Predicting movie revenue from movie meta data

Author: Pinzhi Huang, New York University, New York, NY, 10012, USA.
Email: ph2239@nyu.edu

ABSTRACT

This study employs machine learning techniques to forecast box office movie revenues, utilizing metadata derived from movies. The utilized dataset is derived from IMDb and has been processed based on specific criteria to fit the model requirements. The research framework encompasses four distinct components: an introduction to the research topic, data processing methodologies, model construction, and finally, model application and results. The constructed prediction model is trained on metadata from past movies and applied to predict revenues for newly released movies. The findings indicate that the model's performance significantly improved with the addition of more features. Additionally, the Catboost algorithm outperformed all other algorithms in terms of prediction accuracy. This research underscores the notion that increasing the number of features can augment the predictive accuracy of movie revenue models.

KEYWORDS: *box office revenue, machine learning, meta data, gradient boosting*

1. Introduction

Movies, as an artistic expression, are instrumental in the cultural exchange between countries worldwide. Over time, movies have evolved into profitable commodities, largely due to the pervasive film culture in modern society. However, despite the film industry's capital-intensive nature, the path to success or profitability often remains uncertain, thus making the prediction of movie revenue a compelling research topic.

There has been an uptick in the adoption of machine learning technology to predict a movie's future performance. Prior research has demonstrated that the Pruned Random Forest model could reliably predict a movie's future revenue in China [1]. This model outperforms others in accuracy and offers a useful revenue range prediction, aiding film companies in decision-making. Studies have also found that employing multiple models instead of a single model enhances prediction accuracy [2]. Research by Nikhil et al. highlights that factors like the number of theaters at release, budget, and first-weekend revenue significantly impact box office revenue predictions [3]. Their method combines k-means clustering and linear regression, suggesting that, in some instances, it's possible to predict total revenue with better than 20% accuracy, although some cases are difficult due to database sample limitations.

In another study, a combination of Support Vector Machine (SVM), Neural Networks, and Natural Language Processing was employed [4]. Their technique exhibited varied accuracy for pre-released features, but their findings underlined that the number of screens, Internet Movie Database (IMDb) voters, and budget amounts are critical for predicting a movie's box office revenue. Further, Factorization Machines have been found capable of predicting a movie's future success using rating data from IMDb and social media data for recently released movies [5].

This research utilizes the TMDB Box Office database and LightGBM and Catboost machine learning models to predict movie revenue. The primary focus lies in extracting data from previously released movies to train the machine learning model for future evaluations. This study uses both budget and popularity as features within the machine learning model. Consequently, it aims to deliver a model capable of leveraging these features to predict the total revenue for a pre-release movie with high precision.

2. Data

This research amassed data from approximately 3001 movies, encompassing aspects such as rating, budget, popularity, release date, runtime, and revenue. The dataset was harvested from IMDb (www.themoviedb.org) using their API. The objective of this research was to employ these features to predict the total worldwide box office revenue for the movies.

To enhance the comprehensiveness of our database, we incorporated an additional dataset comprising 2888 movies from The Movie Database. This supplementary dataset furnished us with extra features, such as post-release popularity, total votes, and ratings. We then merged these two datasets to provide a robust training set for our machine learning models.

Table 1.Features

Note: This table shows the features in this study used from IMDb database. The research uses these features both to generate graphs for analyzing their relationships and to train models for the machine learning and regression algorithm.

Feature	Description
IMDb_id	ID of a movie in IMDb database
Total votes	Total number of votes for the rating on one movie.
Rating	A measurement voted by users of how good or popular a movie is.
Budget	Budget of a movie in dollars. 0 values means unknown.
Popularity	Popularity of the movie in floating numbers.
Popularity_2	Popularity measurement of the movies after they released.
Release date	Release date of a movie in mm/dd/yy format.
Runtime	Total runtime of a movie in minutes.
Revenue	Total revenue earned by a movie in dollars.

Table 2.Summary Statistics

Note: This table illustrates summary statistics for the features used in this analysis.

Feature	Average	Minimum	Maximum	Median	Standard deviation
Total votes	993.94	1	18931	18	1744.189
Rating	6.36	1	9	6.2	1.143
Budget	22531334.1	0	380000000	8000000	37026086.4
Popularity	8.46	0.0001	294.34	7.374	12.104
Popularity_2	8.029	0.6	45.15	2.349	5.211

Runtime	107.86	0	338	105	21.887

2.1Criteria

Among the initial 3,000 films surveyed, this study excluded those with a production budget below $1,000, leaving 2,800 films for further analysis. The preprocessing phase necessitated the removal of 11 films from the dataset due to a recorded runtime value of zero. Consequently, following the data cleansing procedure, a total of 2,158 films were retained for the training process.

