Springboard School of Data - Data Science Career Track
Capstone Project 2
Modeling Film Revenue
By: Nicholas Dean
June, 2021

1. Introduction

Film revenue varies wildly from film to film, representing an enormous profit for successful films and potentially huge losses. There are thousands of components that comprise a film, and each can affect the revenue of a film: genre, the other companies which work on a given film, even a film's tagline can affect consumer's willingness to spend money on it for entertainment. The purpose of this project is to model film revenue based on the various qualities of different films available via The Movie Database (TMDB) and the Internet Movie Database (IMDB), with the goal of predicting film revenue within 20% on average, at most.

I could approach this as a classification problem and attempt to classify films as 'sound investments', however that approach leaves much to be desired. The number of features that comprise a film is massive and nuanced, simply classifying a film as a good investment based on its *current* features ignores all of the domain knowledge and expertise stakeholders bring to the table. Consider two films: Film A is classified as a success, but is projected to barely turn a profit as a genre-bending indie film that could become a cult classic or simply fade from public memory after it's release. Film B is also classified as a success, but is projected to turn a healthy profit as an action/adventure film.

Film A will still prove to be a decent investment, but presents little opportunity to build on what is there already. It would be of interest for a studio who specializes in indie films since they often don't turn a profit, but likely not for a larger production company. On the other hand, Film B presents a fantastic opportunity for studios to invest in. It is already projected to have a healthy profit margin which can potentially be built on by a studio which specializes in post-production and can improve the marketing for a film, or a studio who wants to expand its content and has budget and expertise to juice up special effects.

Thus, this project attempts to resolve this uncertainty and actually predict the financial outcome as a function of these ingredients; allowing stakeholders to pick large and small investments for their companies. The stakeholders in this case are studio executives and analysts who are considering their options to invest their company's resources into. The problem of picking winners & losers in the film industry, if solved, could provide them with a powerful tool to avoid massive losses on films.

Treating this as a regression problem rather than a classification problem also gives the stakeholders additional insight into what makes a film 'successful' in a given arena, and introduces additional nuance into the kinds of investments they might make. These 'investments' fall into two categories generally speaking: (1) Scripts in need of full development which are predicted to have significant revenue, (2) Films which have strong foundations upon which marketing or other expertise can be contributed in order to boost revenue further and earn a portion of the increased returns. Further, this allows stakeholders the ability to enter the known features of a film or script and alter other factors they control like budget, cast, runtime, or marketing in order to assess how changes might affect projected revenue.

2. Summary of Results:

Unfortunately the problem proved to be too complex and Linear Regression was not a good option, which means that mathematical coefficients indicating how a change in a film's feature will alter that film's revenue. GradientBoosting regression models provided the best performance with regard to mean absolute percentage error (MAPE = $\frac{1}{n}\sum\frac{|T-P|}{T}\times 100$). Accuracy of predicted revenue as defined by MAPE is the most important factor for this problem, however decision tree based algorithms like this one also allow us to examine which features are the *most important* in terms of determining the revenue of a given film.

The models tested are able to film revenue when considering ALL films to within 293%, when considering specific seasonal releases to within 111%, and when considering specific categories of films to within 36% using a gradient boosting algorithm. In the worst-case scenario (predicting revenue using a model built on every film available with 293% MAPE), 95% of all predictions fall $150m above the predicted revenue or $63m below the predicted revenue. This effectively raises the floor for a 'good investment' for a company to join production on films with revenue projected to be at or above $63m. Modelling on narrower subsets of films improves this uncertainty significantly.

According to the importance of features as measured by information gain within a gradient boosting model budget and the contribution of a top-tier production company towards a film are the two biggest factors in determining revenue for all films examined. These two features are followed by qualities that a production company controls, such as runtime and various marketing indicators.

I also tried a tiered approach to modeling revenue based on smaller subsets of data with great success. Modelling film revenue for movies where Disney was credited enabled the models to predict revenue with a MAPE equal to 36%. However the company-specific approach did not yield the same results when applied to other production companies, when only examining films where Paramount was credited I was only able to reach a MAPE of 188% - worse than modeling based on seasonality. By modelling tiers based on three factors: genre, if a film is part of a larger collection (and thus has a built in audience), and seasonality, The models build report a MAPE of 64% across the films being examined.

This provides a solid roadmap for future work and how to reach the ideal of 20% error--or less. I identified two key issues with the work I've done that contribute to the large error I'm seeing here, which results in recommendations for future work to improve further on the tiered approach.

3. Approach

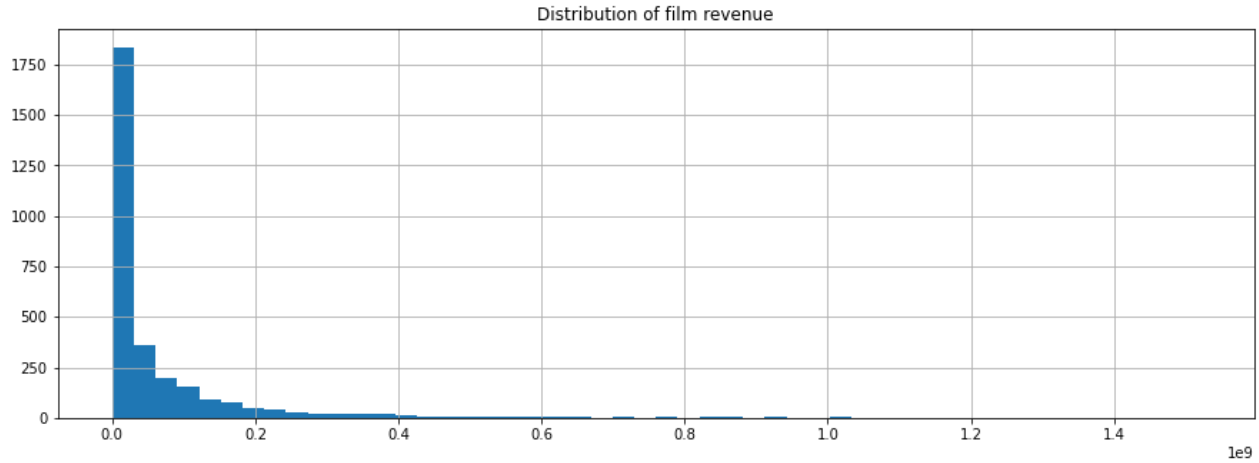
Data acquisition

The data set was initially pulled from Kaggle's TMDB film competition and is available as two separate components, for training and testing models. Film features were stored in a dictionary format and needed to be extracted programmatically in order to view the information in detail, and be able to manipulate the features themselves. Once this had been completed, each feature such as 'genre' contained a list of all of those properties which could apply to a film. For example, 'Hot Tub Time Machine' had a list that contained both the Sci-Fi and Comedy genres.

This offered a high degree of detailed information about each film, but also presented some challenges

The data I had access to was meant to be used to populate the front-end of TMDB's website with information about a film. The result of this was that there was a lot of hyperlink & visual information embedded in each of these categories, as well as marketing copy. While the scope of this project was to predict film revenue, and marketing efforts have a large impact on a film's success, leveraging NLP sentiment analysis for film taglines & plot overviews is beyond the scope of this project. Additionally, while posters and other visual marketing tools are also impactful on a film's success, image processing is not within the scope of this project. I made the decision to not use this information. I did include the length of both a film's tagline and overview as a measure of 'brevity' in marketing copy since a film's tagline and overview are both written *after* a film is produced and would be something that a production company has a degree of control over.

Before going any further I needed to examine the revenue of the films I had information on to ensure there were no missing values or significant errors. The first place to start this is to plot the distribution of film revenue:



This immediately raises some concerns about outliers, however I'm loath to remove those until I know that they'll cause problems with predicting revenue. I did need to examine the distribution of revenue close to zero and determine if they were small values or if they really were zero; upon closer inspection there are some highly suspicious values.

