HarvardX PH125.9x Data Science Capstone: MovieLens

Xiao Guo

7/28/2021

1 Introduction

1.1 Introduction: Background

The MovieLens dataset is from the MovieLens recommender system, where individuals can rate movies that they have seen and in turn receive recommendations on movies that they may enjoy. The data was collected in 1997 by GroupLens Research from the University of Minnesota. Each observation in the dataset consists of an unique user, an unique movie, the rating given, and additional information on the user or the movie. The movies are rated on a half stars to five stars scale, incrementing in half stars, where five stars is the highest rating. The dataset used for this machine learning task is a fraction of the MovieLens dataset (the 10M version), so it is not computationally expensive for a typical personal computer.

1.2 Introduction: Dataset and Goals

The Movielens dataset is split into the edx dataset for model training, and the validation dataset for model evaluation. The edx dataset (first six observations shown below) has six columns. The columns userId, movieId, timestamp, title, and genres are the predictor variables. The column rating is the outcome.

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

The objective of this machine learning task is to use the MovieLens edx dataset and create a recommender algorithm with the baseline-predictors method. The algorithm parameters will be calculated using the outcome and the predictor variables from the edx set. Then, the trained model will predict the outcome (rating) in the validation set using the predictor variables in the validation set. The goal is to construct a model with a prediction RMSE value that is less than 0.86490.

1.3 Introduction: Overview

First, the data is divided into three sets: train, test, and validation. The train set is used to build each model. The test set is used to tune and evaluate the RMSE of each model. The validation set is used to evaluate the performance of the final model on the hold-out data.

Second, the datasets are processed to clean the existing predictor variables and to construct new useful predictor variables. The final predictor variables are userId, movieId, release year, genre, movie maturity, and user maturity.

Third, the train dataset is graphed to uncover patterns in the data. The variation in each predictor variable is visualized by graphing the distribution of the ratings for each predictor variable. The presence of any imbalance in sample sizes is shown by graphing the prevalence of every value in each predictor variable. The predictive powers of the temporal predictor variables are estimated by graphing the correlations between those predictor variables against the ratings.

Fourth, a series of models are constructed using the train dataset. The first model is just the average rating. In each subsequent model, an additional predictor variable is included to improve the RMSE. The final model of this series includes the average rating value plus the effects of all six predictor variables.

Fifth, a model is constructed where all the predictor variables are regularized. This is to minimize the effect of uncertainties from parameters that are calculated from small samples. The regularized model is tuned (the tuning parameter is lambda) to find the optimal degree of regularization.

Finally, the optimized regularized model is trained with all the data in the edx set and then used to predict the ratings in the validation set.

2 Methods

2.1 Methods: Preprocessing

Preprocessing this dataset has three steps: convert predictor variables and the outcomes to the appropriate data type, extract relevant information from the predictor variables, and construct new predictor variables. A function with all the preprocessing steps is used to ensure the same steps are applied to both the edx and the validation datasets.

The edx dataset (below) has five predictors (userId, movieId, timestamp, title, and genres), and one outcome (rating). UserId and movieId are unique identifiers assigned to each user and movie respectively. Timestamp is when a specific user has rated a specific movie. Title includes the title for a specific movie and the year that it was released. Genres are the genre combinations that a specific movie belongs to.

```
##      userId     movieId      rating   timestamp       title      genres
##   "integer"   "numeric"   "numeric"   "integer" "character" "character"
```

The cleaned data (below) has ten columns, six of which are used as predictor variables for model building, one of which is the outcome variable, and three of which are intermediate variables.

```
##        userId       movieId        rating         title        genres
##      "factor"      "factor"     "numeric"   "character"      "factor"
##      viewDate   releaseDate   view1stDate movieMaturity  userMaturity
##     "numeric"     "numeric"     "numeric"     "numeric"     "numeric"
```

Rating and title do not require any changes since the rating data type is already numeric, and the title data type is already character. The userId and movieId data types need to be changed to factor since they are nominal data. The genres data type needs to be changed to factor since it has categorical data. The timestamp data type needs to be changed to datetime since it contains date and time data (this column is later removed).

The new variable viewDate is the year component from timestamp to match the precision of the other temporal data. The new variable releaseDate is extracted from title, and represents the movie release year. The new variable view1stDate represents the first time a specific user watched any movie. This is the first movie that the user watched in the dataset, not necessarily their first movie ever.

The new predictor variable movieMaturity represents the difference in years between the movie release date and the user watch date. This predictor differentiates between a movie that a user eagerly watched as soon as it is released and a movie that a user waited to watch. It is created by taking the difference between the viewDate and the releaseDate. The new predictor variable userMaturity represents the difference in years between the time of the user's first movie rating and the time of the user's current movie rating. This predictor differentiates the harshness of the user's metric when he first started to rate movies and when he has already rated movies for a while. It is created by taking the difference between the viewDate and the view1stDate.

2.2 Methods: Data Exploration

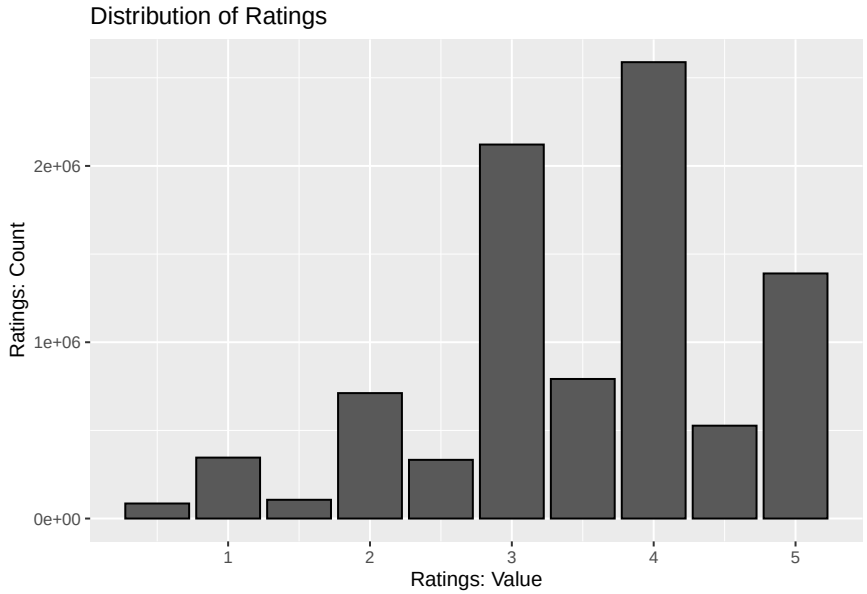
Six predictor variables are hypothesized to influence the rating of a specific movie by a specific user. The predictor variables are movie average (b_i), user average (b_u), genre effect (b_g), movie release year (b_r), movie maturity (b_ti), and user maturity (b_tu). Some preliminary analyses are done on the edx dataset to examine these variables. There are 9000055 total ratings given by a specific user to a specific movie. There are 69878 unique users. There are 10677 unique movies. There are 797 unique combinations of genres. The release date of the movies ranges from 1915 to 2008. The difference in time from movie release date to user watch date ranges from -2 to 93 years. The length in time from when a user rated his first movie and when he rated the current movie ranges from 0 to 12 years.

2.3 Methods: Data Visualization

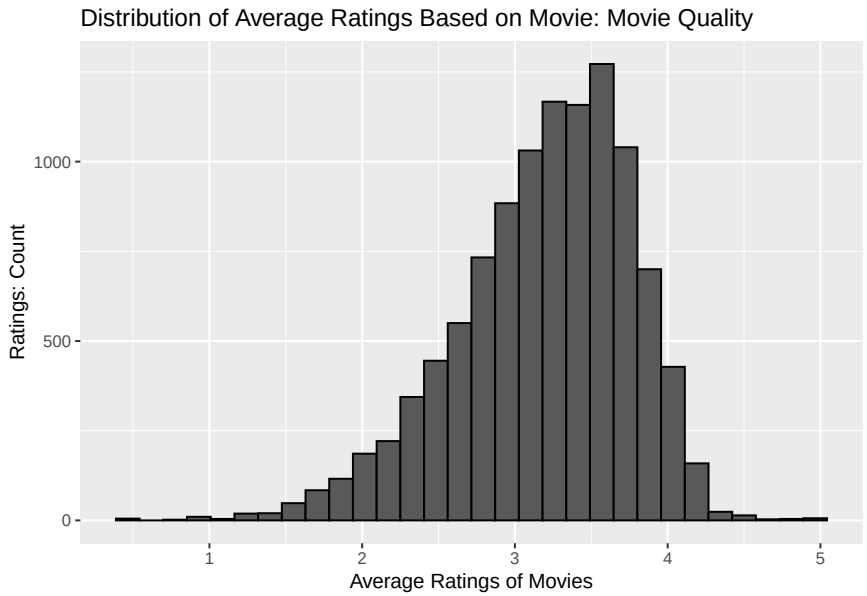
The observations from the preprocessed edx dataset are plotted to reveal patterns in the data. The data are visualized in three ways: examine the distribution of the predictor variables, examine the prevalence of every value in each predictor variable, and examine the correlation between the temporal predictor variables and the outcome rating.

2.3.1 Distribution of Ratings Within Each Predictor Variable The distribution of ratings within each predictor variable is visualized with histograms. The more spread out the histogram, the more the predictor variable influences the rating.

