Money Talks: An Analysis of Profitability with TMDB Dataset

StatR 502

Rebecca Hadi

3/5/2018

Contents

Analysis Objective

Research question: To what extent can the probability that a movie will be profitable be modeled given the predictor variables?

Goals of analysis:

1. Understand ability to predict profitability in this data set.
2. Identify if genre has any impact on profitability in this data set.
3. Identify any other variables that have an impact on profitability.

The Data

This data set was found on Kaggle at https://www.kaggle.com/tmdb/tmdb-movie-metadata/data. It contains various data points from the website "The Movie Database" (https://www.themoviedb.org) for 4803 movies. TMDb is a community built movie and TV database. It is not clear how the sample was derived as there are likely more than ~5000 movies that exist. It's a fair assumption that all movies get made with the intention of being profitable (so that even more movies can be made!), so it would be interesting to understand if there are any significant predictors of profitability.

The initial data set before cleaning contains one record per movie with columns such as budget, revenue, genre, release date, keywords, production company, vote average, vote count, popularity, tagline, title, language, production country, and run time.

With my modeling question in mind and uncertainty around how the data were pulled from the site, I wanted to investigate the data set and remove possible sources of bias and skew. The final data set I used after the cleaning and feature engineering is described below.

Data Cleaning

Data removed (movies can meet multiple criteria):

- Movies that were not released in English. Upon inspection, the data were highly skewed toward English as a release language. In my project proposal I had included original language as a potential predictor, but after examining my data set I decided that there were not enough data for non-English language movies for this to be a meaningful variable.
- Movies that were released prior to the year 2000. It's possible that inflation could skew the input variables of revenue and budget, so I wanted to only look at movies that were released somewhat recently, which I am defining as the year 2000 or greater.
- Movies that had zero revenue. These appear to be missing values from the TMDB data set. For example, the movie "Blades of Glory" was listed as having zero revenue (which was consistent with the TMBD site), but a quick Google search revealed this movie actually had $146M in revenue.
- Movies that had not yet been released.
- Movies with the genre of "TV Movie". This is a different type of movie than what we are trying to model (e.g. revenue from box office)

After cleaning, I am left with 2146 movies for analysis.

Features engineered:

- Profit: Difference between revenue and budget.
- Profitable: Binary indicator of whether or not profit was greater than 0. This is the response variable.
- Release year: Extracted from release date.
- Genre: The original genre data was in JSON format. Upon extracting, it created a data frame that had one row per genre (making a single movie have as many rows as distinct genres). I had a few ideas on how to handle, ranging from picking a primary/arbitrary genre for each movie to force there to be one row, or allowing the data to have multiple rows. I ended up manipulating the data to create a column specific to each genre (e.g. "f.action" is a 0 or 1 if the movie has action as the genre). Then, I created meaningful grouping of genres based on what the most common genres in the data set were and common groupings that exist in popular culture.
- Event: Identifies if the movie was released during a seasonal event (Holiday - Nov/Dec or Summer - June, July, August) or not.