The Alec Baldwin film 'The Getaway' and Mandy Moore film "Chasing Liberty' were listed with a revenue of $30 and $12 respectively; not even enough money to purchase tickets for a family of 4. Ultimately 7% of films had revenue below $500, a highly improbable number that is less than 50 tickets priced at $10. I decided to spot check some revenue numbers against IMDB's revenue information, and found quite a few errors.

First there are films like 'The Getaway' and 'Chasing Liberty', whose revenue was listed as this on TMDB ($30million and $12million), resulting in some sort of error when exporting that information. Second there are true box office bombs, like 'Electra Luxx' whose revenue was listed as $10, but was actually $11,000. After manually checking revenue and finding accurate information on IMDB I chose to import revenue information from their API.

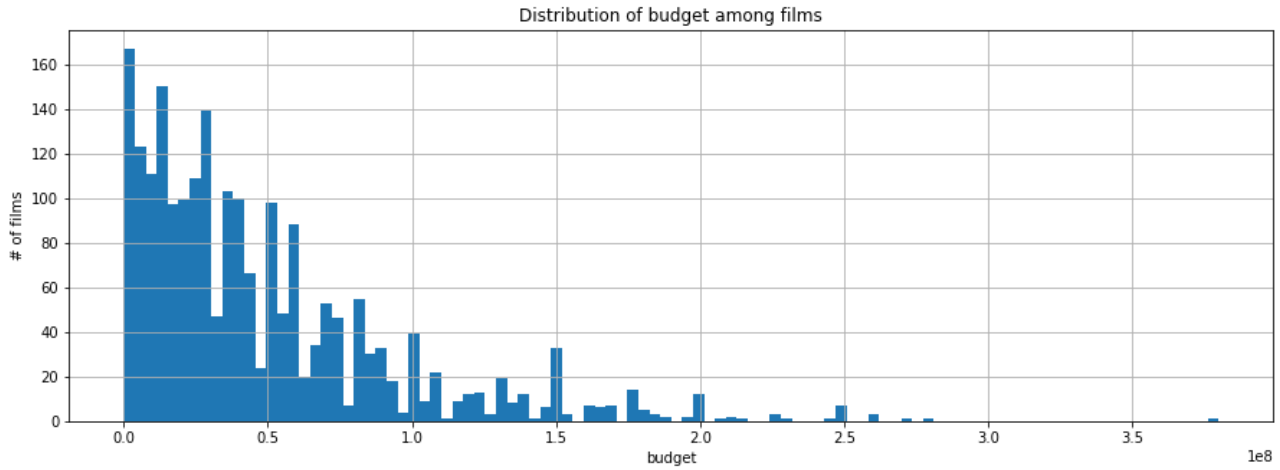
IMDB has two APIs, one is professionally maintained for studio use - I don't have access to this API. The other is an open source Python implementation of an API called IMDBpy. As a result of it's open source nature it took several days to install the API and understand it's data formatting,

but I was able to pull revenue information for most of the films I had data on. However IMDB doesn't have revenue information for all of the films in the TMDB data set, so I chose to combine the TMDB train/test data and keep all films for which IMDB had revenue information on.

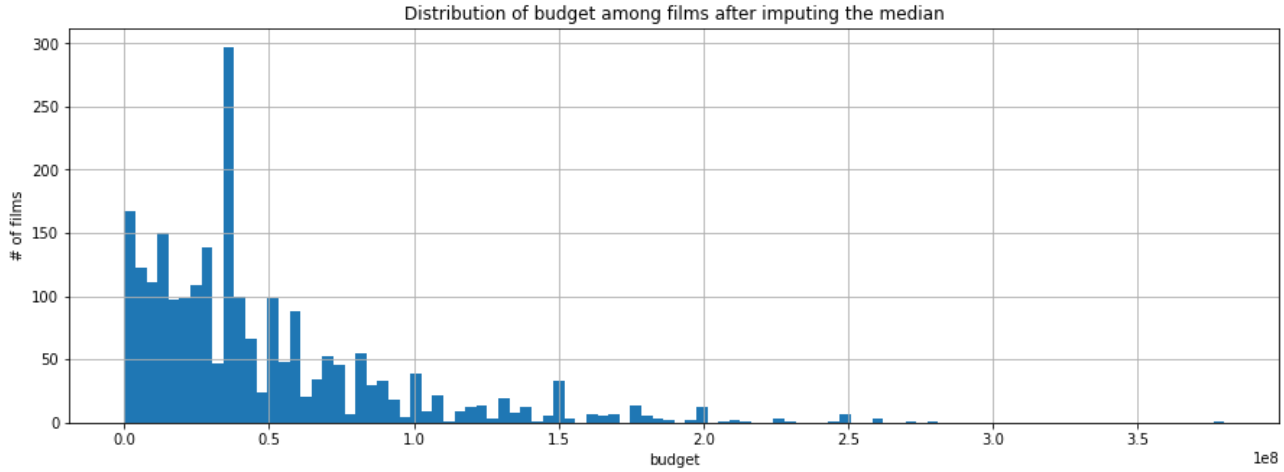
At this point I still had revenue data with an extremely long right tail, but I am confident that I'm working with accurate revenue data. Before making a decision on how to handle the new revenue data I wanted to examine the other features of the data set for missing values or odd distributions and assess how a regression model performed on the data without handling outliers.

Data Wrangling: Numerical Data

In this data set only 23% of the variables are non-categorical, so exploring the few non-categorical variables was a good place to start. The first of these was budget, and I wanted to see if the budget had a similarly skewed distribution to revenue:



Budget distribution seems to mirror revenue distribution, with a less skewed distribution and much smaller range of values. However, this excluded the 194 films for which no budget information was available, given the highly skewed nature of the film budgets I chose to impute the median film budget here ($35,000,000).



This resulted in a large change to the distribution of movie budgets, while the overall structure was preserved. I chose to impute the median for all other missing values in the non-categorical data for

this data set except for runtime for which I imputed the mean, which was the only variable with a normal distribution. There are quite a large number of outliers in this data set, but I chose to leave those until I had come to a decision on how to handle revenue's skewed distribution.

Data Wrangling: Categorical

After handling missing values and outliers in numerical feature variables I needed to decide how to handle the categorical data. All of the categorical variables, after extracting it from JSON format, were represented as a list of all options of a category that applied to a film.

For example, 'The Princes Diaries 2' had a list of these genres in it's genre column: Comedy, Drama, Family, Romance. While this allows the data frame to be compact, it will not work with machine learning models which require all data to be numerical. I needed to create new one-hot-encoded variables from the categorical information.

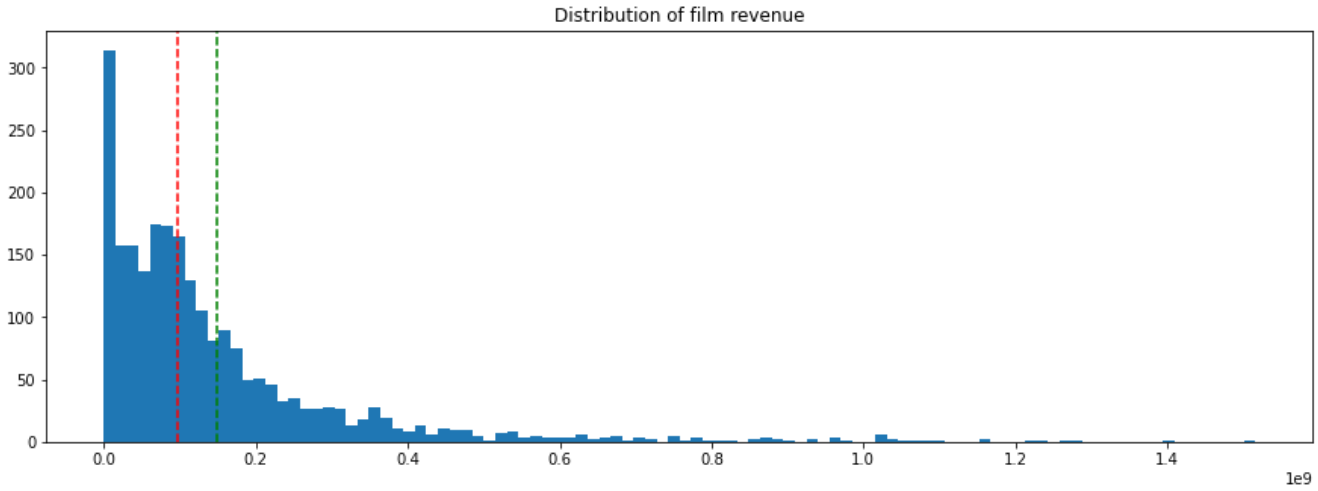
After writing a function to do this for all categorical features (Genre, Cast, Crew, Release Month, Release Year, Production Company, Production Country, Keywords, Original Language, and Spoken Language), I was left with over 10,000 columns. At this time the same film 'The Princess Diaries 2' would have a column for each possible genre. All of those new columns were equal to zero, except for the four genres that described that film (Comedy, Drama, Family, and Romance) which were equal to one.

