UNLOCKING MOVIE MAGIC:

PREDICTIVE FACTORS TO FILM SUCCESS

SHRIYA CHINTHAK, PATRICIA SCHENFELD, LINDSAY STRONG, AGUSTINA ZUCKERBERG

Georgetown University, MS Data Science & Analytics, Statistical Learning for Analytics

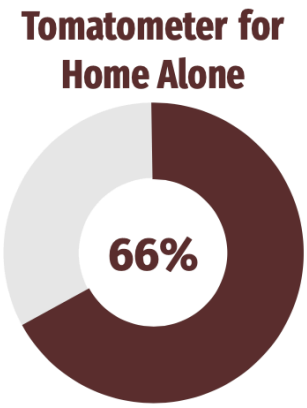
April 2024

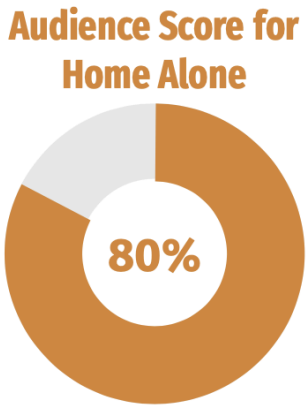
TABLE OF CONTENTS

INTRODUCTION

Lights, camera, action – the timeless allure of the silver screen continues to captivate audiences worldwide. Behind every cinematic masterpiece lies a delicate balance of artistry, storytelling, and commercial viability. Yet, amidst the glitz and glamor of Hollywood, the question lingers: What makes a movie truly successful?

In 2023, total earnings at the box office across the United States and Canada amounted to around 8.91 billion U.S. dollars ("Box office revenue in the U.S. and Canada 2023"). With cinema having such a large economic impact on the entertainment industry, the presence of several movie ranking systems have arisen. Movies are inherently subjective and vary in popularity. Companies like Netflix, IMDb and Rotten Tomatoes rate movies and measure popularity using their own metrics. For example, IMDb has two different metrics, IMDB Score and Popularity Score, while Rotten Tomatoes rates movies using a Tomatometer and an Audience Score. For example, the Rotten Tomatoes ratings for the movie Home Alone (1990) are shown below. Why are these scores different and what metrics are consumers and critics using to rate these movies? Our goal in this project is to understand the nuances of movie popularity and success, and see what features are the leading contributors to higher ratings amongst IMDb and Rotten Tomatoes scores.





Further analysis will allow us to answer our following objectives:

- **What kind of statistical model best predicts the audience score of a movie based on the movie's features?**
- **What features of a movie are most correlated with each other? How might that impact success?**
- **What genres directly affect the success of a movie? Do the types of genres differ for the three metrics of success?**
- **What features are best at predicting Audience Score? Critic Score? Revenue?**
- **Are the features of best prediction between the three metrics of success stay the same or differ from one another?**

- **Are Audience Score, Critics Score, and revenue correlated or influence each other?**
- **Are Audience Score, Critics Score, and revenue reliable metrics for measuring the success of a movie?**
- **Do simpler or more complex models better predict audience score, critics score, and revenue?**
- **Does the time of year that a movie is released impact its success?**
- **Are there trends in audience score and critics score over time?**

Utilizing machine learning and statistical analyses, we will uncover answers to these questions, and provide a thorough investigation into the factors influencing movie success, offering practical insights for industry professionals and movie buffs alike.

DATA GATHERING AND CLEANING

For our analysis, we chose data from The Movie Database (TMDb), a user-editable movie and television database, and Rotten Tomatoes, a movie and television rating website. We pulled movie data from TMDb on April 11, 2024 using TMDb's API. The dataset contained information for 1,022,386 movies with 24 attributes. This data included one of our success metrics, revenue, in addition to information about the movie such as the title, release year, and genre. We removed variables that would not add to the model's performance and only kept the 550,604 english movies in our dataset. We used the tidyverse and dplyr packages in R for data cleaning. We prepared the data for analysis using one-hot-encoding for the movie genres, creating attributes for the number of production companies and spoken languages per movie. Additionally, we changed necessary variables to datetime variables and created an id for each movie based on the title and release year.

