Group Project Report

Successful Movies Prediction

University of Science and Technology of Hanoi

Group 6

Le Anh Tu | BA9-067

Luong Nguyen Viet Son | BI10-156

Nguyen Tu Tung | BI10-187

Tran Hoang Minh | BI10-119

Nguyen Tien Dat | BI10-028

Tran Bao Huy | BI10-079

Pham Hoang Viet | BI10-192

I. Introduction

In this project, we'll try to forecast whether or not a film will be successful. The amount of awards, overall income,movie's rating, and box office can all be used to determine a film's success. However, here we use the rule that a film is profitable if the revenue exceeds the budget. As a result, success will be a boolean feature that holds true if the film's revenue exceeds its budget by twice and it also has the rating score bigger than 5. We'll use two classification methods to try to anticipate this value: regression logistic, K-nearest neighbors.

II. Data Introduction

This is the movies data, which has about 5000 data, consist 12 column:

- id – unique movie ID for TMDB
- title : title of a movie
- budget – rounded to nearest dollar
- revenue – rounded to nearest dollar
- runtime – rounded to nearest minute
- vote_average – average user ratings on a scale of 1-10
- vote_count – number of voters
- Genres : type of the movie
- Countries: Production country
- certification_US – movie rating that determines suitability by viewer age (G, PG, PG-13, R or NC-17)
- year: release year of the movie.
- Success: success = 1 .0, not success = 0.

Intro of dataset:

```
In [55]: ds.head()
```

Out[55]:

	id	title	budget	revenue	runtime	year	genre	certification_US	vote_average	vote_count	country	success
2563	151639	Naran	680000	1800000	165.0	2005.0	Drama	NR	6.1	6	India	1.0
2590	69407	Varalaru	500000	500000	157.0	2006.0	Drama	NR	6.1	6	India	0.0
2651	39170	Prince Vladimir	5000000	5399340	78.0	2006.0	Animation	NR	4.4	6	Russia	0.0
2879	76839	Chennai 60028	470000	3800000	141.0	2007.0	Comedy	NR	6.6	6	India	1.0
2888	46301	Disappearances	1500000	312642	118.0	2007.0	Action	PG-13	4.1	6	United States of America	0.0

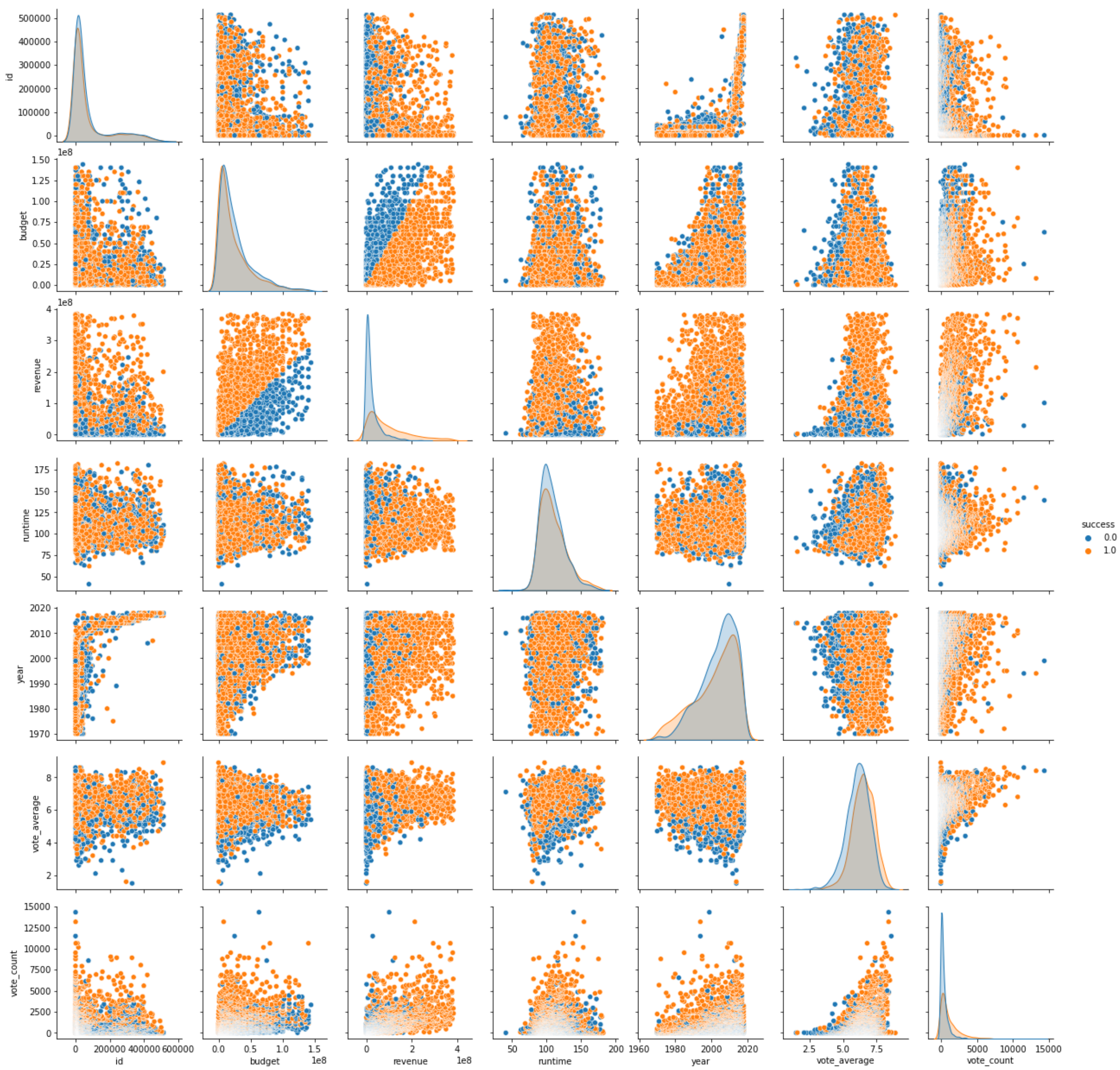
Additional info: The vote count must > 5

