Kyler Bailey, Collin Riddle, Whit Lupoli

CSC-272 Final Project Paper

Hollywood's Hidden Numbers: A Revenue Estimation Project

I. Introduction

Movie ratings and reviews have been one of the most crucial parts of statistical analytics in the movie universe. Our goal with this project was to predict the performance of movies based on a variety of criteria. With this in mind, we aimed to forecast overall movie performance by utilizing previous data to identify key insights into what influences a movie's revenue.

Our analysis focuses on data obtained from Kaggle, a comprehensive repository and data mining competition website. This dataset provided a plethora of attributes that we could use, some excellent, others entirely noisy. We found budget, genres, popularity, production companies, production countries, revenue, runtime, spoken languages, vote average, and vote count to be most applicable. Through these, we would like to discover trends that can encapsulate the wild and often unpredictable nature of movie revenue, further enhancing our understanding of how movies succeed. We wanted to use techniques learned in our data mining course whether that be linear regression, model trees, boosting algorithms, or random forest. At the core we desire to answer a couple primary questions: can a model reasonably predict revenue despite so much variation and external factors? Furthermore, what characteristics should a movie have to maximize its revenue?

In the end, we aim for a definitive and assured method by which movie lovers, like ourselves, or even producers, can answer these types of questions with confidence.

II. Dataset (https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset)

As mentioned previously, our dataset was sourced from Kaggle and provides a vast selection of statistics for over forty-five thousand movies. The dataset includes key attributes such as budget, genres, original language, popularity, production company, production country, production country, production country, revenue, runtime, and so on. But even beyond this, attributes like status, tagline, overview, and more existed but proved to be often noisy or

impractical. Many diverse reasons ultimately resulted in the removal of several attributes. To express this thoroughly, we created two tables detailing those we kept, and those we removed.

Attribute	Reason for Removal
Adult	Biased towards false.
Homepage	Not conclusive, simply website links.
ID	Overfitting
IMDB_ID	Overfitting
Overview	Not conclusive or predictive, very noisy. This is a long description for each instance.
Poster Path	Not conclusive, simply a JPG link.
Tagline	Not predictive, another long string of unique words.
Status	Biased towards “Released” - 99% in that one attribute value.
Video	Biased towards “false” - 99% in that value.
Original Title	Overfitting
Title	Overfitting
Belongs To Collection	90% empty
Release Date	Not central to our research question – very interesting for future consideration!
Original Language	Repetitive with spoken language.

Table 1: All attributes we removed and why we did so.

Attribute	Explanation
Budget (numeric)	Budget allotted for each movie.
Genres (nominal)	Genres of the movies (simplified down from original list.)
Popularity (numeric)	TMDB score quantified by engagement metrics like voting, daily views on TMDB, etc.)
Production Companies (numeric)	Number of companies that produced the movie.
Revenue (numeric)	Amount of gross revenue each movie earned.
Runtime (numeric)	Length of movie (minutes).
Spoken Language (nominal)	Language(s) each movie is in. Can be a single language, multilingual, or unknown.
Vote Average	Average vote score from TMDB.
Vote Count	Total amount of votes from TMDB website.
Production Countries	Countries the movie has been produced in. Classified as either national or foreign.

Table 2: Description of all attributes we kept and what they represent.

III. Preprocessing

Preprocessing was a fundamental step in our project because at the beginning the dataset could not be imported into Weka. This was caused by a variety of reasons, the two primaries being improperly formatted JSON attribute values, and foreign languages that corrupted the CSV data. Due to this, our best approach was to write Python scripts that could effectively convert the messy and dirty information into a clean and importable CSV file for Weka.

Before beginning the description of our Python scripting, I would like to note that ChatGPT-4-Turbo and Claude-Sonnet-3.7 were used as aids in debugging and improvement of

overall code design. That said, below is a table detailing each method, what that method does, and why the method was needed.

