Final Report

Group 13

Jensen Rice, Hyun Choi, Kevin McBride, and Christopher Laughlin

1. Background and Dataset

Problem and Goal: The movie industry makes billions of dollars across the world. The success of these movies, however, is not universal. Some crash at the box office, costing the production company or companies paying for the film millions of dollars while others enjoy hugely successful revenues. The goal of our project is to predict what movies will fail or succeed and to identify key factors in box office success.

Dataset: The data is obtained from The Movie Database. The dataset contains 7,398 movies with each data points contain information about the movies such as budget, director, actors, revenue and more. The data is a schema on write and schema on read; some columns contain movie data in JSON format. The dataset contains high number of missing data values, with important features such as budget having around 2,000 missing values and the feature we are trying to predict, revenue, has more than 4,000 missing values. To address this problem, we obtain data from another data source, and we reduce the number of missing revenue to 1,445 values. This project deals with modeling data with a very small data set as the data does not have high velocity or volume.

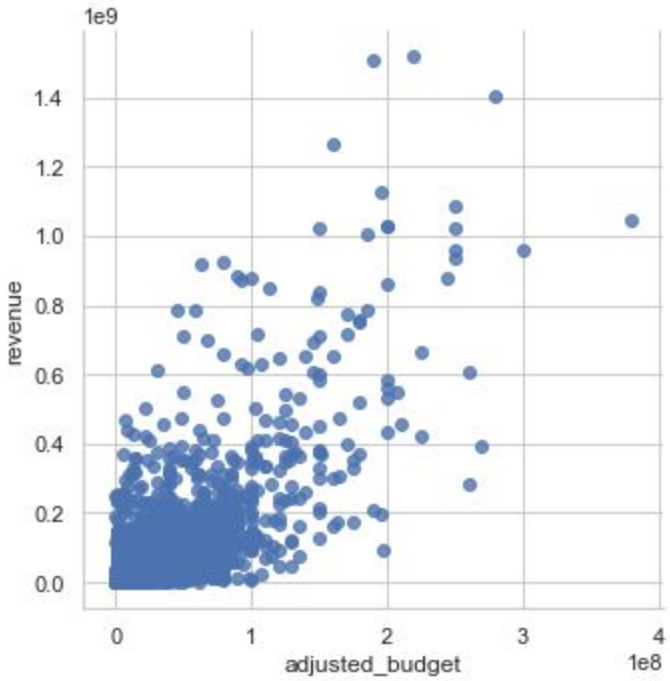
2. Preprocessing

2a - Format the Data: The first step of preprocessing was converting the csv data files into a workable structure for use in our Python environment. Panda was used to read in the csv files into panda dataframes, and the string representation for each feature was converted to a dictionary using AST.

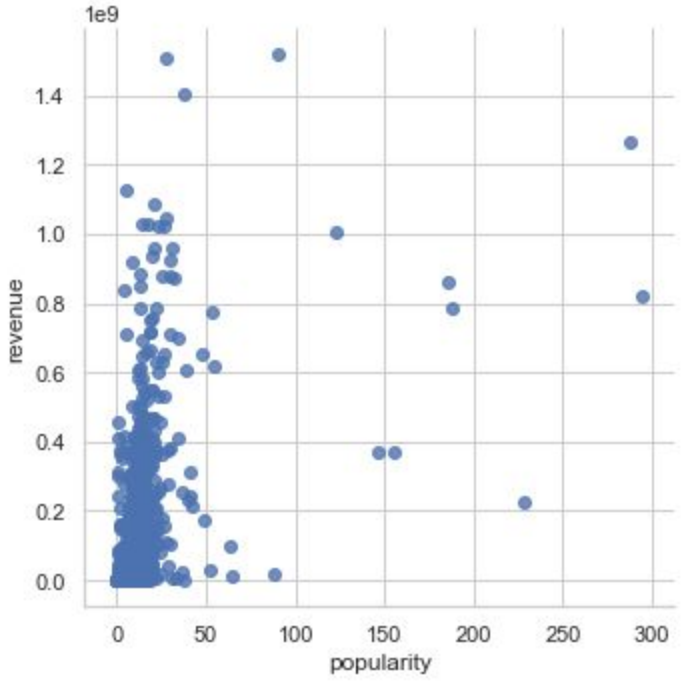
2b - Analyze and Trim the Data: Next, the features included in the data were tested for correlation with the label, revenue.

