INDENG 142 FINAL PROJECT: Predicting Movie Revenue

Team Members: JiHo Bang, Ritvik Gadhok, Siri Phaneendra, Napoleon Vuong, Michael Yun

Motivation:

The objective of this project is to be able to predict movie revenue based on several movie attributes (e.g. release dates, genres, popularity, budget). In doing so, we can discern any patterns that influence how much revenue a movie will generate during its theatrical run. Producing a movie is a significant investment, both in terms of time and money, and studios and producers need the right tools to assess the potential returns on their investments. In an industry where the stakes are high, forecasting revenue offers an edge to directors, producers, and production companies to help them make informed decisions on resource allocation and investments to maximize profitability. Ideally, if revenue is not the sole focus of making movies, it will allow for more creative projects in the industry, enable companies to show diverse stories, and give more opportunities to smaller film-makers, actors, and set designers. By creating a model that predicts movie revenue, we aim to provide valuable insights to filmmakers, production studios, and investors, and help them make more informed decisions about which movies are worth their time. This project aims to reduce the financial risks and maximize the film's revenue by employing a data-driven approach. In general, our motivation is to contribute to the growth of the film industry by ensuring that there is improved decision-making for all stakeholders involved in a movie's production.

Data:

We sourced data from three distinct online repositories, each providing unique insights into the film industry. The primary datasets used were: IMDb, TMDB, and Box Office data, each with its own set of characteristics and information richness.

The first phase of our project involved an extensive data collection process. We began by extracting data from IMDb on Kaggle, one of the most iconic movie databases from 1888 to 2023. This dataset provided us with a historical to contemporary view of 7,414 movies, complete with revenue values and various other attributes such as the movie's genre, language, production company, etc. Our second source was the TMDB dataset, also from Kaggle, which offered metadata for an expansive collection of over 700,000 movies, consisting of various information such as the movie's cast, director, runtime, release date, etc. From this, we focused on a subset of 4,850 movies, again emphasizing non-null revenue figures. And, the final piece of our data collection puzzle was the Box Office data, obtained through web scraping techniques, from the "All Time Domestic Inflation Adjusted Box Office" website, which comprises 17,500 movies with its box office revenue. The web scraping script used the BeautifulSoup, Requests, and Pandas libraries to automate the data scraping to parse the HTML content to retrieve the relevant information and extract and store data from the multiple web pages into a dataframe.

Following data collection, we embarked on the critical task of data preprocessing to ensure the quality and consistency of our datasets. Our initial step involved standardizing movie titles by converting them to lowercase and removing non-alphanumeric characters. This process was significant in maintaining uniformity across datasets. To further refine our data, we eliminated duplicate entries based on unique identifiers such as "imdb_id" and "original_title_lowercase"

columns. This step was crucial in ensuring the accuracy and reliability of our analysis. The three datasets were then merged, creating comprehensive columns such as “all_revenue” and “all_budget” columns. This unification brought together disparate pieces of data to form a more complete picture of the film industry's financial landscape.

Data filtering and engineering were also pivotal components of our preprocessing phase. We performed typecasting to ensure that all data types were appropriate and consistent. Furthermore, we filtered the dataset to focus exclusively on English-language films, thereby maintaining a specific scope for our analysis, excluding non-English-language films such as Korean and French films. An intriguing aspect of our preprocessing was the extraction and cleaning of genre data using regex. This step not only streamlined the genre information but also enhanced its usability for analytical purposes. We also introduced a seasonal dimension to our dataset by creating a function, ‘season_categorizer,’ which categorized months into corresponding seasons (Winter: December, January, February | Spring: March, April, May | Summer: June, July, August | Fall: September, October, November). This seasonal categorization was based on movie release dates (i.e. month) and added another interesting layer to our analysis. Finally, we refined our dataset by including only those films with non-zero and non-NA revenue values. This filtration ensured that our analysis would be based on meaningful and substantial revenue figures.

The processes of data collection and preprocessing were important in setting a strong foundation for our predicting movie revenue project. By gathering and preprocessing data from IMDb, TMDB, and Box Office sources, we were able to construct a dataset that was not only comprehensive, but also tailored to our analytical needs.

