Group 8 DSCI445 Paper DRAFT

Group 8
Kelsey Britton · James Chinnery · Robin Thrush · Kaitlynn Walston

```
## Warning: package 'broom' was built under R version 4.5.2

## Warning: package 'ggplot2' was built under R version 4.5.2

## Warning: package 'glmnet' was built under R version 4.5.2

## Warning: package 'tree' was built under R version 4.5.2

## Warning: package 'randomForest' was built under R version 4.5.2

## Warning: package 'caret' was built under R version 4.5.2

## Warning: package 'rsample' was built under R version 4.5.2

## Warning: package 'tidymodels' was built under R version 4.5.2

## Warning: package 'dials' was built under R version 4.5.2

## Warning: package 'infer' was built under R version 4.5.2

## Warning: package 'modeldata' was built under R version 4.5.2

## Warning: package 'parsnip' was built under R version 4.5.2

## Warning: package 'recipes' was built under R version 4.5.2

## Warning: package 'tailor' was built under R version 4.5.2

## Warning: package 'tune' was built under R version 4.5.2

## Warning: package 'workflows' was built under R version 4.5.2

## Warning: package 'workflowsets' was built under R version 4.5.2

## Warning: package 'yardstick' was built under R version 4.5.2
```

Introduction

Our group chose to work with movie data, and the goal of this project was to predict a movie's gross revenue and identify which predictors are most influential in determining box office performance. The dataset was pulled from IMDb and includes movies released between 1986 and 2016. Some variables included in this dataset are budget, star actor, director, rating, and genre. To make the data and interpretation more digestible, each of us first worked with three predictor variables and saw how they each affected the gross of the movie. Then, we came together and discussed the results and what changes we had to make to the data to build a model. We combined the ways we changed the data to have a final cleaned dataframe. We then each took on a different method that would inform us of which predictors are the most influential to predicting a movie's gross revenue using that final dataframe and discussed the results we got.

Methods

Kelsey:

The variables that I cleaned were release date and movie title. The original release date column had multiple inconsistent format types, so I extracted just the release month as a numerical value. I also used this value to assign each entry a release season (Winter, Spring, Summer, or Fall), which captures seasonal box office patterns while avoiding messy date formats.

For the movie's title, I used sentiment analysis to assign each entry a sentiment score. This was done by combining AFINN-based word sentiment with additional scoring adjustments to account for punctuation, allowing titles with "?" and "!" to reflect a stronger emotional tone in either direction. The final sentiment score (sentiment_best) was kept as a numeric predictor in the combined dataset.

James:

Budget was the main column I worked on. Since over a third of the data was missing I chose to flag it and fill it with the median value for that year. In theory this should allow the models to still use the data from those entries and simply trust the budget less. For the runtime of the movies it was numeric and well behaved so it was left alone. The writers were initially lumped into 5 categories. This was somewhat arbitrary though and a better approach was to lump into two categories since so many writers only have one movie then this threshold could be tuned though I this was changed for some of the methods as they were better equipped to handle target encoded data

Robin:

The variables I worked with were director, company, and country. They are all characters, and unfortunately too unique to rely on the standard dummy encoding. To navigate this, I decided to assign countries to regions, and allow for regions to be dummy encoded because there were way less unique values. When doing this the function automatically made "Africa" the baseline region which caused the other regions to have an unusual standard for the regions and their influence on gross. Africa only had 10 movies associated with it, and all the movies had relatively high gross, and so to fix this I changed the baseline region to be North America which had 5040 movies associated with it, and a much wider range for gross. For the director and company I decided to use frequency encoding which allowed for the data to be digestible for a linear regression model. The caveat with using frequency encoding is that if there are directors that have the same frequency, they will be treated identically, despite being associated with different movie revenues. I transformed the data using log1p() which was supposed to help with dealing with skew, especially with the variables that had an extremely large range (gross, budget, etc). I fit a full model with the director, company, and region. When log transforming data, you interpret the results as how the percent change in the variable relates to the percent change in gross. So for example a 1% increase in director frequency is associated with a 0.807% increase in gross with all other things held constant. The full model with these three variables revealed that director seemed to have more influence than the company or region. I also fit a residual vs fitted plot and QQ plot which revealed that non-constant variance and that the model seemed to be more consistent in predicting high-gross movies than mid to low gross movies. QQ plot mostly normal, left tail was slightly curved, so some non-normality was going on. Overall, okay model with just the three variables, but likely could be improved with the inclusion of other predictors.

Kaitlynn:

The variables I worked with were genre, rating, and stars. Since they were all categorical, I first started out by creating frequency tables to visually see how much each category occurs. After that I cleaned the data by using dummy coding to better represent the categorical variables. The goal, after cleaning the data was, to use the LASSO to help us identify which of the variables have the strongest influence on gross revenue while reducing the noise from the other, less impactful predictors. So, using the cleaned data, I created a LASSO model to try to predict the gross revenue as well as showed the results in a table in order to show the top impactful predictors.

