Early Prediction of Movie Box Office Success Using Social Media

Yang Liu, Shuqiong Chen, Jingchen Lan, Jiawei Xue

ABSTRACT

In 2018, the global box office revenue hit 41.7 billion U.S dollars. In U.S, the number amounted to about 11.9 billion U.S. dollars. United States alone released more than 800 movies in 2018. However, only a small number of movies have high return on investment. Most of them have very limit life time and are dethroned as new starlets are not long in coming. For movie investors, they want to know whether their investment will ultimately lead to returns or profits when the movies are officially released. For film studios, they want to optimize their arrangement of movies by predicting the box office. Knowing which movies are most likely to success and which are most likely to fail could benefits investors and film studio a lot. It could also help them to release movie in a most appropriate way.

1. Introduction

We collected the early period review of movies from IMDB and used Natural language processing (NLP) tools to conduct the movie review analysis. Sentiment analysis will be used to find out people's views about films. The definition of a movie success is relative, some movies are called successful based on its worldwide gross income, and some movies may not be outstanding in business part but can be called successful for good review and popularity, while we believe the business values is always the key point for marketers and investors, so we will use return on investment (ROI) as our target variable. A movie that ROI>1 is regarded as a business successful movie, and vice versa.

As the most popular review hubs, we chose IMDB as the mainly resources for our project dataset. The dataset concludes following information: score, review number and review content, user rating, average rating, director, stars, genres, box office, budget, release date. We first built a baseline model based on movie category features, then we applied text analysis to generate more feature from the early review.

We use the largest number of box office a director/actor has earned through his/her career before the predicted movie is released to quantify person's value. Most of

the research work related to the prediction of movie success use social network analysis to analyze the relationship between different person. They also implied sentiment analysis to gain insight from user's review, but they haven't generated related feature from review text to predict the success of movie box office directly. We imply two method to quantify film's and person's value through text analysis. We first used naïve method to count the frequency of person's name in review text, then we use word2vector to expand the clue words from aspect clue words table. We also used review text to predict the user's rating.

2. Data Collection

We collected raw data from IMDB (https://www.imdb.com), including movie information (movie title, movie length, director, etc.) and movie review text data. We crawled the movie information and movie review text data respectively. The crawl strategy is:

Step 1: We first searched the movie in IMDB website and sorted by U.S box office in descending order. Since the page also provides the URL of each movie, we then can go the website that contains basic movie information.

Step 2: We set: (Link to IMDB) as our started page and collect the top 10,000 movies which contain box-office record and url link.

Step 3: In each movie page, we crawled the detail information, including *Title, Genre, Rating, Length, Release_Date, Director, Writer, Star, Cast, U.S box office, World box office, Budget, Review_URL*.

Step 4: Since we defined ROI:

$$ROI = \frac{\text{movie. Box\_Office\_Final} - \text{movie. Budget\_Final}}{\text{movie. Budget\_Final}}$$

as our target value, we only preserved the movies which contain both budget and box office information in crawling text review steps. Considering the fact that some movies is too aged to analyze, we filter the movies which released after the year of 2000.

Step 5: We use the *Review_URL* and get the review page of each movie.

As a result, we get our raw data with:

- 9962 movies in total, including 4432 movies with full box office and budget value which will be used for further classification model;

- 1023762 reviews in total, including 99419 reviews posted before movie release. See full description in Table 1:

Table 1: Data Description

Table Name	Description
Movie Table 1	Contain 9962 Movies information (some of them don't have budget information)
Movie Table 2	Contain 4432 Movies information (valid budget and box office information), and will be used for final model part
Person Table	Contain Director, Cast, Star, Writer information of each movie, use movie ID map to movie table
Review Table 1	In total, we have 1023762 reviews for 4432 movies
Review Table 2	A subset of Review Table 1, 99419 reviews posted before movie release

3. Feature Engineering

After the data collection and processing part, we believe we already get all the data we need to start modeling. But instead of simply using numerical features and tf-idf matrix, we decide to extract as many new features as we can. From 9962 movies' information, we quantify value of directors, writers and stars with their highest box office; From 99419 reviews, we not only analysis the review content but also predict missing rating value using review texts.

3.1 Quantify values of director, writer and star

To quantify the values of the movie crew (directors, writers and stars), we follow the simple intuition: the higher historical box office a person achieved, the greater value s/he is. For example, in the movie *The Dark Knight Rises*, the value of the director Christopher Nolan is the highest box office of all movies he directed before *The Dark Knight Rises*. The final director value of this movie is determined by the best director, in this case, Christopher Nolan, because he is the only one director of this movie.