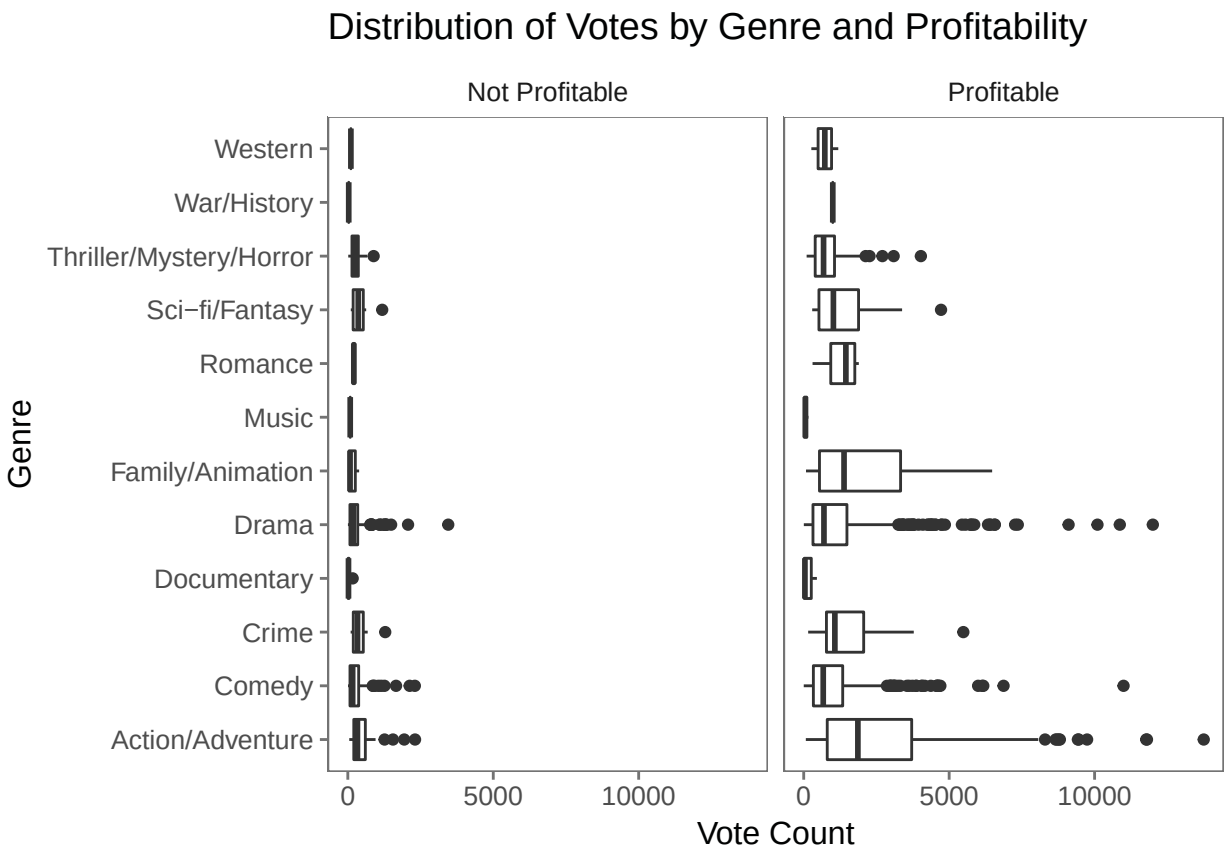
Further, I removed some columns from the data set that were not going to be used in analysis (e.g. keywords, production company).