III. Preprocessing

This dataset has everything we need so we don't have to deal too much with it, we just add a success column to show if the movie is successful or not. I assume that, the movie success if its revenue>= 2* budget and it also has a vote score > 5 . Also, we encoding three columns: Genres, countries, and certification_US; transform them into numbers.

IV. Statistic analysis of data

Below is a scatterplot matrix showing the relationships between the continuous features



As we can see, the revenue and budget plots are biased to the left, indicating that the dataset is primarily made up of low-budget and low-revenue films. Year plots are shifted to the right, so we can see that this dataset is most from the latest year. We can also see that in the budget*year and revenue*years are also shifted to the right, which means with the increasing of year, budget and revenue also increase. That also means that after some time, the budget will increase and so will revenue. We also can see in the revenue*vote_average, it shifted to the right. We can say that the higher the score the film gets, the more revenue the company can get.

V. Identify learning task

To predict a movie's success, we use two classification algorithms : logistic regression, K-nearest neighbors. Each classification algorithm will be using the same target value and predictors. The target value is success, a boolean value that is true if the revenue exceeds twice the budget and also has a vote score > 5 . The predictors are listed below:

budget – total money required to produce movie

revenue – total money obtain from a movie

runtime – total movie time in minutes(additional)

year – release year

vote_average – mean community score

vote_count – number of votes attributing to vote_average

genre – most significant genre

country – most significant production country

certification_US – movie rating that determines suitability by viewer age

VI. Classification model

Overall:

We split the dataset into train set and test set.(80% train, 20% test)

A. Logistic regression

Logistic regression is promising because it works best when the target variable is a boolean value, and our target variable, success, is boolean. In this project, we are using the sklearn library to deal with this data. We assume that solver= 'lbfgs'.

This is the testing accuracy of Logistic regression:

```
In [46]: print(scores)

         0.9600389863547758
```

This is the testing result when we have testing input:

```
In [50]:
dtf= pd.read_csv('tstset.csv')

x_test = dtf.drop(['id','title'],axis=1)
genre = LabelEncoder()
x_test['genre1'] = genre.fit_transform(x_test['genre'])
x_test = x_test.drop(['genre'],axis=1)

country = LabelEncoder()
x_test['country1'] = country.fit_transform(x_test['country'])
x_test = x_test.drop(['country'],axis=1)

certification_US = LabelEncoder()
x_test['certification_US1'] = certification_US.fit_transform(x_test['certification_US'])
x_test = x_test.drop(['certification_US'],axis=1)
x_test
```

Out[50]:

	budget	revenue	runtime	year	vote_average	vote_count	genre1	country1	certification_US1
0	3598902	4001919	108	2018	6.7	137	0	0	1
1	10000000	28646544	84	2018	7.0	139	3	2	0
2	24000000	17506878	124	2018	6.5	140	2	2	1
3	20000000	87054892	117	2018	6.5	363	5	2	2
4	8000000	116470	118	2018	5.9	16	1	1	2
5	5000000	470901	100	2018	5.1	30	1	1	2
6	5000000	10000000	99	2018	4.2	25	4	1	2

```
In [51]: y_predict = model.predict(x_test)
         result = pd.concat([dtf, pd.DataFrame(y_predict)], axis = 1)
         result.columns.values[-1:] = ['success']
```