Continuous Features:

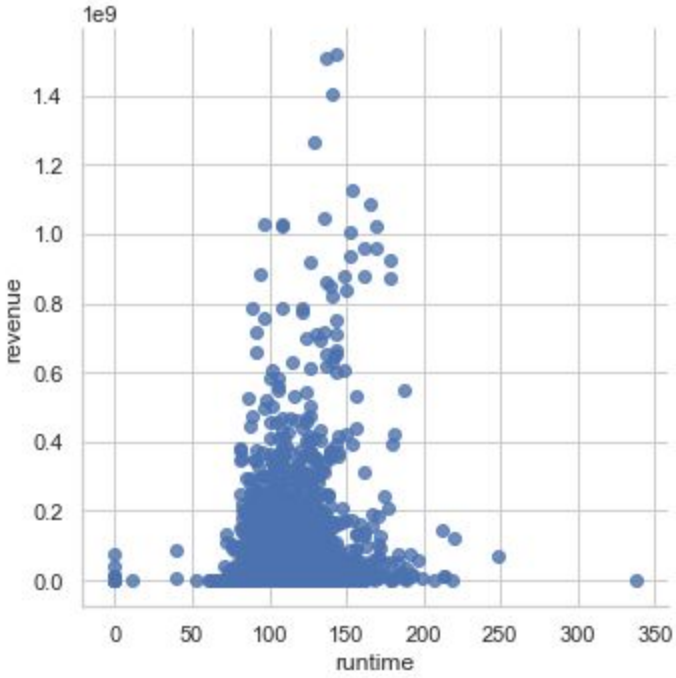
‘budget’: Amount of money spent on creating the film. Since a quarter of the movies were missing budget we attempted to use a linear regression model to predict the budget values of the missing movies. The assumption was that features such as the number of cast and crew members would be highly correlated with the budget. Modelling, however, performed better when filling the budgetless movies with the 30th percentile budget number than the linear regression predicted values. This may be because many of the movies whose budget was not listed were smaller movies, and there is no way to train the model on that feature because every movie that has that feature has a budget of zero.



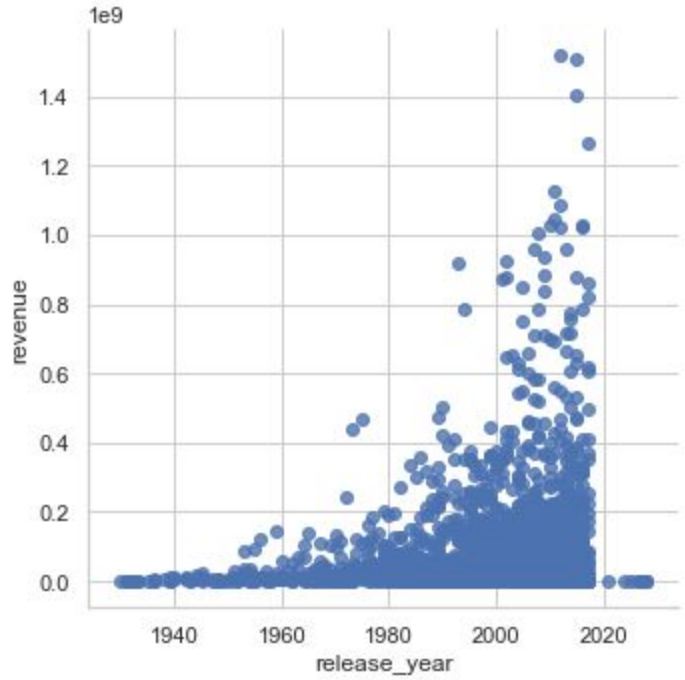
‘popularity’: Feature describing the popularity of a movie.



