CPSC 66 Final Report:
What Makes a Movie? Predicting Revenue Based on Film Features

Jayden Booker JBOOKER1@SWARTHMORE.EDU
Zixi Gao ZGAO1@SWARTHMORE.EDU
Natassia Lowe NLOWE1@SWARTHMORE.EDU
Opeyemi Ogundele OOGUNDE1@SWARTHMORE.EDU

Abstract

Making movies is no easy feat. Oftentimes, it can be hard to understand what aspects of a movie would make it a box office hit, and producers need to know what to invest their time in in order to make a great film that generates the most revenue. Is it the actors? The budget? The marketing? In order to answer address this problem, our group ran 4 distinct models—Neural Networks, Logistic Regression, K-Nearest Neighbors (KNN), and Random Forests—to figure out what feature of a movie carries the most significance in determining its revenue, and therefore, success. After running our algorithms, we found that the budget has the most weight in revenue prediction with our Random Forests and KNN models having the highest accuracy. Lastly, we detail the social implications of our work and possible directions for future researchers.

1. Introduction

The film industry is a highly competitive, multi-billion dollar industry with high expectations and demands. Movies are large and expensive projects that require dozens to hundreds of employees and funding from producers for props, sets, equipment, etc. Producers and production companies usually decide which movies to fund by manually considering if a movie will be successful based on factors like the budget, actors and actresses involved, the director, current trends in the industry, and the script. Box office revenue is universally seen as a definitive metric of success for a movie, and if filmmakers had a way to automatically pinpoint the factors that maximize a movie's revenue or predict whether a new movie would be successful, it could revolutionize the industry. Our project attempts to solve this problem. We used an IMDb dataset containing thousands of movies and employed several machine learning models to predict a movie's revenue based on its features. We focused our analysis on numerical features of the data such as budget, runtime, popularity, vote count, vote average, and year released, rather than categorical features like talent and directors involved. By understanding the relationship between these features and the revenue, filmmakers could gain valuable insights to inform their decisions during the production process.

CPSC 66 Machine Learning Proceedings, Swarthmore College, Spring 2021.

2. Methods

This paper aimed to predict film revenue based on several film features, such as budget, runtime, etc. We developed a pipeline for data analysis illustrated in Figure 1:

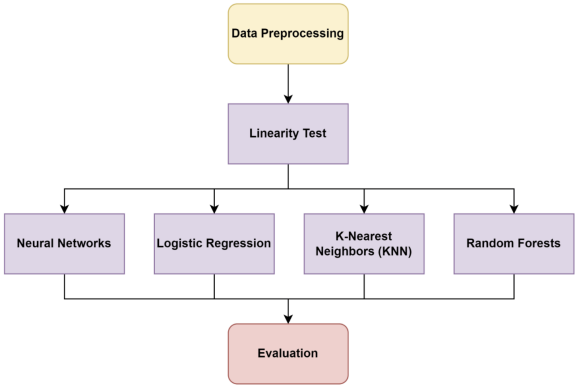

Figure 1. Workflow for film revenue classification. Starting with data pre-processing, followed by assessing linear relationships between features and revenue. Four non-linear machine learning models—Neural Networks, Logistic Regression, K-Nearest Neighbors (KNN), and Random Forests—are applied. The models are then evaluated to determine their effectiveness in predicting film revenue.

2.1. Data Collection and Pre-Processing

Our movie dataset had around 1 million entries that included a movie's title, runtime, budget, producers, revenue, and about 35 more columns of information. However, most of this information was unusable—over 99% of the data had either no budget or revenue listed, were adult films that were not appropriate for this project, were not actual films (e.g. a YouTube video), or had too little information. After filtering, we narrowed our entries to only 7,241 movies. We then consolidated the 4 "Star" columns into one column that listed all the A-list actors in a particular film. Lastly, we adjusted the budget and revenue dollar amounts to their 2024 equivalent due to having release years that span back to 1916. If we used our data without adjusting for inflation, all of our algorithms would be much more inaccurate because the dollar amounts were not properly scaled. For all movies used in our algorithms, none had a budget less than $100,000 USD and a revenue lower than $1,000 USD. All numerical features were normalized and standardized using the `StandardScaler` to ensure uniform feature scaling. Lastly, revenue values were categorized into **two** or **three** classes for classification tasks. For the binary classification task, revenue was divided into:

- **Low**: $\leq 50,000,000$
- **High**: $> 50,000,000$

For multi-class classification, revenue was categorized into:

- **Low**: $\leq 25,000,000$
- **Medium**: $25,000,000 <$ revenue $\leq 120,000,000$
- **High**: $> 120,000,000$

We called `value_counts()` on the **Revenue Class** column to 1) get a list of all unique class labels and 2) print how many examples of said class labels there were to create a balanced train/test split. With this information, we found that the revenues, although ranging from $1,400 to $9 billion USD, were incredibly skewed and did not have a consistent number of examples for each bucket that we tried to create. Because of this, the thresholds for revenue classification were manually selected to ensure a balanced distribution of films across the different classes, thereby enhancing the model's performance on both binary and multi-class classification tasks.

2.2. Linearity Test

To evaluate the initial relationship between individual features and revenue, we performed scatter plot analysis and simple linear regression for interpretable visual and statistical insights. Scatter plots had revenue plotted on the y-axis and each feature on the x-axis to visualize data distribution. Simple linear regression was applied, and the coefficient of determination (R^2) was calculated to measure the percentage of variance in revenue explained by each feature.

The predicted revenue ($\hat{y}$) was computed using the formula:

$$\hat{y} = m \cdot x + c$$

where:

- m (slope) represents the rate of change in revenue for a unit change in the feature:

$$m = \frac{\sum (x - \text{mean}(x)) \cdot (y - \text{mean}(y))}{\sum (x - \text{mean}(x))^2}$$

- c (intercept) is the predicted revenue when the feature value (x) is zero:

$$c = \text{mean}(y) - m \cdot \text{mean}(x)$$

The coefficient of determination (R^2) was calculated as:

$$R^2 = 1 - \frac{\sum (y - \hat{y})^2}{\sum (y - \text{mean}(y))^2}$$

An example of pseudocode for generating the plots and R^2 is as follows:

Algorithm 1 Linearity Validation for Feature and Revenue

Input: Dataset with numerical features and revenue
Output: Combined scatter plots with R^2 values
Initialize a combined visualization graph
for *each feature* x_i *in features* **do**
  Extract feature x_i and revenue yi from the dataset
  Create a scatter plot of all (x_i, yi) pairs
  Annotate the plot with calculated R^2
Save all annotated plots as a combined visualization graph

Our analysis of the scatter plots and R^2 values informed the selection of features for more advanced modeling.

2.3. Model Applications

Building on the results of the linear regression analysis, we implemented the following machine learning models to capture non-linear relationships and interactions among features:

2.3.1. Neural Networks

A neural network is a non-linear machine learning model inspired by the human brain that consists of interconnected layers of nodes to learn patterns through training (Hardesty, 2017). Their suitability for revenue prediction lies in two key aspects:

1. The ability to extract abstract representations in hidden layers, enabling them to handle the complexity of multi-class classification.

2. The capability to handle the large dataset, which supports robust training and enhances the model's capacity to generalize effectively, allowing accurate classification of films into three revenue ranges.

The neural network's performance is evaluated using accuracy and loss plots, while hyperparameters are tuned iteratively to optimize these metrics.

2.3.2. Logistic Regression

Logistic regression is a type of linear model that utilizes stochastic gradient descent to predict the probability of a particular binary outcome. The algorithm works by looping through all of the training examples and calculating the probability of a positive label for each, then calculating the error in the prediction and using that to update the weights of each feature. It takes a hyperparameter (alpha) to control the learning rate of these weights. Logistic Regression can also be adapted for multiclass predictions using the One vs Rest classifier which splits the data into multiple binary class predictions.

2.3.3. K-Nearest Neighbors

K-Nearest Neighbors, also known as KNN, is a supervised learning algorithm that makes a prediction about a particular data point based on the classifications of other data points within its proximity (i.e. its "neighbors"). The *k* in KNN is simply a hyperparameter in which the programmer chooses how many neighbors to consider when trying to predict certain information about their dataset.

KNN is useful for assessing non-linear relationships and, for the scope of our project, helps classify revenue ranges of a particular film based on its surrounding revenue values and their assigned labels. The effectiveness of the KNN algorithm is evaluated by its accuracy rate represented in percentages that are derived from how many predictions the algorithm got correct over the total number of predictions made.