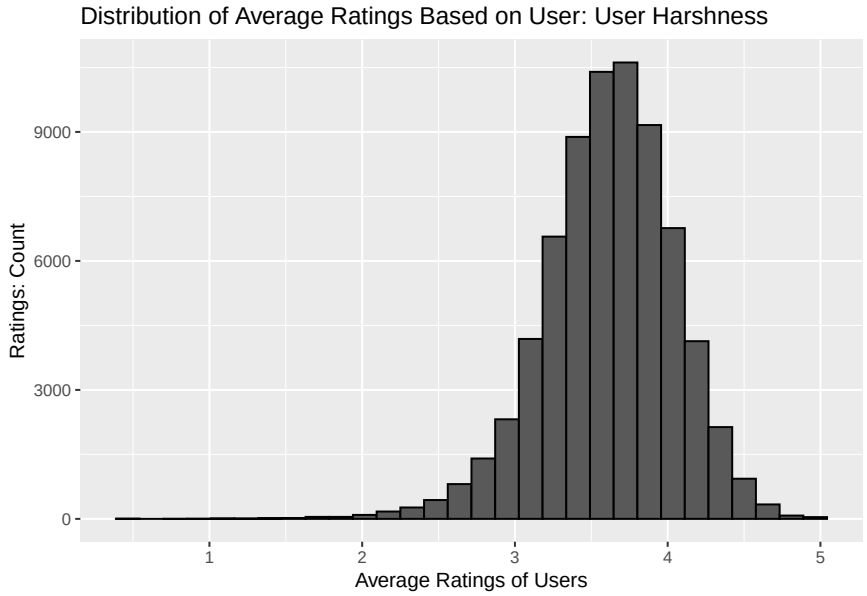
2.3.10 Distribution of Rating Values This plot shows the number of observations for each rating value. It shows there are more whole star ratings than half star ratings.



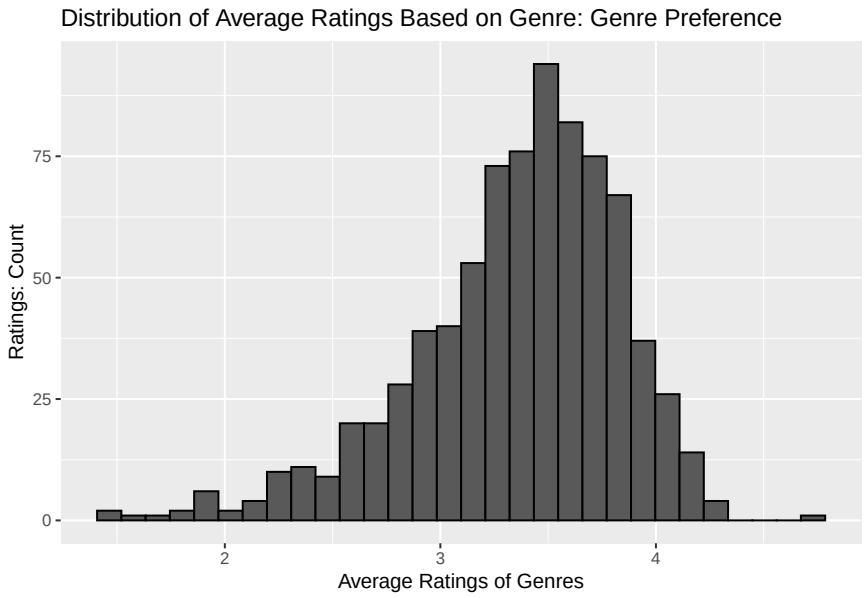
2.3.11 Distribution of Average Ratings Based on Movie This plot shows how the average rating of each movies is distributed. The average rating of the movie represents the quality of the movie, where higher quality movies tend to have higher averages. The plot is quite spread out spanning between 0 and 5 stars. There are very few values at the extremes, and the plot is approximately left skewed normal with the peak between 3 and 4 stars.



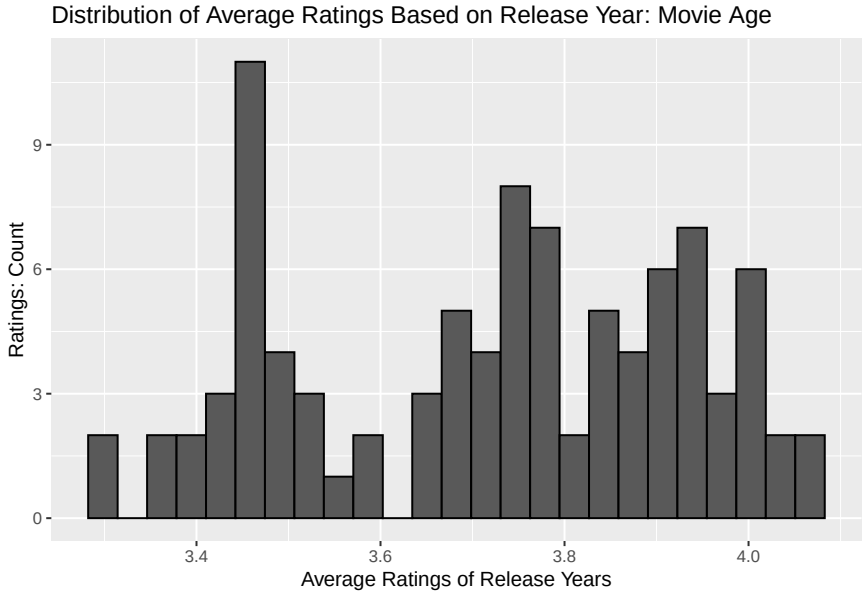
2.3.12 Distribution of Average Ratings Based on User This plot shows how the average rating from each user is distributed. The average rating from the user represents the harshness of the user, where a harsher user tends to give lower averages. The plot is quite spread out spanning between 0 and 5 stars. There are very few values below 2 stars, and the plot is approximately normal with the peak between 3 and 4 stars.



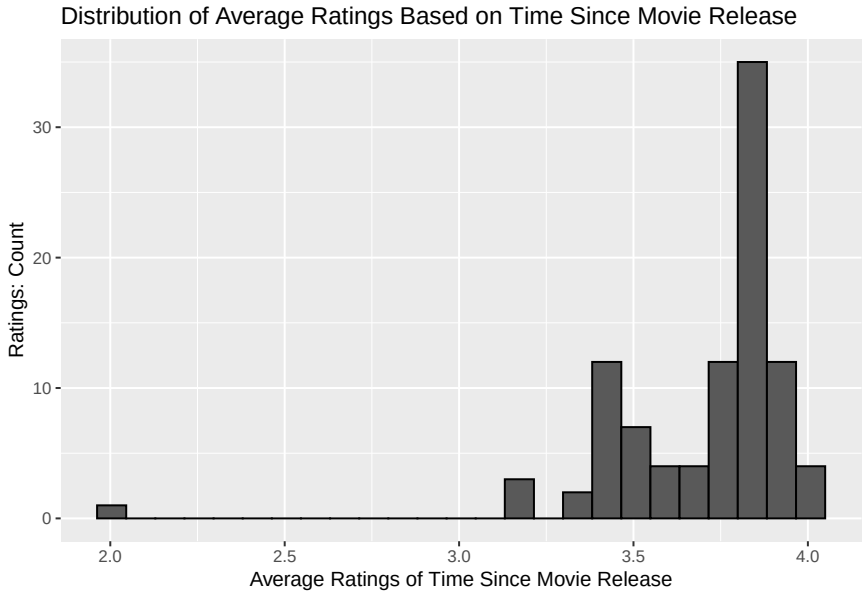
2.3.13 Distribution of Average Ratings Based on Genre This plot shows how the average rating of each genre combination is distributed. The average rating of the genre represents the enjoyability of the movies in a specific genre combination for the average viewer. The plot is quite spread out spanning between 1 and 5 stars. There are very few values at the extremes, and the plot is approximately left skewed normal with the peak between 3 and 4 stars.



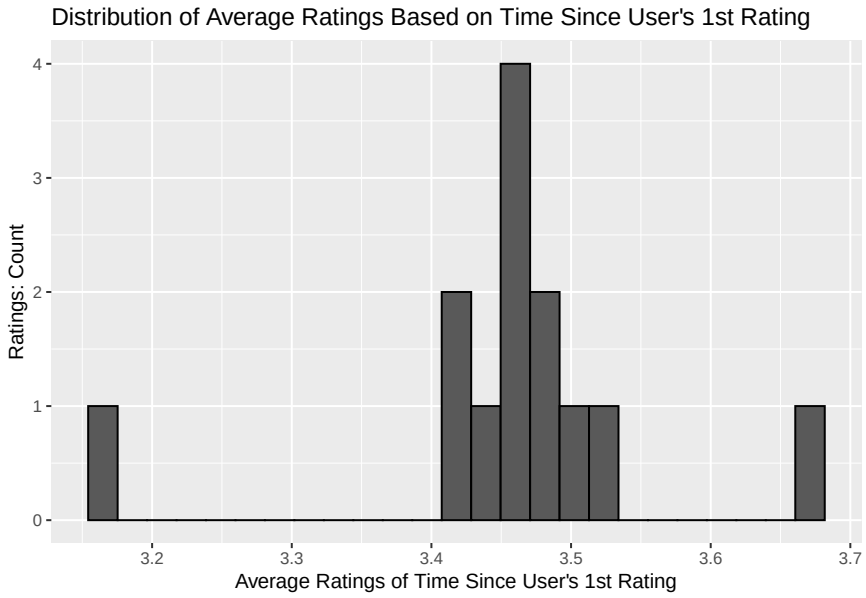
2.3.14 Distribution of Average Ratings Based on Release Year This plot shows how the ratings averaged for each release year is distributed. The average rating of the release year represents the effect of the movie's age. The plot is closely clustered, spanning approximately between 3.2 and 4.2 stars, with little variance. The plot has multiple peaks and is not normally distributed.



