Prediction for the Greatness of a Movie

Jin Huang Taihong Cao Hanzhao Wang

Computer Science and Engineering Department, New York University

New York City, USA

{jh5442, tc2305, hw1445}@nyu.edu

*Abstract*

***The growth of the World Wide Web has resulted in troves of reviews for products we wish to purchase, destinations we may want to travel to, and decisions we make on a day to day basis. Using machine learning techniques to infer the polarity of a product is of great importance in this age of information. Since there are too many movies released every year, we need a reliable way to judge whether a movie is worth to watch. The right combination of talent, timing, all with some good luck can have something to do with the popularity of a movie. So, we use a dataset contains the information of more than 5,000 movies all over the world, analyzing the 28 variables and try to find how different factors influence the success of a movie.***

Keywords—machine learning, prediction, regression

# Introduction

Online review has become crucial to film industry, because most people are relying on reviews and social media like Facebook to judge the quality of a film and to decide whether to watch it or not. The likes on Facebook, actors, director, IMDB review and other factors can influence the popularity of a movie. Thus, what makes a movie successful? Among all the factors, there are important ones and less important ones. The purpose of our project is to use Machine Learning methods to analyze the dataset and furthermore to predict the greatness of a movie and help people to choose good movies to watch.

This topic can be taken as a supervised learning problem, which requires analysis and prediction. Since there are over 30 variables in the dataset, we will first use the correlation matrix to find the most important variables and do analyze to the dataset. We plan to use regression model on the dataset and our main method will include linear regression and logistic regression. To be specific, we would like to predict the score (varies from 1 to 10) to each movie which can present how good it could be, which means the IMDB\_score is the label we want to predict. After finishing this, we will compare the result of the different methods and make a conclusion for which one is better for our dataset.

# Motivation

Feature films are a multibillion-dollar industry. Given plenty number of films produced as well as the level of scrutiny to which they are exposed, it may be possible to predict the success of a film based on publicly available data. As we may know, movie ratings are influenced by many factors, so accurate prediction of new movie ratings can be challenging. In recent years, various kinds of analysis techniques were successfully applied to analyzing user reviews, which in turn were applied to predict IMDB movie ratings (based on IMDB reviews, YouTube movie trailer reviews etc.).

What we are aiming at in the project is to predict the IMDB score of a movie. Why is the IMDB score what we concern much? This is because among the factors of a movie, the IMDB score may be the most intuitive factor for normal audience. Although we care about the director, the promotion PV or the main actor from time to time, it is more regular for us to check the score of a movie and judge whether it is good or not because the IMDB score is to some degree a reliable reference as a quantitative indicator.

We focus on the preprocessing and analyzing of the dataset by midway of the project. This includes the selection of the variables, the clearance of the data, checking the relationship among variables, plotting relevant graphs and so on. The preprocessing is of vital importance in data science and takes most of the workload because the regression must be applied on the dataset, and if there are missing values or none-relevant variables, the result of regression may be unreliable. Furthermore, for a dataset with about 5,000 records, there could exist overfitting problem if too many features are used in training model, so it is very important to decide which features to use by preprocessing.

# Related Work

The movie industry is developing rapidly during recent years, generating approximately $10 billion of revenue annually [1]. The prediction of the success of a movie is becoming a not only entertaining but a commercial topic. Normal audience check the data and reviews to decide whether to watch a movie and the movie producing company use the previous data to anticipate the possible performance. Thus, there has been variable works done during the past year related with the prediction of the greatness of a movie. Pimwadee Chaovalit and Lina Zhou use semantic orientation methods to compare Supervised and Unsupervised Classification Approaches [2]. Mary Margarat Valentine et al built a model to predict the performance of a movie based on online rating and revenue and the prediction accuracy are measured for both the models using forecast accuracy methods MAPE and MSE [3]. And Yew Jin Lim and Yee Whye propose a Bayesian approach to alleviate overfitting in SVD, where priors are introduced and all parameters are integrated out using variational inference [4].

All the above-mentioned works focus on finding new methods to predict the performance of the movie with previously existed data. However, what we want to do is to use recent dataset and built a realistic model which is of practical value. Thus, people can judge the greatness of a movie by closely related factors.

# data

In this project, we use a dataset which can be freely downloaded from Kaggle. It is called “IMDB 5000 dataset”.