Method	Functionality
extract_json_values()	Identifies JSON formatted text and searches for a key, consider the attribute genre, as an example. As long as the data is not empty or corrupted, the code will check if the attribute is a list, and extract each item, in this example, every genre. If not a list, then the given instance only has one genre, and we extract the instance. This was done because JSON formatted information couldn't be properly imported into Weka, so we instead wrote this to aid with extracting the valuable information in these JSON strings.
clean_revenue()	Intended to ensure that any values in scientific notation are converted to standard form. This prevents conversion issues, ensuring these attributes are read as numeric (similarly to **convert_numeric()**).
convert_numeric()	Ensures that all numeric values are read as numbers and not strings (a persistent issue we encountered when importing into Weka).
extract_first_value()	Like **extract_json_values(),** this searches for a given key in a JSON string. The whole goal is to identify if the key is a list, meaning several values exist, and if so, only return the first one in the list. This way we don't have an instance with several attribute values, but just one.
count_production_companies()	Identifies a JSON list and iterates through, counting each instance and returning an integer value of how many production companies a movie has. This was necessary as many production companies corrupted the CSV (due to foreign languages or atypical

	formatting) and therefore this gave an interesting attribute that imported smoothly.
convert_to_multilingual()	Takes all listed languages, counting how many exist, and returns a binary value (single_language or multilingual). This prevents data corruption issues but also allows for a balanced attribute with potentially interesting results.
convert_to_national_or_foreign()	This searches for the U.S. in the **production_countries** attribute. The method returns either national (if produced in the U.S.) or foreign (if not the U.S.). Like above, this provides a more balanced attribute and eliminates data corruption issues from different languages or bad formatting.
remove_zeroes()	Takes in a function and rewrites the previous file into a new file depending on the function. In this case, we pass a function such that if revenue is zero, the instance is removed and not written into the new dataset. Most instances where revenue is zero are either empty, cause obscure exceptions, or are generally not helpful, especially since revenue is our class attribute.
process_csv()	This is the main function which opens the original CSV and writes into a new CSV. For each row in the original CSV, we run every helper method (see above) and as a result write into a new and cleaner CSV with all changes that the previous methods provide included.

Table 3: A table showing each Python script and their functions.

As seen above, cleaning the CSV through Python scripts was very involved and time consuming. But once imported into Weka, we noticed another problem: many numeric attributes like popularity, revenue, budget, and so on, were very skewed. This seemed problematic, so we tried an array of potential solutions to combat these skews.

We began by discretizing these numeric attributes by using equal-frequency binning in hopes of combating outliers and skew. But our classification algorithms seemed to still favor certain bins despite attempts to better spread our results. We could not quite source why, and we did not want to remove outliers as we believed these movies were still critical to our analysis.

We then tried log transformation in hopes that by creating a normal distribution we could combat the skew and improve performance down the road (in our analysis). However, spoiler alert, this only worsened the model. But how come? What we discovered was very interesting in that by performing log transformations, we actually obscured correlations between a given attribute and revenue. Put simply, transformations mess up our relationships and therefore our model as a whole! Consider Figures 1, 2, and 3 below. See how a clear positive linear relationship exists before transformation **(Figure 1),** yet after, the correlation is negatively affected and not as strong **(Figures 2 and 3).** Therefore, log transformation wasn't an option either. We ultimately settled on leaving the skew rather than attempting to correct it.