The explosion in dimensionality is despite a problem that I had with creating dummy variables from the Cast & Crew categories. These two categories had vastly more available options, which proved to be far too much for my laptop to handle. At the time of this project I was travelling while studying, and didn't have access to enough RAM to create the dummy variables from those columns. I would later learn that 10,000 features was far too many dimensions for machine learning to accurately model regardless, and this is one of the mistakes that I'll discuss how to improve at a later point.

Finally after extracting the categorical information into one-hot-encoded variables I had a data frame that would function with a machine learning model, but was also highly sparse. Despite the risk of modelling a highly sparse data frame with so many dimensions, at the time I chose to retain all of the information and move forward.

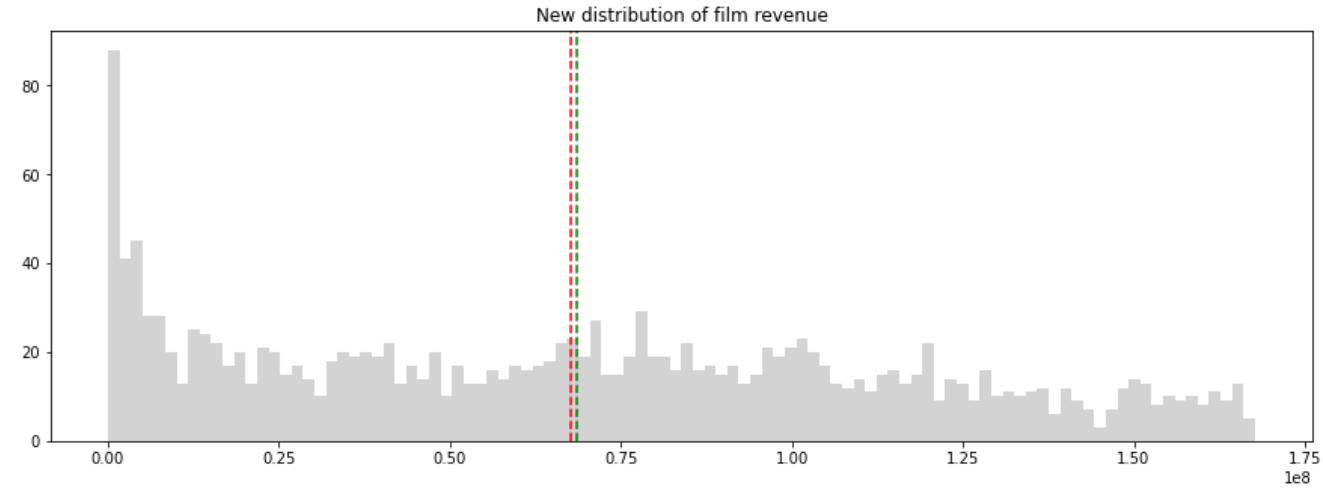
Data Storytelling: Basics of the data set

After getting it from IMDB I'm confident that the revenue numbers make sense and are accurate. However, the revenue for this data set was still highly skewed as you can see in the figure below. The green line denotes the mean revenue of movies in this data set, while the red represents the median revenue.



As you can see there are around 30 films with revenue above $800 million (0.8e9). I needed to remove outliers to prevent serious issues with modelling the data. After trial & error, and examining the relationship between budget (a company's primary financial investment) and revenue it was clear that there was no significant correlation between revenue outliers and any single other variable. I tried removing the top 5% of revenue observations (and the corresponding data on those films), and this resulted in no change to the overall distribution of the data.

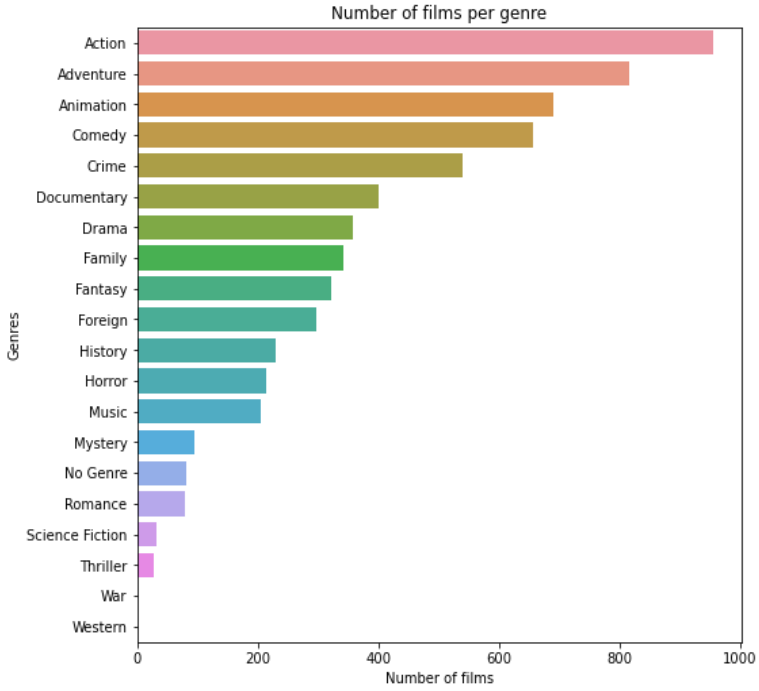
Later after some initial modeling with models reporting a large MAPE, I revisited the outlier problem and tried removing 10% and then 15% of outliers, only seeing an improvement in performance at the 15% mark. After removing the top 15% of films by revenue I was left with a revenue distribution that looked like this:



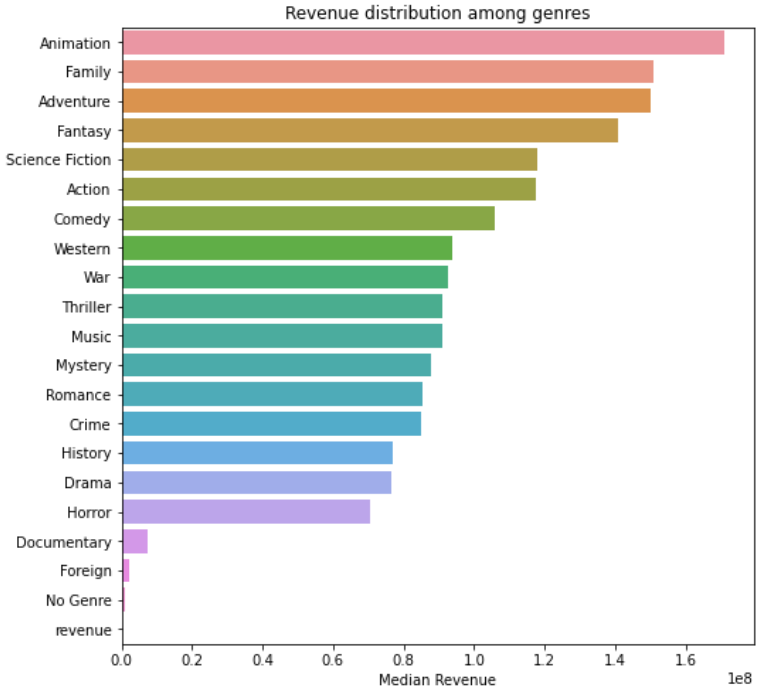
This was much more manageable, and as you can see the red (median) & green (median) lines are much closer to the center of the distribution.