2.3.15 Distribution of Average Ratings Based on Time Since Movie Release This plot shows the distribution of the average rating grouped by the time between the movie release date and the review date. The average rating of the time from the movie release date to the review date differentiates between watching a recently released movie and watching an old movie. The plot is closely clustered, spanning approximately between 3.2 and 4.2 stars, and has an outlier at 2 stars. The plot has multiple peaks and is not normally distributed.

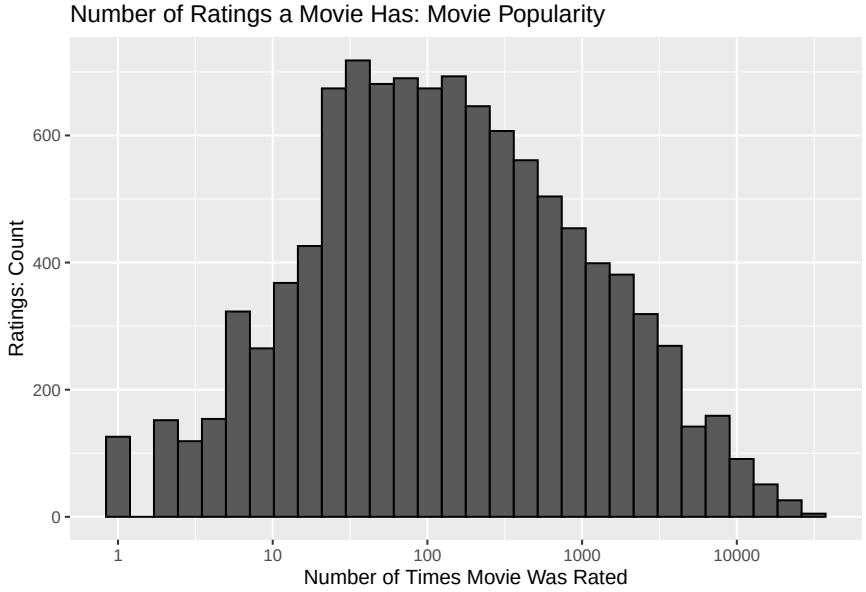


2.3.16 Distribution of Average Ratings Based on Time Since User's First Rating This plot shows the distribution of the average rating grouped by the time between a user's first rating date and the current rating date. The average rating of the the time from user's first rating date to user's current rating date differentiates a user that just started to rate movies and a user that have rated movies for many years. The plot is closely clustered, spanning approximately between 3.4 and 3.6 stars, and has outliers at 3.2 and 3.7 stars. The plot has multiple peaks and is not normally distributed.

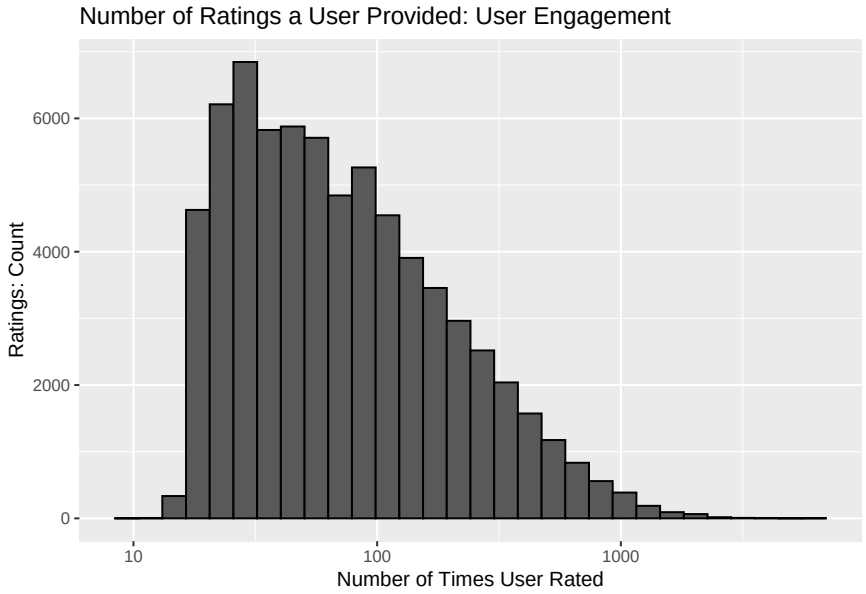


2.3.2 Prevalence of Each Value Within a Predictor Variable The prevalence of each value within a predictor variable is visualized with histograms and barplots. The prevalence of the values for a predictor is imbalanced when some values have a large number of observations and some values have a small number of observations. Data with imbalance prevalence needs to be weighted (using regularization) to reduce the impact of small sample uncertainties caused by small samples. *Notice, the x axis for the first three plots have a log scale, and the y axis for the last three plots have a log scale.

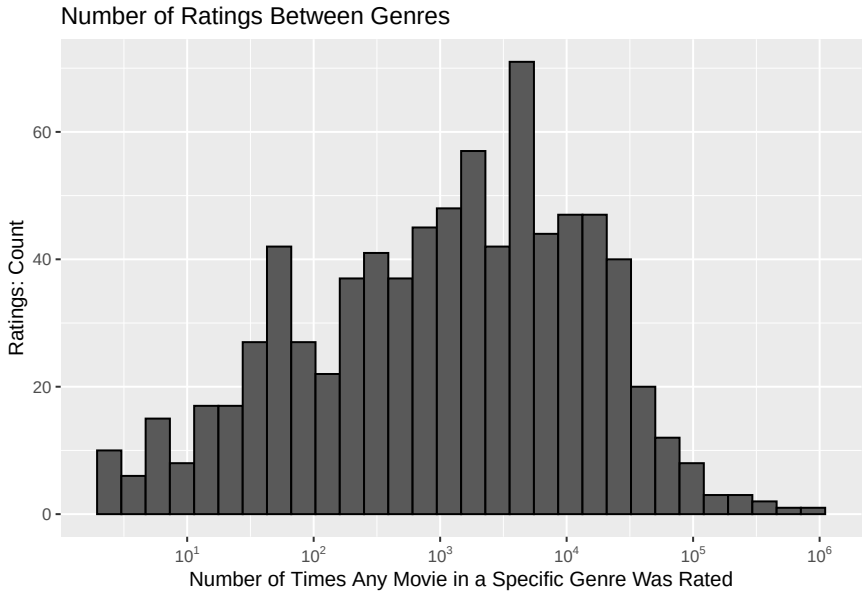
2.3.21 Prevalence of the Number of Ratings a Movie Has The number of ratings a movie has represents movie popularity. The values are imbalanced, where some unpopular movies have a single rating and some popular movies have more than ten thousand ratings. Most movies have been rated between ten and a thousand times.



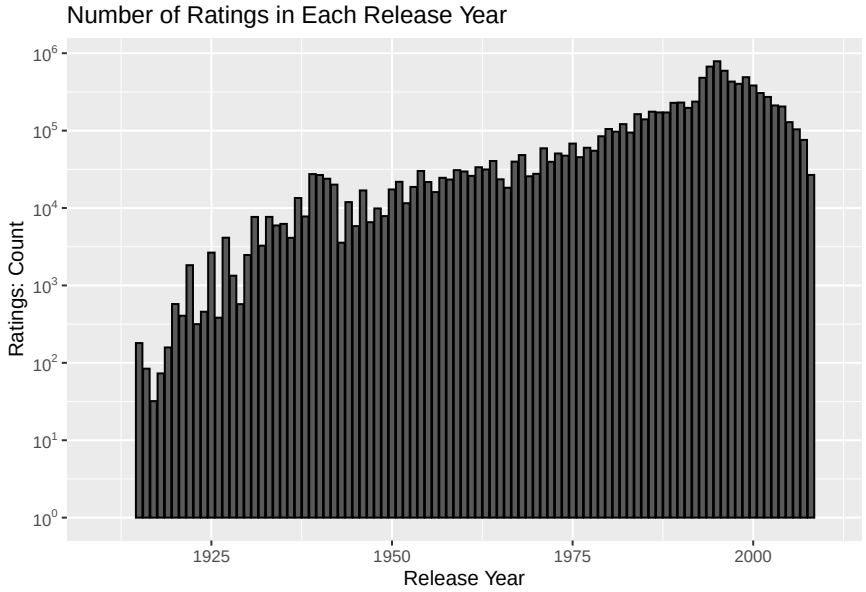
2.3.22 Prevalence of the Number of Ratings a User Provided The number of ratings a user provided represents user engagement. The values are imbalanced, where some users rated around ten movies and some rated thousands. Most people rated between ten and a hundred movies.



2.3.23 Prevalence of the Number of Ratings Between Genres The number of ratings a genre combination has represents genre popularity and the number of movies in a genre combination. The values are imbalanced, where some genres have less than ten ratings and some have millions. Most genres have between a hundred and ten thousand ratings.