2.3.4. Random Forests

Random Forest classifiers are built from an ensemble of Decision Trees used as multiple learners to increase the accuracy results and robustness of the classifier. As a non-linear learning model, the Random Forest classifier reduces variance by aggregating the results of multiple decision trees. This proved to be very useful with our dataset that had high variance due to the lack of great linear relationship within individual features to **Revenue**.

3. Experiments and Results

3.1. Linearity Test

3.1.1. Results

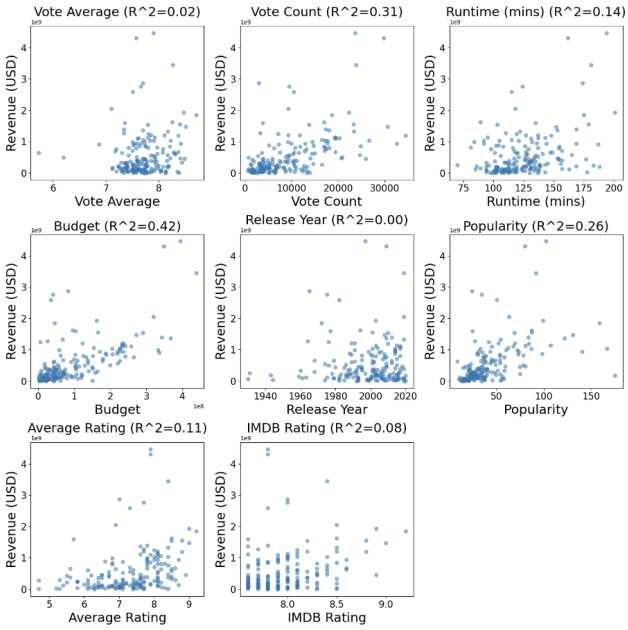

Figure 2. Scatter plots illustrating the relationship between individual features on the x-axis and revenue on the y-axis. The R^2 values indicate the strength of linearity. R^2 values closer to zero indicate weak linear relationships, while R^2 values closer to one indicate stronger linear relationships. The results demonstrate weak to moderate linear correlations between individual features and revenue.

The linearity analysis revealed that no individual feature demonstrates a strong linear relationship with revenue (Fig 1.). Among all features, **Vote Count** exhibits the strongest linear relationship with revenue from the scatter plot. However, the $R^2 = 0.31$ is relatively low, indicating that only a small portion of revenue variance is explained by Vote Count. **Budget** shows moderate correlation with revenue ($R^2 = 0.42$); however, the scatter suggests significant variability for low-budget films. **Popularity** ($R^2 = 0.26$) exhibits non-linear patterns for high popularity scores. **Vote Average** ($R^2 = 0.02$), **Runtime** ($R^2 = 0.14$), **Release Year** ($R^2 = 0.00$), **Average Rating** ($R^2 = 0.11$), and **IMDb Rating** ($R^2 = 0.08$) showed weak or negligible correlations, with scatter plots indicating non-linear trends (Figure 2).

3.1.2. Discussion

The results suggest that no individual feature exhibits a strong linear relationship with revenue. This finding aligns with previous research highlighting the limitations of linear models for revenue classification, given the non-linear interactions among features (Sen Sharma, et al., 2021). Consequently, we concluded that non-linear models are better suited to capture the complex interactions among features and improve revenue classification.

3.2. Neural Networks

To predict film revenue, a neural network was implemented and trained on the dataset containing about 7,200 samples.

3.2.1. Experimental Methodology

Model Architecture and Training A feed-forward neural network with three hidden layers was designed, employing ReLU activation functions, dropout for regularization, and batch normalization to stabilize training. The model was compiled with a stochastic gradient descent (SGD) optimizer and a categorical cross-entropy loss function to handle the multi-class target. A 70/30 train-test split was used for evaluation, and the model was trained for 300 epochs with a batch size of 64.

Evaluation Training and validation accuracy were tracked over epochs to monitor convergence and assess the model's generalization. The final model's accuracy was evaluated on the validation set, and the classification performance was summarized.

