Welsh Movielens Capstone

Lillian Welsh

2023-07-01

I. INTRODUCTION

This report describes the creation of a movie recommendation system using the MovieLens data set as part of the HarvardX Professional Data Science Certificate Capstone Course. In the Machine Learning Course prior to the Capstone, a smaller movielens subset from the dslabs library was used to train and compare five rating prediction models. The smallest Root Mean Square Error (RMSE) from those exercises was 0.881.

For the project described in this Capstone report, a larger subset (MovieLens 10M) was used to train a new and improved machine learning algorithm with a target RMSE of less than 0.8649.

Data Set Description

Code to import and split the 10M version of the MovieLens dataset (http://files.grouplens.org/datasets/movielens/ml-10m.zip) was provided in the course material for this, the first of the HarvardX Capstone projects. Separate columns were created and added (methods described below) for Age (years betwen rating movie release year) and Year (movie release year). Each row in the table represents a rating given by one user to one movie.

The test set is called **final_holdout_test** and is comprised of 10% of the Movielens 10M data. The other 90% is in the **edx** training set, which is described below.

```
glimpse(edx)
```

```
## 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   <int> 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, …
## $ 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…
```

There are 9000055 ratings, ranging from 0.5 to 5, in increments of 0.5.

Rating Summary Stats:

```
summary(edx$rating)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.500   3.000   4.000   3.512   4.000   5.000
```

The average number of ratings for an individual user was 128.7966885.

Below is a sample of ratings by userId created using the pivot_wider function. As there are over 10677 distinct movies and 69878 distinct users, there would be quite a few NAs for ratings by userId.

userId	Glengarry Glen Ross (1992)	Andromeda Strain, The (1971)	Silence of the Lambs, The (1991)	Manhunter (1986)	Nine Months (1995)	Last Action Hero (1993)	Pinocchio (1940)	Rambo: First Blood Part II (1985)	Maltese Falcon, The (1941)	Green Card (1990)
50805	4	NA	NA	NA	NA	NA	NA	NA	NA	NA
42011	NA	3.5	NA	NA	NA	NA	NA	NA	NA	NA
45985	NA	NA	4	NA	NA	NA	NA	NA	NA	NA
19660	NA	NA	NA	4	NA	NA	NA	NA	NA	NA
46073	NA	NA	NA	NA	4	NA	NA	NA	NA	NA
13268	NA	NA	NA	NA	NA	2	NA	NA	NA	NA
6910	NA	NA	NA	NA	NA	NA	2.5	NA	NA	NA
30265	NA	NA	NA	NA	NA	NA	NA	2	NA	NA
65174	NA	NA	NA	NA	NA	NA	NA	NA	3.5	NA
18227	NA	NA	NA	NA	NA	NA	NA	NA	NA	3

Project Goal

The goal of this project is to create a new movie recommendation system. This will be accomplished by training a machine learning algorithm using the inputs in the edx data set resulting in a Root Mean Square Error (RMSE) of less than 0.86490 on the final_holdout_test set.

Key Steps

The Root Mean Squared Error (RMSE) is the typical prediction error of a model. It follows that the goal of model refinement is to minimize the RMSE. Unlike squared residuals, the units of the RMSE match the units of the outcome variable and it is therefore a more readily interpretable metric. In this case for example, an RMSE of 1.0 would mean that the typical prediction would be one point too high or low, on the 5-point movie rating scale.

To select the final model, edx was partitioned in to edx_train and edx_test, using code similar to the partitioning code provided by the course to create final_holdout_test.

Using the information gleaned from EDA, several prediction models were built and tested on the edx data set. The model with the lowest RMSE was then chosen to validate with the final_holdout_test set.

II. METHODS

Feature Engineering

The title and release year in the Title column was split using the stringr package so that only the movie title was retained in the Title column. Separate columns were created for Age (rating year minus movie release year) and Year (movie release year) also using stringr.

