Predicting the Successfulness of Movies in IMDb Using Linear Models, SVMs and Neural Networks

Yunge Hao

Abstract

Predicting and classifying the successfulness of movies yet to be released is of great interest to movie industry. Learning how successful a movie is going to be can help movie companies making adjustments to investment and advertising. In this project, I applied several machine learning techniques to train a movie prediction model and test the results.

1 Introduction

Making predictions to movies, their popularity, revenue and rating, is a very important part of movie production prior to the release of the movies. The movie industries need to know whether they are going to make a profit or loss. Predictions about revenues of movies may not need be too precise, however, since revenue in dollar can vary a thousand without making much difference in overall opinion. Therefore, my goal in this research is not to make accurate revenue predictions, but instead, to make classifications of movies based on the range of their predicted revenues.

The machine learning models I choose to use are ordinary least squares linear regression, logistic regression, linear support vector machine, RBF kernel support vector machine and neural networks. All methods are implemented and tested using sklearn package.

The paper is organized as follows. In section 2, some works related to movie prediction are discussed. In section 3, I will first describe the dataset I used and the selection of the features, then how I handled data preprocessing. In section 4 I will analyse the results obtained from those machine learning models that I use, and give explanations to certain errors. Section 5 is for conclusions and future improvements.

2 Related Work

As mentioned earlier, movie prediction is a very popular research area. There are many people getting good results with various techniques.

Linear models including linear regression and logistic regression, and support vector machines have been studied for predicting revenue of movies by Nithin Vr et al. [1]. Their dataset contains 1050 movies released from 2000 to 2012. In order to get good forecast, they only included movies produced in the U.S. and whose language is English. Their objective is predicting the revenues of movies with a given rate of error tolerance. The result of using linear regression is a success rate of about 51% with 20% tolerance. Logistic regression has a success rate of 42.2% with 12.5% tolerance. SVM approach is even worse, only 39% with 20% tolerance.

Another similar research is done by Yoo et al. [2] on linear regression and logistic regression only. Their linear regression model can get a correlation of 0.7479. Logistic regression is used for classification based on different bucket ranges for the gross. The accuracy is still low, slightly less than 50%. The dataset contains over four thousand movies from 1913 to 2011. They grouped the features into simple features(numeric), complex features(text-based) and sentiment feature represented solely by sentiment score. But the similar accuracy the comparison between their performance yields does not tell much. They also found out that as the number of buckets increases, the accuracy drops dramatically, while the correlations between the revenue and feature sets remain constant.

Abidin, Bostanci and Site have some different approaches [3]. They used classification algorithms including J48, MLP, Random Forest, Bagging, BayesNet, LMT and PART. Random Forest is reported to have the best test performance, of 92.7% success rate, for predicting movie ratings.

Mladen Marovic et al. [4] focused on predicting movie ratings by users in automatic recommendation systems. Their methods are highly

content-based regression tree and neural networks, as well as k-NN in collaborative methods and SVD-kNN in hybrid methods.

3 Dataset

3.1 Description and Features

It can a be very tedious job scraping information about a large amount of moives on the internet without a proper collection of dataset. Fortunately, the Internet Movie Database(IMDb) provides one of the most popular and convenient source for getting information about movies.

all features
id, imdb id, popularity, budget, revenue, original title, cast, homepage, director tagline, keywords, overview, runtime, genres, production companies, release date, vote count, vote average, release year

Table 1: All Features

nominal features
original title, cast, homepage, director tagline, keywords, overview, genres, production companies, release date

Table 2: Nominal Features

numeric features
popularity, budget, revenue, runtime, vote count, vote average

Table 3: Numeric Features

The dataset I used is directly downloaded from Kaggle where data from IMDb is already organized in a table for ease of use.
Table 1 shows the entire features of the dataset. The feature set can be divided into two groups, numeric features and nominal features. The nominal features as shown in table 2 are features that does not have a direct numeric value, but would be great advantageous if their content value is well-studied. In my study, however, only numeric features are involved. All numeric features are shown in table 3. The target for the models to predict is the revenue. All budgets and revenues are already adjusted according to the inflation rate.