3.2.2. Result

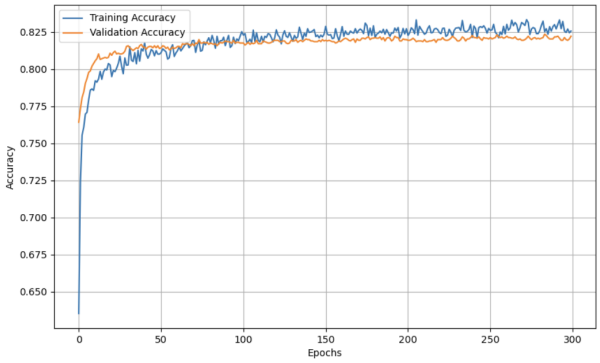

Figure 3. Training and validation accuracy of the neural network over 300 epochs with 2 revenue classes. The x-axis represents training epochs, and the y-axis represents accuracy. The curves demonstrate consistent convergence, with validation accuracy stabilizing at 82%, indicating effective generalization.

Table 1. Classification Report for Neural Network on Film Revenue Classification (Two Classes).

Class	Precision	Recall	F1-Score
Low Revenue (0)	0.81	0.83	0.82
High Revenue (1)	0.83	0.82	0.82
Accuracy		0.82	
Macro Avg	0.82	0.82	0.82
Weighted Avg	0.82	0.82	0.82

We observed the following during the training and evaluation of the neural network:

1. Both the training and validation accuracy curves showed rapid improvement during the initial epochs, stabilizing around 250 epochs (Figure 3).
2. The training accuracy stabilized at approximately 82%, and the validation accuracy remained close throughout, indicating minimal overfitting (Figure 3).
3. The final validation accuracy of 82% suggests that the neural network effectively captures the non-linear relationships among features (Figure 3).
4. The classification report further supports the model's performance. The precision, recall, and F1-scores for all classes are consistent (Precision: 0.81, Recall: 0.83 for Class 0; Precision: 0.83, Recall: 0.82 for Class 1), and the macro-average and weighted-average metrics confirm that the model performs well across all classes, without bias toward any specific revenue range (Table 1).

Table 2. Classification Report for Neural Network on Film Revenue Classification (Three Classes).

Class	Precision	Recall	F1-Score
Low Revenue (0)	0.80	0.75	0.77
Medium Revenue (1)	0.62	0.65	0.63
High Revenue (2)	0.80	0.81	0.80
Accuracy		0.73	
Macro Avg	0.74	0.73	0.74
Weighted Avg	0.74	0.73	0.73

We also ran the neural network for three revenue classes and received lower accuracy and higher variability. Neural network achieves higher overall accuracy (82%) in the binary classification compared to 73% in three-class classification. The three-class classification also exhibits moderate precision and recall variability compared to binary class

classification. In Table 2., the middle revenue range (Class 1) achieves Precision 0.62 and Recall 0.65, which are notably lower compared to the high (Class 2: Precision 0.80, Recall 0.81) and the low revenue range (Class 0: Precision 0.80, Recall 0.75) (Table 4). These results highlight the model failed to classify the middle revenue class compared to the other classes.

3.2.3. Discussion

The results demonstrate that the neural network effectively models the complex interactions among features, achieving **82% accuracy** and balanced performance for two revenue classes, outperforming linear models. This highlights the model's robustness in handling varying data distributions. The close alignment of training and validation curves further underscores the model's ability to generalize effectively.

For the three-class revenue classification, the neural network achieved a lower overall accuracy of **73%**, with greater variability. This variability is expected due to the middle revenue range often sharing characteristics with low- and high-revenue films. The additional complexity also makes it challenging for the neural network to accurately model the boundaries between classes.

Overall, the neural network demonstrates strong performance, with better accuracy and more consistent results in the two-class scenario due to the more distinct separation of classes. However, the specific features contributing most to the classification remain unclear, as the neural network operates as a black-box model.

3.3. Logistic Regression

3.3.1. Experimental Methodology

Logistic regression is a type of linear model that utilizes stochastic gradient descent to predict the probability of a particular outcome. To run this model, we followed the same procedure to clean up the data and load them, separating the revenue into discrete classes. We normalized the data and split them into train and test sets, then fit the model using alpha = 0.01 as the hyperparameter. We also adapted the model for three classes using the One vs Rest classifier.

