2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T) | 978-1-6654-5858-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICPC2T53885.2022.9776798

Movie Box-Office Success Prediction using Machine Learning

Gaurang Velingkar*, Rakshita Varadarajan†, Sidharth Lanka‡ and Anand Kumar M§
Department of Information Technology, National Institute of Technology Karnataka
Surathkal, India 575025
Email: *gaurang.191it113@nitk.edu.in, †rakshitajps.191it140@nitk.edu.in,
‡sidharthlanka.191it251@nitk.edu.in, §m_anandkumar@nitk.edu.in

Abstract—**Being a multi-billion dollar business, the film industry contributes largely to helping sustain a country's economy. A movie's box office (the revenue generated by the number of tickets sold of a movie) is an essential indicator of the movie's popularity. It varies depending upon several factors, including a production company, genre, budget, reviews, ratings, etc. Predicting an approximate value for a movie's box office based upon the various parameters helps investors with this business make intelligent and informed decisions. Thus, this paper designs a machine learning model that can predict the revenue a film will generate based on the information available before the movie's release. It also provides a model capable of taking in the planned genre, the required revenue, and using the Random Forest Regression model, provides recommended budget, runtime, star power, and expected popularity.**

Keywords – movies, box-office, machine learning, revenue prediction, regression.

I. Introduction

Going out to watch movies at the theatres is a popular leisure activity across the globe. The film industry in India generated a box office revenue of Rs 10,948 crore in 2019 [13]. Given the attractive revenue generated by the film industry, it is naturally a massive sector with a significant potential for investments. However, the risky nature of the industry with different revenues generated by different movies makes it complex, with high chances of losing all the investment.

Over the past two decades, exponential technological advancement has provided a vast ocean of data regarding the performance of various movies in the film industry [14]. Predicting a movie's box office success is complex and is hardly straightforward. The measure of success for every movie is relative; some may consider a movie more successful if it is critically acclaimed, while those would give preference to the revenue generated.

Considering success in terms of revenue from ticket sales for the movie (this is called the ”box office” of the movie) is the usual norm. The movie stakeholders primarily define their return on investment in movies by comparing the movie box office revenues and its production budget. The cinema theatres themselves obtain profit from the distributors' cut and entertainment tax subtracted from ticket the movie's ticket sales [15], it is in the investors' best interest to know beforehand box office value of a movie to allow them to make informed and educated decisions, thus reducing the risk involved in the process.

Taking this one step further would be giving these stakeholders the ability to find the required attributes to release a movie generating a specific revenue value generated through the box office. Such a model would be helpful to potential investors having the capability to verify if a movie would generate the required revenue before investing in the same, but also for directors looking for various production companies to work with to approximate the success of their movie beforehand.

The idea of this project is to develop a model to predict the revenue of a film pre-release and another reverse regression model which could take in the revenue that a film producer wants to generate and help design the budget, cast, runtime, and several other vital aspects of the film could be very useful for production companies before the start of film production.

Final Implementation

Based on the core foundation of the project, data relevant to the project was explored and analysed. Analysis was done with different models to see which features affect results the most. Following this, the data was trained with several different regression models, like XGBoost, RandomForest, CatBoost, LGBM, Ridge, and Voting Regressors, to find the one with the best performance, ultimately used in the implementation.

The final model takes the input parameters as the expected revenue from Box-Office and the movie's genre. The output provided by the model is the budget to be allotted to the movie and the runtime of the movie, along with recommended star power. The expected popularity with all these factors considered is also returned.

II. Literature Survey

Current research in the field involves using machine learning models to predict box office success in terms of revenue. Many models have been built for the purpose and analyzed for performance. Research could be broadly divided into two categories - Quantitative and Qualitative. Quantitative research predicts success in numbers, while Qualitative research predicts whether the movie will be successful or not. Another closely related field in the qualitative domain is based on personal choices that build movie recommendation systems.

