Predicting Movie Box Office Gross Revenue: An In-Depth Analysis

Ayato Hisanaga, Geethu Devarajan, Ivy Chen, Madhusmita Oke, Shan Ming Gao, Vera Hu

Information School, University of Washington, Seattle

IMT 574 A: Data Science II: Machine Learning

Prof. Chirag Shah

March 10, 2024[1]

[1] [links to Google Folders for supplementary material] Code and Extra data collected

Introduction

The project aims to use Machine Learning to predict the box office revenue of movies set to premiere in the United States between March 4th and 10th, 2024. We will be predicting the revenue for 13 such movies, as defined in the Appendix.

Importance of the topic: The project provides valuable insights for decision-makers in the entertainment industry. Accurate revenue predictions can significantly influence business outcomes by guiding decisions on advertising budgets, release strategies, and production investments, ultimately leading to increased revenue and profitability. Additionally, our analysis aims to uncover trends in movie performance, audience preferences, and the impact of various features on box office revenue. These insights can inform long-term business strategies and enhance decision-making processes for movie studios, distributors, and cinema chains. By predicting box office performance and understanding audience preferences, filmmakers can create movies that resonate more with audiences, resulting in more high-quality films that viewers can enjoy. This analysis can also extend to other aspects of the entertainment industry, such as digital streaming releases or TV channel movies. Moreover, with better predictions of revenue, producers and studios can better perform contingency planning for future resource allocation, aiding efficient decision-making.

Methodology

Step 1: Data Collection

Training Data: Due to time and scope constraints of the project, and to better capture recent trends and patterns in the movie industry, our group decided to limit the scope to movies released in the US market after 2010. Our data sources include:

- IMDb**:** We used a Python package "Cinemagoer", provided by IMDb, for retrieving financial data like gross box office and movie budget. We first collected the complete list of movie IDs from an offline dataset that is maintained and updated by IMDb daily, and then iterated through all the IDs using APIs in the Cinemagoer package to retrieve the data.

- TMDB: The Movie DB API, with a wrapper available on GitHub, was used to collect movie-related information such as movie title, original language, production companies, genres, IMDb ID and more. Data updates vary (see Limitations).

- Web Scraping: This technique was used to collect Oscar awards data from Wikipedia. Target data included the winners and nominees of every award category. Each year's winners and nominees were organized into a wikitable on their page. Due to the relatively consistent HTML layouts of these target tables across pages, the Requests and BeautifulSoup Python libraries were used to efficiently extract the target data.

- RottenTomatoes: We used RottenTomatoes to gather details like release date, genre and more for movies in the test set.

Test Data: The test data was gathered from the same sources above for movies set to premiere in the United States from March 4th to March 10th. Initially, the test data included a total of 28 movies. However, upon further investigation, we discovered that 12 of these movies were scheduled to debut at various film festivals in the United States and 3 movies were set to release on streaming platforms only. As our goal was to predict box office revenue for theatrical releases, we excluded these movies from our test set. The final test set consisted of 13 movies (listed in Appendix).

Step 2: Data Cleaning

After merging all the related rows of different data sets using the common feature - IMDb ID, we ended up with a final data set of 1418 rows and 14 columns. Cleaning was performed as specified below and model-specific transformations are mentioned later.

- **"Box_office"**: The original data from IMDb was in currency format. Stripped "$" and transformed it into numeric values
- **"Budget"**: This column contained 180 null values. Filled the null values with median value per genre
- **"genre_list"**: Transformed "genre_list" into binary values, with each genre as a single column, ie. one-hot encoding.
- **"release_date"**: Disaggregated into release year, month, and day of week.
- **Post-release data in the test set:** 'popularity', 'vote_average', 'vote_count' and 'ratings' will be available after release, so we estimated these values from genre medians and means.