We also used movie data from Rotten Tomatoes. The dataset contained 143,258 movies with 16 attributes. This dataset included our final two success metrics; Rotten Tomatoes' tomatometer rating and the audience score rating. We changed appropriate data types, creating a release date column based on the release date theaters value and release date streaming value, and created the id for each movie based on the title and release year. We merged this dataset with the cleaned TMDb dataset by movie id. This resulted in 37,298 movies. We removed the duplicate rows and cleaned the movie titles. We converted the dataframe to a csv file to use in our analysis.

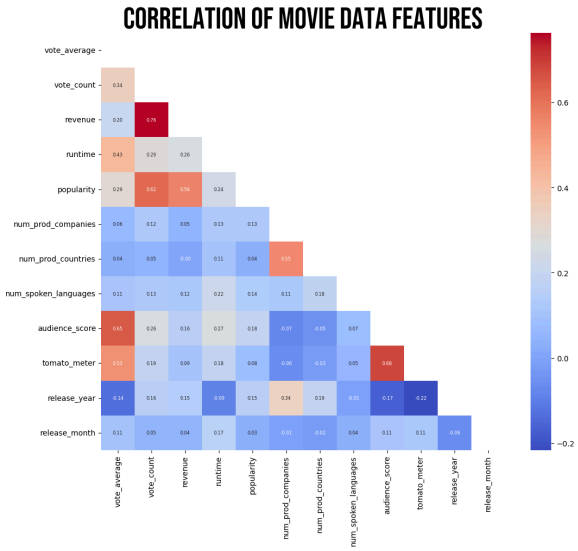
EXPLORATORY DATA ANALYSIS

Exploratory data analysis (EDA) is a crucial initial step in uncovering insights from datasets, providing a foundational understanding of underlying patterns and relationships. In our analysis, we began by looking at some descriptive statistics of the dataset. Particularly interesting figures we found were the differences in average scores for the Tomatometer and Audience rating, and the distribution of revenue and runtime.

	count	mean	std	min	25%	50%	75%	max
revenue	7199.0	7.692569e+07	1.697527e+08	1.0	3178363.5	18200000.0	71958107.5	2.923706e+09
runtime	7199.0	1.054602e+02	2.190553e+01	1.0	94.0	103.0	116.0	3.390000e+02
tomato_meter	7199.0	5.714183e+01	2.839933e+01	0.0	33.0	61.0	83.0	1.000000e+02
audience_score	7199.0	6.209529e+01	2.004661e+01	0.0	47.0	64.0	79.0	1.000000e+02

The average Tomatometer rating in the dataset was around 57, while the average Audience score was 62. Although the averages are not vast in range, the audience ratings skew higher - are critics harsher raters than typical consumers? Further, looking at the distribution of revenue amounts, the mean revenue generated by movies stands at approximately $76.93 million, with considerable variability as indicated by the standard deviation of approximately $169.75 million. To put this into perspective, the maximum generated revenue in this dataset is nearly $3 billion dollars. WIth such a wide range of values for revenue, a prime indicator of success, we wonder what features correlate with revenue. Lastly, looking at runtime, the average movie length is 105 minutes, with the longest being 339 minutes (that's over 5 hours!). Our instincts tell us that runtime could be a potential factor in a higher or lower score.

One of the most valuable parts of exploratory data analysis is creating visualizations due to its effectiveness in being able to identify trends and patterns more easily. We began with creating a correlation heatmap matrix of the numeric (non-dummy) variables in our dataset. This correlation matrix shows how positively or negatively each feature is to one another.



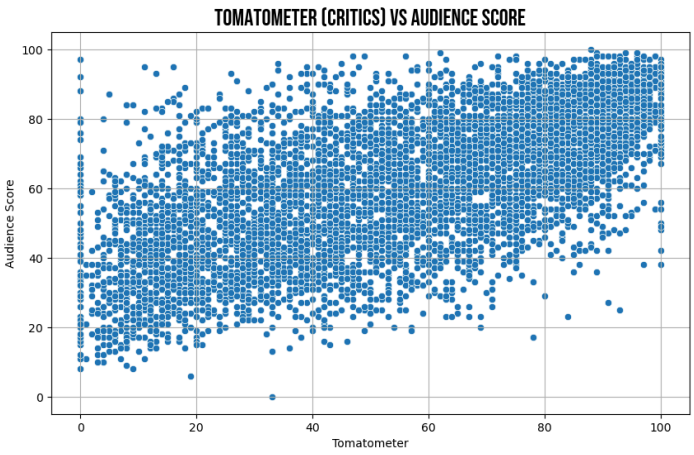
Notable feature correlations include:

- tomatometer and vote_average (IMDb score): .53
- audienceScore and vote_average: .65

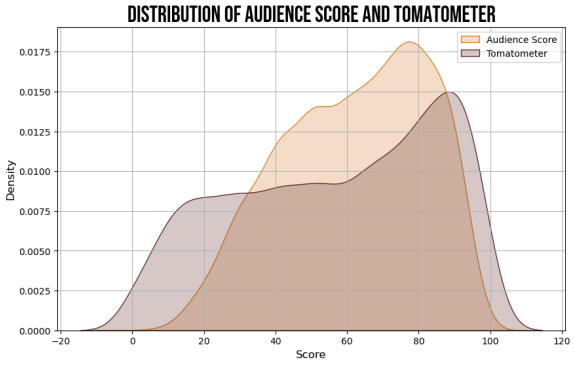
- audienceScore and tomatometer: .68
- popularity and revenue: .56
- vote_average and runtime are positively correlated at 0.43.

This implies that as average votes (IMDb scores) increase, the tomatometer rating increases, when audience score increases, average user votes increases, as the audience score increases, the tomatometer rating increases, as popularity increases, revenue increases, and as runtime increases, average vote increases. These will be interesting to look further into to discern whether correlation equals causation.

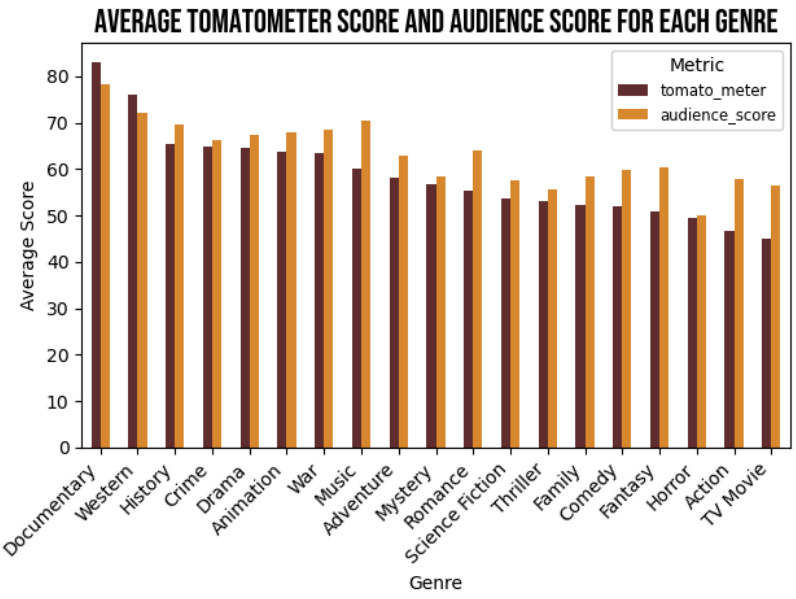
Further, we wanted to see how Tomatometer ratings related with the audience score. In the graph to the left, we plotted Audience Score against Tomatometer to visualize the relationship between these variables. There is an evident positive correlation as the scatterplot trends upward to the right. A positive correlation suggests that movies with higher audience scores tend to also receive higher ratings from critics on the tomatometer. This consistency in perception between audience and critics indicates a shared appreciation for certain aspects of the films. It also suggests a certain level of reliability in both measures. This consistency could enhance the credibility of movie reviews and ratings, providing valuable guidance to audiences in their decision-making process.



In relation, to get a better understanding of the distributions of tomatometer and audience ratings, we created the graph on the right to compare each. Tomatometer ratings exhibit a tendency towards lower scores compared to audience ratings, suggesting a potential disparity in the evaluation criteria employed by critics and general viewers. Moreover, the wider range of values observed in Tomatometer ratings implies a greater variability in critical assessments, possibly reflecting the diverse perspectives and standards among professional reviewers.