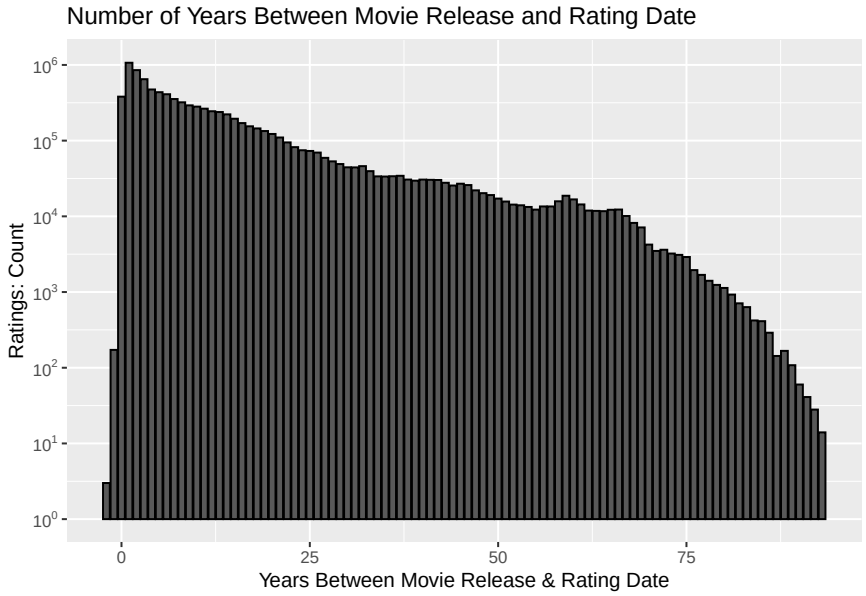


2.3.24 Prevalence of the Number of Ratings in Each Release Year The number of ratings in a specific year is function of the popularity and the number of movies released in that specific year. The plot is heavily skewed to the left (long left tail). This suggests that in general people watch recent movies more than old movies. However, the highest values are in the 1990s rather than the 2000s. This may be because movies from the 1990s have been out for longer than movies from the 2000s, allowing more people to watch them. The values are imbalanced, where some years have less than a hundred ratings and some have near a million.

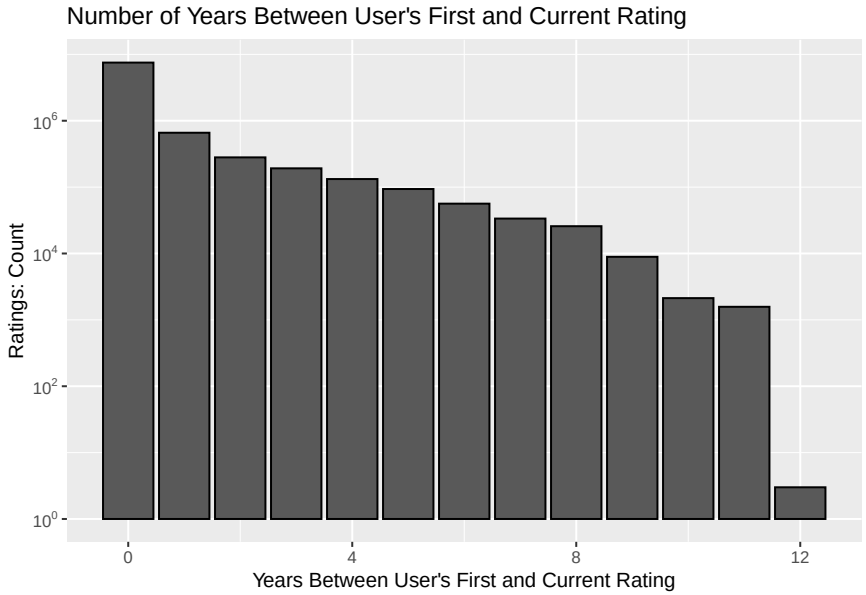


2.3.25 Prevalence of the Number of Years Between Movie Release and Rating Date This plot shows the preference for users to watch recently released movies or old movies. The negative values are likely caused by people that watched movies before their official release date. The number of observations is highest when the time between movie release date and watch date is short. The number of observations

decline almost exponentially as the length of time increases. This shows that people tend to watch movies that are recently released. The values are imbalanced, where some time periods have less than a hundred ratings and some have near a million.

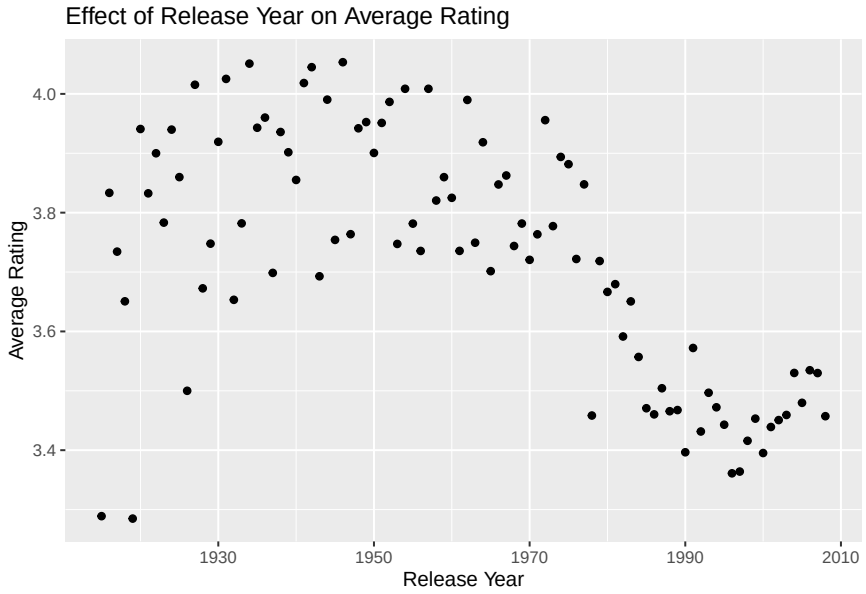


2.3.26 Prevalence of the Number of Years Between User's First and Current Rating Date This plot shows the length of time a user has been rating movies. Almost all the ratings occurred in the same year as the first movie its user rated. This suggests that most of the users have just begun to rate movies. The values are imbalanced, where some time periods have less than ten ratings and some have millions.

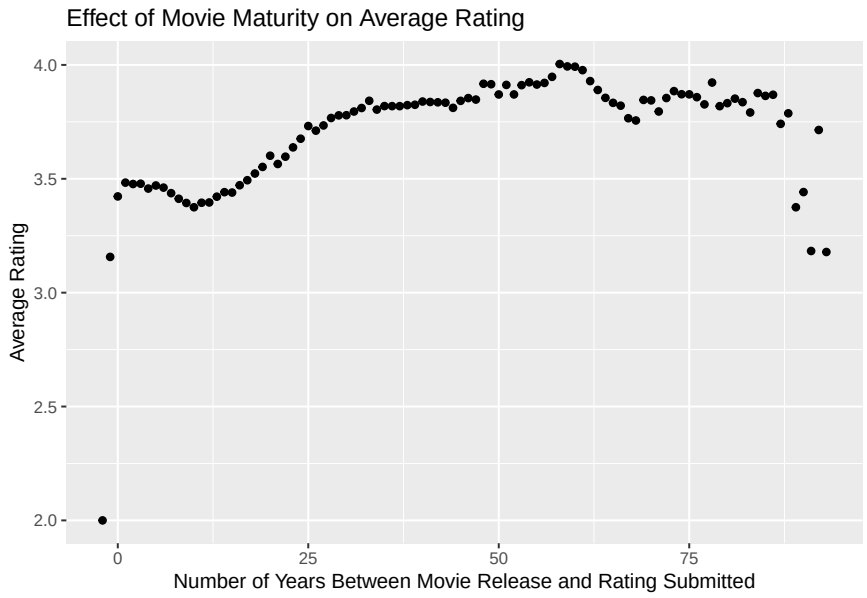


2.3.3 Correlation Between Temporal Predictor Variables and Outcome Rating The correlations between the temporal predictor variables and the outcome rating are revealed with scatter plots. Since different time periods have different sample sizes, the size of the confidence intervals are different for different time periods.

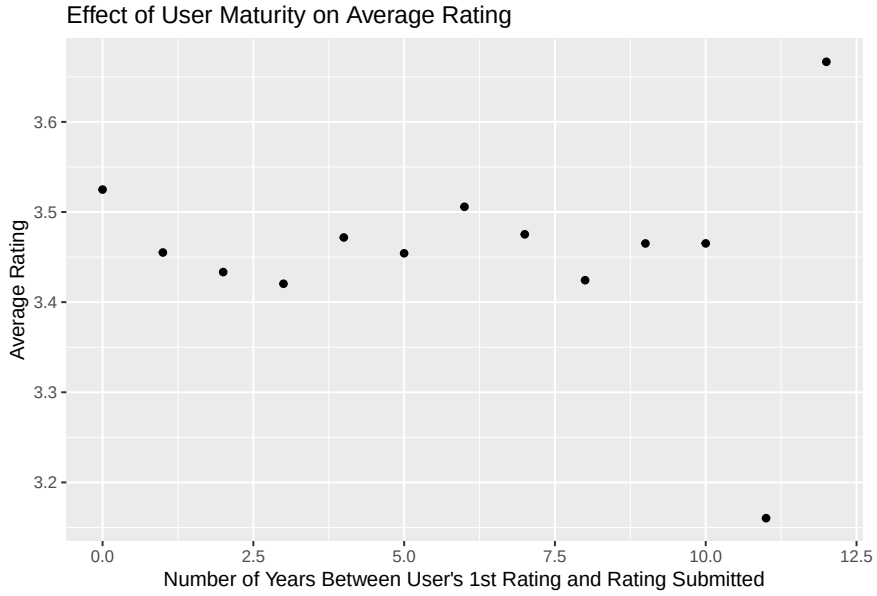
2.3.34 Effect of Release Year on Average Rating The variability in ratings decreases as the release year moves from old to recent. This suggests that the recent films are rated much more often than older films. Moving from old to recent dates, the average movie ratings decrease to a minimum in the mid 1990s then increases after that. The valley in the graph corresponds to the time period with the most ratings. This suggests that the values in the periods before and after 1990s may have high uncertainty due to low sample size. Alternatively, this may suggest that people tend to enjoy a movie if they watch it right when it is released, and people tend to look for old movies if it is a good movie.