Below is a table summarizing all our features:

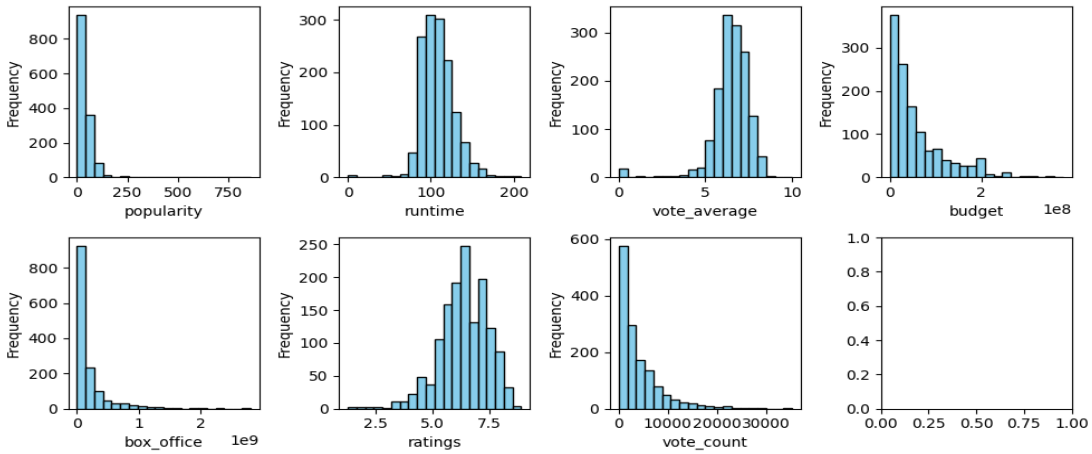
Feature	Description	Data Type
imdb_id	IMDb movie ID	int
original_title	Original title of the movie	string
release_date	Release date of the movie	datetime
budget	Movie budget in USD	float
runtime	Runtime of the movie in minutes	int
popularity	TMDB metric for lifetime user engagement	float
vote_average	TMDB movie rating	float
vote_count	# of user votes received by a movie on TMDB	int
ratings	IMDb movie rating	float
genre_list	A list of genres the movie belongs to	string
oscar_winner	Binary values, 1 meaning the movie won at least one oscar	int

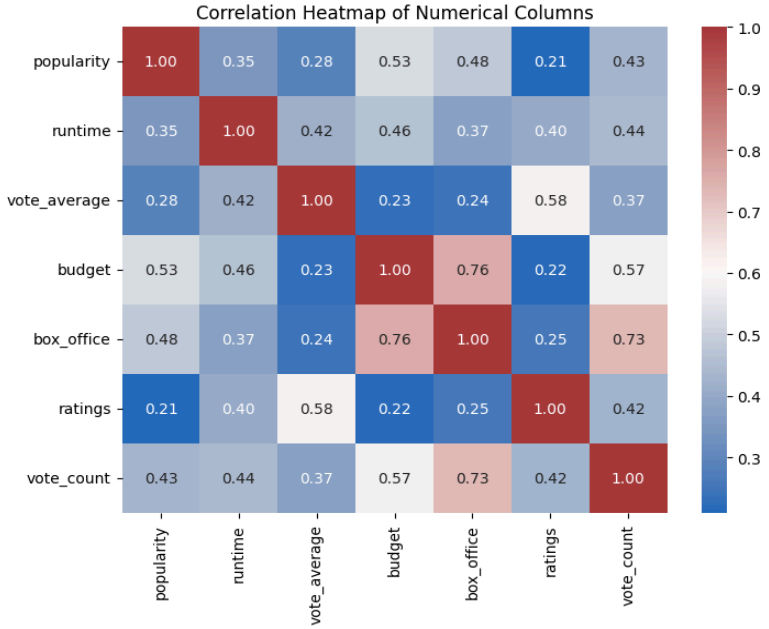
oscar_noms	Binary values, 1 meaning the movie got nominated for oscar	int
earn_class	Based on Return on Investment (%) (3 classes: loser, earner, super-earner). Super_earner at 90th percentile.	string
box_office	Target variable, global movie gross in USD	float

Step 3: Exploratory Data Analysis

Exploratory data analysis was important for identifying patterns to inform feature selection. This process generated insights about the underlying distribution, correlation, and trends within our data set.