The dataset is scraped from the IMDB website using a Python library called “Scrapy”. It obtains all needed 28 variables for 5043 movies and 4906 posters, spanning across 100 years in 66 countries. There are 2399 unique director names, and thousands of actors/actresses[5].

Below are the discription of the 28 variables.

1.” movie\_title”: This is the title of the movie.

2.” color”: This indicates whether this is a color movie.

3.”num\_critic\_for\_reviews”: This is the number of the critic for reviews.

4.” movie\_facebook\_likes”: The number of “likes” the movie gets on Facebook.

5.” duration”: The duration of the movie.

6.” director\_name”: The name of the director of the movie.

7.” director\_facebook\_likes”: The number of “likes” the director gets on Facebook.

8.” actor\_3\_name”: The name of actor3.

9.” actor\_3\_facebook\_likes”: The number of “likes” the actor3 gets on Facebook.

10.” actor\_2\_name”: The name of actor2

11.” actor\_2\_facebook\_likes”: The number of “likes” the actor2 gets on Facebook.

12.” actor\_1\_name”: The name of actor1

13.” actor\_1\_facebook\_likes”: The number of “likes” the actor1 gets on Facebook.

14.” gross”: The gross of the movie.

15.” genres”: Tha genres of the movie.

16.”num\_voted\_users”: The number of voted users.

17.” cast\_total\_facebook\_likes”:

18.” facenumber\_in\_poster”: The face number in the poster of the movie.

19.” plot\_keywords”: The key words for the movie.

20.” movie\_imdb\_link”: The IMDB link of the movie.

21.”num\_user\_for\_reviews”: The number of the user for the reviews of the movie.

22.” language”: The language of the movie.

23.” country”: The country in which the movie is made.

24.” content\_rating”: The rating of the content of the movie.

25.” budget”: The budget of the movie.

26.” title\_year”: The year in which the movie is released.

27.” imdb\_score”: The IMDB score of the movie.

28.” aspect\_ratio”: The aspect ratio of the movie.

# Algorithm(s)

V.I Analysis with categorical variables

As we know, there are numerical and categorical variables in the dataset. For example, the number of the critic for reviews is numerical and the name of a director is categorical. It is a challenging problem to deal with categorical variables during the preprocessing of the dataset due to the difficulty of the quantification of the categorical ones.

We plan to drop the categorical variables and leave the numerical ones in regression. However, we need to take a quick look at these variables before dropping them.

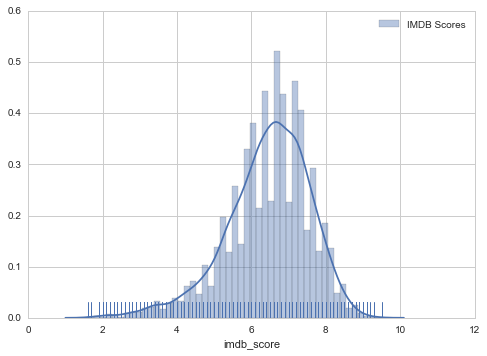


Figure 5-1-1: The Gaussian of imdb\_score

Omitting what has been discussed by midway, we could see from the figure 5-1-1 that the total IMDB score forms a near Gaussian Distribution. The mean of the score is around 7. The curve shows the trend of these data. Further prediction will demonstrate in similar graph to evaluate the model. The curve that most like the original one would be the model that performs best.

V.II Analysis with single variable

Before taking numbers of variables into consideration, we may look at the influence of single variable.

For the convenience, we divided the all the variables about Facebook likes into two groups. One group is the mean of all three actors and the other is the mean of other Facebook likes such as movie Facebook likes and director Facebook likes. From the first graph, we can see that high Facebook likes do not mean that the movie is high of score rating. From the second scatter plot, we can find movie have more Facebook likes tend to have high IMDB scores around 7.5.

We also discussed the difference of other variables by midway, including critics reviews, user reviews, user numbers and so on. We then talked about the relationship among gross, IMDB score and the variables.

V.III Evaluate the importance of variables

Then we go on to analyze the multiple variables for the prediction. Before carrying out analyzing, we need to do preprocessing on the dataset. As has mentioned before, it is very important to clear the dataset for the following analyzing.

Normally, there exist different kinds of data in one dataset, like categorical ones and numerical ones. Also, there exist unknown values or missing values.

We process the dataset with following steps:

(1) We check the missing values in the data frame, and fill them with the same value NaN if there exist missing values.

