IMDB Top 5000 Movie Analysis

Gloria D'Azevedo (gad87@cornell.edu)
Erik Enriquez (eee37@cornell.edu)

May 17, 2017

Contents

1 Abstract

The goal of this project is to find models to accurately predict movie "success" and determine characteristics that are most important in predicting success. This analysis focuses on finding predictors and models that accurately predict the "success" of a movie. The focus is on adjusted film gross and IMDB score (a rating assigned by IMDB that may vary from 1 to 10) . More specifically we attempt to find models that predict movie gross, score and whether a film's adjusted gross ranks above the 90^{th} percentile, where the adjusted gross takes into account inflation.

2 Introduction

"Success" is hard to measure so there are multiple proxies to define it. One such measure is the gross amount of the movie, or the total revenue that the movie generates. Other measures of success include the score assigned to a movie by critics which in this data set is represented by the IMDB score and the amount of "awareness" that it has which is represented by the number of Facebook Likes that the movie has. In addition, instead of looking at exact numbers of these measures, we can also encode them as binary variables to see whether or not they exceed a threshold.

The data set that we are using is the IMDB 5000 Movie data set from Kaggle. The data includes names of the actors and directors, the number of likes on Facebook that they have, the budget for the movie, the score, the gross amount that the movie made, and other information about the movie. (`https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset`) Some of the characteristics included in the data set are breakdowns of Facebook likes for various movie aspects (for the actor or actress individually, for all the actors and actresses, the director, and the movie itself).

In addition we have movie-specific metrics such as the year it was released, the content rating, the IMDB rating, the genre, and the duration. If significant, they can play a role in planning and editing new movies. For example, if the content rating of a movie was lowered from a rating of R to a rating of PG-13, then more people are able to watch it which could increase the overall amount of money that it makes. Similarly, if a movie length was decreased from 150 minutes to 120 minutes, then additional showings of the movie could be scheduled, thus raising ticket revenues. People interested in the success of a movie, such as producers, studio investors, and cast members, will benefit from the results of this project.

During the analysis, a variety of data mining techniques and models will be tested to predict the adjusted gross amount of a movie. Cross validation is used throughout to tune our models. Tests were also made to predict whether adjusted gross exceeded a certain amount or percentile (90^{th}) to be classified as a "box-office hit".

3 Initial Data Analysis and Processing

We decided to remove data points that had missing values from the data set. This way averages and medians can still be computed. Another issue is that there are movies from countries other than the United States and thus the gross amount is reported in different currencies. On top of that, the movies are from different years so the currency exchange rate may differ for certain countries by year. We focused on American English since they make up a majority of the data set. As a result we did not have to deal with exchange rates. We incorporated for inflation so that all movie budget and gross amounts could be compared on equal terms. An inflation factor was calculated based on a movie's title year and used to adjust the gross and budget amounts. Data used to calculate inflation rates for different years was taken from a publicly available US inflation data set (http://www.usinflationcalculator.com/inflation/historical-inflation-rates/).

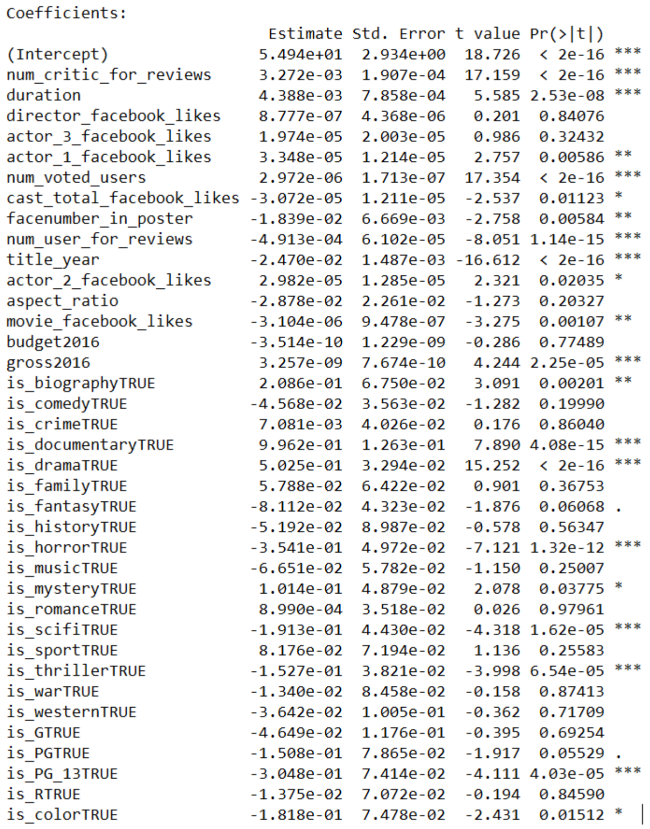
The field "genre" is a list of the categories that describe the movie, divided by the "|" character. Data scraping was used to divide up this text predictor into multiple binary values. Content rating was another categorical variable that was converted to multiple binary values. In the data there are a few predictors that were specific to each movie such as the movie title and the IMDB link to that movie. In addition, some fields such as director's and actors' names and plot keywords are too granular and each only have a few data points that have those values exactly. Such predictors are not considered for the model since they do not imply underlying trends for movies in general.

4 Model Development

We will try to fit a variety of models to better understand the data and make accurate predictions. The models include, but are not limited to, methods such as linear and logistic regression, KNN (K-Nearest-Neighbors) and random forests. Each of the models will try to predict either the adjusted gross amount, whether or not the adjusted gross amount will exceed the 90^{th} percentile, and the IMDB score for the movie.