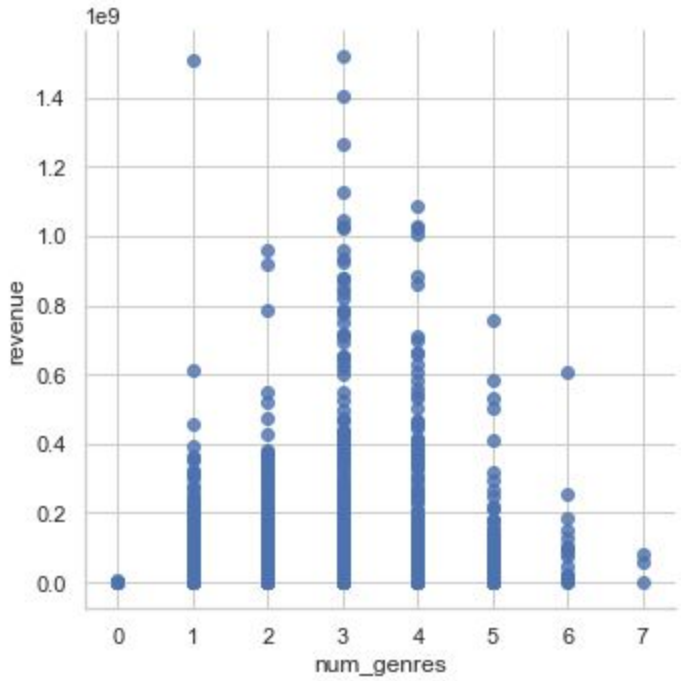
‘runtime’: Length of the movie.



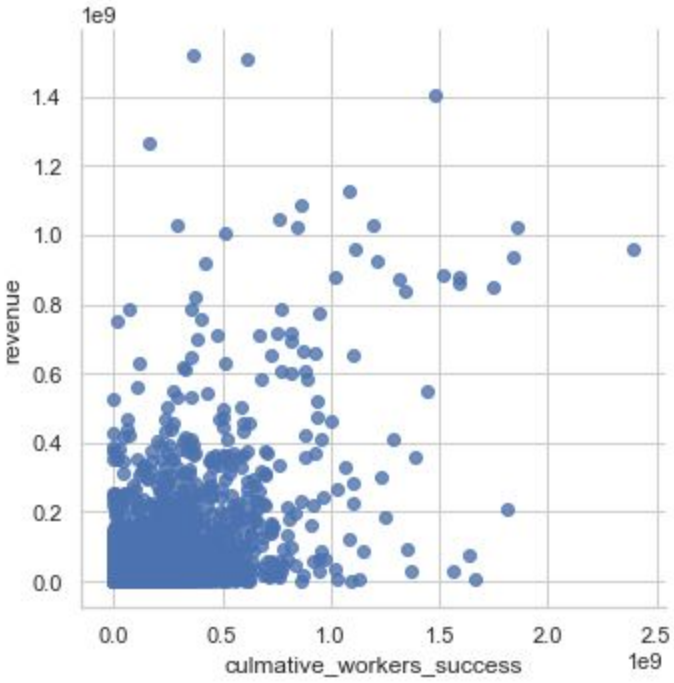
‘release_date’: We processed this into release year for each movie. Taking into account month or day of the week might also be worth consideration. CPIs (consumer price index), which are measurements of inflation, were gathered for each year for an attempt to take into account inflation throughout the years. The modelling performance, however, decreased when using the CPIs instead of realease_date so the feature was dropped.



‘genres’: Parsed into the number of genres a movie has.



‘cast’ and ‘crew’: These listed the personnel involved in a movie, e.g. actors, producers, and directors. These two features are clearly linked to movie success, but not easily translated into a format that a model can handle. The approach we took involved measuring the success of four high impact personnel: the director, producer, and the two lead actors. We created a measurement of success on every movie for each of the four personnel that averaged the revenue of past movies done by the actor or cast member. We limited this feature to only including the revenue of movies made before the release date of the movie on which we are creating these features in order to make the features relevant in a real world scenario where you don’t know the future performance of actors and cast members you hire. The sum of the four measurements for each movie vs. revenue is shown below.