Together:

All of these cleaning methods were combined into a single csv file for use in the modeling.

Results:

Kelsey: Pure Linear Regression

For the pure linear regression model, gross revenue was modeled on the log scale using log(gross+1) to reduce skewness and the influence of extreme blockbuster outliers. Several numeric predictors were also transformed using log(x+1) (budget, votes, runtime, company frequency, etc.) and categorical predictors (genre, rating, region, season) were included as factors.

The model explained about 59% of the variation in log-transformed gross revenue (R^2=0.594). Because RMSE on the log scale is not directly interpretable in dollars, predictions were also converted back to the original revenue scale using exp($\hat{y}$ - 1) to summarize error in revenue units. Overall, the model shows that gross revenue is strongly associated with audience engagement and industry reputation variables, and it provides an interpretable baseline for comparison to shrinkage and tree-based models.

Among the predictors, IMDb vote count and production company frequency (comp_freq) were consistently strong positive predictors, along with star popularity and runtime. Several genre and rating indicator variables were also significant relative to their baselines, indicating that some categories are associated with systematically higher or lower revenue after accounting for other predictors. In total, the model produced a sizeable set of statistically significant terms ($p < 0.05$), spanning popularity measures, budget/runtime, and multiple categorical indicators. Overall, the linear regression model identifies clear, interpretable relationships between movie attributes and gross income, even though more flexible models may achieve slightly better prediction error.

Kaitlynn: LASSO

For the LASSO model, the predictors that impacted the prediction of gross revenue was the movie stars, budget, and genre, having the movie stars the most impactful. The LASSO gave us a baseline predicted gross revenue, holding constant the other predictors, and that baseline was set at gross of $554 million dollars. We can see that the top coefficients were different popular movie stars. Some of the movie star examples that we could see in the model results were: Tom Hanks, Nicole Kidman, Sandra Bullock, and Johnny Depp. Actors like Tom Hanks and Sandra Bullock earned about $28.45 and $23 million over the baseline respectively, however we can also see actors like Johnny Depp and Nicole Kidman are predicted to earn about $16.4 and $27.2 million less than the baseline. Thus showing that depending on who stars in your movie would greatly impact the gross revenue of the movie. The budget_missing indicator also had a noticeable effect ont he gross revenue prediction, since it was able to note that the films without reported budgets tended to earn substantially less than those with reported budgets.

```
## # A tibble: 1 x 3
##   penalty mixture .config
##     <dbl>   <dbl> <chr>
## 1  0.0264   0.474 pre0_mod330_post0
```

Robin: Elastic Net

After group data cleaning, we had 1 comprehensible dataframe with the predictors we would need to run shrinkage and prediction modeling. After cleaning we were still left with 4 columns that were categorical variables: genre, rating, region, and season. To handle this I used dummy encoding, and the model would select the most informative reference categories. I also removed similar column predictors, for example we had director frequency, director count, and director popularity. I treated them as the same and decided to drop director frequency and director count and only used director popularity in my model. Other variables that were handled similarly included writer and star. The remaining predictors were numerical and I log transformed them using log(x +1) to address skewness and handle zero values. To build the model I utilized a bunch of R packages like recipe, workflow, and tune to standardize variables, select the lambda and alpha values, etc. The data was split using 10-fold cross validation, splitting the data 90-10 for training and testing.

A grid of 400 different combinations for lambda (regularization) and alpha (mixture) was created and then the metrics rmse, mae, and rsq for each combination was calculated. The group decided to measure the model's performance using rmse. I was able to pull the associated lambda and alpha which were 0.0263 and 0.4737 respectively. The lambda is small, which indicates weak regularization. The elastic net model could be retaining too many predictors, causing potential overfit. The alpha was almost 0.5 which would have been a perfect mix of the LASSO and ridge penalties.