Next, for the graph below, we aimed to explore the impact of genres on movie ratings. Understanding that individual preferences for movie genres vary widely, we sought to delve into how different genres influence audience and critic perceptions of film quality. From the graph, it is clear that overall, audience scores are higher for nearly every genre compared to the critics tomatometer scores. This leads us to the question, are critics harsher raters than consumers? Further analyzing, we can see that the genres Documentary, Western, and History are the top 3 highest average scores amongst critics. Within that, it's interesting to see that the only two genres where the tomatometer score was higher than the audience score were for the Documentary and Western genres. Documentary, Western, and Historical genres often engage with real-world issues, historical events, or cultural contexts. Critics may appreciate the nuanced exploration of these subjects, whilst audiences tend to have higher ratings for typical fictional genres such as Romance, Music, and Fantasy.



Overall, our exploratory data analysis of our movie dataset has allowed us to gain insights into many different features of the data, guiding our further analysis. We have been able to learn more about the relationships in our data and how they may contribute to building a predictive model for movie success.

METHODS AND ANALYSIS

As we saw through our exploratory data analysis, there are several variables which seem to contribute to audience score, critics score, overall revenue, or all three. Thus, in order to further understand these relationships as well as answer the question of what makes a movie successful, we'll create statistical predictive models to determine what features contribute to the success metrics, audience score, critics score, and revenue. This process can be broken up into two main components; feature selection and statistical modeling.

FEATURE SELECTION

After conducting the exploratory data analysis, we were able to identify the different aspects of the variables in our dataset and gain valuable insights. Considering the number of variables present in the dataset, we previously analyzed the ones that we thought would be most relevant for the analysis. We may have additional variables that may or may not have an impact on the predictor variables. Including these variables in our analysis may increase the complexity of the model and therefore impact the run time. Using the Generalized Linear Models (GLMs) as the baseline model is convenient for several reasons. First, the model calculates predictions by analyzing the linear relationship between features within the dataset. Based on its versatility in different tasks, comparing it to other models can help improve the overall performance. Second, running the GLM model in Python is straightforward using the statsmodels.api library, the GLM function and specifying the target variable and distribution type. After running the model, the summary provides a comprehensive list of features within the dataset along with their corresponding coefficients, standard errors, z-values, p-values and confidence intervals. To determine the significance of each feature, we conducted a hypothesis test as following:

$$H_0: \textit{The coefficient of the predictor variable is not statistically significant for the model.}$$

$$H_A: \textit{The coefficient of the predictor variable is statistically significant for the model.}$$

We evaluate the significance of each feature based on a 5% significance level. The model summary is a valuable resource for interpreting the results of the model compared to other models. It provides a concise overview of the model's reasoning, making it both interpretable and easy to understand. Third, selecting the statistically significant features we concentrate the analysis on the variables that most effectively explain the response variable.

Since we have three response variables in our original data, we created three different datasets, each containing the same features and only one of the response variables. The audience_dataset contains the response variable Audience Score, the critics_dataset contains the response variable Tomatometer and the revenue_dataset contains the response variable Revenue. We split the data into training, validation, and test sets. We normalized the data using the MinMaxScaler() function from the sklearn.preprocessing package. As we saw in the exploratory data analysis, the response variables have approximately Gaussian distributions. Therefore, we define the models for each data set as follows:

$$glm\_model1 = sm.GLM(audience\_train\_target,\ audience\_train\_data,\ family = sm.families.Gaussian())$$

$$glm\_model2 = sm.GLM(critics\_train\_target, critics\_train\_data, family = sm.families.Gaussian())$$
$$glm\_model3 = sm.GLM(revenue\_train\_target, revenue\_train\_data, family = sm.families.Gaussian())$$

After fitting the GLM models, we compute predictions based on the training and validation data and evaluate their performance using the Root Mean Squared Error (RMSE) loss function. Additionally, we developed a helper function to identify statistically significant features for each model, as shown in Appendix - Fig #1. Basically, we save the summary of each model and convert it into a data frame. We perform feature selection by applying a hypothesis test to each variable based on the obtained p-value and a significance level of 0.05. We then save the statistically significant variables in separate lists for each model. Next, we subset the original data to keep only the selected features. The selected features for each model are shown in the figure, where the 'x' represents the selected features and the others were dropped. In addition, we note that the total number of features per model varies. Specifically, the audience model contains a total of 23 features, while the critics and revenue models contain 22 and 16 features, respectively.

