Capstone Project Milestone Report:
Using Machine Learning to Predict Movie Success

Introduction

Problem

The entertainment industry is a high-profile industry with movie producers often investing millions of dollars into making movies. The success of a movie can depend on several factors such as budget, cast, genre, release date, critical reception etc. Understanding the key drivers and predicting movie success can therefore be extremely useful in making key business decisions in the pre-production, production and distribution stages. For example, predicted movie success can be used to determine compensation of cast and crew, marketing as well as other aspects of the budget not yet determined.

Client

Movie producers and investors can help mitigate potential losses and investment risk by better analyzing and predicting movie success. Movie studios can also use this analysis to determine marketing, budgeting and creative decisions.

Dataset

"The Movies Dataset" on Kaggle contains data (cast, crew, plot keywords, budget, revenue, posters, release dates, IMDB rating, languages, production companies etc.) for 45,000 movies. In addition, CSV files of the 50 highest grossing actors and 35 highest grossing directors of all-time was exported from IMDB to add to the analysis.

Approach

I will use supervised learning algorithms to determine the key drivers of movie success and predict the success of a movie in 3 different ways:

a) Predicting *revenue* using a regression model,
b) Predicting *return-on-investment* (i.e. revenue/budget) using regression in order provide more actionable analysis for producers and investors, and
c) Predicting *categorical movie success* (i.e. hit, break-even, flop) using a classification model.

Data Cleaning

Description of Features

Following is a complete list and description of all the original columns/features in the dataset:

File: credits.csv
cast: List of characters and actors in a movie
crew: List of crew members, including the director
id: Movie ID

File: movies_metadata.csv
adult: True/False column identifying whether a movie is an 'adult movie' or not.
belongs_to_collection: Name of collection/franchise a movie belongs to (e.g. Toy Story Collection)
budget: Budget of a movie
genres: Genres associated with a movie
id: Movie ID (same as credits.csv)
original_language: Original language a movie was made in
release_date: Release date of a movie
revenue: The amount of revenue a movie generated in theaters
runtime: Length of a movie
status: Status of a movie (e.g. released, post-production etc.)
title: Title of a movie
vote_average: Average user rating a movie received on TMDB
vote_count: The number of votes a movie received on TMDB

Several columns such as homepage, tagline, video etc. were dropped from the database after being identified as unimportant. As mentioned earlier, a list of top-grossing actors and top-grossing directors was also obtained from IMDB to add to the analysis.

All csv files were converted to pandas DataFrames and any duplicate rows were removed. Column types were checked and changed to the correct data type (e.g. the budget column was changed from type 'object' to a numerical float).

Handling Missing and Erroneous Values

1. There were several missing values within the movies dataset. ~75% of the movies showed budget and revenue values of 0, which were actually indicative of missing values after cross-referencing these movies with TMDB. Since revenue is a predicted variable and budget is assumed to be an important feature, movies with missing values in both these

columns were dropped from the dataset. Similarly, movies with missing data in the runtime, vote_average, and vote_count columns were also removed.

2. The belongs_to_collection feature also consisted of >4000 null values. However, this was not a result of missing data. Instead, the null values simply indicated that the movie did not belong to a collection or franchise. The column was converted to a binary column, with all null values as 0s and all non-null values as 1, indicating that the movie did belong to a collection.

3. After the removal of all null values, the budget and revenue columns still showed several single-digit values, which likely indicates errors in data capture. This was confirmed by cross-referencing a few of these values with their actual revenue and budget values manually. Since most movies have a budget of at least $10,000 and a revenue of at least $1500, all movies with a budget below $10,000 (n = 59) and a revenue below $1500 (n=57) were dropped from this dataset to ensure that all erroneous data was removed.

4. After performing the cleaning steps above, the 'adult' column only contained movies categorized as False. Since this column did not add any additional insight, it was removed from the dataset.

Adding Additional Features: Top Actor and Top Director

1. In order to determine if having a top actor (based on the IMDB top 50 actors list) or top director (based on the IMDB top 35 directors list) affects movie success, the movies and credits DataFrames were merged to add the cast and crew information to the cleaned DataFrame.

2. From the cast and crew columns, 2 additional columns were created: a lead actor column (obtained by extracting the first-listed actor from the cast) and a director column (extracted from the crew).

3. The lead actor and director columns were then cross-referenced with the DataFrames containing the IMDB top actor/director lists to create 2 new binary columns: 'top_actor' and 'top_director'. If the lead actor or director in a movie were found in the IMDB lists, their respective column value would be 1; otherwise, it would be 0.

Dummy Variables - Genre and Release Season

1. Since each movie is often associated with more than 1 genre, dummy genre variables were created to identify the genres associated with each movie. In other words, each genre was converted into a binary column, with a value of 1 if the movie belonged to that genre and a value of 0 if it did not.