978-1-6654-5858-0/22/$31.00 ©2022 IEEE

However, it is essential to note that predicting a movie's success is most relevant at the beginning or before its production. It is even more helpful if the producers are informed about what decisions to make for the movie to succeed. Predicting the requirements for a movie with the required quantified success helps production companies in decision-making. A model that predicts a movie's characteristics based on the revenue to be generated and a particular genre has not been given much attention in the past and forms the motivation for this project. A summary of all the research done is shown in Table I.

III. Methodology

The project was completed in three phases -

- Data Extraction
- Data Processing and Analysis
- Model Building

A. Data Extraction

A significant part of the dataset required for the project was extracted from the global TMDB dataset using its APIs. Following this, the OMDB API was used to extract the MPAA ratings and IMDb ratings and votes of each movie in the dataset. The final dataset has the features - Genres, ID, Original Language, Original Title, Overview, Popularity Rating, Release Date, Title, TMDB Rating, TMDB Vote Count, IMDb ID, Budget, Revenue, Production Companies, Cast, Crew, Production Countries, Spoken Languages, Runtime, Tagline, MPAA Rating, IMDb Rating, IMDb Vote Count and Star Power.

The final dataset has 6065 movies. With regards to Genres, Cast, Crew and Production Companies, the dataset returned a JSON array of responses. The popularity index of the most popular cast in the movie was taken as the star power. Only the top few values in each JSON were considered in the final dataset since these are the elements along with star power that attract majority of the audience.

B. Data Exploration and Modification

The extracted dataset was modified, and some of the essential features were updated. Features like ID, IMDb ID, Original title, and Tagline were not relevant data for the predicting model and were thus removed. All the NULL values in the dataset were removed by changing the NULL values in 'runtime' to the median value and replacing the NULL values in the other columns with an empty set. After removing all the *NULL* values from all entries, the 'release date' feature was modified by splitting it into three distinct features for the day, month, and year of the release. All data except the first three members in the cast of each entry were deleted. Then, a check was done for outliers, which are data points distant from the rest of the data in the dataset. They have the ability to distort the final result and prediction, and thus, were removed.

The given dataset was then explored to understand the relationships between the features given, how they interact with each other, spot anomalies in the data, and find patterns to help build the model. For this purpose, histograms were plotted to study the range of features like runtime, budget, popularity, release-data, IMDb rating, revenue, etc., to study the range of this data. Following this, a correlation matrix was plotted with the same features mentioned above to find the linear interaction between every pair of features. This contained the correlation coefficient between each pair. In addition to this, bar plots and frequency polygons were plotted to study the given data [9].

C. Data Preprocessing

The library 'sklearn', a robust and commonly used library for machine learning problems, was used to preprocess the dataset. SimpleImputer replaced the missing values with median values and a power transformer to transform the data to look more Gaussian, minimize variance, and stabilize skewness for features like budget, runtime, and popularity. After dropping all unwanted features, label encoding was performed on the categorical features, as they needed to be made more expressive. The original category columns were removed, and a transformed version of the dataset was returned, which was suitable for performing modeling. After all the processing, transformation, and label encoding, the final dataset contained eleven features. This data was then split into test and train.

D. Model Building

The preprocessed data was passed through different regression models [12]. Specific functions were written to calculate the model's performance, i.e., the train Mean RMSE and test mean RMSE for the final model. The 15 most important features according to the results of each regression model used were found, and the similarity between the data was analysed. The regression models used are listed and explained below [7].

1) Ridge Regressor: This is a technique for analysing multicollinearity in multi-regression data. When multicollinearity occurs, minimum squares estimate values are generally unbiased, but their variances are enormous. Thus they could be far away when compared to the actual value. Ridge regression reduces standard errors by adding a degree of bias to the regression estimates. It is hoped that the net effect will provide more reliable estimates. This model is used from the sklearn library and handles a regression problem. The loss function is the linear least-squares function, and the L2 normalization is used for regularisation.

2) Random Forest Regressor: This regression algorithm uses a technique that integrates predictions from various machine learning algorithms to get a more accurate prediction than a single machine learning model. During training, a Random Forest constructs many decision trees and outputs the mean of all the classes involved as the final prediction of all the trees. There is no interaction between the decision trees as they run in parallel to each other and perform their learning. This is also implemented using sklearn by fitting several decision trees on sub-samples of the dataset whose size is controlled by the user, and averaging is used to avoid overfitting and improve accuracy [5].

