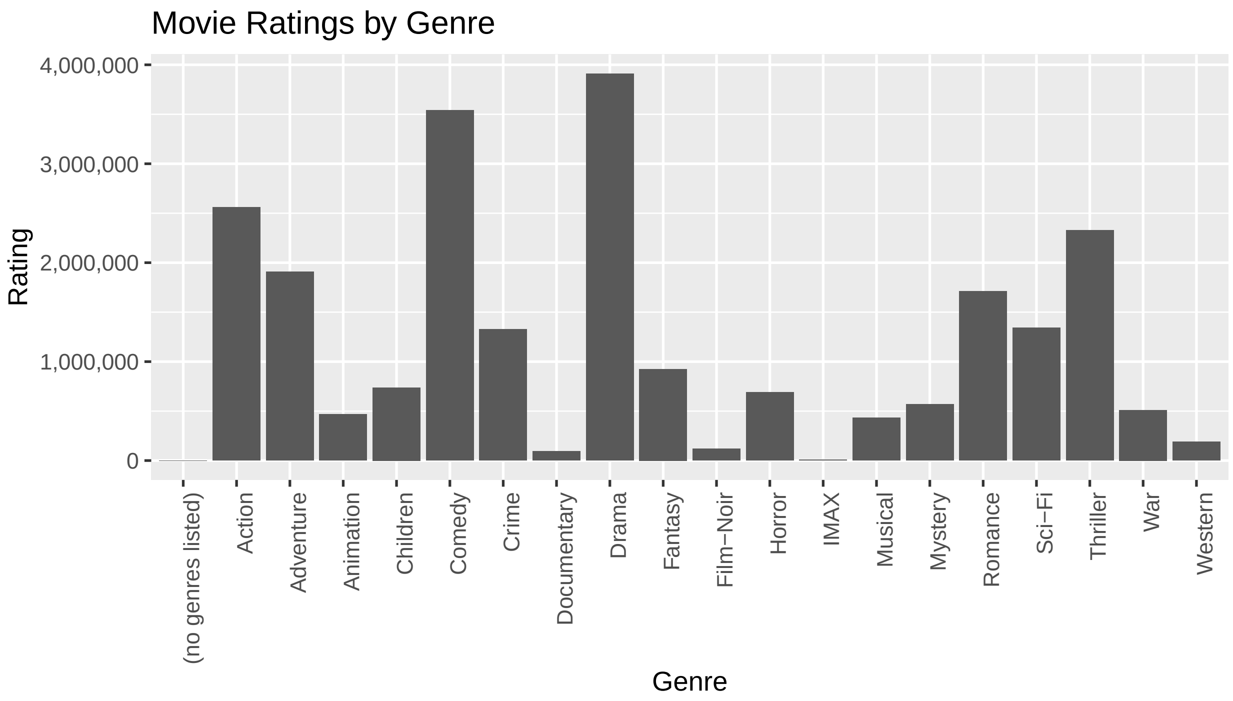
Min/Max Ratings Spread

We've seen what the distribution of ratings over all users, but we can split that down further by looking at the average distribution by an individual user as shown in the first graph below. Another correlation that needs be investigated is that rating lag, that is, the time between the release of the movie and the time the movie was rated by a rated. That correlation is shown in the second graph and shows that rating lag is not a significant feature to be analyzed.





There are 19 genres that can be assign to a movie and a movie can have one or more genres assigned to it. The assignment of genre is usually done by the studio and/or writers and is determined by the literary technique, tone, content and some times by length (Williams 2024). A break down of the movies that are in each genre is below.



Since each movie can have multiple genres, the dataset needs to be expanded so that there is only one genre per rating. This is only to determine the effect of the genre feature on the analysis; for all other features the unexpanded dataset is used.

We now need to examine the data for any anomalies that might cause us problems later. We want to work with tidy data so we need to verify that the training dataset, *edx*, is tidy. To be tidy, three conditions must be met:

- Each variable must have its own column
- Each observation must have its own row
- Each value must have its own cell

Preliminary review The earliest movie in the dataset is The Birth of a Nation that was released in 1915 and the maximum number of ratings (9,000,055) occurred in the year 1994.