Data with NaN is incomplete, so we drop the lines with unknown values. We count the lines in the dataset and know that there are 5043 records initially and 3756 records left after dropping the records with unknown values.

(2) We deal with categorical variables. There are some methods to change categorical data into numerical data or into scores. The mostly used way is changing them into dummy variables [6]. Dummy variable is a good way to deal with categorical variables but the problem is that there are too many directors and actors in the dataset. So, if we apply dummy on the dataset, there will be more than 27,000 variables in the frame and it is very difficult to reduce the dimension since they are already independent of each other. Furthermore, for a dataset with about 4,000 records, it is very possible to cause overfitting problem if there are too many variables. For the convenience of analyzing, we decided to drop the categorical variables and only use numerical variables.

(3) We then split the dataset into training set and testing set. This is a simple but necessary procedure.

(4) Standardizing the data. Different variables have different range of values, larger values like budget or gross many have larger influence on the prediction than we have expected. So, we standardize the data into a unified range.

Then we estimate the importance of different invariables. For this step, we choose to use random forest for evaluation. Random forests or random decision forests are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks, that operate by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the [mode](https://en.wikipedia.org/wiki/Mode_(statistics)) of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of [overfitting](https://en.wikipedia.org/wiki/Overfitting) to their training set [7]. Random forest can measure the importance of each feature; we can then select the most important features based on the importance of indicators.

There are 14 variables left after dropping the categorical variables and the importance of each invariable is shown in Figure 5-3-1.