2.3.35 Effect of Movie Maturity on Average Rating The ratings decrease slightly as the time between movie release date and the view date moves from zero to ten years. This suggests that people tend to enjoy movies if they watch it as soon as it is released. Since this period has the most ratings, the data has higher certainty. The ratings increase steadily as the time between movie release date and the view date moves from ten to sixty years. This suggests that the movies need to be high quality for a person to make the effort to search for them. Since this period has fewer ratings, the data has lower certainty. The ratings decrease slightly and increase in variability as the time between movie release date and the view date moves beyond eighty years. This suggests that those movies are rated very few times, and the values have high uncertainty.



2.3.36 Effect of User Maturity on Average Rating The ratings decrease slightly as the time between the view date of the user's first movie and the current movie moves from zero to three years. This suggests that users tend to get harsher over time. Since this period has the most ratings, the data in this period has higher certainty. The ratings beyond three years increases then decreases again. This may suggest another pattern in the user's harshness, or a lack of data. Since this period has fewer ratings, the data in this period has lower certainty.



2.3.4 Overall Insights The distribution plots showed that the predictor variables movie, user, and genre will have a large impacts on rating, while the predictor variables release year, movie maturity, and user maturity will have smaller impacts on rating. The prevalence plots showed that all the predictor variables have imbalance in their values. This means that all of these predictor variables will benefit from weighed (regularized) values in the algorithm. Lastly, if the data with high uncertainty are ignored, a movie that is watched immediately upon release will be rated higher, and a user will become harsher over time.

2.4 Methods: Create Train and Test Datasets

The edx dataset is split into a training set and a test set, so the models can be evaluated and tuned without touching the validation set. The training set is used to calculate the predictor effects, and the test set is used to evaluate the RMSE for each model. To balance the data required for training and evaluating the models, 80% of the data goes to the training set and 20% of the data goes to the test set.

2.5 Methods: RMSE As Model Evaluation Method

The models are evaluated by measuring the size of their root mean square error (RMSE). The RMSE is calculated by first finding the error as the distance between the true rating and the predicted rating. Then the errors of all the observations are squared and averaged. After that, the square root of the result is calculated. The smaller the RMSE, the more accurate the model. Conceptually, a RMSE value of one means that on average the predicted rating is one star off from the true value.

2.6 Methods: Baseline-Predictors Model Overview

The baseline-predictors model divides the ratings into several components. The rating for each observation is broken down in the following way:

$Y_{movie,user} = \mu + b_{movie} + b_{user} + b_{genre} + b_{year} + b_{movieM} + b_{userM} + \epsilon$
$Y_{movie,user}$ represents the rating from a specific user for a specific movie.
μ represents the average of all the ratings in the training set.
The b parameters represent the effect that a predictor variable has on the rating.
b_{movie}, (b_i) or movie bias is the quality of the movie.
b_{user}, (b_u) or user bias is the harshness of the user.
b_{genre}, (b_g) or genre bias is the effect of belong to a specific genre combination.
b_{year}, (b_r) or release year bias is the effect of the release year (or age) of the movie.
b_{movieM}, (b_ti) or movie maturity bias differentiates between a user that eagerly sees a movie as soon as it was released and a user that waits a while after the movie was released.
b_{userM}, (b_tu) or user maturity bias differentiates between a user that just started to rate movies and a user that may have become more critical or more lenient over time.
ϵ represents independent random errors.

Ideally, each of the predictor variables b is calculated by regression. However, this is not feasible for the typical personal computer, because every value of each predictor variable requires an individual b coefficient. For example, each movie has a separate b_{movie} coefficient, each user has a separate b_{user} coefficient, each genre combination has a separate b_{genre} coefficient, etc.

Therefore, the least square method is used. In this method, the b parameters are approximated by taking the difference between the true rating value, and the rating value predicted by the model that does not include the specific b parameter. For example, the value of b_{movie} for the ith movie is the difference between the average rating of the ith movie (true value) and the variable μ (predicted value).

This least square model minimizes the following sum of squares equation:

$\frac{1}{N}\sum(Y_{movie,user} - \mu - b_{movie} - b_{user} - b_{genre} - b_{year} - b_{movieM} - b_{userM} + \epsilon)^2$

The prevalence of the values in each predictor graph (section 2.3.2) showed that the sample size for different values are not balanced. Therefore, many b coefficients will be calculated from small sample sizes. Any b coefficient that is calculated from only a few observations will introduce a large uncertainty, especially if the b coefficient has a large magnitude. For example, the average rating for a movie (the predictor variable) that

has been rated only a few times (small sample) may not reflect the true quality of that movie. As another example, the average rating from a user (predictor variable) that only rated a few movies (small sample) may not reflect the true harshness of that user.

Regularization is a technique that reduces the effect of b parameters produced from small samples by shrinking their values toward zero with a penalty term. For example, if a movie has only a few ratings but its b_{movie} value has a large magnitude (large positive or negative value), its b_{movie} value will be shrunk toward zero. If the sample size is large, then the penalty term becomes insignificant. Thus regularization only penalizes small sample sizes. The penalty term has the tuning parameter lambda, which controls the degree of penalization.

This regularized model minimizes the following sum of squares equation together with a penalty term:

$$\frac{1}{N}\sum(Y_{movie,user} - \mu - b_{movie} - b_{user} - b_{genre} - b_{year} - b_{movieM} - b_{userM} + \epsilon)^2 + \lambda(\sum b_{movie}^2 + \sum b_{user}^2 + \sum b_{genre}^2 + \sum b_{year}^2 + \sum b_{movieM}^2 + \sum b_{userM}^2)$$

2.7 Methods: Modeling Procedure

The initial model is just the average of all the ratings.

Then the effects from the six predictors (b_{movie}, b_{user}, b_{genre}, b_{year}, b_{movieM}, and b_{userM}) are incorporated into the model one at a time.
The procedure for each subsequent models is the following:
1. Calculate the new predictor coefficients on the training set.
2. Predict the ratings on the test set.
3. Record the RMSE on a summary table.
4. Plot the distribution of the new predictor.

Next, the entire model is regularized to penalize the effect of parameters calculated from small samples.
The procedure to construct the optimized regularized model is the following:
1. Choose a series of lambda values, where every lambda results in one regularized model.
2. With each lambda, calculate the coefficient of each predictor (b_{movie}, b_{user}, b_{genre}, b_{year}, b_{movieM}, and b_{userM}) on the training set.
3. Predict the ratings on the test set.
4. Calculate the RMSE.
5. Choose the lambda with the best RMSE and record its RMSE on a summary table.

3 Results

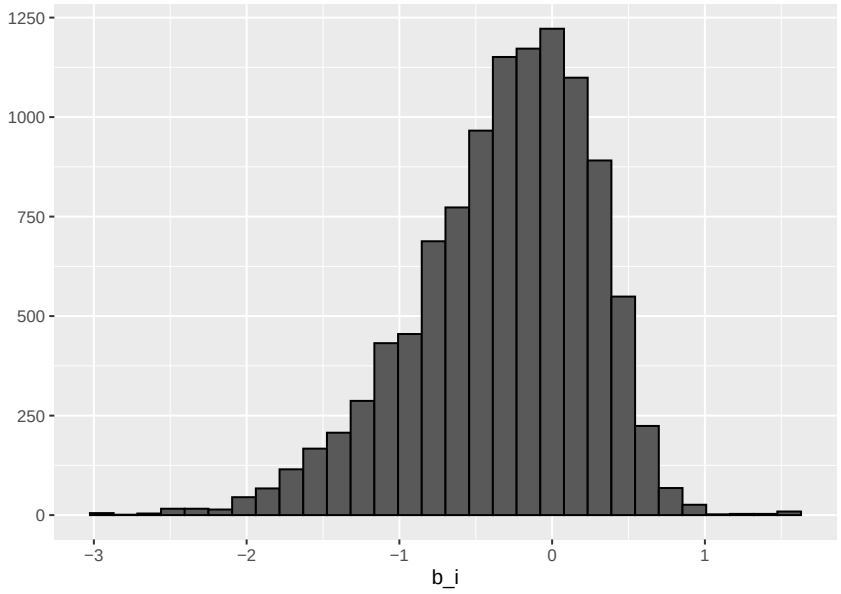
3.1 Results: Non-Regularized Models

3.1.1 Average the Ratings The first model is the mean of all the ratings in the dataset. The RMSE is 1.05990. It serves as the baseline value for this dataset.