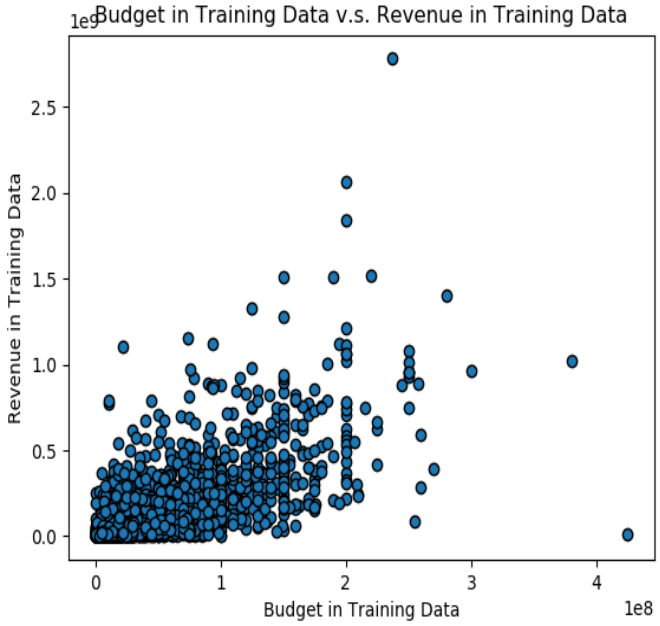

Figure 1: Budget vs Revenue in Training Set

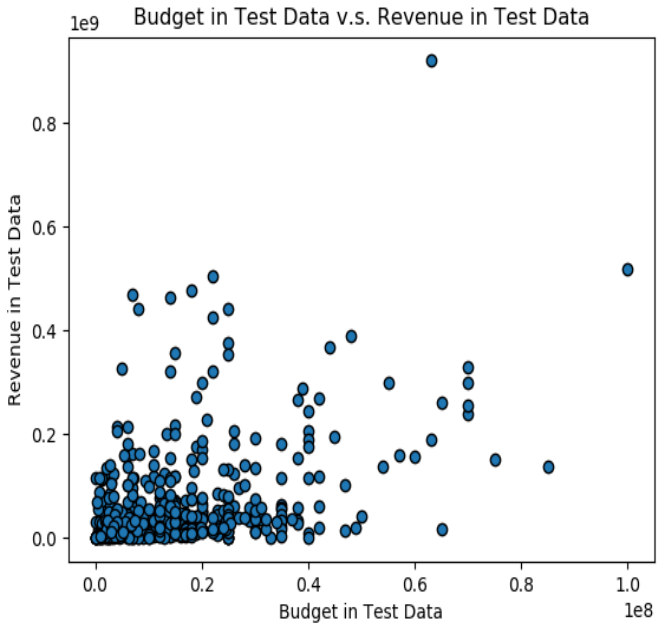

Figure 2: Budget vs Revenue in Testing Set

Figure 1 and 2 plot budget versus revenue, taken natural log. From the plot we can see that majority of the data are distributed near bottom left corner. Movies with high budget or high revenue are very rare.

3.2 Data Preprocessing

The original dataset has 10876 rows in total, but many of them are not complete. Thus, I only extracted rows with complete numeric features, so to make sure no item in the data contains a 0 value in a column. The resulting dataset has 3855 movies. Among them, 500 movies are taken out for testing, and the rest 3355 movies are for training.

3.3 Data Normalization

Since I am training the models to predict for the revenues, and revenues can have a fairly large interval, it looks reasonable to have the data normalized column by column. However, the revenues and budgets are not distributed evenly over their domain, as shown in previous figures. They are more condensed around lower half of the domain. It would cause difficulty to define classes of revenues. Therefore, in my experiment, there is no normalization step. But it would be necessary and beneficial to have a normalization if the predicting target is the rating [3].