2. The ‘release_date’ was converted to type *datetime* and each date was categorized into different release seasons and holidays - Summer, Holiday Season (Thanksgiving, Christmas, New Year’s), Valentine’s Day, Labor Day, and MLK Day). If the date did not fall into any of these aforementioned categories, it’s release season was categorized as Off-Season.

3. These seasons were then converted into dummy variables, with the Off-Season column dropped to avoid problems with multicollinearity.

Generating Additional Target Variables - ROI and Categorical Success

1. As mentioned earlier, apart from revenue, the aim of this project is also to predict ROI and categorial success. As such, a new column was created for ROI (revenue/budget), noting that the budget feature would be ignored in this prediction since budget is being used to directly calculate the target.

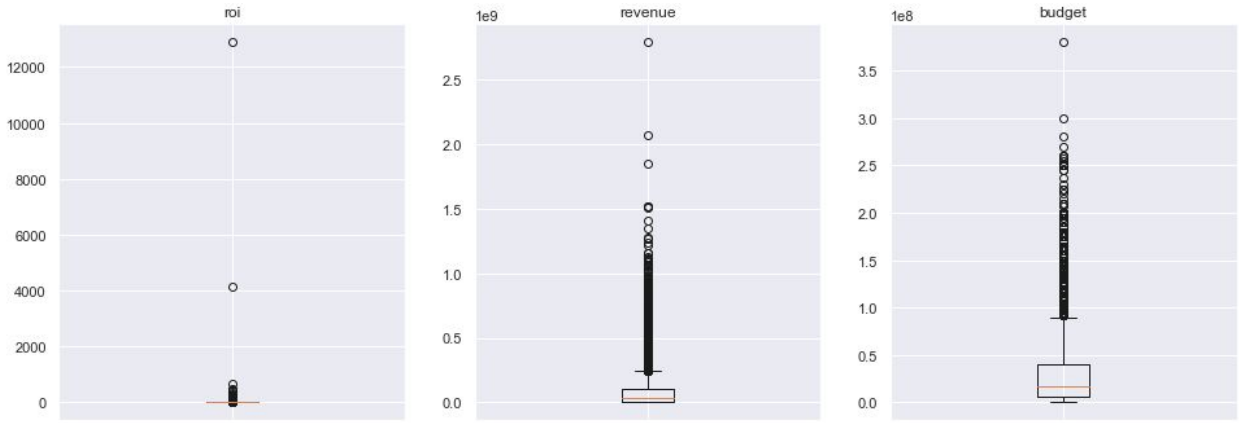
2. The categorical success of a movie was determined using the ratio of revenue to budget, or ROI.

Ratio of Revenue to Budget	Categorical Success
>= 3.5	Hit
>= 1.2 and < 3.5	Break-Even
< 1.2	Flop

3. Finally, the index was reset and a final check was performed to ensure the data looked good, was of the right type, and had no null values. The final, cleaned DataFrame consisted of 5,262 movies.