method	RMSE
Just the average	1.059904

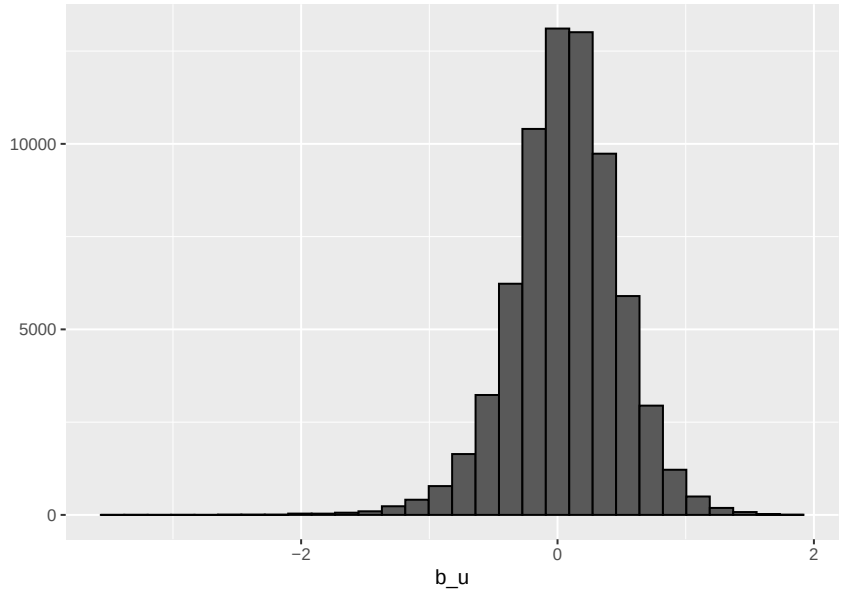
3.1.2 Include Movie Effects b_i The movie effects predictor, b_{movie} (b_i) accounts for the quality of movies. Higher quality movies generally have higher ratings. The distribution of this predictor is left skewed normal with a range of -3.01 to 1.48 stars, median of -0.24 stars, and mean of -0.32 stars. The RMSE is 0.94374, an improvement of 0.11616.

method	RMSE
Just the average	1.0599043
Movie Effect Model	0.9437429



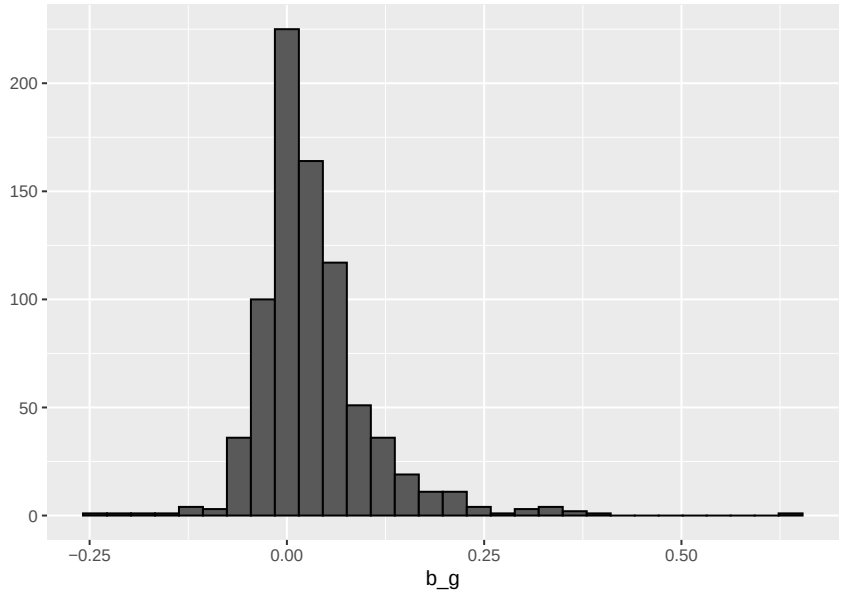
3.1.3 Include User Effects b_u The user effects predictor, b_{user} (b_u) accounts for harshness of users. Harsher users generally give lower ratings. The distribution of this predictor is approximately normal with a range of -3.42 to 1.88 stars, median of 0.72 stars, and mean of 0.06 stars. The RMSE is 0.86593, an improvement of 0.07781.

method	RMSE
Just the average	1.0599043
Movie Effect Model	0.9437429
Movie + User Effects Model	0.8659319



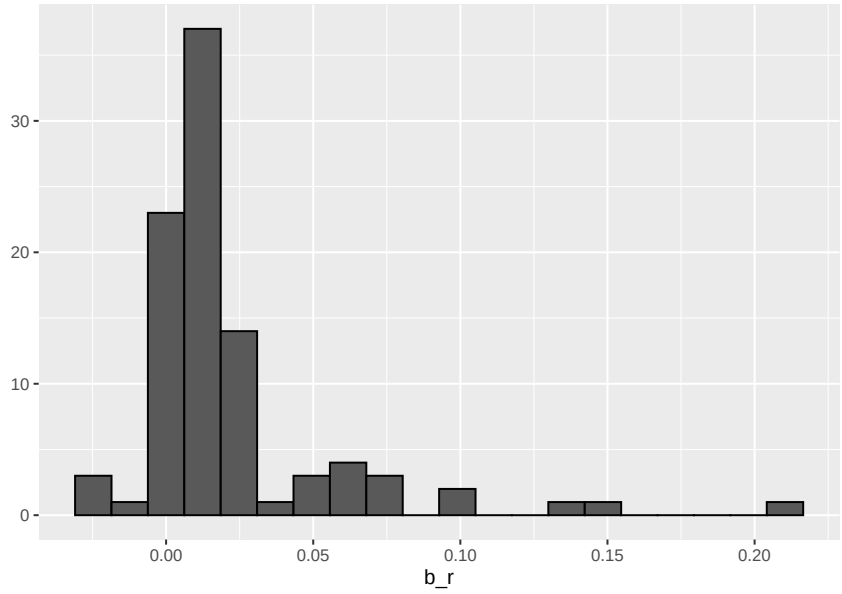
3.1.4 Include Genre Effects b_g The genre effects predictor, b_{genre} (b_g) accounts for any rating bias resulting from different genre combinations. The distribution of this predictor is right skewed normal with a range of -0.25 to 0.63 stars, median of 0.02 stars, and mean of 0.03 stars. The RMSE is 0.86559, an improvement of 3.4×10^{-4}.

method	RMSE
Just the average	1.0599043
Movie Effect Model	0.9437429
Movie + User Effects Model	0.8659319
Movie + User + Genre Effects Model	0.8655941



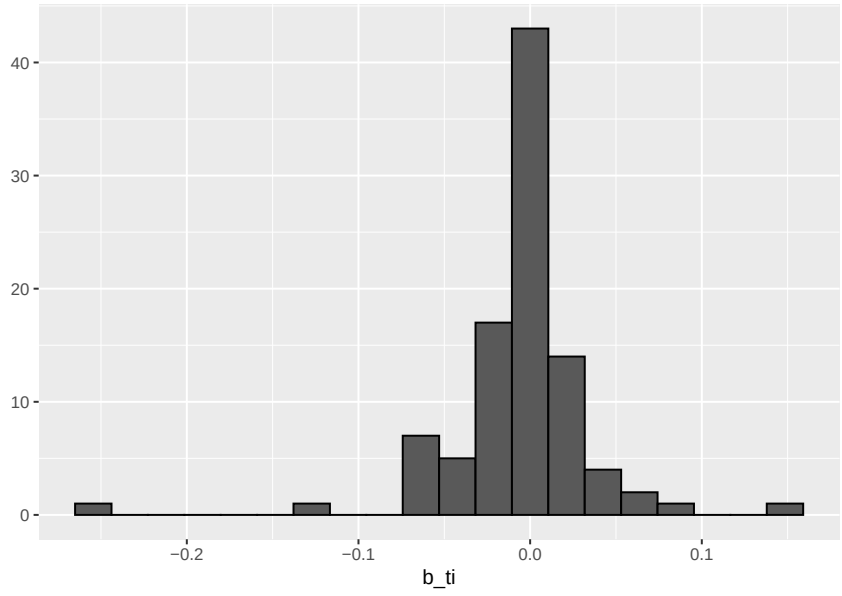
3.1.5 Include Movie Release Time Effects b_r The release year effects predictor, b_{year} (b_r) accounts for any rating bias resulting from the movie's release date. The distribution of this predictor is right skewed with a range of -0.03 to 0.21 stars, median of 0.01 stars, and mean of 0.02 stars. The RMSE is 0.86542, an improvement of 1.7×10^{-4}.

method	RMSE
Just the average	1.0599043
Movie Effect Model	0.9437429
Movie + User Effects Model	0.8659319
Movie + User + Genre Effects Model	0.8655941
Movie + User + Genre + Release Effects Model	0.8654189



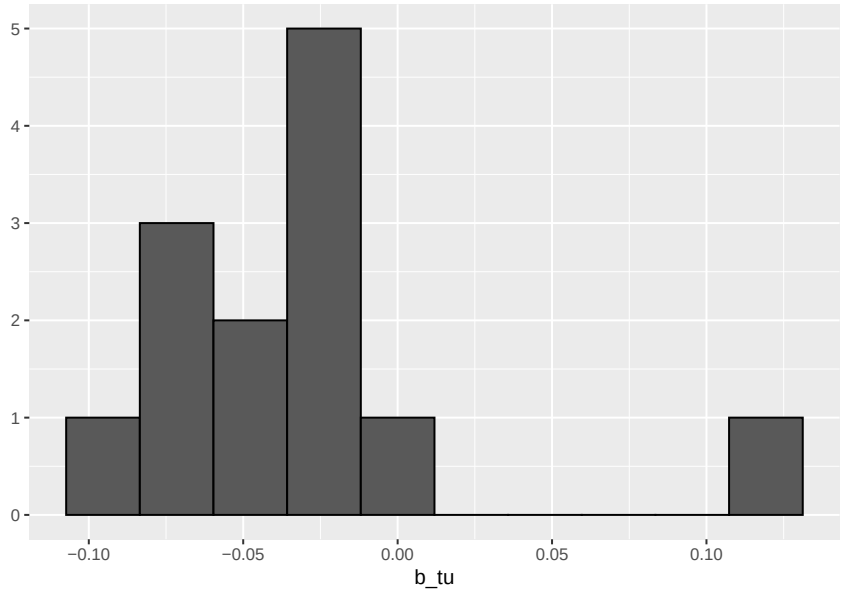
3.1.6 Include Movie Review Time Effects b_ti The movie review time effects predictor, b_{movieM} (b_ti) accounts for the time between the movie release date and the user view date. A user that viewed a movie immediately when it was released rather than waiting for that movie to be more readily available may have a prior interest in that movie. The distribution of this predictor has a range of -0.26 to 0.15 stars, median of 0.00 stars, and mean of -0.01 stars. The RMSE is 0.86510, an improvement of 3.2×10^{-4}.