4 Results and Analysis

There are 5 classes of successfulness to classify, based on the ranges of the revenues. Class 0 indicates the least successful movies, with revenues less than 50 million dollars. Class 1 are movies with revenues between 50 million dollars and 100 million dollars. Class 3 are movies with revenues from 100 million dollars to 300 million dollars. Class 4 are movies with revenues from 300 million dollars to 600 million dollars. Class 5 indicates the most successful movies, with revenue between 600 million dollars and 1000 million dollars.

The results are presented in a confusion matrix for each model. The analysis is based on the following 5 indicators [5].

1. Accuracy

Accuracy for each class is the ratio between the total number of correct classifications for that class and the total number of all classifications. Normally the higher the better. But not necessarily true, if the distribution of the data is not uniform. Then accuracy alone does not make much sense and it cannot represent the information about what we actually care about.

2. Precision

Precision for each class is the total number of correct classifications for that class divided by the total number of all classifications made by the classifier of that class. It represents the ability not to label negative sample positive.

3. Recall

Recall for each class is the ratio between the total number of correct classifications for that class and the total number of all items that are actually in that class. It represents the ability of the classifier to correctly classify true positives.

4. F-Score

A weighted harmonic mean of the precision and recall, 1 is the best and 0 the worst.

5. Support

It is the actual number of items in each class.

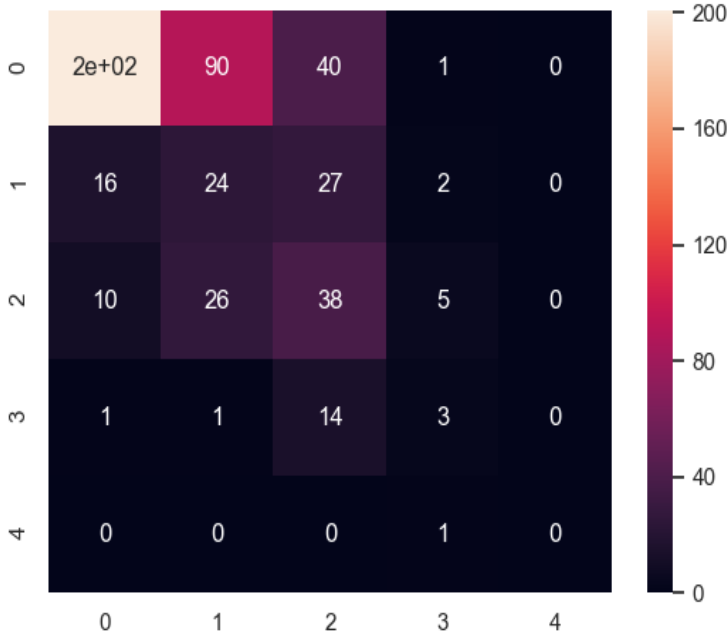

Figure 3: Confusion Matrix for Linear Regression

The accuracy of linear regression model decreases, in general, as the class gets higher, except for class 2 where the accuracy is higher than class 1. Precision for class 0 is highest, then class 2. Class 1 has the lowest precision. Recall and F-score has a similar pattern as accuracy.

It could be attribute to the total number of movies actually in that class, indicated by the support column. Higher classes lack sufficient data to give space for error tolerance, and it kills the performance.

But linear regression behaves more stably

even in making mistakes, compared with other models, when the class is low. By the heap map in figure 3, it tends to make either errors, over classifying as well as under classifying. Its F-score also changes less sharply.

class	accuracy	precision	recall	F-score	support
0	0.402	0.88	0.61	0.72	332
1	0.048	0.17	0.35	0.23	69
2	0.076	0.32	0.48	0.38	79
3	0.006	0.25	0.16	0.19	19
4	0	0.00	0.00	0.00	1