4.1 Linear Regression to Predict Adjusted Gross and IMDB Score

When fitting a linear model to the data, we tried several different methods. The first method was to fit all the relevant predictors to predict the adjusted gross amount and the IMDB score. This model includes the predictors for the number of Facebook likes that the director has, the number of Facebook likes for the top 3 actors, the number of users who voted, the total number of Facebook likes for the cast, the ratings Approved, Not Rated, and Passed, the year of the movie, the Facebook likes of the movie, the adjusted budget, and whether or not the genre includes drama, history, horror, and music. This is the best model that minimizes the training data and may not perform well for new data. For the model predicting adjusted gross, the Adjusted R^2 is 0.493 and the Residual Standard Error is 17,450,000. And for the model predicting IMDB score R^2 is 0.508 and Residual Standard Error is 0.768. We used Leave-One-Out-Cross-Validation (LOOCV) to estimate the test error of the adjusted gross model and K-fold cross validation with 10 folds to predict the test error of the IMDB score model.



```
Coefficients:
                            Estimate Std. Error t value Pr(>|t|)    
(Intercept)                5.494e+01  2.934e+00  18.726  < 2e-16 ***
num_critic_for_reviews     3.272e-03  1.907e-04  17.159  < 2e-16 ***
duration                   4.388e-03  7.858e-04   5.585 2.53e-08 ***
director_facebook_likes    8.777e-07  4.368e-06   0.201  0.84076    
actor_3_facebook_likes     1.974e-05  2.003e-05   0.986  0.32432    
actor_1_facebook_likes     3.348e-05  1.214e-05   2.757  0.00586 ** 
num_voted_users            2.972e-06  1.713e-07  17.354  < 2e-16 ***
cast_total_facebook_likes -3.072e-05  1.211e-05  -2.537  0.01123 *  
facenumber_in_poster      -1.839e-02  6.669e-03  -2.758  0.00584 ** 
num_user_for_reviews      -4.913e-04  6.102e-05  -8.051 1.14e-15 ***
title_year                -2.470e-02  1.487e-03 -16.612  < 2e-16 ***
actor_2_facebook_likes     2.982e-05  1.285e-05   2.321  0.02035 *  
aspect_ratio              -2.878e-02  2.261e-02  -1.273  0.20327    
movie_facebook_likes      -3.104e-06  9.478e-07  -3.275  0.00107 ** 
budget2016                -3.514e-10  1.229e-09  -0.286  0.77489    
gross2016                  3.257e-09  7.674e-10   4.244 2.25e-05 ***
is_biographyTRUE           2.086e-01  6.750e-02   3.091  0.00201 ** 
is_comedyTRUE             -4.568e-02  3.563e-02  -1.282  0.19990    
is_crimeTRUE               7.081e-03  4.026e-02   0.176  0.86040    
is_documentaryTRUE         9.962e-01  1.263e-01   7.890 4.08e-15 ***
is_dramaTRUE               5.025e-01  3.294e-02  15.252  < 2e-16 ***
is_familyTRUE              5.788e-02  6.422e-02   0.901  0.36753    
is_fantasyTRUE            -8.112e-02  4.323e-02  -1.876  0.06068 .  
is_historyTRUE            -5.192e-02  8.987e-02  -0.578  0.56347    
is_horrorTRUE             -3.541e-01  4.972e-02  -7.121 1.32e-12 ***
is_musicTRUE              -6.651e-02  5.782e-02  -1.150  0.25007    
is_mysteryTRUE             1.014e-01  4.879e-02   2.078  0.03775 *  
is_romanceTRUE             8.990e-04  3.518e-02   0.026  0.97961    
is_scifiTRUE              -1.913e-01  4.430e-02  -4.318 1.62e-05 ***
is_sportTRUE               8.176e-02  7.194e-02   1.136  0.25583    
is_thrillerTRUE           -1.527e-01  3.821e-02  -3.998 6.54e-05 ***
is_warTRUE                -1.340e-02  8.458e-02  -0.158  0.87413    
is_westernTRUE            -3.642e-02  1.005e-01  -0.362  0.71709    
is_GTRUE                  -4.649e-02  1.176e-01  -0.395  0.69254    
is_PGTRUE                 -1.508e-01  7.865e-02  -1.917  0.05529 .  
is_PG_13TRUE              -3.048e-01  7.414e-02  -4.111 4.03e-05 ***
is_RTRUE                  -1.375e-02  7.072e-02  -0.194  0.84590    
is_colorTRUE              -1.818e-01  7.478e-02  -2.431  0.01512 *  
```

Figure 1: **Summary of linear regression model fit to predict IMDB score.** Significant variables include number of critic reviews, number of user votes, year, drama, adjusted gross and duration. Interestingly, increasing a movie's, gross, number of more critics, duration, or turning the movie into a drama or documentary will increase the predicted IMDB score (due to positive beta values) while increasing the release year, number of user reviews, movie Facebook likes, or turning the movie into a Sci-Fi, thriller or PG-13 movie will decrease predicted score (assuming all other variables are held constant).

4.1.1 Cross-Validation

LOOCV is a good method to estimate test error since for each of the n data points, a linear model is fitted on the remaining $n-1$ points, then the squared error for the withheld point is calculated between the actual result and the predicted result from the model. Then the squared errors are added over all points to estimate test error. A big advantage of this method is that there is no randomness so overall there is less variability (i.e. returns the same value no matter how many times you run it). However, since a model needs to be fit n times were n is the number of observations, LOOCV can take a long time and uses a significant amount of computing power. In this case, the English movies from United States data with only the major content ratings (G, PG, PG-13, R) is on the magnitude of 3,000 data points which is reasonable for a regular computer to run. The mean squared error for the LOOCV method for predicting gross is high, around 3.054393e+14. For other methods we used variations of k-fold cross validation because of its reduced running time and computing power. This method divides the data into k approximately equal sections, and for each of the k folds, we train the model on the other $k-1$ folds and then predict the values for the withheld fold. Using 10-fold cross validation, we a got a low error of around 0.6 for our linear model that predicts IMDB score.