Included in data set are some obscure movies that could be considered to be removed from the majority of movies. There are 126 (0.0014% of the working dataset) that have only a single rating in the dataset. A table of 15 of these movies and the statistics on their ratings follows:

title	rating	n_rating
1, 2, 3, Sun (Un, deuz, trois, soleil) (1993)	2.0	1
100 Feet (2008)	2.0	1
4 (2005)	2.5	1
Accused (Anklaget) (2005)	0.5	1
Ace of Hearts (2008)	2.0	1
Ace of Hearts, The (1921)	3.5	1
Adios, Sabata (Indio Black, sai che ti dico: Sei un gran figlio di...) (1971)	1.5	1
Africa addio (1966)	3.0	1
Aleksandra (2007)	3.0	1
Bad Blood (Mauvais sang) (1986)	4.5	1
Battle of Russia, The (Why We Fight, 5) (1943)	3.5	1
Bellissima (1951)	4.0	1
Big Fella (1937)	3.0	1
Black Tights (1-2-3-4 ou Les Collants noirs) (1960)	3.0	1
Blind Shaft (Mang jing) (2003)	2.5	1

The summary stsatistics for these obscure movies is:

Avg_Rating
Min. :0.500
1st Qu.:3.357
Median :3.635
Mean :3.614
3rd Qu.:3.903
Max. :5.000

Removing these obscure movies reduces the number of rows in the training dataset to 8,999,929 rows.

Modeling

Data modeling refers to the process of mapping data at typically three levels: conceptual, logical, and physical. It creates visual maps and references that allow data analysts to visualize a data system and is a process by which data systems can be interconnected (Pierson 2024). We will work several models that pertain to features that could influence the recommendation.

Model 0 - Naive model This is a simple, naive models which is predicting average movie rating for all observations. The formula for this model is simply

$$Y = \mu$$

```
mu <- mean(training_set$rating)
zero_rmse <- RMSE(testing_set$rating, mu)
zero_rmse
```

```
## [1] 1.059644
```

Model 1 - Average movie rating model The formula for this model is

$$Y = \mu + b.movie_i$$

where μ is the mean of the data set (model 0), and $b.movie_i$ is an error term that describes the random variability.To improve this RMSE, we will consider adding any effects that the movie may have on this value. There was a hint of this effect when we noticed the more ratings that a movie receives, the higher they often rated. We introduce a movie bias term, b_{movie}, which is the difference between the average of a specific movie$_i$ and the average for all movies.

RMSE Movie Bias Calculation We can now predict the rating with μ, and we can obtain the RMSE for Model 1:

```
predicted_ratings <- mu + testing_set %>%
                     left_join(movie_bias_df, by = "movieId") %>%
                     .$bm
movie_bias_rmse <- RMSE(predicted_ratings, testing_set$rating)
movie_bias_rmse
```

```
## [1] 0.9440223
```

Model 2 - Model 1 with user effect model We've got a RMSE for the average movie rating, but it is over our objective is to be below 0.86490, so we need to continue. We have noticed that a group of users the consistently give low ratings and there are those that consistently give higher ratings. Will use the symbol $b.user_u$ as this bias for user u.

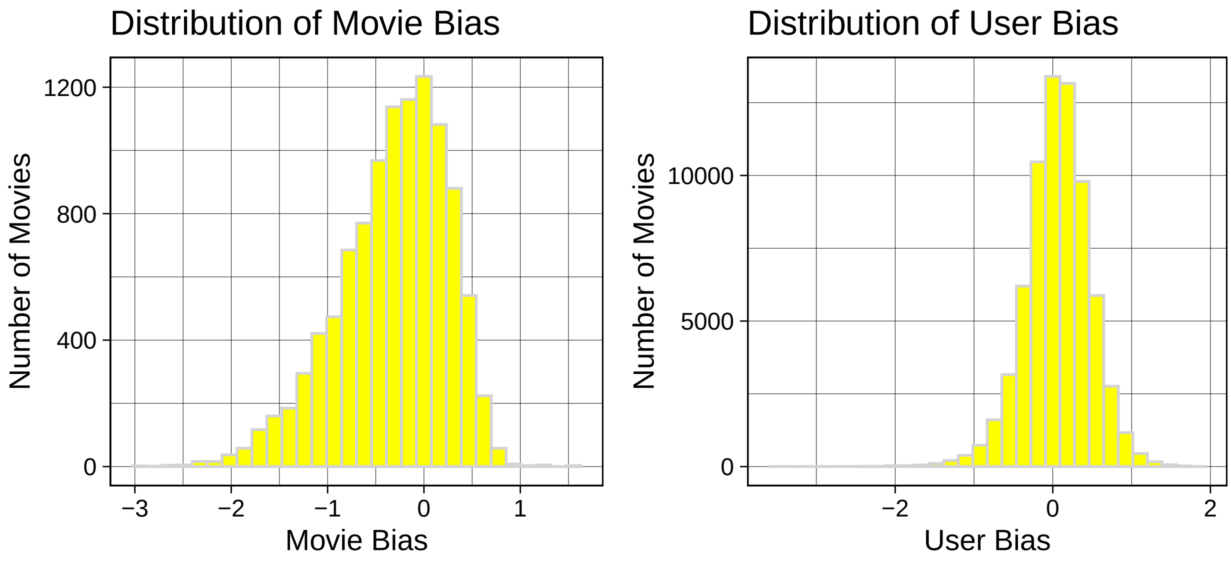
$$Y_i = \mu + b.movie_i + b.user_i$$

```
predicted_ratings <- testing_set %>%
                     left_join(movie_bias_df, by='movieId') %>%
                     left_join(user_bias_df, by="userId") %>%
                     mutate(pred = mu + bm + bu) %>%
                     .$pred
user_bias_rmse <- RMSE(predicted_ratings, testing_set$rating)
user_bias_rmse
```

