Predicting Box Office Success

Yang Li, Rita Zou, Rus Refati, Fei Xu, Xinyu Cao

Abstract – **The primary goal of this project is to correctly predict movie performance classes (blockbuster, successful, moderate success, underperforming, flop), defined by gross box office sales. Three different algorithms are employed: Support Vector Machine, Boosted Tree, and Artificial Neural Network. Boosted tree has best performance for Train data.**

I. INTRODUCTION

Investing in companies from the film industry on the stock market is harder than investing other companies in traditional industries because these film-related companies ‘stocks are strongly affected by the financial performance of their upcoming movies in the short term. Hence, accurate and reliable prediction of the Box Office Revenue (BOR)of a movie before releasing is an in pressing need, as it greatly informs the tough investment decision-making process when facing such a high-risk and high-yield opportunity.

Given our dataset of the Top 1000 IMDB-rated Movies, we attempt to predict the following classes: flop, underperforming, moderately successful, successful, blockbuster. These classes are grouped by ranges in the Gross variable. The Gross value for each class is, respectively: less than 5 million, 5 - 25 million, 25 - 100 million, 100 - 300 million, greater than 300 million. The data has been partitioned into train/test by way of random sampling. A split of 80/20 is employed.

II. DATA

The data being used in this project comes from the IMDB API. The data contains almost 1000 rows and 16 columns of movie attributes (Poster link, Series title, Released Year, Certificate, Runtime, Genre, IMDB Rating, Meta score, Director, Star1, Star2, Star3, Star4, Number of votes, Gross).

We took the following steps to clean the data: text heavy columns “Poster Link” was removed because this project is outside the scope of text-mining.

1. Rows having the “Gross” column as an NA were removed, because this is the response for our algorithms.
2. Gross column is coded to be integer format.
3. Log transformed the Gross column since it was highly skewed to the right.
4. Runtime is coded to be integer format from text format
5. Most movies have multiple genres, so the “Genre” column was split into different columns representing each unique level, with a “1” encoded for it being present, and a “0” otherwise.

Some movies (11%) were only classified into one genre while other movies were classified into multiple genres (22% had two genres; 67% had three genres). Note that this graph does not include self-connections, meaning that it does not show movies with only one genre. Drama was by far the most popular category, followed by Adventure, Comedy and Action. This visualization makes it apparent that the most commonly occurring pairs were Drama-Crime, Drama-Biography, and Action-Adventure. The data contained two potentially useful columns - "Actors" and "Directors", populated with names of corresponding persons. In order to use these columns in our algorithms, two new binary columns "top50actor" and "top50director" were created. Given that at least one Top 50 actor/actress was in the Movie, a "1" would return; given that the movie was produced by a Top 50 director, the same was true for the other binary column.

The variable we are wanting to predict is the gross, but this variable is highly skewed towards larger values. We decided to take the log of this predictor, so it has a more normal distribution.

Given that "Runtime" and "Gross Amount" are continuous variables, the discretization of each would aid in model fitting / creation. "Runtime" has been discretized into four intervals, splitting 90-150 into two separate groupings as the majority of data falls within this range. Runtime column ranged from 72min to 238 min and is split into a new column "Runtime Lengths" with 4 levels: length less than 90 min, 90 min to 120 min, 120 min to 150 min, and more than 150 min.

Our response variable "Box_Office_Performance" is defined by discretizing Gross. The "Gross" column is discretized by 4 thresholds, and they are 5 million, 25 million, 100 million, 300 million, and we recode these 5 intervals into 'flop', 'underperforming', 'mod success', 'successful', 'blockbuster'.

Because this new column is now the response, it is important to verify that our defined levels are distinct. The statistical significance of the difference between levels was verified via Tukey's HSD connecting letters report.

III. TEAM RESPONSIBILITIES

Rus and Rita will be responsible for importing and cleaning the data (discretizing columns, getting rid of unwanted columns, etc.).

Xinyu will be responsible for coding, running, validating, and gathering insights for the SVM model.

Fei will be responsible for coding, running, validating, and gathering insights for the Boosted Tree model

Yang will be responsible for coding, running, validating, and gathering insights for the ANN model.

All members will work on data visualization, the final report, and the final presentation.

IV. TIMELINE OF MILESTONES

Below shows a general timeline for the project:

Week 1 (Oct. 10): Rus will import the data and the rest of the team will familiarize themselves with the data and add additional cleaning if necessary.

Week 2-3 (Oct. 24): Rita will do the data cleaning and pre-processing, and the rest of the team will do the visualization and clustering.

Week 4-5 (Nov. 7): Every member will work on testing their models on the holdout set, gathering insights, and possibly tuning the models.

Week 6-7 (Nov. 21): All members will evaluate the models and keep tuning the models and then review the insights.

Week 8 (Nov. 28): All members will work on and finish the final presentation and final report.