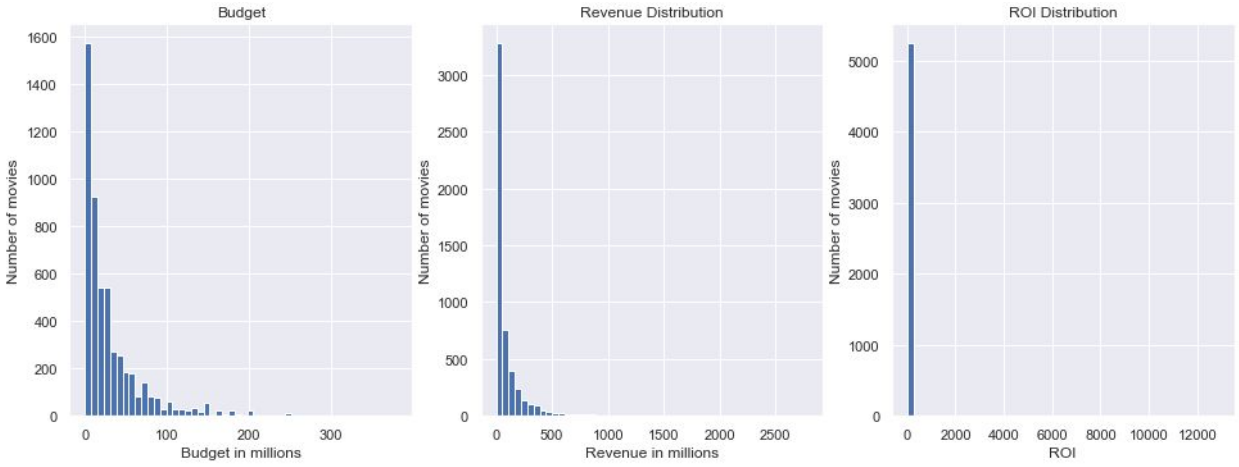
Exploratory Data Analysis

Outlier Analysis for ROI, Revenue, and Budget

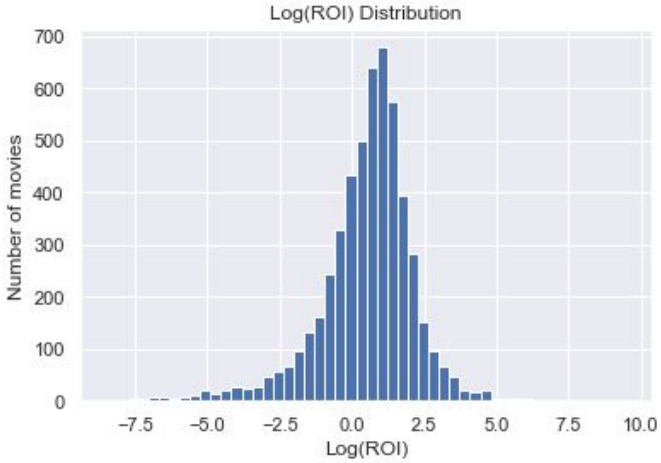


All three sets of data show outliers on the higher end, with the ROI data appearing to have some extremely strong outliers. These extreme outliers in ROI are representative of movies made on smaller budgets that performed exceedingly well at the box-office (e.g. Paranormal Activity, The Blair Witch Project).The revenue and budget data also show a large number of outliers on the high end (greater than ~200 M USD and greater than ~100 M, respectively), although the outliers are less prominent than those observed in the ROI data. Since movie revenues and budgets have been increasing significantly over the years (see analysis below), movies with budgets and revenues typically considered as outliers will still be taken into account in the initial predictive model.

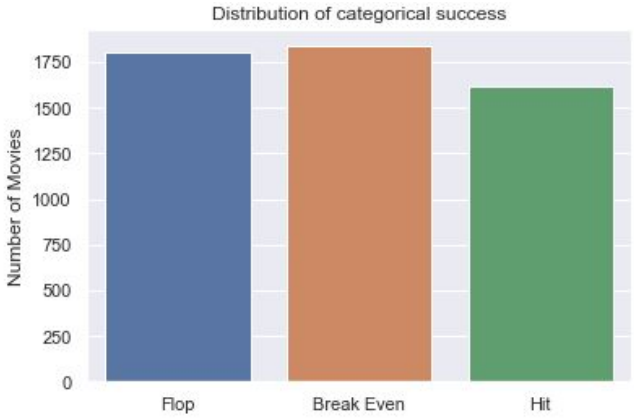
Distributions for Budget, Revenue, ROI, and Categorical Success



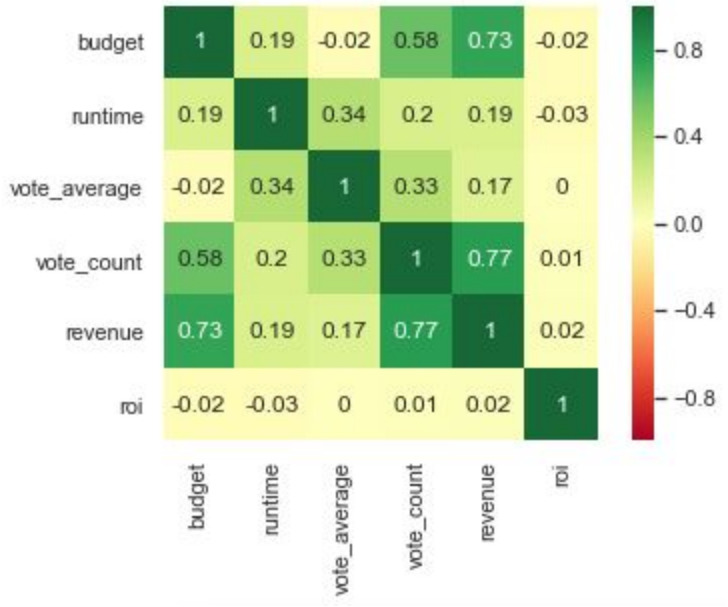
It should be noted that while all 3 distributions of numerical variables are pretty heavily skewed to the right, the ROI distribution is particularly skewed due to outliers. In such extreme cases, a log transformation of the variable might yield better prediction results. The plot below shows that log transformation of the ROI results in a distribution that more closely resembles a normal distribution. However, the initial base model will first be fitted on the original data to better understand the effect of a log transformation.