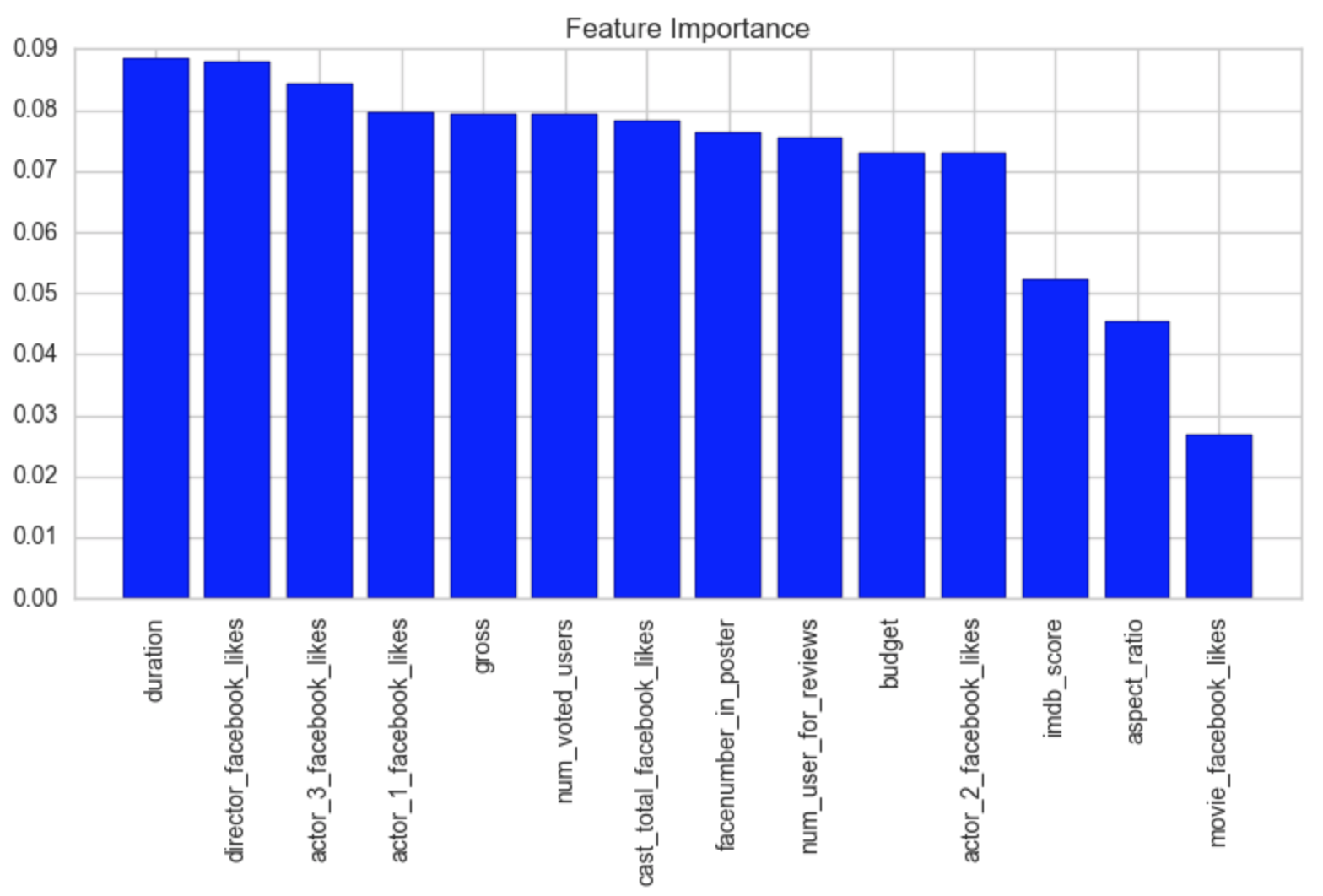


Figure 5-3-1: Feature Importance

We can see that the most important feature for a movie maybe duration and director\_facebook\_likes, similarly. And the least important feature is movie\_facebook\_likes. It is also very interesting that although we often check the IMDB score to judge a movie, it seems that it is not a very important factor among all the invariables.

We plan to take this result as reference and use 8-10 most important variables for regression. This part is still waiting to be discussed.

Finally, we calculate the correlation matrix. The correlation matrix is a square matrix that contains the Pearson coefficients (often abbreviated as Pearson's r), which measure the linear dependence between pairs of features.

The correlation coefficients are bounded to the range -1 and 1. Two features have:

* a perfect positive correlation if r =1
* No correlation if r=0 and
* a perfect negative correlation if r= -1

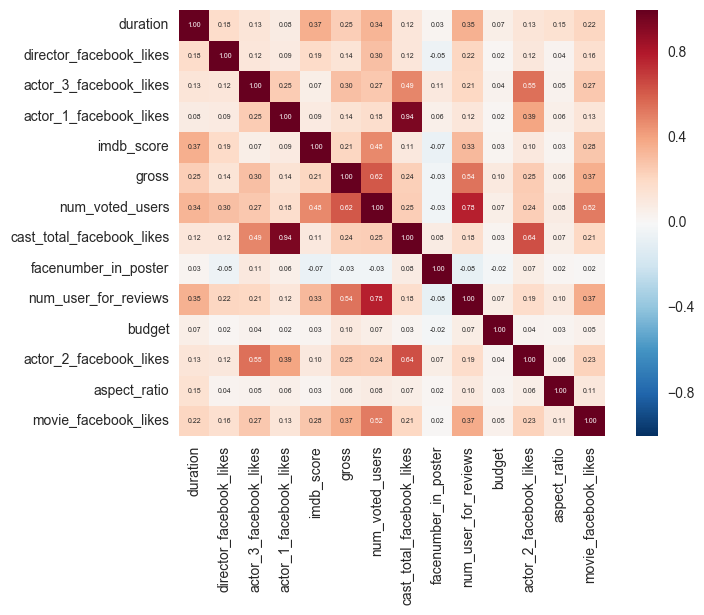


Figure 5-3-2 Correlation Matrix

Visualizing the important characteristics of a dataset, respectively. As mentioned previously, Pearson's correlation coefficient can simply be calculated as the covariance between two features Visualizing the important characteristics of a dataset and Visualizing the important characteristics of a dataset (numerator) divided by the product of their standard deviations (denominator).

The result is shown in Figure 5-3-2.

We can know from the correlation matrix that for our target IMDB score, the most related variables can be ranked as: num\_ voted\_ users, duration , num\_ users\_ for\_ reviews and movie\_facebook\_likes.

The random forest give the importance of variables and the correlation matrix give the relationship among the features, these are all good reference for our predicting the imdb\_score. We will take both into consideration in our following experiments.

V.IV Logistic Regression

Logistic regression can be seen as a special case of the [generalized linear model](https://en.wikipedia.org/wiki/Generalized_linear_model) and thus analogous to [linear regression](https://en.wikipedia.org/wiki/Linear_regression). Logistic regression, or logit regression, or logit model is a [regression](https://en.wikipedia.org/wiki/Regression_analysis) model where the [dependent variable (DV)](https://en.wikipedia.org/wiki/Dependent_and_independent_variables) is [categorical](https://en.wikipedia.org/wiki/Categorical_variable). This article covers the case of [binary dependent variables](https://en.wikipedia.org/wiki/Binary_variable)—that is, where it can take only two values, such as pass/fail, win/lose, alive/dead or healthy/sick [8].

