NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING

Predicting Box Office Success Using Machine Learning Algorithms

by
Sally Kim and Anthony Lam

Introduction to machine learning
and statistical pattern recognition

Spring 2020

Chapter 1. Introduction

For the past few decades, the movie market has been growing larger each year. The film industry generates approximately billions of dollars of revenue annually. The question of what makes a film successful has been asked for over the years. The term “success” in this study will be defined as “breaking even”, where the revenue gained from the movie’s release is equivalent or greater than twice the cost of production (a general rule of thumb of measuring the break-even point). Some questions we ought to answer will include “what qualities does a successful movie have?” and “Does having X factor guarantee the success of the movie?”

Chapter 2. Exploratory Data Analysis

2.1 Feature Analysis

In this study, the publicly available data from Kaggle was used. The data can be accessed from Kaggle.com. The data covers approximately 3000 movies obtained from the Movie Database (TMDB). Data points include cast, crew, plot keywords, budget, posters, release dates, languages, production companies, and countries. Certain data points were omitted during feature transformation and only the features described below in detail were included in our dataset. Table 2.1.1 demonstrates the first 5 rows of the dataset.

	id	belongs_to_collection	budget	genres	original_language	popularity	production_companies	release_date	runtime	title	Keywords	cast	revenue
0	1	[{'id': 313576, 'name': 'Hot Tub Time Machine ...	14000000	[{'id': 35, 'name': 'Comedy'}]	en	6.575393	[{'name': 'Paramount Pictures', 'id': 4}, {'na...	2/20/15	93	Hot Tub Time Machine 2	[{'id': 4379, 'name': 'time travel'}, {'id': 9...	[{'cast_id': 4, 'character': 'Lou', 'credit_id...	12314651
1	2	[{'id': 107674, 'name': 'The Princess Diaries ...	40000000	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	en	8.248895	[{'name': 'Walt Disney Pictures', 'id': 2}]	8/6/04	113	The Princess Diaries 2: Royal Engagement	[{'id': 2505, 'name': 'coronation'}, {'id': 42...	[{'cast_id': 1, 'character': 'Mia Thermopolis'...	95149435
2	3	NaN	3300000	[{'id': 18, 'name': 'Drama'}]	en	64.299990	[{'name': 'Bold Films', 'id': 2266}, {'name': ...	10/10/14	105	Whiplash	[{'id': 1416, 'name': 'jazz'}, {'id': 1523, 'n...	[{'cast_id': 5, 'character': 'Andrew Neimann',...	13092000
3	4	NaN	1200000	[{'id': 53, 'name': 'Thriller'}, {'id': 18, 'n...	hi	3.174936	NaN	3/9/12	122	Kahaani	[{'id': 10092, 'name': 'mystery'}, {'id': 1054...	[{'cast_id': 1, 'character': 'Vidya Bagchi', '...	16000000
4	5	NaN	0	[{'id': 28, 'name': 'Action'}, {'id': 53, 'nam...	ko	1.148070	NaN	2/5/09	118	Marine Boy	NaN	[{'cast_id': 3, 'character': 'Chun-soo', 'cred...	3923970

Table 2.11

Below is a short description of each feature in the data set:

- Belongs_to_collection: Whether the film is part of a series or not. Contains string if yes, NaN if not.
- Budget: Cost (in dollars) of creating the film. Total of 811 data points has 0 as their budget.
- Genres: Includes all the genres that represent the film. Genre types include: Comedy, Drama, Animation, and Thriller etc. There’s also a designated unique id for each genre, represented in a positive integer.
    - [{'id': 16, 'name': 'Animation'}, {'id': 12, 'name': 'Adventure'}, {'id': 10751, 'name': 'Family'}]
- Original_language: The primary language that’s used in the film.

- Popularity: Ranges A float number representing the popularity of the film. Ranges from 0 to 300
- Production_companies: A string including all the names of companies that contributed in producing the film. A unique id is also designated to each company.
    - [{'name': 'Columbia Pictures', 'id': 5}, {'name': 'Rastar Productions', 'id': 13945}]
- Release_date: The Month, Day, and Year of when the film was released
    - 11/10/1989
- Runtime: The total runtime of the film in minutes.
- Cast: Includes the cast id, character name, credit id, gender, and name id and real name of all the actors and actresses that were casted in the film.
    - [{'cast_id': 1, 'character': 'Mia Thermopolis', 'credit_id': '52fe43fe9251416c7502561f', 'gender': 1, 'id': 1813, 'name': 'Anne Hathaway', 'order': 0, 'profile_path': '/jUMOKwSUBnTcMeN1HfhutiY49Ad.jpg'}]
- Revenue: The total revenue of the film

2.2 Correlation Check

- Revenue vs Budget

We can see that there's somewhat a linear correlation between revenue and budget (refer to Figure 2.2.1). This is more evident when we see the correlation between revenue and 2 times the budget. In Figure 2.2.2, we have categorized films either to success (label green) or failure (label red) based on the idea a film will be considered successful if it exceeds two times its budget, unsuccessful if not.