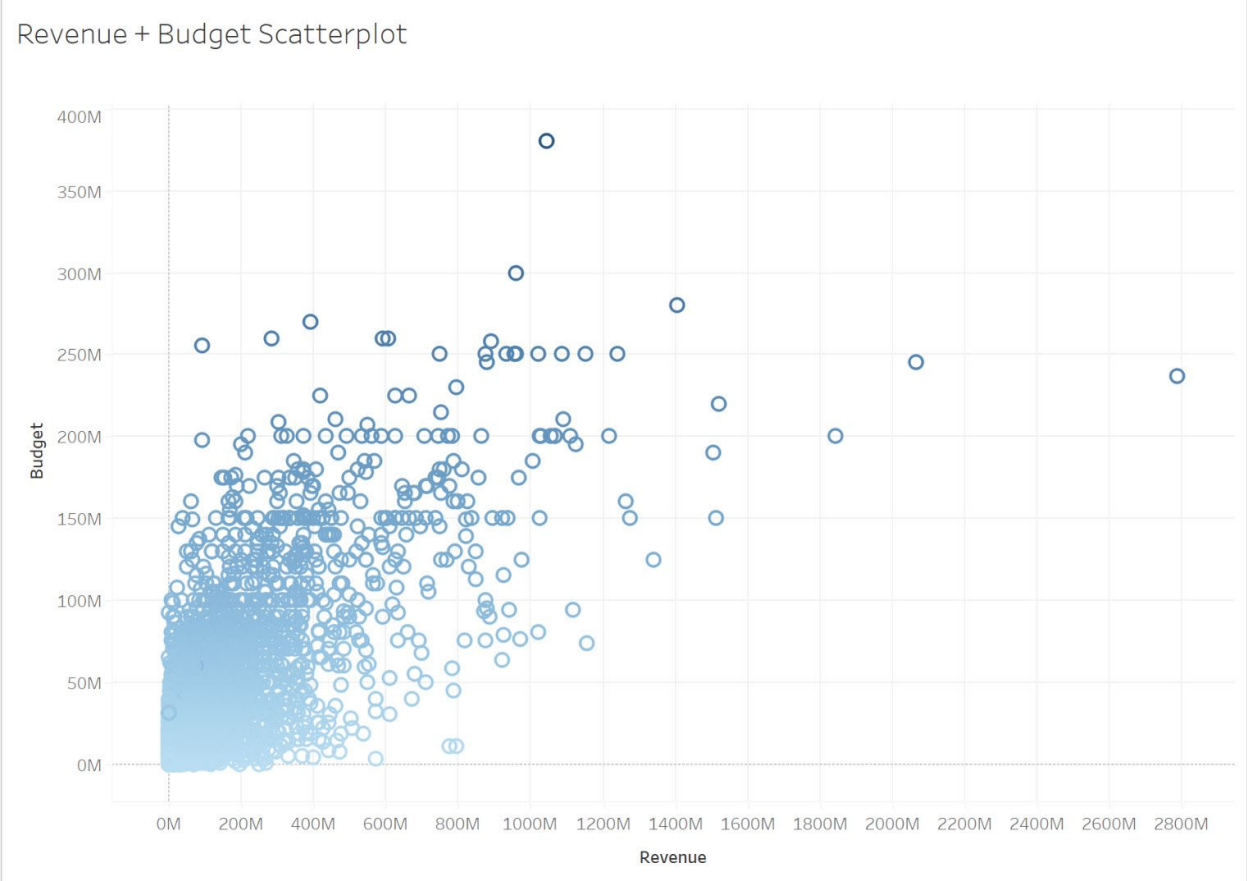

Figure 1: Scatterplot between budget and revenue without log transformations.