	Audience	Critics	Revenue
vote_average	x	x	x
vote_count	x	x	x
runtime	x	x	x
popularity	x	x	x
action	x	x	x
science_fiction	x		
adventure	x		
drama	x	x	x
crime		x	
thriller	x	x	
fantasy	x	x	
comedy	x	x	x
romance	x	x	
western		x	x
mystery	x		
animation	x	x	x
family	x	x	x
horror	x	x	
music	x		
history	x		
documentary	x	x	x
num_prod_companies	x		x
num_spoken_languages		x	
pg_13	x	x	
r	x		x
pg_13		x	x
g		x	x
release_year	x	x	x
release_month		x	

Based on the hypotheses presented in the Exploratory Data Analysis section, we observe that the genres Documentary and Western are significant for the critics analysis but History is not. We can also see that the genres Science Fiction, Adventure, Mystery and Music are not significant for critics but for the audience, while the opposite happens for Crime and Western. We consider two main reasons for this discrepancy. First, critics may prioritize the complexity of a movie's narrative over its attractiveness and popularity, which are more important to audiences. Second, the distribution of genres for all the movies in our dataset is unbalanced, with certain genres having more movies than others, which affects their significance and the model's ability to best describe the target variables. Other features that are important for the Tomatometer score but not for the Audience score include the number of languages spoken in the movie, the movie's rating (specially for

PG13 and G), and the month of the movie's release. On the other hand, to better predict the revenue we consider features related to voting, runtime, popularity, number of production companies, movie ratings, release year, and genres such as Action, Drama, Comedy and Western, among others.

Other methods that can be used to perform feature selection include forward and backward stepwise regression, but for this analysis we focused on generalized linear models. Based on these results we will continue to use the selected features for subsequent analysis using XGBoost and neural network models.

PREDICTIVE MODELS

Through feature selection, utilizing generalized linear regression models, we successfully established a solid foundation for predicting audience score, critics score, and revenue of a film. As we dive deeper into exploring additional predictive models in this project, we will prioritize the features identified as most significant by the GLM, alongside its RMSE serving as a standardized benchmark.

The first type of model we will use is a feed-forward fully connected neural network. This type of predictive model is an ANN where information flows in one direction, from the input layer to the output layer, without any loops or cycles. Each neuron in a layer is connected to every neuron in the subsequent layer. These connections hold weights that are adjusted during the training process to learn complex patterns in the data (Rumelhart et al). After the training process is complete, the hyperparameters of the model are tuned using a validation dataset in order to increase generalization of the neural network as well as prevent overfitting to the training data. To accomplish this type of model, we created a multi-layer perceptron in Python using Keras.

Diving deeper into the architecture of our neural network models, we utilized the Keras, an open-source library in Python. We created a function that takes in the number of hidden layers, the size of the hidden layers, activation function, dropout rate, optimizer, learning rate, and regularization rate from the user and returns a feed-forward model. Using the sequential function in Keras, we created a model that consisted of a dense input layer, a number of hidden layers with node length determined by the user, corresponding dropout layers, and a dense linear output layer. In addition to the structure of the model, since this problem is ultimately a linear regression problem, we used mean squared error (MSE) as a pre-built loss function and later converted that to root mean squared error (RMSE).

With the function that creates the architecture of the neural network, we implemented a number of techniques to tune the model for the best outcome without overfitting. We first split the data to train, validation, and test sets in order to hypertune the model without fitting to the test set. To conduct hypertuning on these models, we used a grid search, a technique used to find the optimal hyperparameters that reduces the overall loss of the model. We implemented a grid search that runs through every iteration of the values found in Fig #2**.** After acquiring the optimal parameters for each neural network, we trained these models using K-fold cross validation, a technique for training in batches to avoid overfitting to the training data. Additionally, we also incorporated early stopping into the training process, a technique which terminates the training process when the validation loss begins to increase for each epoch. Through this process, we are able to find the model for each success metric with the optimal hyperparameters and lowest loss.