4.2 Subset Selection

4.2.1 Forward and Backward Selection to Predict Gross

We wanted to reduce the flexibility of the model due to the high number of predictors (i.e. 38) and hopefully reduce the test error. We were also interested in learning what variables are not significant.

We attempted greedy forward and backward subset selection using the Bayesian Information Criterion (BIC) as the test statistic to predict gross. Forward selection greedily adds one variable at each step and backward selection greedily remove one variable at each step. At each step it considers all models that add/remove a variable and chooses the one that gives the best improvement (i.e. leads to the lowest BIC). After implementing this algorithm the resulting model includes the predictors for the number of critic reviews, number of Facebook likes for the top three actors, the number of users who voted, the total Facebook likes for the cast, content ratings PG, PG-13, and R, the adjusted budget, movie year, IMDB score, and the genres Drama and History. The associated BIC score for the final forward-backward subset selection model is 112,426.

4.2.2 Best Subset Selection to Predict Gross

Forward and backward subset section only finds local solutions. We were interested in finding the global solution so we implemented best subset regression with the Bayesian Information Criterion (BIC) to predict gross. Best subset selection tests all possible combinations of the predictors and finds the model with the lowest training error. The BIC is a metric that we want to minimize and consists of the Residual Sum of Squares as well as a factor for the number of predictors in the model. In Equation 1, the k represents the number of predictors in the model, n is the number of data points in the sample, and $\hat{L}$ is the maximized value of the likelihood function of the model.

$$BIC = k * ln(n) - 2ln(\hat{(L)}) \tag{1}$$

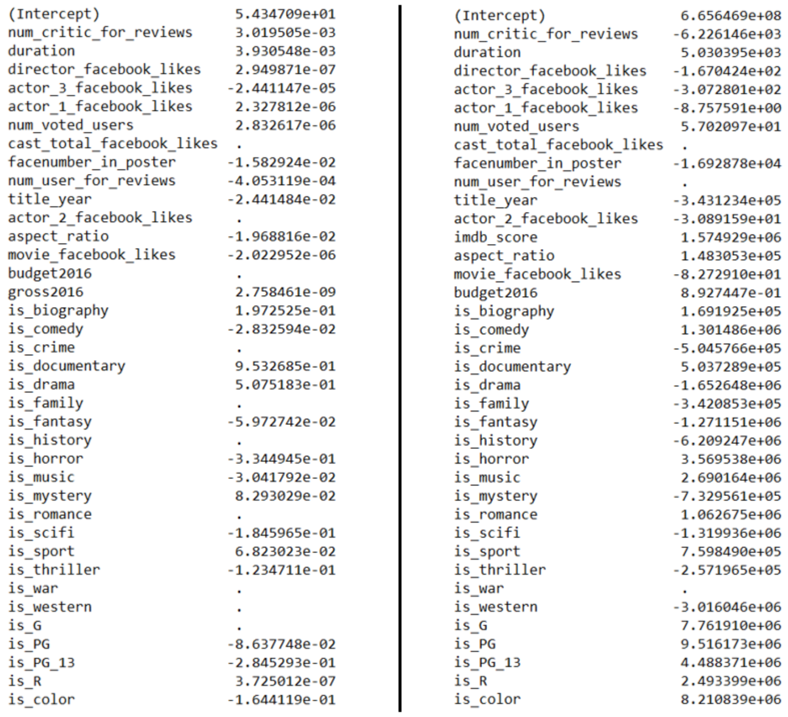
The resulting model uses the predictors for the number of users who voted, the year, the IMDB score, the number of Facebook likes that the movie has, the adjusted budget, whether or not the movie is in color, the ratings G, PG, and PG-13, and whether the genre of the movie includes Comedy, History, or Horror. This model performs similarly to the model that fit all the predictors to the data using LOOCV. However, this algorithm is very computationally heavy since it checks every possible number and combination of variables, 2^K models need to be fitted to the data where K is the total number of predictors that we could add to the model.

We also used k-fold Cross validation; LOOCV is a variation of k-fold Cross validation where the number of folds is n, the number of data points. This method divides the data into k approximately equal sections, and for each of the k folds, we train the model on the other $k-1$ folds and then predict the values for the withheld fold. since we have about 3000 observations, a reasonable number for k is 5 (since each fold would contain about 600 data points) and the resulting model only includes predictors for the number of voted users, movie year, and the adjusted budget. Note that this is a much simpler model than the previously computed models as there are only 3 predictors.

4.2.3 Lasso Regularization

Lasso is a technique that performs both variable selection and regularization in order to reduce flexibility and enhance the prediction accuracy and interpretability. Lasso adds a penalty term that shrinks the coefficient estimates to 0, thus inducing sparsity. As mentioned before having too many variables creates a challenge in model interpretation, especially when the number of variables is large. Given the large number of variables in our dataset we were worried about overfitting and decided to attempt Lasso regularization. To perform