TABLE I
SUMMARY OF LITERATURE SURVEY

Ref No.	Authors	Study Description	Conclusions/Results
[1]	Subramaniyaswamy V., Viginesh Vaibhav M., Vishnu Prasad R. and Logesh R.	This paper predicts the box-office success of movies with the help of multiple linear regression and Support Vector Machine (SVM) Classification taking into consideration factors affecting it such as Wikipedia reviews, trailer views, Wikipedia page views, time of release, and critics ratings.	Multiple variables were trained with SVM, and the final result had a higher accuracy than other previous attempts using SVM on a single variable.
[2]	Sameer Ranjan Jaiswal and Divyansh Sharma	This paper applies machine learning to develop a model capable of predicting a Bollywood movie's success even before it is released. This model has been created with the Bagging Algorithm.	The model developed had an accuracy of 82.01%. However, the dataset size appeared to be smaller than required to be a full-scale model.
[3]	Euna Mehnaz Khan, Md. Saddam Hossain Mukta, Mohammed EunusAli and Jalal Mahmud	This paper proposes new ways of predicting a user's genre preferences in the movie from the user's social media presence, particularly Twitter, with the help of machine learning models.	The models coded were combined, which was found to have higher accuracy than any of the single models developed. The experiments showed that the model was effective in recommending movies to users.
[4]	Nahid Quader, Md. Osman Gani, Dipankar Chaki and Md. Haider Ali	This paper uses Support Vector Machine (SVM), Natural Language Processing (NLP), and Neural Network to predict a movie box office success with the help of features that affect the performance before and after the release of a movie (such as genre, budget, IMDB reviews, etc.)	The results showed that the Neural Network model provided better accuracy than the SVM.

3) XGBoost Regressor: XGBoost (Extreme Gradient Boosting) is a gradient boosting library that has been optimized for efficiency, flexibility, and portability. It uses the Gradient Boosting framework and uses the gradient descent algorithm. It is a parallel tree boosting strategy that improves the accuracy and speed while building the model. It is an ensemble learning method, much like Random Forest, that uses several decision trees in parallel. However, it boosts them by correcting the previous mistakes made and rectifying them to improve future performance. XGBRegressor class of XGBoost is used here to build the model. It contains many hyperparameters like 'subsample', colsample-bytree', 'colsample-bylevel', etc can be tuned to improve results and reduce RMSE [8].

4) CatBoost Regressor: CatBoost is a newly developed Russian open-source ML algorithm similar to XGBoost and works on the concept of gradient boosting and decision trees, performs greedy search sequentially and puts together several basic models to build a robust model that works fast and gives high accuracy predictions. However, during the growing of the decision trees, CatBoost is different from XGBoost. It grows oblivious trees, which implies that nodes at the same level must test the same predictor with the same condition, and so a leaf index may be determined using bitwise operations. The oblivious tree technique is CPU-efficient, while the tree structure acts as a regularisation to discover the optimal solution to avoid overfitting. CatBoostRegressor class is used from CatBoost library, and also contains several hyperparameters which can be tuned, and is a better fit for some datasets than XGBoost regression [6].

5) LGBM Regressor: Light GBM, like XGBoost, is also a high-performance gradient boosting framework based on the decision tree technique that may be used for ranking, classification, and a variety of several other ML applications. It splits the tree leaf-wise because it is based on decision tree algorithms, whereas other boosting methods split the tree depth-wise or level-wise rather than leaf-wise. As a result, when growing on the same leaf in Light GBM, the leaf-wise approach reduces the loss compared to the level-wise algorithm, resulting in significantly higher accuracy than any of the other boosting strategies like XGBoost and CatBoost. It is also much faster than XGBoost, so it is called 'Light' GBM [10].