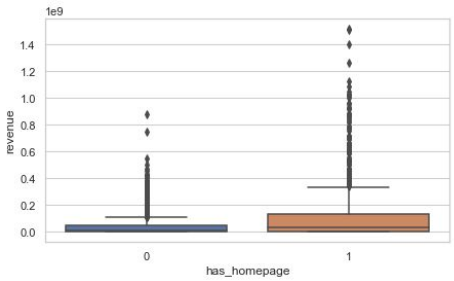
Discrete Features:

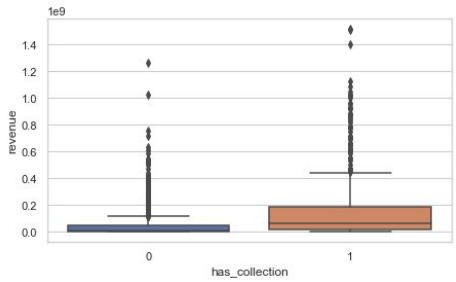
‘homepage’: A feature with a link to the homepage, the movie’s website. Text information in the URL is limited and redundant to other features so only the effect of having a tagline vs. not having a tagline was tested.

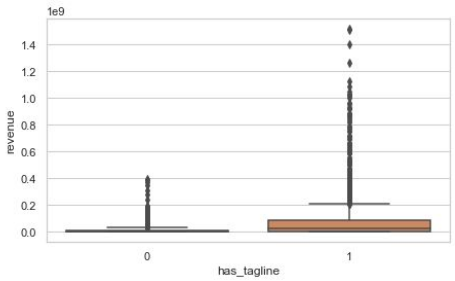
‘collection’: A feature that states the collection a movie belongs to, e.g. “Marvel Cinematic Universe”. The count for movies in the same collection was far too low (maxing out at under ten in our dataset) so only the effect of having vs. not having a collection was tested.

‘tagline’: One sentence ad-line. Tested for the effect of having a tagline vs. not having a tagline.

Box plots (‘has_homepage’, ‘has_collection’, and ‘has_tagline’ vs. revenue)**:**





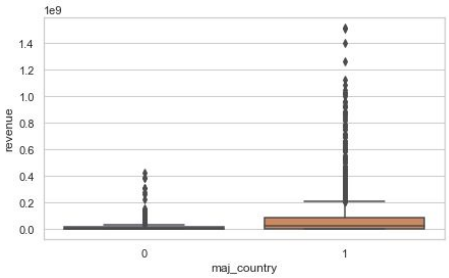


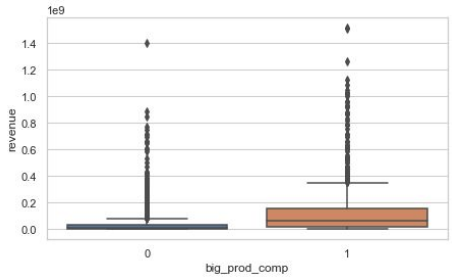
‘production_countries’: Countries the movie was produced by. Movies made by the UK and USA had much higher revenue than other countries. The feature was translated into ‘maj_country’ describing whether the move was produced by one of these two countries or not. This feature may exhibit multicollinearity with original_language.

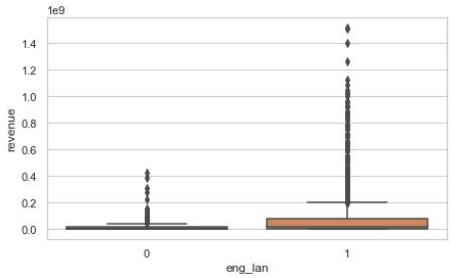
‘production_companies’: Production companies of the movie. The companies 'Universal Pictures', 'Paramount Pictures', 'Twentieth Century Fox Film Corporation', 'New Line Cinema', 'Warner Bros.', 'Walt Disney Pictures' made significantly more movies than the other companies and movies produced by them made significantly more money the feature was translated into ‘big_prod_comp’ describing whether the movie was produced by one of the top production companies or not.

