Analysis of Movie Data to Predict Total Revenue and Box Office Success

Course: CompSci 9114
Date: 6 Dec 2019
Written by: Jingyang Fan, Moustafa Shaker, and Jiaai Xue

Abstract
The growing market of the film industry and the large profits that it brings in annually necessitates the need for robust decision making frameworks to decide on the best movies to invest in. These decisions are traditionally made by high ranking individuals in large studio companies, but the use of a quantitative model can increase profits by aiding in this decision making process. Model fitting on a sample size of 4778 diverse movies using decision tree models were found to be relatively accurate based on a mean squared logarithmic error metric. A random forest and XG boosted regression model obtained values of 2.04 and 1.79 respectively. Performance was particularly good for movies whose earnings were not extreme outliers, but were closer to the average revenue earned by movies. The use of these models and others like them can be used as quantitative methods of corroborating the decision to greenlight a screenplay.

1. Introduction
Art is often evaluated on the basis of subjective criticism from consumers of said art. Evaluating and quantifying the quality and enjoyability of a piece is often a vain endeavor that can invoke the wrath of dedicated fans. It is often the case that as a following grows for a piece and it gains popularity, individuals that otherwise would never have paid it a second thought begin to notice. Vincent Van Gogh, painter of the famous "The Starry Night" died penniless, yet his works are now being sold for hundreds of millions of dollars. Who could have predicted the success of his work? Fortunately, the case is different for the modern media of cinema where information about a movie is used to advertise and entice individuals to watch. Similar to a painting where knowing the artist of a piece can immediately lend credibility and value, the director or lead actor/actress in a movie can give viewers confidence that they will get their money's worth. The U.S. film industry is currently growing at an incredibly fast rate and is expected to grow by 2% in the next 5 years (Watson, 2018). This rate outpaces the growth of the overall U.S. economy. Additionally, more and more of the film industry's total revenue is being retained as total profits, further exacerbating its growth rate (Robb, 2018). With this sort of growth and the promise of wider profit margins, it is becoming increasingly valuable to be able to forecast the performance of a movie before release. Given a set of variables about a movie's production and advertising framework, executives of large studios would want to how well a movie is projected to perform in order to drive investment decisions (Dhir and Raj, 2018) (Lee and Park, 2016).

This study aims to use regression techniques to accurately predict the total revenue earned by a movie prior to release. In particular, the use of random forests or gradient boosted trees would be the primary techniques used for prediction. The reasoning behind this decision is because complex data sets, such as the one used in this study, are often non-Gaussian and violate the linearity and equal variance assumptions. Additionally, decision trees are both simple to use and robust methods of predicting on datasets where linear regression models fail (Tsansas and Xifara 2012). The overall goal of the study is to create a predictive model that can be applied to a real up-and-coming movie in order to accurately predict its total revenue and box office success.

2. Data
The Movie Database (TMDB) has an open API that allows users to freely search and query information about movies. Github user "alicexja" queried the data from 4778 different movies, which was used as the data set in this study (Kaggle 2019). The data set comes from a very diverse set of movies. The release date of the movies queried range from as early as 1916 to as late as 2017. Additionally, there are a total of 20 different genres that a movie can be categorized in, and each can

belong to multiple groupings. The data set includes a total of 22 variables, in which the response variable is the total revenue earned by a movie in dollars.

There are 3 types of predictive variables. The first are variables that contain a dictionary of values. These include: cast, crew, genres, keywords, production companies, production countries, and spoken languages. These variables can simply be treated as lists since a single movie can have multiple values. It is notable that the values are not unique, and multiple movies can have the same values. For example, movies can share genres and actors/actresses. The second type of variables are strings. These include: title, homepage, original title, overview, status, original language and tagline. Most of these string variables are pseudo-unique, like title and original title where most titles are unique, but sometimes there are movies that happen to have the exact same name. On the other hand, there are other string variables where most of the variables are the same. These would include the original language and status of a movie. Finally, there are string variables that are completely unique, and these would include homepage, overview, and tagline. The third type of variables are numerical ones. These include: movie id, budget, popularity, release date, revenue, runtime, vote average, and vote count. Movie id is a unique code for each movie and thus does not have any predictive power on total revenue. The release date variable is in YYYY/MM/DD format and is the date that the movie was released. Runtime is a float in units of minutes and represents how long a movie is. A full list of each variable and its corresponding description can be viewed in table 1.