Lasso regularization we made sure all predictors were normalized. We combined Lasso regularization with 10-fold cross validation to tune the "penalty parameter" and then estimated the beta coefficients and test error. Lasso regularization performed just as well as basic linear regression in predicting IMDB score and performed worse than linear regression in predicting adjusted gross. The optimal "penalty parameter" was 53,343.78 and 0.004617489 for the adjusted gross and IMDB score model, respectively.



```
(Intercept)                5.434709e+01
num_critic_for_reviews     3.019505e-03
duration                   3.930548e-03
director_facebook_likes    2.949871e-07
actor_3_facebook_likes    -2.441147e-05
actor_1_facebook_likes     2.327812e-06
num_voted_users            2.832617e-06
cast_total_facebook_likes  .
facenumber_in_poster      -1.582924e-02
num_user_for_reviews      -4.053119e-04
title_year                -2.441484e-02
actor_2_facebook_likes     .
aspect_ratio              -1.968816e-02
movie_facebook_likes      -2.022952e-06
budget2016                 .
gross2016                  2.758461e-09
is_biography               1.972525e-01
is_comedy                 -2.832594e-02
is_crime                   .
is_documentary             9.532685e-01
is_drama                   5.075183e-01
is_family                  .
is_fantasy                -5.972742e-02
is_history                 .
is_horror                 -3.344945e-01
is_music                  -3.041792e-02
is_mystery                 8.293029e-02
is_romance                 .
is_scifi                  -1.845965e-01
is_sport                   6.823023e-02
is_thriller               -1.234711e-01
is_war                     .
is_western                 .
is_G                       .
is_PG                     -8.637748e-02
is_PG_13                  -2.845293e-01
is_R                       3.725012e-07
is_color                  -1.644119e-01
```

```
(Intercept)                6.656469e+08
num_critic_for_reviews    -6.226146e+03
duration                   5.030395e+03
director_facebook_likes   -1.670424e+02
actor_3_facebook_likes    -3.072801e+02
actor_1_facebook_likes    -8.757591e+00
num_voted_users            5.702097e+01
cast_total_facebook_likes  .
facenumber_in_poster      -1.692878e+04
num_user_for_reviews       .
title_year                -3.431234e+05
actor_2_facebook_likes    -3.089159e+01
imdb_score                 1.574929e+06
aspect_ratio               1.483053e+05
movie_facebook_likes      -8.272910e+01
budget2016                 8.927447e-01
is_biography               1.691925e+05
is_comedy                  1.301486e+06
is_crime                  -5.045766e+05
is_documentary             5.037289e+05
is_drama                  -1.652648e+06
is_family                 -3.420853e+05
is_fantasy                -1.271151e+06
is_history                -6.209247e+06
is_horror                  3.569538e+06
is_music                   2.690164e+06
is_mystery                -7.329561e+05
is_romance                 1.062675e+06
is_scifi                  -1.319936e+06
is_sport                   7.598490e+05
is_thriller               -2.571965e+05
is_war                     .
is_western                -3.016046e+06
is_G                       7.761910e+06
is_PG                      9.516173e+06
is_PG_13                   4.488371e+06
is_R                       2.493399e+06
is_color                   8.210839e+06
```

Figure 2: **Coefficients for Lasso model.** Coefficients for predicting IMDB score are on the left and coefficients for predicting adjusted gross are on the right. After running Lasso we had a better idea of the variables that are and are not important in predicting IMDB score. In the chart above, variables that are not important had their coefficients reduced to 0 (indicated by a dot).

4.3 Logistic Regression to Predict Adjusted Gross Exceeding the 90^{th} Percentile

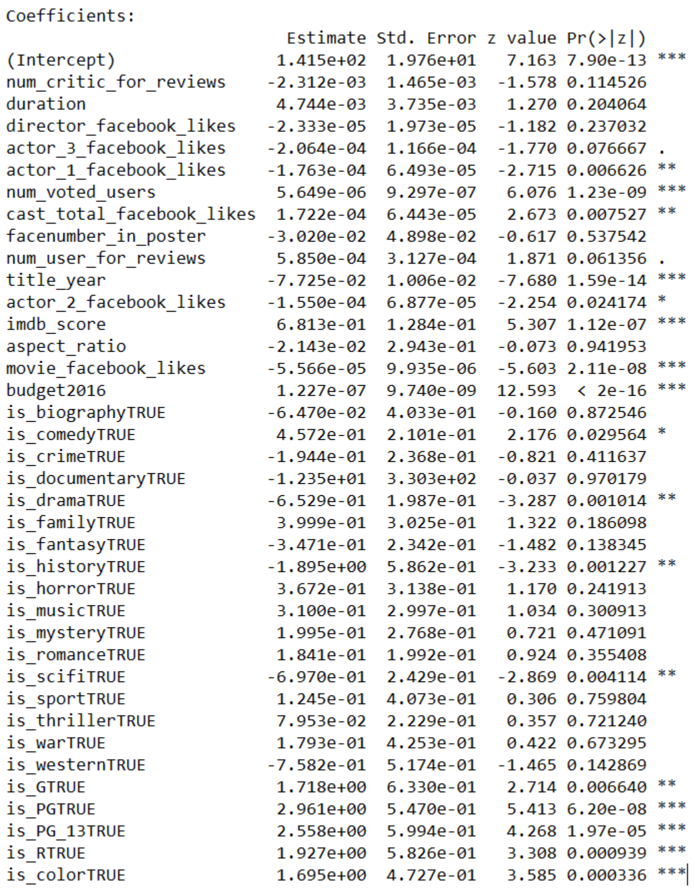
We performed logistic regression, a variation of linear regression. Instead of estimating the adjusted gross amounts directly, we estimate the log-odds ratio and checked whether it exceeded a threshold. For English movies from the United States, the 90^{th} percentile of adjusted gross amounts is \$29,723,854. Using this value we want to create a model that estimates whether or not new movies will be in the top 10% of movies based on adjusted gross amounts. The motivation behind this model is that stakeholders for films do not care as much about estimating the exact amount that the movie will make, rather they just want to know if it will be "successful". The variable Top_10_Movies is added where the value for a movie is equal to 1 or TRUE if the movie exceeds the 90^{th} percentile that we defined earlier, and 0 otherwise. Then LOOCV, best subset regression, and forward and backward subset selection was implemented, similar to the linear regression approach.