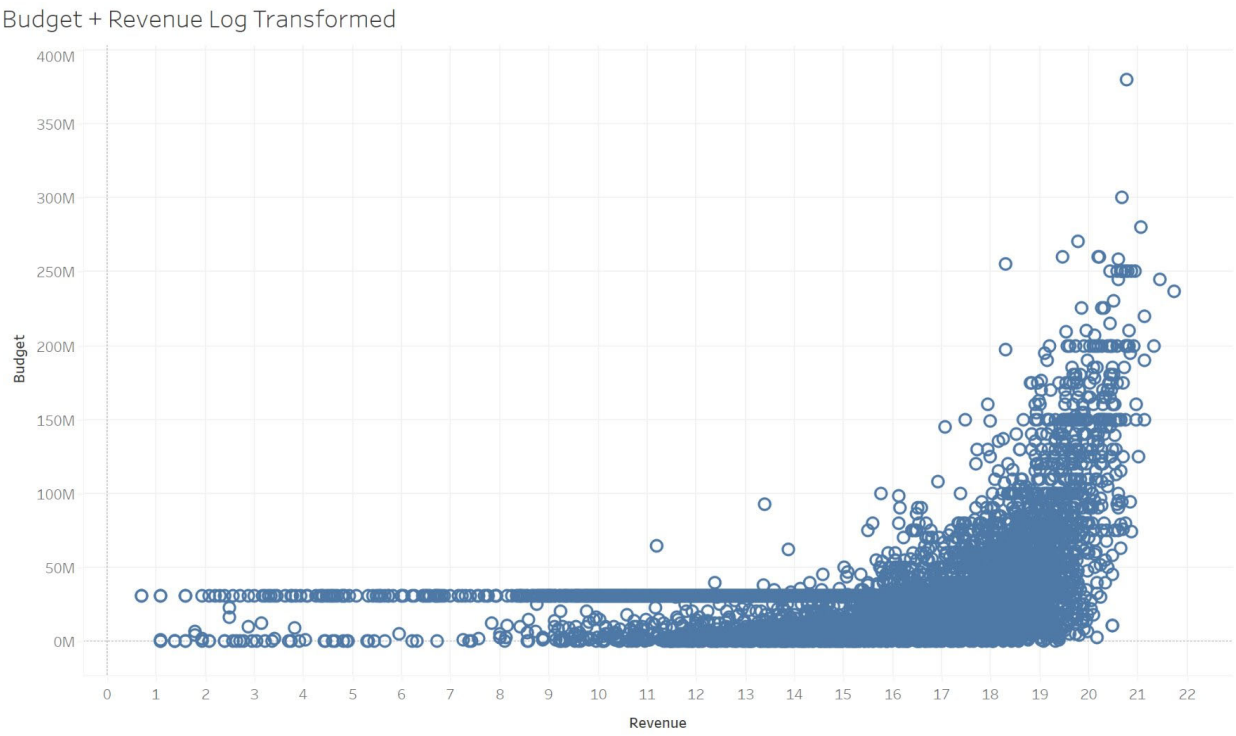

Figure 2: Scatterplot between budget and revenue with revenue log transformed.

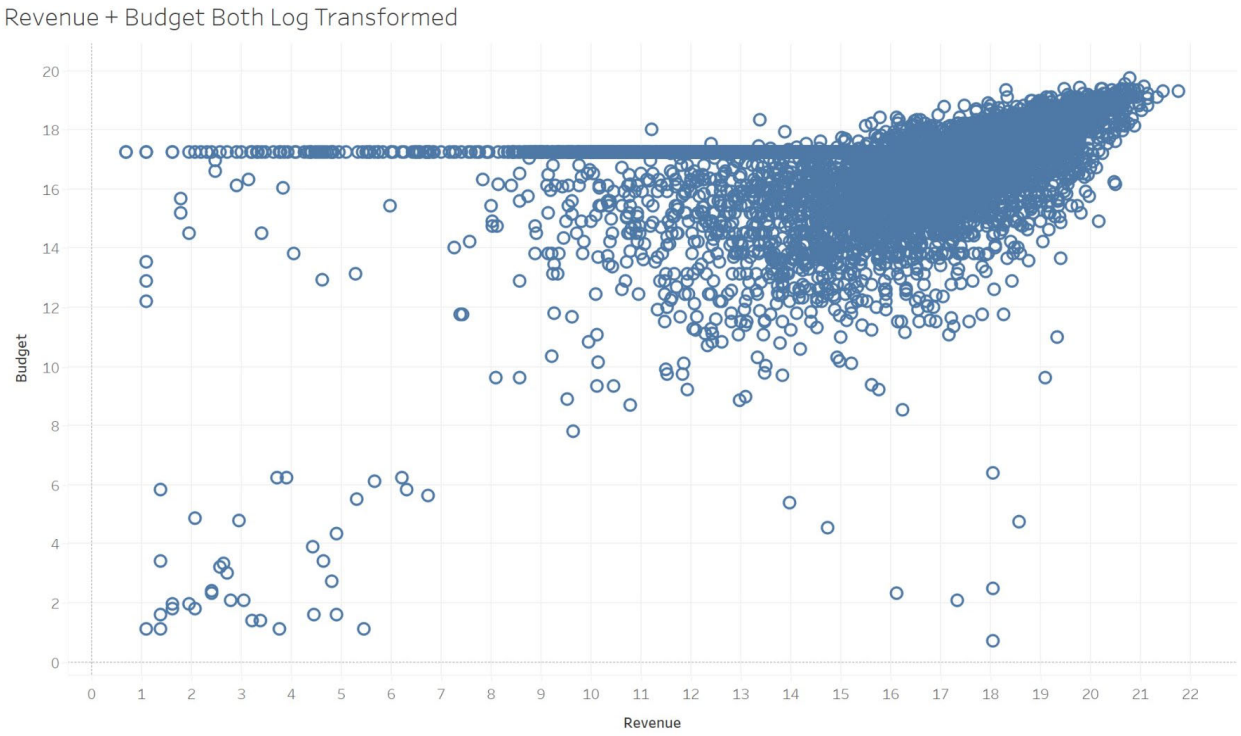

Figure 3: Scatterplot between budget and revenue with both attributes log transformed.