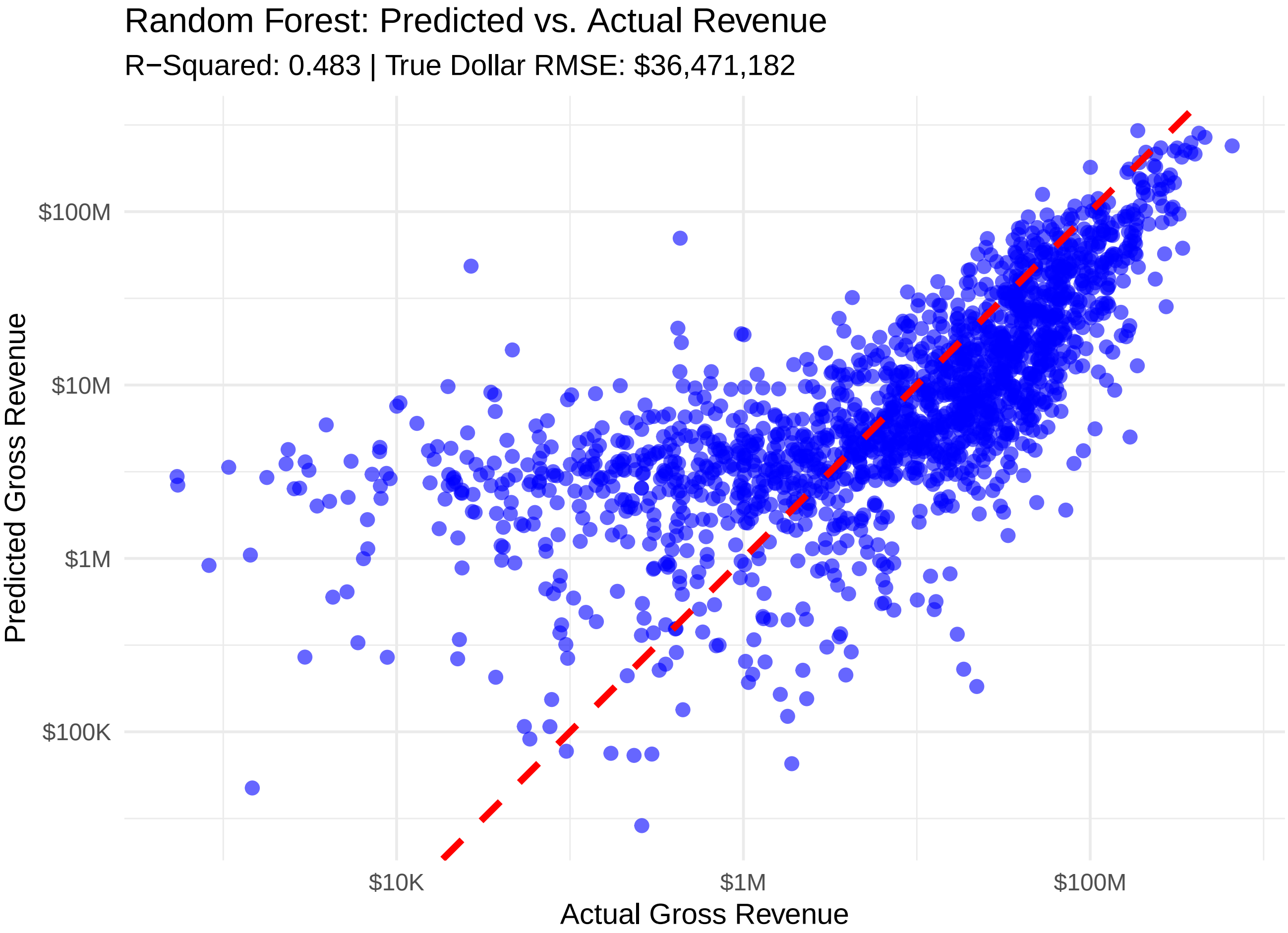
Then the “best” elastic net model was fit using the values we found for lambda and alpha. A prediction model was made inspired by the final fit. For comparison to see how the predict model performed we measured up the metrics we collected earlier rmse, mae, and rsq. From these findings we were able to tell that the predictive model performed well. The rmse value from the training and the test rmse were just about the same, meaning that the model generalized well (38.5 million and 37.7 million respectively) The training mae and the test mae are a bit further apart in value, but they are similar enough, which just corroborates the conclusion that the model generalized well (19.3 million and 18.4 million respectively).

Lastly, to identify the most influential predictors, I isolated the predictors and their coefficients. Filtered out the ones that were dropped to zero, and ordered the ones that were leftover by their magnitude. This is because there were predictors that were associated with changes in achieving higher or lower gross. Though we want to know which predictors are the most influential in predicting higher-gross, it was interesting to see the ones that had a negative influence (like having an “unknown” rating). The elastic net model went reduce the original 53 predictors to 39. 14 were dropped to zero. The top three predictors that were the most influential for higher-gross were: votes, company frequency, and movies that were made in North America.