To get more accurate values, we use the full 9688 movies we crawled, instead of the preprocessed 4432 movies.

3.2 Analyze review content

In the previous work, we calculated some aggregated review features such as review counts and average review length. Here we want to explore more about the contents about the reviews, trying to find out what aspects attract people most for a specific movie.

We first use a naïve method to count the appearance of director/writer/actor in review text. Our idea is: in the early review, if there is high percentage of review that have mentioned the director name, then it will converted to a high confidence that this director has a relatively high influence for this movie. So, we tokenized all the reviews that have posted before movie is released. For each review, if director's first name or last name appeared in the review, then we count 1 for this review. So, for each movie, the director's influence is equal to the number of reviews mentioned director's name over the total number of reviews. For actors, and writer, we also use the same method.

Many researchers have done a lot of work in the area of aspect-based sentiment analysis. We use a clues words table (Table 2) that is summarized by researcher.

Table 2: Clues Words Table

Aspect Category	Aspect Clue words
Acting	Chemistry, performance, Charm, comedian ...
Direction	Director, filmmaker, vision ...
Screenplay	Sequence, script, lines, editing, screenwriting ...
Sound effect and music	Score, music, vocals, audio ...
Story	Mystery, spoof, thriller, twist, shock ...
Visual Effects	Effects, 3d, scenery, photography, camera, cinematography
Film on the whole	Flick, remake, sequel, classic, entertainment ...

Once we have this table, we can use word2vect to expand the clue words. We find out the Top 10 words similar to each word in this table. Then we use the naïve method we have used before, to calculate the influence of each category (*Table 3*).

Table 3:Influence of Each Aspect

Movie_ID	acting_content	direction_content	screenplay_content	sound_content	story_content	visual_content	film_content
tt0499549	0.121951	0.215447	0.447154	0.178862	0.231707	0.829268	0.971545
tt1825683	0.143541	0.133971	0.382775	0.248804	0.110048	0.454545	0.909091
tt4154756	0.094488	0.053543	0.181102	0.059843	0.135433	0.236220	0.856693
tt0369610	0.198068	0.154589	0.516908	0.198068	0.280193	0.608696	0.985507
tt0848228	0.176056	0.227700	0.530516	0.068075	0.192488	0.485915	0.988263
tt3606756	0.036364	0.090909	0.163636	0.127273	0.181818	0.090909	0.872727

3.3 Review Rating Classification

In the review table (Table 4), we also have a rating score columns which represent the user's rating for this movie.

Table 4: Review Table

review_text	rating_score
Considering the fact that Johnny Depp and Ange...	4.0
Imagine a movie, imagine that movie stars two ...	6.0
This ugly little piece of slam dunk marketing ...	2.0
(Synopsis) Elise (Angelina Jolie) randomly sit...	4.0
Considering the previous great movie of this d...	5.0

For each movie, we use the mean rating score of their reviews as their rating. However, some reviews don't have rating score. So, we use the all the review text to predict the null rating score. We convert it into a ten-class classification problem and use LinearSVC classification model to predict the rating score.

4. Modeling

As we mentioned before, our goal is to train a model that can predict a movie will be success or not base on movie information or reviews. As the result of feature engineering, we organized our features into four different sets.

The basic movie features including the movie budget, release day and the person values that we have calculate in the feature engineering part. We will first build a baseline based on the basic movie features using Logistic Regression Algorithm.

4.1 Logistic Regression Model Performance

Logistic Regression is an extension of linear regression to predict qualitative response for an observation. It defines the probability of an observation belonging to a category or group.

4.1.1 Basic Movie Features Prediction

We use basic movie features to predict the target value, we split the data with 70% training set and 30% test set (always this way if not mention), the result of prediction for test set was shown in Figure 1:

```
classification_evaluation(LogisticRegression(), cols=original_cols)
```

```
LogisticRegression
Accuracy: 0.6460843373493976
              precision    recall  f1-score   support

       False       0.67      0.84      0.75       829
        True       0.55      0.32      0.41       499

   micro avg       0.65      0.65      0.65      1328
   macro avg       0.61      0.58      0.58      1328
weighted avg       0.63      0.65      0.62      1328
```