After my target revenue was appropriately dealt with to remove extreme outliers I wanted to examine the categorical variables and how they interacted with specific films and the revenue variable if possible. I wrote a function that would allow me to count the number of films each

individual feature would appear in. Below you can see an example of how films were distributed amongst the genre category - one of only two categories to not exhibit a highly skewed exponential distribution:



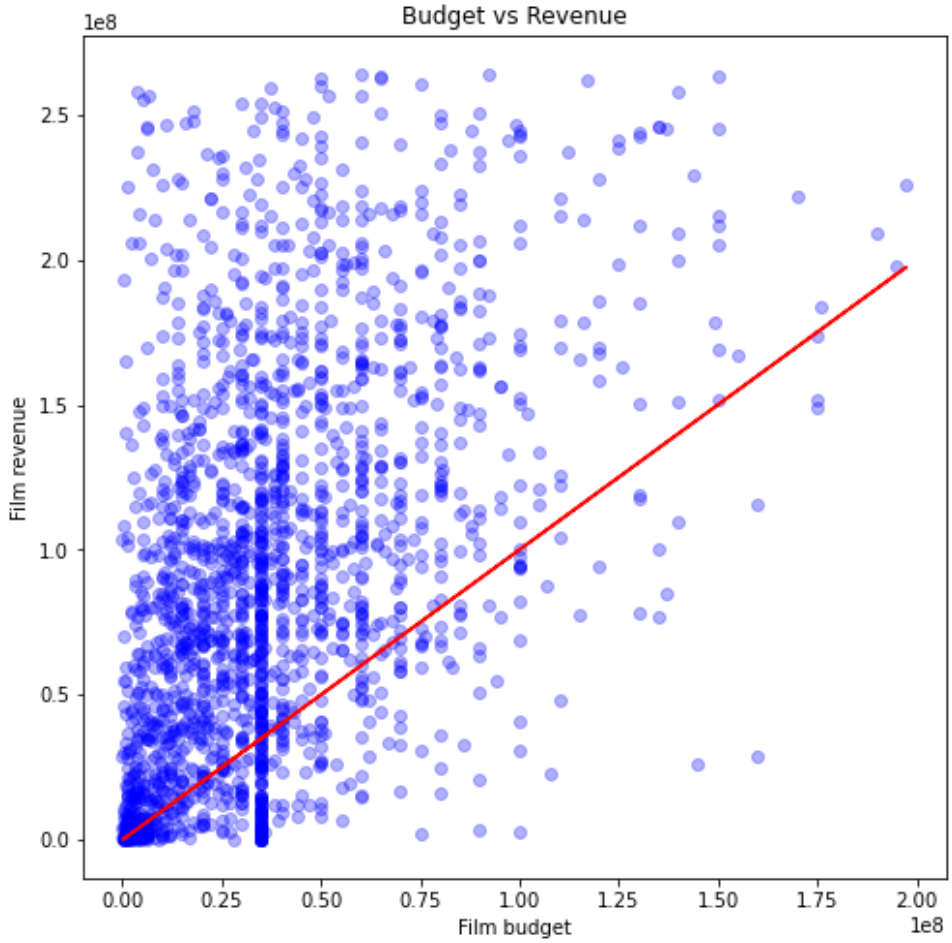
This same function would then plot a graph from an engineered data frame showing the distribution of revenue among those features, with an extra bar equal to zero for the revenue column in the engineered data frame. You can see the corresponding graph for genre below:



While extremely useful for Genre (and Release Month), this approach was only helpful for examining a features distribution among films and revenue since the large number of variables made plotting with any real labels impossible as you'll see shortly when I look at specific subsets of the data.

Data Storytelling: Relationships to Revenue

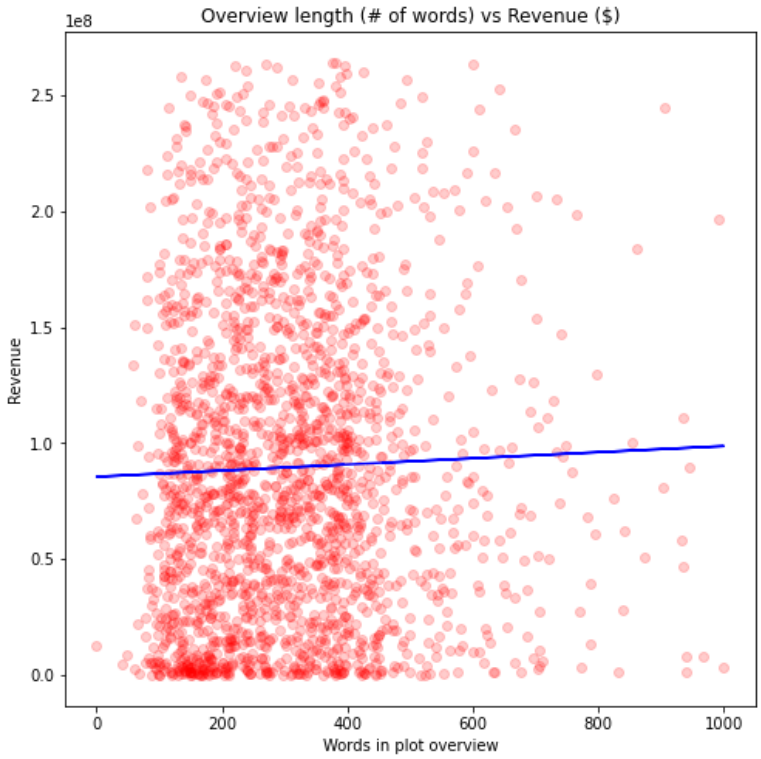
Before examining my approach to baseline modeling it will be valuable to examine more closely the relationship between several key features and revenue. Budget, as will later be shown, is one of the most important features when it comes to determining revenue; both budget and revenue possess extremely long right tails that leave a lot of room for outliers. This only leads to significant predictive power for both medium-low budget films, as shown in the graph below.



In this graph, the red line indicates the 'break-even' point for films where revenue is the same as budget. Below this line lie the 20% of films which represent losses for their production companies, and above are the remaining 80% of films. There does seem to be some correlation here, it's also immediately obvious that while budget is correlated with revenue it cannot describe revenue

outcomes on it's own. There are no films at all in this data set with a budget over $200m, but around 15% of films have revenue that exceeds $200m, and does so at all levels of budget.