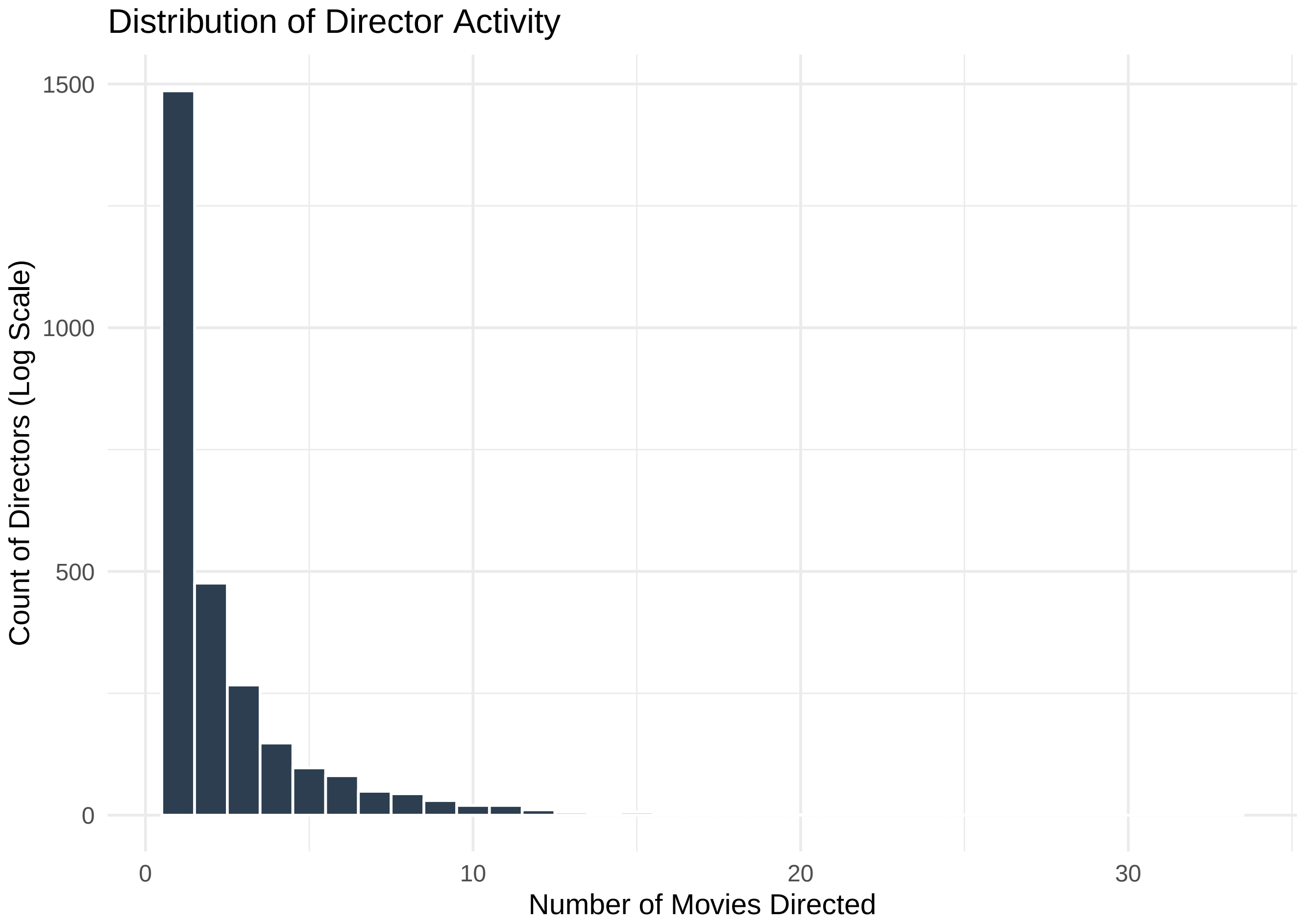
James: Tree Boosted/ Bagged Random

I approached the tree based models with a slightly different approach, since tree based models theoretically choose the ideal point to split the data at each node I wanted to give them the chance to choose this point rather than relying on the arbitrary splits we created with the frequency based binning approach outlined in the methods section. To this end I initially fit the tree based models using target encoded data. This kept the dimensional low and hopefully added some extra information for the ensemble models to pick up on. This did risk over fitting because the target encoding can cause problems when there are a low number of samples making the target value just the value. Despite this I thought the trees would be resistant to this for a few reasons. The package used to do the target encoding has smoothing factor to trust the means of poorly represented features less. In addition to that in theory the ensemble based methods should be highly resistant to over fitting in this manner by nature of being the average of many tree. Finally all hyper parameters were tuned using 5 fold CV. 5 Folds were chosen for training speed since these models ran somewhat slowly. I hoped the combination of these techniques would reduce over fitting to an acceptable level.

