Exploring and Modeling with the IMDB Movie Dataset

Sayma Akther, Md. Kauser Ahmmed,
Soujanya Chatterjee, Anjana Tiha, Md. Azim Ullah
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]

1.1 Goal Description

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 their next movie given the features. This work will be useful to both the audience and the movie makers. 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

[1] Please find all the figures in the Appendix section below

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,

[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 Methods and Experimental Evaluation

3 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, numcriticforreviews, 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.

3.1 Prediction of IMDB Score

- 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 5. 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* since they fit the *imdbscore* better than the other text/categorical features (MSE = 0.016 and residue squared = 0.1). 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 6, 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 7 shows the fitted vs actual data plot for the three methods used.

3.2 Prediction of Gross

Next we use the movie data set to predict gross of a movie. Both categorical and numerical features of the data have been used to predict gross. It has

Table 1: Gross Prediction

Evaluation Metrics	Random Forest Regression	Decision Tree Regression
Mean Absolute Error	0.0398	0.0456
Mean Square Error	0.0048	0.0065
R-Square	0.4628	0.3225
Explained Var Score	0.468	0.265

been treated as regression problem. First, data has been pre-processed. The numerical features have been scaled. We have used 5 numerical features; 3 top actors and directors Facebook likes, budget, and 7 textual/categorical features; 3 top actors names, directors name, country, content rating, language for gross prediction. Textual features have been labeled for each column separately and then they have been transformed to binary form.

Random Forest Regression and Decision Tree Regression has been used to predict gross. Cross Validation has been applied to evaluate the regression model performance. 5-Fold Cross Validation has used. Cross validation with Mean Absolute Error(MAE) and Mean Squared Error(MSE) has been calculated. Figure 8 shows the plot for actual v predicted gross when using Random Forest Regression. Figure 9 shows the plot for actual v predicted gross when using Decision tree Regression. Table 1 displays the results.

4 Classification

In this section we discuss the two classification problems addressed in this work. First, we build classification models for prediction of high rated or popular ($imdbscore > 8$) and low rated or not so popular movies ($imdbscore \leq 8$). Second, we also build classification models to predict profitable ($profit > 0$) v not profitable movies ($profit < 0$).

To build the model, 11 numerical and 3 text/categorical features were used (same as used in prediction of imdbscore).

- Prediction of popular or not so popular movies - First we worked with only the numerical data. Several classification models were employed. We evaluated the models via 5-Fold cross-validation. Table 2 includes all the results. We observe that decision tree and random forest classifiers work the best. Next we added the text/categorical feature along with the numerical features. Table 3 includes all the results. Random forest classifier again performs the best.

- Prediction of profitable v not profitable movies - First we worked with only the numerical data. Several classification models were employed. We evaluated the models via 5-Fold cross-validation. Table 4 includes all the results. We observe that decision tree and logistic regression classifiers work the best. Next we added the text/categorical feature along with

Table 2: Classification of High rated movie(Numerical features)

Model Name	Precision	Recall	F1-score
Decision Tree method	1.00	1.00	1.00
Extra Trees method	0.98	0.98	0.98
Random Forest method	1.00	1.00	1.00
SVC method	0.95	0.95	0.92
Logit method	0.93	0.95	0.93
GaussianNaiveBayes method	0.92	0.91	0.92

Table 3: Classification of High rated movie(Numerical+ categorical featuress)

Model Name	Precision	Recall	F1-score
Decision Tree method	0.95	0.95	0.95
Extra Trees method	0.96	0.96	0.95
Random Forest method	0.96	0.96	0.96
SVC method	0.95	0.95	0.92
Logit method	0.94	0.92	0.93
GaussianNaiveBayes method	0.95	0.93	0.94

the numerical features. Table 5 includes all the results. Support Vector classifier again performs the best.

- Observation - In both the cases we observe that classification performance drops on addition of text/categorical features.

5 Clustering

We used clustering to build a simple movie recommender system. Consider a scenario, where an user would like to find a similar movie to the one he/she has already watched, based on *genre*, *plot key words*, *cast and crews*. This system will be able to suggest top 10 similar movie given the movie watched based on

Table 4: Classification of profitable movie(Numerical features)

Model Name	Precision	Recall	F1-score
Decision Tree method	0.97	0.97	0.97
Extra Trees method	0.86	0.85	0.85
Random Forest method	0.89	0.89	0.89
SVC method	0.75	0.55	0.40
Logit method	1.00	1.00	1.00
GaussianNaiveBayes method	0.54	0.55	0.46

Table 5: Classification of profitable movie(Numerical+ categorical features)

Model Name	**Precision**	**Recall**	**F1-score**
Decision Tree method	0.68	0.68	0.68
Extra Trees method	0.68	0.68	0.68
Random Forest method	0.70	0.70	0.70
SVC method	0.75	0.55	0.40
Logit method	0.70	0.70	0.70
GaussianNaiveBayes method	0.66	0.56	0.44

k-means clustering algorithm. To build the system, first we produce unique clusters given the features, *genre*, *plot key words*, *cast and crews*, where *cast and crews* includes all the actors and directors. Next, given a movie, we search for other movie in the cluster. Once we have the movie list, then we suggest to the user the top 10 best movies based on their *imdbscore* or *gross*. We also have the autocorrect feature, that is if the movie provided is not in the database, then we suggest the best matches with respect to the provided title. We show a screen shot of the system in the figure 10 and figure 11

We developed several visualization in the process of performing exploratory analysis of the movie data set. These visualizations help us in understanding the data, extract useful information from the data and then use them for modeling purposes.

First we build a web based interactive visualizer using the *bokeh* library of python. It includes various options which the user can select from in order to obtain the data visualization according to their query. For example, if one wants to know the relationship between *number of critic reviews* and *imdbscore* for movies starring Brad Pitt then the user can simply select the required options to get the data visualization displayed on the web page itself. Figure 12 shows a screen shot for this visualization.

Next we also developed several interesting data exploratory analysis, represented by different plots.

- Top 10 rated movies based on IMDB rating - Shawshank Redemption. Figure 13 displays the result.
- Top 10 directors based on critic reviews - Steven Spielberg tops the list Figure 16 displays the result.
- Top 10 actors based on critic reviews in IMDB - Jhonny Depp on the top Figure 17 displays the result.
- Top 10 actors based on number of user votes in IMDB - Morgan Freeman on the top Figure 18 displays the result.
- Top 10 actors based on gross in IMDB - Robert Downey Jr. he seems to enjoy a whole lot of money Figure 19 display the result.