The distribution of categorical success, shown below, is fairly even with the number of hits being slightly slower than flops or break-evens.



Correlation Matrix of Numerical Features and Targets



The correlation matrix of continuous variables shows the biggest linear predictors of revenue to be budget (0.73) and vote_count (0.77). Runtime and vote_average each show a linear correlation of 0.19 and 0.17, respectively, with revenue. None of the continuous variables have a high linear correlation with ROI. In terms of feature collinearity, runtime and vote_average have a correlation of 0.34, while vote_count and vote_average has a correlation of 0.33. In general, correlations coefficients over than 0.7-0.8 between features indicates feature dependency, which should be avoided.

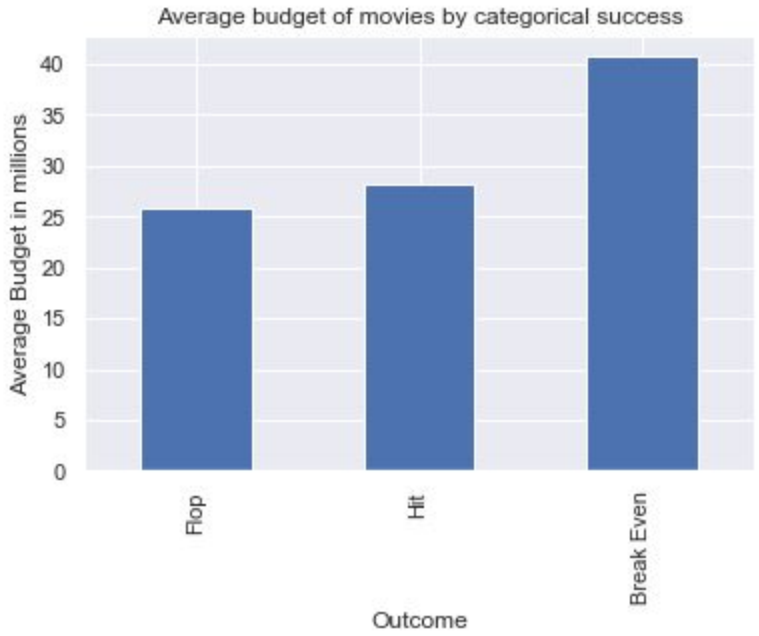
However, a correlation matrix only shows the direct, linear relationship between two variables. It is possible for multicollinearity to still exist between features, which could result in an unstable model. In order to assess the existence of multicollinearity within our data, we can look at Variable Inflation Factors (VIFs) of the feature variables. There are different thresholds for VIFs for features to be considered multicollinear - in this case, we will be using a threshold of 10.

	VIF Factor	features
2	38.221020	vote_average
1	36.378651	runtime
0	3.405193	budget
10	3.131515	drama
3	2.413300	vote_count

The VIF analysis shows vote_average and runtime to have high inflation factors, indicating that predictors are highly correlated with each other. In order to avoid problems of multicollinearity,

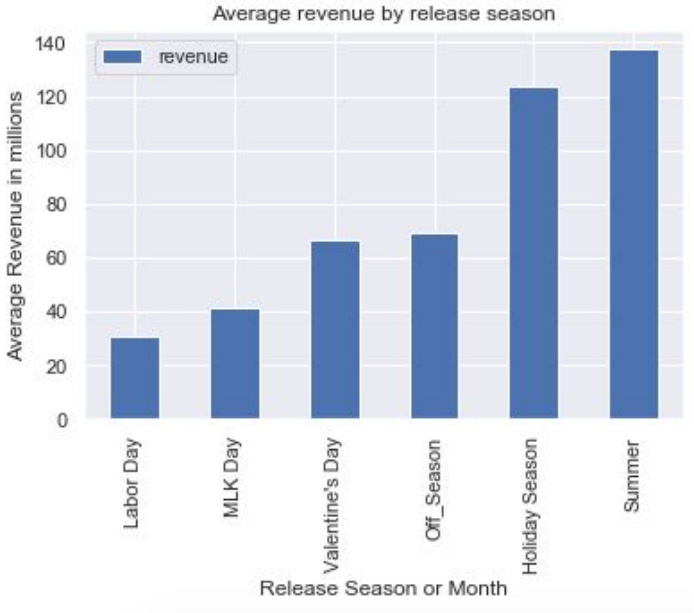
one of these variables should be removed from the analysis, preferably the variable with the higher VIF. Removing vote_average, lowers the VIF of runtime to <10. Therefore, vote_average will be removed as a feature in our models.