We fitted almost all variables using logistic regression to predict whether or not the adjusted gross amount exceeds the threshold, the model predictors used were number of Facebook likes that the top two actors have, the number of users that voted, the total Facebook likes that the cast has, the movie year, the IMDB score, the number of Facebook likes that the movie has, the adjusted budget, the sub genres Comedy, Drama, History, Sci-Fi, whether or not the movie is in color, and the content ratings G, PG, PG-13, and R. The misclassification rate for this model using a threshold probability of 0.5 is $\approx 7.7\%$. A threshold of 0.5 implies that we classify a prediction as TRUE if the probability is at least 0.5. Computing LOOCV on the logistic regression model from above, we get a misclassification rate of $\approx 5.9\%$. This result implies a good fit to the model since LOOCV provides an estimate of the test error of the model.

Running LOOCV takes a long time so we also considered using the k-fold cross validation method with $k = 5$. The resulting model only has 3 predictors: the number of users that voted, the movie year, and the adjusted budget and is consistent with the size of the model of the 5-fold cross validation method for the regression case to estimate the adjusted gross amount.

We implemented the less computationally expensive forward-backward subset selection on the data to predict the response for whether or not a movie will exceed the 90^{th} percentile, the resulting model had 13 predictors. These predictors are the number of users that voted, the movie year, the IMDB score, the number of Facebook likes that the movie has, the adjusted budget, sub genres Drama, History, and Sci-Fi, whether or not the movie is in color, and the content ratings (i.e G, PG, PG-13, and R). The associated BIC value for this model is 1276.3. The misclassification rate for the resulting model is $\approx 7.8\%$.

We then computed the best subset using the BIC as the criterion to get the global solution. The resulting model has 12 predictors: the number of users that voted, the year, the IMDB score, the number of Facebook likes that the movie has, the adjusted budget, the sub genres comedy, history, and horror, whether or not the movie is in color, and the ratings G, PG, and PG-13. This set of predictors is quite similar to the set of significant predictors that resulted from the model that used all predictors.



```
Coefficients:
                          Estimate Std. Error z value Pr(>|z|)    
(Intercept)              1.415e+02  1.976e+01   7.163 7.90e-13 ***
num_critic_for_reviews  -2.312e-03  1.465e-03  -1.578 0.114526    
duration                 4.744e-03  3.735e-03   1.270 0.204064    
director_facebook_likes -2.333e-05  1.973e-05  -1.182 0.237032    
actor_3_facebook_likes  -2.064e-04  1.166e-04  -1.770 0.076667 .  
actor_1_facebook_likes  -1.763e-04  6.493e-05  -2.715 0.006626 ** 
num_voted_users          5.649e-06  9.297e-07   6.076 1.23e-09 ***
cast_total_facebook_likes 1.722e-04  6.443e-05   2.673 0.007527 ** 
facenumber_in_poster    -3.020e-02  4.898e-02  -0.617 0.537542    
num_user_for_reviews     5.850e-04  3.127e-04   1.871 0.061356 .  
title_year              -7.725e-02  1.006e-02  -7.680 1.59e-14 ***
actor_2_facebook_likes  -1.550e-04  6.877e-05  -2.254 0.024174 *  
imdb_score               6.813e-01  1.284e-01   5.307 1.12e-07 ***
aspect_ratio            -2.143e-02  2.943e-01  -0.073 0.941953    
movie_facebook_likes    -5.566e-05  9.935e-06  -5.603 2.11e-08 ***
budget2016               1.227e-07  9.740e-09  12.593  < 2e-16 ***
is_biographyTRUE        -6.470e-02  4.033e-01  -0.160 0.872546    
is_comedyTRUE            4.572e-01  2.101e-01   2.176 0.029564 *  
is_crimeTRUE            -1.944e-01  2.368e-01  -0.821 0.411637    
is_documentaryTRUE      -1.235e+01  3.303e+02  -0.037 0.970179    
is_dramaTRUE            -6.529e-01  1.987e-01  -3.287 0.001014 ** 
is_familyTRUE            3.999e-01  3.025e-01   1.322 0.186098    
is_fantasyTRUE          -3.471e-01  2.342e-01  -1.482 0.138345    
is_historyTRUE          -1.895e+00  5.862e-01  -3.233 0.001227 ** 
is_horrorTRUE            3.672e-01  3.138e-01   1.170 0.241913    
is_musicTRUE             3.100e-01  2.997e-01   1.034 0.300913    
is_mysteryTRUE           1.995e-01  2.768e-01   0.721 0.471091    
is_romanceTRUE           1.841e-01  1.992e-01   0.924 0.355408    
is_scifiTRUE            -6.970e-01  2.429e-01  -2.869 0.004114 ** 
is_sportTRUE             1.245e-01  4.073e-01   0.306 0.759804    
is_thrillerTRUE          7.953e-02  2.229e-01   0.357 0.721240    
is_warTRUE               1.793e-01  4.253e-01   0.422 0.673295    
is_westernTRUE          -7.582e-01  5.174e-01  -1.465 0.142869    
is_GTRUE                 1.718e+00  6.330e-01   2.714 0.006640 ** 
is_PGTRUE                2.961e+00  5.470e-01   5.413 6.20e-08 ***
is_PG_13TRUE             2.558e+00  5.994e-01   4.268 1.97e-05 ***
is_RTRUE                 1.927e+00  5.826e-01   3.308 0.000939 ***
is_colorTRUE             1.695e+00  4.727e-01   3.585 0.000336 ***
```

Figure 3: **Summary of logistic regression model fit to predict whether adjusted gross exceeds the 90^{th} Percentile** Significant variables include budget, year, number of voted users and IMDB score. All of these variables except year increase the log-odds (assuming all other variables are held constant.

4.4 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors algorithm is a supervised but nonlinear learning technique for either regression or classification settings. The only parameter needed for this algorithm is the integer k which is the number of neighbors used to predict the response of a new point. A variety of distance functions can be used such as Euclidean distance for real values (or 2-norm), hamming distance or absolute distance (1-norm). If k is small then the model may be too flexible for estimating new points but if k is too large, then the model may be too inflexible and just return the values close to the average. Typically, k is chosen by cross validation to estimate and minimize the test error. Variations of K-Nearest-Neighbors can also weight the importance of neighbors by the proximity to the new point, where a common weight is $\frac{1}{d}$ where d is the distance from the new point to the neighbor. These variations are not investigated in the scope of this paper. When implementing the KNN algorithm, we first scaled the data to have mean of 0 and standard deviation of 1 this way some predictors will not dominate the result if their variances are higher than the other predictors.