Exploring the data

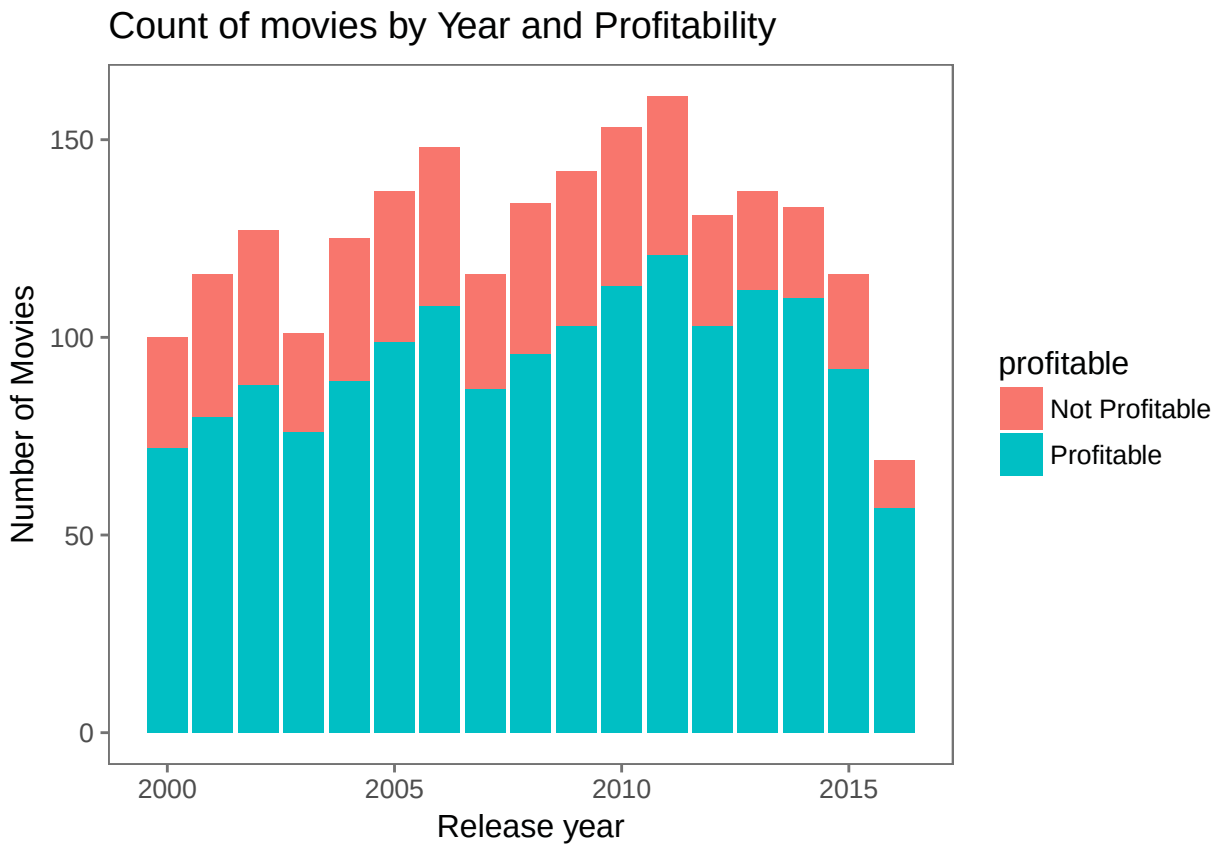
The final data set I am working with looks as follows:

title	budget	revenue	profit	profitable	profitable.ind	release_date	release_year	genre	runtime	vote_average	vote_count	event
Avatar	2.37e+08	2787965087	2550965087	Profitable	1	2009-12-10	2009	Action/Adventure	162	7.2	11800	holiday
Pirates of the Caribbean: At World's End	3.00e+08	961000000	661000000	Profitable	1	2007-05-19	2007	Action/Adventure	169	6.9	4500	na
Spectre	2.45e+08	880674609	635674609	Profitable	1	2015-10-26	2015	Action/Adventure	148	6.3	4466	na
The Dark Knight Rises	2.50e+08	1084939099	834939099	Profitable	1	2012-07-16	2012	Drama	165	7.6	9106	summer
John Carter	2.60e+08	284139100	24139100	Profitable	1	2012-03-07	2012	Action/Adventure	132	6.1	2124	na
Spider-Man 3	2.58e+08	890871626	632871626	Profitable	1	2007-05-01	2007	Action/Adventure	139	5.9	3576	na

The response variable is profitability, so let's take a look at how the data looks by genre based on the number of votes.



In the data set, we can also see how the number of movies and how the proportion of profitability varies over time.



Selecting a model

For my research question, I want to understand the relationship between the various predictor variables in my data set and my *binary* response variable: profitability. Because the response variable is binary, I'm going to be using **logistic regression**, otherwise known as a Generalized Linear Model with a binomial link function. My predicted value will then be a measure of the probability of profitability.

For simplicity and to establish a baseline, I started with the following model:

```
## glm(formula = profitable ~ budget, family = binomial(link = "logit"),
##     data = movies.pr)
##             coef.est coef.se
## (Intercept) 0.66     0.07
## budget      0.00     0.00
## ---
##   n = 2146, k = 2
##   residual deviance = 2353.7, null deviance = 2421.2 (difference = 67.5)
```

This model offered some improvement from the null model, but I have other potentially meaningful variables and transformations to consider.

Model Refinement

In my project proposal, I was planning on including both *budget* and *revenue* as predictor variables. However, when I applied this to the model, I received a warning indicating that the algorithm did not converge. I then used the vif() function to evaluate the col-linearity of the metrics, both of which had *huge* values when

included in the model. After some critical thinking, I realized that because profit is a calculation based on revenue and budget, that these perfectly explained the outcome. As a result, the models I fit included either budget *or* revenue, but not both.