Analytics Models:

To achieve our goal for this project, we decided to build various models such as a linear regression model, CART models, Random Forest models, and a Boosting model, using supervised learning techniques. The first model we built was a simple linear regression model, using P-values and VIF scores to optimize the model and avoid issues regarding collinearity. During this process, we tested the strength of specific variables using variables with p-values under 0.05 and avoided multicollinearity by examining variables with VIFs under 5. After selecting our features, we found that the strongest predictors for revenue were the popularity metric, vote count, the season summer, and certain genres like comedy, animation, and family. This led us to an r-squared value of 0.557 and an OSR^2 value of 0.62. Moreover, the RMSE for the linear regression model was 73664345.95. A more detailed view of the results of this model can be seen in Figure 1. These initial results led to optimism due to outperforming the baseline models in every metric. Of course, the 0.62 OSR^2 was significantly better than 0.0, but the RMSE also outperformed the baseline (which had an RMSE of 120000832.45) by 46336486.55, a significant amount. However, we also realized that there was room for improvement in terms of predictive power and this was likely because the relationship between predicting revenue and the data is more complex than just linear.

Due to the results of the previous linear regression model, we proceeded to build a CART model. First, we specified the hyperparameters ccp_alpha (0, 0.10, 201), min_samples_leaf (5), min_samples_split (20), and max_depth (30). We used GridSearchCV to perform 10-fold cross-validation and find the optimal hyperparameters based on the negative mean absolute error

as our scoring parameter. We then trained the decision tree regression on the training dataset using these hyperparameters. As we are using RMSE to evaluate the different models' performances, we calculated the RMSE from the negative mean absolute error and plot it in a scatter plot against various ccp_alpha values. We noticed that the graph showed a horizontal line, indicating that as the ccp_alpha values change from 0 to 0.10, the RMSE is not affected and is constantly 14696698.76 throughout (Figure 2). We also calculated our OSR^2 to be 0.982 by using R^2 as an alternative scoring parameter. This was very close to 1, indicating a very high level of predictive accuracy. To analyze this further, we plotted the distribution of training and test values with two overlaid histograms (Figure 3). Based on this, we determined that the y_train and y_test distributions both have a right-skewed distribution and are very similar to each other, which explains why the OSR^2 was very high.

Though the results from the CART model looked promising, we wanted to see if a more robust implementation of the decision tree model via a random forest method could improve our model performance. To do this, we conducted hyperparameter optimization using GridSearchCV. The hyperparameters we set for the model included 'max_features' (iterated through values 1-27, 27 being all columns in our dataset), 'min_samples_leaf' (5), 'n_estimators' (500), and 'random_state' (88 for result reproducibility). We also conducted 5-fold cross validation through GridSearchCV and found the OSR^2 and RMSE values for the model by defining the scoring method to be r-squared when optimizing for highest average OSR^2 on the validation sets and negative mean squared error (MSE) for the lowest average RMSE value on the validation sets. The model with the scoring method on r-squared had the best hyperparameters set to 'max_features': 26, 'min_samples_leaf': 5, 'n_estimators': 500, 'random_state': 88 with an OSR^2 of 0.982 (refer to Figure 4). The model with the scoring method on negative mean squared error had the same resulting best hyperparameters and resulted in an RMSE of 14245684.55 (Figure 5). This result shows a slightly improved performance compared to the previous best performing decision tree model with an RMSE value of 14696698.76, thus making this Random Forest Model the now best performing model amongst the baseline, Linear Regression, and Decision Tree Models.

Another model we built was a Gradient Boosting Regressor. To do this, we created a Gradient Boosting Regressor instance and set our hyperparameters to n_estimators=200, learning_rate= 0.1, max_leaf_nodes=100, max_depth=100, min_samples_leaf=10, random_state=88, verbose=1. After some experimentation, we decided that these parameters would give us the highest overall OSR^2. The model was then trained on a dataset, represented as X_train_dummies for features and y_train for the target values. The model's performance was then evaluated using OSR^2, where we achieved an OSR^2 of 0.978. We thought this could be improved, so we also conducted 5-fold cross validation through KFold and found the OSR^2 and RMSE values for the model to be 0.964 and 17357157.87, respectively. This ended up performing worse compared to the Gradient Boosting Regressor model.