With our dataset appropriately cleaned, and a consensus to keep the skew, we completed our preprocessing and began analysis.

IV. Analysis

Beginning our process of analysis, we recognized that our available options regarding models were a bit more limited, given we chose numeric estimation. Before continuing, I would like to note that we did run a few baseline algorithms, like IBk, but they functioned very poorly, and I chose not to include them in our analysis as the results were not very helpful. Equally important to note is that we initially chose ten-fold cross-validation as the method of assessing the accuracy of our models. However, we concluded that using the holdout method on the data was more appropriate (after our presentation), given the data's vast size. Therefore, we trained on 4/5 of the data and tested on 1/5 of the data. This split was specifically chosen because this seemed to lead to better model performance overall. All outputs reflect this methodology for assessing the models.

Getting into the algorithms, we knew that a variety of very reliable options existed, one of those being linear regression. When we executed the algorithm, we received a relatively promising output (**Figure 4**).

```
Correlation coefficient                  0.8552
Mean absolute error               41804257.9858
Root mean squared error           75492085.5518
Relative absolute error                 51.9679 %
Root relative squared error             51.8409 %
Total Number of Instances            1482
```

Figure 4: Weka output using Linear Regression as the model.

There was a strong correlation, but our two error percentages exceeded fifty percent, and this was seemingly problematic as we certainly believed we could do better. What we determined as the primary issue for lack-luster performance was that a good number of our attributes violated linearity. Because of this, we figured that a linear regression model likely would not be the best.

Our second inclination was to choose something that might have higher toleration to linearity violations and big skews. We settled on Weka's Additive Regression algorithm which, given its boosting nature to continually improve upon previous models, we were hopeful. Through this, we figured boosting might be able to better handle the outliers. To an extent, we were correct! With a decent bit of classifier experimentation and crashing Weka a few too many

times by doing so (thanks Random Forest), we settled on the model below which is using M5P as a classifier.

```
Correlation coefficient                  0.8727
Mean absolute error                34204291.9888
Root mean squared error            71323224.7
Relative absolute error                 42.5202 %
Root relative squared error             48.9781 %
Total Number of Instances            1482
```

Figure 5: Weka output using Additive Regression as the model.

As you can see, an optimistic decrease in error rate occurred and a pretty substantial increase in the correlation coefficient as well. This was good progress, and certainly serves as a good model, but could we do better?

With a bit more experimentation we settled on the Random Forest algorithm in Weka. This is a good choice given the dataset because of the underlying nature to use bootstrapping and averaging, therefore creating robustness to outliers. Furthermore, Random Forest takes little concern with linearity violations, a previous issue we had run into. When running this algorithm, we opted for two hundred and fifty iterations as this did slightly improve performance. The resulting model was our best-to-date output, in fact, this was better than the one we used in our presentation (because this was tested with the holdout method). Boasting a strong 0.8981 correlation coefficient and our lowest errors, we settled on this specific model (**Figure 6**).

```
Correlation coefficient                  0.8981
Mean absolute error                32097483.8299
Root mean squared error            64404815.0384
Relative absolute error                 39.9012 %
Root relative squared error             44.2272 %
Total Number of Instances            1482
```

Figure 6: Weka output using Random Forest as the model.

After completing the above model, we wanted to know what attributes were best facilitating the process of predicting our class attribute revenue. To do so, we ran feature selection, specifically the WrapperSubsetEval method paired with the BestFirst search strategy and the primary predictive model within WrapperSubsetEval being the Random Forest algorithm. This combination was deliberately chosen over alternatives such as CfsSubsetEval

and GreedyStepWise. WrapperSusbetEval was favored because this algorithm evaluates subsets of attributes using the actual learning algorithm, allowing the algorithm to better capture how the selected features interact with the specific model: Random Forest. In contrast, CfsSusbsetEval selects features based on how highly correlated they are with the class attribute which overlooks the feature combinations.