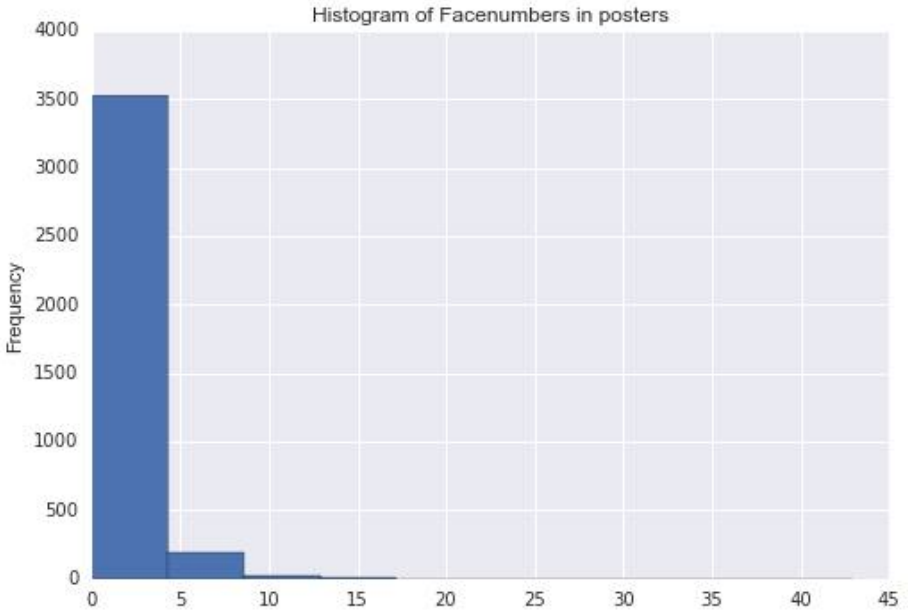

Figure 1: No. of faces on poster

- Top 10 countries having most movie budget - USA tops the list. Figure 20 displays the result.
- Box and Whisker plot to show the distribution of imdb score based on selected year and countries - The year 2007 shows highest median movie ratings. USA produces the maximum films annually however the average imdb scores is far less than those films produced in other countries. Figures 14 and 15 display the results.
- Top 10 profitable movies - Avatar, Jurrasic World and Titanic tops the table. Figure 21 displays the result.
- Histogram of face number of posters - Interestingly we find that almost 98% of posters have less than 5 faces. Figure 1 displays the result.
- Top 10 directors raking highest gross - Steven Spielberg tops the list. Figure **??** displays the result.
- Top 10 directors raking highest average budget - Joon-ho Bong from South Korea. Interestingly South Korea makes less movies but all of them are of substantial budget. Figure **??** displays the result.