method	RMSE
Just the average	1.0599043
Movie Effect Model	0.9437429
Movie + User Effects Model	0.8659319
Movie + User + Genre Effects Model	0.8655941
Movie + User + Genre + Release Effects Model	0.8654189
Movie + User + Genre + Release + MovieM Effects Model	0.8650974



3.1.7 Include User Review Time Effects b_tu The user review time effects predictor, b_{userM} (b_tu) accounts for the time between the view date of the user's first movie and the view date of the current movie. The harshness of a user may change as more movies are watched over time. The distribution of this predictor has a range of -0.10 to 0.11 stars, median of -0.03 stars, and mean of -0.03 stars. The RMSE is 0.86502, an improvement of 8×10^{-5}.

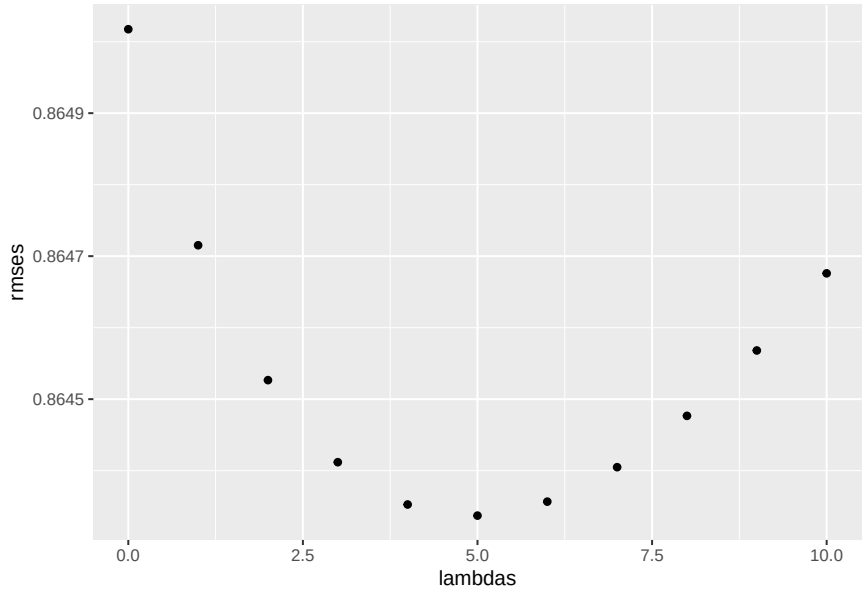
method	RMSE
Just the average	1.0599043
Movie Effect Model	0.9437429
Movie + User Effects Model	0.8659319
Movie + User + Genre Effects Model	0.8655941
Movie + User + Genre + Release Effects Model	0.8654189
Movie + User + Genre + Release + MovieM Effects Model	0.8650974
Movie + User + Genre + Release + MovieM + userM Effects Model	0.8650174



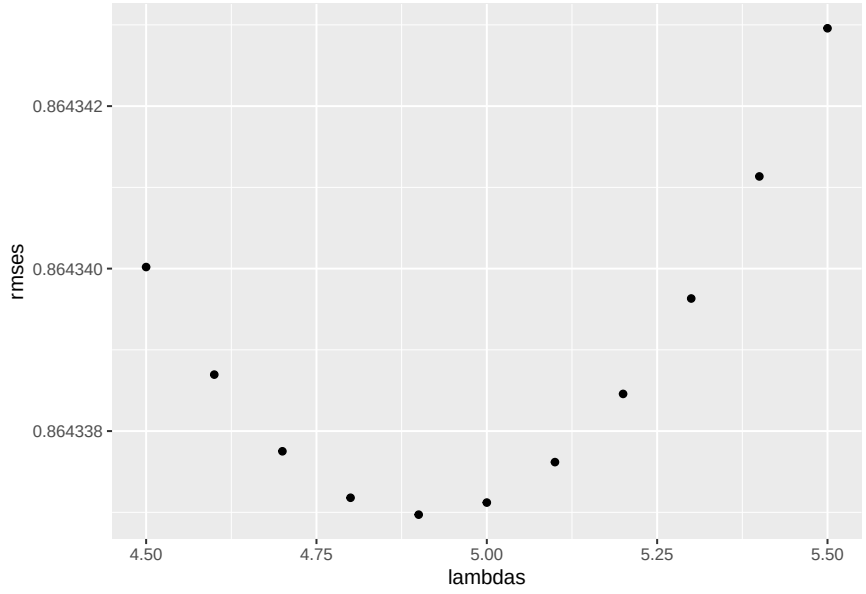
3.1.8 Non-Regularized Models Insights The first two predictors captured most of the variation in the data, and provided the greatest RMSE improvements. This is shown in the large range of their b coefficients. The following four predictors had very small variation, and only offered small RMSE improvements. This is shown in the small range of their b coefficients. The distribution of each of the predictor coefficients mirror the corresponding plot for the distribution of the rating averages for each predictor variables (section 2.3.1).

3.2 Results: Regularized Models

3.2.1 Optimize Tuning Parameter Lambda The regularized model is optimized by calculating the RMSE from different lambda values. Lambda values from 0 to 10 in intervals of 1 are first tested. The lambda value of 5 produced the lowest RMSE.



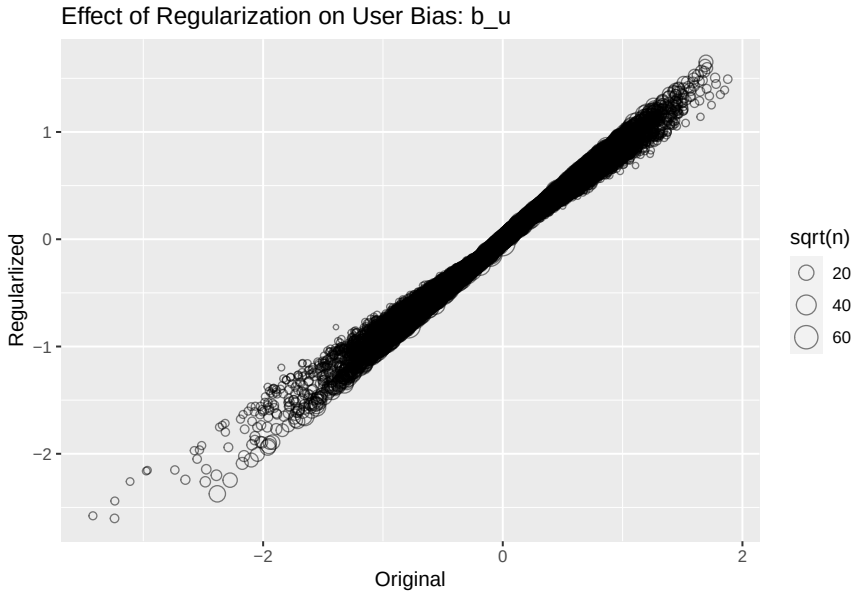
Afterwards, lambda values from 4.5 to 5.5 in intervals of 0.1 are tested. The lambda value of 4.9 produced the lowest RMSE.