Aside from our neural network models, we also ran XGBoost Regression models to predict audience score, critics score, and revenue. XGBoost Regression is a powerful modeling technique for building regression models using decision trees. Where it excels is in speed, handling large datasets, and offering interpretability through feature importance scores (Yan et al). While our neural networks are versatile and able to make decent predictions, in theory, XGBoost can be better at dealing with limited data and computational resources. Thus, using the XGRegressor package in Python, we created a similar function to the neural network model that takes in hyperparameter values and outputs a trained mode (Fig #3). We performed grid search with cross-validation to find the best hyperparameters based on the negative mean squared error. We then fit the best model on the training data, evaluated its performance on validation and test sets, and printed the root mean squared errors for a final evaluation. The optimal models for all success metrics for both neural networks and XGBoost Regression architectures can be found in Fig #4.

Thus, we used a variety of predictive models to infer the audience score, critics score, and revenue of each film. Now, we will review the results of all the models in comparison to the GLM as a benchmark.

RESULTS

In our exploration to define what makes a movie successful, we established predictive models to further the discussion of whether movie features are imperative to the success of a movie. Firstly, we defined a generative linear model (GLM) as a baseline model to measure the

efficacy of more sophisticated algorithms, namely XGBoost and Neural Networks. The comparison of these models was measured using the RMSE metric, serving as a yardstick for predictive accuracy across these models. For ideal comparison, the RMSE values across all three models were normalized around the GLM's initial performance. Thus, a normalized RMSE value lower than 1 indicates a better performing model than the GLM.

When comparing the models, XGBoost consistently outperformed both the Neural Network and GLM models, demonstrating a more adept grasp of intricate data patterns. Conversely, the Neural Network model struggled to match the predictive prowess of XGBoost and even lagged behind the GLM model, particularly concerning revenue prediction. This performance disparity persisted across both training and testing datasets, indicating a systemic trend and potential pitfall in the features of the dataset when estimating revenue. A table of the normalized RMSE scores for all models, all metrics, and all dataset can be found in Fig #5.

CONCLUSIONS

Throughout our exploratory data analysis, feature selection, and predictive models, we were able to find the features that determine a movie's success. We found that Rotten Tomatoes' audience score and tomatometer have a strong positive correlation while revenue shows minimal correlation with either. This positive correlation between audience score and tomatometer shows the similarity between audience and critics ratings of movies. It also shows the reliability of both metrics. Revenue is slightly negatively correlated with both metrics which shows that revenue is not a predictor of the success of a movie. This is useful knowledge for movie producers as a lower audience and critic score during an initial screening does not suggest that the movie will make less revenue or vice versa.

During exploratory data analysis, we also found that tomatometer ratings are negatively correlated with release year. This suggests that tomatometer ratings are more critical than they were in the past or the quality of movies has decreased. We also found that longer movies tend to receive higher viewer ratings due to a positive correlation between vote average and runtime. This information could be useful for movie producers as making their movies longer could result in increased ratings.

From our feature section, we found that significant features for each model differ mainly in genre and ratings and these features were used to train our predictive models. For example, audiences' preferred adventure and science fiction while critics' scores were heavily influenced

by crime and western genres. Additionally, the number of spoken languages was more significant in the critic's tomatometer score than the audience score. The critics also preferred G and PG-13 movies while the audience preferred R-rated movies. These significant features show the differences in audience and critic preferences in a successful movie. When Rotten Tomatoes users are deciding on a movie to watch, it is important for them to be aware of these preferences. For example, users who are looking to watch an R-rated adventure movie should value the audience score more than the tomatometer score as audiences prefer R-rated movies and adventure movies compared to critics.

In our predictive models, the XGBoost Regressor performed the best at predicting all three metrics of success compared to the benchmark model. Thus, we can say that the simpler predictive model was more effective in our analysis. We used our feature selection in our predictive models and we were able to predict the audience score, critics' score, and revenue. In our best model, the audience score had the lowest loss, followed by the revenue and the critics' score. Thus, the model predicted the audience score the best and the critics' score the worst. This could be due to critics being more subjective compared to a larger audience with diverse opinions. These results could be useful to Rotten Tomatoes or other movie rating companies if they decided to use predictive models to generate their scores. They should generate audience scores rather than the tomatometer as the model performs better with audience scores.

In closing, we were able to create predictive models that predict the success of movies based on our definition. Our analysis showed the validity of the metrics Rotten Tomatoes uses, the differences in audience and critics' ratings of movies, and their relationship to each other. These insights could be leveraged by stakeholders in the entertainment industry for successful movies in the future.