Relationship between categorical success and budget



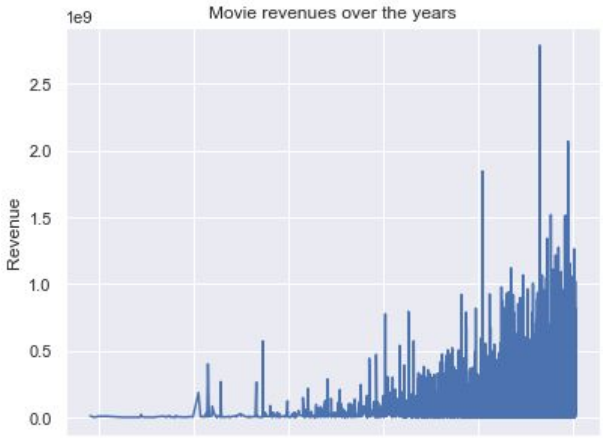
The average budget is lowest for movies that are flops and hits, and ~15M higher for movies that broke even. This phenomenon can be interpreted in two different ways. Note that categorical success is based on the **ratio** of revenue to budget (ROI). While having a lower budget can increase the ROI, having too low of a budget can affect the quality, reputation, and star power of a movie, thus resulting in significantly lower revenues, which is the case for flops. On the other hand, having a lower budget also allows the movie to recover their costs more easily and be categorized as a hit. Whether a lower-budget movie is a hit or flop likely depends on other factors such as genre, star power, release season etc.

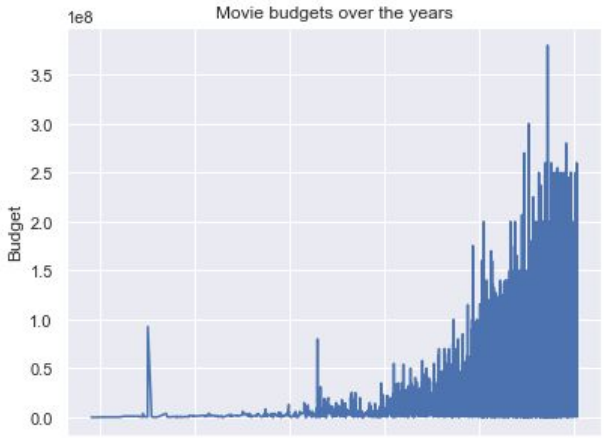
Impact of Release Season on Revenue



Movies released in the summer make the most money on average followed by the holiday season. Holidays like Labor Day, Valentine's Day, or MLK Day don't appear to have much of an impact.

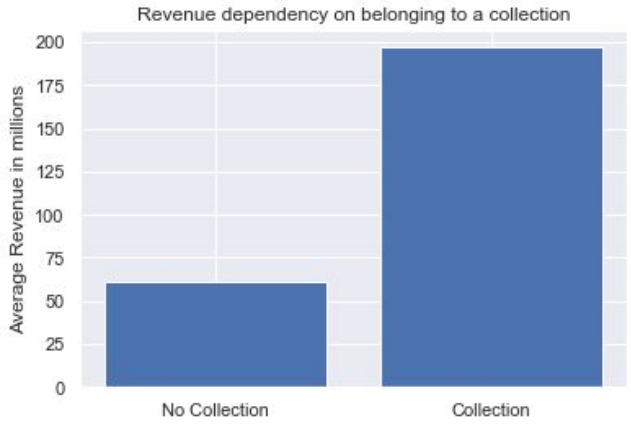
Evolution of movie revenues and budgets over the years





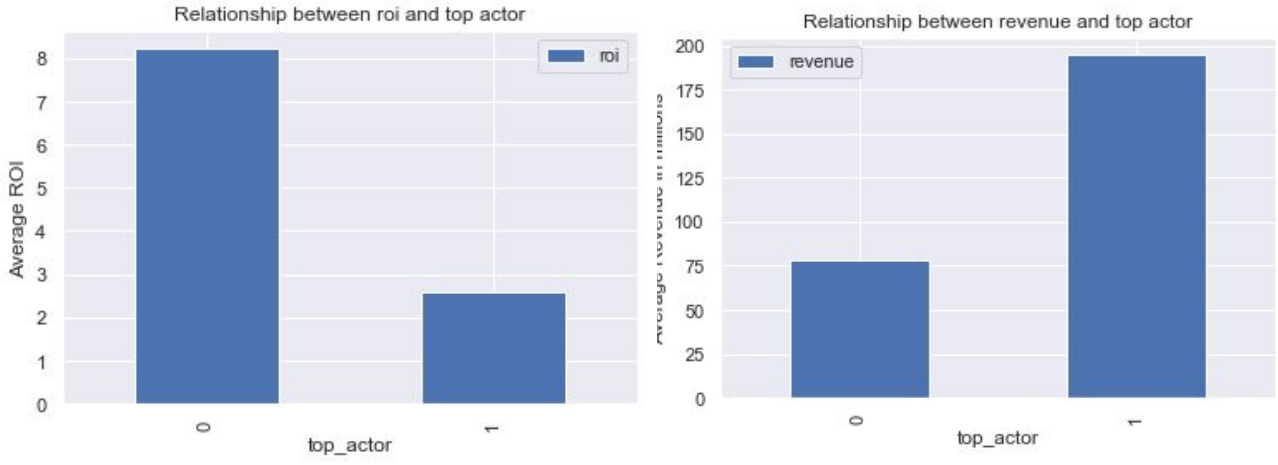
Overall, both movie revenues and movie budgets have been steadily increasing since the 1980s, with the budgets seeing a steeper increase since the 2000s. Keeping this in mind, it may be important to keep the outliers in the budget and revenue data since they could be representative of the general trend of higher budgets and revenues.

