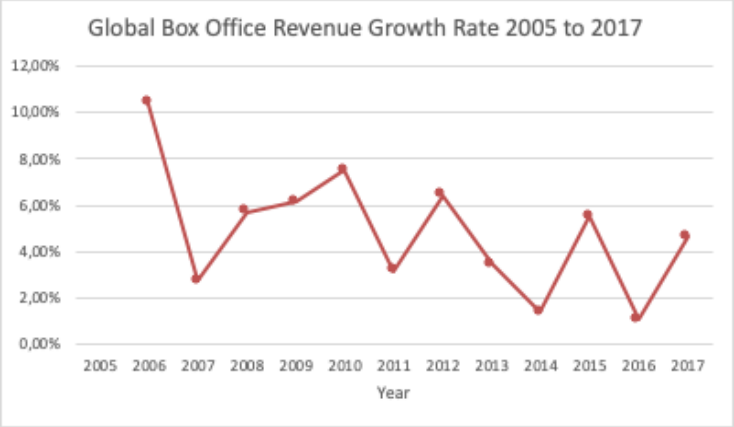
What Kind of Movie Is Worth the Investment?
- A Box Office Revenue Analysis

Authors : Suyu Chen, Claire Feng, Siqi Li, Shefali Pradeep Mhadadalkar, Yizhou Zhang
Columbia University
IEOR 4650 Business Analytics

Introduction

The film industry by nature has a low rate of return - from scripting and casting to finally reaching the cinema, the process usually takes months and occasionally even years. To add onto the pressure, recent years have seen a stagnation in box office revenue growth.



(Data Source: MPAA)

With the rising popularity of online streaming, increasingly more people prefer enjoying movies in the comfort of their home to sitting inside a movie theater with the AC cranked up too high and the air filled with the smell of popcorn and chicken nuggets. It is therefore crucial for production companies to invest smartly: Would action movie bring in more cash than romance? Maybe acting skills not matter much in generating revenue? Could movie in English sell better than movie in foreign language? To answer such questions, we delved into analyzing historical data on movies across different genres and languages. We hope that our results could help studios better decide on how to allocate their resources at the birth stage of future productions.

Data

For this project, we manipulated data scraped from the IMDb website. Our base dataset is a pre-compiled list of 5000+ movies. These 5000+ movies range from 1916 to 2016, from action to documentary, from English to Persian. To adapt the data to analytical models, we took several steps to reconstruct the set.

It is common sense to assume that the director and cast of a movie impact its box office turnout. To transform the crew's names into variables suitable for quantitative models, we decided to calculate scores for each movie's director and first three actors/actresses. Each director's score corresponds to the average IMDb rating of his/her works. Each actor/actress' score corresponds

to the average IMDb rating of the movies in which he/she starred. Each movie then receives a cast score taken as the average of its first three actors/actresses' scores. By using Python and Excel, we matched as many directors and actors/actresses mentioned in our base dataset as possible with the rated movies in the IMDb datasets. The resulting average scores range from 0 to 10. Hopefully, our models could tell us whether an actor with a horrible 3 out of 10 could actually ruin the box office of his movie.

Remove irrelevant variables

After supplementing our base dataset with these average scores, we moved towards removing variables assumed to be irrelevant or unreliable. Since we now have scores for the crew, we removed columns concerning the names of directors and actors/actresses. Some other variables we deemed irrelevant include the movie's IMDb link, plot keywords - which naturally correlate with genres, - aspect ratio, and more.