There are several variables that cannot be used as predictors of total revenue due to them being irrelevant or not being available prior to a movie's release. For example, the website homepage of a movie should not be used since it has overlapping information with the title variable. On the other hand, the popularity of a movie would be an extremely valuable predictor, but unfortunately it would not be available prior to a movie's release. Some variables are also problematic and difficult to work with. How can the tagline of a movie be used as a predictor when each tagline is unique? Some processing needs to be done on these types of variables to extract information out of them. This will be explored in the methods section.

3. Methods

This section will discuss the data science techniques used to predict the revenue of a movie given a set of features. How and why these techniques were applied will also be discussed.

3.1 Data cleaning

"Garbage in, garbage out" is a well-known concept in Data Science that elaborates on the importance of having high quality input data to reach accurate conclusions. The data set came in separate files in comma separated and JSON format. The data in JSON format was changed to comma separated format for convenience and consistency Additionally, turning some of the features into dummy variables is a straightforward process when the data is in comma separated format.

Non-values in each feature were checked and either replaced by the mean or median based on the data trend. In some cases, the mean and median were not consistent with the general data trends in a scatter plot, so the data record was deleted. Furthermore, nonsensical values such as $0 for budget and $0 for revenue were either replaced by median, mean or their record was deleted.

3.2 Data exploration

Before diving into the implementation of advanced data science techniques, exploratory data analysis was conducted to better understand the statistical properties of the variables in the dataset. Histograms were plotted using the Python Seaborn visualization library to obtain non-parametric density

estimates for each variable. Features such as popularity, revenue, and budget were shown to be non-Gaussian in nature. Multiple linear regression and random forest regression models make no assumptions with regard to the distribution of the independent variables. Therefore, transformations were not necessary for this type of regression.

A scatter plot for each input variable and revenue was plotted to understand whether their relationship was linear, non-linear, or independent. The scatter plots used normalized data to enable meaningful comparison between the variables. Furthermore, a heat map was plotted to explore the correlation between explanatory variables with each other and the revenue variable. The variables budget, popularity, and the total ratings of a movie were shown to have the strongest positive correlation with revenue. On the other hand, original title word count, title word count, and overview word count showed the weakest positive correlation with revenue. Among the explanatory variables, crew count and popularity had the highest correlation with each other.

3.3 Feature engineering

The original features of the data set included variables such as title, overview, original title, crew names, and etc. that were written words. Since our analysis was not focused on word processing, the format of those features were changed. This was to enable the extraction of insights from the features to make better predictions on revenue without the complications of word processing. Features such as word count of title, overview word count, tagline word count, and etc. were constructed from the features that were written words.

Since time series analysis was not the focus of the study, features that were in date formats were dropped and new features were constructed using the information in the dates. For example, the release day of the week variable was constructed from the release date. Table 2 in the appendix describes all the features that were engineered.

3.4 Defining performance matrix

Mean squared logarithmic error (MSLE) was used as the criterion to measure performance. The MSLE was used because it incorporates both the variance of the estimator and its bias. The natural logarithm was used to allow for better comparability since the revenue of movies can output extremely large values. For each model, the MSLE was calculated by using the built-in functions in Python.

3.5 Training/Testing Split and Cross Validation

Data was split into a training and a testing set. The training set was used to train the model to find the best parameters. In this project, k-fold cross validation was used with the parameter k set to 5. Cross validation further divided the training data into validation and training sets. This technique was implemented to reduce bias and find a model that generalized better than a model which was fit using only the training data. The test dataset was used to provide an unbiased evaluation of the final model. It provided the final estimate on how the model would generalize on new data. The splitting of the data and the cross validation was done using the scikit-learn machine learning library in Python.

3.6 Multiple Linear Regression for Base-line Performance

A multiple linear regression model was fit using the original and engineered explanatory variables to predict revenue. Highly correlated explanatory variables were not dropped because the goal was prediction and not inference. Using the functions in the scikit-learn machine learning library, the training data set was used in a cross-validation function to estimate the best parameters of the model. Following that, the model performance was evaluated using the test data set to get an unbiased estimate

of the model performance on new data. The unbiased mean square error estimate of the multiple linear regression was used as a baseline performance for the other regression models in this project.

3.7 Random Forest Regression

A random forest regression was fit using 2000 trees in the forest and 2 minimum samples to split an internal node. Furthermore, the random state variable in the scikit-learn function was set to 1 to control for the randomness of the bootstrapping of the samples used when building trees and sampling of features to consider when looking for the best split at each node. This control was done to enable the replication of the results if necessary. After fitting the random forest regression model, cross validation was used to estimate the out of sample MSLE.

The GridSearch function in scikit-learn function was used to perform an evaluation of possible values for the minimum number of samples required to split an internal node(min_samples_split) and also the minimum number of samples required to be at a leaf node(min_samples_leaf). The evaluation of possible values for the parameters "min_samples_split" and "min_samples_leaf" was done to optimize the random forest regression model. The GridSearch function evaluated 2, 5, and 10 as possible values for the parameter "min_samples_split" and evaluated 0.001, 0.005, 0.01 as possible values for the parameter "min_samples_leaf".

A new random forest regression model was fit using the optimal parameters found by the GridSearch function and 5000 trees in the forest. The performance of the new model was evaluated using 5-fold cross validation to estimate the out of sample mean square error.

3.8 XGBoosting

XGBoosting was used to fit a gradient boosting regressor. The gradient boosting regressor function from scikit-learn library was used. In the initial fit the contribution of each tree was set to shrink by 0.1, the minimum number of samples required to split an internal node was set to 2, and the maximum depth of individual regression estimator was set to 3. Using a 5-fold cross validation the estimate the out of sample mean square error was calculated for this model.

Using GridSearch function in scikit-learn function to perform an evaluation of possible values for the learning rate to shrink the contribution of each tree(learning_rate), the minimum number of samples required to split an internal node(min_samples_split), and the maximum depth of the individual regression estimator(max_depth). The evaluation of possible values for the parameters "learning rate", "min_samples_split", and "min_samples_leaf" was done to optimize the XGBoosting. The GridSearch function evaluated 3, 5, 10 and 15 as possible values for "max_depth", 2, 5 and 10 as possible values for "min_samples_split", and finally 0.05 and 0.1 as possible values for "learning_rate".

A new gradient boosting regressor model was fit using the optimal parameters found by the GridSearch function and 5000 trees in the forest. The performance of the new model was evaluated using 5-fold cross validation to estimate the out of sample mean square error.

4. Results

The baseline performance for our project was the multiple linear regression model. The model's performance on the test data outputted a MSLE value of 2.23. Comparatively, the random forest regression and the XGBoosting regression models outputted MSLE values of 2.79 and 2.34 respectively. This is without any parameter tuning for the random forest regression and the XGBoosting regression. Following parameter tuning, the random forest regression and the XGBoosting regression out of sample mean square error improved to be 2.15 for the random forest regression and 2.25 for the XGBoosting regression (Table 3).

Using a 5-fold cross validation, our baseline multiple linear regression MSLE was 2.06, while the random forest regression and XGBoosting regressions had MSLEs of 2.61 and 2.01 respectively. This is before tuning the parameters of the random forest regression and the XGBoosting regression. Following the parameter tuning the mean square error of both the random forest regression and the XGBoosting regression improved. The random forest regression had mean square error of 2.04 while the XGBoosting regression had 1.79 (Table 3).

5. Discussion

From the three regression models that were created, the XG Boosted model obtained the greatest performance, followed by the random forest model, and then the multiple linear regression model. This finding aligned with the expectations of how decision tree based models would perform on the dataset compared to a linear model. The most surprising finding was that the multiple linear regression model actually had really good performance. It ran much faster than the decision tree models, and even performed better than them before their parameters to tuned and optimized. This seems to show that a linear model is actually incredible useful if a model needs to be trained quickly, whereas a decision tree model (particularly an XG Boosted model) would give better performance but require more time and effort.

Based on the models created, it was found that the strongest predictor of the total revenue of a movie was the budget allocated to it. This intuitively makes sense since movies with higher budgets would be able to afford better talent and create a larger scale movie that would appeal to a larger audience. However, there may be another reason that the budget of a movie was found to be such a strong predictor. Large budgets are given to movies that are already likely to succeed. This is because production companies have their own internal-based rating systems and decision hierarchies to pick and choose projects that have elements that would allow them to do well. These elements might include a really compelling and original plot, or an incredible director picked it up, or maybe it is a new installment to a franchise that historically does incredibly well (e.g. Marvel Cinematic Universe). These are the sorts of movies that production companies will invest in and give a large budget to. In this way, the use of the budget variable is actually a reflection of the decisions that production companies have already made.

Another interesting finding about the performance of the models was that they had a lot of trouble accurately predicting the revenue of movies that were wildly successful. A simple explanation of this phenomenon would be because these incredibly successful movies are outliers and do not fit into the general trend. Most movies that come out are not nearly as successful as the biggest movies in this dataset, even if they end up being profitable investments. Additionally, as stated in the beginning of this paper, the popularity or fan base of a franchise can massively increase visibility and generate a greater audience. Therefore, a movie that is doing well will garner more attention and do even better. This is further exacerbated in the modern climate due to the increased rate of information transfer through social media. The spread of internet memes and news coverage of the hottest new movie allows it to gain even more popularity, and thus increase its revenue. Furthermore, since the regression models were based on a dataset containing a wide variety of movies, regardless of success or popularity, they have trouble predicting on movies that are outliers of these qualities.

There were some limitations to our study that limits the power of the models created. One of these limitations is the data available to work with. Although the data set used was fairly thorough, there was still some missing information that would be incredibly useful for prediction. For example, there was very little information on the advertising campaign of the movie and how much it increased the movie's visibility prior to release. The only piece of data relating to this was the tagline variable, which

was extremely limiting. Additionally, the advertising budget was included in the total budget, but there was no information on exactly how much was used for advertising. All movies have some sort of trailer they use to grab the attention of audiences. However, there's no sort of quantitative measure that can be used to evaluate how "good" a trailer was and how well it did to "hype" up audiences. One of the limitations behind the methods used for analysis was how strings were treated. Particularly how the movie titles, overviews, and taglines were treated. The information behind the meaning of these strings were lost through the method used to only measure the word counts. This was done in an attempt to simplify the analysis, but resulted in the sacrifice of a lot of information.

Moving forward, the limitations faced in this study could be improved upon to enhance the models created. Collection of more data about each movie would provide more information that could be used for prediction, particularly variables pertaining to how a movie advertised or garnered attention from viewers. Additionally, since many successful movies build upon the success of previous movies, a variable that indicates a movie's connection to current franchises could be a powerful predictor since that movie will already have the advantage of having an established audience. Finally, language processing techniques could be used on long string variables to extract more information out of the meaning of the words and sentences. With these changes, more accurate models can be created and be used to aid decision making in the filmmaking industry.

References

Dhir R., Raj A., Dec 2018. Movie success prediction using machine learning algorithms and their comparison. *2018 ICSCCC*

Lee K., Park J., Aug 2016. Predicting movie success with machine learning techniques: ways to improve accuracy. *Information Systems Frontier* **20**(3): 577–588

Robb D. Jul 2018. U.S. Film Industry Topped $43 Billion In Revenue Last Year, Study Finds, But It's Not All Good News. *Deadline*: https://deadline.com/2018/07/film-industry-revenue-2017-ibisworld-report-gloomy-box-office-1202425692/

Tsanas A, Xifara A. Jun 2012. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings* **49**: 560–567

Watson A. Dec 2018. Film Industry - Statistics & Facts.
Statistica: https://www.statista.com/topics/964/film/

May 2019. TMDB Box Office Prediction. *Kaggle*: https://www.kaggle.com/c/tmdb-box-office-prediction/overview

Appendix Attached Separately