Exploratory Data Analysis

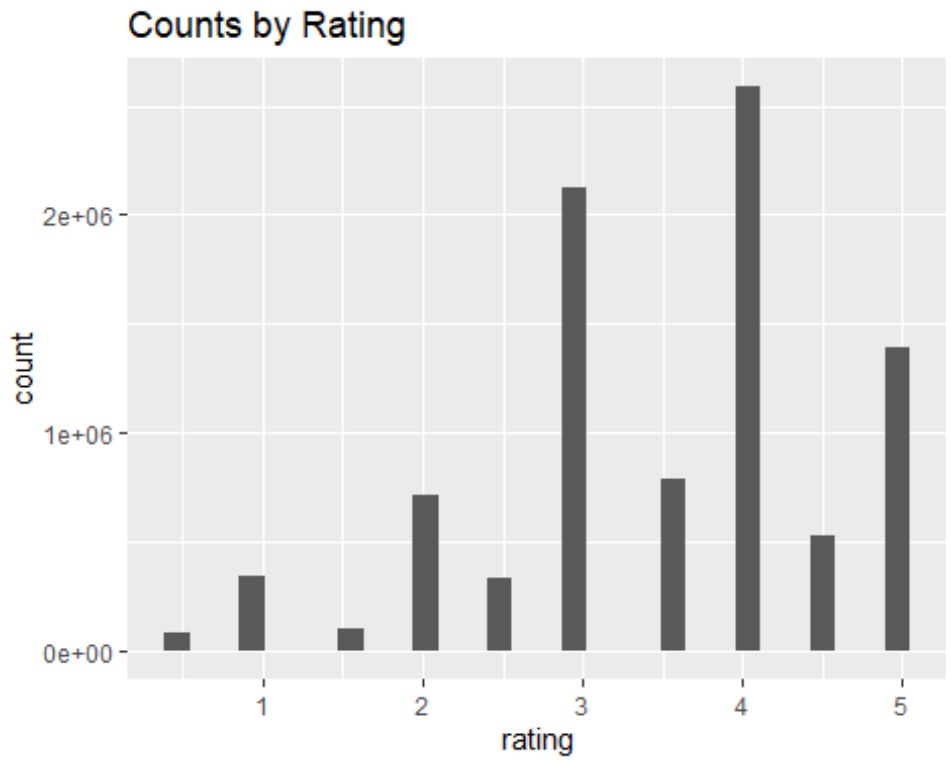
Describing the response variable, Rating

On average, each movie was rated 842.9 times.

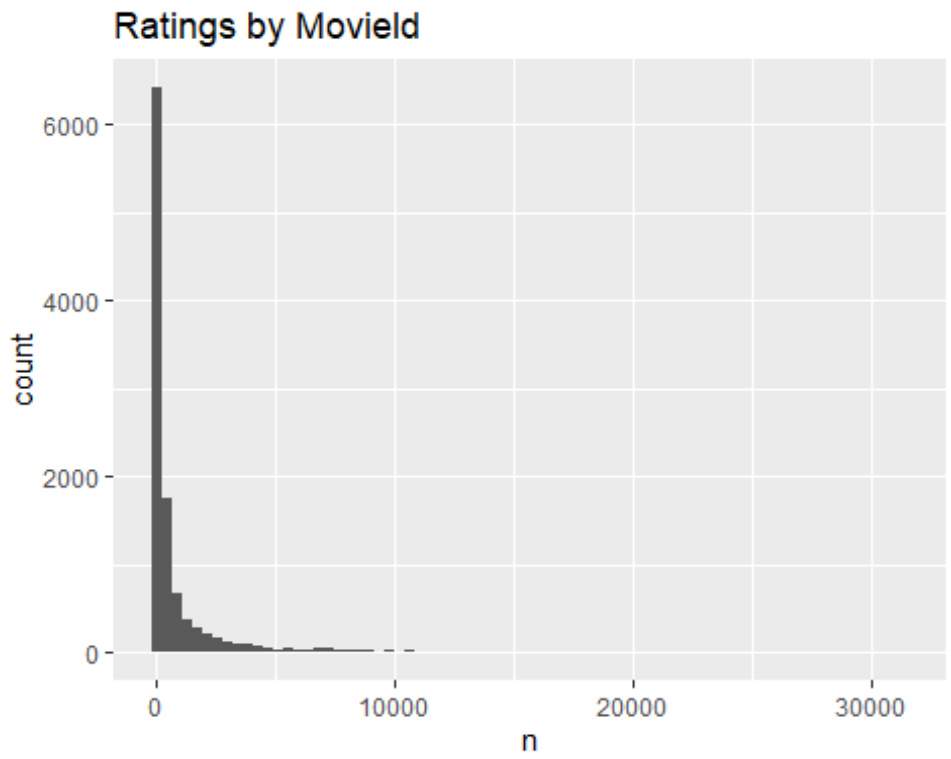
Half-ratings, those ending in .5, were given less often than whole-number ratings.