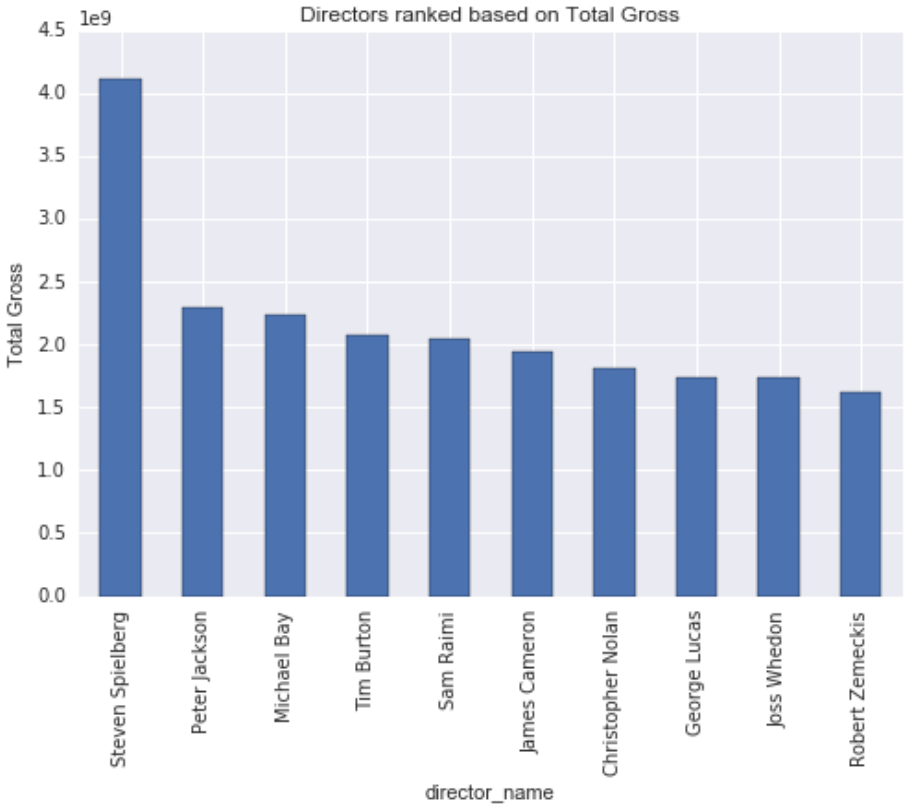

Figure 2: Top 10 Grossing Directors

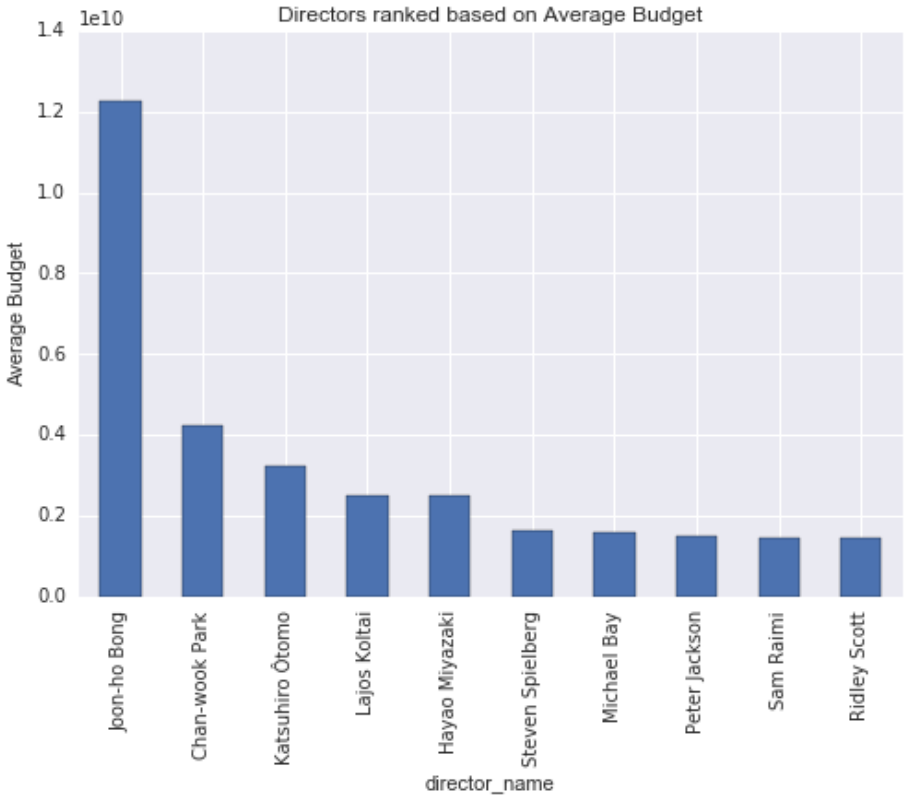

Figure 3: Top 10 Directors by average Budget

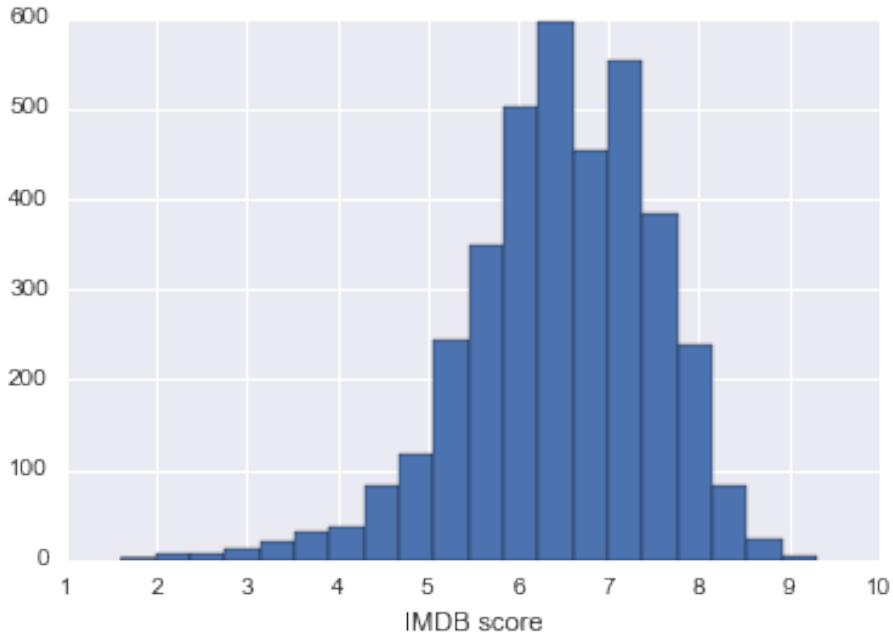

Figure 4: Histogram plot for imdbscore

6 Contributions

- Sayma Akther - Classification (profitable v non profitable movie and popular v non popular movie) model development, statistical analysis, evaluation and results.
- Md. Kauser Ahmmed - Clustering and text/categorical feature and statistical analysis for prediction of IMDB Score.
- Soujanya Chatterjee - Correlation analysis of numerical features, comparative regression model development for prediction of IMDB score, evaluation and results.
- Anjana Tiha - Data analysis, Visualization for gross prediction using regression models, evaluation and results.
- Md. Azim Ullah - Interactive web-based visualization development and exploratory statistical data analysis.