The other variables I have to consider contain the following:

- Release date
- Release year
- Genre
- Run time
- Vote average
- Vote count
- Event

After evaluating several models using BIC, I found that *genre*, *event*, and *release year* were not improving model fit and as a result were removed from the model. From an inference perspective, it could make sense to include these predictors to understand how the data varies across the factor levels.

Transformations Considered

Upon visual inspection of my data set, I noticed that revenue and budget did not appear to be normally distributed so I applied a log transformation to see if that improved fit. Surprisingly, the BIC was higher in the log transformed models than the non-transformed models, so I ended up *not* using the log transform for budget, but it did make a slight improvement for the revenue model so it was kept there.

For interpret-ability of coefficients, I centered and scaled the following variables:

- Vote count
- Run time (I had also evaluated a log transform on this variable due to non-normality but it did not improve model BIC)
- Vote average

Because these are linear transformations, they will not impact model fit.

Regarding outliers, I had previously trimmed my data set for especially low profits or budgets (less than 10,000 dollars).

Final Model

I arrived at this final model by adding in each of the predictor variables listed above and evaluating the BIC. I'm using BIC because I want to penalize not only for the number of coefficients estimated, but also the number of observations. I'm able to use BIC as a measure to compare the model performance because I'm not altering the response variable or link function (all use the binomial link function).

```
##
## Call:
## glm(formula = profitable.ind ~ log(revenue) + runtime.z + vote_average.z +
##     vote_count.z, family = binomial(link = "logit"), data = movies.pr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0770  -0.0077   0.2071   0.4723   3.2561
```

```
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -23.01044    1.44701 -15.902   <2e-16 ***
## log(revenue)     1.40996    0.08185  17.227   <2e-16 ***
## runtime.z       -0.96133    0.09902  -9.708   <2e-16 ***
## vote_average.z   0.81335    0.08864   9.176   <2e-16 ***
## vote_count.z     0.61698    0.24668   2.501   0.0124 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2421.2  on 2145  degrees of freedom
## Residual deviance: 1283.4  on 2141  degrees of freedom
## AIC: 1293.4
##
## Number of Fisher Scoring iterations: 7
```

Model Interpretation

- Intercept: The probability that a movie is profitable assuming log revenue is 0, average run time, average vote score, and average number of votes is 0%. This probability is small because revenue of 1 unlikely to occur in the data set, and also very unlikely to be profitable if it did exist.
- Log Revenue: For every 1 unit increase in log revenue, the log odds of profitability increases by 1.41. Calculating at the mean log revenue (controlling for all other variables), the probability it is profitable is 88.02%.
- Run time (centered around mean and scaled by 1 std dev): If the run time increases by 1 standard deviation, the log odds of profitability decreases by -0.9613263.
- Vote average (centered around mean and scaled by 1 std dev): If the vote average increases by 1 standard deviation, the log odds of profitability increases by 0.8133531.
- Vote count (centered around mean and scaled by 1 std dev): If the vote count increases by 1 standard deviation, the log odds of profitability increases by 0.6169778.