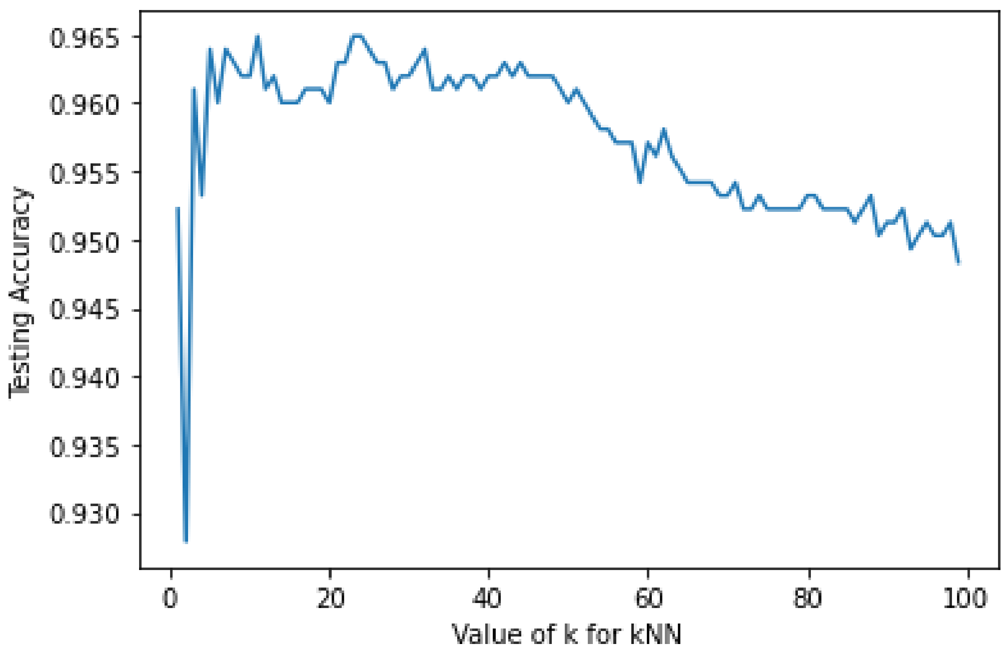
In [52]: `result`

Out[52]:

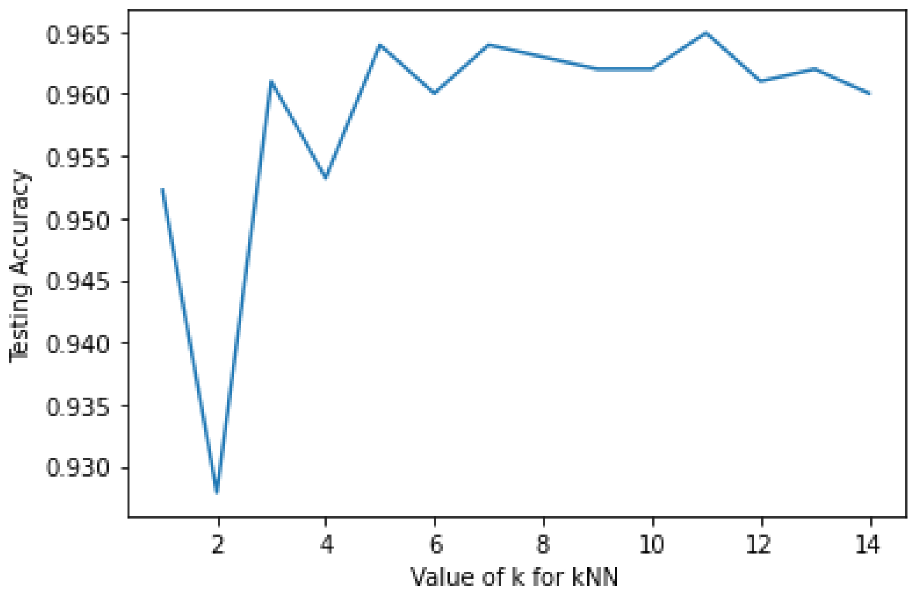
	id	title	budget	revenue	runtime	year	genre	certification_US	vote_average	vote_count	country	success
0	1	Bleach	3598902	4001919	108	2018	Action	PG-13	6.7	137	Japan	0.0
1	2	Teen Titans Go! To the Movies	10000000	28646544	84	2018	Family	PG	7.0	139	United States of America	1.0
2	3	Operation Finale	24000000	17506878	124	2018	Drama	PG-13	6.5	140	United States of America	0.0
3	4	A Simple Favor	20000000	87054892	117	2018	Thriller	R	6.5	363	United States of America	1.0
4	5	London Fields	8000000	116470	118	2018	Crime	R	5.9	16	United Kingdom	0.0
5	6	Mara	5000000	470901	100	2018	Crime	R	5.1	30	United Kingdom	0.0
6	7	Random1	5000000	10000000	99	2018	Random	R	4.2	25	United Kingdom	0.0