Table 4: Report on Linear Regression

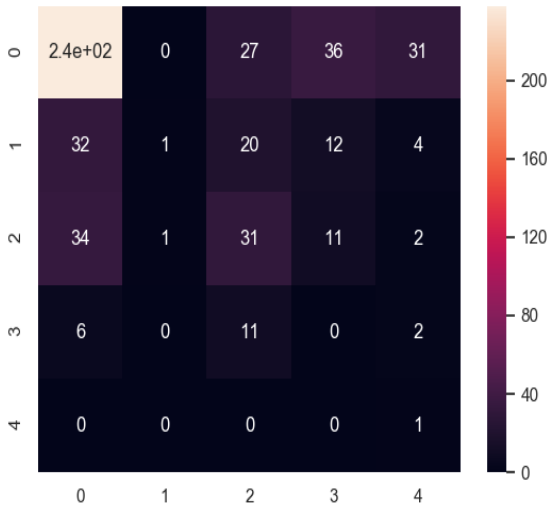

Figure 4: Confusion Matrix for Logistic Regression

Logistic regression has best performance on class 0, then class 2. The fact that class 4 has only 1 occurrence of revenue does not necessarily imply a good classifier for being able to find this one when classifying. But considering all other classifiers fail to do so, i.e., a zero accuracy and precision, logistic regression is indeed a better one in extreme case. The mistake logistic regression most likely to make is labeling a lower class.

class	accuracy	precision	recall	F-score	support
0	0.476	0.77	0.72	0.74	332
1	0.002	0.50	0.01	0.03	69
2	0.062	0.35	0.39	0.37	79
3	0.00	0.00	0.00	0.00	19
4	0.002	0.03	1	0.05	1

Table 5: Report on Logistic Regression

Linear SVM is very good for class 0 where most low revenues occurred. But it does not function on higher classes. It has a strong tendency to make classifications higher than the actual class.

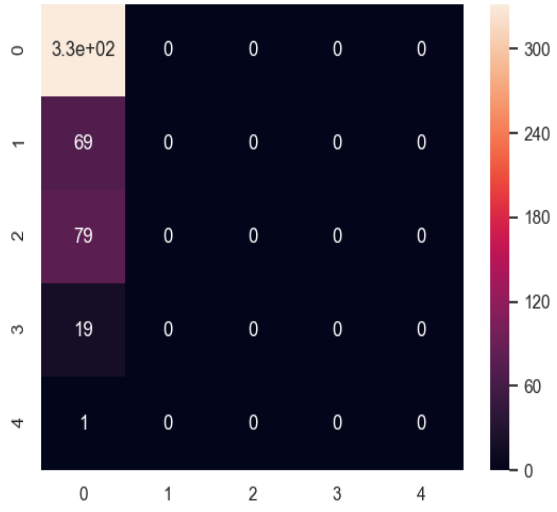

Figure 5: Confusion Matrix for Linear SVM

class	accuracy	precision	recall	F-score	support
0	0.656	0.66	1.00	0.80	332
1	0.00	0.00	0.00	0.00	69
2	0.00	0.00	0.00	0.00	79
3	0.00	0.00	0.00	0.00	19
4	0.00	0.00	0.00	0.00	1

Table 6: Report on Linear SVM

Kernel SVM is like linear SVM, the same problem of classifying higher than the truth.