7 Apendix

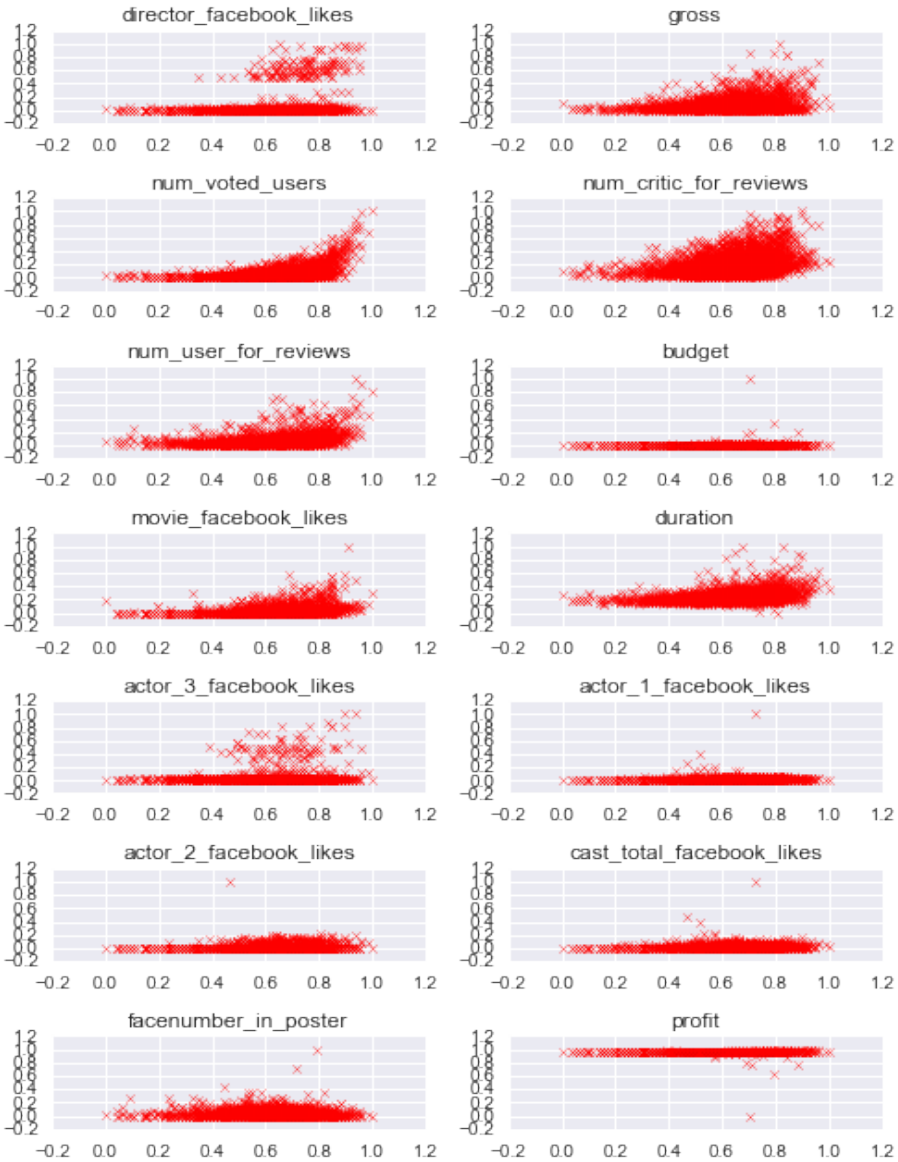

Figure 5: Scatter plot for imdbscore v numerical features

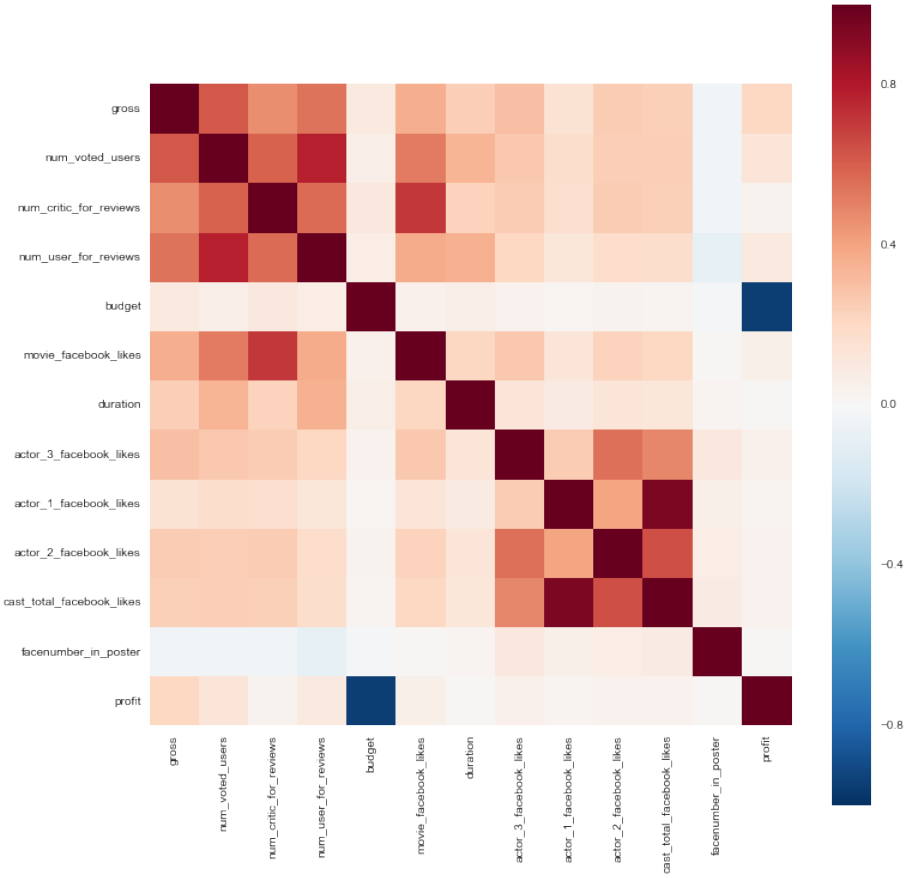

Figure 6: Correlation of Numerical features

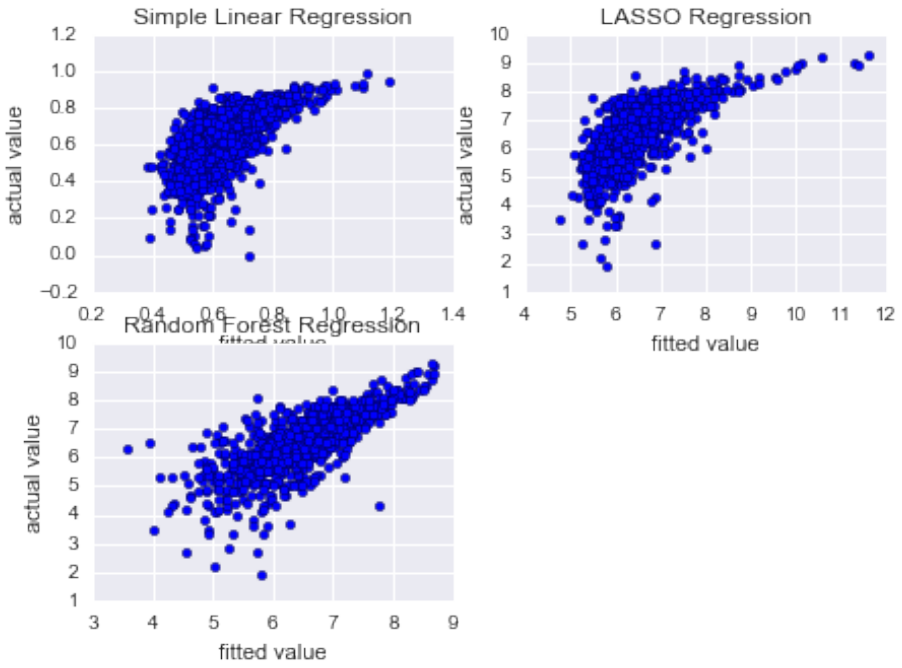

Figure 7: Fitted vs Actual data plot