3.3.2. Results

After running the logistic regression model on the data, the weights of each feature are as shown in Table 3.

Feature (two classes)	**Weight**
Vote Average	1.001
Vote Count	1.254
Runtime (mins)	1.054
Budget (USD, Adjusted for 2024 Inflation)	1.278
Release Year	0.978
Popularity	1.124
Average Rating	1.026
IMDB Rating	0.892
Bias	1.045
Accuracy	**0.81**

Feature (three classes)	**Weight**
Vote Average	-0.996
Vote Count	-1.088
Runtime (mins)	-1.067
Budget (USD, Adjusted for 2024 Inflation)	-1.076
Release Year	-1.046
Popularity	-1.118
Average Rating	-1.069
IMDB Rating	-0.395
Bias	0.762
Accuracy	**0.63**

Table 3. Logistic Regression Results for Movie Data.

3.3.3. Discussion

Running the model using two classes for the revenue produced the expected results. Budget had the highest weight, followed by the vote count. These results are logical as higher budget movies tend to produce a higher revenue, and higher number of people rating a movie (vote count) means that more people likely went to see that movie. However, when running the model with the revenue split into three classes, popularity (weekly IMDb site hits) had the highest weight, followed by vote count and budget. This is inconsistent with our expected conclusions but the differences between the weights were not significant. Additionally, the accuracy dropped from 81 percent with two classes, to 63 percent with three classes. This was expected because the model gets the answer "correct" less often when there are more class options to choose from.

3.4. KNN

3.4.1. Experimental Methodology

Three runs of the KNN algorithm were conducted using the SciKit Learning and Pandas libraries. The preprocessing and the label classifications of revenues as discussed in section **2.1** allowed us to create a new column in the movie list dataset called **Revenue Class**. We ran two KNN models—one had two classes where the range of predictions

were either 0 or 1, where 0 was a **Low** value and 1 was a **High** value. The other had three classes where the range of predictions were 0, 1, or 2, where 0 was a **Low** value, 1 was a **Medium** value, and 2 was a **High** value. We manually tuned k and assessed the accuracy of the models when $k = 1$, $k = 5$, and $k = 10$.

3.4.2. RESULTS

As shown in **Table 4**, the KNN models produced the highest accuracies with only a binary classification and were less accurate for multi-class labels.

Table 4. Classification Report for KNN on Film Revenue Predictions with Two and Three Classes.

Accuracy %	$k = 1$	$k = 5$	$k = 10$
Two Classes	81.5%	83.6%	83.6%
Three Classes	67.6%	71.4%	71.2%

3.4.3. DISCUSSION

Based on the accuracies, the ideal k–value would be between 1 and 5, inclusive. The revenue values are heavily skewed and the number of ideal neighbors being ≤ 5 means that the KNN algorithms we used are prone to overfitting due to a low k–value. This model for both 2 and 3 classes may have a higher level of variance—if the revenue values were to become less skewed, it would dramatically change the accuracy of the algorithm. There may also be a lower level of bias due to the sheer amount of entries and the fact that small details are at risk of being overlooked as the model runs so it can make a prediction. Lastly, a small k–value means that the classes may have very complex boundaries that are highly sensitive to outliers, which ties back into having a higher level of variance. In order to get a larger k–value, we think that choosing a set of movies with revenues that follow a more normal or even distribution could increase the accuracies up to 85-90% or higher.

3.5. Random Forests

3.5.1. EXPERIMENTAL METHODOLOGY

To implement the Random Forest Classifier, we used the SciKit Learn sklearn.ensemble module which utilizes bootstrap aggregating. This classifier utilized the same preprocessing that was explained in Section 2.1. There was no need to perform normalization as the model is built on multiple decision trees.