2.2Release date

As the years progress, there is a notable increase in the number of film releases. Over nearly a century (1920-2016), the total box-office revenue from movies has generally trended upward, albeit with periods of fluctuation. It's important to highlight that there was a dramatic surge in total revenue between 1972 and 1976, reaching a pinnacle in 1974. Additionally, a discernible seasonality effect is evident, with film revenue typically peaking in the months of June and December, while noticeably lower in other months.

2.3Rating count

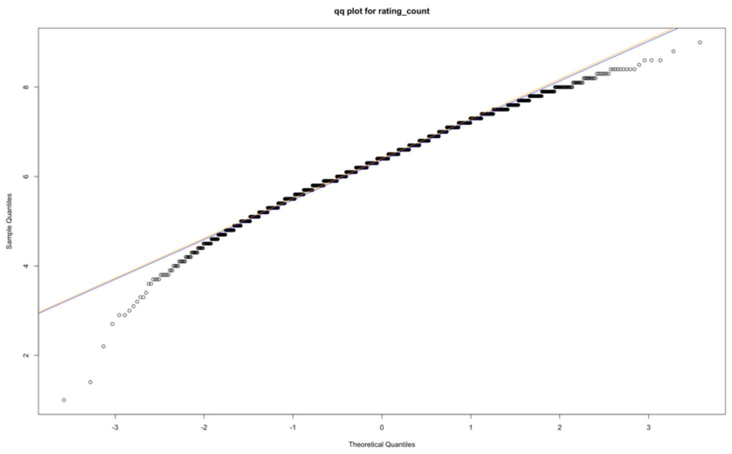

Figure1.Q-Q Plots for rating count

Note: This figure presents a Quantile-Quantile (Q-Q) plot for the rating count feature. A Q-Q plot serves as a graphical tool used to compare two datasets, aiming to elucidate their relationship by assessing their similarity or differences in distribution.

Q-Q plots facilitate a comparison between the quantiles of two distributions. As illustrated in Figure 1, this study displays quantiles derived from a theoretical normal distribution along the horizontal axis. These are juxtaposed with the distribution of actual sample data—rating counts—plotted on the vertical axis. The closely fitting line through the data points suggests that the rating count closely adheres to the expected pattern of a normal distribution. This distribution is symmetric, with a majority of data concentrated around the mean and median, while exhibiting fewer data points on either tail.

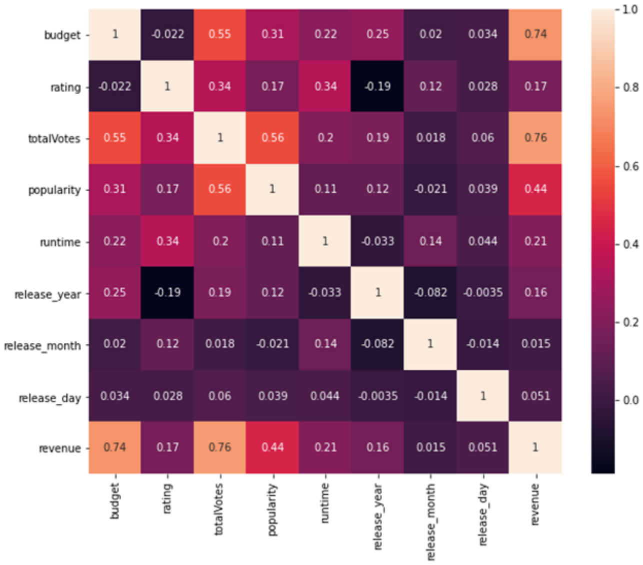

Figure2. Correlation of Features

Note: This is a heat map for the correlation of features used in this analysis. The bar on the right with different colors represents different degrees of the correlation.

The heatmap depicted above showcases the extent of correlation among various factors, signified by coefficients ranging from 0.0 to 1.0. Movie revenue demonstrates a strong correlation with the budget, indicated by a coefficient of 0.74. This suggests that films with higher investment levels tend to yield higher box-office returns. Similarly, movie revenue exhibits a robust association with total votes, reflected by a correlation coefficient of 0.76. This indicates that revenue is impacted by audience sentiment; films that receive higher ratings are more likely to generate larger profits. The metric of popularity also emerges as a significant predictor of high movie revenue, boasting a correlation coefficient of 0.44.