Additionally we ran into "memory" issues for the grid search cross validation that did not allow us to iterate through intervals of n_estimators = np.linspace(50, 75*50, 75, dtype='int32') and

max_leaf_nodes = np.linspace(2, 10, 9, dtype='int32'). This led us to try and generalize the model via K-fold cross validation with k=5 to minimize.

Overall, our final best model was the decision tree with an OSR^2 of 0.982 and an RMSE of 14696698.76. Even with Cross Validation, the model still held suspiciously large test set accuracies, beating both our random forest and boosting model. We believe that this is due to our data distribution across our Train-Test split, where both the training and testing sets had nearly identical distributions of data points (as seen in Figure 3). Even when changing random states and ensuring no stratification in the train test split, this still led to similar results.

In the future, we can extend our analysis by incorporating data from other data sources and look at other features that are more customer centric, such as viewer preferences and trends. In addition, we could break the data up into industries to get more granular insights.

Impact:

By predicting movie revenue, we believe our project could have a positive impact on the film industry. With the recent "Hollywood" strike in 2023 that involved the Writers Guild of America and Screen Actors Guild of America, we have seen how important it is for workers within the film industry to have a consistent source of work and projects. Accurate predictions on revenue would allow production studios to better gauge how much of a budget they should be allocating for their next project. This would ideally lead to faster production for projects which would result in more content, more revenue, and a more optimal, well-rounded movie release schedule. On the other hand, this could also introduce some negative consequences. If the model is able to create perfect predictions, companies may begin to converge on the type of movies that are created solely for profit. For example, if revenue is maximized when certain actors are included and a certain genre is correlated with higher revenue, then the over-saturation of that actor and genre may lead to less creative efforts from production companies and higher effort just to achieve the factors that lead to higher predicted revenue. This would lead to the opposite of the desired goal of trying to create a more positive work environment and inclusive film industry.

Our project mainly affects two groups in different ways. For the production companies, our model would optimize resource allocation, provide a schedule for marketing and releases, and minimize financial risks for projects. For customers, the more evenly-distributed movies throughout the year and ideally cheaper cost of movies allow for more and cheaper movies to be enjoyed. To expand the scope of our analysis, we could gather more data on smaller studios, actors, and independent film-makers. Understandably, there is currently less information on this lesser-known population. However, if we were able to put more emphasis on them, we could make more accurate predictions and provide insights that would benefit this crucial, yet often overlooked part of the film industry.

Final Notebook:
https://deepnote.com/workspace/ritvik-gadhok-c6335963-8997-440b-a872-8ea1cf1d59cc/project/IEOR-142-926459a9-a790-4e67-bb88-49003f77b39f/notebook/Final%20Notebook-3347da30f4704f56b9d6d6c58fa4558d

Sources of Data:

1. IMDb: https://www.kaggle.com/datasets/komalkhetlani/imdb-dataset/data
2. TMDB: https://www.kaggle.com/datasets/akshaypawar7/millions-of-movies
3. Box Office: https://www.the-numbers.com/box-office-records/domestic/all-movies/cumulative/all-time-inflation-adjusted

Appendix:

Figure 1: OLS Summary

OLS Regression Results

Dep. Variable:	all_revenue	R-squared:	0.554
Model:	OLS	Adj. R-squared:	0.553
Method:	Least Squares	F-statistic:	379.3
Date:	Sat, 16 Dec 2023	Prob (F-statistic):	0.00
Time:	01:51:29	Log-Likelihood:	-53906.
No. Observations:	2753	AIC:	1.078e+05
Df Residuals:	2743	BIC:	1.079e+05
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.082e+09	3.54e+08	14.348	0.000	4.39e+09	5.78e+09
all_budget	0.4624	0.051	9.087	0.000	0.363	0.562
vote_count_meta	3.896e+04	1520.299	25.624	0.000	3.6e+04	4.19e+04
Released	-2.5e+06	1.78e+05	-14.070	0.000	-2.85e+06	-2.15e+06
Rank	-1.051e+04	541.175	-19.429	0.000	-1.16e+04	-9453.363
season_Summer	9.567e+06	3.5e+06	2.733	0.006	2.7e+06	1.64e+07
genres_meta_Adventure	1.351e+07	6.12e+06	2.207	0.027	1.5e+06	2.55e+07
genres_meta_Animation	3.857e+07	1.09e+07	3.553	0.000	1.73e+07	5.99e+07
genres_meta_Comedy	1.382e+07	3.69e+06	3.749	0.000	6.59e+06	2.1e+07
genres_meta_Family	3.77e+07	1.37e+07	2.762	0.006	1.09e+07	6.45e+07