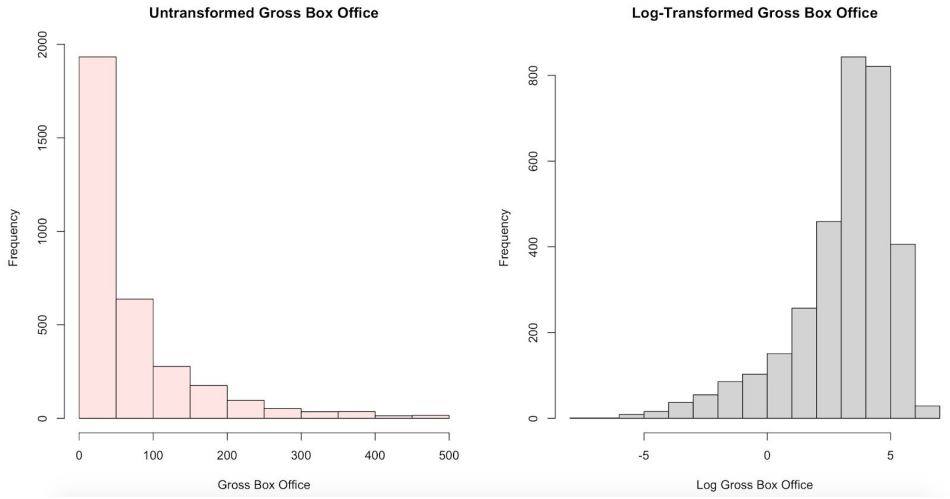
One interesting family of factors is Facebook Likes. We originally considered it a useful metric for a movie's popularity, which could affect the movie's box office revenue. Yet upon closer inspection, we decided to disregard data on Facebook Likes due to their unreliability. Not only a great number of movies lack data on such attributes, but the ranges are also big - considering some of the movies were released decades before Facebook's existence. Thus, we removed all columns pertaining to Facebook Likes. We also removed movies from before 1950 because, "Who would use data from a century ago to predict the performance of a movie today? " Besides eliminating movies before 1950, we didn't select IMDb scores' as one of x variables. IMDb scores always came out after the box office and didn't have much impact on the box office revenue. Thinking from a producer perspective, we believed that the scores were not useful to predict gross revenue.

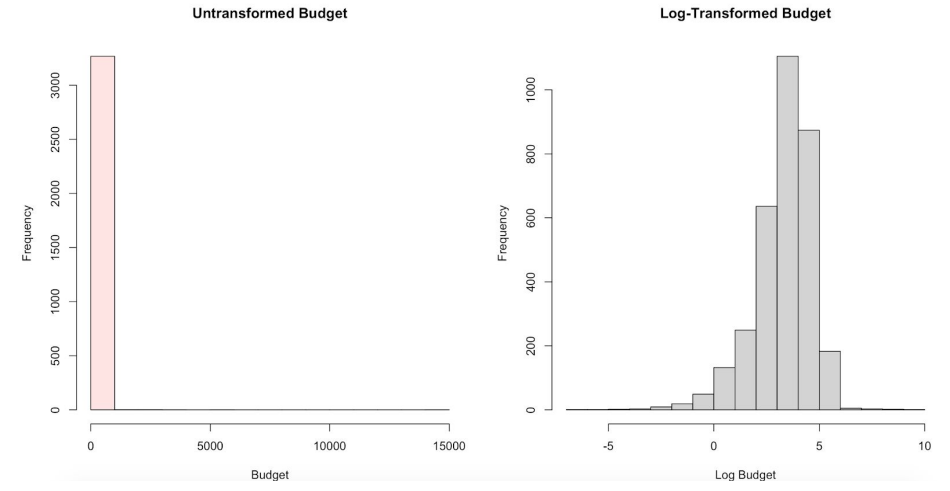
Build categorical variables

The next step was to transform categorical variables into binary/dummy variables appropriate for modelling. The categorical variables include: content_rating, duration, color, language, country, and genres. For variables that are already binary (i.e. color), we coded one attribute as 0 (i.e. Black & White) and the other as 1 (i.e. Color). For variables with multiple layers, we either created a dummy for each layer (i.e. for genres: is this an "adventure" movie? Yes = 1, No = 0) or further categorized them and then created dummies. Take the attribute "language" as an example. The thousands of entries in the dataset initially include 48 different languages. On one hand, there is English. On the other hand, there is Kannada, a regional language spoken in India. Since a vast majority of the movies recorded are in English, we created a dummy where English is coded as 1 and all other languages as 0. The model could then capture whether producing a movie in English increases box office revenue.

Convert budget and gross to present values & remove the outliers

One further problem lies within gross and budget variables, which we assumed to hold nominal values. Since we could not treat $1 in 1950 the same as $1 in 2016, we used the Consumer Price Index (CPI) to adjust for inflation so that all movies' gross and budget correspond to dollar value in 2016. The specific values and method used in this step could be found in Appendix A. (Sahr, 2017). After that, based on the histogram of gross, we considered the data points which had gross more than 500 millions as outliers and decided to remove them.Since the budget and gross data were very skewed even after converting the dollars values to present values, we continued to normalize our data by changing gross and budget to log(gross) and log(budget).





The result from countless sleepless nights and cups of coffee (and sweets)? A table with 3274 movies and 40 attributes.

