Exploring and Modeling with the IMDB Movie Dataset

University of Memphis

December 06, 2016

1 Introduction

Movies can be of different stories, genre, from different countries, having different actors (both male and female), directors and others. There are several sources in the internet today that stream our favorite movies on demand, that is whenever you want and as often you want, Netflix for example. We all love to watch movies. It is one of the best entertainer during the leisure hours.

1.1 Goal Description

We are interested to predict whether a movie is going to be popular or not or what will the gross income of a particular movie before its release or what will be the IMDB score of a movie or to suggest you a movie quite similar to the one who have just watched. This seems to be exciting, since based on our relied prediction one (from the audience) can go and watch a movie (predicted to be popular and entertaining) or based on the prediction of the gross, the movie makers can change their marketing regime for a particular movie, before its release. In this work, we have done an exploratory analysis and modeling with IMDB movie data set available in an online data science platform, Kaggle. Namely, we have performed visualization, 'value' prediction or regression in the form of predicting the imdb score and gross of a movie, 'class' prediction or classification in the form of predicting the popularity of a movie, and clustering in the form of a basic recommender system.

1.2 DataSet Description

We used the IMDB 5000+ movie dataset for our project. The dataset consists of 5043 unique movies and their attributes (28 variables). The collection spans over 100 years and 66 different countries. There are 2399 unique director names and thousands of actors and actresses. The following is the list of 28 variables from the data set,

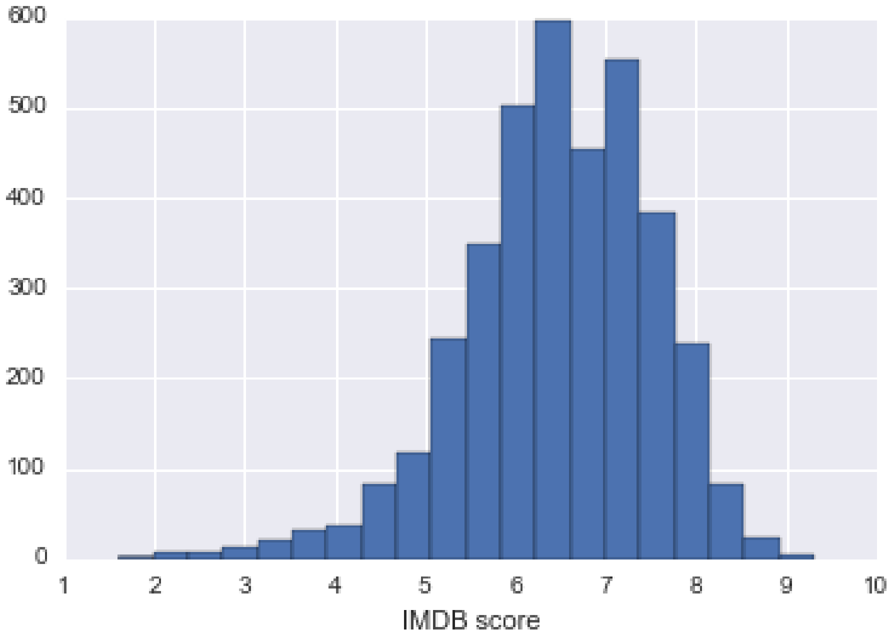

Figure 1: Histogram plot for imdbscore

[movietitle, color, numcriticforreviews, moviefacebooklikes, duration, directorname, directorfacebooklikes, actor3name, actor3facebooklikes, actor2name, actor2facebooklikes, actor1name, actor1facebooklikes, gross, genres, numvotedusers, casttotalfacebooklikes, facenumberinposter, plotkeywords, movieimdblink, numuserforreviews, language, country, contentrating, budget titleyear, imdbscore, aspectratio.]