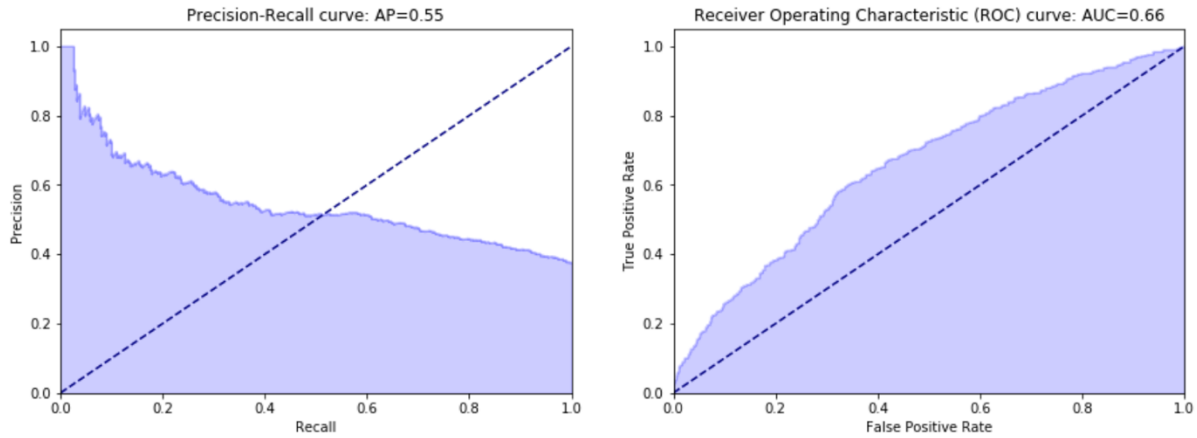

Figure 1: Result of Basic Features for Logistic Regression

The accuracy is 64.6% and from the ROC curve we can get the Area Under Curve (AUC) is 66% which is definitely not an ideal result. So we tried to add more features in to the model to see if it can be improved.

4.1.2 Naïve Method Features Prediction

Beside basic movie features in the first step, we added *review count, average review length, average rating* and 3 features generate using naïve method (count of the appearance of director/writer/actor in review text):

```
classification_evaluation(LogisticRegression(), cols=text_naive_cols)
```

```
LogisticRegression
Accuracy: 0.6890060240963856
              precision    recall  f1-score   support

       False       0.72      0.83      0.77       829
        True       0.62      0.45      0.52       499

   micro avg       0.69      0.69      0.69      1328
   macro avg       0.67      0.64      0.65      1328
weighted avg       0.68      0.69      0.68      1328
```

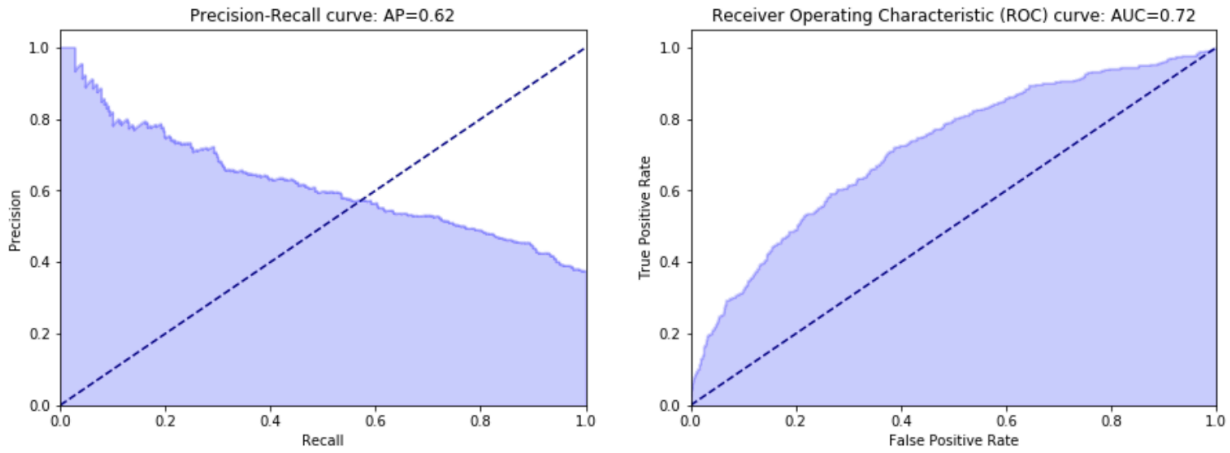

Figure 2: Result of Naïve Bayes features for Logistic Regression

Accuracy improve to 68.9%, in the meantime, AUC improve to 72%.