Model

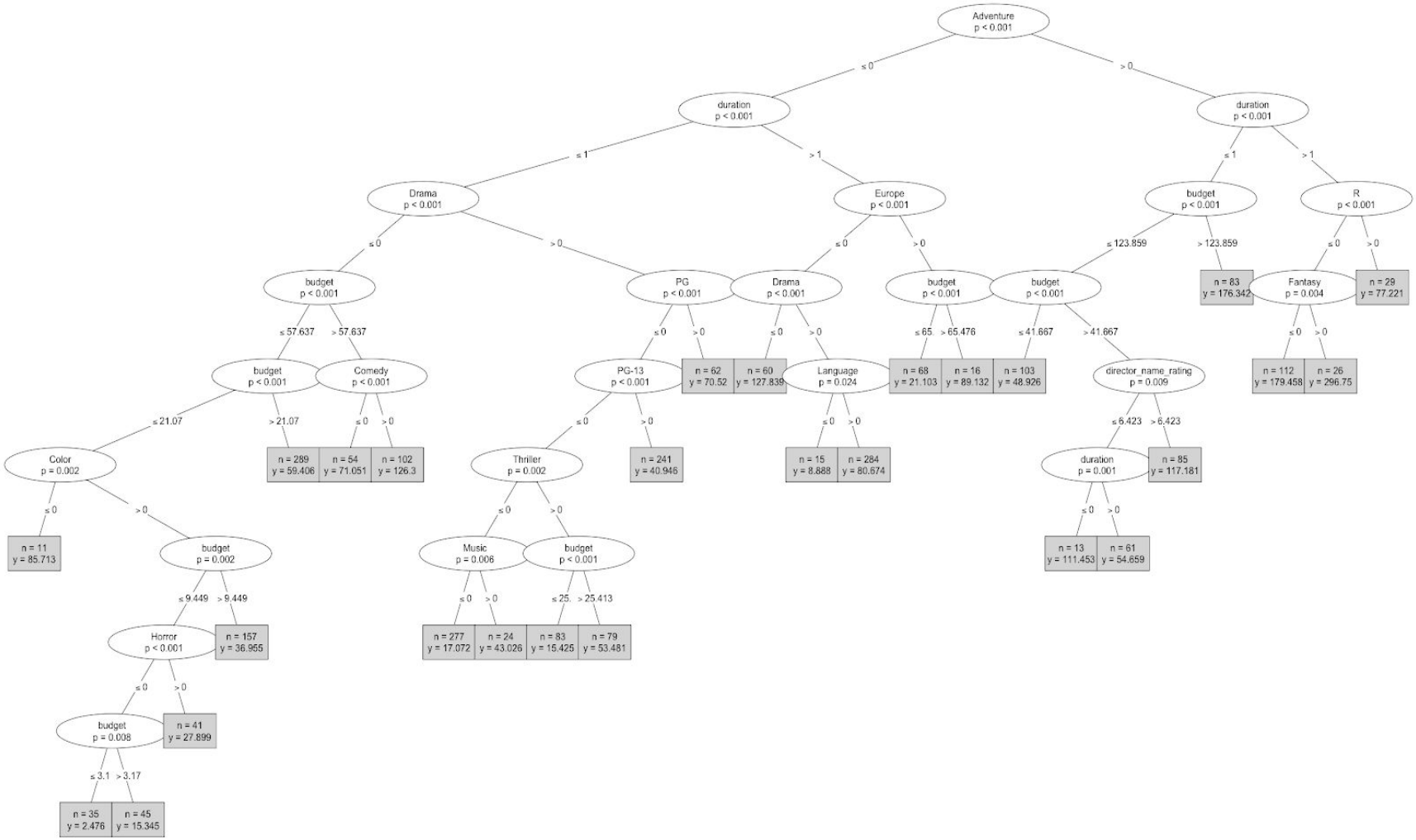
Best Subset and Lasso Variable selection

Right up the front, we fitted a multiple linear regression across all variables to use as our basis. The adjusted R squared we got is 0.4453 ,which was not good enough. The full model gave us 19 insignificant variables based on p values (see Appendix B1 for details). Instead of manually removing all those insignificant variables, we decided to apply some techniques to help us make such selection. We ran best subset selection under 10-fold cross validation. The best size with minimum mse was 37. The selection kept all variables we selected from the beginning and the model gives us an rmse of 1.24 (see Appendix B3 for details). We suspected the model given by best subset selection was overfitting and decided to move to Lasso regression under leave-one-out cross-validation (LOOCV) to see if it would remove any insignificant variables (see Appendix B2 for details). However, Lasso regression still gave us 37 variables. Considering the nature of our dataset, mainly consisted of categorical variables and skewed continuous variables even after log transformation. We decided to move to non-parametric method.