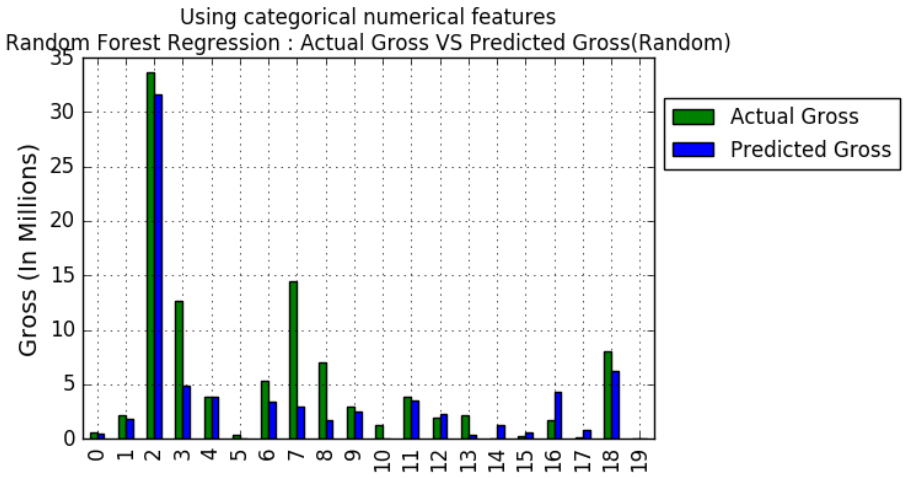

Figure 8: Movie Recommendation according to IMDB Score

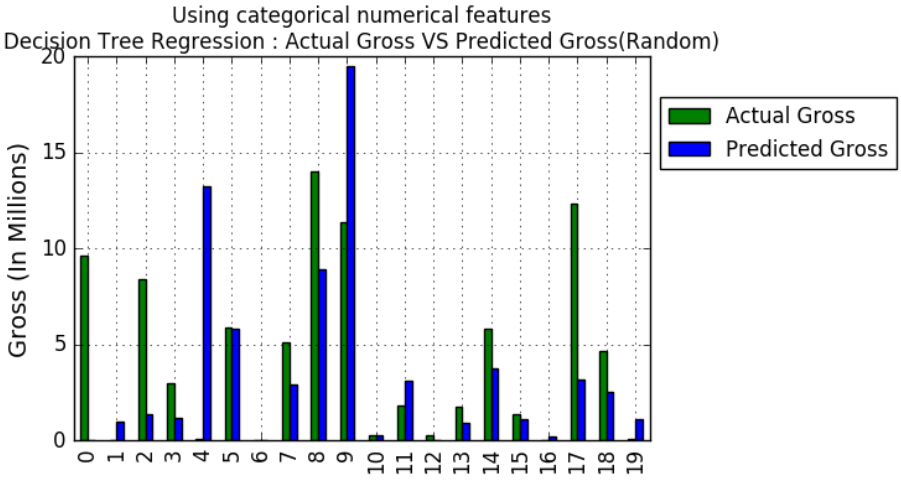

Figure 9: Movie Recommendation according to IMDB Score

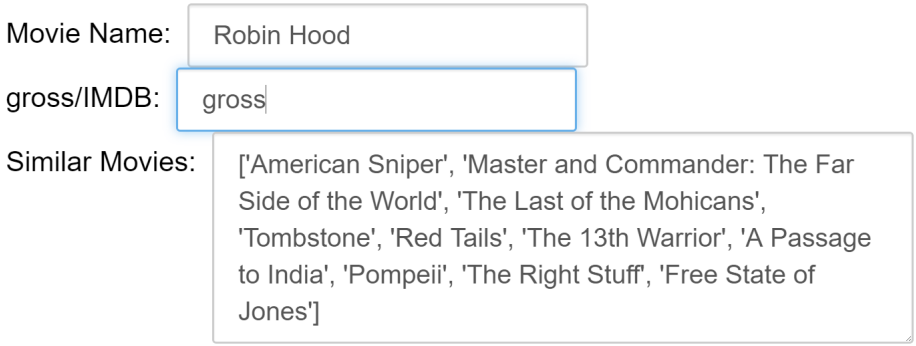

Figure 10: Movie Recommendation according to gross

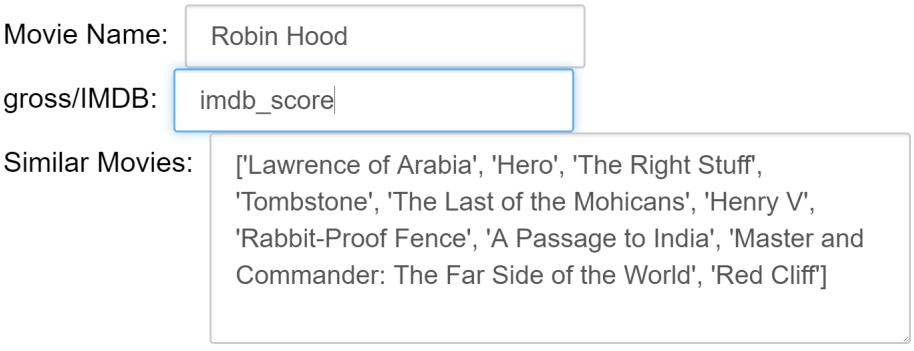