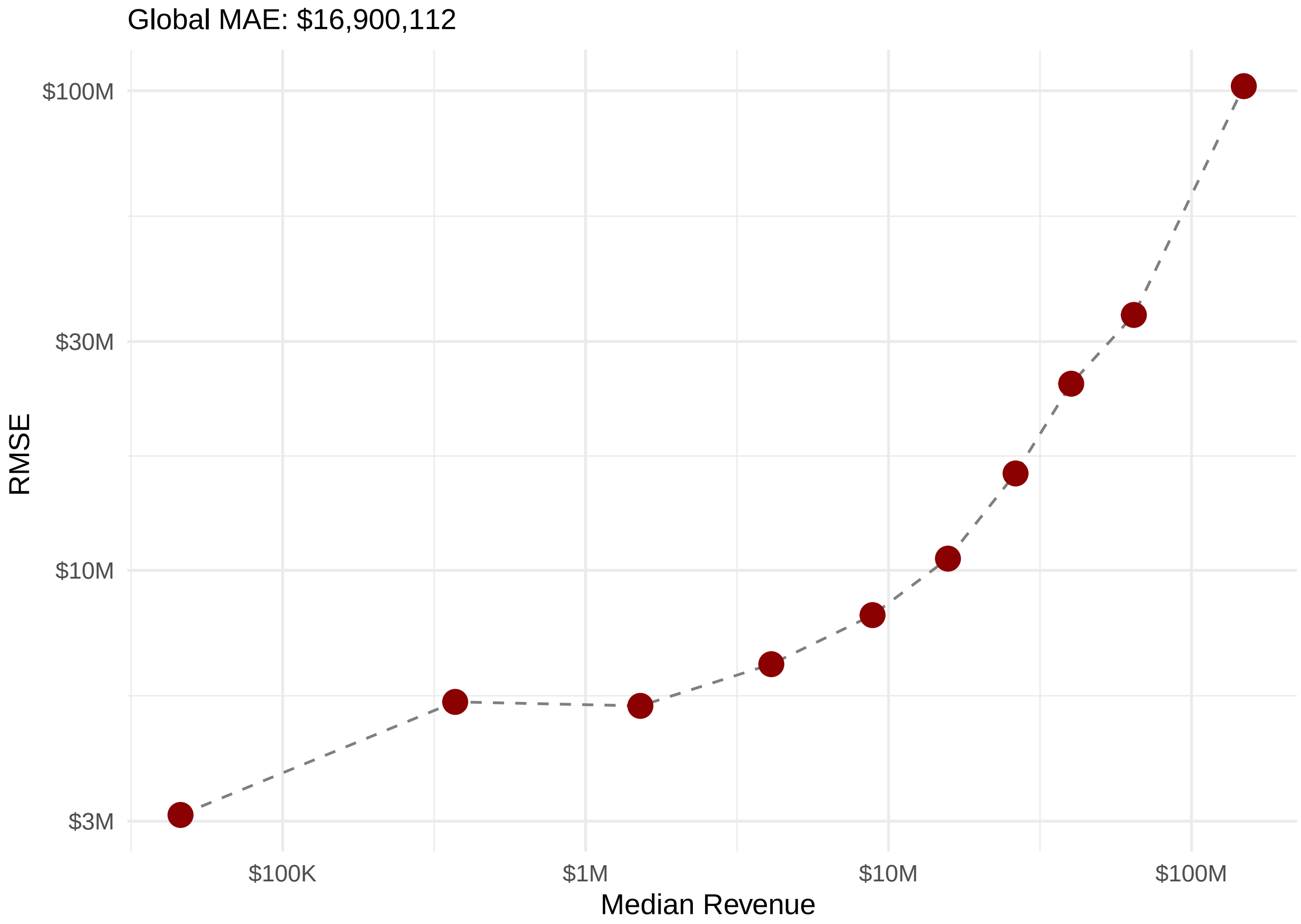
Using this method I fit both a random forest and a boosted tree. The Random Forest was tuned for the minimum data points required for a node split as well as the number of predictors sampled for the sub trees. Similarly the boosted tree was also tuned on the minimum points for a node split, in addition to both depth and learn rate.Both the Forest and Boosted Tree were tuned on 100 trees. This number was fixed for speed and only used for tuning. The final model was fit with 1000 trees, I decided to push it high for the final fit as these models don’t really change if there are too many trees instead just running slowly this was acceptable for the final model given it was only a single model being fit. The final model was only fit on the random forest since the preliminary results showed both trees being extremely similar with a CV error of 1.81 and 1.80 (log scale) for the Random forest and boosted tree respectively. The full Random Forest did not improve this meaningfully, this result showed the tree based models running notably worse that the linear regression based models. My hypothesis as to why was that the presence of of functionally unique data points made the data too noisy. The predicted vs actual plot echos this sentiment. In addition to that you can also see somewhat of a curved trend line in the cloud of predictions pointing towards this model poorly representing our data.