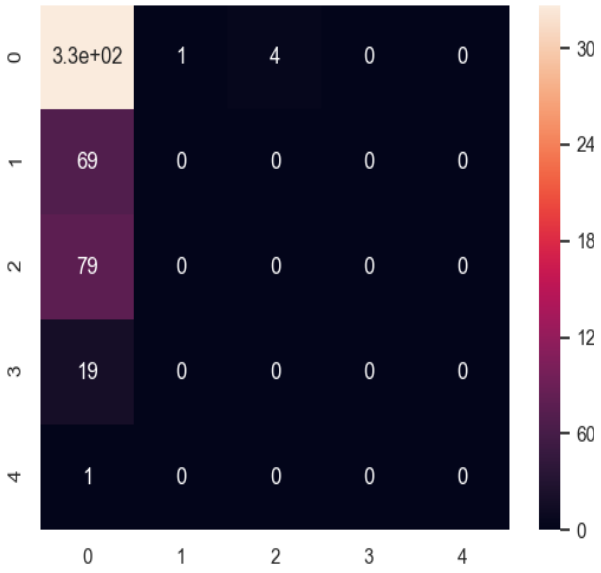

Figure 6: Confusion Matrix for RBF Kernel SVM

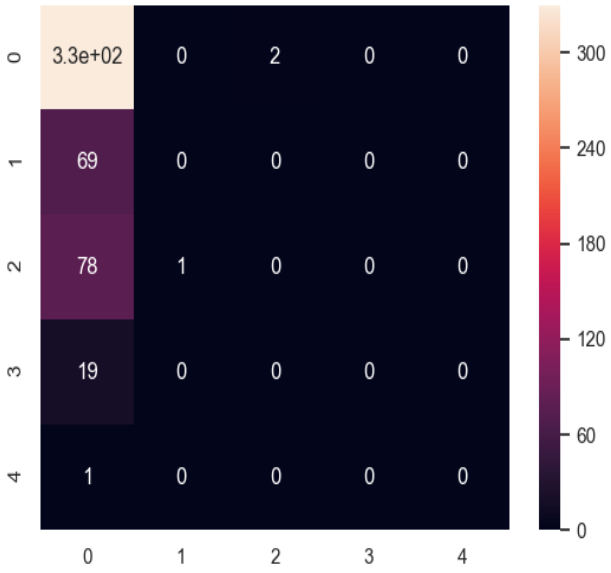

Figure 7: Confusion Matrix for Neural Networks

Neural networks also has the same issue as kernel SVM and linear SVM, only differ slightly in classifying class 2 a class 0.
Overall, linear regression and logistic regression are better at classifying middle class of range of revenues. Linear SVM, RBF kernel SVM and neural networks have desirable performance for class 0, the lowest revenues, but do not have any use for the rest of the classes.
We can exclude the discrepancy due to inflation rate since the budges and revenues are already adjusted according to the rate. The causes for this problem are firstly that the dataset does not have enough movies with high budget and high revenue to train the models, and secondly I only made use of numeric features while text features and content features can have huge effects.

class	accuracy	precision	recall	F-score	support
0	0.654	0.66	0.98	0.79	332
1	0	0.00	0.00	0.00	69
2	0	0.00	0.00	0.00	79
3	0	0.00	0.00	0.00	19
4	0	0.00	0.00	0.00	1

Table 7: Report on RBF Kernel SVM

class	accuracy	precision	recall	F-score	support
0	0.632	0.66	0.99	0.80	332
1	0	0.00	0.00	0.00	69
2	0	0.00	0.00	0.00	79
3	0	0.00	0.00	0.00	19
4	0	0.00	0.00	0.00	1

Table 8: Report on Neural Networks

5 Conclusions and Future Work

Predicting and classifying movies by revenue and ratings has been popular as a research topic in data science, many promising results have been achieved. But the test error can be sensitive to misleading features [6]. This can be solved by sequential forward selection as stated in [6]. The first thing I could do to improve the performance is to make adjustments of the data. If more movies with high budgets and revenues can be collected, a normalization is feasible.

But for the current dataset, normalization

will only help if a transformation of the data can be made, for example, log function, to reduce the imbalance. Then of course the feature set is a big problem. I should include text-based features, if too hard to do it by scraping, then should find a dataset that have them prepared. Training purely based on numeric feature is impossible to get high quality models.

References

[1] V R, N. (2017). Predicting Movie Success Based On Imdb Data. *International Journal for Research in Applied Science and Engineering Technology*, [online] V(X), pp.504-507. Available at: https://www.researchgate.net/publication/265407660_Predicting_Movie_Revenue_from_IMDb_Data.

[2] Yoo, Steven; Kanter, Robert; Cummings, David; Maas, Andrew. (2019). Predicting Movie Revenue from IMDb Data.

[3] Abidin, D.; Bostanci, C.; Site, A. In *Movie Rating Prediction with Machine Learning Algorithms on IMDB Data Set*; Safranbolu, 2018; pp 231–235.

[4] Marovic, Mladen; Mihokovic, Marko; Miksa, Mladen; Pribil, Sinisa; Tus, Alan. (2011). Automatic movie ratings prediction using machine learning.. 1640-1645.

[5] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[6] Armstrong, Nick; Yoon,. (2008). Movie Rating Prediction.