3. Methodology

The Proposed methodology is illustrated in Figure 3. It contains following steps:

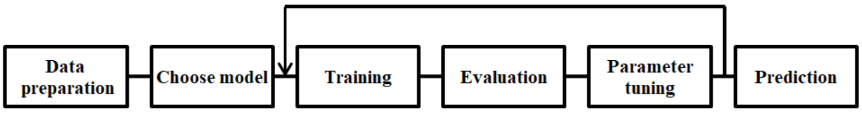

Figure3.methodology

Note: This figure presents a flowchart detailing the methodology utilized in this study. The flowchart portrays the research steps in the form of interconnected boxes, with arrows indicating the order of operation. It comprises six steps in total, outlining the research approach to problem-solving.

3.1Data preparation

The dataset, provided by IMDb and sourced from www.themoviedb.org (TMDB) Open API, forms the basis of this research. Upon importing the data, the study undertakes an analysis of the dataset, followed by data cleaning. The dataset contains irrelevant features and missing values, which were addressed by focusing on features exhibiting strong correlation with revenue. To handle missing values, the IMDb dataset was merged with the TMDB data, allowing for the completion of incomplete fields and the filtering out of incorrect information.

3.2Model Selection

The objective of the research is to train three models — Gradient Tree Boosting (GBT), Light GBM, and Catboost -- on the same training set to compare performance. The selected models are further elaborated upon below.

GBDT

The Gradient Boosting Decision Trees (GBDT) model generalizes boosting to arbitrary differentiable loss functions. GBDT is a high-performing, versatile tool applicable to both regression and classification problems across various fields, particularly in number predictions or ranking tasks. Figure 4 illustrates the GBDT algorithm's training process. During the learning and weighting phase, residuals are calculated by comparing the difference between the actual data (from the previous decision tree) and the predicted data (from the current decision tree). The next decision tree is then built based on these residual values. This procedure continues iteratively: each classifier trains on the residuals from the last iteration to reduce errors. The final output combines all constructed trees to form a more accurate ensemble model.

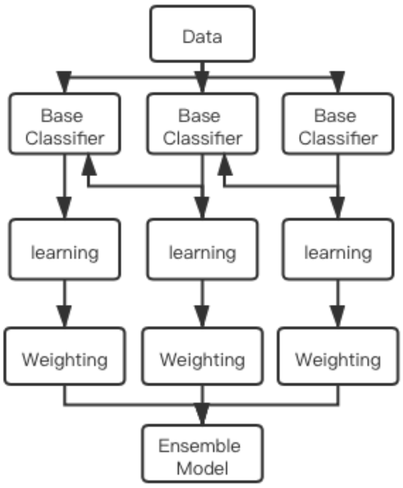

Figure4. GBDT Algorithm Training Process

Note: Figure4 shows a flowchart for Gradient boosting Algorithm, it contains base classifier, learning and weighting process, the goal is to produce a prediction model and give a prediction result.

LightGBM

Another machine learning algorithm applied in this research for forecasting movie box-office revenue is LightGBM (Light Gradient Boosted Machine), a highly efficient gradient boosting decision tree based on a histogram algorithm. The creators addressed the issue of high computational complexity, a problem that often arises due to the sheer volume of features and instances when predicting with GBDT. They developed LightGBM, a novel GBDT algorithm that employs Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to handle large numbers of data instances.

LightGBM facilitates parallel learning across both features and data. The parallel algorithm aids in forming the decision tree by determining the optimal data split, thereby greatly enhancing the efficiency of the training process. Unlike traditional decision tree-based algorithms, which typically adhere to a level-wise tree growth principle, LightGBM employs a leaf-wise tree growth strategy. The algorithm selects the leaf with the maximum delta loss and continues to develop on that specific leaf node.

Experimental results clearly demonstrate that LightGBM offers a superior trade-off in terms of training efficiency and memory usage compared to the GBDT algorithm. LightGBM is a gradient boosting algorithm that employs tree-based learning strategies, and it's designed to be distributed and efficient, offering the following advantages:

- Superior performance during the training process.
- Lower memory space requirements.
- Greater accuracy compared to similar models.
- Compatibility with GPU learning for accelerated performance.
- Capability to process metadata concurrently.