Random Forest

However, since our output variables of interest - gross box office - crowd toward below $100 million, we thought it helpful applying a model with less assumptions. The idea of decision tree came to light. A decision tree returns an easily interpretable prediction structure and could function like a regression without the need to transform data. It grows by factors that split input data into two groups with the most significantly different means. For example, in the figure from the following page, the tree first tries to split gross box office with every potential factor and find that when the movies were split by the genre *Action*, the two groups' means differ the most compared to splitting by other factors, such as duration, budget, and more. The tree grows by one layer and then repeat the process until further splitting would not create sufficiently different group means.

Taking one step further brings us to the idea of random forest. This algorithm could plant hundreds of trees of different sizes and structures. The forest grows by planting increasingly big trees while randomizing the factors considered - think of it as using different fertilizers on the same seed, and this randomization factor contributes to our confidence in future predictions. The growing process equates to the learning process of our old friend - linear regressions. The end product of a forest are the mean predictions from individual trees. A bundle of features would travel through the forest and come out - transformed! - as one gross box office value.


Adventure
p < 0.001
≤ 0
> 0
duration
p < 0.001
duration
p < 0.001
≤ 1
> 1
≤ 1
> 1
Drama
p < 0.001
Europe
p < 0.001
budget
p < 0.001
R
p < 0.001
≤ 0
> 0
≤ 0
> 0
≤ 123.859
> 123.859
≤ 0
> 0
budget
p < 0.001
PG
p < 0.001
Drama
p < 0.001
budget
p < 0.001
budget
p < 0.001
n = 83
y = 176.342
Fantasy
p = 0.004
n = 29
y = 77.221
≤ 57.637
> 57.637
≤ 0
> 0
≤ 0
> 0
≤ 65
> 65.476
≤ 41.667
> 41.667
≤ 0
> 0
budget
p < 0.001
Comedy
p < 0.001
PG-13
p < 0.001
n = 62
y = 70.52
n = 60
y = 127.839
Language
p = 0.024
n = 68
y = 21.103
n = 16
y = 89.132
n = 103
y = 48.926
director_name_rating
p = 0.009
n = 112
y = 179.458
n = 26
y = 296.75
≤ 21.07
> 21.07
≤ 0
> 0
≤ 0
> 0
≤ 0
> 0
≤ 6.423
> 6.423
Color
p = 0.002
n = 289
y = 59.406
n = 54
y = 71.051
n = 102
y = 126.3
Thriller
p = 0.002
n = 241
y = 40.946
n = 15
y = 8.888
n = 284
y = 80.674
duration
p = 0.001
n = 85
y = 117.181
≤ 0
> 0
≤ 0
> 0
≤ 0
> 0
n = 11
y = 85.713
budget
p = 0.002
Music
p = 0.006
budget
p < 0.001
n = 13
y = 111.453
n = 61
y = 54.659
≤ 9.449
> 9.449
≤ 0
> 0
≤ 25.
> 25.413
Horror
p < 0.001
n = 157
y = 36.955
n = 277
y = 17.072
n = 24
y = 43.026
n = 83
y = 15.425
n = 79
y = 53.481
≤ 0
> 0
budget
p = 0.008
n = 41
y = 27.899
≤ 3.1
> 3.17
n = 35
y = 2.476
n = 45
y = 15.345

Results and Interpretation

The random forest model sorted our variables with descending importance (Please see the full list with contribution to accuracy value in Appendix C) . Among all 30+ factors, the 10 most important ones by descending order are: *Budget, Duration, Family (genre), R (rating), Adventure (genre), Rating of the Director, Drama (genre), Action (genre), PG-13 (rating), and Europe (region).* Among binary predictors, genre is the most critical. The client should take those leading variables into account when deciding which movie to be put into production. For example, if a director came in with a basic movie proposal, the client could use those new data to predict gross revenue and eventually calculate potential profit margin. Another way to interpret the model results is by looking into a tree, such as the one above. The biggest output value is around 297 millions, implying that the most profitable movie would be characterized as an Adventure movie with duration greater than 1 (i.e. longer than 90 minutes), not R-rated and some fantasy elements involved. Clients can make better movie selection based on those higher expected gross revenue path.

