Predicting Movie Box Office Sales

by

Ekaterina Pesherov

Dr. Piotr Szczurek, Advisor

Masters project submitted in partial fulfillment of the

requirements for the Master of Science in Data Science degree

in the College of Aviation, Science, and Technology of

Lewis University

*Abstract*

The goal of this project is to analyze the Box Office Dataset and the Rotten Tomatoes Movie Dataset to create a model that can predict the all-time Domestic Box Office sales based on Rotten Tomatoes critics ratings. Prior to movie release in theaters, critics are invited to view the movie and provide reviews on several online and paper forums. In the first step of this analysis, I identified which components are the most influential in predicting Domestic Box Office Sales. Second, I showed that it is possible to build a powerful prediction model using regression. Lastly, I confirmed that the content of critics’ reviews can positively increase the accuracy of predicting Domestic Box Office Sales.

*Index Terms*—Box Office Sales, critic ratings, movie, natural language processing, prediction analysis, regression, Rotten Tomatoes

# Introduction

Performing Box Office Revenue prediction and analysis has been a long-standing practice for the film industry. Production companies are constantly making predictions and calculating risks to determine which investments will give the highest return. However, previous statistical models published online have not taken into consideration the value of critic ratings in predicting Box Office Sales. Critics have a good understanding of what audiences will like and a critics rating can be an invaluable indication of how much a movie can make in the Box Office.

Companies have many uses for early predictions of Box Office revenue. One scenario where having a projected estimate on total Box Office Revenue is crucial, may be in negotiations for Television rights, Video-On-Demand, and Streaming deals. Revenue from these after-market options are continually becoming a major revenue source for production companies [12].

In this analysis, I will be developing a predictive model that can predict the total Domestic Box Office Sales prior to the wide distribution of the movie by using the Rotten Tomatoes critic reviews. In addition, I will be coupling these ratings with some other movie attributes such as movie rating, genre, director, number of famous actors, etc. to increase predictive accuracy. By using this unique combination of datasets and features, I hope to provide an improvement to the previous Box Office prediction models.

## Ethical Considerations

In this paper, the 6 pillars of ethics were taken into consideration. Voluntary participation and Informed consent did not apply to this study, since there was no live participant interaction. Anonymity and Confidentiality was considered with the data and no names or identifiers to any persons was released. In addition, the data is public knowledge, and did not require consent to access. Considering the above, there was no Potential for Harm. In addition, I cited all references used and did not violate the Results Communication.

# Data Description

For this analysis, I combined 3 datasets (‘The Numbers’ Domestic Box Office Dataset, Rotten Tomatoes Movie Dataset and Rotten Tomatoes critic review dataset).

## Domestic Box Office Dataset

This dataset contains information on 3,822 movies. I scraped the data from The Numbers website to create a dataset which includes the top 100 best performing movies released each year between 1980-2020 [1]. Table I contains the description and attributes of the movies including the movie title, distributor, Domestic Box Office sales, Opening Weekend Box Office sales, and number of theaters showing the movie [1].

1. Features OF Domestic Box Office Sales

| Attribute | Type | Example Value | Description |
| --- | --- | --- | --- |
| Rank | Numeric (int) | 1 | Movie Rank according to Box Office Total |
| Movie Title | Nominal (string) | Dolittle | Title of the movie |
| Distributor | Nominal (string) | Warner Bros | Name of the Distribution Company |
| Domestic Box Office | Nominal (string) | $204,417,855 | Total amount of Box Office Revenue for that year. \*The Domestic Market is defined as the North American movie territory (consisting of the United States, Canada, Puerto Rico, and Guam). |
| Opening Weekend Box  Office | Nominal (string) | $62,504,105 | Total amount of Box Office Revenue for the opening weekend. \*The Domestic Market is defined as the North American movie territory (consisting of the United States, Canada, Puerto Rico, and Guam). |
| Number of Theaters Playing | Numeric (int) | 3,000 | Number of Theaters playing this movie title |
| Release Year | Numeric (int) | 2020 | Year that the movie was released |
| Adj Domestic Box Office | Numeric (float) | 204417855.0 | Adjusted Domestic Box Office Sales based on inflation rate |
| Adj Opening Weekend Box Office | Numeric (float) | 62504105 | Adjusted Opening Weekend Box Office Sales based on inflation rate |

First thing to note about this data, is that because it has been scraped from a website and the website is updated daily to reflect the latest studio reports, the values represented in this dataset are only accurate up to the date of extraction, December 7, 2021. In addition, I wanted to clarify when referring to the Domestic Market in this dataset, it is referring to the North American movie territory (consisting of the United States, Canada, Puerto Rico, and Guam) [1]. Lastly, the Box Office Sales column contains data that has not yet been adjusted for inflation. I used the Consumer Price Index table from the Federal Reserve Bank of St. Louis (FRED) [4] to adjust the sales column to ensure the data is comparable over the years.

## FRED Economic Data Inflation Dataset

1. Features of the Fred economic Data inflation Dataset

| Attribute | Type | Example Value | Description |
| --- | --- | --- | --- |
| Year | Ordinal (date) | 1995 | Year in comparison for current inflation value |
| CPIAUCSL | Numeric (int) | 22.33 | A measure of the average yearly change in the price for goods and services paid by urban consumers between any two time periods. |
| Inflation Adjustment | Numeric (int) | 11.59 | Calculated Value; current year/comparison year |

The Data Inflation dataset (Table II) contains the attributes year and US CPI Index and ranges from the years 1947-2020 [4]. The “Inflation Adjustment” attribute is a calculated column that I added to the dataset using a simple adjustment formula (1) to make an inflation factor for each year [5]. I used that value to calculate the ‘Adj Domestic Box Office’ and ‘Adj Opening Weekend Box Office’ columns in the Box Office dataset using the adjusted sales price equation (2) [5].