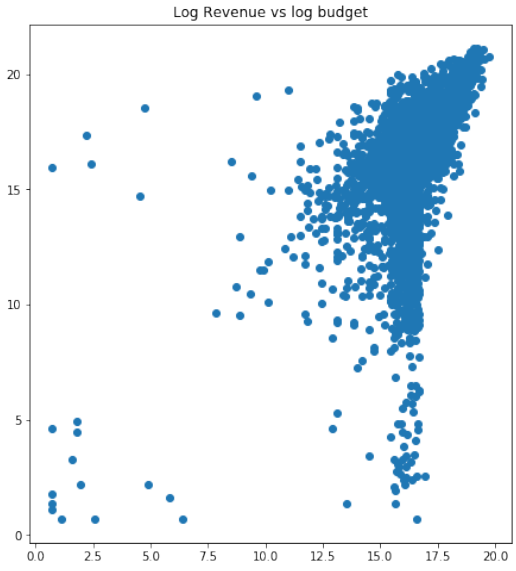

Figure 2.2.1

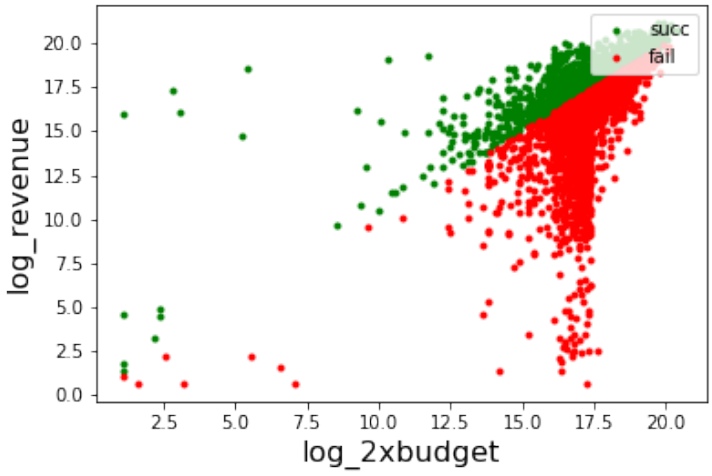

Figure 2.2.2

- Revenue vs Popularity and Revenue vs Runtime

In Figure 2.2.3 and 2.2.4, we've also plotted to see if there's a correlation between revenue and popularity. Almost all of the films fall in the popularity range of 0-25 and doesn't seem there's much correlation between these features. Likewise, because most typical films have 1.5-2 hour runtime for their movie, we can see in Figure 2.2.5 that many of both "successful" and "failed" movies fall in this slot. But it's useful to note that the films with highest revenue are also in this range.