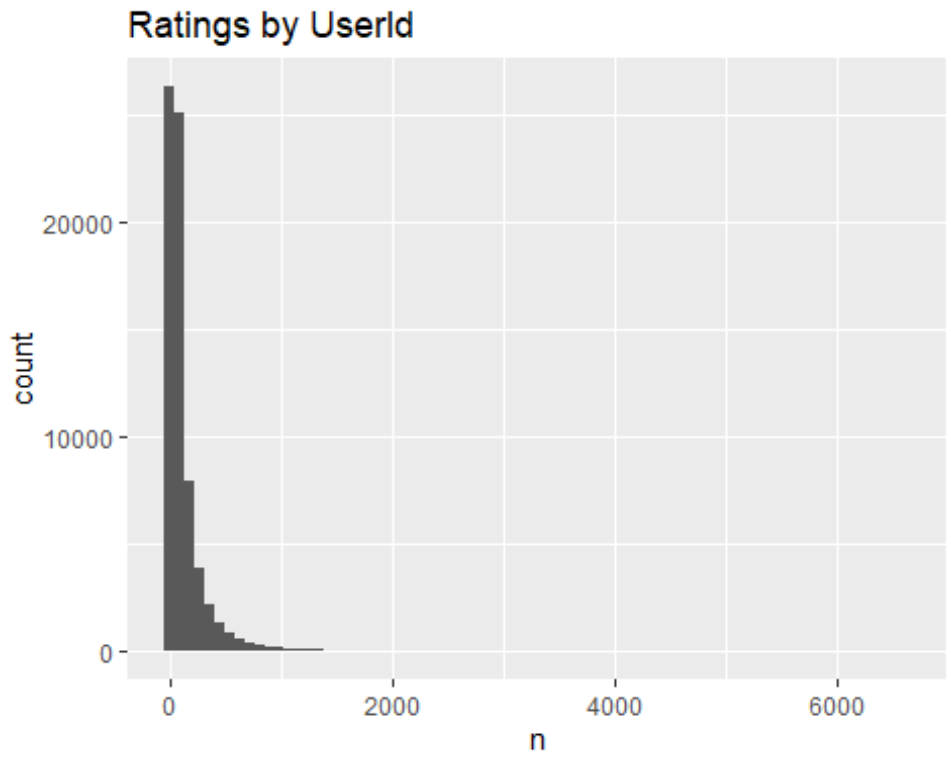
Count by Rating

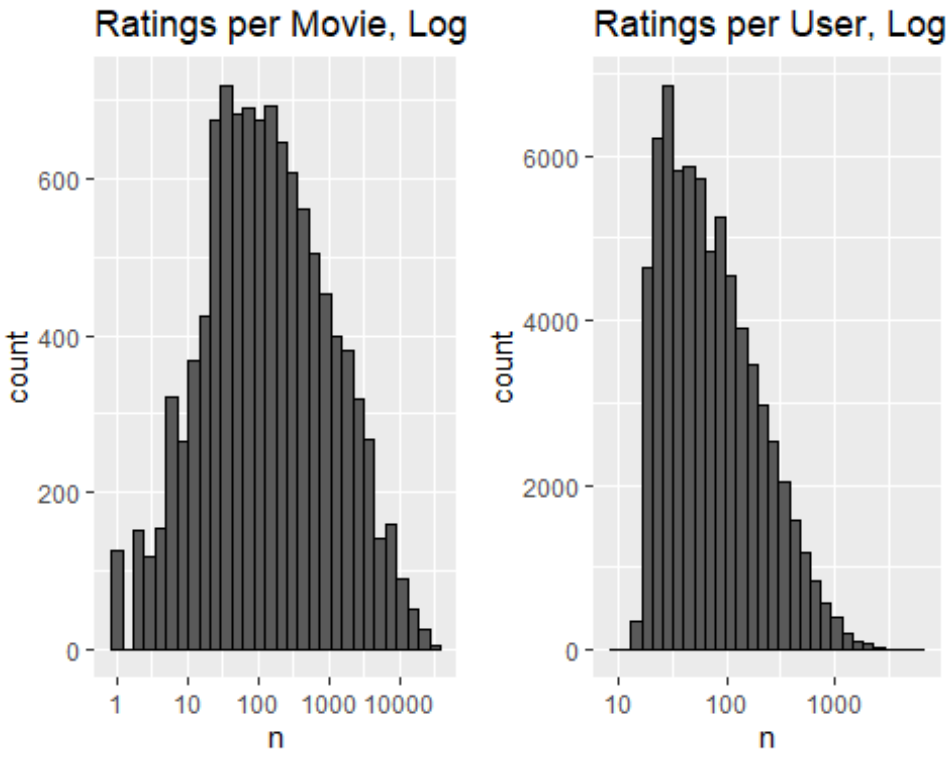
rating	n
0.5	85374
1.0	345679
1.5	106426
2.0	711422
2.5	333010
3.0	2121240
3.5	791624
4.0	2588430
4.5	526736
5.0	1390114



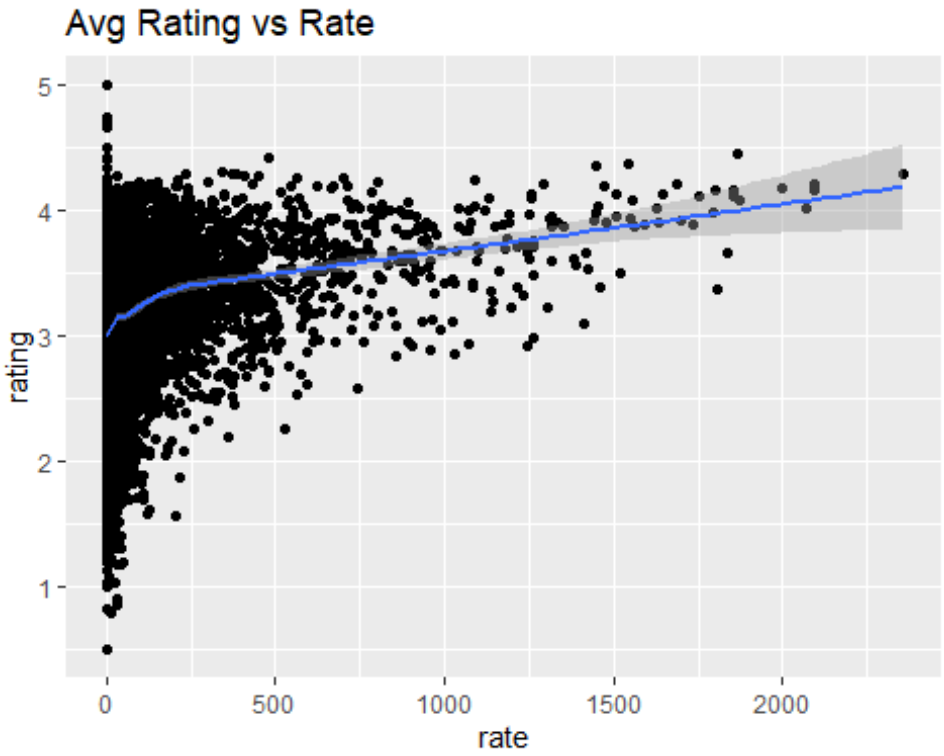
Ratings by movie and by user are both right skewed.







The more often a movie is rated, the higher its average rating. This was calculated using rate= number of ratings over years since movie release to 2009.



Describing the Explanatory Variables

MovieId

There are 10677 movies included in the edx data.

movieId Summary Stats

min	max	distinct
1	65133	10677

UserId

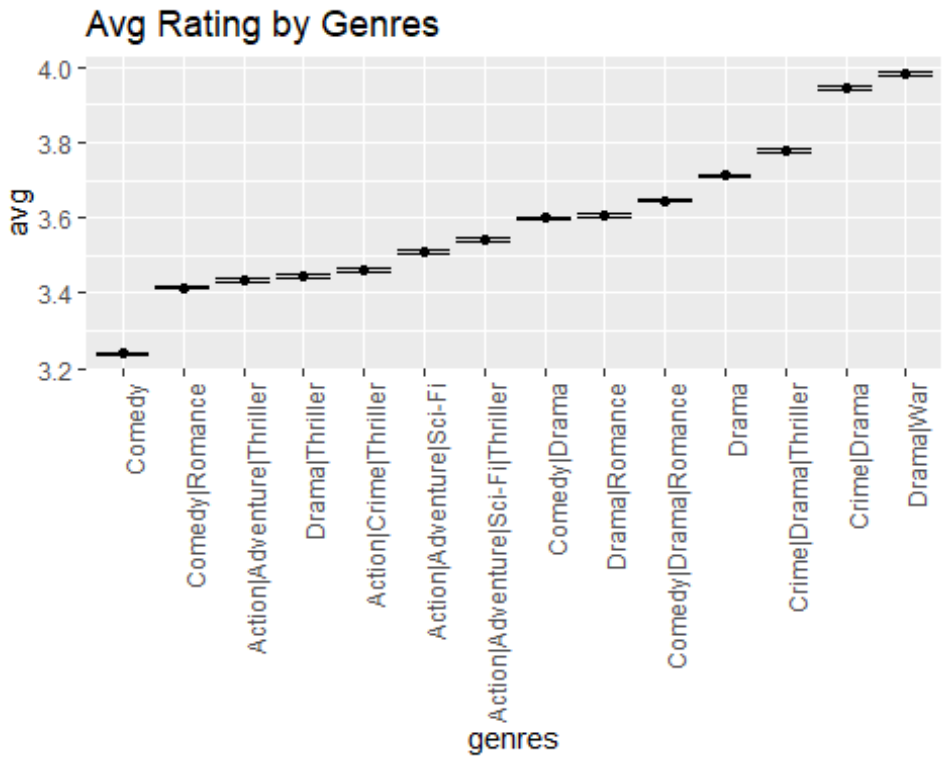
There are 69878 users included in the edx data.

userId Summary Stats

min	max	distinct
1	71567	69878

Genres

There is evidence of a strong relationship between Genres and avg movie Rating. The graph below is filtered by Genres with over 100,000 ratings as there are 797 distinct genres, many of which are combinations of 15 distinct single-genres. Genre categories that included Drama were the most highly rated.



Ratings per Single Genre

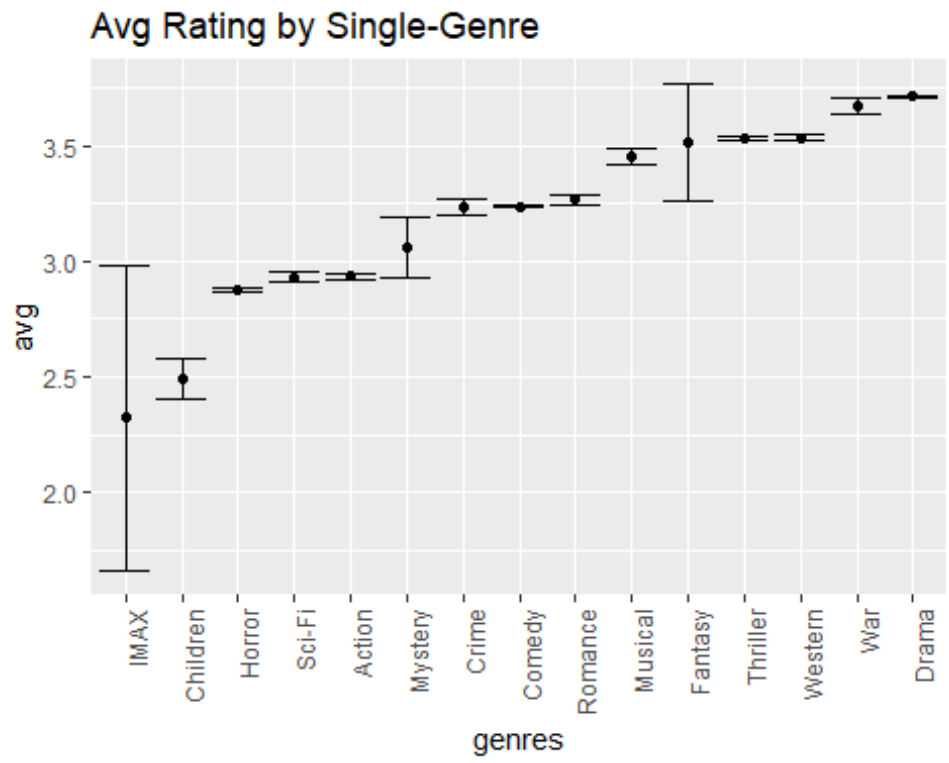
genres	n
Action	24482
Children	745
Comedy	700889
Crime	3197
Drama	733296
Fantasy	86
Horror	68738
IMAX	14
Musical	3851
Mystery	246
Romance	8410
Sci-Fi	10125
Thriller	94662
War	2300
Western	15300

When evaluating the summary stats for single-genre movies only, the skew seen in all movies was lessened, as the average rating was similar while the median rating decreased from 4 to 3.5.

Single-Genre Movie Stats

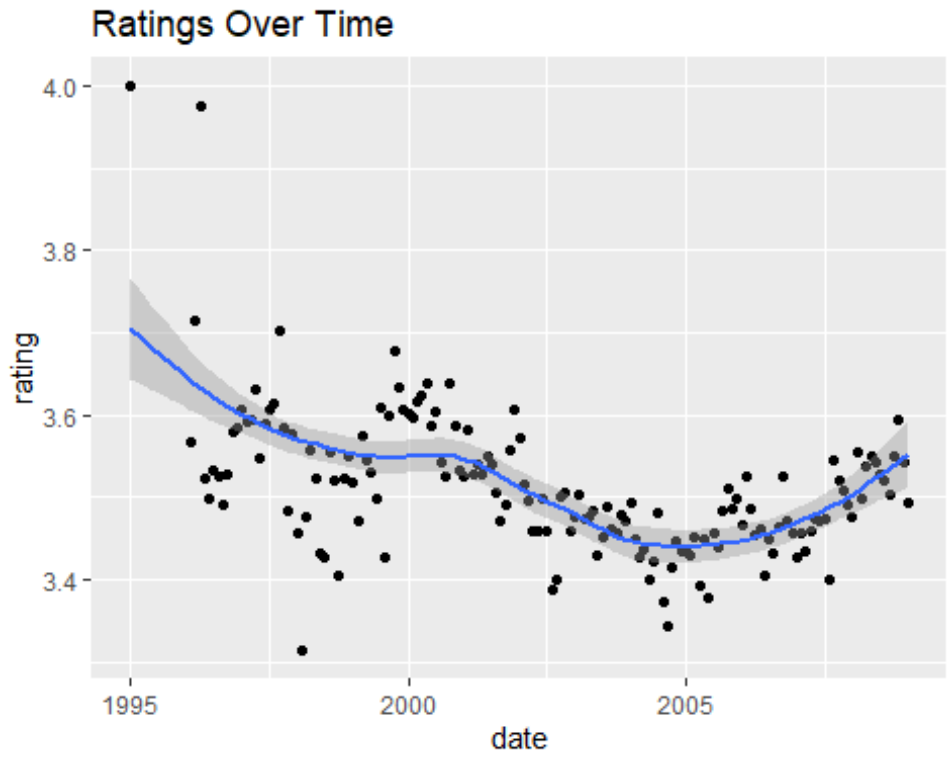
avg_rating	med_rating	stdev	var
3.445779	3.5	1.084475	1.176085

These single category, non-combined Genres exhibited a similar effect on Rating, with greater variation by genre, especially those genres with fewer ratings such as IMAX and Fantasy.



Date

There appeared to be a minor effect of rating Date (grouped here by month) on Rating



Modeling Approach

The first method assumed the same rating for all movies and users. As the predicted Rating, Y, minus the average Rating, mu, equals the error, epsilon…

$Y_{u,i} - \mu = \epsilon_{u,i}$
Pred-Avg = Error
where u= user and i= movie

it then follows that the predicted rating would equal the error plus mu.

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

Using this least squares estimate model, all differences would be explained by random variation, resulting in a naive RMSE of 1.06.

Biases

The naive model can be further improved upon by adding movie-specific (b_m) and user-specific (b_u) effects.

The *Movie Effect* model accounted for the fact that some movies are just generally rated higher than others. The more often a movies was rated (the more popular movies), the higher the average rating.

The *Movie+ User Effect* model also accounted for user-specific variability (users who tended to rate movies substantially higher or lower than the average).

method	RMSE
Just the average	1.0600537
Movie + User Effects Model	0.8646843

As shown in the Exploratory Data Analysis, there was also strong evidence of a Genres, Age, and Year effects. Here, in the UMGAY (User, Movie, Genres, Age, Year) model, those biases were each accounted for.

method	RMSE
Just the average	1.0600537
Movie + User Effects Model	0.8646843
UMGAY Model	0.8637526

Regularization

Looking at the predictions for top and bottom 10 movies by rating shows that these are very obscure movies.

10 Best Movies

x
Hellhounds on My Trail
Satan's Tango (Sátántangó)
Shadows of Forgotten Ancestors
Fighting Elegy (Kenka erejii)
Sun Alley (Sonnenallee)
Blue Light, The (Das Blaue Licht)
Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva)
Life of Oharu, The (Saikaku ichidai onna)
Human Condition II, The (Ningen no joken II)
Human Condition III, The (Ningen no joken III)

10 Worst Movies

x
Besotted
Hi-Line, The
Accused (Anklaget)
Confessions of a Superhero
War of the Worlds 2: The Next Wave
SuperBabies: Baby Geniuses 2
Disaster Movie
From Justin to Kelly
Hip Hop Witch, Da
Criminals

This obscurity makes sense because the predictions are based on very few ratings. In most cases, these movies had only one reviewer.

Number of Users Rating Predicted Top 10:

```
##  [1] 1 2 1 1 1 1 4 3 4 4
```

Number of Users Rating Predicted Bottom 10:

```
##  [1]   2   1   1   1   2  56  32 199  14   2
```

