# Movie IMDB Score Prediction

## CSCI 630 Final Project Report

Jingyang Li, Haonan Yin - April 19, 2017

## Introduction

This project aims at making analysis and prediction on movie IMDB score based on the dataset called "IMDB 500 Movie Dataset"[1] from Kaggle. In order to do that, we have applied some machine learning and data mining methods in this project such as data preparation and data cleaning, feature selection and extraction, min-max algorithm, linear regression, logistic regression and so on.

The motivation for us to choose this topic as our final project is that it is closely relevant to our daily lives, since everyone must have watched a lot of movies with different stories, by different crews. Moreover, IMDB score is believed to be the fair way of rating movies by public. Therefore, we think this will be a very meaningful project to do.

The main problem of this project is to find the most relevant features which are not present in the dataset, then use a proper training model to make prediction on testing dataset. This means we need to extract and generate hidden features from available ones, then try different models to test since it is unknown to us which will have better results beforehand.

## Method

The method can be divided into several parts:

1. Feature selection
2. Data cleaning and preparation
3. New feature generation
4. Model training and testing

Each of the steps will be discussed individually in following parts.

Before starting these steps, the most important thing for us is understanding our dataset well. This dataset has 5043 instances and 28 attributes. The attributes include information of movies, directors and actors, and also the public response to them on the

Internet. However, it is not a perfect dataset because there are some missing values in it. Therefore, handling these missing values properly would be vital to us.

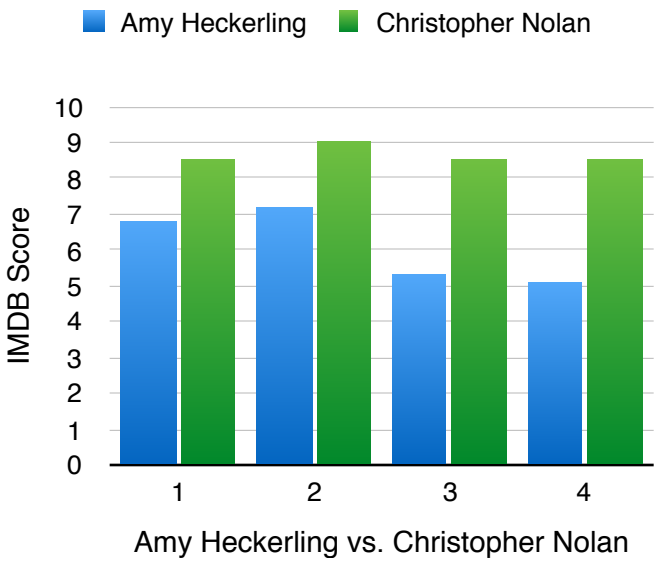
## Feature Selection

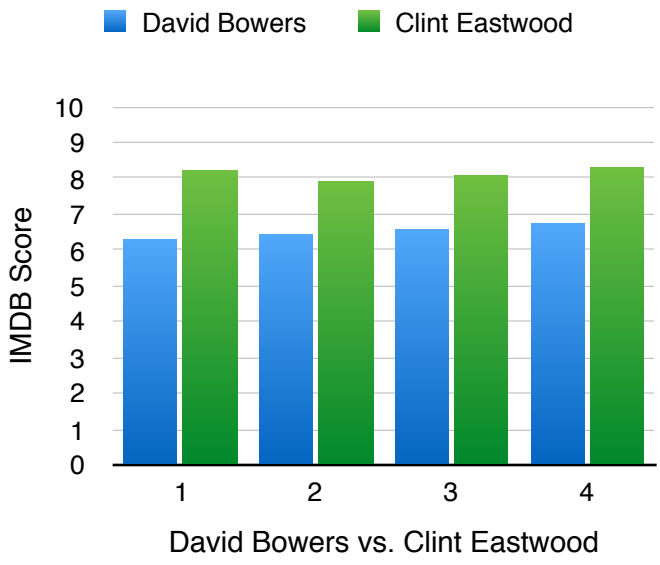
It is obvious that the target variable is "imdb_score". When it comes to features, things become complicated. As we discussed before, there is no numeric feature directly related to imdb score. Therefore, we study the dataset and draw a conclusion as follows:

The movie's plot keywords, director and actors are the most relevant features to its imdb score. The reasons are as follows:

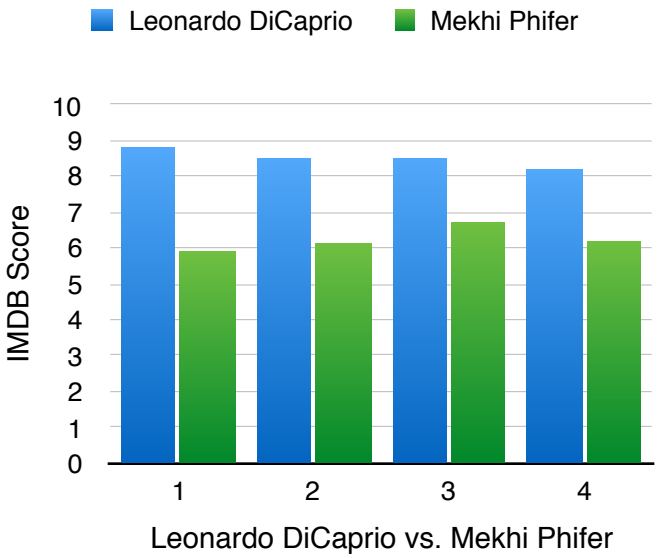
The most important feature in our opinion would be plot keywords because they present the essence of the whole movie. Audience can simply understand the story based on these keywords and also be attracted by them. Some people might argue that movie title is the most important feature, but as a matter of fact, it is a very constrained feature which cannot give more information than plot keywords. For example, the eighth episode of "Fast and Furious" franchise is called "Fate of the Furious" which sounds totally strange to us, but the plot keywords must be all the same.

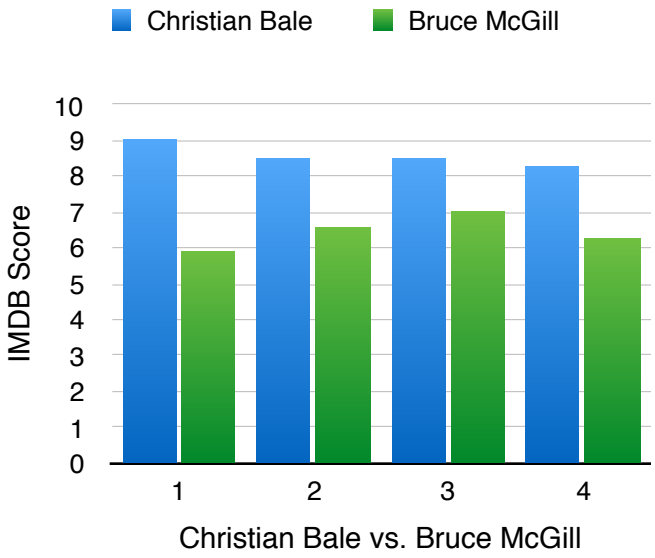
These following two figures show a comparison between two pairs of directors, we can easily tell that Christopher Nolan and Clint Eastwood who are well known to all clearly produce movies with higher IMDB score than the ordinary directors. This is a consistent and steady data pattern throughout the whole dataset.



Similarly, the next two figures illustrate that famous actors are more likely to have higher rating movies than the others. This is also common sense in real life, because skilled and experienced actors can make the movie more touching and impressive.




Above all, we decide to use these three features as key features in this project. However, the biggest issue is to properly and scientifically convert nominal data into ratio data in order to give a good prediction result. The solution will be discussed in "New Feature Generalization" part.

## Data Cleaning and Preparation

### Data Cleaning:

First of all, we use Pandas to read in the csv file as a dataframe. Then drop all the instances with duplicate information. And also, drop the ones with missing values in either one of the three features. At last, reset the index of this dataframe.

### Data Preparation:

In this part, we extract different features for different models because some of them requires single feature while others require multiple features. Apart from this, we decide to use five fold cross validation which means we use eighty percent of data as training set, the left twenty percent of data as testing data. We believe the result would be most reliable using this method of cross validation.

## New Feature Generalization

We count appearance of directors, actors and keywords. Sum up each one's corresponding imdb score, then get the mean value for each of them. Thus, for each movie, we can compute its keyword's score, director's score and actors' score. In order to normalize the data, we apply Minmax algorithm as follows:

$$V_{norm} = \frac{V - V_{min}}{V_{max} - V_{min}} \times 100\,\%$$

We normalize this because we are seeking for relations between three features and the target variable. One feature cannot be converged with another attribute if they have no common basis for comparison. With normalization we are able to keep relative relations of each instance in the same attribute.