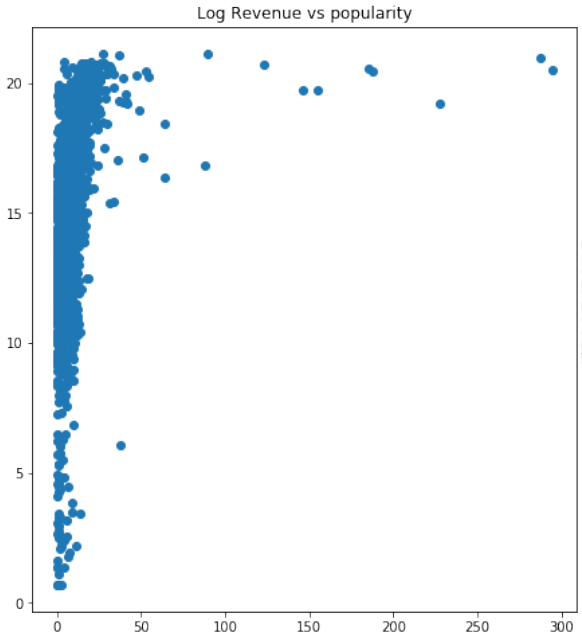

Figure 2.2.3

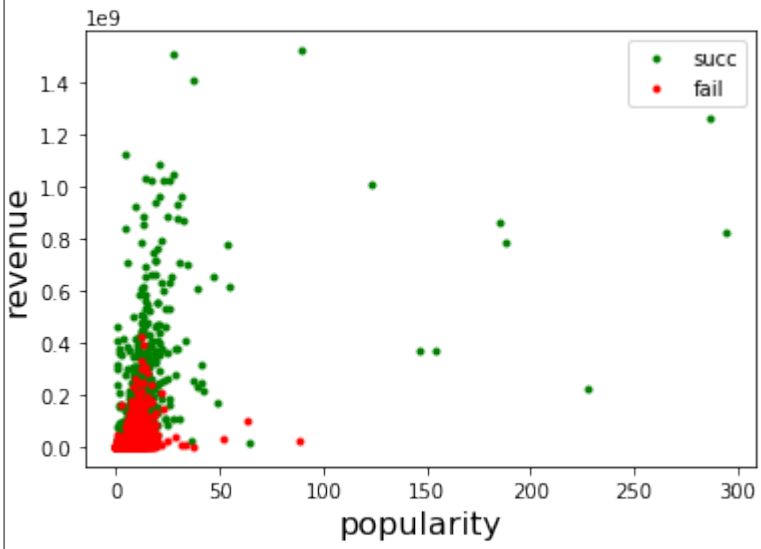

Figure 2.2.4

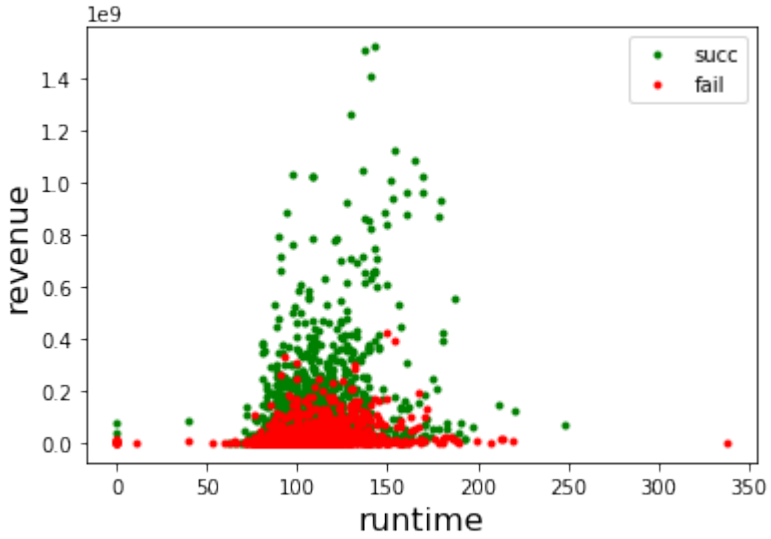

Figure 2.2.5

Here, in Figure 2.2.6, we've taken the mean value of the revenue for each original language and plotted to see where the movies with highest revenue fall in which language. As there are more English films in our dataset, we see that they have a higher range of values. But not only English movies have the highest revenues, but there are also a few non-Eng. movies that have high revenues.

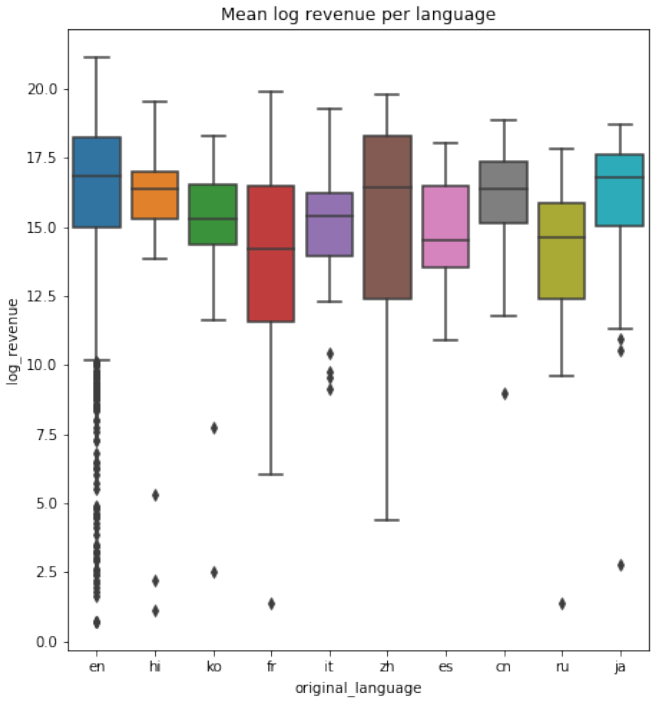

Figure 2.2.6

2.3 Feature Transformation

Within our data set, we obtained a variety of features that each movie had categorized. However, most of the data was unusable in its original form, having properties that made it difficult for an algorithm to translate with. The ultimate goal here was to convert these features that were obtained into numerical values that could be used to define each feature to their respective movie.