Figure 11: Movie Recommendation according to IMDB Score

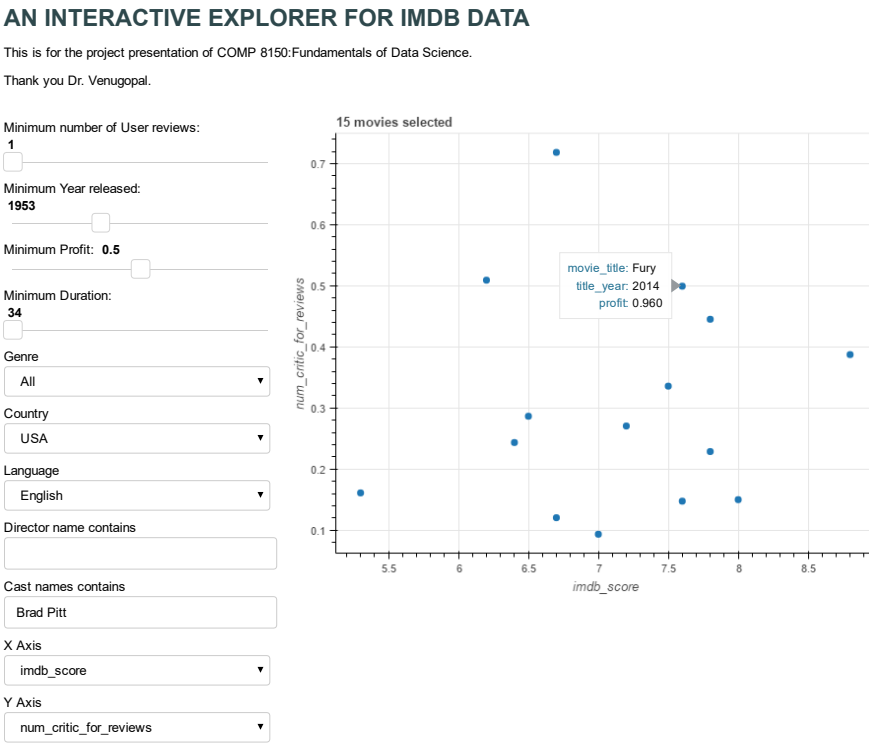

Figure 12: Interactive web-based Visualizer snapshot

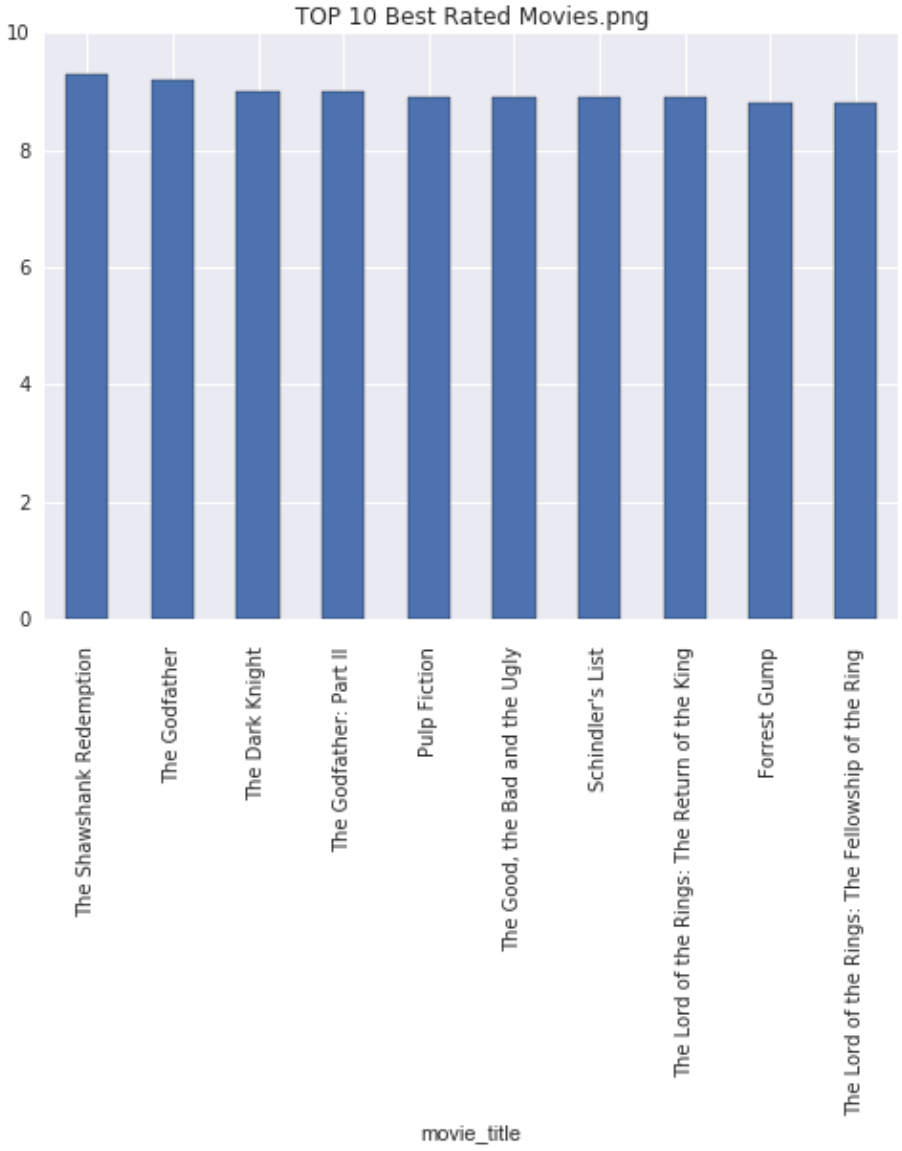

Figure 13: Top 10 Best Rated Movies

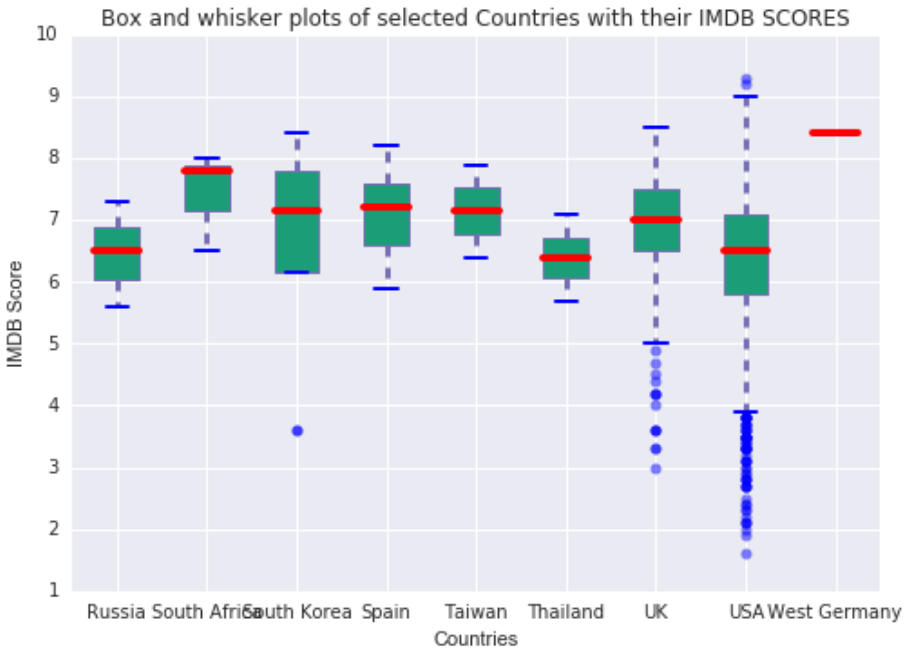