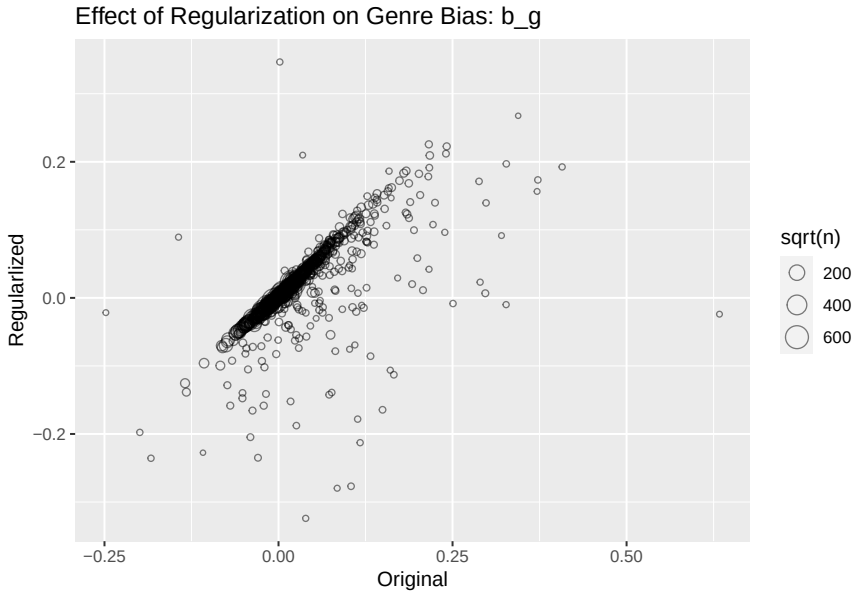


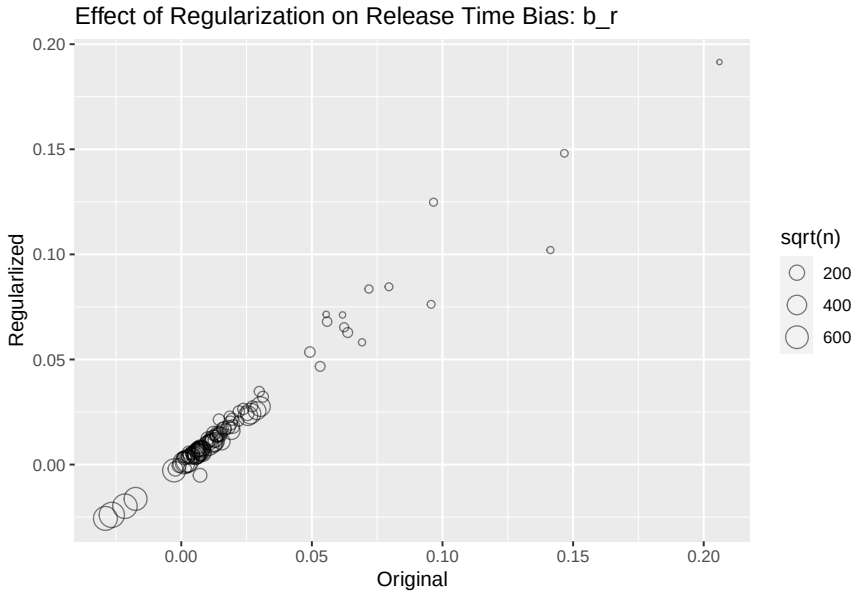
3.2.2 Visualize Effect of Regularization The effect of regularization is visualized for each predictor by plotting the value calculated with the least square algorithm (x axis) against the value calculated with the regularized algorithm (y axis).

The larger circles represent b coefficients from larger sample sizes, and the smaller circles represent b coefficients from smaller sample sizes. For example, a movie that was rated many times has a large circle, while a movie that was rated only once has a small circle. As another example, a user that rated many movies has a large circle, while a user that only rated one movie has a small circle.

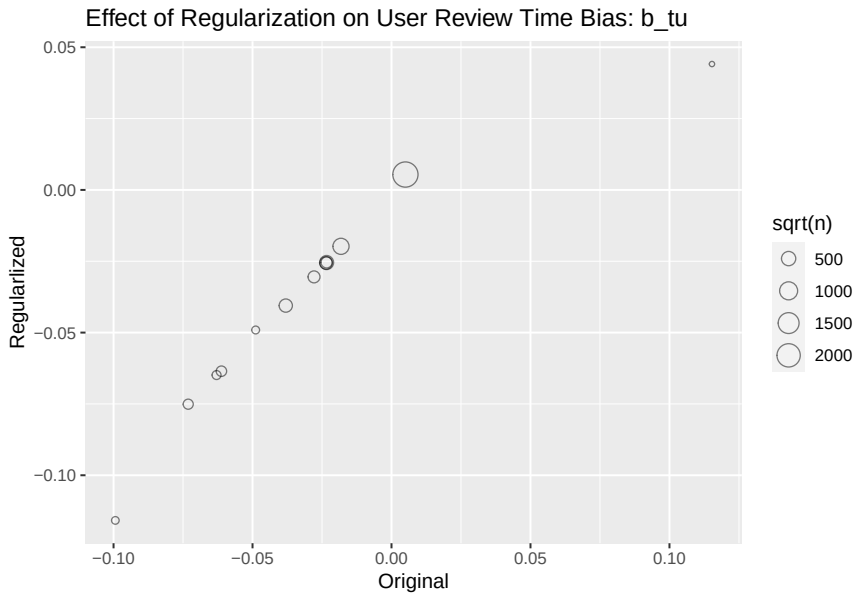
If most of the points fall on a line, then regularization had a small impact on the specific predictor variable. If a lot of the points are spread out away from the trend line, then regularization had a large impact on the specific predictor variable.


Effect of Regularization on User Bias: b_u
Regularlized
Original
sqrt(n)
20
40
60
1
0
−1
−2
−2
0
2


Effect of Regularization on Genre Bias: b_g
Regularlized
Original
sqrt(n)
200
400
600
0.2
0.0
−0.2
−0.25
0.00
0.25
0.50


Effect of Regularization on Release Time Bias: b_r
Regularlized
Original
sqrt(n)
200
400
600
0.20
0.15
0.10
0.05
0.00


Effect of Regularization on Movie Review Time Bias: b_ti
Regularlized
Original
sqrt(n)
250
500
750
0.1
0.0
−0.1
−0.2



The general trend of all the graphs is a linear line with a positive slope of 1. A point that falls directly on this line means that it has the same value in both the least square non-regularized and the regularized model. The larger circles are only slightly affected by regularization, thus they gather close to this trend line. The smaller circles are greatly impacted by regularization, thus they are spread further away from this trend line. The min, max, median, and mean values of the coefficients of each predictor shrink toward 0 when comparing the regularized model to the non-regularized least square model. Viewing the spread of the points, these plots show that regularization has a significant impact on movie, and genre effects, and less impact on user, release date, movie maturity, and user maturity effects.

3.2.3 Regularized Model RMSE

method	RMSE
Just the average	1.0599043
Movie Effect Model	0.9437429
Movie + User Effects Model	0.8659319
Movie + User + Genre Effects Model	0.8655941
Movie + User + Genre + Release Effects Model	0.8654189
Movie + User + Genre + Release + MovieM Effects Model	0.8650974
Movie + User + Genre + Release + MovieM + userM Effects Model	0.8650174
Regularized Model	0.8643370

The RMSE of the tuned regularized model is 0.86434, an improvement of 6.8×10^{-4}. This is the RMSE of the final model predicting the ratings on the test set. However, this is not the true RMSE of the final model because the test set was also used to optimize the lambda parameters. To find the true RMSE of this model, predictions needs to be made on the validation set, which is the hold-out dataset.

3.3 Results: Predicting the Validation Set

To predict the ratings in the validation set, the tuned regularized model is used. The data in the validation set is processed to have the same format as the edx set. The entire edx dataset (not just the training set) and the tuned lambda parameter are used to calculate the coefficients for each predictor variable (b_i, b_u,

b_g, b_r, b_ti, and b_tu). Then, a rating is predicted for each observation in the validation set. Each predicted rating is compared to the true rating to calculate the RMSE.

The final RMSE on the validation set using the regularized model is 0.8638377.

This value is even lower than the one produced from training the model. This is likely because there is less variation in the validation set than the edx set since all the users and movies in the validation set exist in the edx set, but the inverse is not true.

4 Conclusion

4.1 Conclusion: Summary

The MovieLens dataset is a list of ratings that specific users gave to specific movies. There are six predictors, and ten possible rating values. To predict the ratings given to a specific movie by a specific user, baseline-predictors models are used. The first model is just the average of all the ratings. Each additional model incorporates an additional predictor to explain part of the residual. All the predictors in the model are then regularized to minimize the effect of any predictor coefficients that are calculated from small samples. During training, the initial model has a RMSE of 1.05990, the model with all the predictors has a RMSE of 0.86502, and the regularized model has a RMSE of 0.86434. The largest reduction in RMSE are a result of the introduction of the predictor variables movie effects, user effects, and genre effects, as well as regularization. The improvement in RMSE from the initial model to the final model is 0.19556, almost a fifth of a star. When the final model is trained on the entire edx dataset to predict the ratings in the validation set, the RMSE is 0.86384. Therefore, the RMSE goal of 0.86490 or lower has been achieved.

4.2 Conclusion: Limitations

The first limitation of this study is that the predictive power of this model is limited to only the users and movies in this dataset. However, a similar model can be applied to similar datasets with different movie and user combinations. The second limitation is the model looks at the values in each predictor variable in isolation. For example, the movie predictor examines the quality of each movie for all users in isolation. It does not cluster movies that are rated similarly together. It does not investigate if certain movies are rated similarly. As another example, the user predictor only considers the harshness of the user for all movies. It does not question if a user has a preference for a certain type of movie over another. It does not investigate if a certain group of users prefers a specific type of movies and dislikes another type, while another group of users may display the opposite preference. The third limitation is each predictor variable is calculated using the least square method. Although regression is more computationally expensive, it is possible that regression may be able to produce more optimal values. This will involve building neighborhood models, as an extension of the baseline-predictors models.

4.3 Conclusion: Future Works

Regression methods can be used if more powerful computers are available. Other algorithm such as matrix factorization may find patterns in each predictor. For example, movies may be clustered into categories such as blockbusters, cult following, or academy award winning. As another example, users may be clustered to groups that choose movies based on popularity, availability, critically acclaimed, genre, or presence of specific actor or director. A neighborhood model that clusters similar observations can further improve the performance of the machine learning task.

5 Citation

Harper, F. M., & Konstan, J. A. (2016). The MovieLens Datasets. ACM Transactions on Interactive Intelligent Systems, 5(4), 1-19. doi:10.1145/2827872

Irizarry, R. A. (2020). Introduction to data science data analysis and prediction algorithms with R. Boca Raton: CRC Press.

Koren, Y. (2009, August). The BellKor Solution to the Netflix Grand Prize. Retrieved from https://netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf

Winning the Netflix Prize: A Summary. (n.d.). Retrieved from http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/