RMSE User Bias Calculations

```
## [1] 0.8653934
```

Graphs for both the movie bias and user bias are shown below.



The RMSE is looking better. It is below 1 but still above our goal. Another model is in order.

Model 3 - Model 2 with genres effect model To the movie 2 model, we will add a genres effect term.The formula now becomes:

$$Y_{u,i} = \mu + b.movie_i + b.user_u + b.genre_{u,i}$$

```
predicted_ratings <- testing_set %>%
                      left_join(movie_bias_df, by="movieId") %>%
                      left_join(user_bias_df, by='userId')  %>%
                      left_join(genres_bias_df, by="genres") %>%
                      mutate(pred = mu + bu + bm + bg) %>%
                      .$pred
genres_bias_rmse <- RMSE(predicted_ratings, testing_set$rating)
genres_bias_rmse
```

RMSE Genres Bias Calculation

```
## [1] 0.8651024
```

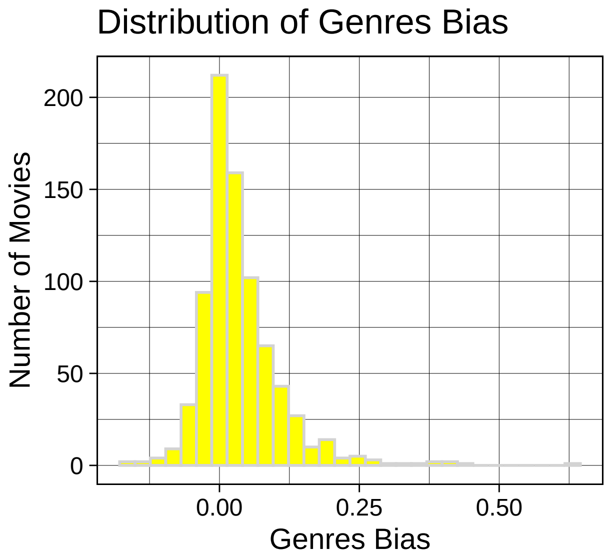
Model 4 – Model 3 with year of release bias The graph for the distribution of the year of release bias on the data and indicates that some improvement would be made if this bias was included in the analysis.

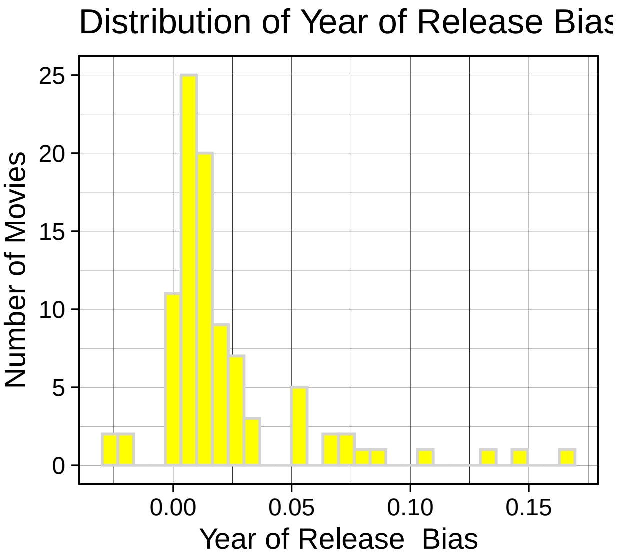
```
predicted_ratings <- testing_set %>%
                     left_join(movie_bias_df, by="movieId") %>%
                     left_join(user_bias_df, by='userId') %>%
                     left_join(genres_bias_df, by="genres") %>%
                     left_join(year_bias_df, by = "releaseYear") %>%
                     mutate(pred = mu + bu + bm + bg + by) %>%
                     .$pred

releaseYear_rmse <- RMSE(predicted_ratings, testing_set$rating)
releaseYear_rmse
```