Histograms revealed a blend of skewed and normal distributions within our numerical features:

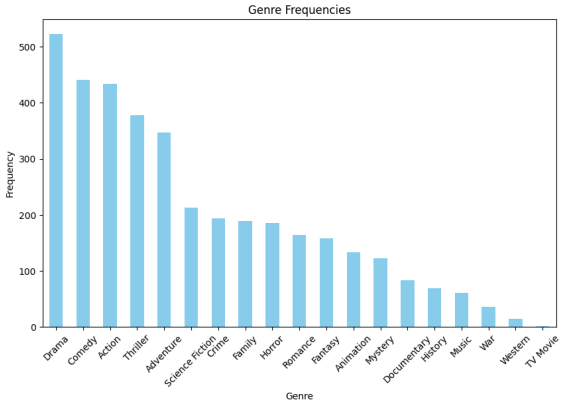


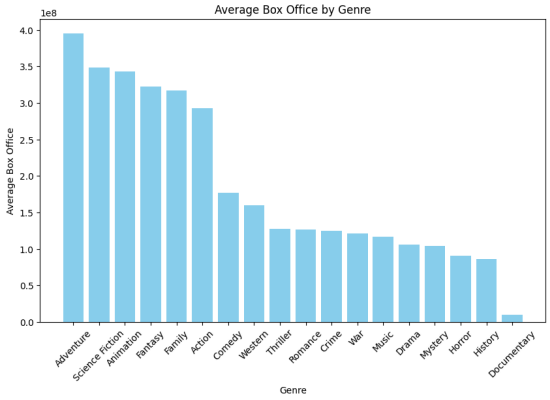


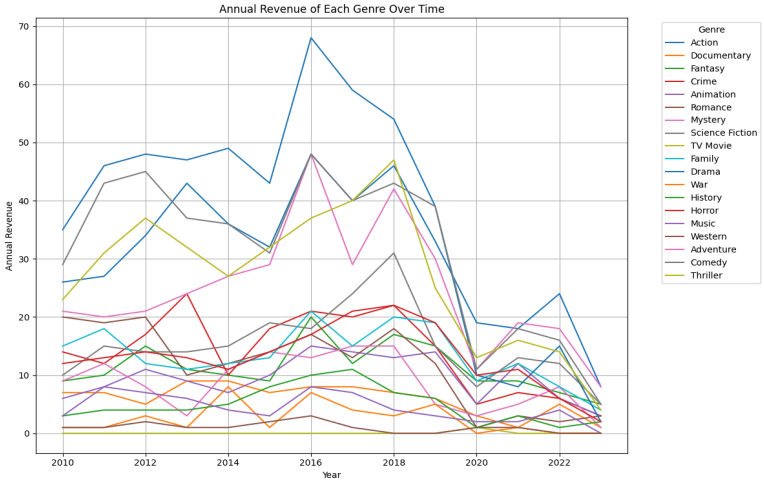
A heatmap of the correlation matrix among features was generated.

- Features most related to box office: budget, vote_count
- Features least related to box office: vote_average, ratings

Isolating each genre into its own binary column (one-hot encoding) revealed that the distribution of movie genres to be unequal:



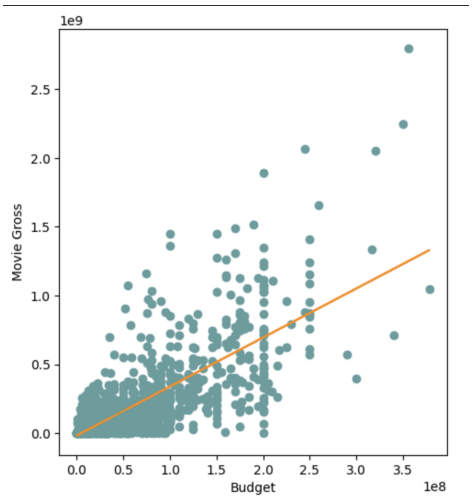
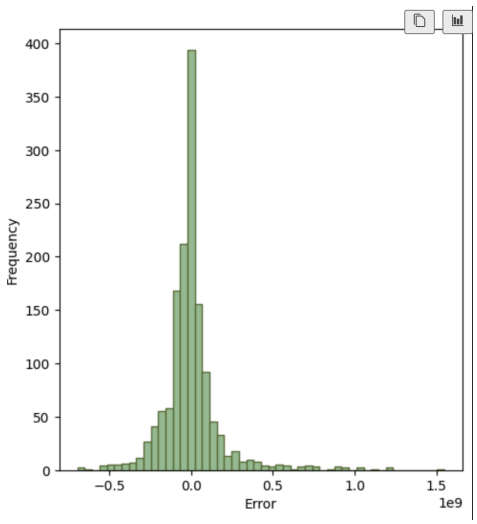




Moreover, plotting the annual revenue of each genre reveals a sharp dip across all genres in 2020 – most likely caused by the global COVID-19 pandemic:

Baseline model: We built a baseline regression model using only "budget" as the predictor variable to predict box office, which serves as a benchmark model for the future more advanced model. We chose "budget" because it has the highest correlation with the target variable.

- The model yields an R-square of 0.577, and RMSE of $182185565.69.

Model Fit (Gross Revenue by Budget)	Model Errors
	

These revelations raised important questions about how to factor historical contexts, skew, crisis anomalies, and varying distributions into our model.

Comparative Analysis of Modeling

Overview of Modeling: Equipped with actual gross revenue values, we utilized supervised machine learning, training the model on historical to learn the relationship between various features and box office gross, and then using this trained model to predict the gross revenue of current movies. We adopted a series of regression and classification techniques known for their robustness in predictive analytics. The choice of continuous models—Linear Regression, kNN, and XGBoost Regressor—stems from the quantitative nature of our target variable: the gross revenue, which is a continuous financial metric.

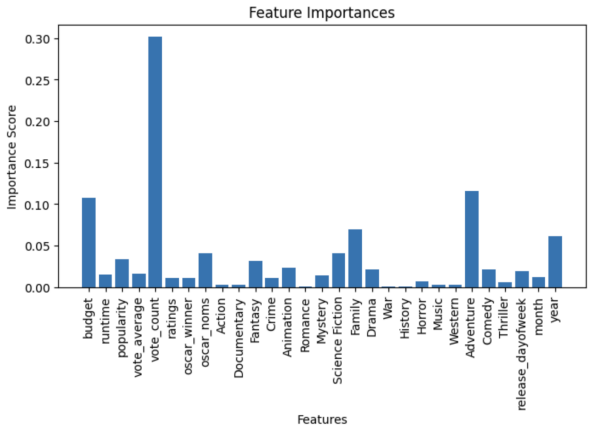
Linear Regression is the most intuitive model due to its simplicity, interpretability, and efficiency, especially in drawing direct relations between our predictors and the gross revenue. The kNN algorithm was considered for its ability to find natural clusters within the dataset, whereas XGBoost offers a gradient-boosting framework optimized for speed and performance. For a categorical interpretation of our primary question, we utilized Decision Tree and Random Forest modeling, both capable of tackling outliers and skew, as seen above in our EDA.