Surprisingly 'Overview Length' proves to have some importance for predicting revenue, although the correlation between the two is not strongly evident when plotting their relationship. The blue line in this plot is the line of best fit for these two variables:

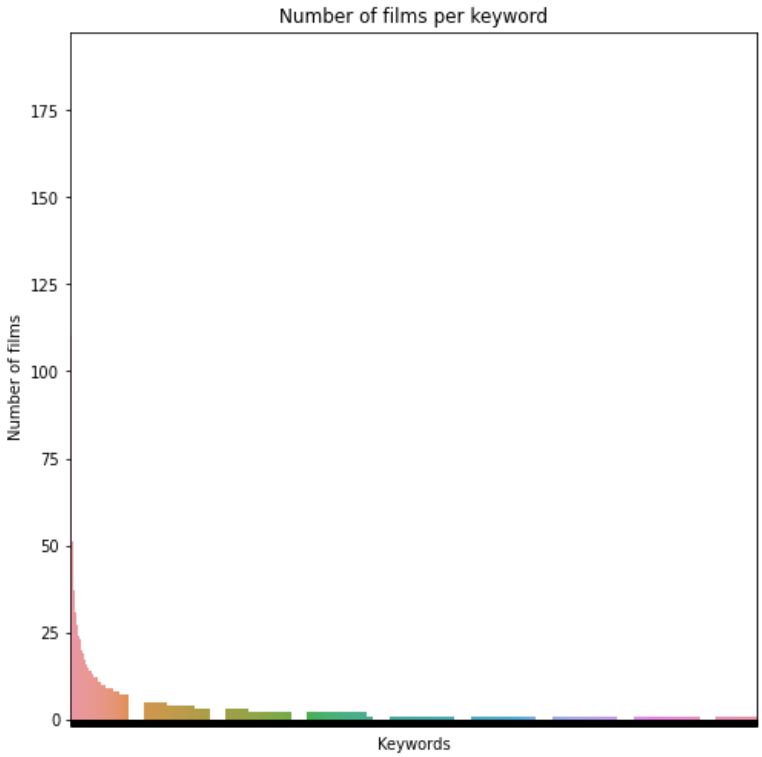


Overview length is an engineered feature that records the length of a film's plot overview used for marketing purposes. It's companion feature "Tagline Length" also had significance for predicting revenue. The purpose was to capture the value of brevity in 'pitching' a film to audiences; particularly since this data set contained little information in the way of marketing efforts on behalf of films.

Data Storytelling: Categorical Data Distribution

Almost all of the categorical data followed the same distribution. A relatively small number of each category (production companies, keywords, etc.) appeared in multiple films. The vast majority of the options for these categories appeared in less than 5 films in the entire data set of 2,000 films.

A good example of this is the Keywords category. There were originally over 7,000 different keywords contained in this data set. Only three keywords appeared in more than 0.5% of films, and there were more than 7,000 keywords in this data set, making it nearly impossible to graph the distribution with any real precision:



This extremely skewed distribution was not quite as dramatic with regards to revenue, but still preserved when examining Keywords and their median revenue.

After some initial modeling I chose to reduce the dimensionality of this data set dramatically and will discuss that strategy later. Only the genre and release month categories did not follow the exponential distribution that keywords demonstrate, all other categorical information followed the same trend.

Modeling Revenue: Baseline Modeling

The first model I chose to use was a linear regression model. The goal was to be able to provide stakeholders with coefficients for each feature that would inform their ability to alter revenue by including certain categories or changing features like budget & runtime. Unfortunately, because this data is organized into a sparse data frame, this resulted in massive error, well over 30,000%. I tried to assess if I could get a more accurate prediction on the data using only a smaller subset of the features in this data set and eliminate any overfitting.

Initially I tried modelling revenue exclusively on the numerical data, then forward selecting and adding in successively more and more complex categorical information. This approach proved to be an improvement but ultimately fruitless. The best accuracy that a linear regression model was able to provide was 2,546% error. I needed to take a different approach that didn't rely on linear regression models.

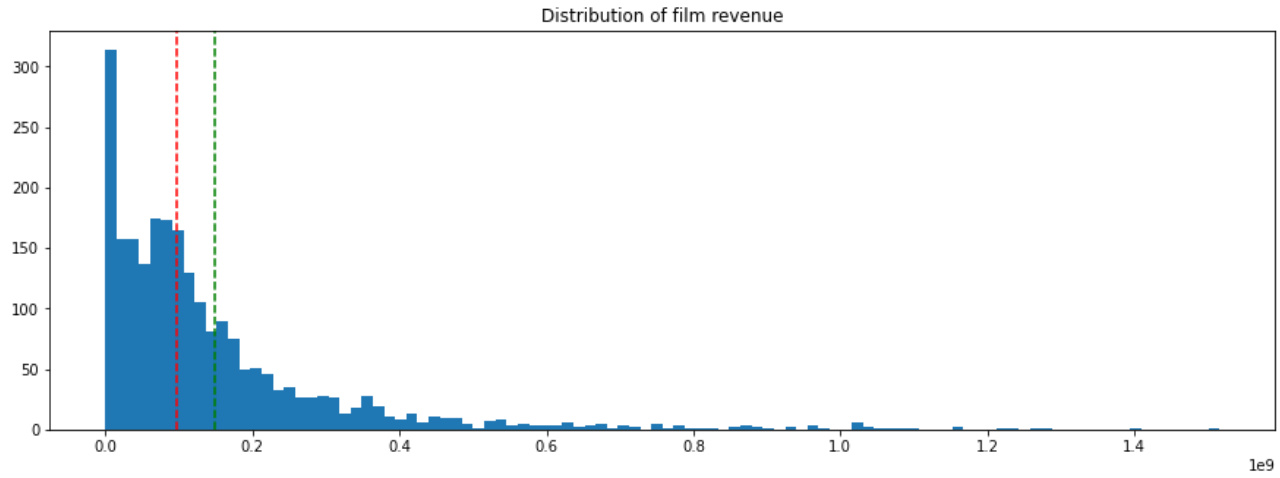
Random forest models are the models that I was most familiar with besides linear & logistic regression, so I started there. Using this model reduced error from 2,546% down to 1,238%, a

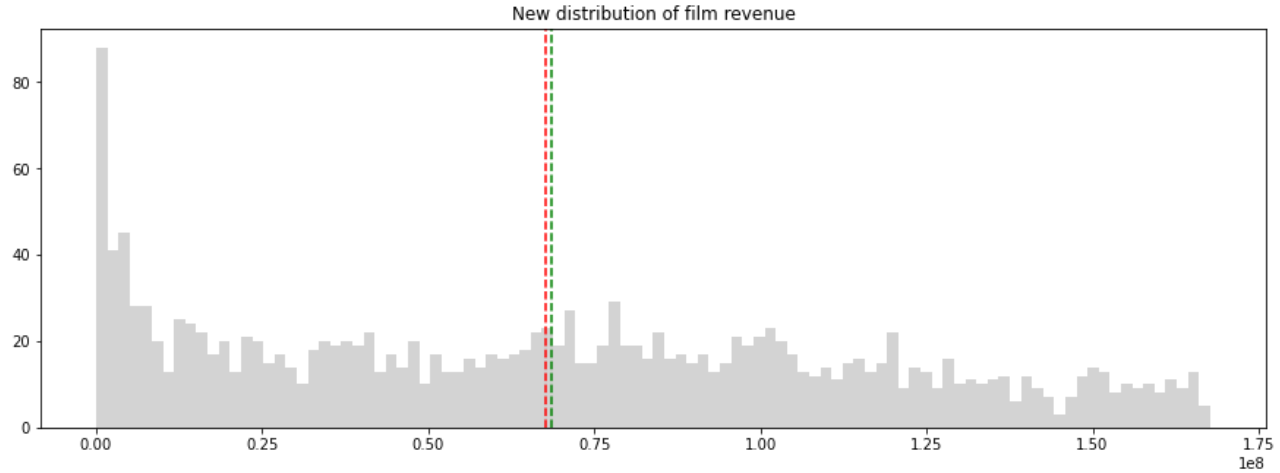
massive improvement but still not usable. Trying a decision tree based model produced improved error but failed to eliminate the issue of significant MAPE entirely.

I needed to revisit the data I was working with and find a way to improve what I was feeding to the ML models without losing highly predictive information. I had two strategies I could use to improve modeling error; I could remove revenue outliers, and I could work to reduce the 10,000+ dimensions I have in the data set.

Extended Modelling: Handling Outliers

The first and easiest of these two options was to remove revenue outliers from the data set. I showed the results of this effort previously, and used trial & error to assess the error improvements from removing a different percentage of films that were outliers. Initially I chose to remove the top 5% of outliers. This did reduce MAPE somewhat to 900%, but I needed to improve the error much more in order to be able to accurately predict revenue. I ultimately decided to remove 15% of the film data after also assessing the impact of removing 10% of outliers. I've included smaller figures here that show the distribution of film revenue before and after removing outliers:





This reduced the MAPE to 289%, another massive improvement but still not usable. While this was an improvement I still would need to reduce the dimensionality of this data set significantly.