RMSE Year Released Bias Calculations

```
## [1] 0.8649158
```

The graphs for the genres and release year biases look different than the previous graphs.



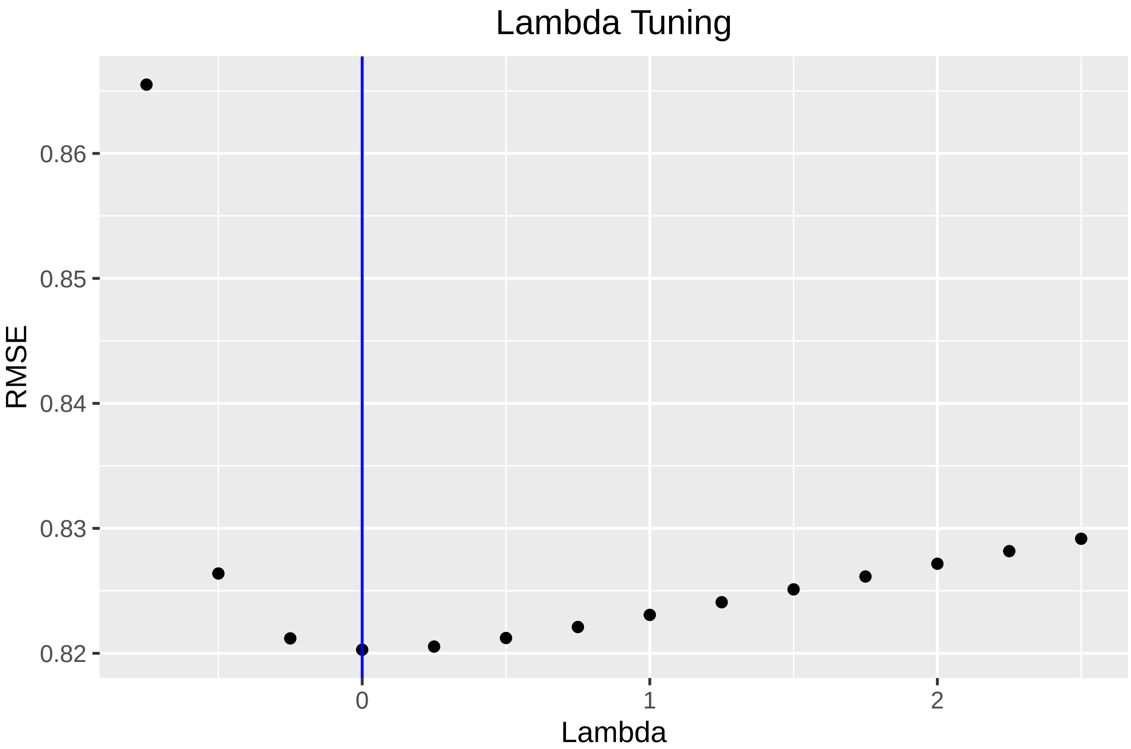


Regularization of Model 4

This is better than before but still above the goal of 0.84690. To bring that value down below the goal, we will apply regularization that adds constraints to total variability of the various effects. Regularization is a set of methods for reducing over fitting in models such as this one. Typically, regularization exchanges a marginal decrease in training accuracy for an increase in generalizability (Muriel. J. 2023). We implement regularization by include a new term lambda, λ into our equation which now becomes

$$\frac{1}{N}\sum_{u,i}(y_{u,i} = \mu - b.movie_i - b.user_u - b.genres_{u,i})^2 + \lambda(\sum_i + b.movie_i^2 + \sum_i b.user_u^2)$$

where $b.movie_i$ is influenced by movies that have jut a few ratings. $b.user_u$ is influenced by those users who only rated a small number of movies. $b.genres_{u,i}$ is influenced by the genres that are not often associated with a movie such as IMAX. The use of regularization allows us to penalize these effects. Consequently, we can use λ as a tuning parameter that, by adjusting, allows us to minimize the RMSE. A range of λ's were evaluated to determine witch value produced by lowest accuracy. The graph of these tests is shown below.



```
## Minimum lambda:  0
```

```
## [1] 0.8202878
```

Results

Now that we have trained the algorithm with the training dataset, it is time to validate it. For this we use the final_holdout_test dataset which has not been used up to this point. We apply the same preparation to this dataset as was applied to the training dataset, i.e., converting the timestamp, adding rating lag and year release, removing obscure movies, and expanding the genres column. We then apply the training algorithm that was developed.