Linear Regression Model: Our linear regression model was meticulously built on a training set encompassing historical data of movies' performances. The model's effectiveness, reflected in an R-squared value of 0.747 (MSE approximately 2.01e+16) indicates that approximately 74.7% of the variance in movie revenues is explained by our model (represented below). The predicted revenues for the 13 upcoming movies ranged from $3 million to $534 million (see Conclusion). Factors such as budget and vote count had the greatest effect on the prediction.

```
Linear Regression Equation: box_office = 15511844.59771 + 807813963.96377*budget + 36575696.40068*runtime + 20629611
0.72043*popularity + -72835962.30318*vote_average + 1058997979.39201*vote_count + -72747988.09603*ratings + -4694063
7.98096*oscar_winner + -8180391.45946*oscar_noms + -24573943.51473*Action + 59326843.28092*Documentary + -53761995.05
986*Fantasy + -20324819.15919*Crime + 30724477.77278*Animation + 6642740.29434*Romance + -33062981.05622*Mystery + -5
9195732.40263*Science Fiction + 50257202.22367*Family + -222896.58249*Drama + -11181836.26747*War + -43417078.32489*H
istory + 19897896.27489*Horror + -857941.12350*Music + -128922191.82465*Western + 9474540.51989*Adventure + -4661680.
55390*Comedy + -12388640.77480*Thriller + 15718261.81953*release_month + 13075806.95771*release_year + -25725510.0139
0*release_dayofweek
```

kNN Regressor: When predicting box office revenue, a continuous variable, the kNN regressor is more suitable than the kNN classifier, for continuous versus categorical outcomes, respectively The kNN method groups movies with similar characteristics and performance trends, capturing complex, non-linear relationships between features and the target variable (box office revenue). Using kNN regressor, the model was trained with features including budget, popularity, vote count, runtime, oscar_winner, oscar_noms, and genre-specific features. The highest R-squared (R-squared = 0.576) occurred with k = 5, with an average MSE = 4.203E+16. Predictions below (see Conclusion).

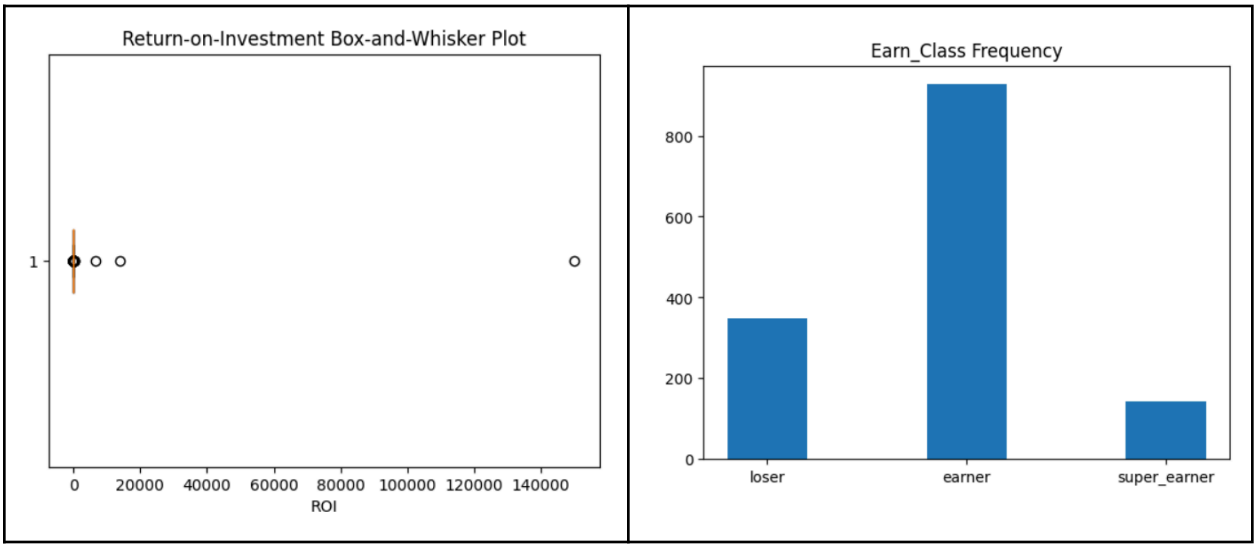
XGBoost Regressor: XGBoost Regressor is a gradient-boosted tree-based machine learning model used for regression predictive modeling. It works well with non-linear relationships present in the data, null values, and skewed data. The underlying principle behind XGBoost Regressor is Extreme Gradient Boosting. Gradient Boosting is a type of ensemble model used for classification or regression predictive modeling. Ensembles are constructed from decision trees – each tree added corrects the residual errors of the previous tree model. The models are fit based on differentiable loss function or gradient descent optimization. Regularization parameters used with the algorithm prevent overfitting such as gamma – minimize loss reduction for splitting on a leaf node, alpha (L1 regularization) – weight /importance of a particular feature, lambda – (L2 regularization) reducing complexity of model. Using XGBoost Regressor, the model was trained with features including budget, popularity, vote count and genre-specific features.