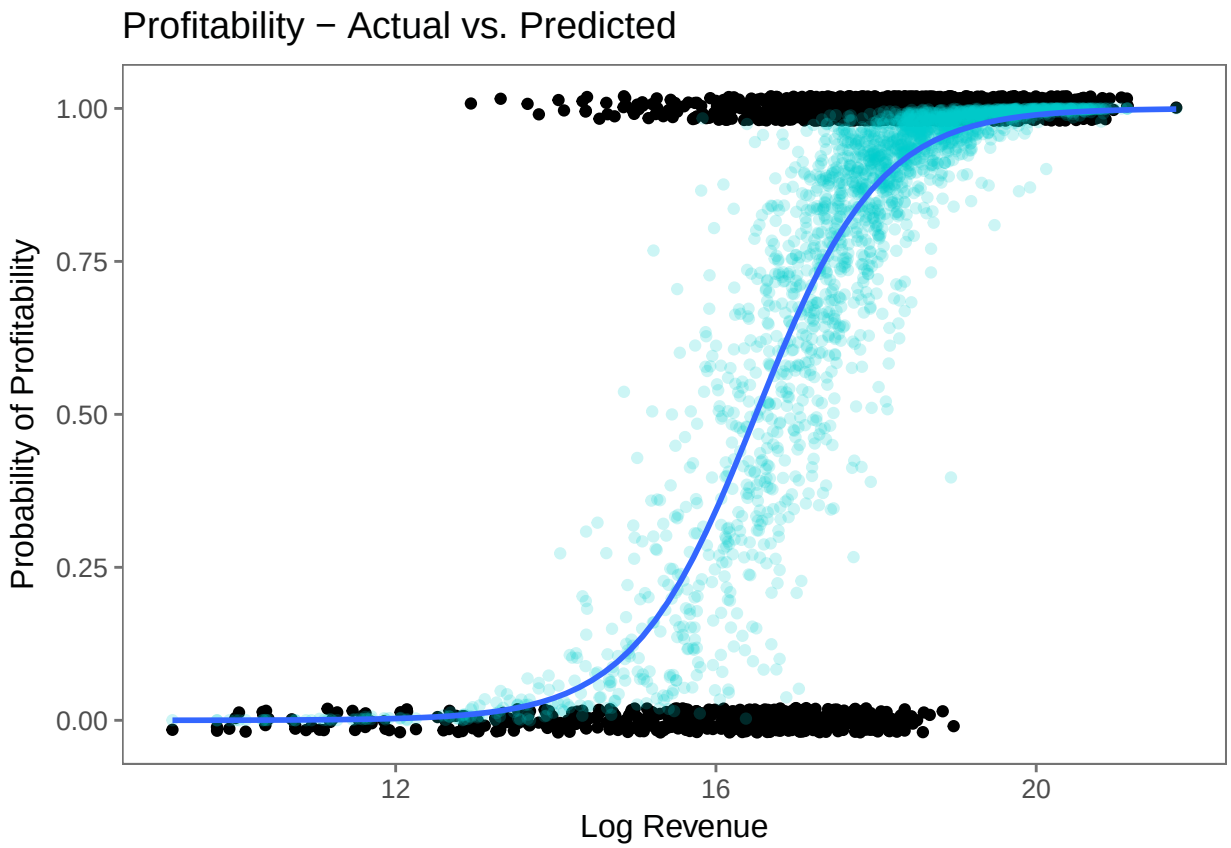
Model Evaluation

Table 1: BIC output of Initial model vs. Final model

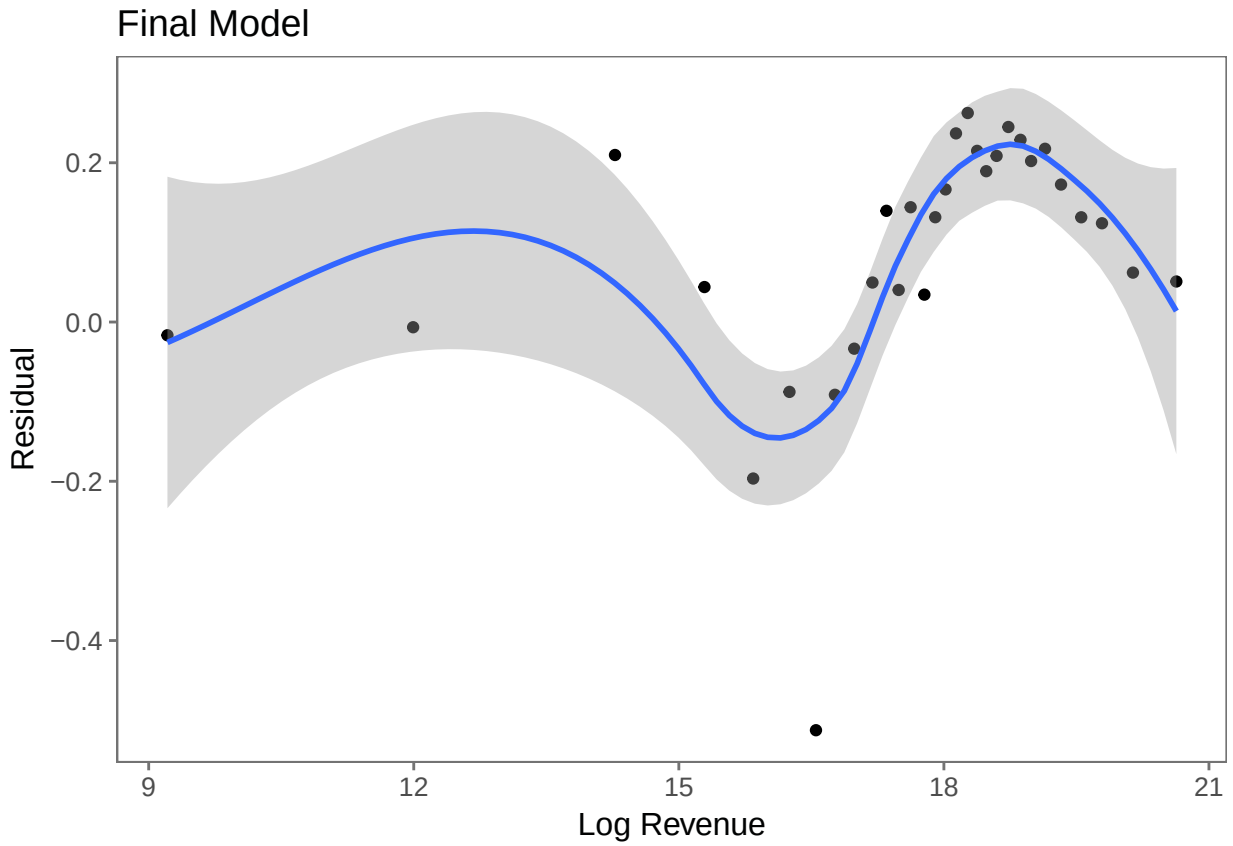
	df	BIC
mod1	2	2369.062
final.mod	5	1321.803

The BIC of the final model is considerably lower than the initial model fit. The threshold for *strong* evidence of improved fit is a difference in BIC of at least 10, and it is clear that these results show an improvement of 1047.

Below is a plot that makes use of the geom_smooth() function given one predictor, log revenue. This is not the final model I ended up with, but is helpful to visualize the probability compared the observed outcome. The blue dots are the fitted probabilities based on the final model.



To evaluate the model, I want to see how the residuals look. Since the response variable is binary and the predicted value is a probability, to get a reasonable picture of the residuals I want to create bins to look at the average residual across a number of groups.