Conclusion

Box office prediction is always a hot topic, yet limited number of analysis were from the views of producers on the market nowadays. Our report investigated the critical influences that will help company and producers making decisions in the early stage. Through the model and algorithm, producers are able to gain more market insights. This model answers the question such as what the new trend is? How big the profit scale will be for a specific movie.

Limitations

While the random forest algorithm is friendly to our dataset, it does come with much noises. Regression tree by nature cannot generate continuous output, even if the output variable is a continuous variable, such as gross box office in our case. Consider the final nodes as “exits” if we see a tree as a “maze.” When a new movie appears at the entrance of the maze, the Force of the tree - the algorithm - checks whether the movie has traits ‘Action’, ‘English’, ‘PG-13’ and sends it to “exit - $50 million” This movie now expects to gross $50 million, even though it might end with only $30 million and that the champion from this exit pockets $70 million. Therefore, when evaluating the performance of our tree, we are essentially comparing 50 numbers to thousands of different values, giving us a high rate of error. However, if we adjust our objective to know whether a movie with certain traits would be a blockbuster, of mild success, or a total flop, regression tree would be able to do a much better job. In fact, the prediction generated this way is more helpful to our client. We don’t need an exact prediction of the gross revenue, but a idea of how profitable this movie can be.

Future Directions

However, the success of one movie relies on more practical factors. From a survey in 2018, the largest source of revenue for production studios came from foreign distribution(36.1%), surprisingly, and distribution through additional forms(39.1%). Moving forward, it would be insightful to dig into features like the increasing effort of advertising, the impact from online streaming providers such as Netflix and Amazon Video, and studio name. We could also potentially train the model on bigger dataset to give our clients a more comprehensive system of gross revenue prediction.

Reference

MPAA. (n.d.). Global box office revenue from 2005 to 2017 (in billion U.S. dollars). In *Statista - The Statistics Portal.* Retrieved December 8, 2018, from https://www.statista.com/statistics/271856/global-box-office-revenue/

Sahr, Robert (2017). Conversion Factors in 2016 Dollars for 1774 to estimated 2027. In *Individual Year Conversion Factor Tables.* Retrieved December 9, 2018, from https://liberalarts.oregonstate.edu/sites/liberalarts.oregonstate.edu/files/polisci/faculty-research/sahr/inflation-conversion/pdf/cv2016.pdf

Robb, David (2018). U.S. Film Industry Topped $43 Billion In Revenue Last Year, Study Finds, *But It's Not All Good News. Retrieved December 11, 2018, from https://deadline.com/2018/07/film-industry-revenue-2017-ibisworld-report-gloomy-box-office-1202425692/*

Appendix A

Consumer Price Index (CPI) Conversion Factors for Dollars of 1774 to estimated 2027 to Convert to Dollars of 2016

CAUTION: Estimates for 2017-2027 are based on the average of OMB and CBO estimates as of early 2017. They will be revised in 2018.

To convert dollars of any year to dollars of the year 2016, DIVIDE the dollar amount from that year by the conversion factor (CF) for that year. For example, $1000 of 1945 = $13,333 dollars of 2016 ($1000 / 0.075). Rounding is strongly recommended.

Notes: Conversion factors are based on final 2016 annual average CPI: 2.40007, re-based so that 2016 = 1.000.

To reverse the process, that is, to determine what a 2016-dollar amount would be in dollars of another year, simply MULTIPLY the year 2016 amount by the conversion factor for that year. For example, $1000 of 2016 would be about $75 in dollars of 1945 ($1000 x 0.075 = $75).

Data series since 1912 have changed periodically, so numbers are not all precisely comparable. Therefore it is recommended that numbers be ROUNDED to four (or, more cautious, three) significant digits. So, $13,333 in the example above becomes $13,330 or $13,300. For years prior to 1913, rounding to three (or more cautious, two) significant digits is recommended, e.g. $13,333 becomes $13,300 or even $13,000. ALMOST ALWAYS, ROUNDING TO DOLLARS AND CENTS SUGGESTS MORE PRECISION THAN THE DATA ALLOW.