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a [logistic function](https://en.wikipedia.org/wiki/Logistic_function), which is the cumulative logistic distribution. Thus, it treats the same set of problems as [probit regression](https://en.wikipedia.org/wiki/Probit_regression) using similar techniques, with the latter using a cumulative normal distribution curve instead.

In our project, we use Logistic Regression to predict the imdb\_score of the film.

V.V Linear Regression

In [statistics](https://en.wikipedia.org/wiki/Statistics), linear regression is an approach for modeling the relationship between a scalar [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable) *y* and one or more [explanatory variables](https://en.wikipedia.org/wiki/Explanatory_variable) (or independent variables) denoted *X*. Linear regression is a type of regression analysis to be studied rigorously, and to be used extensively in practical applications. We use relative variables to predict IMDB score.

The results for LR model will be discussed in section VI.

V.VI Other Methods Implemented

Besides the Linear Regression and Logistic Regression, we also implemented some other models, including K-Neighbors, Decision Tree Classifier and Random Forest Classifier. We also compare the performance of different models.

The results for all methods will be given and discussed in section VI. RESULT.

# VI. RESULT

VI.I Logistic Regression

We choose two variables 'director\_facebook\_likes' and 'num\_critic\_for\_reviews', which are the most related variables by analyzing the trend of variables as our training set. Because it is logistic regression, we must classify all movies to two parts, one is the movies whose scores more than 7.0 and the other part is the movie whose score less than 7.0. Thus, we can mark the movies whose scores more than 7.0 as 1 and otherwise as 0. The accuracy of the prediction with above variables is 69.55%.

Secondly, by computing the accuracy of test set, we can get a score for logistic regression.

Then, we add another variable 'duration'. which is the most related variable by doing the random forest. We can see the score of accuracy is larger than before, which is 72.61%.

We also calculate the MSE for training and testing set, which are 0.8776 and 0.8740. This is a satisfied result, which implies that we may use these three variables to predict IMDB scores with other models.

VI.II Linear Regression

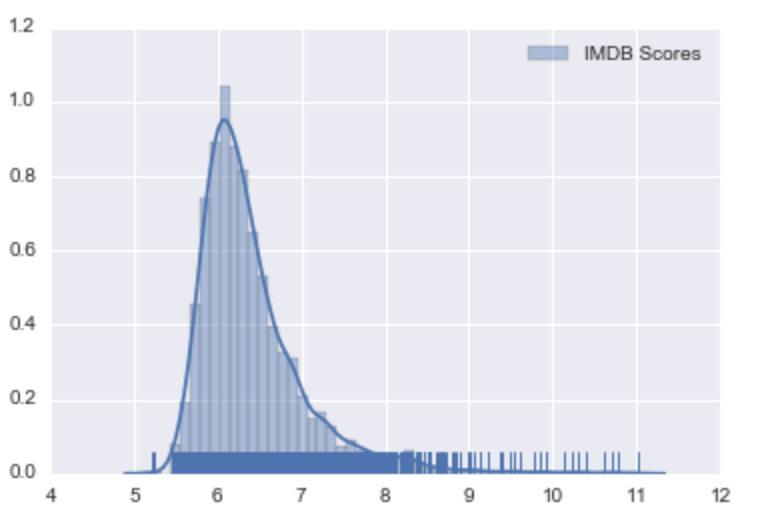


Figure 6-2-1 Linear Regression

Comparing with the original distribution, we can see that the linear regression model does not perform well. From the graph, we could see that it loses the scores that are smaller than 5. Although the mean of the data predicts well, the trend of the result is far away from the original data in that it ranges smaller as shown in Fig. 6-2-1.