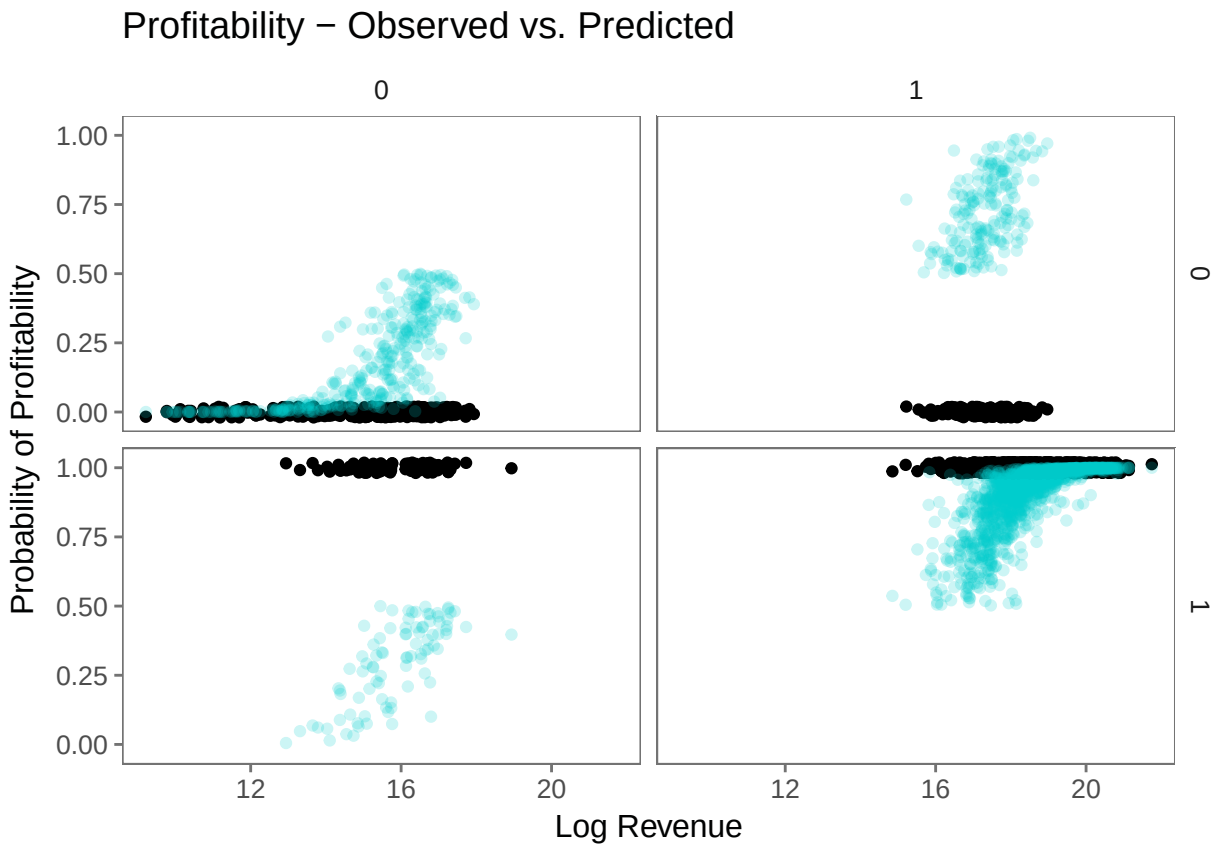
Based on this residual plot, it appears there may be some underlying pattern that is not being addressed in the model. Generally, the residuals fall within -0.2 and -0.2.

Another method of evaluating the model is to compare the predicted values to the observed values. Since the predictions are in terms of probability, I am rounding to arrive at whether the model predicted the movie to be profitable or not (e.g. 60% probability of being profitable would be rounded to a probability indicator of 1).

The below table and plot summarize the comparison.

Table 2: Confusion Matrix - Observed (row) vs. Predicted (column)

	0	1
0	350	190
1	89	1517



Based on the plot and table above, the model accurately predicted the profitability outcome for 1867 movies. The model was more likely to predict that a movie was profitable when it wasn't.

Conclusions

The conclusions I draw from the model are the following:

- Movies with higher revenue are more likely to be profitable.
- Movies that are significantly longer than average have a decreased profitability of being profitable.
- Movies with significantly more votes than average have an increased probability of being profitable.
- Movies with a significantly higher rating (vote average) than average have an increase probability of being profitable.
- Genre and event did not appear to be significant predictors of profitability.
- Pattern of average residuals indicate there may be some variable not being addressed.

These conclusions are fairly intuitive, although the effect of run time was less obvious to me.

Potential Next Steps

Cleaning the data set to extract genre was significantly more complicated than I had initially anticipated and planned for, especially since it was my first time working the JSON data. If I had more time, I would extent this project in the following ways:

- Gain a better understanding of how the sample of 5,000 movies was generated.

- Cast genre as a random effect using mixed level modeling and evaluate model performance.
- Develop a more precise method of determining "event". In the interest of time, I set this up at the month level. If I were to be more thorough, I would find specific movie event dates for each year (e.g. Thanksgiving, Christmas, Valentine's Day) and also do more research on events that tend to be large movie releases.
- Better diagnose the pattern in residuals from the final model.
- Identify a way to incorporate both budget and revenue in the model while still having the model converge. I'm not sure if this is possible statistically, but it makes sense that these would *both* be relevant predictors.

Shiny app

An interactive way to explore the data set and evaluated models can be viewed by running the following code:

shiny::runGist("3f70ad037e1960c61d9200ef70582bbe")