Relationship Between Belonging to a Collection and Revenue



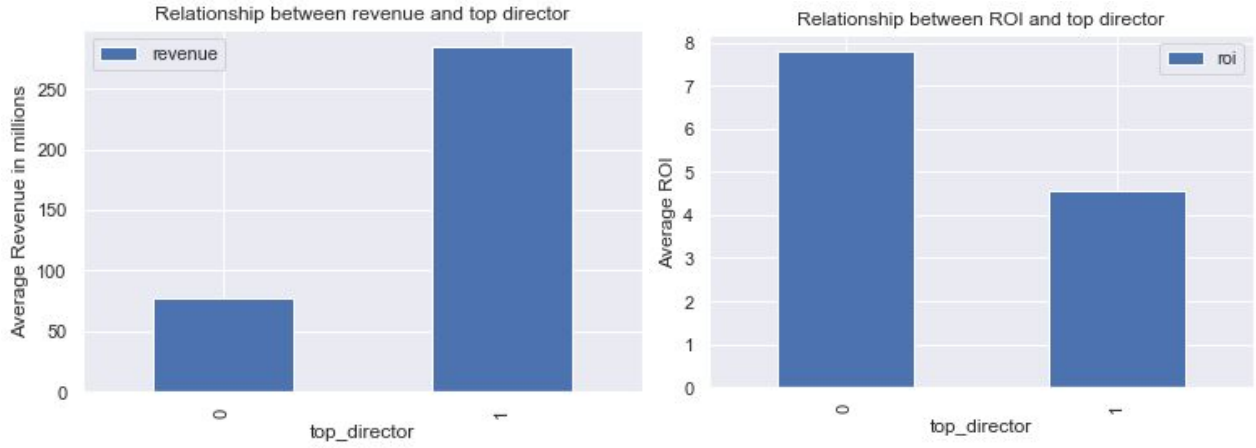
Movies that belong to a collection (i.e. franchise) gross, on average, almost 4 times higher than movies that do not belong to a collection. Statistically, this difference in mean revenue is significant as observed by the t-test, which resulted in a p-value close to 0 indicating that the means for these 2 groups are not equal.

Relationship Between Top Actor and Revenue/ROI



A top actor is defined as any actor appearing on highest grossing actors list on IMDB. While having a top actor as a lead does result in a much higher revenue on average (~200M vs 50M USD), it also results in a lower average ROI (~75% decrease), likely due to the higher compensation top actors demand. The difference in revenue and ROI distributions are statistically different for movies with and without top actors, as confirmed by the non-parametric Mann-Whitney U test, which resulted in a p-value close to 0 for both revenue and ROI.

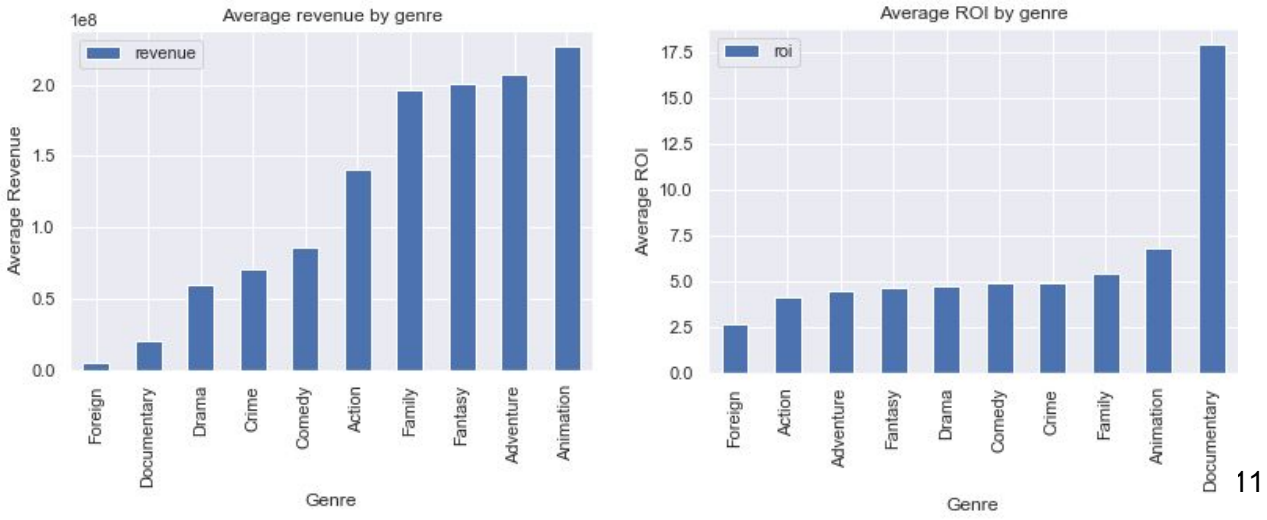
Relationship Between Top Director and Revenue/ROI