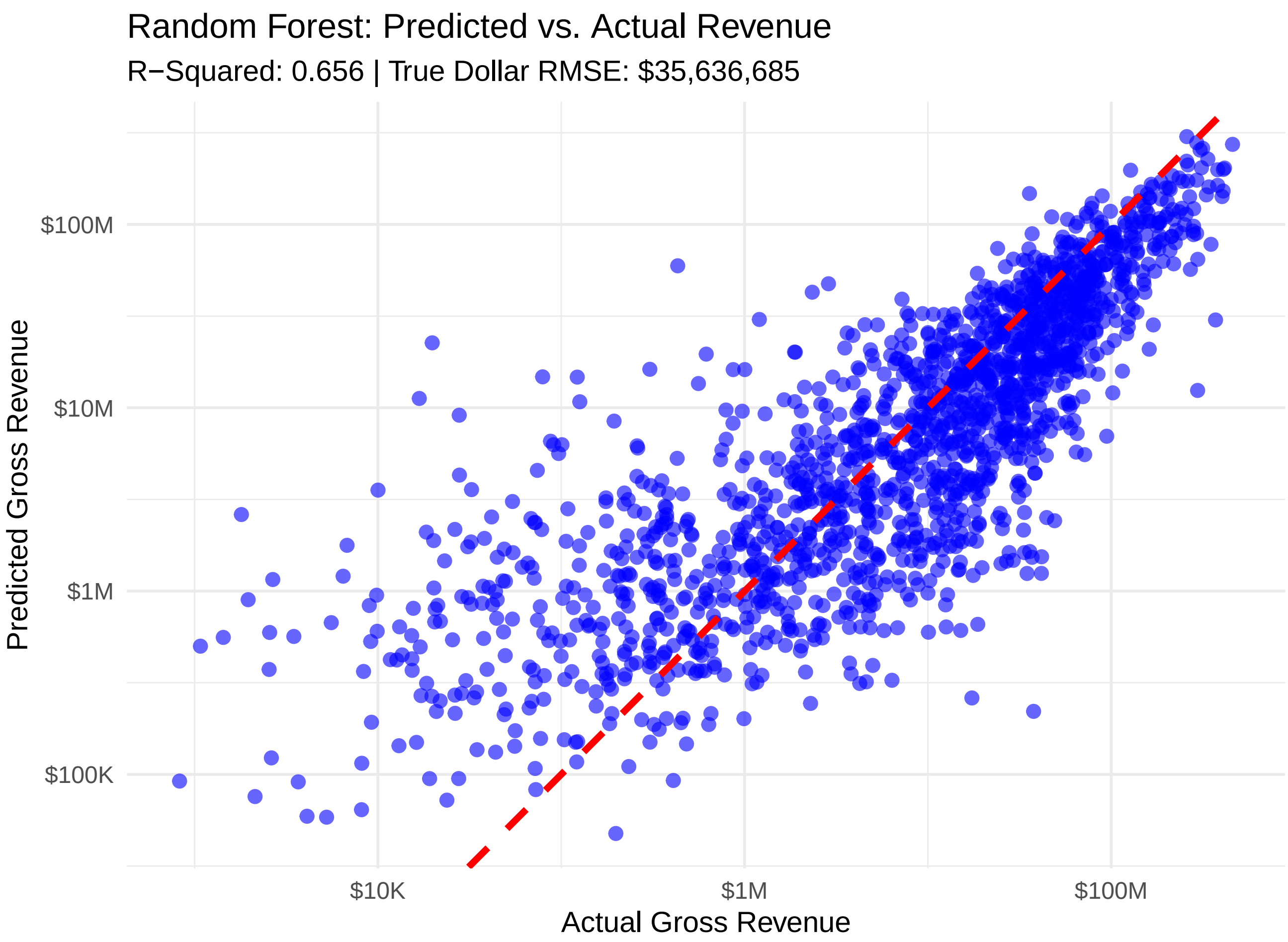
To test this I tried to encode the data in the lowest variance way possible to see if that would improve the model. To this end, rather than my initial approach of target encoding to give the models the most information I instead transformed the high cardinality data into a simple binary of if they were present for more than 3 movies this was chosen according to the histogram below and under the assumption that variables like writer and star followed a simliar distribution. Further experimentation could be done to find the ideal cutoff for each variable with CV. I did not do this as the results were not indicative of major potential gains.



Fitting a Random forest on this was much better, but still left a lot of room to improve. The tuning was the same as the prior with 5 fold CV where an RMSE of 1.46 ($35.6M) was achieved along with an Rˆ2 of .65, the highest any of our models achieved. Rˆ2 is a slightly better metric here as we know the mean heavy statistics will be skewed by the massive hits that make in excess of 100M as such we know the RMSE will be inflate. As such looking at the MAE may be more beneficial as its less sensitive to the outliers. Globally on this model once adjusted back to dollars its 16.9M which is very poor still given this is larger than the median gross a movie makes. Furthermore as seen in the plot below our error is just globally poor especially near the outliers at the top rather than making good predictions for the low gross movies and struggling at the top end.This can be seen in the below plot which compares RMSE to budget.