MSE: 2.023e+16, Avg R-squared value computed: 0.771, ie. 77.12% of the variance in the outcome can be explained by the model.

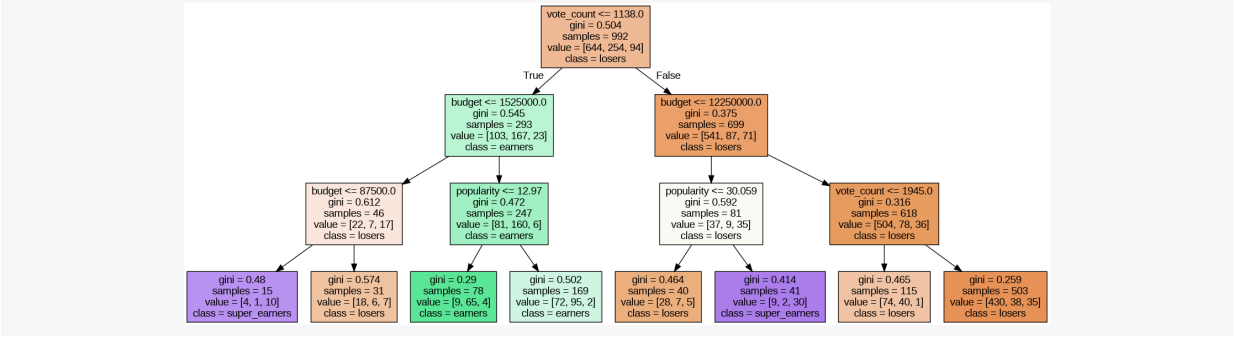
Decision Tree and Random Forest: Recall our objective to predict the gross revenue of movies in dollars. A less precise, but perhaps closer to the intended business value way of asking this question is to interrogate whether movies did or did not make up their budget investment. In other words, we can try to predict whether our movies are to be a positive or negative return on investment (ROI), "loser" and "earner" respectively. Additionally, as stakeholders may be interested in whether they have a record-setting revenue generator on their hands, we created a third category, "super-earner", for

predicting movies comparable to the 90th percentile of top ROI-earning movies in our training data.



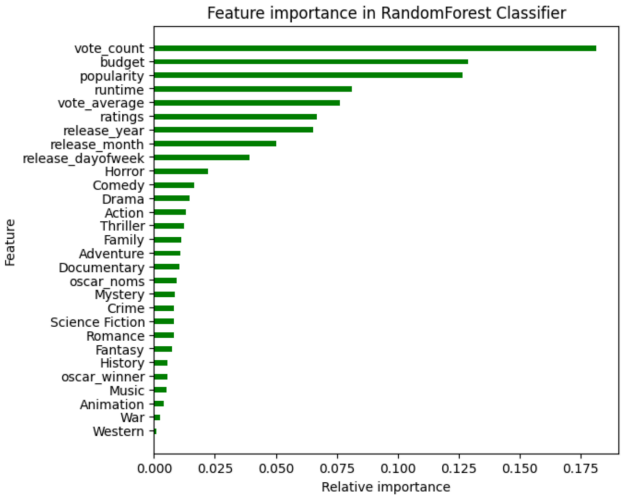
With a median of 1.441 (ie. about 44% profit) and a mean of 123.754, ROI exhibits substantial right skew. In our data, 24.6% of movies did not make up their budget (ie. the "losers"), while 65.4% did remake their budget and were less than the 90th percentile. To be of the 90th percentile, a movie had to have a ROI of 7.55 and above. Examining the top super_earners, high ROI's were often a result of low independent ("indie") movie budgets, rather than high popularity. With worries regarding outliers, the nature of decision trees and random forest models imparts more robustness by relegating outliers to distant nodes early in tree building[2].