4.1.3 Word2vector Features Prediction

Instead of using characters' count of appearance, we use features that represent the influence of each movie aspect (which was shown in Table 3):

```
classification_evaluation(LogisticRegression(), cols=text_w2c_cols)
```

```
LogisticRegression
Accuracy: 0.6965361445783133
              precision    recall  f1-score   support

       False       0.73      0.83      0.77       829
        True       0.62      0.48      0.54       499

   micro avg       0.70      0.70      0.70      1328
   macro avg       0.68      0.65      0.66      1328
weighted avg       0.69      0.70      0.69      1328
```

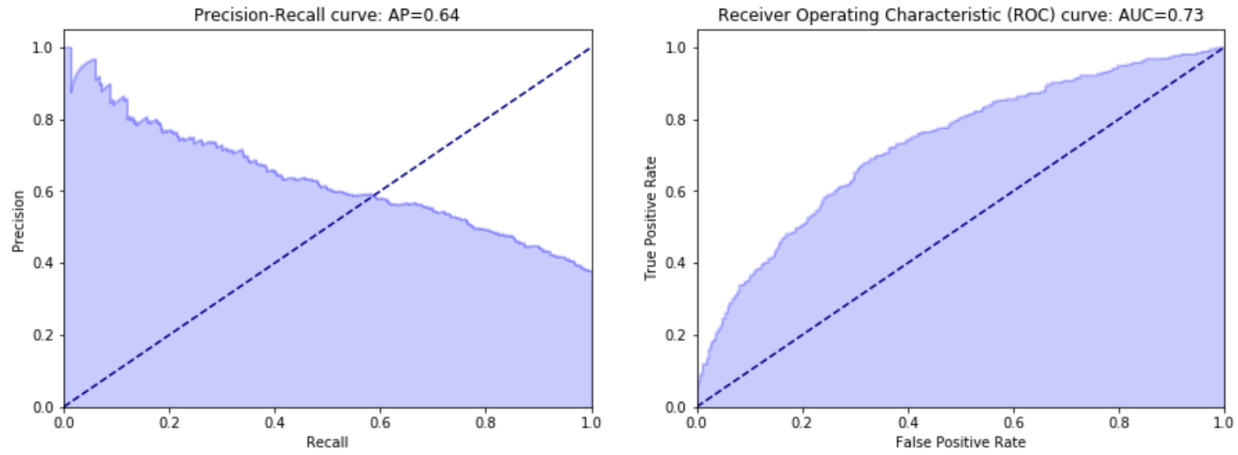

Figure 3: Result of Word2vector features for Logistic Regression

In Figure 3 we can see the model was improved once again:

Accuracy = 69.7%; AUC = 73%

4.1.4 Prediction with Predicted Review Rating

Base on the accomplishment in 4.1.3, we add predicted review rating we generate in 3.3 to see if this feature can improve the model or not:

```
classification_evaluation(LogisticRegression(), cols=text_predicted_rating_cols)
```

```
LogisticRegression
Accuracy: 0.6980421686746988
              precision    recall  f1-score   support

       False       0.73      0.83      0.77       829
        True       0.63      0.48      0.55       499

   micro avg       0.70      0.70      0.70      1328
   macro avg       0.68      0.66      0.66      1328
weighted avg       0.69      0.70      0.69      1328
```

