CONCORDIA UNIVERSITY

FINAL REPORT

COMP6321

Movie Success Forecaster with Intelligent Feature Engineering

Author:	*Author ID:*
Somayeh GHAHARY	40106359
Mehrnoosh AMJADI	40091264
Bikash JAISWAL	40115186

Submitted to:
Dr. Andrew DELONG

April 13, 2021

1 Introduction

With the unprecedented growth of the film industry, stakeholders and producers have become very interested in investing in this field. However, there is always the concern that whether this investment will be profitable for them or not. In this regard, the existence of a system that can predict the success rate of a film before its release, allows producers to invest in this field with a better perception like [8] and [5]. This project is dedicated to predict how successful a movie would be prior to its release by efficient machine learning methods.

Our primary objective in this project is to learn how to apply different Machine learning models for predicting the category of success in which a movie belongs, prior to its release in box offices. The secondary goal is to observe the importance of the cast and crews involved in movies towards its success in box office. To be specific, we want to know what is the effect of the Director's revenue and actor's popularity features with some additional features in prediction and does it make sense that they invest on hiring popular actors and actress or highest revenue directors.

The rest of the report is organized as follows: Section 2 provides a discussion on relevant information about datasets, preprocessing, feature engineering and visualization. Section 3 reports the implementation, evaluation of different training models on our datasets. Then the conclusion of our analysis is provided in section 4. Finally future works are mentioned in section 5.

2 Methodology

2.1 Dataset

The dataset for this project is "The Movie Dataset" [6], which is a combination of dataset obtained from TMDB [4] and GroupLens [1]. It includes two dataset files, the main file "movies_metadata.csv" consists of 45,000 movies with 23 features like budget, revenue, release dates, languages, production, countries, etc,. Whereas, another dataset "credits.csv" has cast and crew information for all our movies in the main dataset. The both dataset files contain a variety of heterogeneous data recorded in text, json, numeric and web links formats.

2.2 Tools Used

We use the following tools and technologies for implementation of the project in table 1.

2.3 Preprocessing

The first step of any machine learning project is to understand datasets. We extract eight useful information from "credits.csv" dataset, mainly the top three lead actors and director's name and gender of each movie, and also eight main features of "movies_metadata.csv " like genre, release_date, popularity, revenue. We remove some irrelevant features (based on our intuition) from datasets like overview, poster_path, production_companies, production_countries, runtime and etc. Then we merge useful features based on their unique IDs, as a unique feature of the two dataset. To make our training and prediction process efficient, we remove those movies whose 'budget' or 'revenue' information are missing because these two features are critical for prediction.

One of the most crucial parts of the preprocessing is handling the null and empty values. In this regard, first of all, we substitute the blank and zero data with 'nan' values and then remove all the 'nans' from the entire dataset due to having a clear data. The data set contains 45,460 records, which drops to 5,393 records after the 'nan' values are removed[1]. As we categorize our data based on profit_rate values, and because we only have 5,393 records available for this label, we decide to remove all the 'nan' values to make an accurate prediction. We normalize the final dataset since normalization is a essential part of machine learning to scale raw data and make them suitable for training on the machine learning models. We use StandardScaler method for normalization from scikit-learn library [9].

2.4 Feature Engineering

2.4.1 Adding New Features

We propose to add some new features to analyze if they would help to increase the performance of the prediction of the models.

- **Revenue_director_mean and popularity_actor_mean:** Since the cast and crew have

[1] After consulting with the professor about only 5,393 available target data, he suggested removing all 'nan' values from the entire dataset.

Tools	Purpose
Python3, Anaconda, Jupyter Notebook	Programming Language and IDEs
Numpy, Pandas	Numerical Analysis
Matplotlib, Seaborn	Visualization and Plotting
Scikit-learn [9]	Machine Learning Packages

Table 1: Tools and technologies used

a huge impact on the success of a movie in box-office, movies that are directed by directors who have earned more profit from their past movies or have famous actors/actresses might be predicted as more successful movies. So we calculate the mean revenue of each director and mean popularity of lead actor/actress based on their other movies records [12].