One Decision Tree Instance (Accuracy = 0.758, Max Depth = 3)



Decision Tree: Our highest accuracy for our decision tree was about 0.75 with a max depth of 3 (max depth found through finding the most accurate depth from 1 to 10, over 5-10 iterations). The decision trees consistently showed that vote_count, budget, popularity, and ratings yielded the most useful initial information gain. A more systematic examination of feature importance below through random forest modeling.

[2] https://stats.stackexchange.com/questions/187200/how-are-random-forests-not-sensitive-to-outliers



movie	earn_class
Love Lies Bleeding	earner
Kung Fu Panda 4	earner
Imaginary	earner
Cabrini	earner
Accidental Texan	earner
The 3_2 Pulldown	earner
American Dreamer	earner
Night Shift	earner
5lbs of Pressure	earner
The Ballad of Davy Crockett	earner
Space: The Longest Goodbye	loser
The Piper	earner
Glitter & Doom	loser

Random Forest: When 1000 or 10,000 trees were used to build the random forest model, accuracy was likewise around 0.75 (0.756 and 0.749, respectively). A beneficial component of the random forest model is the ability to evaluate feature importances over the forest of 1000 trees; we see the same features considered important in the individual decision tree instances as well as runtime, vote_average, and features related to time of release. Genre features were relatively less important, and the inclusion of Oscar nominees or winners were not important to the model.

Limitations and Future Directions

1. ***Test Data and Incompleteness:*** As the test data included unreleased movies at the time of analysis, we faced difficulties in obtaining certain features for them. While 'popularity', 'vote_average', 'budget', 'vote_count', and 'ratings' were present for movies in the training data, these features were often missing for movies in the test set. To handle this issue, we organized the movies by genre and computed the median values for these features within each genre using the training data.
2. ***Feature Flux:*** In addition to the genre-based value substitution for nulls, even data that does exist for the upcoming test movies was often in flux due to the nature of the novelty of their movie pages on the data source sites. An expansion of our analysis in the future could include time-series data to understand and utilize how changes in values created slightly before, recently after, and longer after the release date relate to lifetime gross revenue. (For example, change in popularity before the movie releases and over the first 30 days). Additionally, we are limited by our data sources' methodologies for database updates[3] in ensuring data accuracy.
3. ***Gross Revenue, Movie Lifetime, Distribution Model***: The gross revenue of each movie has accrued over varying amounts of time depending on each one's release date. Thus, an expansion of our analysis could be to predict the gross revenue of a movie after a defined

[3] https://www.themoviedb.org/talk/597b9bc59251414bdb000d93 and https://www.themoviedb.org/talk/5c8215bd0e0a2643015febbc .

time (eg. 50 years), essentially predicting a gross revenue rate for the lifetime of the movie. Our analysis is also unable to accommodate the varying possible revenue streams of a movie available today, eg. theater release, streaming only, limited release, etc.