Catboost

Catboost algorithm is a novel machine learning algorithm developed by Yandex, a famous Russian searching engine. Research applied ordered boosting with ordered TS (target statistics) to solve the problem of prediction shift due to the defective currently existing gradient boosting algorithms [8].By comparison, the ordered boosting mode of Catboost performs better than XGBoost and LightGBM, especially in the accuracy performance. CatBoost combines the categorical features and gradient boosting, also based on the decision tree library. It is a powerful machine-learning technique that achieves state-of-the-art results in a variety of practical tasks. Catboost has algorithm for transforming the categorical data into numerical data and takes advantage of dealing with them during training as opposed to preprocessing time. Another advantage of the algorithm is that it uses a new schema for calculating leaf values when selecting the tree structure, which helps to reduce the overfitting. Steps for the Catboost training process are to get the residuals from each data point and apply the trained model on all remaining data. The research is to train the model with the residuals of each data point, these steps are repeated until the algorithm converges.

4. Results

4.1Training and parameter tuning

The research split the dataset into 80% train and 20% test. After training the model and evaluate the result to adjust the model parameters through an exhaustive parametric grid search. Grid search cross-validation is used to train a machine learning models for each combination of model parameters to determine the optimal combination of parameters which yields the highest model performance. Model performance is measured through the R^2 regression score as the performance metrics in this research to help us to choose the best model of all time.

4.2Prediction

Using the holdout data set (20%) The research is to make predictions for all three models and compare r-squared (R^2) score and other performance measures from each model (Table 3).

Table 3.Performances for each model

Note: This form draw a comparison between three models, and shows the R^2 score as well as running time they have on making predictions. The training and tuning time is in seconds (s).

	GBDT	LightGBM	Catboost
R2 score(without additional feature)	0.64	0.53	0.64
R^2 score	0.76	0.73	0.78
Training time	34 secs	15 secs	8 secs
Parameter Tuning Time(for 27 fits,500 iteration）	315secs	168secs	238secs

Catboost delivered the most impressive performance. None of the models performed well without the inclusion of additional features such as rating, total votes, and popularity. This underlines the significance of feature quantity in enhancing the performance of machine learning models in predictive tasks.

It is evident that LightGBM made the most notable improvements once these additional features were incorporated. This suggests that LightGBM leverages less important features more effectively than the other models. The study indicates that CatBoost provides the highest accuracy on the test set and requires the least training time.

GBDT typically performs well and requires fewer parameter adjustments compared to the other models, making it the simplest model to set up. LightGBM, while offering the shortest parameter training time, achieves a marginally lower accuracy score compared to the others.

5. Conclusion

This research successfully developed machine learning models to facilitate movie revenue prediction. By comparing the performance of these models, it was determined that Catboost performed optimally for this particular dataset. However, it's important to note that these results may vary if the dataset is expanded to include more features or a larger number of movies for analysis.

Future research could consider identifying additional factors that might influence revenue. We are confident that, with a more extensive dataset, this research could enhance both the reliability and accuracy of the predictive model.

References

[1]. Z. Guo, X. Zhang, and Y. Hou, “Predicting Box Office Receipts of Movies with Pruned Random Forest,” in *Neural Information Processing*, Cham, 2015, pp. 55–62.

[2]. G. He and S. Lee, “Multi-model or Single Model? A Study of Movie Box-Office Revenue Prediction,” in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Oct. 2015, pp.

321–325.

[3]. Nikhil Apte, Mats Forssell, Anahita Sidhwa “Predicting Movie Revenue“http://cs229.stanford.edu/proj2011/ApteForssellSidhwa-PredictingMovieRevenue.pdf

[4]. N. Quader, Md. O. Gani, D. Chaki, and Md. H. Ali, “A machine learning approach to predict movie box-office success,” in *2017 20th International Conference of Computer and Information Technology (ICCIT)*, Dec. 2017, pp. 1–7.

[5]. B. Çizmeci and Ş. G. Ögüdücü, "Predicting IMDb Ratings of Pre-release Movies with Factorization Machines Using Social Media," *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, Sarajevo, 2018, pp. 173-178.

[6]. J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” The Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.

[7]. G. Ke et al., “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” p. 9.

[8]. L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “CatBoost: unbiased boosting with categorical features,” p. 11.