VI.III K Nearest Neighbors

The k-Nearest Neighbors algorithm (or k-NN for short) is a [non-parametric](https://en.wikipedia.org/wiki/Non-parametric_statistics) method used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression](https://en.wikipedia.org/wiki/Regression_analysis).In both cases, the input consists of the k closest training examples in the [feature space](https://en.wikipedia.org/wiki/Feature_space). The output depends on whether k-NN is used for classification or regression:

* In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive [integer](https://en.wikipedia.org/wiki/Integer), typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.
* In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

k-NN is a type of [instance-based learning](https://en.wikipedia.org/wiki/Instance-based_learning), or [lazy learning](https://en.wikipedia.org/wiki/Lazy_learning), where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all [machine learning](https://en.wikipedia.org/wiki/Machine_learning) algorithms.

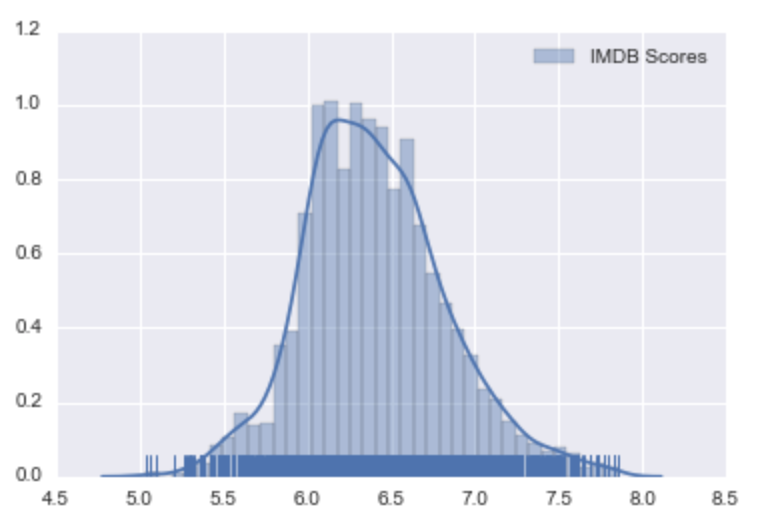


Figure 6-3-1 KNN (n=15)

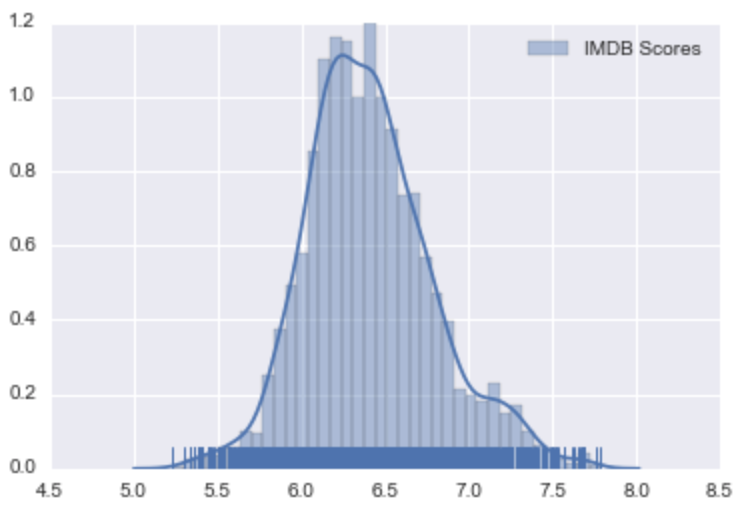


Figure 6-3-2 KNN (n=25)

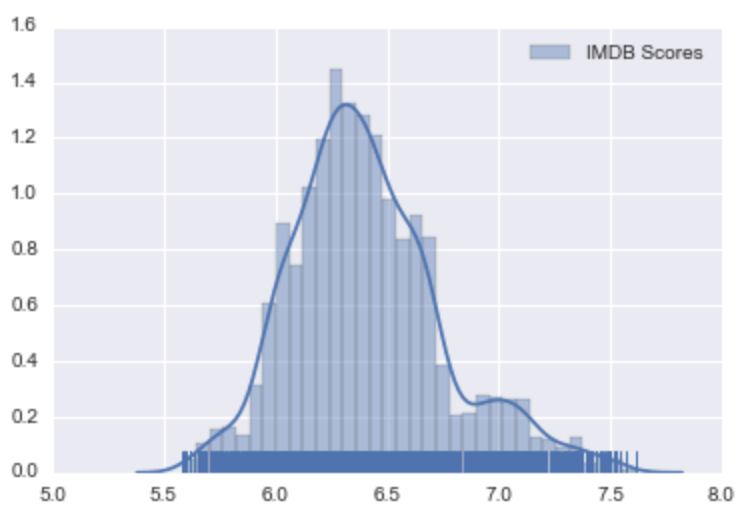


Figure 6-3-3 KNN (n=50)

In general, KNN performs better than linear regression. We have tried three different neighbors in KNN methods, namely 15,25,50. The figures above show the prediction range. Although normally increasing n should lead in better performance. In this case n=25 performs best in that it has relatively low Mean squared error. The general curve is much more like the original data. Yet, still it does not predict the score less than 5 or more than 8.