A top director is defined as any director appearing on highest grossing actors list on IMDB. Similar to hiring a top actor, a top director also results in a much higher average revenue (~280M vs 75M USD). However, while the ROI is lower with a top director, it still almost twice as large as the ROI with a top actor. This could indicate that hiring a top director can result in higher revenues on lower budgets (i.e. higher ROIs), compared to hiring a top actor.

Similar to the analysis for top actors, the Mann-Whitney U test can be used to compare the differences in revenue and ROI since both samples follow a non-normal distribution. The test concluded that the revenues for both samples (movies with top directors and movies without) are from different population distributions. The same conclusion was reached for the ROI data. Therefore, there is a statistical difference between the two distributions, indicating that revenue/ROI values for movies with top directors are statistically likely to be different form the revenue/ROI values for movies without top directors.

Relationship between Genre and Revenue/ROI/Categorical Success



Animated movies appear to have the highest average revenue followed closely by family, fantasy, and adventure. Foreign films, documentaries and dramas appear to have the lowest average revenues.

Documentary films have the highest ROI by far, followed by animated movies. Most other genres (except for foreign) result in similar ROIs on average.

The chi square test of independence resulted in a p-value close to 0, thus rejecting the null hypothesis of independence between genre and categorical success. It can, therefore, be concluded that there is a statistically significant correlation between the two variables.

Model Building

Scaling of features

After removing vote_average, we have 3 continuous variables as features - vote_count, budget, and runtime. Since the range of these variables are very different, it is important to scale these variables prior to linear regression. All numerical variables were scaled to have a minimum value of 0 and a maximum value of 1.

Machine Learning

As mentioned earlier, the data will be used to predict 3 different outcomes: revenue (regression), ROI (regression), and categorical success (classification).

Predicting Revenue

The data was split into a training and test set. Each model was fit and cross-validated on the training data. 5-fold cross-validation was performed to calculate R^2 and RMSE. Model selection, parameter tuning, and feature selection was performed only based on the training data. The model's performance was compared to the test set only to assess it's predictive performance on unseen data and to ensure no overfitting was occuring. 5 different algorithms were compared to determine which yielded the best performance for predicting revenue: Linear Regression, kNN Regression, Random Forest Regression, Decision Tree Regression, and Gradient Boosting. Performance was based on the R^2 value and root mean squared error (RMSE) derived from the training set.

Base Models					
	Linear	**kNN**	**Random Forest**	**Decision Tree**	**Gradient Boosting**
Train R2	0.736	0.52	0.77	0.72	0.76
Train RMSE	7.5e+15	1.4e+16	6.4e+15	1e+16	6.7e+15
Test R2	0.731	0.56	0.78	0.61	0.79
Test RMSE	8.5e+7	1e+8	7.9e+7	1e+8	7.6e+7

Outlier and influence analysis was performed using boxplots and Cook's Distance and any high influence points were removed from the dataframe. The best-performing algorithms, linear regression, Random Forest, and Gradient Boosting, were then re-run after the removal of the outliers (results shown in table below). Random Forest Regression had the best performance on the training data, with the highest cross-validated R^2 value of 0.78 and the lowest RMSE of 5.9e+15. The model performed well on the test data as well, with a test R^2 value 0.77 and an RMSE of 7.8e+07.

Without Outliers			
	Linear	**Random Forest**	**Gradient Boosting**
Train R2	0.743	0.783	0.777
Train RMSE	6.8e+15	5.76e+15	6.0e+15
Test R2	0.735	0.76	0.77
Test RMSE	8.4e+7	7.9e+7	7.8e+7

Feature selection was performed based on the Gini importance derived by the Random Forest regression model. The model identified vote_count and budget as the most important features, with runtime far behind. However, just using these variables as features reduced the training R2 and increased the RMSE significantly.