Equation (1)

Equation (2)

## Rotten Tomatoes Datasets

To pair with the Box Office dataset, I extracted data from the Rotten Tomatoes Movies dataset found on Kaggle.com [2]. This dataset contains movie ratings and reviews scraped from the publicly available Rotten Tomatoes website (https://www.rottentomatoes.com) as of 2020-10-31 [2]. The Rotten Tomatoes dataset has 2 tables, a table called “rotten\_tomatoes\_movies” including detailed movie information such as movie title, description, genres, duration, director, actors, users' ratings, and critics' ratings and a second table called “rotten\_tomatoes\_critic\_reviews” containing each critic review, rating, publication date, and review content [2]. One thing to note between the reviews and movies datasets, is that it took a few days to scrape the reviews dataset from the website due to the large amount of content available so there may not be full consistency between some columns of the movies and reviews dataset such as “tomatometer\_count” and "tomatometer\_top\_critics\_count" because the reviews dataset was scraped first.

To join the Box Office dataset and the Rotten Tomatoes Movie dataset, I will be using a combination of attributes (movie ‘title’ and ‘release year’) to uniquely match the datasets.

## Rotten Tomatoes Movies Dataset

The Movies dataset (Table III) contained 17,712 rows of movie information and summarized audience and critic review scores dating from 1914-2020.

1. Features from the Rotten Tomatoes Movie Dataset

| Attribute | Type | Example Value | Description |
| --- | --- | --- | --- |
| rotten\_tomatoes\_link | Nominal (primary key) | m/1021312-three\_amigos | Unique item identifier. Link from which the movies data has been scraped – e.g., the record " m/1021312-three\_amigos " has been scraped from "https://www.rottentomatoes.com/ m/1021312-three\_amigos " |
| movie\_title | Nominal (string) | Three Amigos | Title of the movie as displayed on the Rotten Tomatoes website |
| movie\_info | Nominal (string) | “Three cowboy movie stars from the silent era -- Dusty Bottoms (Chevy Chase), Lucky Day (Steve Martin) and Ned Nederlander ...” | A brief description of the movie |
| critics\_consensus | Nominal (string) | “Three Amigos! stars a trio of gifted comedians and has an agreeably silly sense of humor, but they're often adrift in a dawdling story with too few laugh-out-loud moments.” | Comment from Rotten Tomatoes |
| content\_rating | Nominal (string) | PG-13 | Category based on the movie suitability for audience |
| genre | Nominal (string) | Comedy | Genre category of the movie. Movie genres separated by commas, if multiple |
| directors | Nominal (string) | Steven Spielberg | Name of the director(s) |
| authors | Nominal (string) | George Lucas | Name of the author(s) |
| actors | Nominal (string) | Samuel L. Jackson | Name of the actors |
| original\_release\_date | Ordinal (date) | 12/12/86 | Date when the movie was released for the Box Office |
| streaming\_release\_date | Ordinal (date) | 10/22/20 | Date when the movie was released for streaming |
| runtime | Numeric (int) | 124 | Movie runtime in minutes |
| production\_company | Nominal (string) | Disney | Name of the production company |
| tomatometer\_status | Nominal (string) | Fresh | "Rotten" (less than 60% positive reviews), "Fresh" (at least 60% of positive reviews), and "Certified Fresh" (at least 75% of positive reviews, at least 80 reviews of which at least 5 from top critics) |
| tomatometer\_rating | Numeric (int) | 65 | Percent of critics gave positive rating |
| tomatometer\_count | Nominal (string) | 244 | Number of critic ratings counted for the calculation of the tomatomer status |
| audience\_status | Nominal (string) | Spilled | Audience value of "Spilled" (less than 60% of users gave a rating of at least 3.5) or "Upright" (at least 60% of users gave a rating of at least 3.5) |
| audience\_rating | Numeric (int) | 53 | Percent of audience that rated the film positive |
| audience\_count | Numeric (int) | 2400 | Number of ppl in audience that rated the film |
| tomatometer\_top\_critics\_count | Numeric (int) | 43 | Number of top (“certified”) critics that rated this movie |
| tomatometer\_fresh\_critics\_count | Numeric (int) | 73 | Number of critics that rated the movie positive |
| tomatometer\_rotten\_critics\_count | Numeric (int) | 76 | Number of critics that rated the movie negative |
| release\_year | Numeric (int) | 2020 | Extracted “year” value from “original\_release\_date” |
| ActorPopularityScore | Numeric (float) | 1.289524 | Calculated value of the sum of actor’s popularity |
| ActorPopularityScore | Numeric (float) | 0.745462 | Calculated value of the sum of directors’ popularity |

First, I evaluated the dataset for missing values. I replaced null values in the attributes genre, directors, actors, production company with the string “unknown”. Since the data in these columns is categorical, adding the “unknown” string will allow the data to claim a value where data is not available. For the “runtime” attribute, I decided to fill the missing values in the column with the mean because this column is numerical. One column where I was not able to replace missing values in was the “original\_release\_date” as this value was used to confirm the unique identity of the movie, so rows with a missing value in the column needed to be dropped. Afterwards, I was able to extract the release year from the date string into a new column “release\_year”.