Tab

Method	RMSE	
No biases applied - Median Alone	1.05964	
Median + Movie Effects	0.94402	
Median + Movie + User Effects	0.86539	
Median + Movie + User + Genres Effects	0.86510	
Median + Movie + User + Genres Effects + Release Year	0.86492	
Median + Movie + User Effects + regularization	0.82029	
final_holdout_test	0.81796	
Using a λ of 0, we determine a rmse of 0.820288	for the training set. It is now time	for the grandiose test – ou

Other tests An analysis of the supplied dataset with the obscure movies removed modified is complete. Other different, but similar, datasets are available from the grouplens website, https://grouplens.org/datasets/movielens/ which contain more data or are more recent. The analysis described here was used to analyze some of those datasets and the results are published here:

Dataset	Rows	Obscure Removed	Training Set	final_holdout_test Set	Date	Time
ml-10M100K	8,999,929	Yes	0.82029	0.81796	2024-10-22	6.20 mins
ml-10M100K	10,000,054	No	0.81992	0.81796	2024-10-23	5.95 mins
ml-32M-download	28,749,912	Yes	0.83488	0.81451	w024-10-23	25.89 mins
ml-32M-download	TBD	No	0.81598	0.81446	2024-10-21	24.97 mins?
ml-latest	33,832,162	Yes	0.80875	0.80921	2024-10-21	30.00 mins
ml-latest	30,419,985	No	0.80875	0.80921	2024-10-23	34.51 mins
ml-latest-small	87,436	Yes	0.65521	0.68537	2024-10-22	0.12 mins
ml-latest-small	91,112	No	0.65892	0.67877	2024-10-22	0.12 mins

Conclusion

For this project, we built a machine learning model that used the MovieLens Dataset to forecast movie ratings that takes into account any user movie bias, user bias, or genres bias and release year bias. The goal was to obtain an accuracy of 0.86490 or lower, which was accomplished.

Appendix

System Information RStudio: version 2024.09.0 Build 375
R: version 4.3.2 (2023-10-21 ucrt) – "Eye Holes"
Windows 11 Pro Version 22H2 OS Build 22621.4317 64 bit operating system 64 bit processor
Dell 5431 Processor 2.20 GHz Memory 32 GB

Table 6: Parameters for This Analysis

Parameter	Value
testing_set_percent	0.1
subset_percent	0.1
save_files	FALSE
large_dataset	FALSE
remove_obscure	TRUE
expand_genres	TRUE
dataset name	ml-10m

```
## [1] "Elapsed time for this analysis: 5.3028 mins"
```

Bibliography

Frost, J. 2023. *Room Mean Square Error (RMSE).* https://statisticsbyjim.com/regression/root-mean-square-error-rmse/.

Harper, Konstan, F. M. 2015. "The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (Tiis)." https://doi.org/10.1145/2827872.

Hodson, T. O. 2022. "Hodson, t. O.: Root-Mean-Square Error (RMSE) or Mean Absolute Error (MAE): When to Use Them or Not." *Geossci Model Dev.*

Muriel. J., E., Kavlakoglu. 2023. *What Is Regularization.* https://www.ibm.com/topics/regularization.

Pierson, L. 2024. "What Is Data Modeling – Data Modeling Vs Data Analysis 101." https://www.data-mania.com/blog/what-is-data-modeling-data-modeling-vs-data-analysis-101/.

Wickham, H., M. Centinkaya-Rundel, and G. Grolemund. 2023. "R for Data Science." Sebastopol, CA: O'Reilly Media, Inc.

Williams, E. R. 2024. "How to View and Appreciate Great Movies." https://www.thegreatcourses.com/courses/how-to-view-and-appreciate-great-movies.

Willmott, Matsuura, C.j. 2005. *Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance. Clim. Res.* Vol. 30. https://doi.org/10.3354/cr030079.