B. K-nearest neighbor

This is the testing accuracy when testing a success movie when k in range 100



This is the testing accuracy when testing a success movie when k in range 10



Its max accuracy is: 0.9649122807017544 at k = 11.

As we can see from the chart, the accuracy will deeply decrease at the even values of k(k=2,4,6,8…). So the kNN algorithm will work best with this data when k is odd.

This is the predict result when we input dataset

In [97]:
```
dtf= pd.read_csv('tstset.csv')

x_test = dtf.drop(['id','title'],axis=1)
genre = LabelEncoder()
x_test['genre1'] = genre.fit_transform(x_test['genre'])
x_test = x_test.drop(['genre'],axis=1)

country = LabelEncoder()
x_test['country1'] = country.fit_transform(x_test['country'])
x_test = x_test.drop(['country'],axis=1)

certification_US = LabelEncoder()
x_test['certification_US1'] = certification_US.fit_transform(x_test['certification_US'])
x_test = x_test.drop(['certification_US'],axis=1)
x_test
```

Out[97]:

	budget	revenue	runtime	year	vote_average	vote_count	genre1	country1	certification_US1
0	3598902	4001919	108	2018	6.7	137	0	0	1
1	10000000	28646544	84	2018	7.0	139	3	2	0
2	24000000	17506878	124	2018	6.5	140	2	2	1
3	20000000	87054892	117	2018	6.5	363	5	2	2
4	8000000	116470	118	2018	5.9	16	1	1	2
5	5000000	470901	100	2018	5.1	30	1	1	2
6	5000000	10000000	99	2018	4.2	25	4	1	2

In [98]:
```
y_predict = knn.predict(x_test)
result = pd.concat([dtf, pd.DataFrame(y_predict)], axis = 1)
result.columns.values[-1:] = ['success']
```

In [98]:
```
y_predict = knn.predict(x_test)
result = pd.concat([dtf, pd.DataFrame(y_predict)], axis = 1)
result.columns.values[-1:] = ['success']
```