In the regression case, the response of a new point is the average of the responses from the k nearest neighbors based on the distance calculation. For the classification case, the response is chosen by the "voting" technique where instead of averaging the responses of the k nearest neighbors, the majority class of the k nearest neighbors will be chosen for the new data point. This method applies to response variables with multiple classes as well.

We performed the regression method of KNN to predict IMDB score. A scaled version of the data was divided into a training and test set where the size of the training set is approximately two-thirds of the data set. We found the optimal k to be $k = 10$ from testing 1,3,5,10,20 and 50 as possible values of k. A plot of error vs. k value is shown in Figure 4. The error found with $k = 10$ was 0.8098. This is worst than the linear regression and Lasso methods.

We then used the same method to predict adjusted gross. Dividing the scaled data into a training and test set where the size of the training set is approximately two-thirds of the data set. This time we found the optimal k to be $k = 3$ from testing 1,3,5,10,20 and 50 as possible values of k. A plot of error vs. k value is shown in Figure 4. The error was found to be 2.714498e+14 which is better then the previously attempted methods.

When predicting whether or not the adjusted gross value exceeds the 90^{th} percentile, we used the classification method of KNN. Again, the scaled data was divided into a training and test set with the same sizes as before. Once we found k, we ran LOOCV on the test set fitting a KNN model with $k = 10$ at each iteration.
The resulting misclassification rate is $\approx 9.4\%$. A plot of error by value of k is shown in Figure 4.

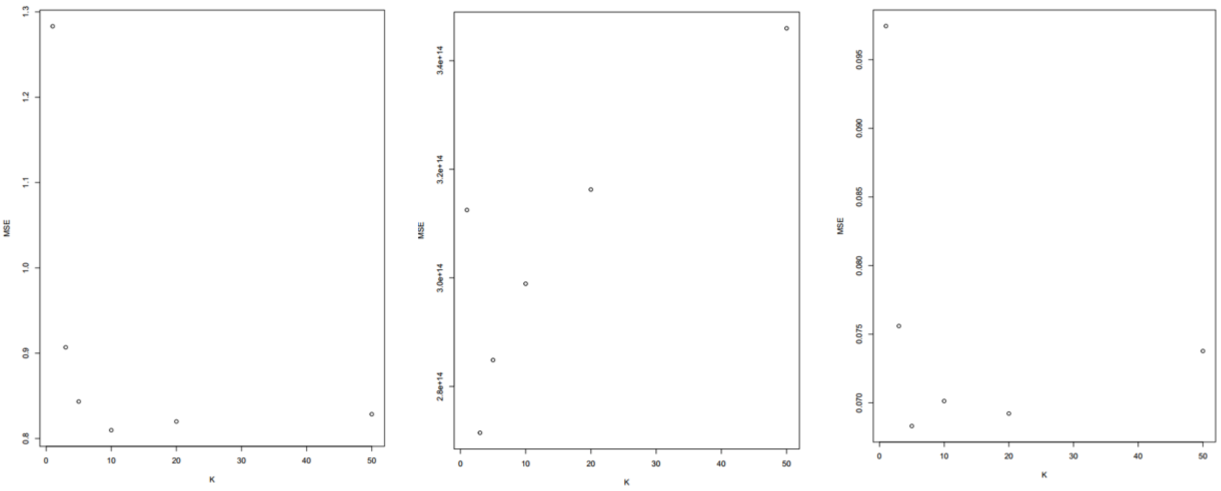


Figure 4: From left to right: error rates for various K when predicting IMDB score, adjusted gross and top 10^{th} percentile