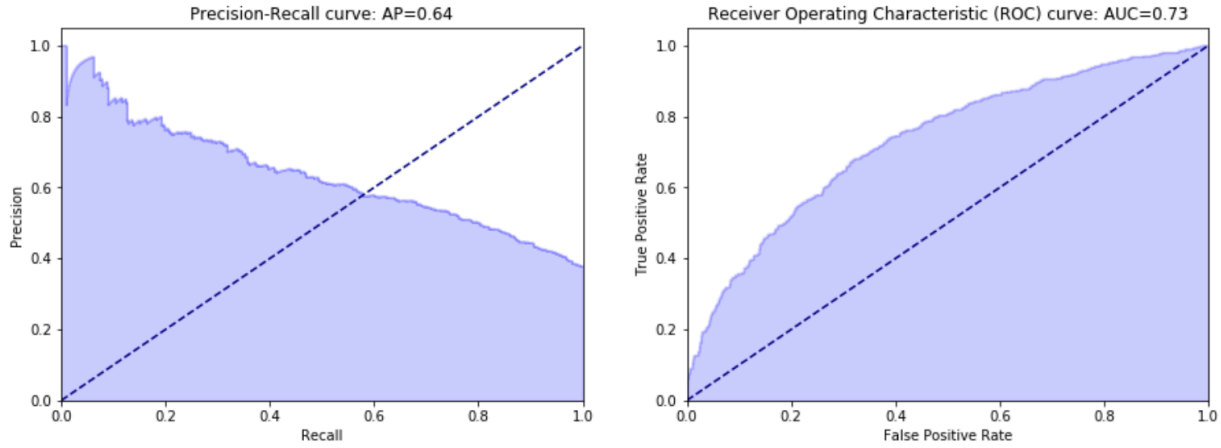

Figure 4: Result of prediced rating feature for Logistic Regression

From Figure 4 we know: the Accuracy or AUC barely didn't improve.

4.2 Random Forest Model Performance

Random Forest(RF) is a supervised learning algorithm. Like you can already see from its name, it creates a forest and makes it somehow random. The forest it builds is an ensemble of Decision Trees. Here we use RF as our second model. And the modeling steps are exactly same with Logistic Regression, so we will skip the process and just show the result of RF model below:

```
classification_evaluation(RandomForestClassifier(n_estimators=200), cols=original_cols)
```

```
RandomForestClassifier
Accuracy: 0.6837349397590361
              precision    recall  f1-score   support

       False       0.70      0.85      0.77       829
        True       0.62      0.40      0.49       499

   micro avg       0.68      0.68      0.68      1328
   macro avg       0.66      0.63      0.63      1328
weighted avg       0.67      0.68      0.67      1328
```

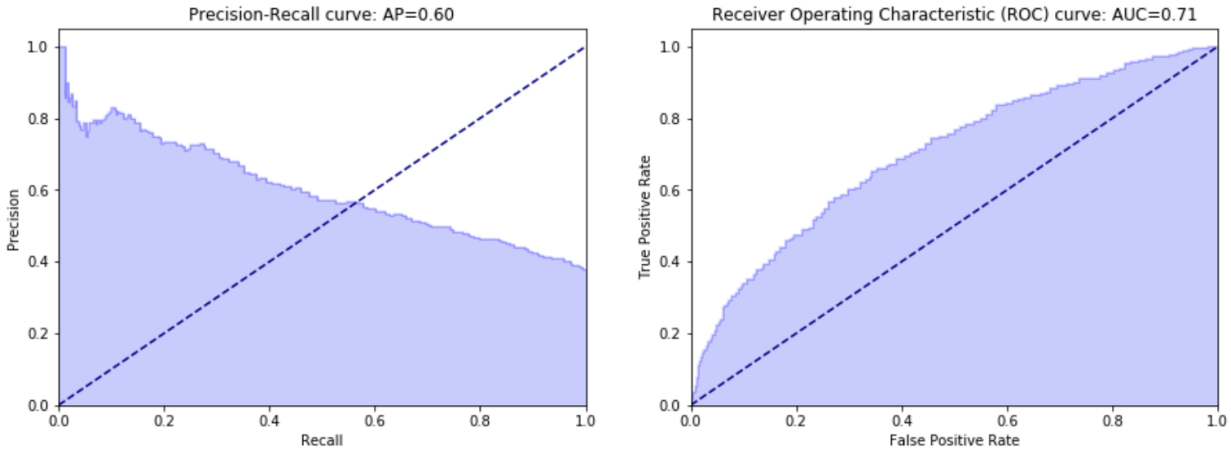

Figure 5:Result of Basic Features for RF

```
classification_evaluation(RandomForestClassifier(n_estimators=200), cols=text_naive_cols)
```

```
RandomForestClassifier
Accuracy: 0.7198795180722891
              precision    recall  f1-score   support

       False       0.73      0.87      0.79       829
        True       0.68      0.47      0.56       499

   micro avg       0.72      0.72      0.72      1328
   macro avg       0.71      0.67      0.68      1328
weighted avg       0.71      0.72      0.71      1328
```