Extended Modeling: Reducing dimensionality

My next option was to reduce the dimensionality of the data set, since I had a large number of features that were providing information on an extremely small number of films. This resulted in a highly sparse data set. A large number of features doesn't necessarily mean a large amount of information, there has to be a way to summarize some of this information in fewer categories.

The first step was to eliminate the 41 year categories and reduce these down to five categories representing the decade of a film's release year. Production countries presented another category that was relatively simple to reduce. Originally 63 features representing the different countries films in this data set were shot on, I reduced this to 7 features representing the global regions a film was produced in.

Spoken language was the easiest option to limit, over 87% of films in this data set had English as their spoken language, and I simply eliminated the other categories to indicate if a film was shot in English or not.

Keywords were the largest category to be reduced. Originally over 7,000 features, only 24 keywords appeared in more than 50 films. I chose to drop all keywords that appeared in less than 50 films, and created a new column that indicated the number of keywords that were assigned to a film.

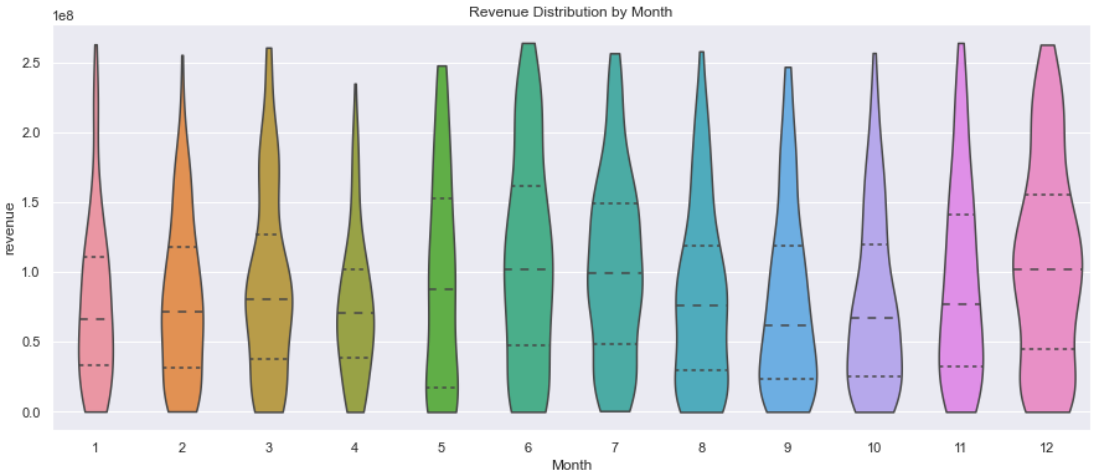
Finally I needed to approach how to reduce the dimensionality of Production Companies, which contained over 2,000 companies. This is an important feature to handle since not only do they control a lot of the work that goes into a film, but also how a film is produced. I chose to break this category down to four options based on the average revenue of each film a company worked on (Quartiles 1, 2, 3, and 4).

I also created a new feature that indicated the number of production companies that worked on a film, since larger films would have more companies brought in to work on them. The data set now had a total of 85 features, down from 9,976. Now that I had fully reduced dimensionality to a more manageable level I needed to test regression models for performance.

I chose to test Random Forest and Gradient Boosting models to capture the performance of decision tree based ensemble methods. I also tested the performance of K-Nearest Neighbors and Support-Vector Machine regressors in order to assess the performance of non-linear and non-decision tree based models. While this improved scores marginally for linear regression and random forest models, it didn't bring the modelling error down to the 20% MAPE goal and actually increased the gradient boosting MAPE by 2%

Extended Modeling: Tiered Approach

Finally, in order to improve error to a manageable level I chose to approach this with a tired method. The first idea was to model revenue of films based on the seasonality of their release. After examining revenue distribution of films among seasons I chose to eschew the conventional Q1-4 approach.



As you can see, the revenue of films released in Jan-April is similar, while May-July, Aug-Oct, and Nov/Dec can all be grouped as having similar revenue. This proved to be promising! While I wasn't able to improve MAPE at all for the Holiday season, this did reduce error for the August-October season to 101% after applying some hyperparameter optimization.

I took this tiered approach one step further, and began to construct 'micro-tiers' that would model a group a film's revenue in a similar fashion to the seasonal method above. Given that oftentimes a production company can control when a film is released, and that it is something that's often decided *after* stakeholders would have chosen to join production, a seasonal-based model may not be the best option.

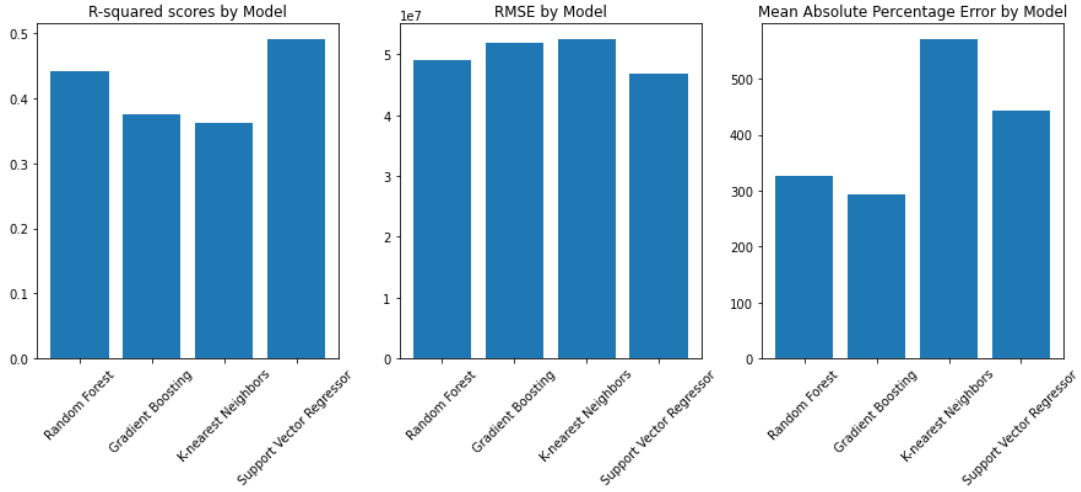
I did start with a tier that included seasonality and modeled action films which are part of a film collection released in 'blockbuster' months (summer blockbuster of june/july and christmas season) resulted in a MAPE of 64%. Ultimately the most effective method was to model films based on the major production company that is working on it (Disney), resulting in MAPE as low as 36%, but this approach was not as successful with other production companies. The second best approach was to model films by genre, resulting in MAPE around 65%, which did generalize well to other tiers.

4. Findings

Results: Model Selection

Of the models tested on the entire data set, Gradient Boosting performed the best with regards to the mean absolute percentage error target that was set at the outset of this project. In

addition to testing a random forest, gradient boosting, KNN, and SVM regression models I also tested the performance of a linear regression model, however its performance didn't increase at all after the data transformations, and it's error metrics were so poor that they distorted the