6) Voting Regressor: Voting Regressor is an ensemble method present in sklearn library, whose final model and prediction is a combination or ensemble of a group of trained models, which, in this case, are the models discussed above, like Random Forest, XGBoost, Ridge, CatBoost, and Light GBM regression models. The final prediction is an aggregate of all these above predictions and is equal to the mean predicted target value of the above regression models. Thus, due to combining all of these models, the Voting Regressor model often performs better than the remaining individual models, and it is best to use this as the final model.

After training the dataset with the models mentioned above, the RMSE error of the test and train data was checked for each of the models. The five most critical contributing features for each regression model were plotted. The final best model was then put together, i.e., the voting regression model, which combined all the above models and gave out the mean. This column was then used to predict the revenue for the entire dataset and plot the scatter plot graphs for several features against actual and predicted revenue values and study the model's accuracy [11].

E. Backtracking From Revenue

After making revenue predictions for the dataset, a model was built to solve the reverse problem. The revenue that a producer wishes to generate is taken as input. The model attempts to accurately predict the essential features required by the movie to achieve this result. The input features are the revenue predictions achieved in the previous section and the genre feature, which has been labeled encoded. Random Forest Regression was used for this problem, taking the above input features and training separate models for different output features, i.e., the budget, movie runtime, star power required, and the popularity index. After the four models were trained, the interaction between the four different predicted features was studied using graphs and correlation matrices to allow modifying the models as required to ensure that the predictions have correlations and are not independent of each other.

The runtime and budget models only use the above two input features. In contrast, the popularity and star power predictors take in the predicted budget from the previously trained model and use that as an additional feature to generate their final predictions.

The movie genre and the expected revenue are the user input, and the budget required, preferred runtime, star power, and the expected popularity index are predicted by the model built.

IV. Results and Analysis

The correlation matrix table was constructed using the features in the dataset to study the linear correlation between the revenue and other features in the dataset before starting training and is shown in Fig 1. It was found that the highest correlation coefficients in the matrix with respect to revenue was the budget (0.71), followed by IMDb votes (0.59), followed by the popularity, runtime, and star power.

The dataset was trained with different regression models, and the Root Mean Squared Error (RMSE) for both the test and train data derived from each of the models is noted in Table II.

TABLE II
RMSE for different regression models

Model	Train RMSE	Test RMSE
Ridge Regressor	384.162719	379.516703
RandomForestRegressor	271.665345	357.068054
XGBRegressor	139.052521	348.237095
CatBoostRegressor	220.020737	344.352039
LGBMRegressor	351.395364	362.607178
VotingRegressor	265.167752	351.252695