‘original_langauge’: Original language of the movie. English was the most common and also had the highest correlation with revenue. The created feature, ‘eng_lang’, describes whether or not the original language of the movie was english.

Box plots (‘maj_country’, ‘big_prod_company’, and ‘has_tagline’ vs. revenue)**:**





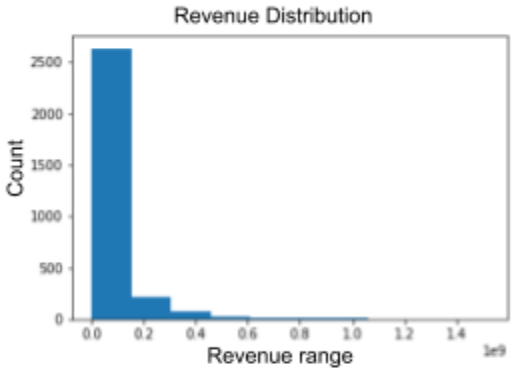


Genres:

Genres were one-hot encoded and genres that had a significant difference in means of the revenues with the rest of the genres were included. The genres that we found to have significant difference in the means were, drama, adventure, fantasy, western, thriller, animation, action, mystery, foreign, science.

Revenue:

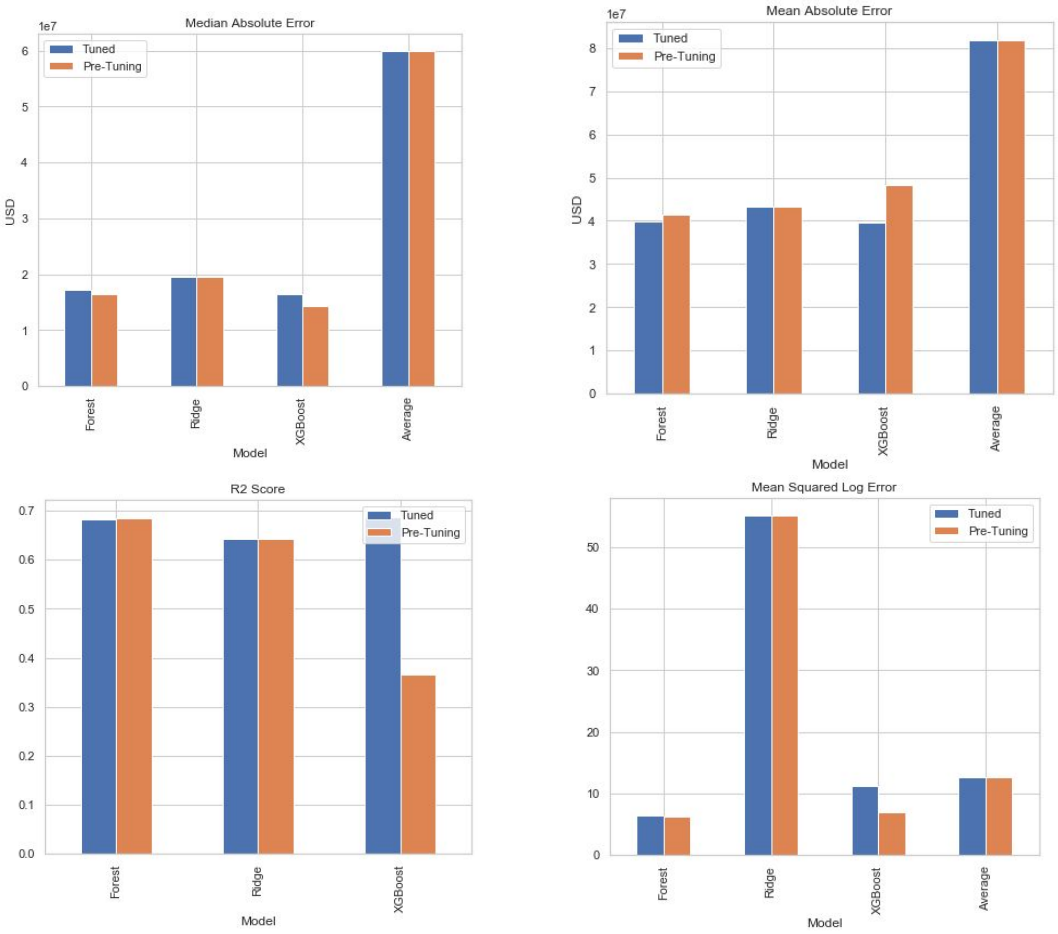
We performed a log transformation of revenue so that the revenue would be distributed more evenly. The revenue before the transformation contains many movies that are within the lower end but with the transformation it has a shape closer to a normal distribution.