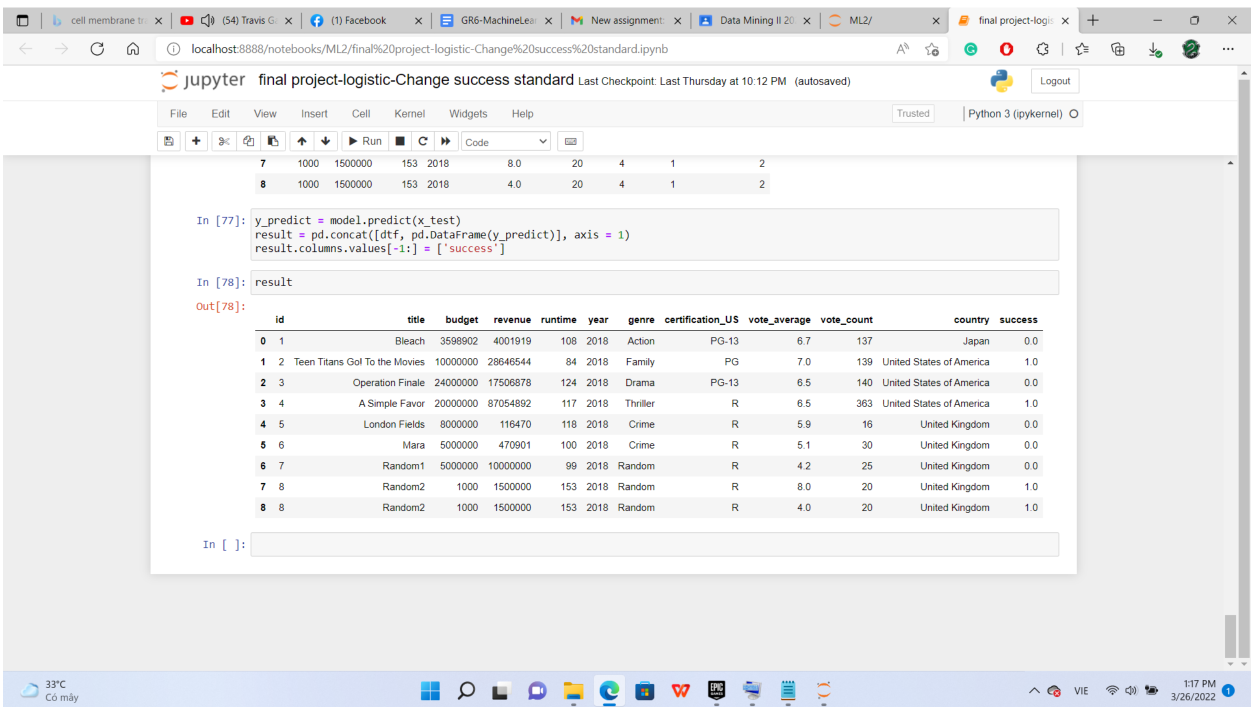
In [99]:
```
result
```

Out[99]:

	id	title	budget	revenue	runtime	year	genre	certification_US	vote_average	vote_count	country	success
0	1	Bleach	3598902	4001919	108	2018	Action	PG-13	6.7	137	Japan	0.0
1	2	Teen Titans Go! To the Movies	10000000	28646544	84	2018	Family	PG	7.0	139	United States of America	1.0
2	3	Operation Finale	24000000	17506878	124	2018	Drama	PG-13	6.5	140	United States of America	0.0
3	4	A Simple Favor	20000000	87054892	117	2018	Thriller	R	6.5	363	United States of America	1.0
4	5	London Fields	8000000	116470	118	2018	Crime	R	5.9	16	United Kingdom	0.0
5	6	Mara	5000000	470901	100	2018	Crime	R	5.1	30	United Kingdom	0.0
6	7	Random1	5000000	10000000	99	2018	Random	R	4.2	25	United Kingdom	0.0

As we can see in the results of 2 algorithms, only 2 movies are 'successful' and we can clearly see that those film budgets are <= revenue/2. So this is a correct prediction.

C. Wrong prediction:


jupyter final project-logistic-Change success standard Last Checkpoint: Last Thursday at 10:12 PM (autosaved)

7	1000	1500000	153	2018	8.0	20	4	1	2
8	1000	1500000	153	2018	4.0	20	4	1	2

In [77]:
```
y_predict = model.predict(x_test)
result = pd.concat([dtf, pd.DataFrame(y_predict)], axis = 1)
result.columns.values[-1:] = ['success']
```

In [78]:
```
result
```

Out[78]:

	id	title	budget	revenue	runtime	year	genre	certification_US	vote_average	vote_count	country	success
0	1	Bleach	3598902	4001919	108	2018	Action	PG-13	6.7	137	Japan	0.0
1	2	Teen Titans Go! To the Movies	10000000	28646544	84	2018	Family	PG	7.0	139	United States of America	1.0
2	3	Operation Finale	24000000	17506878	124	2018	Drama	PG-13	6.5	140	United States of America	0.0
3	4	A Simple Favor	20000000	87054892	117	2018	Thriller	R	6.5	363	United States of America	1.0
4	5	London Fields	8000000	116470	118	2018	Crime	R	5.9	16	United Kingdom	0.0
5	6	Mara	5000000	470901	100	2018	Crime	R	5.1	30	United Kingdom	0.0
6	7	Random1	5000000	10000000	99	2018	Random	R	4.2	25	United Kingdom	0.0
7	8	Random2	1000	1500000	153	2018	Random	R	8.0	20	United Kingdom	1.0
8	8	Random2	1000	1500000	153	2018	Random	R	4.0	20	United Kingdom	1.0

In [ ]:

Although our model have high accuracy, but it will have some wrong prediction, as we can see in a picture, there is some mistake in predicting a movie which has revenue>2*budgets but its vote average is<5 but the computer still sees it as a successful movie. So we still have to update our model to minimize the error.

VII. Conclusion

As we can see, the kNN and logistic regression algorithm somehow have decent accuracy and it predicts perfectly. We can use both of them to predict the success of a movie. I believe that, thís model can work well with some movie data like this, but there will be some inaccurate value that will appear when your data have the budgets*2<= revenue but the vote_average <5, it will make some wrong predictions.

VIII. References

Dataset source: Movie-Success-Predictor/moviesDb.csv at master · timothyng-164/Movie-Success-Predictor (github.com)

Source code: nguyentutung/final-dt2 (github.com)