To join the Box Office dataset and the Rotten Tomatoes Movie dataset, I used the fuzzywuzzy library from Python and iterated through each title in the Box Office dataset to find all potential string partial matches from the Rotten Tomatoes dataset. If the movie title had a 90+% match with the original title and the movie was released on the same year then the data was merged. Of 3,823 Box Office titles, I was able to find 3,444 matches in the Rotten Tomatoes Dataset (90% match).

To double check the matching was accurate, I checked if the 2 movie title strings were an exact match in the new merged dataset. I found that only 330 movie titles were not an exact match. I manually scanned through those 330 titles and removed the row of any incorrect matches (about 30 rows).

## Feature Encoding

Some of the features I wanted to use in the predictive model were categorical strings. To be able to apply many algorithms, it is required to convert string values to numerical values. The ‘content\_rating’ feature was transformed from a string to an integer using ordinal feature mapping. The “genres” column contained a string of applicable genres to each movie. I started by converting the string to an array and one-hot encoding the genre labels.

In addition, I generated 2 new features the ‘ActorPopularityScore’ and the ‘DirectorPopularityScore’. The movies dataset contains a column of containing a string list of actor names. I parsed the actor’s names from the list and created a python dataframe containing the actor’s name total movie count based on the number of times they have been listed in this database. This dataset only contains the top 100 best performing movies in the Box Office so by counting how many times each actor starred in these movies we can consider it a rating of popularity. The actors with the largest count, who played in the most top 100 movies, can be considered the most popular. I found that Samuel L. Jackson (count: 67) and Bruce Willis (count:55) were the 2 most popular actors in the dataset. I then used the sklearn preprocessing MinMaxScaler() to transform the actor count values to normalized values between 0-1. The aggregate value of actor popularity for each movie is stored as the column ‘ActorPopularityScore’. Since we can only say that actors who are often in top 100 Box Office movies are popular, I chose to have the popularity\_score be only positive float. In this case, the weight or popularity of each actor could only increase the overall actor popularity score and unpopular actors can not detract from the popularity\_score. This function was also performed on the ‘directors’ column and the score was saved as the ‘DirectorPopularityScore’.

One error I saw was that ‘Jr’ and ‘Unknown’ were flagged as having one of the largest counts (aka most popular). To remedy the situation, I changed the counter and popularity score to 0 for those values so they didn’t carry any weight when summing the popularity score.

## Rotten Tomatoes Critics Review Dataset

Table IV contains the attributes and descriptions for the Rotten Tomatoes Critics Review Dataset. The dataset contains 1,130,017 rows of critic’s movie reviews. Each movie can have multiple reviews, so this dataset is significantly larger than the movies dataset. I was able to filter this dataset by performing a one-to-many merge between the movies and the critics dataset based on the ‘rotten\_tomatoes\_link’ unique identifier. After the merge the dataset contained only 474,631 rows of relevant review content.

1. Features from the Rotten Tomatoes Critic Reviews Dataset

| Attribute | Type | Example Value | Description |
| --- | --- | --- | --- |
| rotten\_tomatoes\_link | Nominal (primary key) | m/1021312-three\_amigos | Unique item identifier. Link from which the movies data has been scraped – e.g., the record " m/1021312-three\_amigos " has been scraped from "https://www.rottentomatoes.com/ m/1021312-three\_amigos " |
| critic\_name | Nominal (string) | Roger Ebert | Name of the critic who rated the movie |
| top\_critic | Boolean Value | True | Boolean value that clarifies whether the critic is a top critic or not, per Rotten Tomatoes guidelines (https://www.rottentomatoes.com/critics/top\_critics) |
| publisher\_name | Nominal (string) | New York Times | Name of the publisher that the critic works for |
| review\_type | Nominal (string) | Rotten | Was the review rotten or fresh |
| review\_score | Nominal (string) | 3/5 | Review score provided by the critic |
| review\_date | Ordinal (date) | 11/20/20 | Date of the review |
| review\_content | Nominal (string) | “Three Amigos! stars a trio of gifted comedians and has an agreeably silly sense of humor, but they're often adrift in a dawdling story with too few laugh-out-loud moments.” | Content of the review |
| review\_content\_unique | Nominal (string) | [‘stars’, ’gifted’, ’comedians’, ‘agreeably’, ‘silly’, ’moments’] | Extracted list of “significant” words from ‘review\_content’ |
| review\_vec\_norm | Numeric (float) | 1.3184 | Normalized vector representing the ‘review\_content\_unique’ |

To clean the review column of this dataset and prepare it for natural language processing, I removed the punctuation from the ‘review\_content’ and converted all the words to lowercase. I then tokenized the words by splitting the string into a list of words and removed any “stop words” (defined by the nltk library). I took a look at the frequency of words available in the reviews. I removed the top 50 most popular words that don’t contribute to the review such as “movie”, “film”, “one”, “like”, “story”, etc. This will reduce the background noise in the algorithm. I also removed the least popular words from the dataset. These are words that were mentioned less than 1000 times. This left me with approximately 900 “significant” words to use for NLP. I created a new column called ‘review\_content\_unique’ and for each review removed any nonsignificant words such that I was only left with a list of unique words.