Except for the director, there are three actors and multiple plot keywords per movie.

Therefore, the way of calculating their weights is sum their values up for each movie after getting their individual mean score. In this way, we will have a cumulative value of this feature for this movie. Then apply normalization as stated above.

## Model Training and Testing

We have tried several different models in this project such as linear regression, multinomial logistic regression, stochastic gradient descent and multi-class adaboosted decision trees. Only linear regression gives us a good accuracy while others do not.

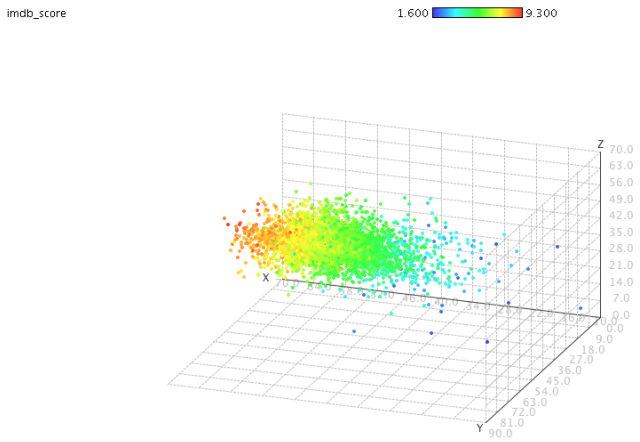
For multinomial logistic regression, stochastic gradient descent and multi-class adaboosted decision trees, the input features are "plot_keywords_weight", "director_weight" and "actor_weight". However, none of them give us a satisfying result. Test results (prediction accuracy) for these methods are as follows:

Multinomial Logistic Regression: 30.8%

Multi-class Adaboosted Decision Tree: 24% - 25%

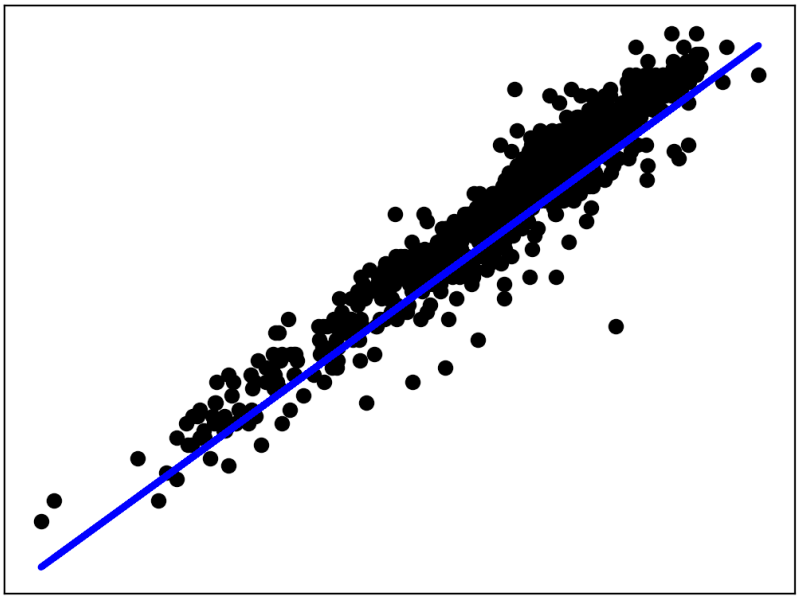
Stochastic Gradient Descent: < 33%

Since these methods are all performing bad on prediction, we try to seek for a better model which can fit this dataset well. We soon realize that the three features have some linear relations to each other which means linear regression might be a better way to try. The evidence shows as follows:




x-axis is plot_keywords_weight
y-axis is director_weight
z-axis is actor_weight
color represents imdb score

What we did was using these three key features to generate another mixed feature and use this as the input of linear regression. We are doing this because linear regression can only take one feature as input. Therefore, we used the mean value of these three feature to have a test run. The result is 84.4% accuracy which is surprisingly good and encouraging. Resulting figure is as follows:



## Conclusion

We have learnt a lot from this project. Apart from the matching learning algorithms and methods, a very precious thing for us is that it trains us the way of seeking to solve problems. More significantly, it seems like we successfully find a scientific method of predicting the imdb score of a movie. We hope to apply this method to predict coming movies in the future, and see if it is a solid method in a long run.

## Reference

1. "IMDB 5000 Movie Dataset": https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset