

Stat 410 Aidan's Movies

Aidan Gerber

April 29th, 2022

Contents

Project statement	1
Data description	2
Exploratory data analysis	2
Data analysis	4
Models	5
SLR	5
MLR	6
Transformed and Polynomial MLRs	7
Polynomial and Interaction	8
Ridge and Lasso	8
Ridge	9
Lasso	10
Comparison	11
Steps with AIC and BIC	11
General Additive Models	13
Large GAM	13
GAM with BIC Variables	13
Comparison	13
Summary and discussion	14
Predictions	15
References	16
Code and data	16

Project statement

What is the purpose of your analysis? What effect are you investigating? Why? The purpose of my analysis is to determine what movies I will enjoy the most. I will do this by creating a model with the response variable of how much I liked a movie out of 100 and predicting it with data on various movie reviews, genres, cast and crew, and more. I am doing this so that I can figure out which movies I will want to

watch because I can be confident that I will enjoy them.

Data description

How were the data collected? How do these data help you answer the relevant research questions? The data was collected over time by me. A partial dataset in json form is available on my website here: https://www.tradethisandthat.com/movies/api/all_movies/. In the references there is python code I wrote to turn the mySQL database into a csv. Data from TMDB was collected using their open API. Data from IMDB was collected by web scraping their page for awards and finding any Oscars as well as scraping their page for rating distributions. Much of the data is gotten with python code that I run on movie addition with Django. My rating and metacritic ratings are collected by me.

Exploratory data analysis

What are the basic features of the data? What are the variables? How are the variables related? Are there any unusual patterns? None of the variables are too highly related to my rating. The highest correlations are to `imdb_rating`, `tmdb_rating`. The other variables above 0.6 correlation are variables that are highly correlated with `imdb_rating` like `imdb_arithmetic_mean`, `imdb_us_rating`, `imdb_top_1000_rating`, and `imdb_not_us_rating`. Something important to note is that most variables in the dataset make sense linearly such as `my_rating` and `imdb_rating`. Since `my_rating` is on a 0 to 100 scale, in an MLR setting other variables within a specific range work best if they are not transformed such as IMDB rating, between 0 to 10, TMDB rating between 0 to 10, or Metacritic rating between 0 and 100. When attempting to use uncapped data, they were insignificant. This is evident when looking at IMDB count vs IMDB Top 1000 Count. When using a log transformation, the full IMDB count becomes more significant which makes sense since it includes more data.

This R code creates the main dataset that I am analyzing. The first thing I did was deciding to interpolate null values rather than removing them. The only category with null values was Metacritic rating since some movies are not on Metacritic. To interpolate Metacritic ratings, I created a basic MLR to predict them from movie characteristics that are not related to me or my ratings.

The next step was to make `mpaa_name` a numeric factor allowing it to be used as a dummy variable. The most common MPAA ratings for movies that I have seen are PG-13(209), PG(179), R(135), G(27), and NR(21).

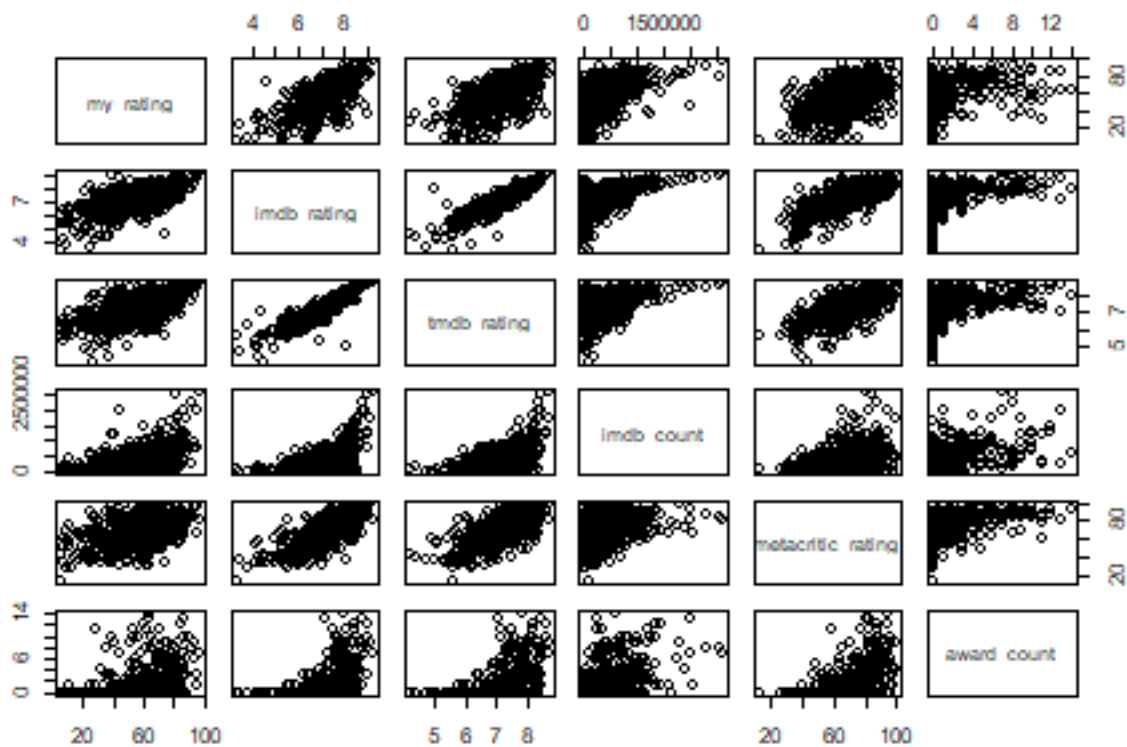
After that, I wrote a function to check if a person of a specific id appeared in a movie and return 1 or 0. This combined the separate credits and movies tables. Credits are 1-to-1 related with both movies and people so I checked those columns.

Finally, it was time to use dplyr to get the dataset. I began by arranging by IMDB rating since Metacritic ratings are most related to IMDB ratings. Then, I selected the desired columns from the larger dataset. After that, I created dummy variables for both genres and crew. I chose every genre that I have seen more than 100 movies from.

- Comedy(266)
- Adventure(239)
- Action(213)
- Drama(167)
- Family(166)
- Science Fiction(132)
- Animation(126)
- Thriller(102)

Choosing cast and crew members was more difficult. I selected people that I thought had the ability to make movies more than the sum of their parts. I decided to choose many more people than I expected to significant and have them whittled down through AIC and BIC.

Looking at the correlation matrix of my rating to all the variables in the final dataset, I noticed that `imdb_rating` is most significant. Barely behind are `tmdb_rating`, `imdb_arithmetic_mean`, `imdb_top_1000_rating`, `imdb_us_rating`, `imdb_not_us_rating`, `imdb_rating_percentile`, and `tmdb_rating_percentile`. None of these are surprising and each of them are correlated with each other. There were also reasonably strong correlations between `my_rating` and `imdb_count` with less strong ones to `award_count`, `runtime`, and `revenue`. In terms of genres, the highest positive correlation was to drama with large negative correlations to comedy and family. For people, the highest positive correlations were to John Williams, Wally Pfister, Christopher Nolan, Harrison Ford, George Lucas, and Matt Damon.

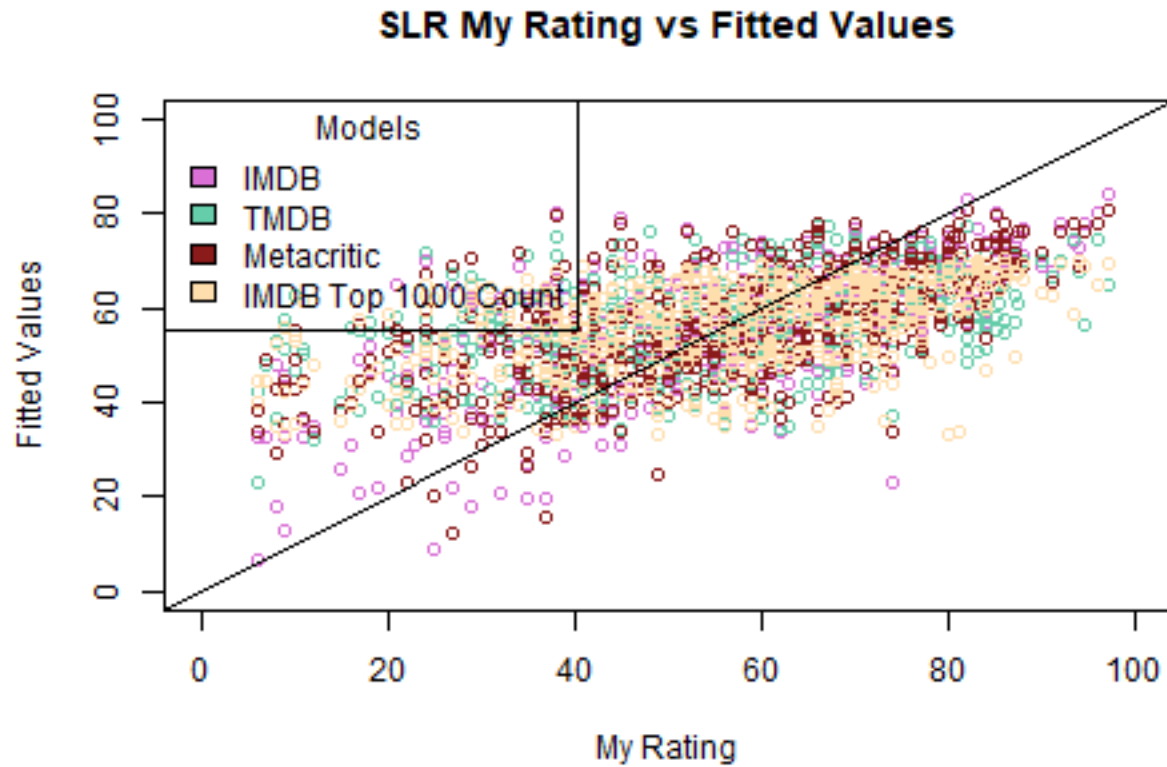


Data analysis

What is the statistical model (or models) that you are using? Why is this an appropriate model to use? Do the model diagnostics contradict the model assumptions? How do you interpret the results from the statistical analysis in the context of your research question? I tried fitting many different models including SLRs, MLRs, polynomials, montone transformations, ridge, lasso, and GAMs.

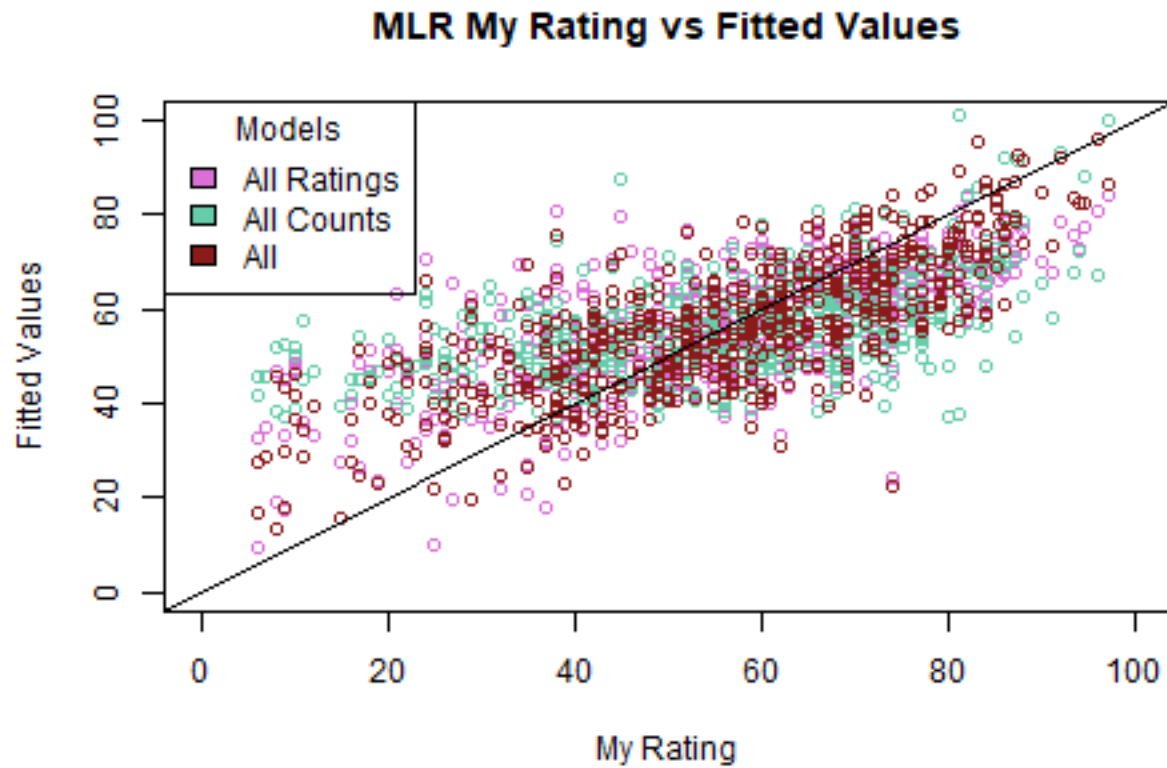
Models

SLR



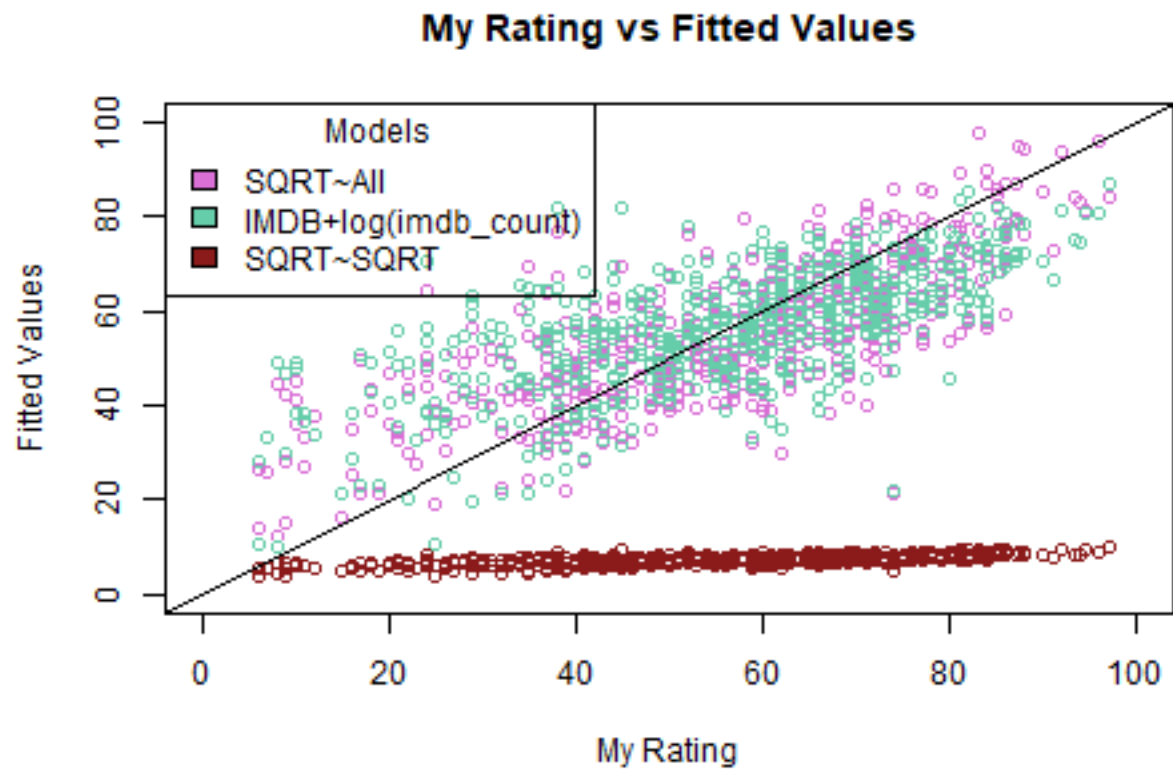
Each of these SLRs have substantial flaws. IMDB ratings are most predictive with an adjusted r squared of 0.4314773, higher than the other models. Each of the models have relatively low betas. This is because RSS highly penalizes particularly far values so it places values toward the center of the range. This is especially apparent for Metacritic which has a $\hat{\beta}$ of 0.6126243. This means that the total range of predictions 23.1598334 to 77.0707686 despite my rating range going between 6 to 97.

MLR

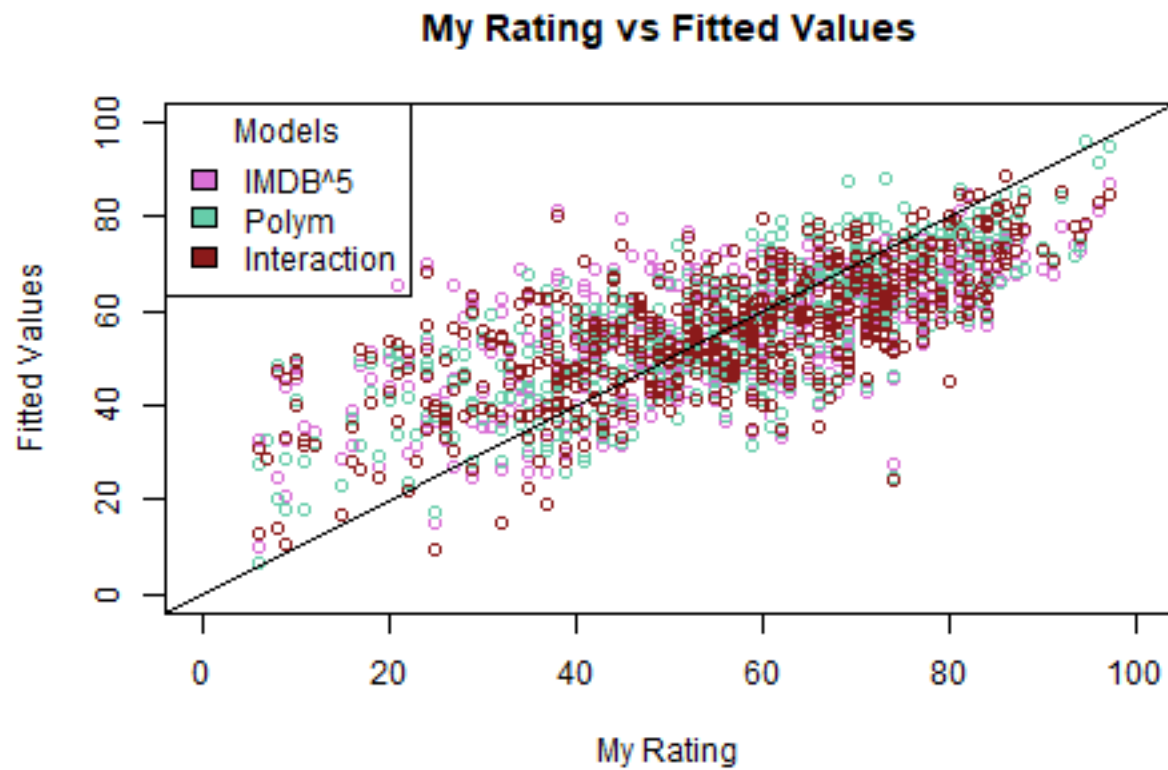


The full MLR is an improvement but all ratings and all counts models are not particularly effective. Looking at the ratings MLR, the only significant term is the IMDB rating and the adjusted r squared only increases 8.2000213×10^{-4} between the 2 models. Although many of the counts are significant, the overall model r squared is quite low at 0.2895117. This occurs because of the uncapped counts that I mentioned in data analysis.

Transformed and Polynomial MLRs



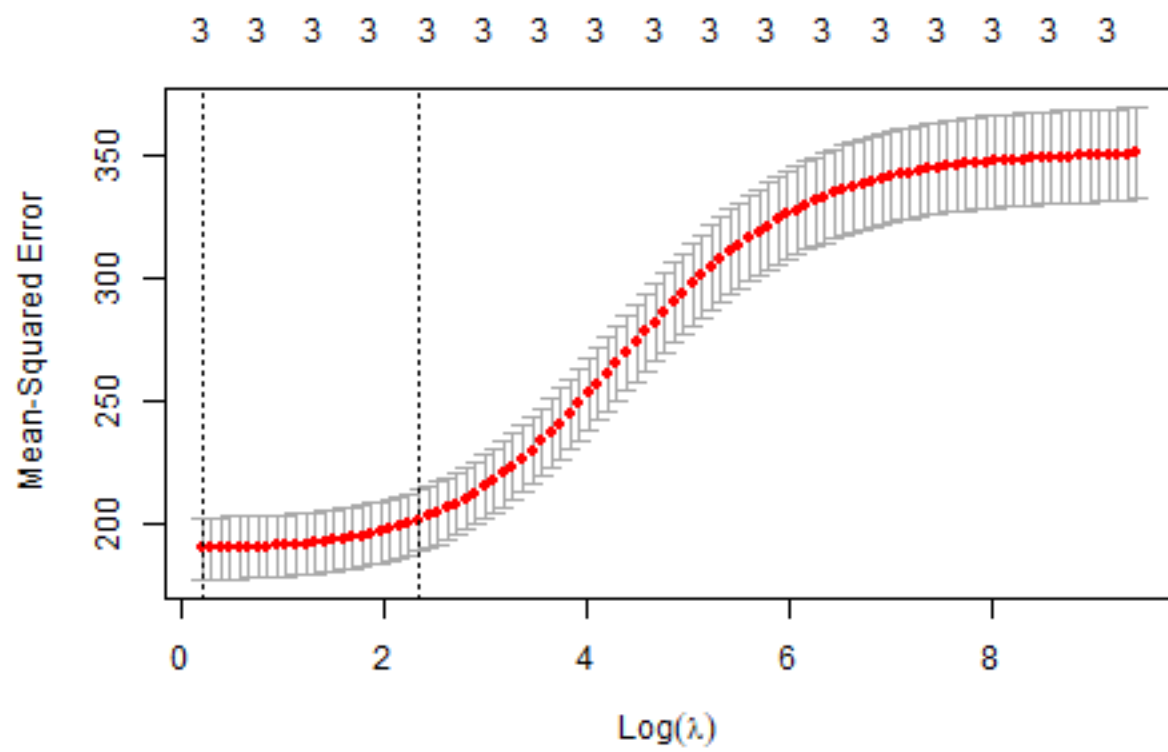
Polynomial and Interaction



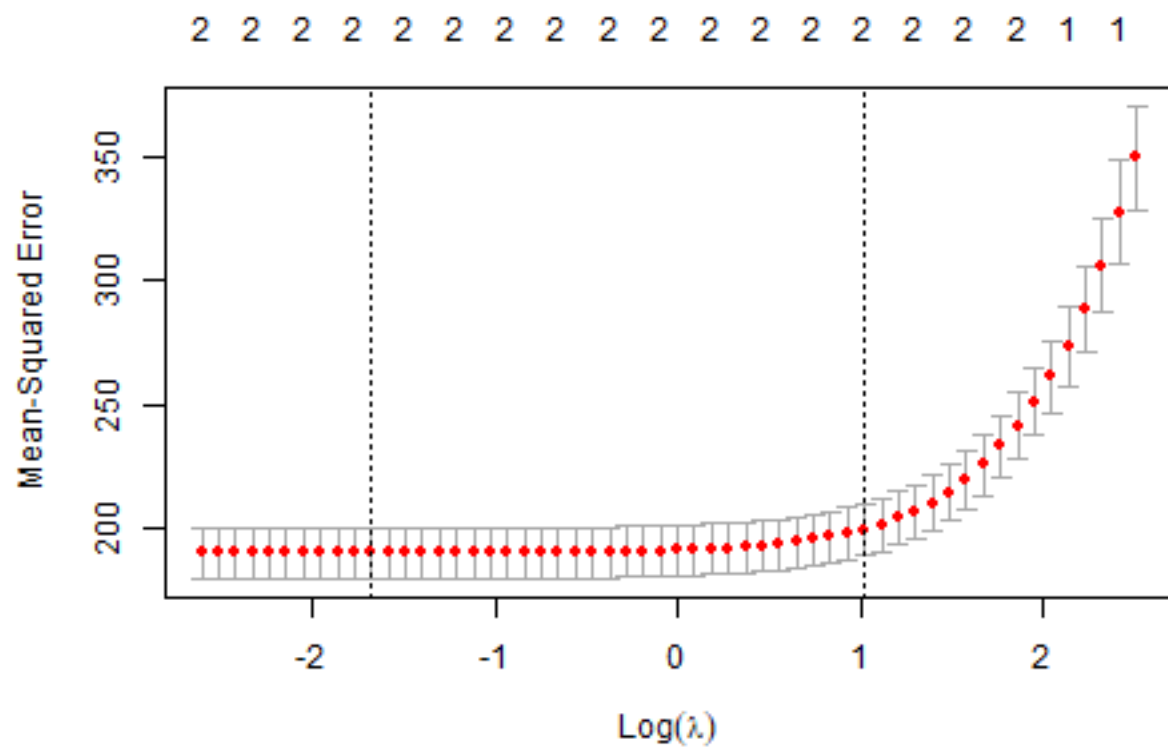
Ridge and Lasso

Create data matrix to be used for both ridge and lasso.

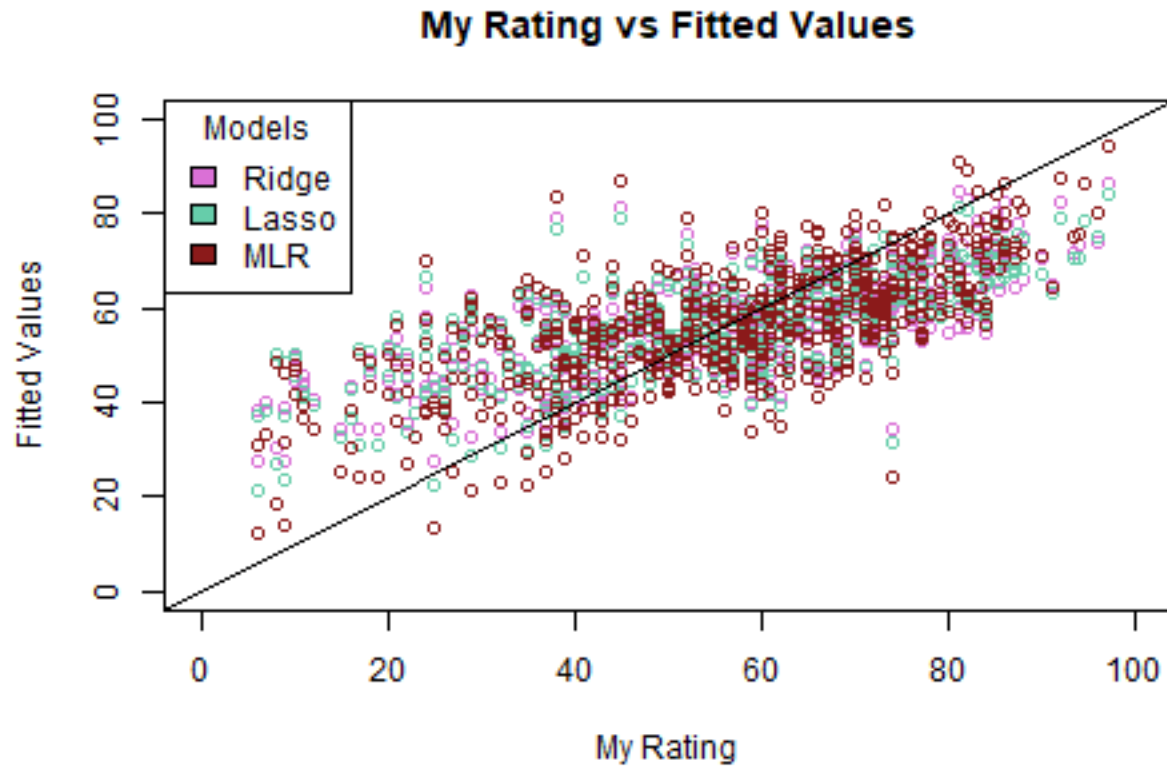
Ridge



Lasso



Comparison



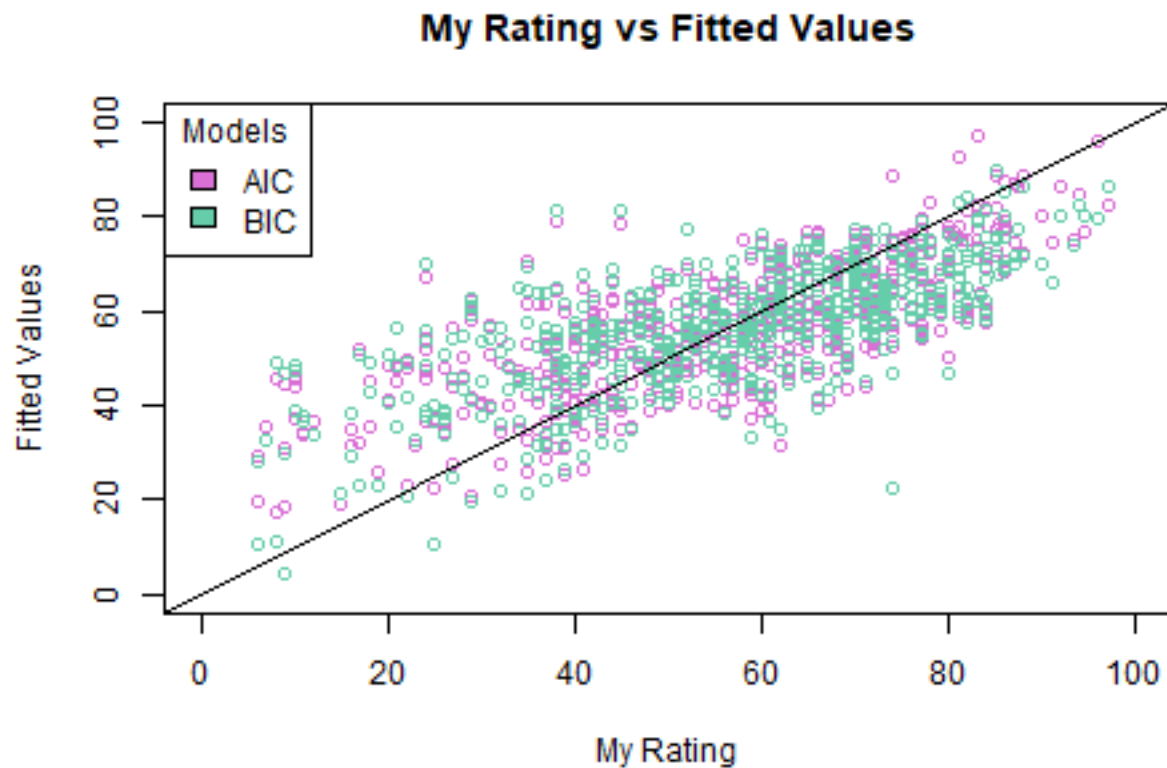
Adjusted R Squared at 1 standard deviation for ridge and lasso:

- ridge: 0.4658365
- lasso: 0.4671242
- mlr: 0.4644028

Lasso and ridge result in a slight improvement over the MLR since they take advantage of the bias variance tradeoff. Looking at the beta values, the different models change the coefficients a lot despite getting similar results. MLR values IMDB count and IMDB rating much more than ridge or lasso. Lasso actually removes $\sqrt{\text{award_count}}$ to not much of a detriment.

Steps with AIC and BIC

Because the p-value for the full model is significant in comparison to the null model, it makes sense to continue with AIC and BIC to find a significant model. The use of AIC and BIC is to penalize adding more parameters because any parameter improves the fit.



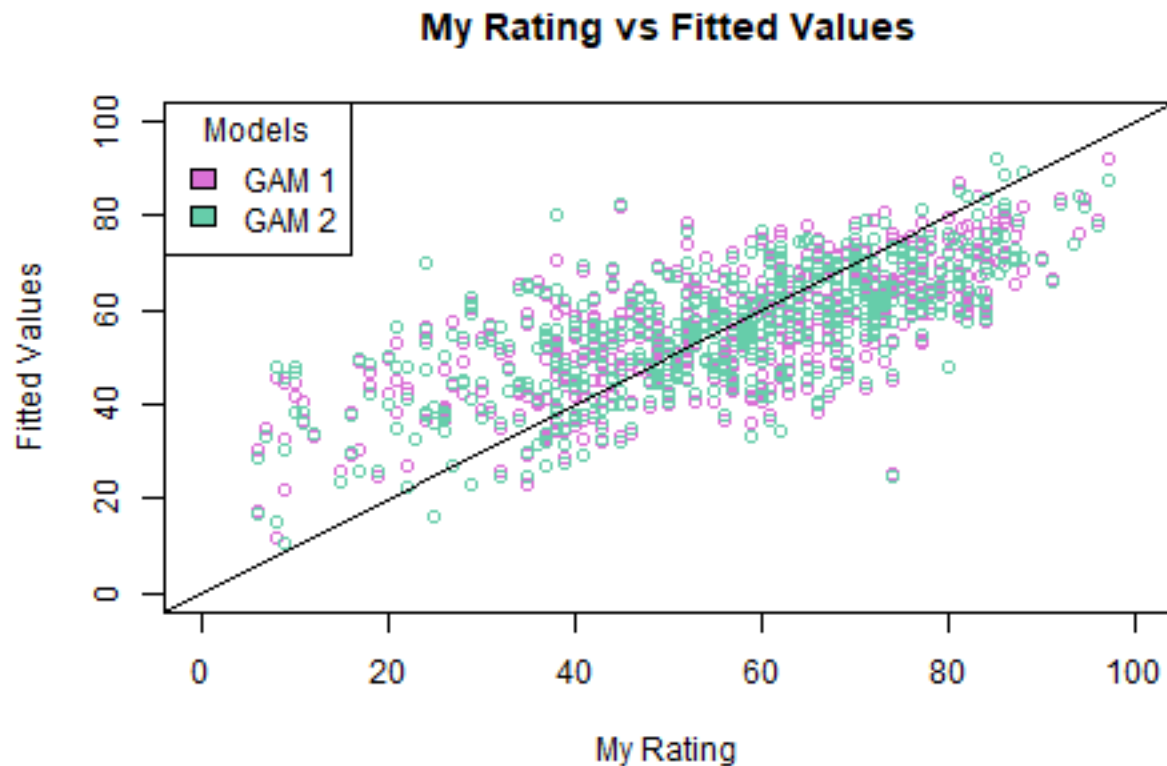
AIC and BIC had very different results. The AIC resulted in my highest adjusted R^2 of any model (0.5080743). The final AIC selected 21 coefficients. This is very different than the BIC which selected 4. Only 3 variables in the AIC have negative betas, `is_jack_nicholason`, `is_adventure`, and `imdb_rating`. This is unique to this model and is possible because lots of people have highly positive coefficients and `log(imdb_count)` is also highly positive. Moreover, `imdb_rating` does not have a significant p-value in this model. The BIC chooses a much sparser with just `imdb_rating`, `log(imdb_count)`, and `is_john_williams`. John Williams is not only an excellent composer but links many of my favorite movies such as Indiana Jones and Star Wars while only appearing in 2 movies I rated below a 60: Close Encounters of the Third Kind and The Lost World: Jurassic Park. Even with such different models, both the AIC and BIC had similar adjusted r squared values and their adjusted r squared values were generally in line with other models.

General Additive Models

Large GAM

GAM with BIC Variables

Comparison



The fit plots have been withheld for spacing reasons. For the first gam: IMDB rating and metacritic rating are linear. TMDB rating is linear until the rating reaches about 6 before increase in slope between 6.5 and 7.5 before flattening. This means that the impact of an increase in TMDB rating from 7.5 to 7.6 is expected to have a larger impact on my rating than an increase from 6.4 to 6.5. IMDB count has a very steep slope until it reaches around 500,000 before flattening out. This impact is fixed by using a log transformation. Finally, runtime actually flips. When runtime is under an hour, an increase in runtime leads to an expected increase in my rating. However, as runtime passes 150 minutes, a higher runtime means an expected decrease in my rating. IMDB arithmetic rating is a new addition to this model and actually has a negative beta model. This exemplifies the slight difference in the way the public IMDB rating is calculated vs the arithmetic mean. The public IMDB rating is calculated using a secret formula to prevent review bombing and is actually a weighted average. This reveals that using the weighted mean is more powerful than the arithmetic mean and

that the weighted average used by IMDB is useful.

For budget, there is a quick slope upward at the beginning revealing that a percentage increase in budget is more important than a dollar amount increase. However, this should be taken with a grain of salt as there are flaws to using budget as a prediction metric. Firstly, movies have been made at all different times and budgets are not normalized for inflation. This harms the predictive power of using budget. Moreover, some movies do not have public budget information which is related to their popularity. The mean IMDB count for a movie with a budget that is not 0 is 4.3154913×10^5 while for movies with a budget that is 0 the mean count is 3.8796132×10^4 , more than 10 times smaller.