3. Modeling

Methods Used: We used three different models on our data: scikit-learn's Ridge model (a linear regression model with L2 regularization), scikit-learn's RandomForestRegression model, and XGBoost (decision tree based models). We measured the performance of these models using scikit-learn's regression metrics: mean absolute error, median absolute error, explained variance score, mean square error, mean squared log error, and R2 score. For each model we ran cross-validation using scikit's RandomizedSearchCV and parameter grids that focused the search for the best hyperparameters for each model.

Results: The results of four of the metrics we tested for on the tuned and untuned models are shown below (average represents predicting the average revenue for every movie in order to get a sense of the impact of the modelled predictions):



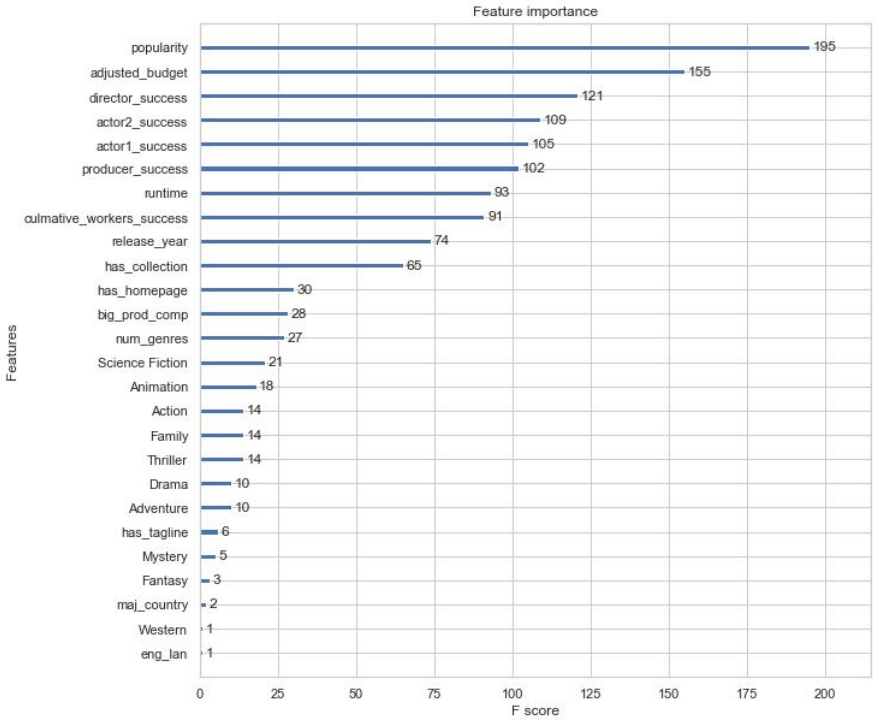
Analysis: These results give insight into the interaction between our model and our data. The median absolute errors for our models are roughly between 14 and 19 million dollars, but the mean absolute errors are roughly between 39 and 49 million. Congruently, our R2 scores are generally low, between 0.6 and 0.7 except for the pre-tuned XGBoost at only 0.35 (which outperformed every other model in the median absolute error metric). These three results point to an inability to reliably predict outliers and high error values skewing the results. A high mean and a low median point to this because it shows an imbalance towards the higher end of the spectrum. The low R2 scores show this because not being able to predict outliers can ruin the wellness of fit. Untuned XGBoost significantly outperformed every other model on the median absolute error metric likely due to its low R2 score; the model doesn't fit the outliers well so it performs better on the remainder of the data.