Conclusions and Takeaways

Summary of Predictions by Linear Regression, kNN Regressor, XGBoost Regressor
Note: We are most confident in predictions by XGBoost Regressor for reasons and with the limitations mentioned below.

movie	budget	LR Predict ($)	LR_ROI	kNNR Predict($)	kNNR_ROI	XGB predict ($)	XGB_ROI	RF Predict (earn_class)
Love Lies Bleeding	247800000	534591335	1.16	562267595	1.27	105777760	-0.57	earner
Kung Fu Panda 4	160000000	369112691	1.31	454976321	1.84	250660060	0.57	earner
Imaginary	13000000	91714640	6.05	35323376	1.72	100146830	6.70	earner
Cabrini	50000000	100119156	1.00	101824406	1.04	26659300	-0.47	earner
Accidental Texan	10000000	21756308	1.18	71654079	6.17	42818496	3.28	earner
The 3_2 Pulldown	1300000	70817789	**53.48**	46268189	**34.59**	72286784	**54.61**	earner
American Dreamer	40000000	66927417	0.67	116278073	1.91	79604320	0.99	earner
Night Shift	20000000	80041821	3.00	62175405	2.11	62907676	2.15	earner
5lbs of Pressure	30000000	107867255	2.60	21377546	-0.29	51473488	0.72	earner
The Ballad of Davy Crockett	75000000	235208981	2.14	9210767	-0.88	194965580	1.60	earner
Space: The Longest Goodbye	1550000	3654450	1.36	1522205	-0.02	16639981	**9.74**	loser
The Piper	10000000	86765660	**7.68**	35254967	2.53	95083088	**8.51**	earner
Glitter & Doom	28000000	65733534	1.35	75641268	1.70	36235776	0.29	loser

Legend:
loser
earner
super_earner

We used R-squared as the comparison metric for selecting the optimal model amongst Linear Regression, kNN Regressor, and XGBoost Regressor as it explains the variability in the data. Based on the comparison, XGBoost Regressor had the highest R-squared value (0.771) and thus was chosen as the best model for predicting gross movie revenues. As such, we expect 2 movies to gross less than their budget investment and the rest to earn back their investment (with 3 occupying super-earner status). For most movies in the testing set (7 of 13), there is congruence in all four models over whether the budget will be remade. As mentioned above, a substantial limitation is the fluctuation of engagement metrics, which likely have changed to varying extents since our data was pulled.

Applications of our predictions can benefit industry stakeholders, such as filmmakers and business owners to improve their content as well as adopt profitable business strategies according to the movie trends. This information can be used to create awareness among the masses by creating content/movies in popular genres. For instance, a documentary on climate change would not garner the required attention from the audience. However, using a popular genre or profitable one like a thriller or creative dramatic movie centered around climate change may attract a larger audience with a diverse demographic. Lastly, investment in engagement monitoring and streaming analytics may contribute to data for a more accurate, precise expansion of our analysis, improving future predictions.

References

1. Steven Skiena. (n.d.). Stonybrook.edu. http://www3.cs.stonybrook.edu/~skiena/591/final_projects/movie_gross/
2. kNN classification and regression.(n.d).Kaggle.com. https://www.kaggle.com/code/sashikanthreddy1598/knn-classification-regression

Data source references:

1. IMDb*:* http://www.imdb.com,
2. IMDb datasets: https://datasets.imdbws.com/
3. Rotten Tomatoes: http://www.rottentomatoes.com
4. The Movie Database (TMDB): https://www.themoviedb.org/?language=en-US
5. The Numbers : http://www.the-numbers.com/
6. Github: https://github.com/celiao/tmdbsimple
7. IMDB PY GitHub : https://github.com/cinemagoer/cinemagoer ->can look at readme for available list of parameters
8. Rotten tomatoes web scraper api https://pypi.org/project/rotten-tomatoes-scraper/
9. Cassese, M. (2018). *Movie Dataset: the 23 Best Data Sets Related to Cinema and TV.* https://www.lafabbricadellarealta.com/open-data-entertainment/

Appendix

Movies Defined for Prediction

(04MAR2024 to 10MAR2024 Theatrical Release)

1. Love Lies Bleeding
2. Kung Fu Panda 4
3. Imaginary
4. Cabrini
5. Accidental Texan
6. The 3_2 Pulldown
7. American Dreamer
8. Night Shift
9. 5 lbs of pressure
10. The Ballad of Davy Crockett
11. Space: The Longest Goodbye
12. The Piper
13. Glitter & Doom