Figure 14: IMDB Score based on Countries

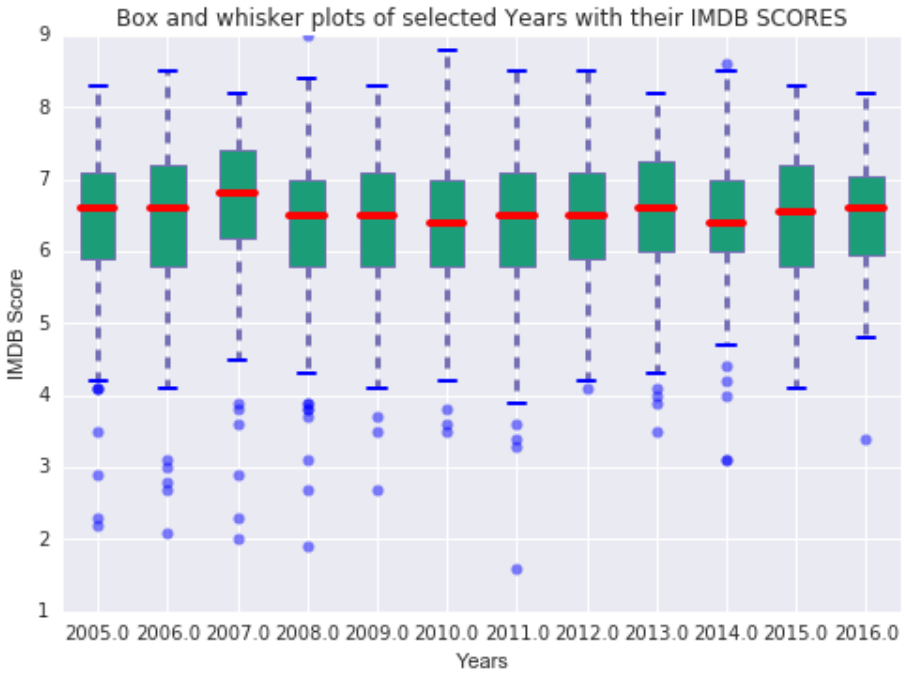

Figure 15: IMDB Score based on year

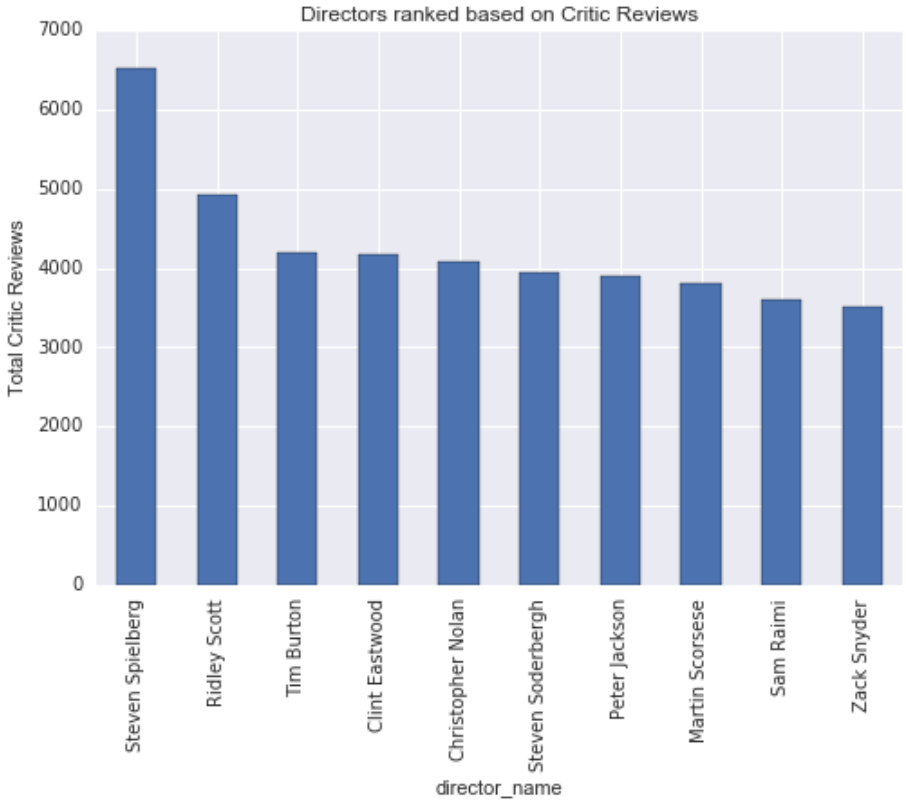

Figure 16: Top 10 Critics Choice Directors

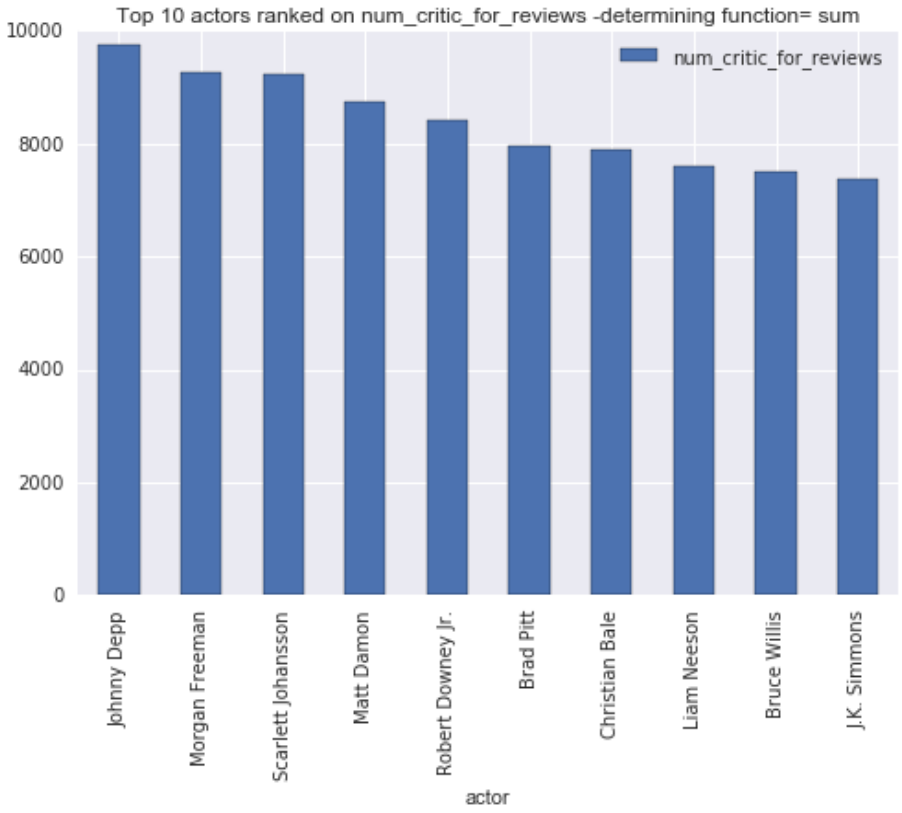

Figure 17: Top 10 Critics Choice Actors