This analysis is further supported by looking at the highest error value for the RandomForestRegressor (arguably the best performing model), which is 60 times higher than its own median error and 86 times higher than the best median absolute error obtained by our models. These errors steeply drop off relative to the size of the testing set, but even a few errors this large are heavily impactful on the results.

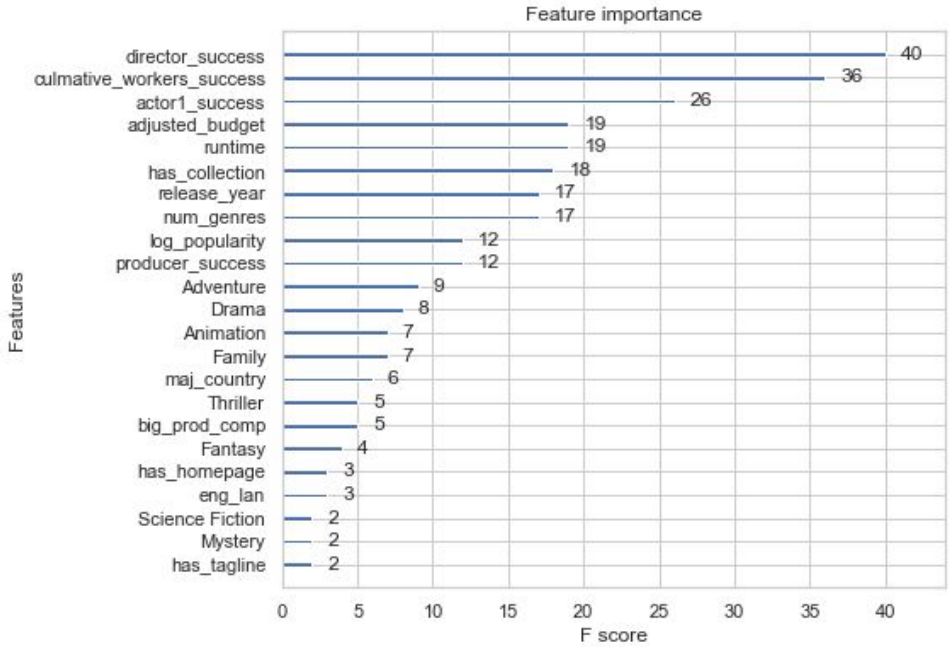
There are two possibilities for the cause of our inability to predict outliers: either outliers cannot be predicted by our data and decided instead by the many intangibles of movies, or our data simply doesn't have the volume required. My hypothesis is that it is a mixture of these two factors, predictions for movies will always be flawed but a higher volume of data would greatly improve performance.

Feature Importance: To test what features were most important to modelling success we used XGBoost's feature importance function to see what features had the highest impact on XGBoosts's predictions. The bar charts below show feature importance on the tuned and untuned models. The tuned feature's importance ratings were closer to expected: popularity and budget both heavily influenced the linear regression model. Both models highly rate the worker success features, which is slightly unexpected due to their significant but small impact on the linear regression model. The untuned model in particular rates worker success features as the top three features used for predicting revenue.

Tuned XGBoost Feature Importance:



Untuned XGBoost Feature Importance:



4. Post Presentation Additions and Improvements

After the presentations, we decided to test if our models were overfitted. To check our results we trained our models with the hyperparameters that had the best results with k fold cross validation (k=9). The results of the cross validation show that the models were a little bit overfit as the R2 scores for each of the models (linear, random forest, xgboost) all decreased by around 0.1.

5. Member Contributions

The preprocessing and data visualizations were done by Hyun and Jensen. Modelling was performed by Christopher and Kevin, working with the later stages of preprocessing by giving performance feedback and feature relevancy.

6. Github Link

https://github.com/CRLaughlin/CAP4770-Group_13