The first feature that needed transformation was what the baseline for success and failure was based on, budget. In the dataset, there were many movies that had 0 for their budget attribute, meaning any revenue received would put that movie as a success. The idea was to move away from that and rather use the collective data to create a bracket of possible "realistic" budget values and re-assign all movies with a budget of 0 to a randomized value within the bracket. The bracket was determined to be around 5 million to 18 million. Prior to transformation, the dataset had around 800 points of 0 budget movies which would have definitely skewed the algorithm towards a lower budget than normal as a success rating.

The simplest features to transform are ones that can be converted into a binary system. This is what was done with the “Belongs_to_Collection” because it could be identified as either having a sequel, prequel, or be a part of a series or it could be held as a solo movie. For this it was simple as the information was parsed through our program, searching for cells with any sort of value and attributing a 1 if the cell had information and a 0 if it did not.

The next feature that was tackled was release_date. The idea behind this transformation was finding out the impact that the season of release had on the movie. A year was divided into seasons; months 3-5 as spring, 6-8 as summer, 9-11 as winter, and 12-2 as winter. From here, it was easy to parse the information as the release data was in a mm/dd/yyyy format, meaning only the first part needed to be parsed and then the feature details could be redefined, seasonally.

After these first two transformations, it became a bit more difficult as the data showed the features: genre, production_companies, and cast to be a bit more information that needed to be parsed. In addition, the way that the dataframe was parsed made it a little more difficult as it was parsed as a string rather than a list format as shown on the data sheet. However, it was still achievable and a baseline for categorizing the features was made.

For genres, the idea was to calculate the total number of iterations that each genre appeared in the entirety of the datasheet and calculating the percentage of “volume” that each genre took up. This would resemble a percentage value of occurrences of which was later used to attribute to each movie’s genre feature. For movies with multiple genres, the average “volume” was taken where the sum of all the percentage of occurrences of genres was taken and divided by the number of genres that each movie had, similarly to the example.

```
[{'id': 35, 'name': 'Comedy'}]
[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'name': 'Drama'}, {'id': 10751, 'name': 'Family'}, {'id': 10749, 'name': 'Romanc
e'}]
[{'id': 18, 'name': 'Drama'}]
```

```
13.7
11.3
20.4
```

The next feature was production_companies. The idea behind this transformation was based on equating the number of production companies a movie had to its success level under the belief that a movie that had more production companies meant it would have a higher backing and a higher range of credibility in its success. So the general approach here was to count the number of production companies a movie had in its production and use that value as its definition of the feature. So a movie with 5 production companies backing it would have a feature value of 5.

The most difficult feature transformation was the cast feature. The idea was relatively simple and similar to genre, however, the execution was very poor and difficult to parse quickly. The idea was to put each distinct actor into their own dictionary key and keep track of their appearances throughout the entirety of the datasheet. This was to gauge their “level of power” based on how many movies they premiered in. The more movies they appeared in, the more points awarded to them. From here, the movies had their points tallied based on their actors and the appearance count, leading to a numerical redefinition of the feature from the original string type.

Example:
[{'cast_id': 4, 'character': 'Lou', 'credit_id': '52fe4ee7c3a36847f82afae7', 'gender': 2, 'id': 52997, 'name': 'Rob Corddry', 'order': 0, 'profile_path': '/k2zJL0V1nEZuFT08xUdOd3ucfXz.jpg'} . . . {'cast_id': 66, 'character': 'Bridesmaid', 'credit_id': '59ac02cd925141079d02b1b4', 'gender': 1, 'id': 129714, 'name': 'Kisha Sierra', 'order': 24, 'profile_path': ''}] = 48.0

Chapter 3. Modeling

The original approach for this topic was to use linear regression to find a pattern within the features analyzed and gain some sort of idea which features had a direct impact on the success rating of the movie. This would be found by first contemplating a baseline for an acceptable success level which is understandably at 50% because it would make sense for an algorithm to have a guessing accuracy greater than half of the time. Additionally, it was planned to include some form of linear classifiers to help sort out and identify successful movies from ones that were not. However, data plotting showed trends that wouldn't fit a linear regression, but rather a logistic one. So the next approach was to apply a logistic regression model on the features to determine which group of features were best suited for the model.