Top_actor and top_director were identified as less important feature by the Random Forest model. Removing both features slightly worsened the R2 and RMSE values. Since the removal of these features did not significantly enhance model performance, the final model hyperparameters were tuned with top_actor and top_director in the final model. The final Random Forest model had a training R2 of 0.8 and a test R2 of 0.75 with 1111 estimators and a maximum tree depth of 50. The maximum number of features were determined by taking the

square root of the total number of features. In order to enhance model explainability, Shapley values were used on individual predictions to identify the most influential features for each prediction. An example for Prometheus is shown below.

From the Shapley values, the average impact of each feature on the overall model could also be calculated to determine overall feature importance. It should be noted that top_director was the third most important feature for the model in terms of the average Shapley values.

Predicting Return-on-Investment (ROI)

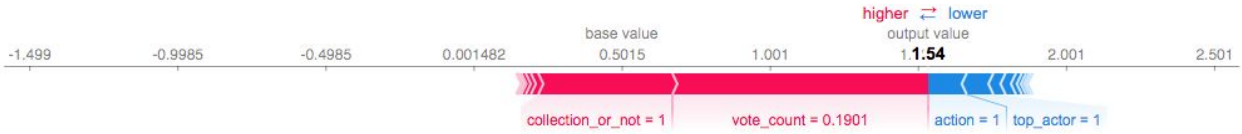
The same 5 algorithms were used to predict ROI, and their performances were compared based on the training R^2 and RMSE values, similar to the analysis conducted for revenue prediction. Outlier analysis was performed before any model building for ROI based on Cook's Distance. Budget was no longer included as a predictor since ROI was calculated as the ratio of revenue and budget.

Since the distribution of ROI was extremely skewed, none of the models resulted in a decent R2 value, with linear regression yielding a training R^2 of 0.006 and an RMSE of 412. EDA showed that log transformation of ROI resulted in a less-skewed and more normal distribution so this transformation was applied to the data prior to fitting the model. Log transformation of ROI in linear regression resulted in a higher R^2 of 0.13 and a much lower RMSE of 2.39. Log transformation was applied for all the algorithms with Gradient Boosting ultimately outperformed all other algorithms with a low cross-validated training R2 value of 0.2 and an RMSE value of 2.19.

Log-Transformed ROI Models Without Outliers					
	Linear	**kNN**	**Random Forest**	**Decision Tree**	**Gradient Boosting**
Train R2	0.13	0.12	0.133	0.18	0.2
Train RMSE	2.39	2.24	2.37	2.26	2.19
Test R2	0.14	0.12	0.15	0.15	0.19
Test RMSE	1.52	1.54	1.51	1.48	1.48

Feature selection was performed based on the Gini importance scores determined by the Gradient Boosting model. Vote count was the most important feature, with a steep drop in

importance between vote count and the next two important variables - collection_or_not and runtime. Top director and top actor were identified to be quite low in terms of Gini importance so the model was run with the removal of those features, one by one. However, the model R2 is already fairly low and removing either of the features resulted in a slightly lower R2 and a slightly higher RMSE so the features were ultimately retained in the final model. The low R2 value indicates that it is likely that the predictor variables simply don't explain the variance in ROI very much. The final model had a training and test R2 of ~0.2. Once again, Shapley values were used to determine the local feature importance. An example for Mission: Impossible is shown below.



Predicting Categorical Success

As mentioned earlier, categorical success consists of 3 categories: Flop, Breakeven, and Hit. In order to best predict the outcome of a movie, the following classification algorithms were compared: kNN Classification, Logistic Regression, Random Forest Classification, Gradient Boosting Classifier, and Linear and Non-linear SVM. Prediction performance was measured in terms of accuracy, precision and recall.

Classification Models						
	Logistic	**kNN**	**Random Forest**	**Gradient Boosting**	**SVM (Linear)**	**SVM (Non-Linear)**
Training Accuracy	0.54	0.46	0.51	0.55	0.5	0.47
Test Accuracy	0.52	0.48	0.5	0.53	0.5	0.47

The Gradient Boosting Classifier demonstrated the best performance with an overall cross-validated training accuracy of ~55%. The average precision and recall scores were ~0.53 with the model having the hardest time predicting movies that broke even. The precision and recall scores for this class were ~0.43, probably due to a higher likelihood of misclassification as a result of being the middle category. The most important features, based on the Gini Importance, were identified to be vote_count, collection_or_not, and runtime with vote_count being significantly more important than the other two. While top_actor and top_director were deemed less important features, removing these features did slightly decreased the overall accuracy, precision or recall scores. Therefore, the features were retained in the final model. The final model had a training accuracy of 0.55 and generalized well to unseen data with a test accuracy of ~0.52.