# Methodology

I used many different techniques to approach the analysis of this dataset. In previous sections I have discussed how I cleaned and prepared the data to be used, now I will be applying different tools to generate an optimized prediction model.

## Data Examination

The first step in this analysis was to perform a Univariate Profile and Bivariate Profile on some the features in this dataset. I plotted the data onto graphs such as histograms and scatterplots to examine the relationships between variables in the dataset and help support our hypothesis that we can predict Box Office Revenue Sales based on critic’s reviews, cast popularity, movie type, etc.

A common way to visualize the attributes in multivariate analysis is to use the scatterplot or correlation matrix. This matrix contains all combinations of variables and displays the data using scatterplots, histograms, and correlation coefficients. Plotting the variables in this way, I can examine their relationships and identify which variables will be the most useful in this analysis. The lower left quadrant of the matrix contains scatterplots, the diagonal contains histograms, and the upper right quadrant contains correlation coefficient values. The correlation coefficient is a value between -1 to 1 which represents the measure of linearity. A value near 1 indicates a relationship where the values trend together and a value near -1 indicates a relationship where 1 variable increases and the other variable decreases in a fixed proportion. A value closer to 0 indicates no relationship.

Our goal when examining the correlation matrix is to identify if there is enough significant correlation in the dataset to warrant factor analysis and to get a quick view of which variables will play the largest role in determining the factors. If a variable has very low correlation value, then it may not play any part in the factor analysis and if a variable has a large number of correlations, it may be part of several factors.

## Exploratory Factor Analysis

In addition, to the correlation matrix, I used the Measure of Sampling Accuracy (MSA) and partial correlations to determine which variables to use in the Factor Analysis. Factor analysis will always derive factors, but it is important to maintain a base level of statistical correlation between variables that are chosen for the analysis to ensure that the resulting factor structure has some objective basis [6]. Using the MSA test we can identify which variables lack enough significance in their correlation and should be removed. Generally, an MSA value of 0.5 or higher is acceptable.

After verification and selection of significant variables we can implement the factor model. The correlation matrix is then transformed through estimation of a factor model to obtain a factor matrix containing factor loadings and communality values for each variable on each derived factor. The loadings of each variable on the factors are then interpreted to identify the underlying structure of the variables. Before applying the model, we must determine the number of components to be used in the model. We use the eigenvalues for each variable to understand their relative explanatory power. The latent root criterion requires that factors to be considered should have a value above 1.0. Another way to confirm the number of factors is the use of the Scree Test. The Scree test is a line plot of the eigenvalues. The elbow of the curve on the plot determines the number of factors that should be used in the component analysis. Combining these techniques yielded a more holistic result in choosing the number of components for the model.

Lastly, I evaluated the unrotated and rotated factor matrices for significant factor loadings and adequate communalities. Factor loadings, in either the unrotated or rotated factor matrices, represent the degree of association of each variable with each factor. The goal of this analysis is to maximize the association of each variable with a single factor. The communality for each variable shows the amount of variance in a variable that is accounted for by the two factors taken together. I used communality values to determine if all the factors belonged in the factor analysis. The factor loading is how much influence the variable contributes to that factor [6]. By studying these patterns, one can determine which variables are best explained by which factors. The goal of Factor Analysis in this paper was to identify the most important features and themes in the dataset for predicting Domestic Box Office sales.

## Linear Regression

After identifying the core factors of this analysis, I split the dataset into testing and training sets with an 80/20 random split. Starting off, I always try to see if I can fit my dataset into the simplest predictive model, Linear Regression. In any type of analysis, I prefer to use the simplest model, if possible, to do so without sacrificing accuracy. The simpler a model is, the easier it is to understand it and make any changes to it, as needed. In this case, I used the Stepwise Linear Regression with Ordinary Least Squares from Python’s statsmodel.api to perform this initial examination. In stepwise estimation, one variable is added to the model at a time until maximum model accuracy is achieved. By adding the variables stepwise, I can incrementally check the results of the model and make any adjustments (insertion or deletions of variables) as necessary. The first variable is chosen based on the highest bivariate correlation with the dependent variable and the following variables are chosen based on the highest partial correlation.

To measure the model’s performance, I mainly used the R square, Adjusted R square, and F statistic. R square, also known as coefficient of determination, is the correlation coefficient squared. This value indicates the percentage of total variation of the dependent variable explained by the regression model consisting of independent variables. The adjusted R square, additionally, takes into account the number of independent variables included in the regression model and the sample size. Usually, the addition of independent variables will cause the R square to rise, but the adjusted R square may fall if the added independent variables have little explanatory power or if the degrees of freedom become too small. The F-statistic is a ratio of the model mean square and the residual mean square. The F-statistic reveals how much predictive capability the model has by comparing the model to the null hypothesis which is assumed to have no predictive power.

In addition, to looking at the statistics of the overall model, I used several measures to analyze each independent variable such as the regression coefficient, standard error of the coefficient and the t value. The regression coefficient represents the amount of change on the dependent variable for each unit change in the independent variable. We can use this value to compare how much each variable is influencing the result of the dependent variable. The standard error of the regression coefficient is an estimate of how much the regression coefficient will vary between samples of the same size taken from the same population. In a simple sense, it is the standard deviation of the estimates across multiple samples. Lastly, the t value is the significance of the partial correlation of the variable. This measure indicates the confidence level that the coefficient is not equal to zero. The t value can be used to determine if the independent variable should be dropped from the model if the variable falls below the threshold.

To evaluate if the linear model was a good fit for the dataset, we must also evaluate the variate for meeting these assumptions. These assumptions include linearity, homoscedasticity, independence of the residuals, and normality. The main measure used to evaluate the regression variate is the residuals. The residual is the difference between the predicted value and the actual value. I plotted the residuals in a scatter plot to check for linearity and homoscedasticity. Homoscedasticity refers to consistency of residual values across all values, thus we are looking for the scatter plot to have no defined pattern.

## Non-linear Regression

If the assumptions for linear regression analysis aren’t met, I can test the data against non-linear Regression models. Python’s sklearn library contains the following regression functions to easily run the following algorithms and compare their R2 values for: Linear Regression, Decision Tree Regression, Random Forest Regression, Gradient Boosting Regression and XGBoost Regression.

A Decision Tree model uses decision-based logic to classify samples. Each node has a condition of a feature (aka a rule) and the node decides how to navigate down the tree. Once the leaf node is reached, an output is predicted [14]. At the top of the Decision Tree is the best predictor also known as the ‘root’ node [13]. As you traverse down the tree, the features become more nuanced and less influential in the prediction. For the correct prediction to be made, generally the right sequence of conditions must occur. One major issue is that Decision Trees make specific rules that work extremely well on the training dataset, however, don’t apply well to the testing dataset because model tends to overfit [13].

The Random Forest Model uses multiple decision trees to operate as an ensemble [15]. Each tree is responsible for a class prediction and the tree with the most votes becomes the prediction [15]. By using multiple decision trees, we can compensate for any individual errors. This model is less prone to overfitting like the Decision Tree Model because we use the majority vote to select the predicted output [14].

Gradient Boosting regression is very similar to the Random Forest Model. It uses multiple decision trees to operate as an ensemble for prediction [13]. The additional benefits of using Gradient Boosting is that contains an optimized loss function and can convert weak learners into single strong learners in a stepwise fashion [13].

XGBoost regression is a type of Gradient Boosting regression. This specific model has become very popular in the data science community and is well known for high efficiency and flexible performance [16]. Some unique features of this algorithm is that is uses Newton boosting, extra randomization, regularization on leaf weights and more [17].

## Natural Language Processing

Lastly, I wanted to use one more technique to see if I can improve the predictive accuracy of the non-linear models by including the critic review content. Using a natural language processing algorithm called “bag-of-words” I converted critics’ reviews into a vectorized number. I first cleaned the “review\_content” column as described in Section II of this paper by tokenizing the review and selecting the most unique words from the review. I then performed a text-to-number transformation using the bag-of-words model from Python’s gensim library [11]. Once I had every review in vector form, I used a norm function to assign it a number equivalent to the vector space. Then for each movie, I aggregated the review vectors to get the mean review vector for each movie and used that value as an additional attribute in the dataset called ‘review\_vec\_norm’. With the updated database, I reran the sklearn regression models and see if there was an improvement in the prediction accuracy with the addition of the new variable ‘review\_vector’.

# Results and Discussion

## Data Examination

I started this analysis, by first examining the features available in these datasets. To get a quick view of multiple variable combinations, I used a scatterplot matrix (Fig. 1). The lower left quadrant of the matrix contains scatterplots which can be used for bivariate profiling, the diagonal contains histograms used for univariate profiling, and the upper right quadrant contains correlation coefficient values which represent a numerical value of the scatter plots. I selected a few variables from the dataset which I suspected of having a potential to be influential in predicting Box Office Sales. These variables included the 'Adj Domestic Box Office', 'Adj Opening Weekend Box Office', 'Number of Theaters Playing', 'tomatometer\_rating', 'tomatometer\_top\_critics\_count', 'ActorPopularityScore', and 'DirectorPopularityScore'. I found that all of variables I chose had a fairly high correlation coefficient (>0.2) with the dependent variable “Adj Domestic Box Office” sales. The best correlation in the matrix was with the “Adjusted Opening Weekend Box Office” sales and variable “Adj Domestic Box Office” sales with a value of 0.762. Looking at the correlation plot, I can estimate that the relationship may even be linear. Another interesting correlation was between ‘Adjusted Opening Weekend Box Office’ and ‘tomatometer\_top\_critic\_count’ which had the second highest correlation with a value of 0.431.

Looking at the histograms for the variables, I can see that most of the distributions appear normal. This indicates that we may be able to use linear regression for predicting the Box Office Sales in this dataset. One histogram from the matrix particularly sticks out in Figure 1, I can see that the histogram of “tomatometer\_rating” attribute is fairly widespread. This indicates that there is a fairly even selection of positive and negative critic reviews in the dataset and that the dataset is not skewed towards only positive reviews.

A picture containing diagram

Description automatically generated

Figure 1: Scatterplot Matrix of variables ['Adj Domestic Box Office', 'Adj Opening Weekend Box Office', 'Number of Theaters Playing', 'tomatometer\_rating', 'tomatometer\_top\_critics\_count', 'ActorPopularityScore', 'DirectorPopularityScore']. The lower left quadrant of the matrix contains scatterplots, the diagonal contains histograms, and the upper right quadrant contains correlation coefficient values.