4.5 Random Forest

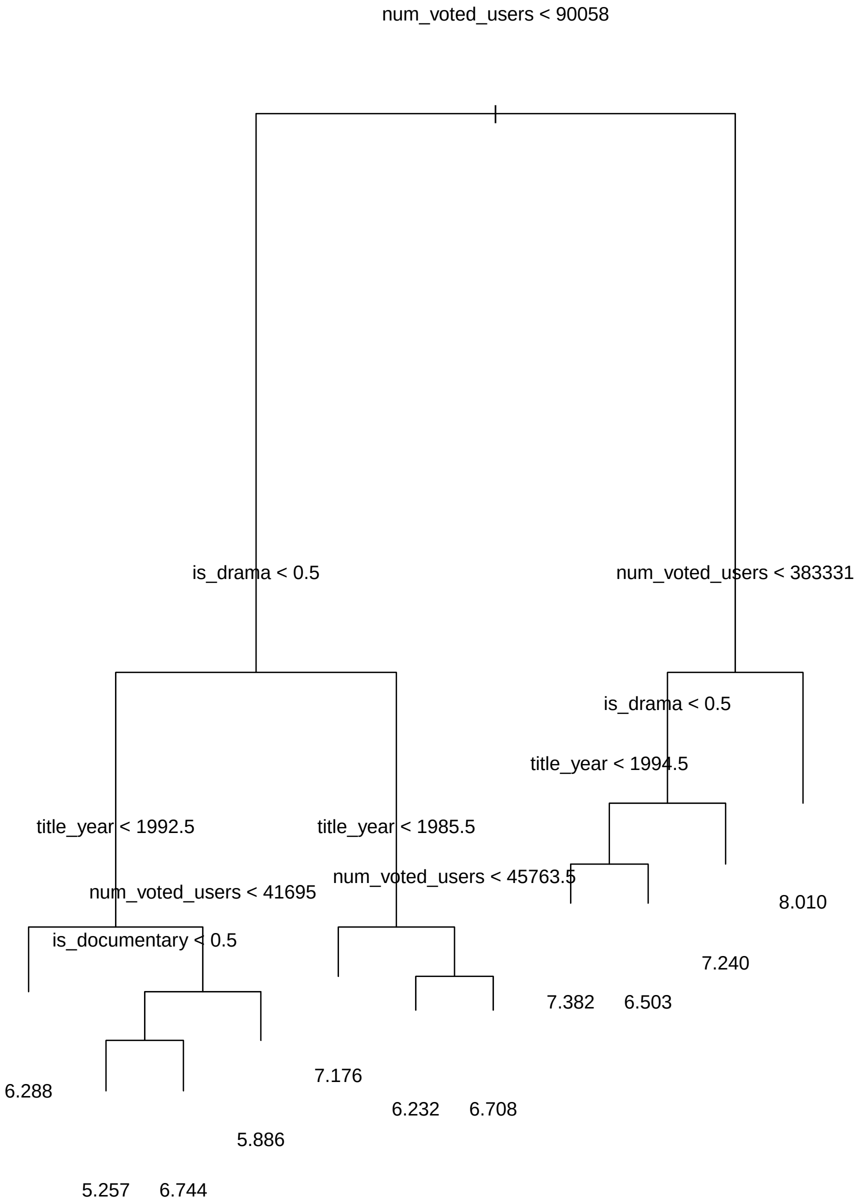


Figure 5: We wanted to try a model that is easy to interpret so we fitted a pruned decision tree to predict IMDB score. The tree with 12 terminal nodes had the lowest cross-validation error rate with mean squared error of 0.6675. The more important variables appear in the earlier/higher splits such as number of voted users and drama.

We were interested in learning whether an unsupervised, non-linear method like random forest would outperform the other previously mentioned methods in predicting adjusted gross, top 10^{th} percentile gross and IMDB score. We were also interested in learning which variables are the most important in terms of improvement in accuracy of predictions.

Random Forest requires two parameters: the number of bootstrapped training sets used (ntree) and the number of predictors to consider at each split (mtry). We used the default (500) number of trees and used cross validation to determine the optimal number of predictors to consider at each split. We did not use cross validation for the number of trees since it is not as important so long as the number of trees is large. Below we plotted the cross validation error (MSE) for values of mtry varying from 1 to 38 to predict adjusted gross. We did the same to predict IMDB score and whether a movie gross was above the 90th percentile, except in this case using the mis-classification rate. The lowest cross validation error was achieved with 30, 26 and 11 as mtry for IMDB score, gross and the 90^{th} percentile respectively.

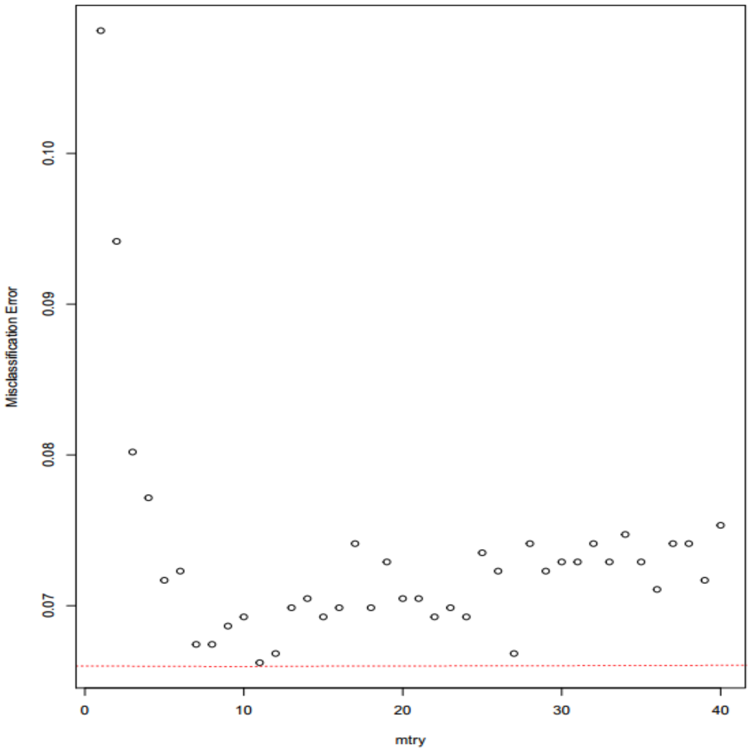


Figure 6: Error Rates for various values of mtry on the cross validation data when predicting whether adjusted gross amount falls in the top 10th percentile. The red dotted line denotes the lowest MSE.

We were also able to take a look at which variables are the most significant. The results were similar for predicting adjusted gross and the indicator for being above the 90^{th} percentile of adjusted gross. Adjusted budget, number of user votes, IMDB score and movie Facebook likes were among the most important predictors. Surprisingly, the title year and number of user reviews were also important to determining the adjusted film gross. For improving IMDB score prediction accuracy, the most important variables were number of user votes, drama, title year, number of user reviews and film duration. This time budget, which was the most important variable for gross, was not as important, meaning that budget is much more important when predicting gross than IMDB score.