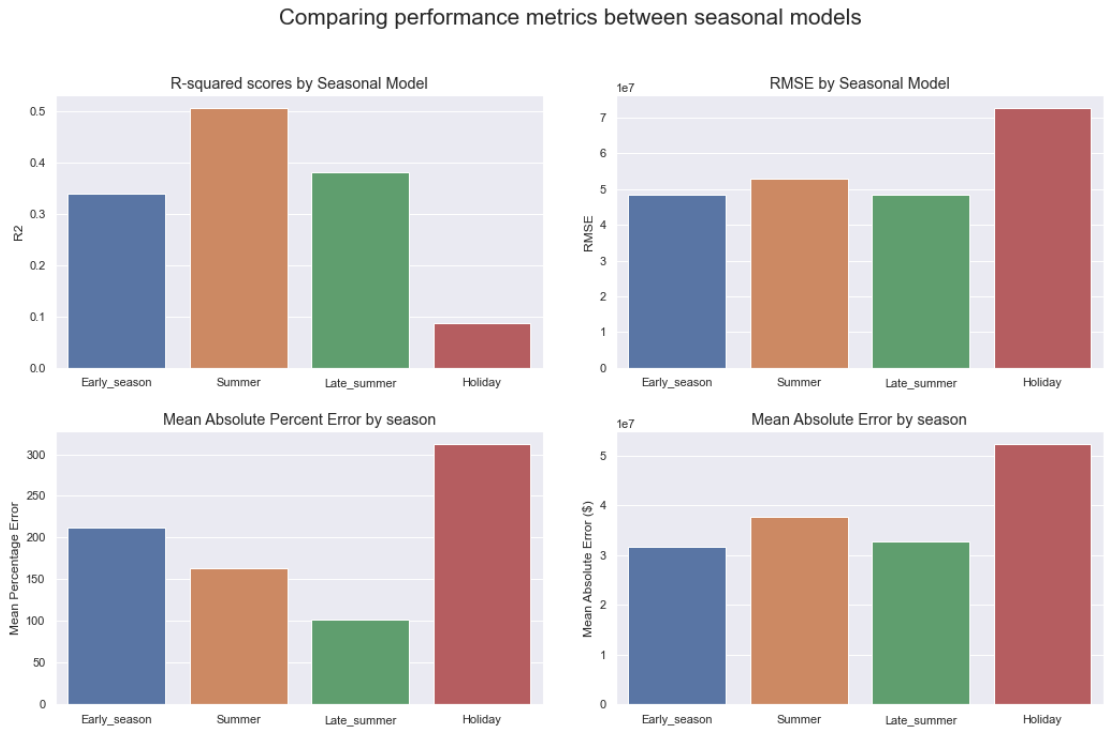
visualization of the other models' scores.

As is evident, while gradient boosting barely edged out the random forest model for performance on MAPE, the support vector regression model performed best using an R-squared and root mean squared error (RMSE) metrics. This means that SVM works well for explaining the variance in revenue, and performs slightly better at predicting large revenue values.

The discrepancy between the two models can be explained by what the metrics are measuring. MAPE is evaluating how well a model predicts revenue and it is scale-independent - an error of $50m is very different if a film's true revenue was $500m vs $50m. RMSE is evaluating the actual dollar amount residual between a model's predicted revenue and the true revenue of a film. As a consequence, RMSE penalizes a 10% error on a $500m film much more than a 10% error on a $50m film.

Results: Comparative Analysis Modeling Seasons

Based on the model performance on the entire data set, I chose to use a gradient boosting model while I tried modeling revenue on different tiers. The first concept to model by tiers was using seasons that were crafted based on which month's revenues were similar to each other. I then modeled all of the resulting seasons with a gradient boosting model, scored them for accuracy, and extracted the importance of various features for each season. You can see the metric results from this approach below:

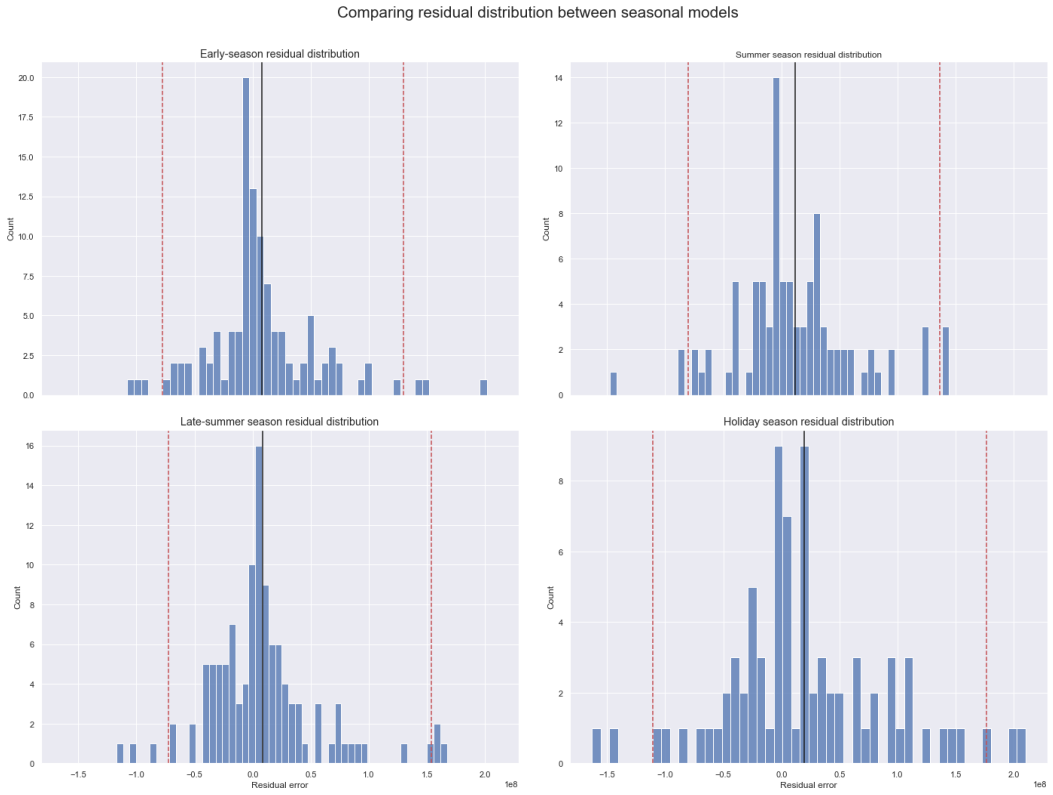


There are several key takeaways from this metric analysis. First, Holiday and Summer season scores differ significantly across the board, but particularly with the R-squared metric. This is somewhat surprising since both were considered ‘blockbuster’ seasons when I created them, and their average revenue differed by less than 1%. What this means, however, is that there is some factor that is causing films to vary in revenue in the Holiday season that is not being captured by this model.

Second, the Late Summer season had the lowest MAPE at 101%. This is still nowhere close to our target of 20%, but represents a significant improvement over the 1,238% error I started with using an out-of-the box random forest model.