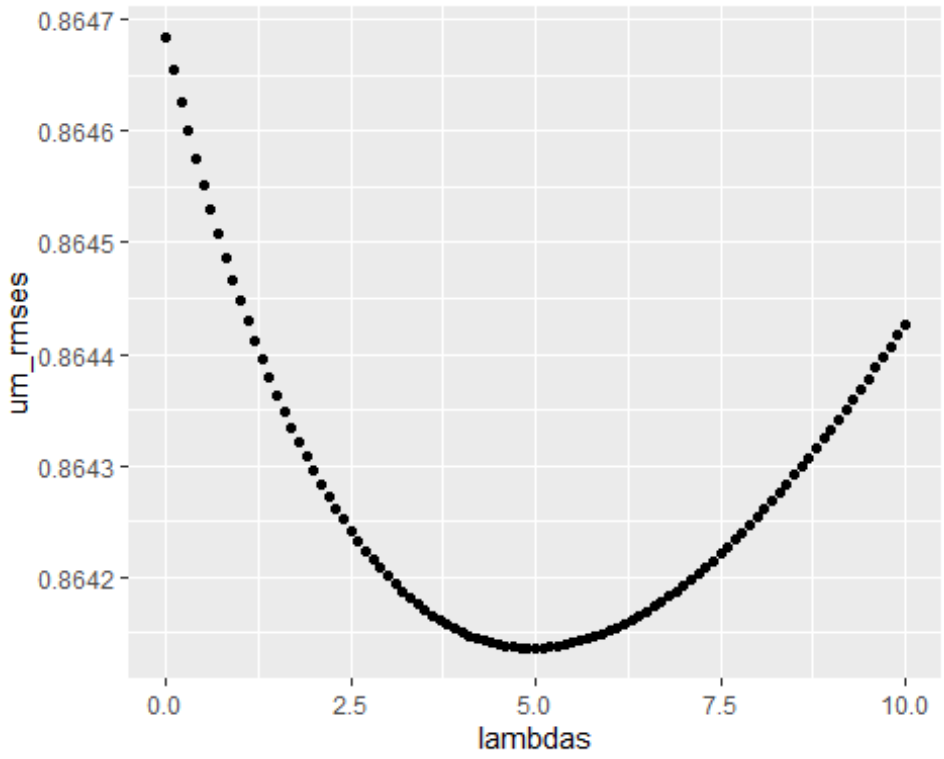
Estimates hold a lot of uncertainty when based on few users. Regularization penalizes large estimates formed using small sample sizes by dividing the sum of the residuals (sum(rating-mu)) by the number of ratings for the movie plus a penalty term, lambda (n_i + lambda). Using cross-validation, a range of lambdas can be tested. The lambda which yields the smallest RMSE is the one used to optimize the model.

Below are the predictions for 10 highest and lowest rated movies after regularization of movieId with lambda=3. The movies are no longer obscure.

x
Shawshank Redemption, The
Godfather, The
Usual Suspects, The
Schindler's List
More
Casablanca
Rear Window
Sunset Blvd. (a.k.a. Sunset Boulevard)
Third Man, The
Double Indemnity

x
SuperBabies: Baby Geniuses 2
From Justin to Kelly
Pokémon Heroes
Disaster Movie
Carnosaur 3: Primal Species
Glitter
Pokemon 4 Ever (a.k.a. Pokémon 4: The Movie)
Gigli
Barney's Great Adventure
Hip Hop Witch, Da

Choosing Lambda for Optimization:




```
Optimal lambda

## [1] 4.9
```

method	RMSE
Just the average	1.0600537
Movie + User Effects Model	0.8646843
UMGAY Model	0.8637526
Regularized User + Movie Effect Model	0.8641362
Regularized UMGAY on Train	0.8632248

III. RESULTS

The Regularized UMGAY model trained with edx_train and tested on edx_test yielded the lowest RMSE (0.86322), an 18.6% improvement on the naïve RMSE.

The Regularized UMGAY model was therefore selected to train the full edx data set and then tested on the final_holdout_test set. The result was an RMSE of 0.86386, below the project goal of less than 0.86490.

method	RMSE
Regularized UMGAY on Test	0.8638649

IV. CONCLUSION

The RMSEs from the train and test results were similar, indicating that the model was neither over or under fitted.

The selected UMGAY Regularization model was effective and predicted ratings better than the previous models described in the course. This project had technical limitations in terms of RAM due to the size of the data set. Therefore, linear regression modeling with the caret package was not within the scope of this analysis. Similarly, while a simple regression tree was constructed in trials, parameter tuning for more informative random forest ensembles was prohibitive.

This project could be expanded with the exploration of these types of ensembles using packages such as parsnip in tidymodels with a high-performance operating system. An additional task could be using the Rating information to fill in the NAs for each user and building a recommendation system with that data.

Another interesting extension could include analysis on demographics of users, which was not a part of this movielens dataset. Also, examining unusual trends in user ratings to identify potential bots or other artificial ratings could be used in conjunction with the regularization methods used here. This type of identification and penalization process is increasingly necessary for public-facing systems.