2 Methodologies

2.1 Regression or 'Value' Prediction

In this work we have developed regression models for predicting, *imdbscore* and *gross* of a movie. First we describe the analysis and methods used for *imdbscore* prediction. *imdbscore* is a score or rating provided to movies by critics by an online databased platform for movies, IMDB[]. This score is able to quantify how popular a movie is. Figure **??** shows a histogram plot for the imdb scores. From the plot we can observe a normal distribution of the score, with a mean = 6.465±1.056, minimum value of 1.6 for the movie *Justin Beiber: Never say never* and maximum of 9.3 for the movie *Shawshank Redemption.* In the data set, we have 28 variables, out of which there are 13 useable numeric features, namely, [directorfacebooklikes, gross, numvotedusers, num-

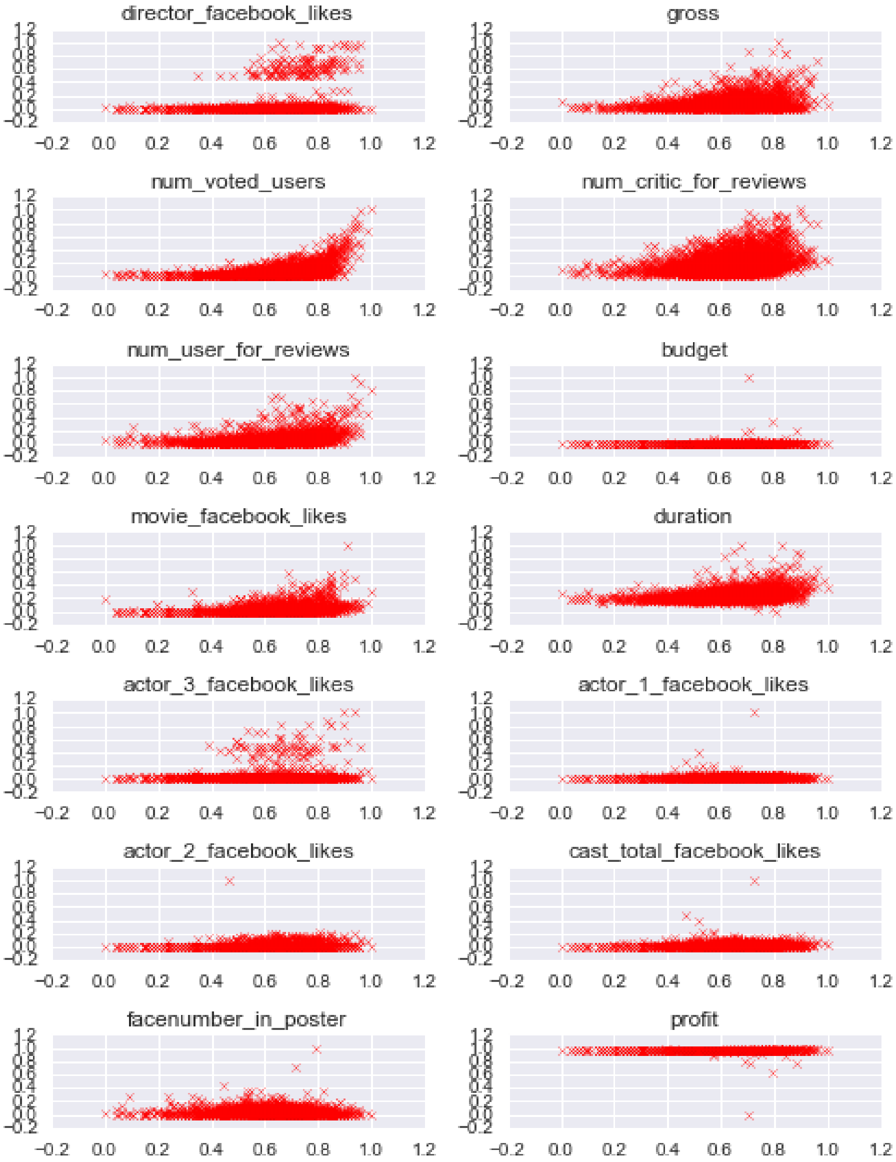

Figure 2: Scatter plot for imdbscore v numerical features

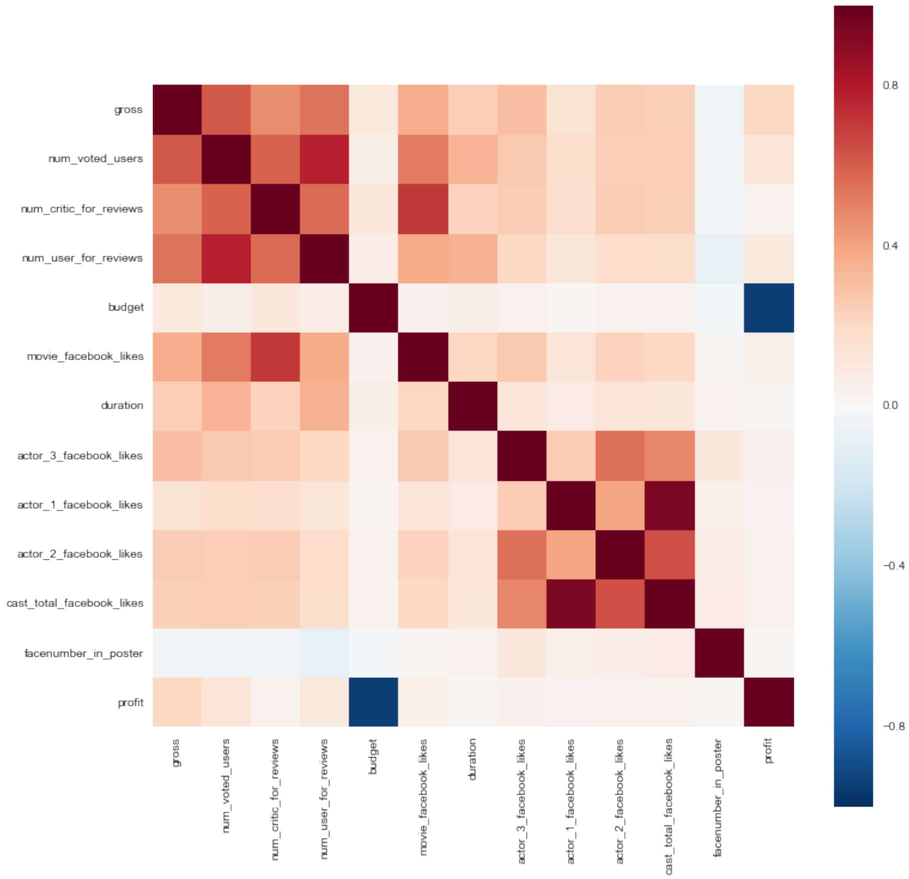

Figure 3: Correlation of Numerical features

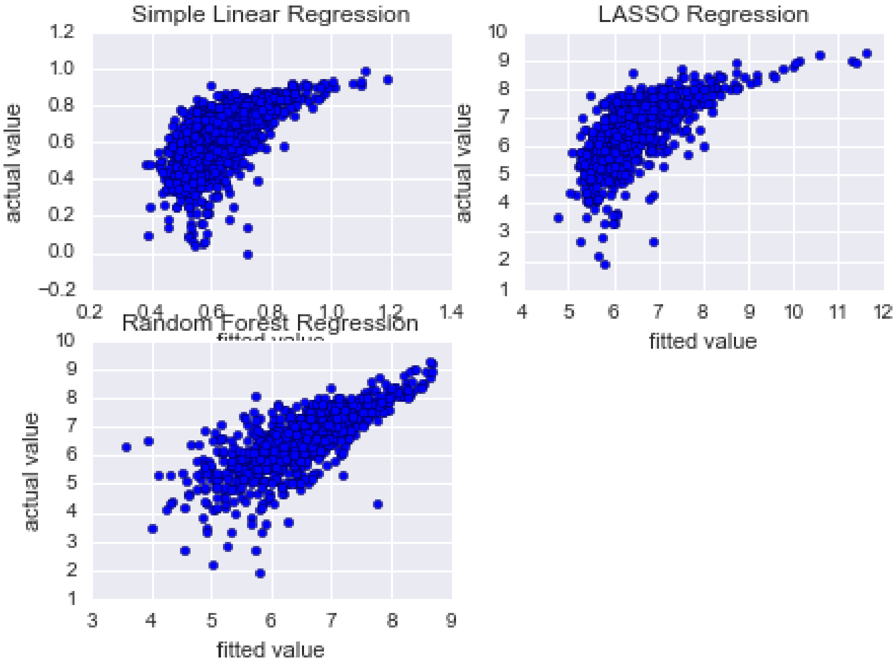

Figure 4: Fitted vs Actual data plot

criticforreviews, numuserforreviews, budget, moviefacebooklikes, duration, actor3facebooklikes, actor1facebooklikes, actor2facebooklikes, casttotalfacebooklikes, facenumberinposter], and we have included the profit of a movie, which is calculated by subtracting the budget from the gross. First, we have cleaned the data by dropping those samples having NaN values (using 3756 out of 5043 for modeling) and scaled all the features. For all analysis we have used the numpy, scipy, sklearn, pandas, matplotlib, seaborn and bokeh libraries of python.

2.2 Prediction of ImdbScore

- Initial step: Since this is a regression problem, we just run a simple linear regression using all the numerical features. To be noted for all the linear regression models, we have performed evaluation by running different folds of the training and test samples, that is with trainsize = 0.5, 0.66 and 0.75 and report the mean residue square value and the mean square error. We observe the following results. The mean MSE is 0.013 and mean residual square is 0.296.

- Second step: Next we analyzed the correlation of these features with the *imdbscore* as shown via the scatter plots in figure 2. We observe that *numvotedusers* has the highest correlation with *imdbscore*, 0.482 and *facenumberinposter* has a negative correlation of -0.065. We observe that for *directorfacebooklikes* (642 out of 3756 are 0) and *moviefacebooklikes* (1742 out of 3756 are 0) has a lot of missing data, hence, we remove them from our next analysis. Next we again performed a simple linear regression for predicting the *imdbscore*, with the new set of features. We observe the following results. The mean MSE is 0.0126 and mean residual square is 0.323. We observe an improvement from the first step.

- Third step: Next we observe that there are a few text features that might be useful in predicting the *imdbscore*, namely *genre*, *plotkeywords* and *country*, we need to add these to the numerical features. For this, first we obtain the tfidf scores with the tfidf vectorizer and use todense() to convert each feature from sparse to dense matrix. Next we append these feature along with the numerical features to obtain a new feature list to predict the *imdbscore*. We observe the following results. The mean MSE is 0.011 and mean residual square is 0.455. We observe an improvement from the second step.

- Fourth step: Next from the figure 3, we observe that some of the features are correlated to each other. For example, *numvotedusers* and *numcriticforreviews* have some correlation, similarly *gross* and *numvotedusers* have some correlation, hence in this step we have used LASSO regression, since this method does feature selection by assigning 0 coefficient for those features which are redundant and hence reduces overfitting via first norm regularization. We use aic/bic for evaluating the model. We have found the optimal alpha from a range of alpha values of the LASSO model for which the aic and bic are minimum. We performed cross validation to obtain a mean cross-validation score of 0.471. We observe an improvement from the third step.

- Fourth step: Next from the figure 3, we observe that some of the features are correlated to each other. For example, *numvotedusers* and *numcriticforreviews* have some correlation, similarly *gross* and *numvotedusers* have some correlation, hence in this step we have used LASSO regression, since this method does feature selection by assigning 0 coefficient for those features which are redundant and hence reduces overfitting via first norm regularization. We use aic/bic for evaluating the model. We have found the optimal alpha from a range of alpha values of the LASSO model for which the aic and bic are minimum. We performed cross validation to obtain a mean cross-validation score of 0.471. We observe an improvement from the third step.

- Fifth step: Next we attempted using the Random Forest Resgression to perform the prediction of the *imdbscore*. We performed 5-fold cross validation for the evaluation. We observed the mean cvScore to be 0.511. Another important aspect of random forest is that it provides us with the feature importance list. We note the following top 10 features for *imdbscore* prediction. [numvotedusers, genres, budget, duration, numuserforreviews, gross, numcriticforreviews, directorfacebooklikes, actor3facebooklikes, profit].

Hence we conclude that random forest regression works best among all the methods used to predict the value of *imdbscore* given the features of a movie. Figure 4 shows the fitted vs actual data plot for the three methods used.

2.3 Prediction of Gross