- **Imdb_rating:** Rating of a movie is an important factor to determine its popularity among the public. If the ratings are low, most probably, viewers wouldn't prefer to go to the box-offices to watch the movies. But giving rating to movies based on individual 'vote count' or 'vote average' would result in an injustice to the ratings as average rating of 9 with just 2 vote count is less effective then average rating 8.2 with 12 vote count. So we adapted the IMDb weighted rating [2] methods to calculate the weighted rate (WR) of each movie. The methods use the following formulas which provides a true 'Bayesian estimate', which utilises number of votes for the movie, average rating of the movie, mean vote across the whole dataset and minimum votes required to be on the list (at least 90% more vote than other movie vote).

$$WR = \frac{v}{v+m}R + \frac{m}{v+m}C \tag{1}$$

  - v: the number of votes for the movie
  - m: the minimum votes required to be listed in the chart
  - R: the average rating of the movie
  - C: the mean vote across the whole report

Finally we consider two X datasets to train and evaluate them separately to observe if the new features would improve the prediction process. X1 dataset has the new features revenue_director_mean, popularity_actor1_mean and imdb_rating. While the other dataset X2 does not have the new features and only have features from metadata.csv files (only raw features). We split both the data into training, validation, and test data with the ratio 60:20:20 respectively.

2.4.2 Labeling Target

To label the data, we create a new variable called profit_rate. Profit_rate is defined as the rate of revenue over the budget:

$$profit\_rate = \frac{revenue}{budget} \tag{2}$$

From distribution of profit_rate we get information such as minimum, first quartile, median, third quartile, and maximum. We use the first four pieces of information to categorize the y label. In general, four categories are created and data is labeled based on these divisions. In the following, we consider these four categories as loss, normal, successful and hit. We encode this four categories to 0, 1, 2 and 3 respectively. As shown in 2, for example, movies with profit_rate in the range of minimum and first quartile are classified as a loss.

2.5 Visualization

We provide a bunch of plots to visualize the data and get a better sense of them as it is usual at most of the machine learning projects. In phase 3 of Jupyter Notebook, we picture our target data profit_rate from different aspects. In section 3.12 of Jupyter Notebook, we can see distribution of profit_rate. We also observed that around 15% of data seems like outliers. Since these outliers influence presentation of other data, we decide to remove them from this plot. Moreover we label the target data based on this plot. Also we plot the sum of profit_rate of movies that were released in each month from 2010 to 2017 in plot 3.13 of Jupyter Notebook. Refer to phase 3 of Jupyter Notebook for more information about description and analysis of other plots.

y values vs pr range	$min \leq pr < 1^{st} quartile$	$1^{st} quartile \leq pr < med$	$med \leq pr < 3^{rd} quartile$	$3^{rd} quartile \leq pr$
y (value)	0	1	2	3
y (tag)	loss	normal	successful	hit

Table 2: y labeling

Model Name	Parameters	Best parameter X1 includes new	Best parameter X2 has only raw	Score of best estimator train X1	train X2	validate X1	validate X2	test X1	test X2
Logistic Regression	C=[0.001, 0.01, 0.5, 0.85, 0.9, 1] max_iter=1000 random_state=0	C=0.5 max_iter=1000 random_state=0	C=0.96 max_iter=1000 random_state=0	0.83	0.83	0.84	0.84	0.82	0.81
Neural Network MLPerceptron	batch_size = [200, 250, 300, 350, 400, 450, 500] random_state=0 hidden_layer_sizes =(30,20) solver='Adam' learning_rate_init =0.09 momentum=0.9	batch_size=450 random_state=0 hidden_layer_sizes =(30,20) solver='Adam' learning_rate_init =0.09 momentum=0.9	batch_size=200 random_state=0 hidden_layer_sizes =(30,20) solver='Adam' learning_rate_init= 0.09 momentum=0.9	0.86	0.89	0.84	0.89	0.83	0.86
SVM	C = [0.1, 10, 50, 100,250, 350, 425, 500] gamma=0.001 max_iter=1000 random_state=0	C=409.14 gamma=0.001 max_iter=1000 random_state=0	C=409.14 gamma=0.001 max_iter=1000 random_state=0	0.82	0.81	0.82	0.81	0.79	0.79
Random Forest	max_depth=[1, 5, 15, 25, 50, 100] random_state=0	max_depth=15.16 random_state=0	max_depth=19.43 random_state=0	1	1	0.82	0.85	0.78	0.83
Decision Tree	max_depth= [1, 5, 15, 20, 50, 75, 90] splitter='random' random_state=0	max_depth=20.92 splitter='random' random_state=0	max_depth=18.48 splitter='random' random_state=0	0.99	0.98	0.76	0.84	0.78	0.83
Ada Booster	n_estimators=[4, 128, 256, 512, 1024, 2048, 4093] algorithm ='SAMME' random_state=0	n_estimators=4090 algorithm ='SAMME' random_state=0	n_estimators=4093 algorithm ='SAMME' random_state=0	0.55	0.55	0.52	0.52	0.54	0.54