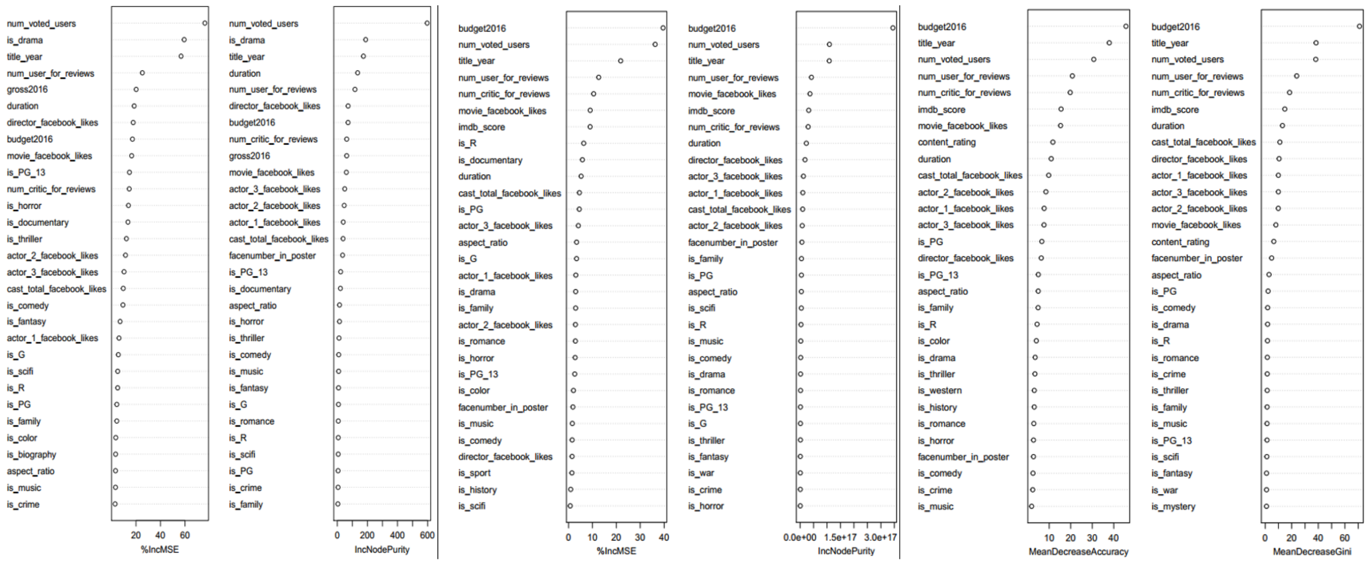
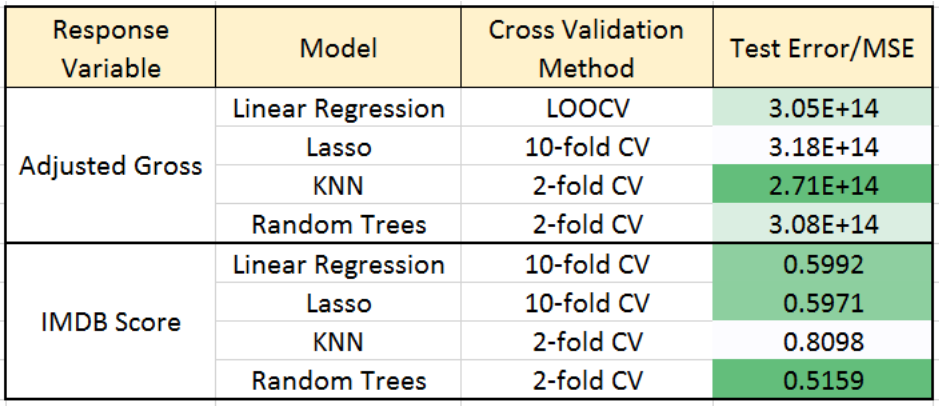


Figure 7: From left to right: important variables for predicting IMDB score, adjusted gross and top 10th percentile of gross.

5 Results Summary



Response Variable	Model	Cross Validation Method	Test Error/MSE
Adjusted Gross	Linear Regression	LOOCV	3.05E+14
	Lasso	10-fold CV	3.18E+14
	KNN	2-fold CV	2.71E+14
	Random Trees	2-fold CV	3.08E+14
IMDB Score	Linear Regression	10-fold CV	0.5992
	Lasso	10-fold CV	0.5971
	KNN	2-fold CV	0.8098
	Random Trees	2-fold CV	0.5159

Summary of Prediction Models for the Top 10^{th} Percentile Response

Model	Cross Validation Method	Misclassification Rate
Logistic Regression	None	7.7%
Logistic Regression	LOOCV	5.9%
Logistic Regression and Best Subset Selection	None	7.8%
Logistic Regression and Forward Backward Subset Selection	None	7.8%
KNN	LOOCV	9.4%
Random Forest	2-fold CV	6.7%

6 Conclusions

The best model for predicting IMDB score based on test error is a random trees model that considers 30 variables at each split. The test error for this model is 0.516 which is quite small. We recommend using this model over linear regression, Lasso and KNN in the future. Figure 7 shows which variables are most important in respect to accuracy when using this model. Number of user votes, drama, title year and duration are among the most important variables. These variables also appeared as splits in our fitted pruned tree (see figure (5). From the linear regression model, we learned that increasing number of user votes, duration and being a drama lead to higher predicted scores (assuming all other variables are held constant). Among these variables drama had the highest estimated beta coefficient value with a value of 0.502, meaning that switching a movie's genre to drama increases the predicted IMDB score by 0.502.

The best model for predicting adjusted gross based on test error is linear regression. However the test error for all models are quite large so it might be better to instead predict whether gross will exceed a threshold, such as the 90^{th} percentile. Exceeding the 90^{th} percentile will tell you whether a movie could be a hit. Logistic regression performed the best in predicting whether gross will be in the top 10^{th} percentile so we would recommend using this model to predict that. The misclassification error was quite small so its results are reliable.

The logistic regression model to predict whether or not a movie's adjusted gross value is in the top 10% has a lower misclassification rate than using K-Nearest-Neighbors for classification and random forest. In addition there was not a significant improvement in the misclassification rate when implementing best subset selection versus simple logistic regression with LOOCV.

For full R code and scripts go to https://github.com/gloriadazevedo/IMDB_Data_Mining_Project to view R code and corresponding code comments.