## Exploratory Factor Analysis

Using the correlation matrix, I can definitively say that there is enough significant correlation within the dataset to warrant factor analysis. Next, I created a partial correlations table with the measure of sampling adequacy (MSA) value along the diagonal of the matrix (Fig. 2). Values that are significant are shown in darker colors (red for positive values and blue for negative) and values that are close to 0 are shown in a very light color or white. I used the MSA test to identify which variables lacked enough significance in their correlation and should be removed.

I can see that the overall MSA value is above the acceptable range (>0.5) with a value of 0.63. Looking at the MSA value for individual variables, I can see that variables 'tomatometer\_rotten\_critics\_count' and the ‘western’ genre have MSA values below 0.5. To maintain good statistical correlation among the variables, I first omitted 'tomatometer\_rotten\_critics\_count' which had the lowest MSA value and recalculated. I saw that the feature ‘western’ was still below the threshold and I removed that variable from the analysis as well.

Chart, bar chart

Description automatically generated

Figure 2: Partial Correlation Matrix with MSA value along the diagonal

Next, I extracted the eigen values into a table (Fig. 3) and used the Scree Test (Fig. 4) to understand the number of components best suited for the model. The Scree Test results did not come out as expected. The graph appeared to have multiple elbows and did not plateau as usual. However, based on the first elbow I determined the number of components should be 4 or 5.

To confirm the number of components to be used, I referred to the Eigenvalue table (Fig. 3). Using the latent root criterion, I saw that 4 factors have an eigenvalue above 1.0 so 4 factors should be retained. When looking at the eigenvalue of Factor 5, a value of 0.0, this value was too low to be included thus we retained only 4 factors. The main issue I found was that with those 4 factors, only 36% of the variance in the dataset was represented. I suspect the main reason for these poor results are due to using mixed data in this analysis. Ideally, it would be best to have attributes that are all continuous values and comparable, however this dataset has continuous, ordinal, and binary values.

Table

Description automatically generated Chart, line chart

Description automatically generated

Figure 3: Eigen Value Extraction Table Figure 4: Scree Test For Component Analysis

Despite the low values in the eigen value chart, I decided to proceed with the Varimax Rotated Factor Analysis using the FactorAnalyzer library from Python. Since I was performing the factor analysis for exploratory purposes and didn’t intend to perform dimensional reduction, the low values were acceptable. Figure 5 show the results of VARIMAX Rotated Factor Matrix. Adding the rotation to the Factor Matrix allowed me to redistribute the factor loading pattern. In addition, I removed any values in the matrix that were below the significance threshold of 0.35. In Figure 5, the pattern of significant factor loadings is visible, and each variable has a clear factor that it belongs to. However, 3 factors in the figure are cross-loaded (tomatometer\_rating, Adj Opening Weekend Box Office, and Number of Theatres playing). I removed each factor 1 at a time to see how the factor loading was affected. After removing tomatometer\_rating and Adj Opening Weekend Box Office, the matrix had only single loaded factors and was suitable (Fig. 6).

Calendar

Description automatically generated Calendar

Description automatically generated with low confidence

Figure 5: Varimax Rotated Component Analysis Figure 6: FINAL Varimax Rotated Component Analysis

The use of exploratory factor analysis in data analysis is to find themes in the data. Based on the groupings from Figure 6, the main proponents in predicting the Box Office Sales are the following themes: Factor 1: Movie Popularity and Critical Acclaim, Factor 2: Family Friendly Content, Factor 3: Actor and Director Popularity and Factor 4: The “Blockbuster” Formula (Comedy, Action, Suspense).

## Linear Regression

For the next step in this analysis, I used a Stepwise Linear Regression model to predict Domestic Box Office Sales. If possible, it is always best to use the simplest models with the least number of attributes as necessary to get an accurate prediction. As a baseline in my analysis, I attempted to fit my dataset into a Linear Regression model. In a stepwise fashion, I added each variable to the model and evaluated the results. I removed any variables were the t-value and regression coefficient were too low, or if the addition of the variable caused the Adjusted R2 or F-statistic to go decrease instead of increase. The final results of the optimized regression model are in Figure 7.

A screenshot of a computer

Description automatically generated with low confidence

Figure 7: Linear Regression with Ordinary Least Squares Results

The OLS Linear Regression model performed poorly with a final R2 of 0.353. The variables that were used were chosen because they had the highest correlation coefficients and were selected from the Factor Analysis. An indication of poor performance of this model was that the coefficients in the table were below 0.001. This indicates that these variables are not predicting the variance of the Adjust Box Office Sales well. I suspected that the problem was that the model was not linear. To check for linearity of the model, I plotted the residual values vs the predicted value (Fig. 8). If the model was linear then we would expect to see a random pattern in the plot. However, the plot in Figure 8 does not appear to have a random pattern. This suggests that the equation we have made for the regression model is nonlinear and thus a linear regression model is not an appropriate fit.

Chart, scatter chart

Description automatically generated

Figure 8: Scatter Plot of Residual Values from Linear Regression

## Non-linear Regression

Since the Linear Regression model did not perform well, I tested a few non-linear regression models from Python’s sklearn library on the dataset. I ran the dataset on the following models: Linear Regression, Decision Tree Regression, Random Forest Regression, Gradient Boosting Regression, and XGBoost Regression. To compare the results of these algorithms, I used the R2 value to determine which had the best accuracy in predicting Box Office Sales. These results can be seen in Table V. The Random Forest model performed the best on the testing dataset and had an R2 of 0.53. Gradient Boosting was the second-best performing algorithm and had an R2 of 0.52. The Linear Regression results in this model are comparable to the results from the OLS model.