Table 3:

3 Implementation and Results

3.1 Model training

In this project, we have chosen six most popular machine learning models to evaluate the prediction on training, validation and test data. We use RandomSearch cross validation technique [7] for choosing an optimal model based on some hyperparameter. In table 3, model name, hyperparameters used to extract the best model, best model's parameter and score of training, validation and test data returned by the best model, are recorded.

3.2 Observations and Results

As a result of model training step, we come to following considerations: The best model based on our results would be the Neural Network since it has the highest training, validation and test accuracy in both X1 and X2 experiments relative to other models. Despite having nearly 100% accuracy in training data, Random Forest and Decision Tree has resulted in lower accuracy in prediction of new movies. Also, among all the training models, AdaBoost has performed the worst, with accuracy in range of 50%.

For Logistic Regression, adding new features improves the score of the test dataset, although for train and validation, it is the same. While performance of SVM and AdaBoost is fixed even after adding these new features. The score of other models, Neural Network, Random Forest and Decision Tree decrease a bit on X1, the dataset with new features.

4 Conclusion

We might think that the popularity of the lead actor or mean revenue of director of a new movie can be a driving force towards the success, however, our observation refutes this assumption. We might assume there must be various other factors that need to be taken under consideration like movie scripts, cinematography and others, before investing a lot by the production team. With reviewing various research [11][10] and also the results that we obtained from our experiment, we came to this conclusion, the Neural network is the model that can rely on movie predictions.

5 Future Work

We could use semi-supervised learning [3] to add the values to revenue and budget features so we have not had to remove the movies with Nan value. This could improve the model prediction. Moreover Utilizing Deep Learning might improve the performance of the prediction. Also predicting based another target value may lead to better and more precise outcomes. Instead of analyzing on just one dataset, it would be possible that merging multiple open datasets available on different platform.

References

[1] MovieLens Latest Datasets. 1

[2] IMDB Rating. 2

[3] Semi-Supervised Learning. 4

[4] The Movie Database (TMDb) API. 1

[5] S. Adhikari. How to use machine learning approach to predict movie box-office revenue / success? https://medium.com/analytics-vidhya/how-to-use-machine-learning-approach-to-predict-movie-box-office-revenue-success-e2e688669972, 2020, May 01. Accessed: October 09, 2020. 1

[6] Rounak Banik. The Movies Dataset, Nov 2017. 1

[7] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(null):281–305, February 2012. 4

[8] Sharda R. Delen D. Predicting box-office success of motion pictures with neural networks. *Expert Systems with Applications*, 30(2):243–254, 2006. 1

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 1, 2

[10] Nahid Quader, Md. Osman Gani, Dipankar Chaki, and Md. Haider Ali. A machine learning approach to predict movie box-office success. *2017 20th International Conference of Computer and Information Technology (ICCIT)*, 2017. 4

[11] Travis Ginmu Rhee and Farhana Zulkernine. Predicting movie box office profitability: A neural network approach. *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016. 4

[12] Willacy. Director and Actor's Total Gross and IMDB Score, Jan 2020. 2