V. MODEL BUILDING AND RESULT

A. Support Vector Machine (SVM):

Support-vector machine classifiers are one of the popular algorithms in the machine learning field, which are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. In this project we choose train() and "method='svmLinear'"& "method='svmRadial'"in the package "caret" for the actual classifications. In this model, we use Box_Office_Performance as response variable, Released_Year and Runtime as numerical predicting variables, also genre, top50actor and top50director as binary predicting variables. For the parameter, we set the cross validation as 10. The result shows accuracy was used to select the optimal model using the largest value and the tuning parameter 'C' was held constant at a value of 1. When choosing the value of C, it is important to note that large values of C result in lower bias and higher variance, and therefore become overfit as C gets larger. Our model with a C parameter of 1 gave us an accuracy on the training set of 43.3%. While when we used the model to predict we finally got an accuracy of testing set at 41.3%. Radial kernel's tuning parameter 'sigma' was held constant at a value of 0.002606789. The final values used for the model were sigma = 0.002606789 and C = 1.

However, it gives a worse performance, a percentage about 8.4%, insinuating that "svmRadial" is not applicable to this dataset.

B. Boosted Tree:

Boosted trees incrementally build an ensemble by training each new instance to emphasize the training instances previously mis-modeled. And for this project we use xgboost since it is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solves many data science problems in a fast and accurate way. For the implementation, we follow and use the R-code in lecture notes. In this model, we use Box_Office_Performance as response variable, Released_Year and Runtime as numerical predicting variables, also genre, top50actor and top50director as binary predicting variables. For the parameter, we set the cross validation as 10. The result shows accuracy was used to select the optimal model using the largest value and tuning parameter 'gamma' was held constant at a value of 0 parameter 'min_child_weight' was held constant at a value of 1. The best tune used for the model were nrounds = 50, max_depth = 1, eta = 0.3, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample = 0.75 giving us an accuracy of training set at 44.3%. While when we used the model to predict we finally got an accuracy of testing set at 41.9%.

C. *Artificial neural network*

ANN Classification is an example of Supervised Learning. Known class labels help indicate whether the system is performing correctly or not, and it is the process of learning to separate samples into different classes by finding common features between samples of known classes. For the implementation, we follow and use the R-code in lecture notes. In this model, we use Box_Office_Performance as response variable, Released_Year and Runtime as numerical predicting variables, also genre, top50actor and top50director as binary predicting variables. For the parameter, we set the cross validation as 10. The result shows accuracy was used to select the optimal model using the largest value and the final values of best tune used for the model were size = 5 and decay = 0.1 giving us an accuracy of training set at 44.1%. While when we used the model to predict we finally got an accuracy of testing set at 33.6%. Also, we noticed that the accuracy of the testing set decreased substantially from the accuracy of the training set which means there might be an overfitting issue. It may improve the performance if we try to evaluate the classification error during the process of re-adjusting weights or scale the data. Another thought is we could also try to use different predictors in the model by using importance of variables or omitting one or more variables to see if there can be an improvement in models’ performance.

VI. EXPECTED OUTCOME

The expected deliverable is a functioning model that uses the key info of a movie to predict its revenue or success with acceptable accuracy and evaluation metrics (MSE or F1 score). We will also provide recommendations on what factors are more important in movie production. Based on the learned features, we train a mutually enhanced prediction and ranking model to obtain the box office revenue prediction results. Finally, we apply the framework to the film market and conduct a comprehensive performance evaluation using real-world data. We will use experimental results to demonstrate the superior performance of both extracted knowledge and the prediction results.

VII. INSIGHTS AND CHALLENGES

While the dataset used in this report was extensive, it is reasonable to acknowledge a few shortcomings. The data fails to include the movie's budget, marketing spend, or production studio, all of which could more accurately contribute to a measure of a movie's success. Big-budget movies may be correlated with blockbusters, whereas smaller budgets would be more indicative of class B movies. The production studio may be significant, as the studio determines the distribution of a movie. The marketing spend may have a big impact in that the more marketing a movie receives, the greater the box office is expected to be. The inclusion of these metrics in the data would provide a more accurate depiction of success.

The dataset does include release year but fails to provide release month. Given the seasonality of movie releases, this could be an important variable that was omitted. It is a common assumption that blockbuster movies are released around holidays (Fourth of July, Thanksgiving, Christmas) as families are together and searching for something to do. The inclusion of months in our dataset might bolster the accuracy of some algorithms.

In real life, the prediction of a high grossing movie can be somewhat easy at times. Given that the movie has a notable star, a high budget, summer release date, and a general public excitement already established (think Harry Potter, Star Wars), it would be a safe assumption to guess that it would be a success. This is especially true in the case of movie sequels.

REFERENCES

[1] https://www.ranker.com/crowdranked-list/the-most-oscar-worthy-directors-of-all-time

[2] https://www.ranker.com/crowdranked-list/the-greatest-film-actors-and-actresses-of-all-time?ref=collections_page