This dataset contains the same variables used in the factor analysis except the Adj Opening Weekend Box Office was removed. Due to the concept of data leakage, I was not able to use the Adj Opening Weekend Box Office attribute to predict total Box Office sales because the Opening Weekend sales are included in the calculation of the total Domestic Box Office attribute. Data leakage occurs “when the data you are using to train a machine learning algorithm happens to have the information you are trying to predict.” [18]. The main issue is that data leakage can cause invalid predictive models that are too good to be true. For this analysis, I’ve decided to remove this attribute prior to training the model so it did not incorrectly inflate the accuracy results of the model.

1. Accuracy of Regression Models

|  |  |  |
| --- | --- | --- |
| **Regressor** | **Training Dataset (R2 )** | **Testing Dataset (R2 )** |
| Linear Regression | 0.36463 | 0.39079 |
| Decision Tree Regression | 1.00000 | 0.23132 |
| Random Forest Regression | 0.92214 | 0.52887 |
| Gradient Boosted Regression | 0.68266 | 0.51770 |
| XGBoost Regression | 0.83619 | 0.49347 |

Utilizing the best performing model, Table VI consists of the most influential features used in the Random Forest Model. This list only includes features that had an importance value of 0.002 or higher. Of the 18 most important features in Table VI, a majority of the features match with those identified by the Factor Analyzer, however 5 new columns were added as significant: ‘Romance’, ‘Classics’, ‘Musical & Performing Arts’, ‘Documentary’, and ‘Gay & Lesbian’

1. Feature Importance From Gradient Boosted Regression

Table

Description automatically generated

## Natural Language Processing

In the last step of this analysis, I wanted to see if I could improve the performance of the regression models by adding a new attribute. In addition, to the critic’s overall movie tomatometer\_rating and number of critics who post reviews on the movie, I wanted to analyze the content of the critic’s reviews to identify patterns of speech that could improve the Box Office Sale predictions. I first tokenized the critic reviews, identified which words had the most significance, and removed any nonsignificant words. I then applied the ‘Bag of Words’ algorithm to convert the words into a vector and used Python’s linalg library to normalize the vectors into a number. For each review, I now had a number representing the content of the review. To use this value with the Rotten Tomatoes movies dataset, I had to aggregate all of the review norm. vectors for each movie into a single value, the mean of the normalized vectors.

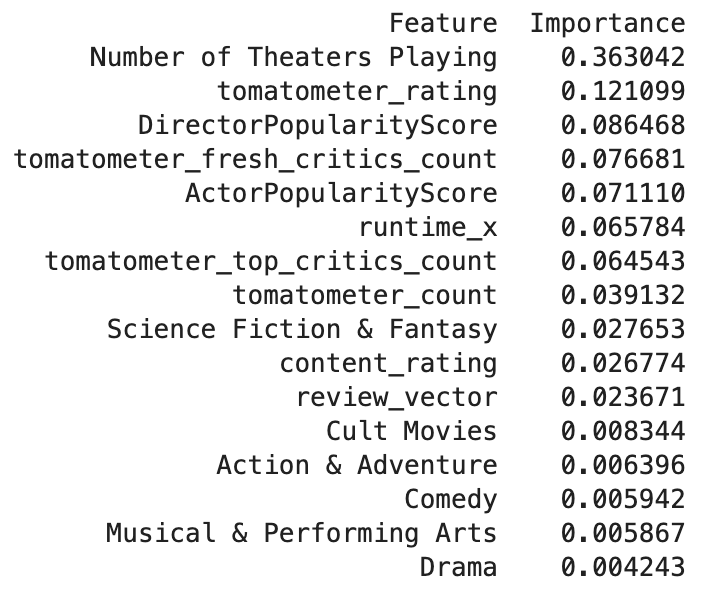
With this new attribute added to the dataset, I repeated the 5 regression algorithms run previously. I found that again that Random Forest Regression performed the best with an R2 of 0.56. This was a 3% improvement to the previous value of 0.53. In fact, all of the regression models saw a boost of about 3% to their R2 value. That is confirmation that the text-to-number transformation of the ‘review\_content’ feature was successfully able to increase the model’s ability to predict Box Office Sales. This is also further proof on the ability to replace a vector of text into a number without significant data loss as demonstrated in a previous paper by Sergey Burukin [11].

1. Accuracy of Regression Models Post Addition of Review Vector

|  |  |  |
| --- | --- | --- |
| **Regressor** | **Training Dataset (R2 )** | **Testing Dataset (R2 )** |
| Linear Regression | 0.36463 | 0.40164 |
| Decision Tree Regression | 1.0000 | -0.16295 |
| Random Forest Regression | 0.92214 | 0.56427 |
| Gradient Boosted Regression | 0.68266 | 0.53321 |
| XGBoost Regression | 0.83619 | 0.54924 |

For further confirmation on the performance of this new attribute, I can see that the ‘review\_vector’ was included as #11 on the feature importance list in Table VIII. Not only has the addition of NLP increased the accuracy of the Random Forest Regressor but it has also simplified the important features from 19 features to 16.