Summary and discussion

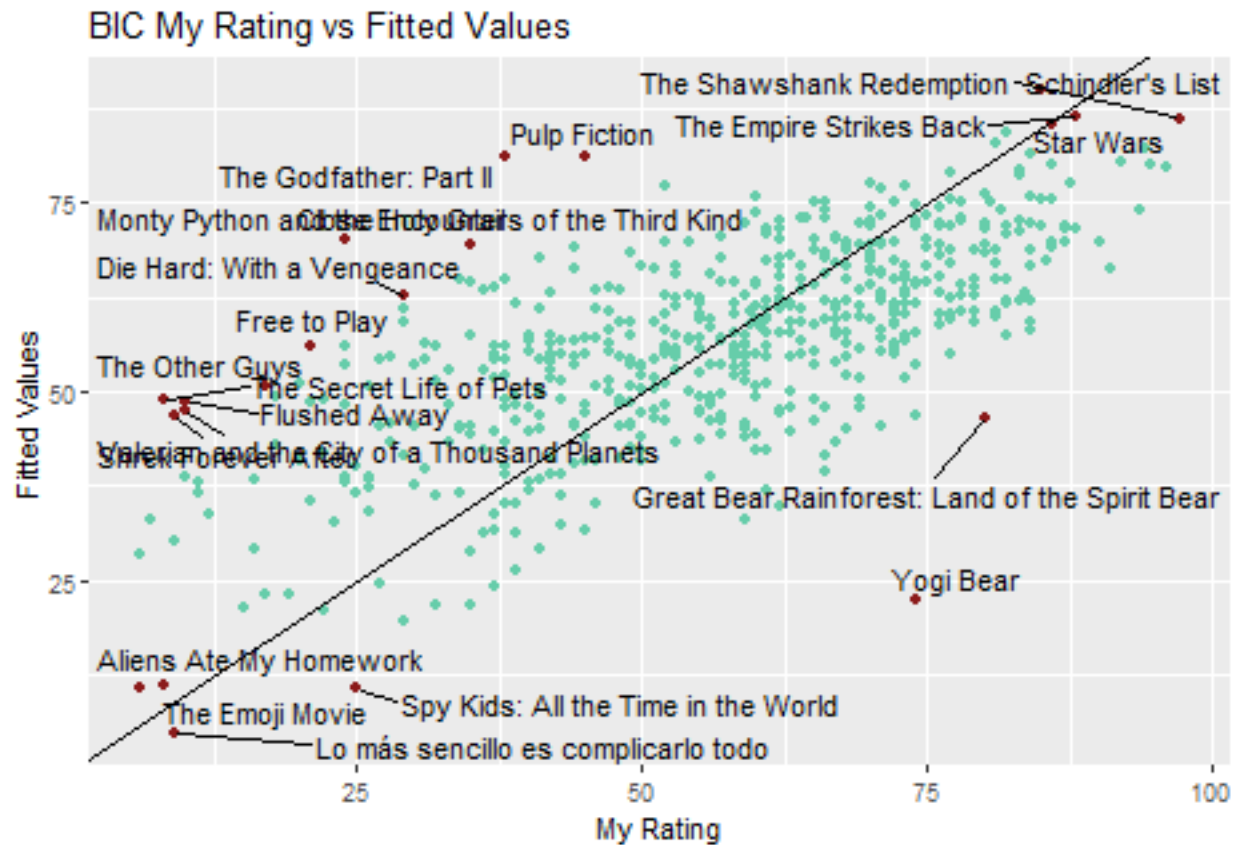
What are the main conclusions of your analysis? What are the main limitations? What might be investigated in future research? Ultimately, the model that I ended up liking the most was the BIC. This model includes `imdb_rating`, `log(imdb_count)`, and `is_john_williams`. The model has an adjusted r squared of 0.4735121. This is very similar to the GAM version of the model which includes a smoothed version of `imdb_rating` and `imdb_count`, a standard dummy variable of `is_john_williams` and an interaction term between `is_john_williams` and `log(imdb_count)`. This resulted in an adjusted r squared of 0.4749977. Since the simple MLR found by BIC is much simpler and has a very similar r squared, it is the better choice for interpretability.

Interpretation. A movie with an IMDB rating of 0, that has been seen by 1 person, and does not have John Williams would be expected to have a rating of -53.7527001. For each 1 unit increase in IMDB rating, holding all else equal, my expected rating increases by -53.7527001. For each 1 unit increase in `log(IMDB count)`, holding all else equal, my expected rating increases by 10.9579689. If John Williams is credited in the movie, holding all else equal, my expected rating increases by 2.5708082.

My largest conclusion in this project is that predicting my ratings is very difficult and that adjusted r squared of around 0.47 are the best I am going to do. I reached this conclusion since I fit many different models and ended up with results consistently similar despite highly different betas. Even with these varied betas, models consistently overpredicted low values and underpredicted high values. This makes sense considering each of the things that I am predicting based on are aggregate metrics. When trying to turn these aggregate metrics back to an individual prediction, the model hedges and predicts values generally closer to the center. This allows the model to miss by less when it is not particularly close.

Fit BIC Outliers

- Yogi Bear: 51.6646477
- Great Bear Rainforest: Land of the Spirit Bear: 33.4587349
- The Smurfs: 27.2795077
- Robots: 27.0933446
- Misha and the Wolves: 26.4569976
- Monty Python and the Holy Grail: -45.9879978
- The Godfather: Part II: -42.9105868
- The Secret Life of Pets: -40.8222339
- Flushed Away: -38.702369
- Shrek Forever After: -37.6574857



Predictions

The movies being predicted are the most reviewed movies on IMDB that I have not seen. In order they are Inglorious Basterds, The Silence of the Lambs, Saving Private Ryan, The Departed, Shutter Island, The Green Mile, Titanic, American Beauty, American History X, Braveheart, and The Hottie & the Nottie.

These predictions are mostly reasonable but they do fall within very large ranges. On the high end, the predictions get above 100 and on the low end they get below 0. I feel confident that I would rate nearly all of these movies within the given range; however, the ranges encompass over half of the rating space. These ranges means the model must hedge because it cannot continuously predict values of 90 for movies with ranges between 63 and 117 because the range on the upper end is impossible and the model will attempt to avoid it.

Overall, predicting my movie ratings is very difficult to make highly accurate, but it is certainly possible to do much better than random guessing, and with careful variable, it is possible to get a relatively accurate model with high interpretability.

References

What are the sources for the data and any additional statistical resources that you used to support your analysis? Data sources:

- TMDB API
- IMDB Website
- Personal Movie Website

Code and data

Include the code and data to reproduce your analysis. The code should include clear comments and should run correctly without errors. Code and data included throughout.

Datasets available on Github and on the hosted version of the html: movie.csv, credit.csv, continuous__movies, and pdf form