Table1: The performance of KNN

|  |  |  |  |
| --- | --- | --- | --- |
| N | Score | MSE | Explained Variance Score |
| 5 | -0.11 | 1.30 | -0.11 |
| 15 | -0.01 | 1.09 | -0.01 |
| 25 | 0.01 | 1.16 | 0.02 |
| 50 | -0.01 | 1.18 | -0.01 |

VI.IV Other Models

We also apply different methods to classify whether a movie is a good movie or a bad movie. First we suppose movies that have imdb\_score above average are good movies. In this case, movies have score over 6.5 are good ones. Then we use Gradient Boosting Classifier, Decision Tree Classifier, K-Neighbors Classifier and Random Forest Classifier respectively to classify our data. Using roc\_auc\_score to rank these methods, we found that Gradient Boosting Classifier performs best, following by Random Forest Classifier. K-Neighbors Classifier peforms worst. So, when classify a movie is bad or good, we shall apply Gradient Boosting Classifier.

The comparison of accuracy of different models are listed in the table 2.

|  |  |
| --- | --- |
| Model | Accuracy |
|  |  |
|  |  |
|  |  |
|  |  |

# VII. Code

Our code is uploaded to our Github page for the project. The programming and analyzing is based on the python and jupytor notebook in Anaconda. Each of the group member is responsible for different parts of analyzing.

You can find more details and analysis by running our kernel and check the contents. But please read the readme file and pay attention that the dataset we use is a local one so each of us has our own path. If you want to run the kernel by yourself, please remember to change the path of the dataset file at the beginning of the code. The dataset file is also uploaded to the Github page, which is called “movie\_metadate.csv”.

Here is the link for our Github page:

<https://github.com/KiyoshiKAWASAKI/ML-Project>

# VIII. VIDEO LINK

@@@@@To be finished

IX. EVALUATION

In this report, we finished the analysis of the dataset as scheduled. We firstly take a quick look at the categorical invariables and then give analysis based on single invariant. Then, we only care the numerical invariables to simplify the problem. We mainly use the random forest and correlation matrix to evaluate the importance and relationship among the features, by which we find some important features like duration, director\_facebook\_likes, actor3\_facebook\_likes and actor1\_facebook\_likes. Thus, we can use important invariables to build our regression model. After we finish the regression model, we may hopefully find which method is better for the prediction of the movie based on our dataset.

# X.CONCLUSION

# XI.Reference

[1] "The Numbers - Movie Market Summary 1995 to 2011." The Numbers - Movie Box Office Data, Film Stars, Idle Speculation. Web.

<http://www.the-numbers.com/market/>.

[2] Movie Review Mining: A Comparison between Supervised and Unsupervised Classification Approaches, Pimwadee Chaovalit and Lina Zhou, Proceedings of the 38th Hawaii International Conference on System Sciences, 2005

[3] A Model for Predicting Movie’s Performance using Online Rating and Revenue, Mary Margarat Valentine, Veena Kulkarni, and R.R. Sedamkar, International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September-2013, P277-282.

[4] Variational Bayesian Approach to Movie Rating Prediction, Yew Jin Lim and Yee Whye The.

[5]https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset

[6]https://ljalphabeta.gitbooks.io/python-/ content/categorical\_data.html

[7] [Hastie, Trevor](https://en.wikipedia.org/wiki/Trevor_Hastie); [Tibshirani, Robert](https://en.wikipedia.org/wiki/Robert_Tibshirani); [Friedman, Jerome](https://en.wikipedia.org/wiki/Jerome_H._Friedman) (2008). [*The Elements of Statistical Learning*](http://www-stat.stanford.edu/~tibs/ElemStatLearn/) (2nd ed.). Springer. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [0-387-95284-5](https://en.wikipedia.org/wiki/Special:BookSources/0-387-95284-5).

[8] <https://en.wikipedia.org/wiki/Logistic_regression>