1. Feature Importance From Gradient Boosted Regression Post Addition of Review Vector



# Conclusion

## In this analysis, I took the Domestic Box Office dataset and the Rotten Tomatoes movie and critics’ reviews dataset to see if I could create a model to predict Domestic Box Office Sales. I was able to identify the most influential features in predicting Box Office Sales, created a model that could accurately predict Box Office Sales prior to movie release, and determined the content of the critics’ reviews could be transformed from text-to-number and improve prediction accuracy. Being able to accurately predict Box Office Sales before movies are distributed to audiences can be crucial for many companies. This paper proved that this was not only possible but also can be greatly improved by including critic reviews in the analysis. Critics are movie experts and can prove to play a crucial role in determining a movies’ total Box Office Sales.

## For further analysis, I would have liked to improve how I performed the Bag of Words algorithm. I would have liked to separate the reviews by positive and negative scores and create vector values that reflected if the words were associated with a positive or negative review. I could then perform a better aggregation of the vectors and further improve the performance of Gradient Boost Regression.

References

[1] *Top 2020 movies at the Domestic Box Office*. The Numbers. (n.d.). Retrieved December 8, 2021, from https://www.the-numbers.com/box-office-records/domestic/all-movies/cumulative/released-in-2020.

[2] Leone, S. (2020, November 4). *Rotten tomatoes movies and critic Reviews Dataset*. Kaggle. Retrieved December 8, 2021, from https://www.kaggle.com/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset.

[3] Brownlee, J. (2020, August 17). *How to use feature extraction on tabular data for Machine Learning*. Machine Learning Mastery. Retrieved December 8, 2021, from <https://machinelearningmastery.com/feature-extraction-on-tabular-data/>.

[4] U.S. Bureau of Labor Statistics, Consumer Price Index for All Urban Consumers: All Items in U.S. City Average [CPIAUCSL], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/CPIAUCSL, December 9, 2021.

[5] *Adjusting a data series for inflation - youtube*. (n.d.). Retrieved December 9, 2021, from https://www.youtube.com/watch?v=mHyKL-m-rsk.

[6] J. F. Hair, W. C. Black, B. J. Babin, and R. E. Anderson, Multivariate data analysis. Andover, Hampshire: Cengage, 2019.

[7] Aher, A., Kannan, R., & Vispute, S. (2021, August 7). *Data Analysis and price prediction of Black Friday sales using Machine Learning Techniques*. International Journal of Engineering Research & Technology. Retrieved December 10, 2021, from https://www.ijert.org/data-analysis-and-price-prediction-of-black-friday-sales-using-machine-learning-techniques.

[8] Dabbura, I. (2019, December 15). *Predicting loan repayment*. Medium. Retrieved December 10, 2021, from https://towardsdatascience.com/predicting-loan-repayment-5df4e0023e92.

[9] Tavoosi. (2019, July 19). *Predicting box office revenue with Random Forest*. Kaggle. Retrieved December 10, 2021, from <https://www.kaggle.com/tavoosi/predicting-box-office-revenue-with-random-forest>.

[10] Longnickel, M. (2018, June 16). *Analyzing movie review data with Natural Language Processing*. Medium. Retrieved December 10, 2021, from https://medium.com/@MarynaL/analyzing-movie-review-data-with-natural-language-processing-7c5cba6ed922.

[11] Burukin, S. (2019, March 6). *NLP-based data preprocessing method to improve prediction model accuracy*. Medium. Retrieved December 10, 2021, from https://towardsdatascience.com/nlp-based-data-preprocessing-method-to-improve-prediction-model-accuracy-30b408a1865f.

[12] Zipin, D. (2021, December 7). *How exactly do movies make money?* Investopedia. Retrieved December 18, 2021, from <https://www.investopedia.com/articles/investing/093015/how-exactly-do-movies-make-money.asp>

[13] *Machine learning black Friday dataset*. DataCamp Community. (n.d.). Retrieved December 18, 2021, from https://www.datacamp.com/community/tutorials/ml-black-friday-dataset

[14] D. Varghese, “Comparative study on Classic Machine learning Algorithms,” Medium, 10-May-2019. [Online]. Available: https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222#:~:text=Decision%20tree%20vs%20KNN%20%3A,KNN's%20expensive%20real%20time%20execution

[15] T. Yiu, “Understanding Random Forest,” Medium, 14-Aug-2019. [Online]. Available: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

[16] Ander, R. van den. (2019, August 14). *What makes a successful film? predicting a film's revenue and user rating with Machine Learning*. Medium. Retrieved December 18, 2021, from https://towardsdatascience.com/what-makes-a-successful-film-predicting-a-films-revenue-and-user-rating-with-machine-learning-e2d1b42365e7

[17] Synced, Synced, About Synced Machine Intelligence | Technology & Industry | Information & Analysis, Synced, A., Machine Intelligence | Technology & Industry | Information & Analysis, & Name. (2017, October 24). *Tree boosting with XGBoost – why does XGBoost win "Every" machine learning competition?* Synced. Retrieved December 18, 2021, from https://syncedreview.com/2017/10/22/tree-boosting-with-xgboost-why-does-xgboost-win-every-machine-learning-competition/

[18] *Ask a data scientist: Data leakage*. insideBIGDATA. (2014, November 26). Retrieved December 19, 2021, from https://insidebigdata.com/2014/11/26/ask-data-scientist-data-leakage/