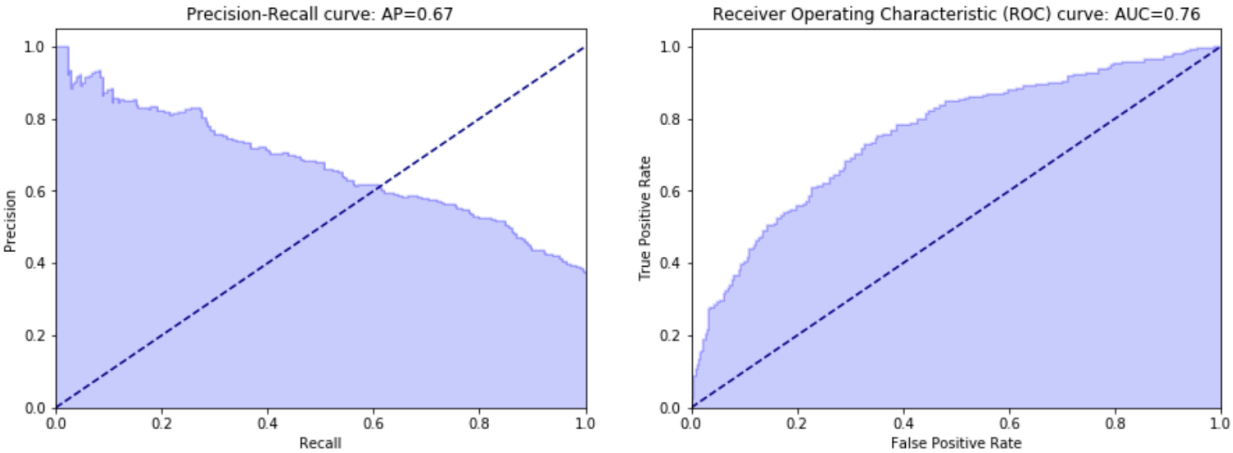

Figure 6:Result of Naïve Bayes features for RF

```
classification_evaluation(RandomForestClassifier(n_estimators=200), cols=text_w2c_cols)
```

```
RandomForestClassifier
Accuracy: 0.7206325301204819
              precision    recall  f1-score   support

       False       0.74      0.86      0.79       829
        True       0.68      0.48      0.57       499

   micro avg       0.72      0.72      0.72      1328
   macro avg       0.71      0.67      0.68      1328
weighted avg       0.71      0.72      0.71      1328
```

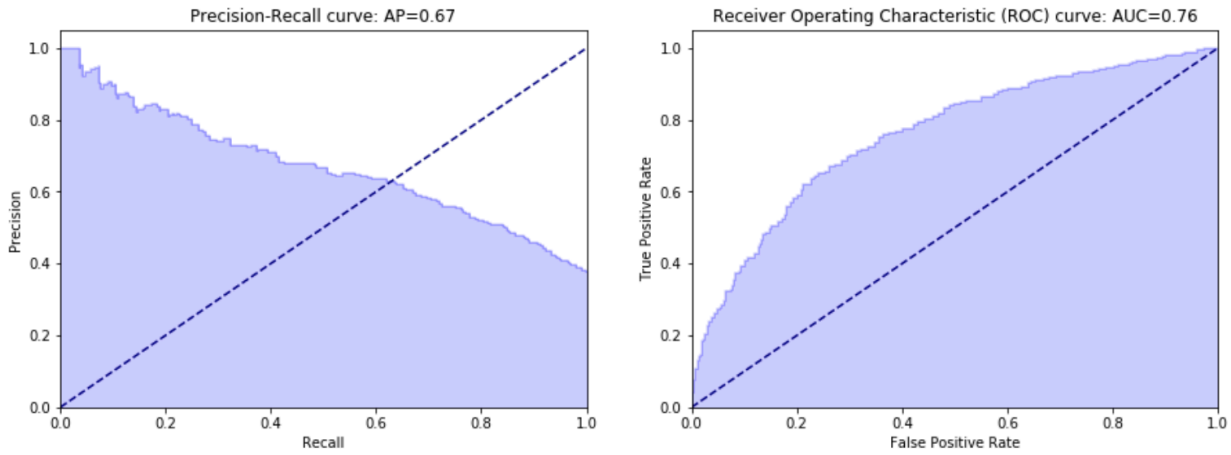

Figure 7:Result of Word2vector features for RF

```
classification_evaluation(LogisticRegression(), cols=text_predicted_rating_cols)
```

```
LogisticRegression
Accuracy: 0.6980421686746988
              precision    recall  f1-score   support

       False       0.73      0.83      0.77       829
        True       0.63      0.48      0.55       499

   micro avg       0.70      0.70      0.70      1328
   macro avg       0.68      0.66      0.66      1328
weighted avg       0.69      0.70      0.69      1328
```