3.5.2. RESULTS

The Random Forest Classifier provided similar results to the weights shown in Logistic Regression as shown in the Tables below:

Feature	**Importance**
Vote Average	0.100
Vote Count	0.169
Runtime (mins)	0.085
Budget (USD, Adjusted for 2024 Inflation)	0.247
Release Year	0.111
Popularity	0.152
Average Rating	0.075
IMDB Rating	0.066
Accuracy	**85.21%**

Table 5. Random Forest Classifier Feature Importance Results on Film Revenue (2 Classes)

Feature	**Importance**
Vote Average	0.097
Vote Count	0.165
Runtime (mins)	0.086
Budget (USD, Adjusted for 2024 Inflation)	0.261
Release Year	0.100
Popularity	0.139
Average Rating	0.082
IMDB Rating	0.070
Accuracy	**71.86%**

Table 6. Random Forest Classifier Feature Importance Results on Film Revenue (3 Classes)

These feature importances were calculated using the "feature_importances_" attribute for the SciKit Learn Random Forest Classifier.

3.5.3. DISCUSSION

TThe results show that **Budget** held the highest feature importance on the Random Classifier Model for both class Distinctions as it has **24.7%** importance when run on 2 classes (Figure 5), and **26.1%** when run on 3 classes (Figure 6). These feature importances reveal the relative importance of each feature within each tree of the ensemble, more specifically, how much each feature contributes to the model's predictions. With this, and the higher accuracy of **85.21%** of the Random Forest Classifier when ran on two classes (Figure 5), this model shows us that Budget has the greatest impact on the Revenue class. In other words, the Budget is a logical indicator for how much revenue a movie may make. These results are similar to those in the Logistic Regression Model with the only difference being that **Budget** maintained the highest level of importance between both class distinctions.

4. Social Implications

Key stakeholders in our project include filmmakers and producers. Access to a model that reliably predicts a movie's revenue based on its features would revolutionize filmmaking, since producers would be able to predict which movies were likely to be successful and use that information to choose which movies to fund. This could also pose potential drawbacks: If these models were relied on too heavily, the budgets, runtimes, etc. that produced the highest predicted revenue could dominate the industry, leading to increased homogeneity. For example, if higher budget films or movies with shorter runtimes consistently make more money, it might discourage filmmakers from pursuing passion projects and only big blockbuster movies would get funded.

Another drawback of this project is that there are potentially many other important factors that contribute to a movie's box office turnout that are not considered in our analysis. For example, certain actors or directors being involved in a movie could drastically impact turnout, but these are not as well suited to Machine Learning analysis as the numerical features due to a lack of data, so we did not include them.

We believe it important to also mention that Hollywood and the wider film industry is riddled with misogyny, racism, ableism, and other types of institutional discrimination. It may very well be the case that movies that have a higher number of white and/or male cast members, for example, could make more in revenue due to the prevalence of misogyny and racism in the film and marketing industry. Overall, we believe that if producers were to rely on our models to decide which movies to fund, it could lead to a lack of variety in the film industry, create a negative feedback loop of less diverse actors/actresses on screen, and may influence the creation of films that were made with the priority of maximizing revenue instead of creativity and fresh, new ideas.

5. Conclusion

To summarize, our group found that the budget of a film is the most significant factor in predicting its revenue. This makes sense because the higher budget a movie has, the more it can spend on resources to produce. The models that achieved the most accuracy in predicting revenue data from our numerical features were our Random Forest and KNN algorithms. Due to our time and resource constraints, we believe that a possible avenue future researchers could explore would be experimenting with more robust hyperparameter tuning and how to include categorical features into these machine learning models to observe how the accuracy is affected.

References

Hardesty, L. Explained: Neural Networks. *MIT News Office*. Archived on 18 March 2024. Retrieved on 2 June 2022, 14 April 2017.

Scikit-Learn: "1.11. Ensembles: Gradient Boosting, Random Forests, Bagging, Voting, Stacking." Accessed 11 Dec. 2024. https://scikit-learn/stable/modules/ensemble.html

Sen Sharma, A., Roy, T., Rifat, S. A., & Mridul, M. A. Presenting a Larger Up-to-date Movie Dataset and Investigating the Effects of Pre-released Attributes on Gross Revenue. *arXiv preprint*, arXiv:2110.07039 [cs.IR], 13 October 2021. https://arxiv.org/abs/2110.07039

Shubham Chandra. (2024). IMDB & TMDB Movie Metadata Big Dataset (over 1M) [Data set]. *Kaggle*. https://doi.org/10.34740/KAGGLE/DSV/9111436

U.S. Inflation calculator: 1635-2024, department of labor data. (n.d.). Retrieved December 11, 2024. From https://www.in2013dollars.com