Omnibus:	2230.519	Durbin-Watson:	1.913
Prob(Omnibus):	0.000	Jarque-Bera (JB):	109240.072
Skew:	3.465	Prob(JB):	0.00
Kurtosis:	33.072	Cond. No.	1.27e+10

Figure 2: DTR Validation Results

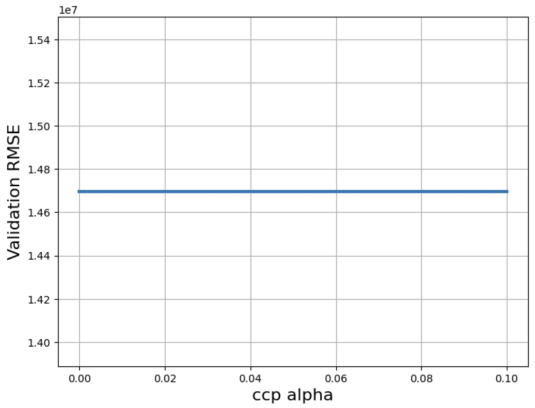

Figure 3: Train Test Distribution

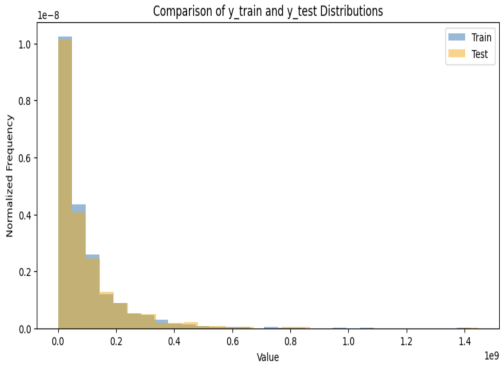

Figure 4: RF Regressor Validation R2

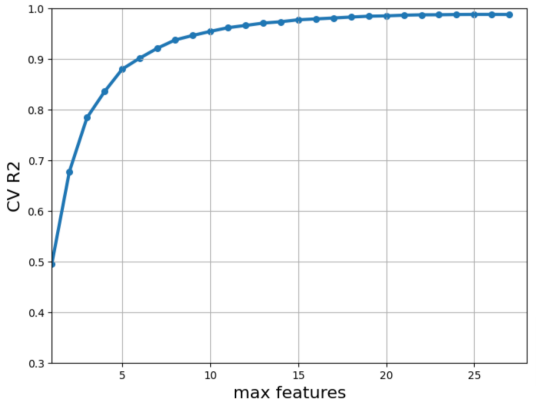

Figure 5: **RF Regressor Validation RMSE**

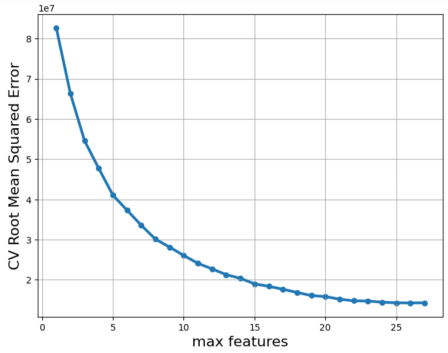

Figure 6: Model Results

model object	OSR2 float64	RMSE float64
Baseline	0	120000832.5
Linear Regression	0.982	73664345.95
Decision Tree Re...	0.987	14696698.76
Random Forest	0.982	16140461.14
Gradient Boosting	0.978	17357157.87