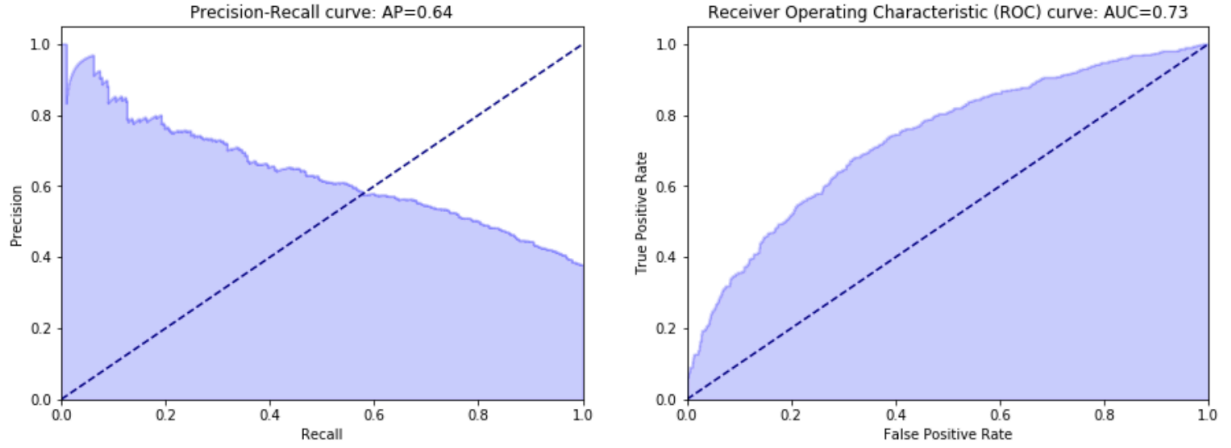

Figure 8:Result of prediced rating feature for RF

5. Conclusion and Future Work

Both Naïve method and 'word2vetor' method can improve model's performance:

Form both Logistic Regression and Random Forest we get the same conclusion which is features generate with Naïve method and 'word2vetor' method in 3.2 accomplish a big progress in prediction performance.

From our point of view, movie is the product of not only human's intelligence but also their taste of art, stories in their own life. So, the director/actor/writer's influence can be really critical for a movie will acquire great amount of audiences or not. Count of appearance in the review text can be the reflection of influence made by movie's director/actor/writer. Let's assume if two movies are under same condition, audiences would like to go into the theater to see a movie that is direct by Clint Eastwood and act by Bradley Cooper, Emma Stone rather than by nobody.

A lot of movies are released every year. Whether the quality of scene and sound are good or not? Is it a great story? Did the actors or actresses perform well? Audiences want to know these answers before they go to see that movie. So, the features related with movie quality are important in predicting box office success.

Using review text to predict the rating score didn't improve model's performance:

For both models, predicted rating scores didn't improve the performance in predicting the target value. We consider there's two reasons: first, when we predict the review rating using LinearSVC, the highest accuracy is 49% (for 1 in 10 classes), it's not good enough. Second, we use 90% exist rating to predict 10% missing rating, the predicted rating must share the same distribution with the rest 90%. So, it is reasonable that predicted rating has no improvement in modeling.

Random Forest performed better than Logistic Regression:

Compare the two models' result we used in prediction. Random Forest performed better than Logistic Regression. Logistic Regression worked well in linearly separable data. For our problem, we start with logistic regression because that will be our baseline, followed by non-linear classifier such as Random Forest. Besides, our data is unbalanced, so RF can be a better choice. We will show the feature importance gathered from RF in Figure 9 below.

In Figure 9, We can see *review count* is the most important feature, and *avg_review_length* in the fourth, which means the popularity of the movie is crucial (the more and longer reviews a movie has, the more popular it is). In most instances, movie's popularity is directly proportional to the advertising investment.

Budget_Final take the second place, our target is ROI, so the investment of a movie will not only decide the Return ratio but also decide the quality of a film in some cases. *avg_rating, story_content* and *character_content* are placed in important positions as well. These features stand for review authors' views about this movie, which proved can be the factors that influence the movie's success in our project. *writer_value*, *star_value* and *director value* shows the influence of movie makers have strong directionality to the audience.