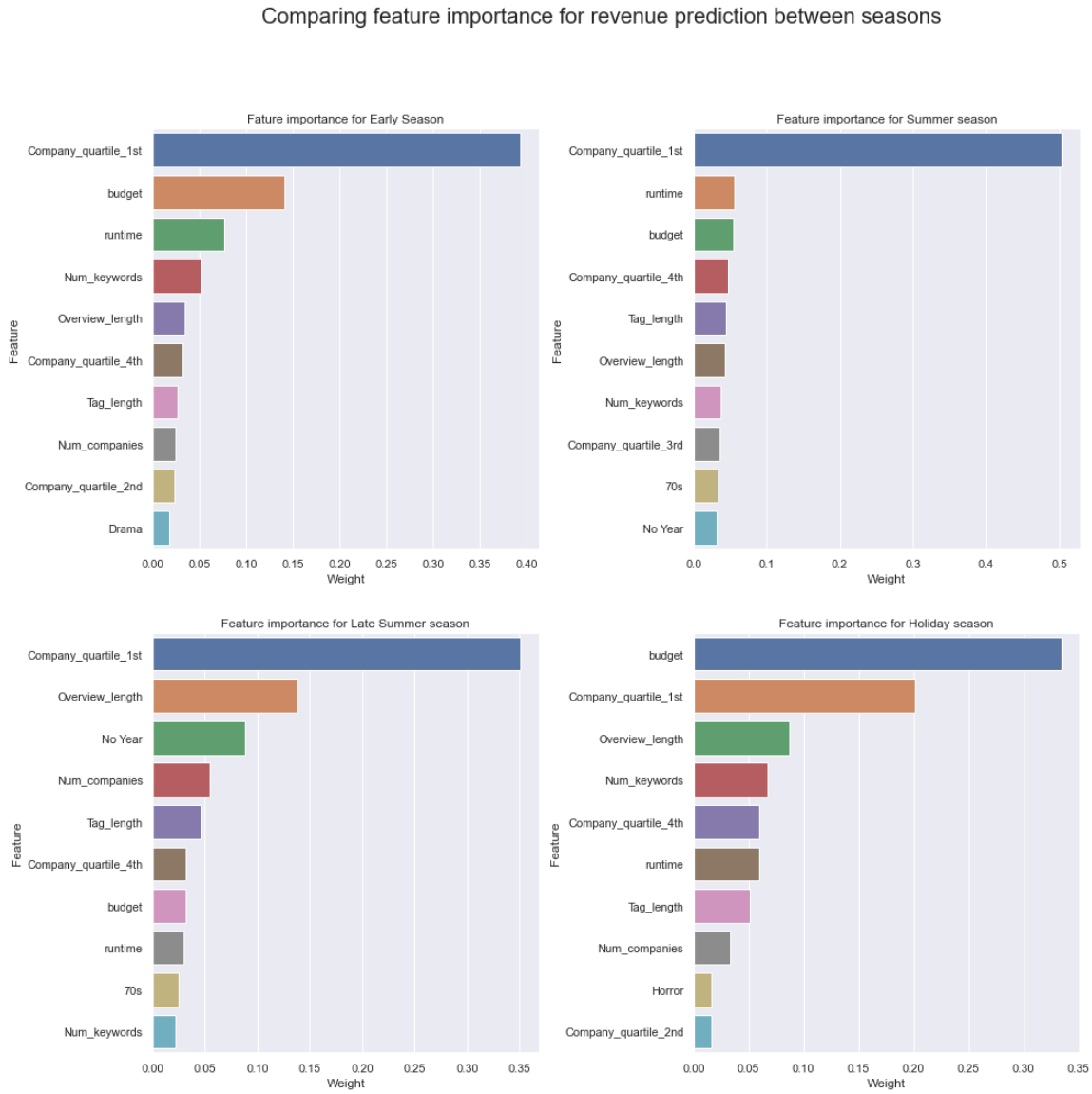
Third, the Holiday season model is the only model that predicts revenue less accurately than a gradient boosting model that predicts revenue for all films. This makes sense since the model is least able to explain the variance in holiday season price based on the R-squared score.

If we examine the distribution of the residual error after modeling these seasons we can see that there is some noticeable difference between the seasons:



While all models tend to over predict films, the residuals do demonstrate the most optimal distribution shape for the Late-Summer model. Further, we can see that the 95% confidence intervals (denoted by the dotted red lines) are somewhat narrower for the Early-season model. This would seem to contradict the results with lower percentage error for Summer & Late Summer seasons; however residual analysis is concerned with actual dollar amounts, and the Early Season had the lowest average revenue by $20m. While the confidence intervals are smaller, those residuals are larger when compared to the actual revenue being generated by films released between January and April.

Finally, and more interestingly, using a gradient boosting model gives us access to the feature importance of each variable in the model:



It is unsurprising that whether or not a company that is in the top 25% of revenue producers (by median film revenue) is an important factor in predicting a film's revenue. What is surprising, is that of the other three company quartiles, only the bottom quartile appears in the top 10 most important features for all four seasons. Intuition would suggest that if knowing one quartile is important knowing the other three would also be somewhat important.

Findings: Tiered Approach Using Smaller Subsets

In order to take this tiered approach one step further I tried modelling even smaller 'tiers' in an attempt to determine if this approach works on more specific subsets of film data. There were several positive green shoots here. Modelling revenue based on seasonal release presented

some improvement in score, I wanted to see if there was a way to improve on this further and strike a balance between # of models needed to predict accurately and accuracy. I was able to model films consistently with accuracy as high as 64% by creating tiers based on a film's genre, seasonality of release, and status as part of a film collection or not.

While I was able to model films more accurately if only examining films where Disney was created with a 36% MAPE score, this approach was not as successful when applied to other production companies. Notably, for all of these tiers either budget or the contributions of a company in the 1st quartile for revenue were also the most important factors in determining revenue.

5. Conclusions

Conclusions: Next Steps

In future one could build models for different movie subsets based on genre and production company; while not near the target of 20% this approach produced the best results thus far at 36%. This could allow you to find a specific model for features of a movie that are immutable, and then test different results by altering variables the production company controls like marketing, budget, release time, etc.

Given more time I would like to explore this approach by using an unsupervised learning model to first cluster films together to create the 'tiers' that will be used. The success of this tiered approach demonstrates that there is potential to also build 'engagement' predictive models. The hyper-specific model approach could also work for streaming services, and build the model based on subsets of customers. (EG: launching a new crime series like Lupin you could model it's performance on the general Netflix audience, then also its performance on different customer segmentations to ascertain who to present it to.)

Conclusions: Recommendations

Before building out a fully tiered approach to modeling film revenue I recommend that we revisit the raw data and extract additional information.

First and foremost I would find a way to unpack the cast & crew data without running into a memory error. Simply including this information, and partitioning the cast/crew into quartiles as we have for companies would add little in the way of dimensionality and provide additional powerful predictors. I would also extract a datetime column that indicated the day in each year where a film was released, allowing me to select for holiday releases with more accuracy and select seasons for modelling with more precision.

Secondly, I would look for additional ways to engineer new features from the raw data. While it's unsurprising that whether or not a film is worked on by a top production company is a big factor in determining its revenue, it does represent a bias towards past success for determining future performance. I would like to pull additional features out of the data that will capture more information about who works on a film besides just the production companies themselves.

This project and final approach to modelling films shows a lot of promise, but needs to be refined further. The inclusion of additional data will likely be a large boost for accuracy, particularly

if accounting for major movie stars working on a film. Modeling films based on smaller and more specific tiers will also provide stakeholders with a more fine-grained look at which features of a film are most important to predicting revenue as well, providing additional insight into how to handle the qualities of a film that remain in the control of stakeholders prior to a decision being made.

Consulted Resources:

Data Sources:

Kaggle. (May 2019). TMDB Box Office Prediction, V1. Retrieved 04/16/2021 from https://www.kaggle.com/c/tmdb-box-office-prediction/data.

IMDbPy. Retrieved 04/21/2021 from https://imdbpy.github.io/

Mentors Consulted:

AJ Sanchez - Springboard Mentor. Provided overall guidance and direction for this project, strategized on ML models to use, suggested reducing dimensionality and the tiered modeling approach.

DJ Sarkar - Springboard Technical Mentor. Provided guidance on many technical aspects of this project, most notably on writing functions for use in this project, log transforming data, and helped me overcome a massive problem with package management.

Kenneth Gil-Pasquel - Springboard Technical Mentor. Provided guidance on debugging my code/data.

Frank Fletcher - Springboard Alumni. Provided guidance on troubleshooting Anaconda, installing Mamba, and managing packages.

Projects and Other Consulted Code:

Referenced to better understand first steps for exploratory analysis: EDA for Categorical Variables - A Beginner's Way. Consulted 05/2021: https://www.kaggle.com/nextbigwhat/eda-for-categorical-variables-a-beginner-s-way

Referenced to understand log-transforming the target variable: StackExchange - Getting Negative Predicted Values after Linear Regression. Consulted 05/2021: https://stats.stackexchange.com/questions/145383/getting-negative-predicted-values-after-linear-regression/145387

Referenced to understand how to convert feature weights into a format for graphing: StackOverflow - Create 2 dimensional array with 2 one dimensional array. Consulted 06/2021: https://stackoverflow.com/questions/17710672/create-2-dimensional-array-with-2-one-dimensional-array

Referenced to better understand how to assign colors to seaborn bar plots: StackOveflow - How to manually assign color to type of categorical variable? Consulted 06/2021: https://stackoverflow.com/questions/59024513/how-to-manually-assign-color-to-type-of-categorical-variable

Documentation Consulted:

Sci-kit Learn

Seaborn

Matplotlib

Pandas

Numpy

IMDbPy