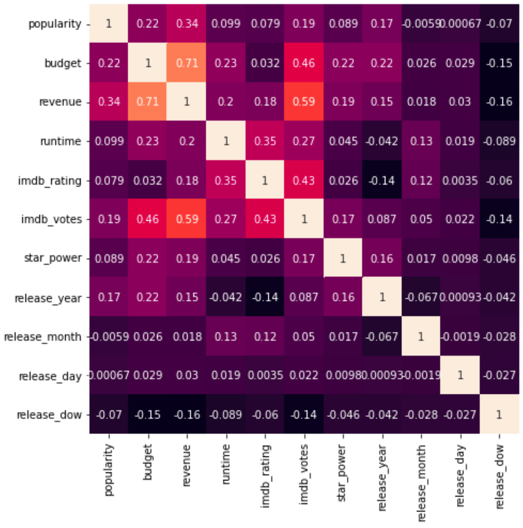

Fig. 1. Correlation between features

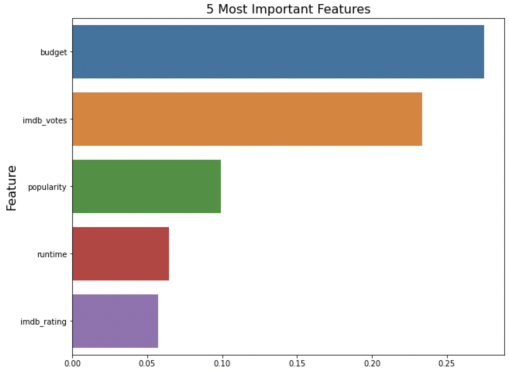

Fig. 2. Random Forest Feature Importance

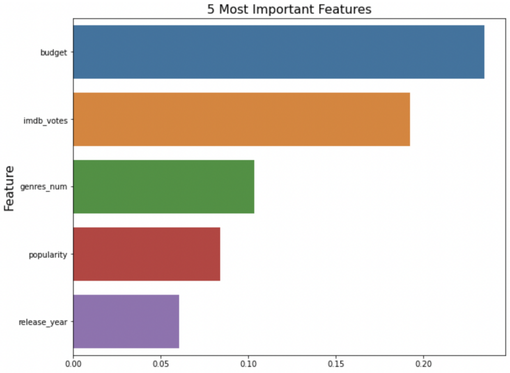

Fig. 3. XGBoost Feature Importance

From the above table (Table II), the following observations can be made :

1) It is observed that the model that gives the most significant error and performs the worst is the Ridge model, and both it is Train and Test RMSE are higher than the other models. This is because this model is generally

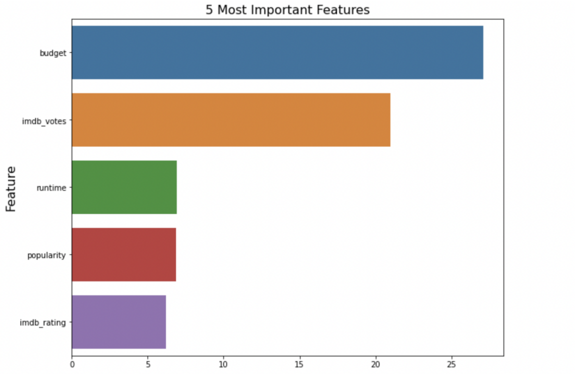

Fig. 4. CatBoost Feature Importance

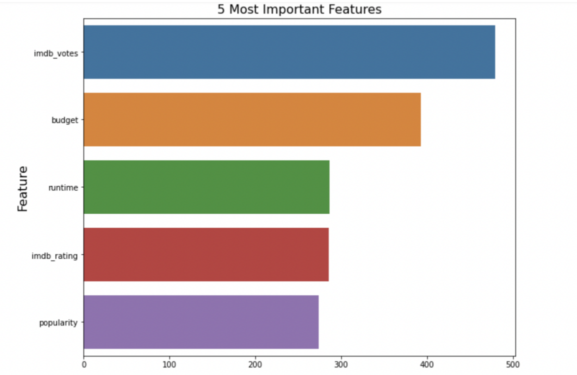

Fig. 5. Light GBM Feature Importance

preferred for datasets where the features have a high correlation with each other.

2) The Random Forest Model has a much lower error than Ridge due to its ensemble method and decision tree process for arriving at an optimal solution.
3) XGBoost and CatBoost both use gradient boosters to improve speed and accuracy compared to random forest regression, and thus, both have a better Train and Test RMSE. It is noticed that XGBoost has a very low Train RMSE compared to CatBoost but a higher Test RMSE than it. This is because the XGBoost model seems to have overfitted the dataset and is too specific to the training data. CatBoost does a better job of avoiding overfitting with the given dataset.
4) LGBM Regression model does not perform as well as either CatBoost or XGBoost. This could be because the model has not been fine-tuned well enough, and it may be underfitting the given dataset.
5) Voting Regressor model is the final model used for predicting the revenue data as it was taking the mean of all the previously used models.

The RMSE values are seen to be high and this can be due to the irregular distribution of revenues and the fact that the dataset has a lot of features that do not strongly influence the predicted values. The lesser important features introduce the high error during prediction. Since features like Budget, IMDb Votes and Popularity are much more significant than the others as discussed below, these are the ones considered in the Backtracking Model.

The five most important features according to each of the models were plotted and are shown from Fig 2. to Fig 5. Since the most important features vary based on the model, the Voting Regression was used to combine all these results.

The final model is used to predict the revenue for the entire dataset. To check how accurate the actual revenue data is compared to the predicted data, scatter plots for Revenue vs Budget are plotted (Fig 6). It can be seen that the regions of predicted and actual values are very similar.