Base on our result, movie theater companies like AMC, BOW-TIE and marketing companies for the movie could analysis the reviews and movie features before it is released. Then decide how many sessions should arrange or how much should be put in advertising.

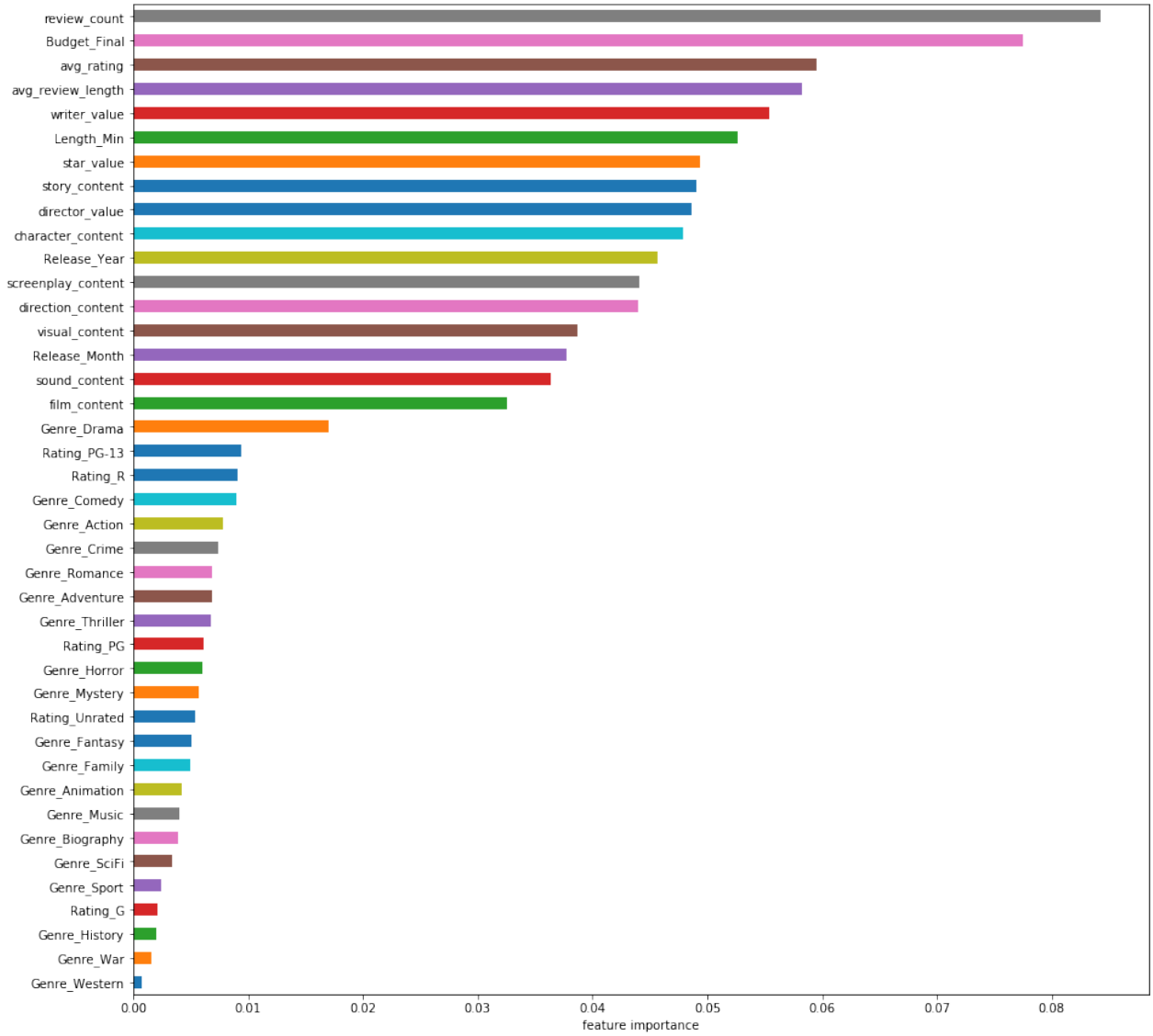

Figure 9: Feature Importance

Construct neural network model to gain deeper insights into the features of items which interest the users by mining reviews:

Although we have made some progress, our mining of the film review text is obviously not enough. In the next step, we will build an artificial neural network, hoping to find some features that may have been overlooked in previous process.

And we may also use more sentiment analysis tools to find the author's attitude towards film/director/actor, and combine them in our model.