REFERENCES

"Box office revenue in the U.S. and Canada 2023." *Statista*, Statista Research Department, 3 January 2024, https://www.statista.com/statistics/187069/north-american-box-office-gross-revenue-since-1980/ . Accessed 22 April 2024.

Rumelhart, D., et al. “Learning representations by back-propagating errors.” *Nature*, vol. 323, 1986, pp. 533–536, https://doi.org/10.1038/323533a0 .

Yan, Zhiqiang, et al. “XGBoost algorithm and logistic regression to predict the postoperative 5-year outcome in patients with glioma.” *National Library of Medicine*, 2022. *NIH*, 10.21037/atm-22-3384.

APPENDIX

Fig #1: Models Notebook - Feature Selection Function for GLM Model

```python
# Function to select features using GLM model
def glm_feature_selection(glm_result):

    # Create an empty list to store the significant features
    glm_summary = []

    # Get the summary of the GLM model
    summary = glm_result.summary()

    # Get the second table from the summary
    summary = summary.tables[1]

    # Convert the summary table to a DataFrame
    summary = pd.DataFrame(summary.data)

    # Set the first row as the column names
    summary = summary.rename(columns=summary.iloc[0])

    # Drop the first row
    summary = summary.drop(0)

    # Format nan values to np.nan
    summary["P>|z|"] = summary["P>|z|"].replace("   nan", np.nan)

    # Convert P>|z| column to numeric
    summary["P>|z|"] = pd.to_numeric(summary["P>|z|"])

    # Ignore constant term
    summary = summary[summary[""] != "const"]

    # Select features with p-value less than the significance level
    significant_features = summary.loc[summary["P>|z|"] < 0.05, ""]

    # Store the significant features in glm_summary list
    glm_summary.extend(significant_features)

    # Return the list of significant features
    return glm_summary
```

Fig #2 - Hyperparameter Options for Neural Networks

Name	Values
Hidden Layer Size	[2,3]
Number of hidden layer nodes	[64,128]
Dropout Rate	[0, 0.1]
Regularization	[L1,L2]
Regularization Rate	[0.01, 0.001]

Fig #3 - Hyperparameter Options for XGBoost Regression

Name	Values
ETA	[0.1, 0.01]
Max Depth	[3, 4, 5]
Subsample	[0.7, 0.8, 0.9]
Column Sample per Tree	[0.7, 0.8, 0.9]
Lambda	[0.01, 0.1, 1]
Alpha	[0.01, 0.1, 1]

Fig #4 - Hyperparameters of Final Models

Model Type	Final Parameters
Neural Network - Audience Score	[Hidden Layer Size: 2; Number of hidden layer nodes: 64; Dropout Rate: 0.1; Regularization: L2; Regularization Rate: 0.01]
Neural Network - Tomato (Critics) Score	[Hidden Layer Size: 2; Number of hidden layer nodes: 64; Dropout Rate: 0.1; Regularization: L2; Regularization Rate: 0.01]
Neural Network - Revenue	[Hidden Layer Size: 3; Number of hidden layer nodes: 128; Dropout Rate: 0.1; Regularization: None; Regularization Rate: 0]

XGBoost Regression - Audience Score	[ETA: 0.1; Max Depth: 4; Subsample: ; Column Sample per Tree: 0.8; Lambda: 1; Alpha: 0.01]
XGBoost Regression - Tomato (Critics) Score	[ETA: 0.1; Max Depth: 4; Subsample: 0.8; Column Sample per Tree: 0.9; Lambda: 0.1; Alpha: 0.01]
XGBoost Regression - Revenue	[ETA: 0.1; Max Depth: 4; Subsample: 0.7; Column Sample per Tree: 0.8; Lambda: 1; Alpha: 1]

Fig #5 - Results of Predictive Models

Normalizd RMSE Comparison

	GLM	XGBoost	Neural Network
Audience - Train	1	0.686	0.853
Audience - Validation	1	0.668	0.779
Audience - Test	1	0.682	0.776
Critics - Train	1	0.849	0.944
Critics - Validation	1	0.820	0.902
Critics - Test	1	0.850	0.923
Revenue - Train	1	0.794	1,090
Revenue - Validation	1	0.802	0.990
Revenue - Test	1	0.750	1,307