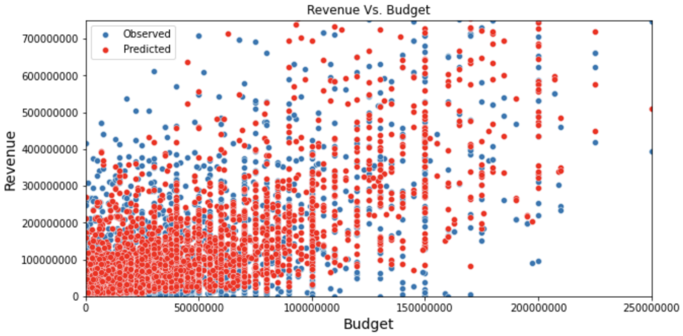

Fig. 6. Revenue vs Budget

A. Revenue Backtracking

After the revenue is predicted for the dataset, the other features are backtracked by passing the revenue and genre of the film as input. The model is trained using Random Forest Regression, which is ideal since only two features (genre and expected revenue) are passed as inputs to predict the budget required, runtime preferred and star power required to achieve input revenue. The model was found to provide accurate results for the three output features. The scatter graphs of the actual and predicted values of each of the features to the revenue were plotted below from Fig 7. to Fig. 10, and we can observe that the regions of predicted and real values are very similar, but the predicted value plot does not have too outliers in the graph.

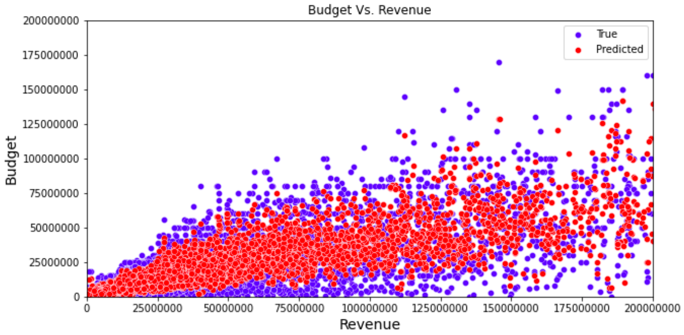

Fig. 7. Budget vs Revenue

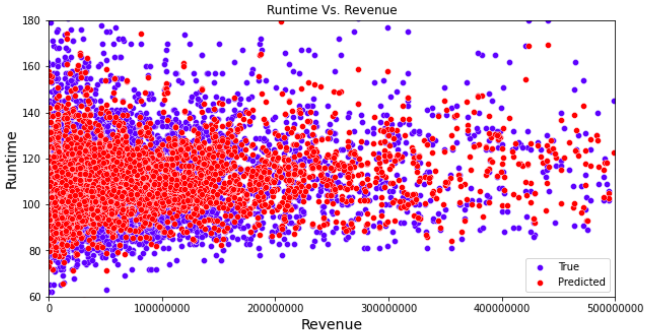

Fig. 8. Runtime vs Revenue

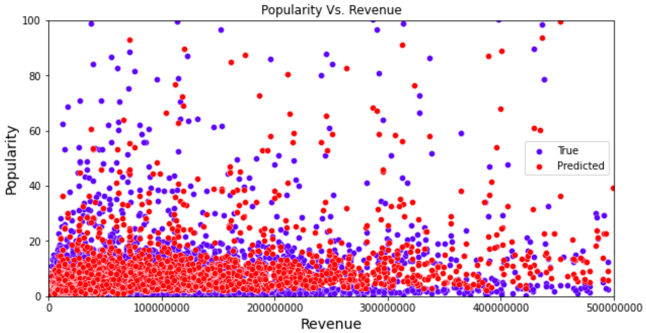

Fig. 9. Popularity vs Revenue

V. Conclusion

This paper successfully implements a Box Office Success Recommendation System, which can be used by Production Companies across the globe to assist them with decision making at the beginning or before the production of a movie. The machine learning model that taken in an expected revenue and is able to give the required features for achieving that goal would be extremely useful during Film Productions, and can help Companies make more well informed decisions while setting out to make a new film.