3.1 Logistic Regression

The second approach, logistic regression proved to show some accuracy ratings that were sufficient enough to say they "worked". This was done first by running each individual feature on the algorithm to determine its solo accuracy rating through a train-test split of 80-20. From the table, the logistic regression accuracy ratings can be seen for each individual feature

Feature	**Accuracy Rating**
Budget	0.575
Belongs_to_Collection	0.6583
Genre	0.572
Popularity	0.597
Release_Date	0.575
Revenue	0.425
Cast	0.673

As it can be seen, most of the features accurately predict above a 50% rating, up to a 67.3% rating. The feature, revenue, was interestingly only at a 42.5% accuracy rating even though it was our baseline for determining whether or not the movie was considered successful or not. This might be attributed to the randomness that was applied to the budget feature as the revenue for each randomized budget was not adjusted, adding on another layer of uncertainty since the revenue had to be 2x the budget to be considered successful. For the other individual ones, they passed the baseline test of being at least 50% accurate. However, the idea was to find a group of features that would have an impact in predicting box-office success. The initial decision to apply all of the features seen in the table above with the same 80-20 split in a multivariable logistic regression model actually proved to be detrimental with a resulting accuracy score below 50%. We also tried testing combinations with the most features included and with the best bet at accurately predicting successful movies was between the features: "Belongs_to_Collection", "Genre", "Popularity", Release_Date", and "Production_Companies". With this, the accuracy rating of 58% was achieved. Although it was not higher than some of the individual features, like the cast feature with the highest accuracy rating, the goal of this test was to find out a combination of features best suited to help the logistic regression algorithm predict box-office success.

In addition to this accuracy test, recall and precision tests were run on the same dataset, resulting in a recall rating of 79.6% and a precision rating of only 40.8%. As it can be seen, the precision rating is quite low which is why the approach shifted from just using logistic regression to also sampling between SVM and RBF model analysis to see if a different model would result in higher ratings.

3.2 Support Vector Machines

SVMs are another algorithm for finding linear classifiers which is as popular as logistic regression. First, we tried running the sklearn's SVM classifier (called SVC) with a linear kernel. The accuracy result came out to be the exact same value as that of Logistic Regression, and this is expected because our data is linearly separable. We then tried k-fold cross validation on the same linear kernel model. Since our data points (roughly 3000) aren't that many, we predicted using cross validation should give us better accuracy results than when we're using train-test split. And this prediction was correct. Having 3 K folds, we've got [0.66708385, 0.65957447, 0.67293233] as our accuracy score.

Furthermore, we also tried to see how our SVM model performed using the RBF kernel. We used the same k-fold cross validation here also, and we see that the RBF kernel provides more accurate results: [0.6795994993742178, 0.6770963704630788, 0.6879699248120301].

Chapter 4. Conclusion

In this exploration of data, there were quite a few challenges and improvements that could have been made. Some challenges included the difficult parsing and manual editing of the

datasheet so that it could actually be parsed properly as well as the “useless” features that were included. With more time, it would mean more opportunity to find ways in utilizing these “useless” features in some way that could actually prove to help reanalyze the logistic regression and SVM models. Additionally, we could have sought for different datasets that still contained most of the information we needed, but also some newer features that we would not have thought to use in this exploration. On model usage, with more time means more exploration with the different types of models that are available rather than having to stick with a logistic regression (which makes the most sense), we could try applying a different model that we wouldn’t originally about and transform our data to fit that. In a sense, moving our data to conform to a different model than what we were originally doing, conforming a model to our data. It was relatively easy to determine a well-fitting model that would make sense with our data, but maybe spending more time thinking outside of the box and applying a model like a neural network to our data.

Trying to answer the questions we sought to answer through our models, we came to a conclusion that the ‘X factor’ that has the most effect on making the film successful is the cast feature. We can also say that if the movie belongs to a collection, it ought to have a higher chance of gaining box office success (refer to Table 3.1). In other words, moviegoers would be more likely to watch the movie in a theater if the film is casted with popular actors/actresses. Also, if the new movie is part of a series, people tend to have more fondness and positive expectations about the movie that’s coming out.