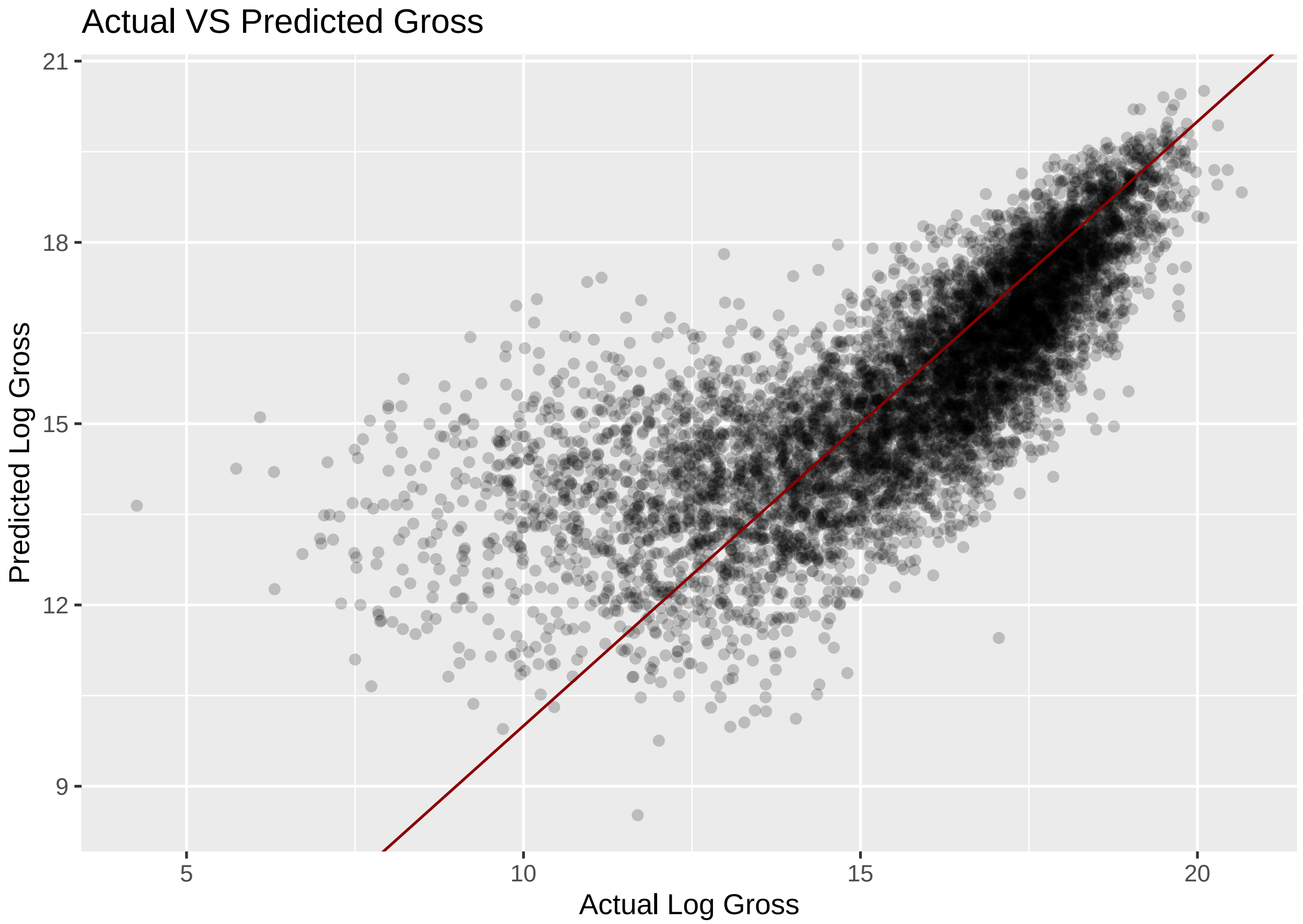


This better, but still bad behavior can also be seen in the fitted vs residuals plot. Note how relative the the higher variance version, of the random forsest the group in is tighter and the concerning curved trend is lesser as well