This paper has been able to perform detailed data exploration, model building using several algorithms, study of the important features in revenue prediction, graphically represented the predicted and true data to understand the performance of the model, and has performed detailed data extraction to achieve a high accuracy. Also, the advantages and differences between each of the different regression models were observed and analysed with respect to their performance. The reverse prediction system taking an input of revenue was observed to work with good accuracy, despite the fact that there were very few input features and a basic regression model was used.

In the future, instead of making each of the features like budget prediction and runtime prediction during backtracking completely independent, the predicted output features could interact with each other and update their results according to each other's predictions. More complex training algorithms like CNNs and RNNs could be used to give better results. Also, an alternative approach to using the revenue to predict all the features could be done using Generative Adversarial Networks (GANs). GANs are a semi supervised learning algorithms that would be able to generate completely different data each time, and applying GANs instead of supervised learning algorithms, could give far more promising results.

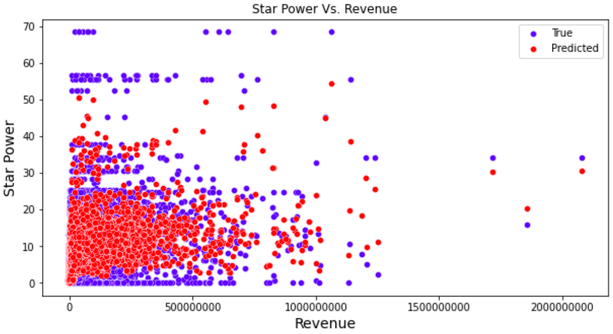

Fig. 10. Star Power vs Revenue

References

[1] V. Subramaniyaswamy, M. Viginesh Vaibhav, R. Vishnu Prasad and R. Logesh, "Predicting movie box office success using multiple regression and SVM", *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, Dec. 2019, pp. 182-186

[2] S. R. Jaiswal and D. Sharma, "Predicting Success of Bollywood Movies Using Machine Learning Techniques", *Compute '17: Proceedings of the 10th Annual ACM India Compute Conference*, Nov. 2017, pp. 121-124

[3] E. M. Khan, Md. S. H. Mukta, Md. E. Ali and J. Mahmud, "Predicting Users' Movie Preference and Rating Behavior from Personality and Values", *ACM Transactions on Interactive Intelligent Systems*, vol. 10, Sept. 2020, pp. 1-25

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

[5] J. Ali, R. Khan, N. Ahmad, I. Maqsood, "Random Forests and Decision Trees", *IJCSI International Journal of Computer Science Issues*, vol. 9, Sept. 2012, pp. 272-278

[6] A. V. Dorogush, V. Ershov, A. Gulin, "CatBoost: Gradient boosting with categorical features support", *arXiv:1810.11363*, Oct. 2018

[7] S. B. Jha, R. F. Babiceanu, V. Pandey, R. K. Jha, "Housing Market Prediction Problem using Different Machine Learning Algorithms: A Case Study", *arXiv:2006.10092*, Jun. 2020

[8] T. Chen, C. Guestrin, "XGBoost: A Scalable Tree Boosting System", *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 785–794

[9] J. DSouza, S. S. Velan, "Using Exploratory Data Analysis for Generating Inferences on the Correlation of COVID-19 cases", *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2020

[10] G. Ke1, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T. Y. Liu *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*, *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems* Dec. 2017, pp. 3149–3157

[11] J. R. Quinlan (1986, March), "Induction of Decision Trees", *Machine Learning 1, 1986*, Mar. 1986, pp. 81-106

[12] Z. Qin, L. Yan, H. Zhuang, Y. Tay, R. K. Pasumarthi, X. Wang, M. Bendersky, M. Najork, "Are Neural Rankers still Outperformed by Gradient Boosted Decision Trees", *International Conference on Learning Representations (ICLR)*, 2021

[13] Gaurav Laghate (2020, February), *Indian box office crosses Rs 10,000 crore mark in 2019*, *The Economic Times*

[14] *A Very Short History of Cinema*, (2020, June), *The Science and Media Museum*

[15] Stephen Follows (2017, February), *How is a cinema's box office income distributed?*, *Stephen Follows Film Data and Education*