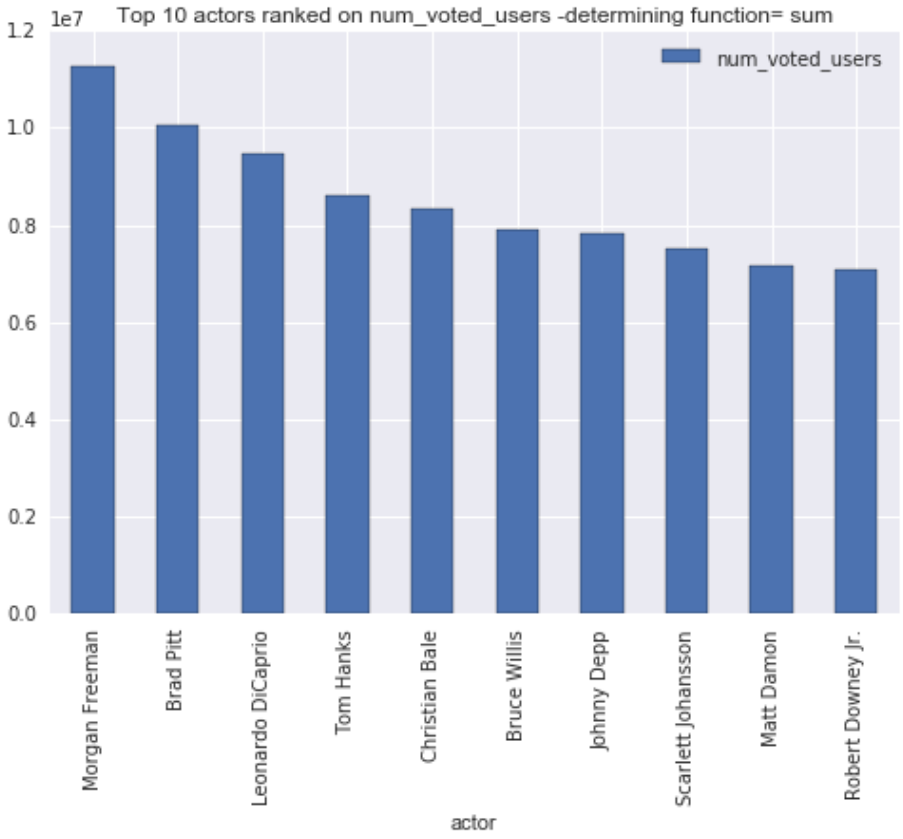

Figure 18: Top 10 User Voted Actors

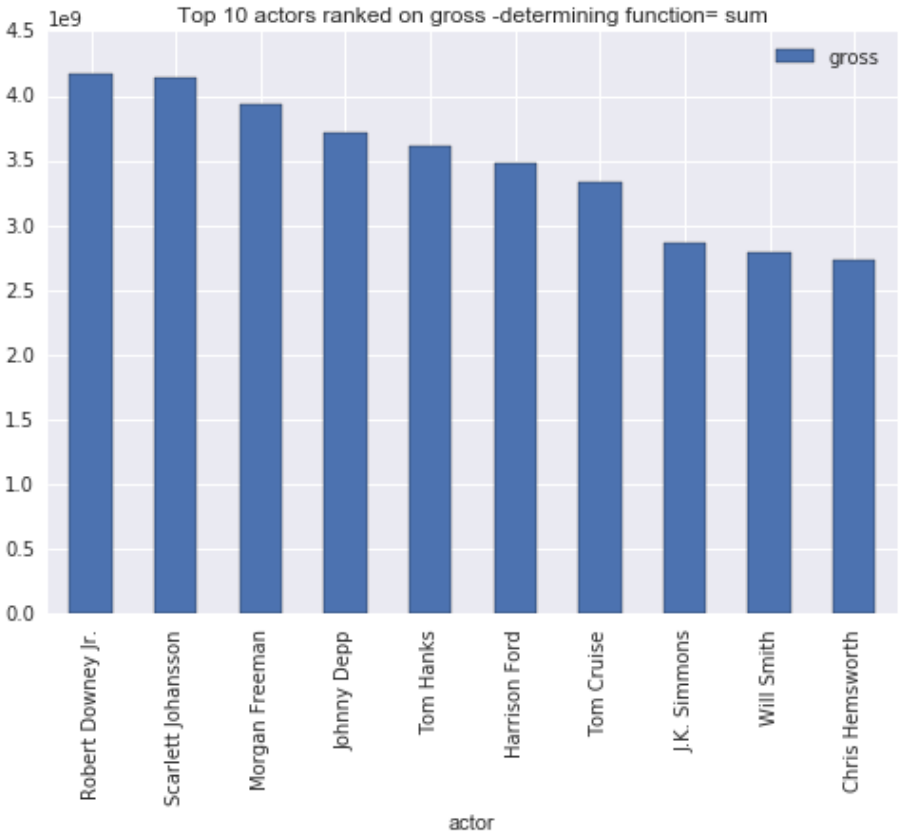

Figure 19: Top 10 Grossing Actors

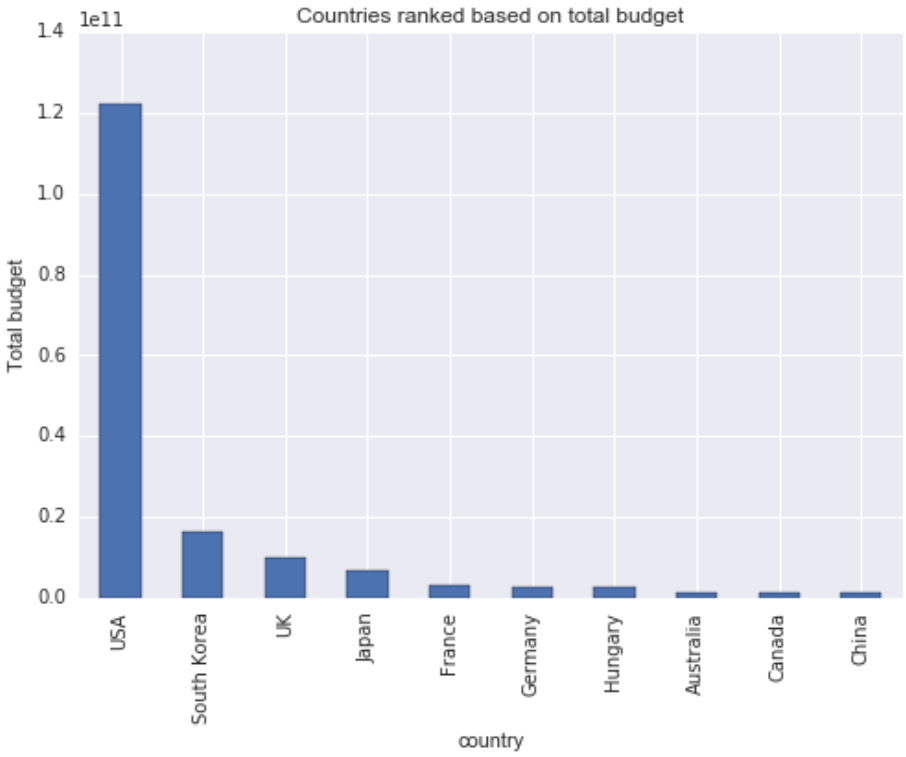

Figure 20: Top 10 Countries by Budget

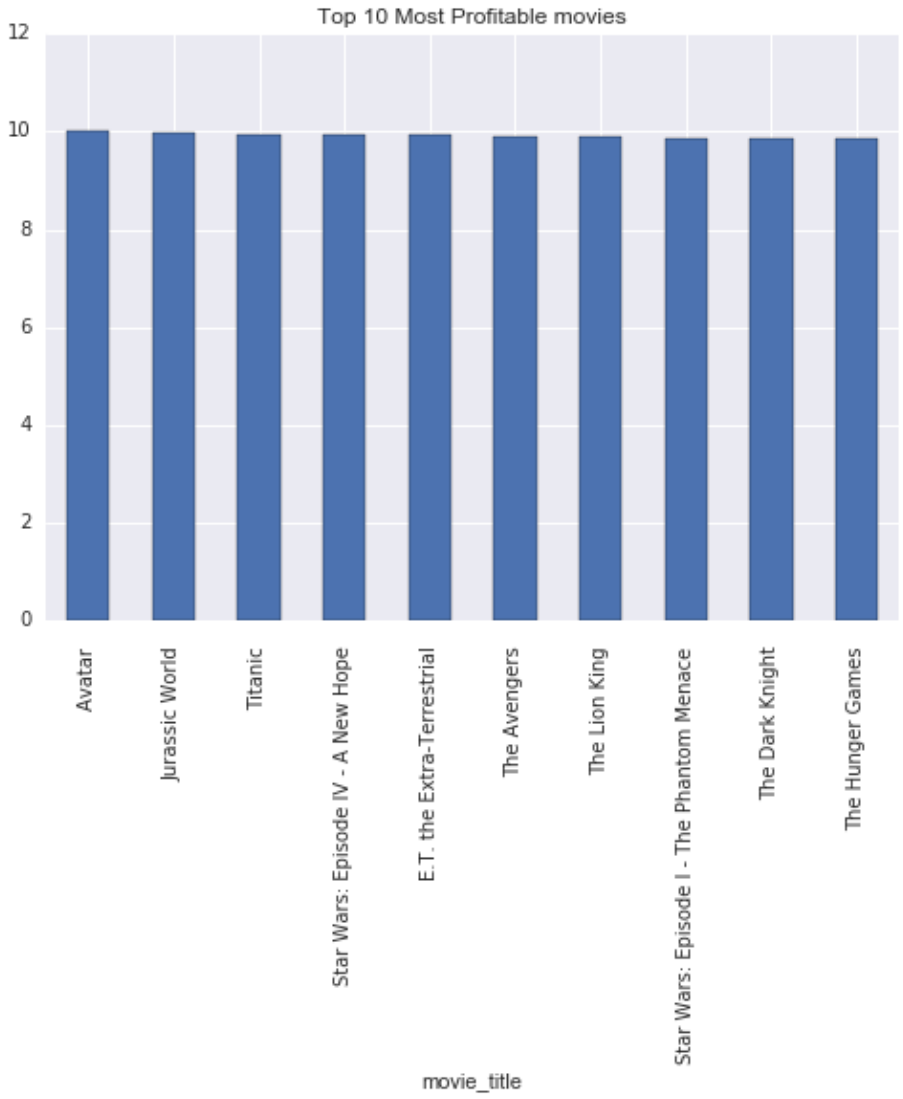

Figure 21: Top 10 Profitable Movies