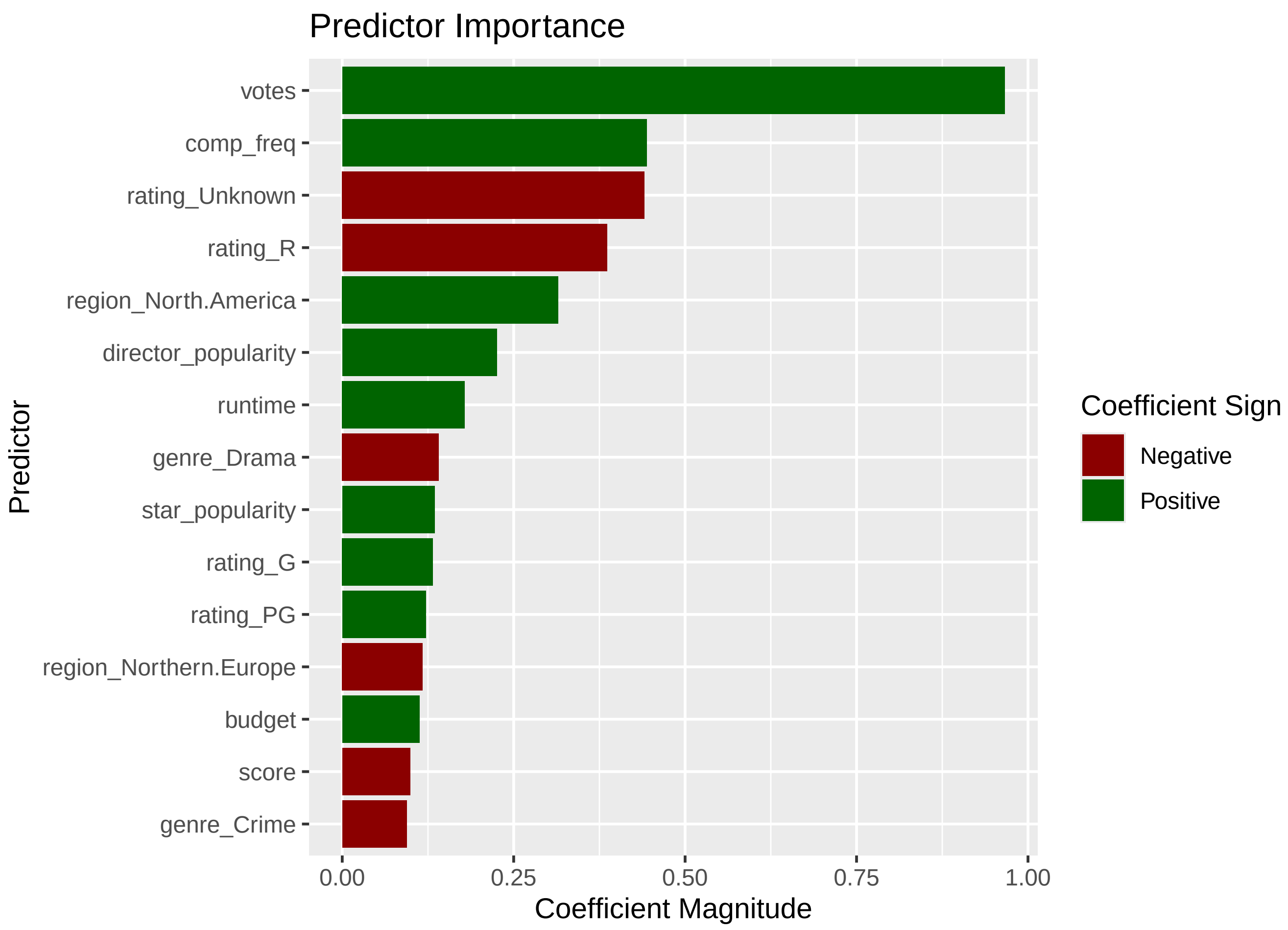


Figures

The figure below is the actual versus the predicted plot of the elastic net model. We can see that it is more accurate in predicting higher gross films.



The barplot below shows how influential each predictor was for films gross. This visual is useful for justifying our final claim that audience engagement and industry reputation are most important for high gross films.



Conclusion/Changes if we did it again:

Across all modeling approaches a consistent set of predictors were revealed as the most influential in determining gross. This would be votes, company frequency, director popularity, and star popularity. In terms of performance, we used RMSE as our measure of comparing models. Linear Regression came in at the highest RMSE, and following was Tree-based models, LASSO, and Elastic Net. Their respective RMSEs being $–, $–, $–, and $37.7 million. Notably, despite using a variety of different approaches the RMSE's and Rˆ2 values all clustered around similar values. This points to the data being the main culprit behind the relatively poor results of the models. This makes sense in practice because on account of how many people may only show up on the list once in six thousand entries leading to points where theres really no trend to learn as there's only one example. The end result is data where the true trends have a very high amount of extraneous information that essentially amounts to noise. The only fix here really is more data which given the finite number of movies isn't very feasible.

Despite the different approaches we took, the models agreed on the most influential predictors. We can generalize the strongest predictors as: audience engagement and industry reputation. For real world application we could say that for films to be financially successful they should heavily advertise, cast popular actors, and have known directors direct the film. This result isn't particularly surprising through notable it should be mentioned how how of the most impact predictors are really just side effects of raising the budget. Getting a household name in a given movie isn't cheap in most cases. That said we didn't see this correlation with budget directly, leading to the conclusion that although the best way to make a hit movie is expensive, money alone is not enough. It gives insight into how film makers should allocate their budget to ensure box office success.

If we were to continue this work there are some notable gaps that could be addressed. Notably our dataset lacked data on marketing effort as well as how many theaters a movie played in. This data could potentially help distinguish between films that just couldn't be a massive commercial success because they outright lacked the potential audience. In addition to that getting more data on the budget of movies could substantially help our model. Movies with a budget of 0 do exist and by employing a blanket policy of just setting them all to the median there was some data lost that may have been useful in predicting lower performing films. Finally, though difficult theres lots of potential in finding a way to quantify to encode the plot of the movie. Though we had features that helped quantify how good a movie actually was, there was no way to tell if the plot of a movie had mainstream appear perhaps sentifment analyis of a summary of the movie could provide some of this insight.