Similarly, BestFirst was chosen over GreedyStepWise due to the algorithm's ability to explore the attribute space more thoroughly. GreedyStepWise progresses in a linear fashion and may overlook the best subset of overall features settling for the local optimum. Inter-relationships between attributes like production company, budget, vote count, and others proved to be both crucial and complex. The combination of WrapperSubsetEval and BestFirst thus enabled the creation of several features that were used in all ten-folds.

The results of feature selection yielded budget, production company, vote average, and vote count, indicating that they are the most influential in predicting a movie's revenue (see **Figure 7**). **Vote count** and **budget** have the strongest relationships, with R-squared values of **0.61** and **0.50**, respectively, indicating they explain a large proportion of the variability in revenue. **Vote average** and **production companies** also show moderate predictive power, each with an R-squared around **0.40**. All attributes have p-values well below 0.01, confirming that the observed relationships are statistically significant.

To further illustrate feature selection, see **Figure 8,** showing that **budget**, **vote count**, **vote average**, and **production companies** are all reasonable predictors of revenue (keeping in mind that skew does challenge these relationships). In all, these findings prove to be quite reasonable, as we'll explain further in the results section of this paper.

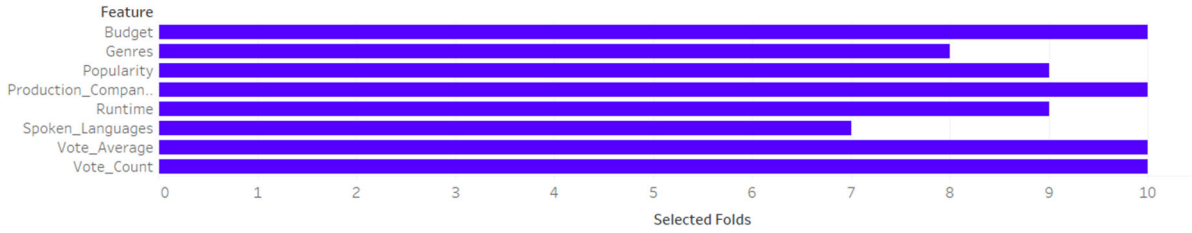

Figure 7: Bar Chart showing the total number of folds a given attribute was used in for the model.

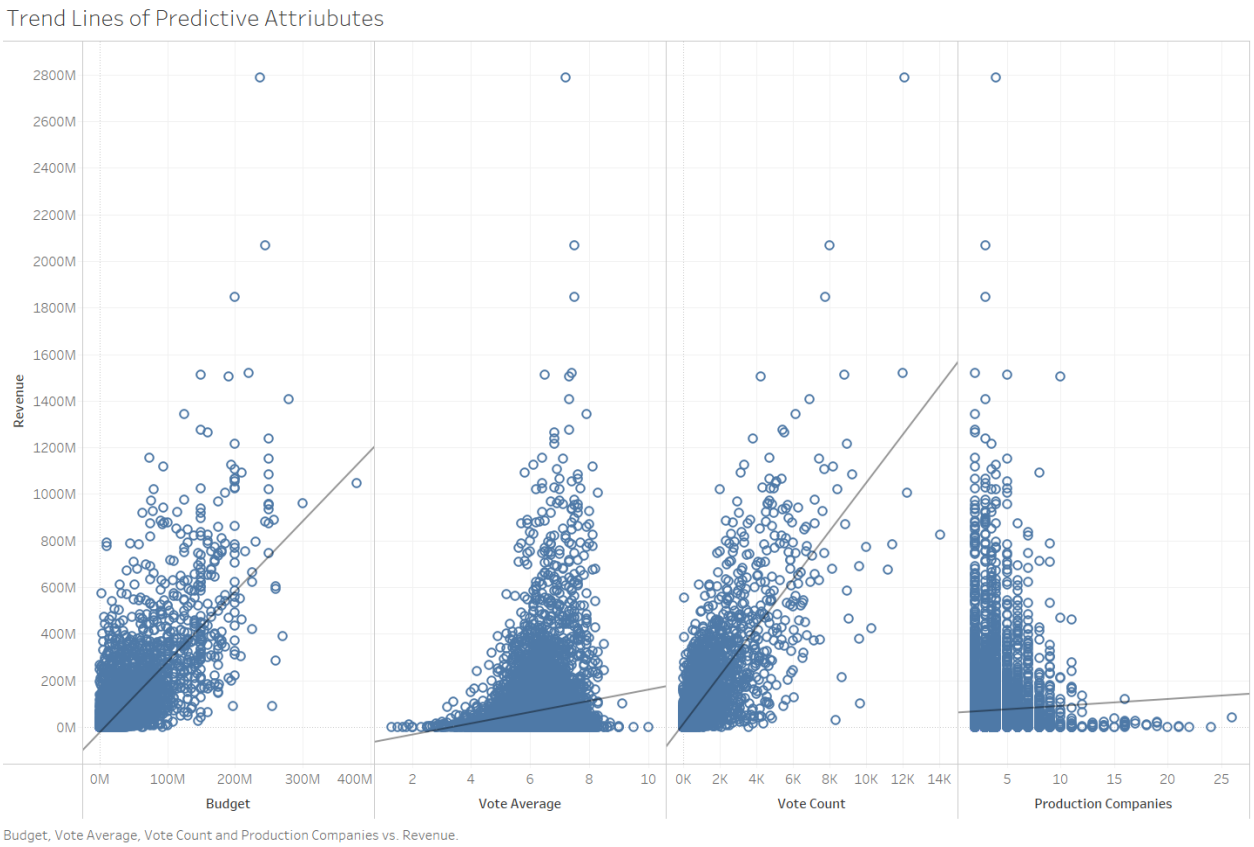

Figure 8: Scatterplot with trend lines showing relationships with revenue.

V. Results

Given all the information about model experimentation and predictive attributes, what are the key results? Well, let's begin with the model where we ultimately settled on Random Forest given the robustness to outliers and seamless handling of non-linear distributions. As seen in **Figure 6,** we got a strong correlation coefficient of 0.8981 indicating Random Forest was able to predict values quite closely to the actual ones. Moving down, we should take a moment to discuss what's going on with the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). When first seeing these values, we were exceedingly concerned about what this could imply and if something was seriously wrong with our models. However, after spending some time with our data and beginning to understand what these measurements were, we understood that these do make logical sense. But how so? Well, we found that MAE and RMSE are scale-dependent values, meaning they are measured upon the same scale in which the variable they are predicting. In our case, that is revenue, which explains why MAE and RMSE are so large. Revenue ranges anywhere from one or two dollars up to several billion dollars, therefore our MAE and RMSE will be inherently large.

That said, let's focus on our Relative Absolute Error (RAE) of 39.9012% and Root Relative Squared Error (RRSE) of 44.2272% as they are normalized and expressed in percentages, giving clearer insight into model performance. Both are below fifty percent, which is quite good! Of course, this is not perfect, and there are plenty of models out there with much better precision and accuracy. However, we discovered that movie prediction in and of itself is simply challenging! Movies themselves warrant unpredictable reactions. Sometimes movies are hyped up but then perform awfully in the box office whether that be the movie's fault or even something like a very popular competing release. Consider The Minecraft Movie, which had little to no anticipation, yet on the release weekend, they made over one hundred and sixty million dollars. These kinds of outliers or unexpected turn-of-events are hard to account for, but that's part of the challenge when trying to predict something like movie revenue.

Before concluding this analysis, we still need to provide results to a key question. What attributes, or characteristics, should a movie have to succeed? Recall previously that we ran attribute selection and yielded the following in all ten folds: budget, production company, vote average, and vote count.

For starters, the budget of a film often correlates directly with production quality, marketing, and overall visibility, which are crucial factors in driving box office performance. We theorize that higher budgets generally enable wider releases and more aggressive promotional campaigns, contributing to increased revenue potential. Production companies were also found to be significant. Certain studios have a proven track record of producing high-grossing films due to their access to superior resources, talent, and distribution networks. We speculate that this feature allows the model to account for organizational factors that often predict commercial success.

In terms of audience reception, the vote average serves as a proxy for viewer satisfaction, while the vote count reflects the breadth of a film's reach. A higher vote average would seem to indicate favorable reception, which can support sustained earnings through word-of-mouth and repeat viewings. Simultaneously, a high vote count appears to be indicative of a wide viewership, likely translating to higher gross earnings. Together, we can see why these features enhance the model's ability to predict both initial and long-term financial outcomes of cinematic releases.

VI. Conclusion

With this project, we have determined that building a model to reasonably predict movie revenue is hard but possible. We now understand that movies are variable and perform in unexpected ways. However, if we pay close attention to characteristics like their budget, who's producing it, and how their engagement metrics are doing (ex., vote count), then we might have a shot at predicting how much the movie will make, whether that be two dollars or two billion dollars.

Additionally, we would also like to speak about what we would change and how this project might be continued further, given more time. For starters, this dataset was quite messy, and while this dataset proved to have lots of interesting movies (that many datasets lacked), we would like to consider using a cleaner one in the future. This may aid in not only easier analysis but also allow us a bit more time in general as preprocessing was very involved and time-consuming.

Speaking more about the dataset, we want to consider the release date as we believe it likely has an impact on how much a movie makes. Adjusted for inflation, this would be interesting to see how movies and their revenue have changed over time. This is certainly a component we would like to consider in the future as it provides lots of interesting insights and may allow us to answer an array of different questions about movies more generally.

The last thing we'd recommend for anyone who may further continue this analysis is to consider streaming services and their original movies. We're curious to see if a movie's revenue may be affected by factors regarding streaming services. For example, does a movie being shown on streaming services and not in theaters have any correlation with revenue? What about if a movie is released in theaters but will quickly be available for streaming? What impact could this have? As seen in **Figure 8**, streaming services are quickly rising in popularity while movies in theaters don't seem to be. We believe that streaming services should be strongly considered for future analysis of movies as they're certainly becoming a bigger component today and could provide interesting insights.

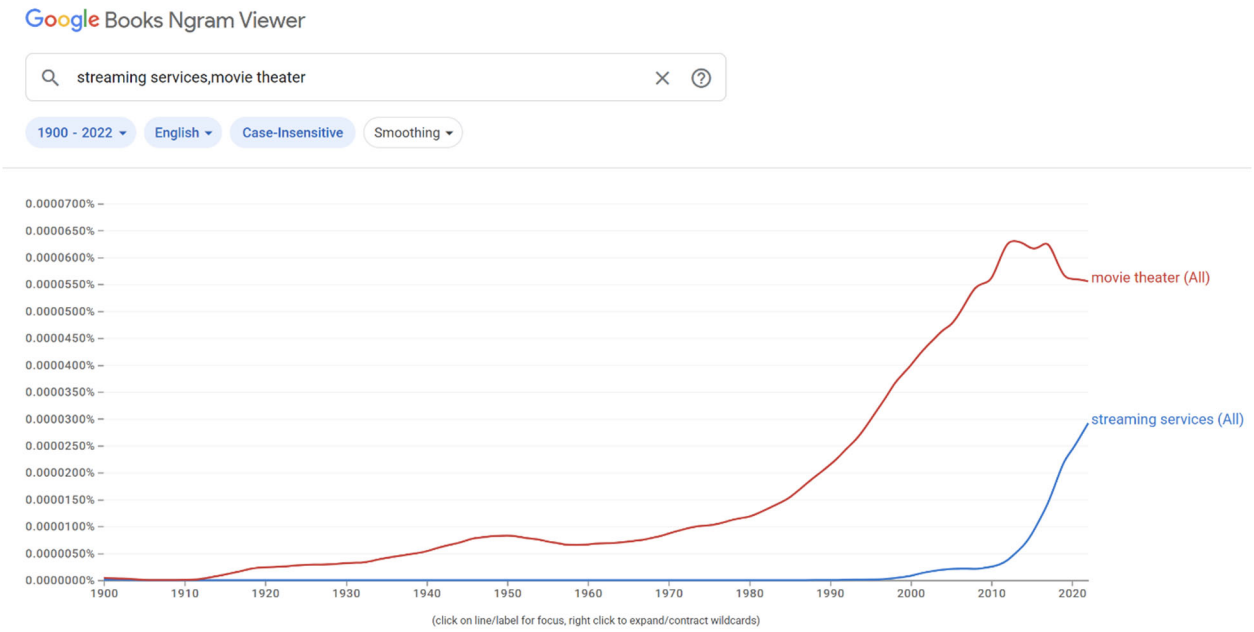

Figure 9: Google Ngram Viewer graph comparing the usage of movie theater and streaming services.

In all, we believe that our questions have been sufficiently answered, and we've found some interesting insights that tell us more about the hidden numbers of Hollywood.