Year	CF	Year	CF	Year	CF	Year	CF	Year	CF	Year	CF	Year	CF
1774	0.034	1814	0.073	1854	0.035	1894	0.036	1934	0.056	1974	0.205	2014	0.986
1775	0.032	1815	0.064	1855	0.036	1895	0.035	1935	0.057	1975	0.224	2015	0.988
1776	0.036	1816	0.059	1856	0.035	1896	0.035	1936	0.058	1976	0.237	2016	1.000
1777	0.044	1817	0.055	1857	0.036	1897	0.035	1937	0.060	1977	0.252	2017	1.025
1778	0.057	1818	0.053	1858	0.034	1898	0.035	1938	0.059	1978	0.272	2018	1.049
1779	0.051	1819	0.053	1859	0.035	1899	0.035	1939	0.058	1979	0.302	2019	1.073
1780	0.057	1820	0.049	1860	0.035	1900	0.035	1940	0.058	1980	0.343	2020	1.097
1781	0.046	1821	0.047	1861	0.037	1901	0.035	1941	0.061	1981	0.379	2021	1.123
1782	0.050	1822	0.049	1862	0.042	1902	0.036	1942	0.068	1982	0.402	2022	1.149
1783	0.044	1823	0.044	1863	0.052	1903	0.037	1943	0.072	1983	0.415	2023	1.176
1784	0.042	1824	0.040	1864	0.065	1904	0.037	1944	0.073	1984	0.433	2024	1.204
1785	0.040	1825	0.041	1865	0.068	1905	0.037	1945	0.075	1985	0.448	2025	1.232
1786	0.040	1826	0.041	1866	0.066	1906	0.037	1946	0.081	1986	0.457	2026	1.261
1787	0.039	1827	0.042	1867	0.062	1907	0.039	1947	0.093	1987	0.473	2027	1.291
1788	0.037	1828	0.040	1868	0.059	1908	0.038	1948	0.100	1988	0.493		
1789	0.037	1829	0.039	1869	0.057	1909	0.038	1949	0.099	1989	0.517		
1790	0.038	1830	0.038	1870	0.055	1910	0.040	1950	0.100	1990	0.545		
1791	0.039	1831	0.036	1871	0.051	1911	0.040	1951	0.108	1991	0.567		
1792	0.040	1832	0.036	1872	0.051	1912	0.040	1952	0.110	1992	0.585		
1793	0.041	1833	0.035	1873	0.050	1913	0.041	1953	0.111	1993	0.602		
1794	0.046	1834	0.036	1874	0.047	1914	0.042	1954	0.112	1994	0.617		
1795	0.052	1835	0.037	1875	0.046	1915	0.042	1955	0.112	1995	0.635		
1796	0.055	1836	0.039	1876	0.045	1916	0.045	1956	0.113	1996	0.654		
1797	0.053	1837	0.040	1877	0.044	1917	0.053	1957	0.117	1997	0.669		
1798	0.051	1838	0.039	1878	0.042	1918	0.063	1958	0.120	1998	0.679		
1799	0.051	1839	0.039	1879	0.042	1919	0.072	1959	0.121	1999	0.694		
1800	0.052	1840	0.036	1880	0.042	1920	0.083	1960	0.123	2000	0.717		
1801	0.053	1841	0.036	1881	0.042	1921	0.075	1961	0.125	2001	0.738		
1802	0.045	1842	0.034	1882	0.042	1922	0.070	1962	0.126	2002	0.750		
1803	0.047	1843	0.031	1883	0.042	1923	0.071	1963	0.127	2003	0.767		
1804	0.049	1844	0.031	1884	0.041	1924	0.071	1964	0.129	2004	0.787		
1805	0.049	1845	0.032	1885	0.040	1925	0.073	1965	0.131	2005	0.814		
1806	0.051	1846	0.032	1886	0.039	1926	0.074	1966	0.135	2006	0.840		
1807	0.048	1847	0.034	1887	0.040	1927	0.072	1967	0.139	2007	0.864		
1808	0.052	1848	0.033	1888	0.040	1928	0.071	1968	0.145	2008	0.897		
1809	0.051	1849	0.032	1889	0.038	1929	0.071	1969	0.153	2009	0.894		
1810	0.051	1850	0.032	1890	0.038	1930	0.070	1970	0.162	2010	0.909		
1811	0.055	1851	0.032	1891	0.038	1931	0.063	1971	0.169	2011	0.937		
1812	0.055	1852	0.032	1892	0.038	1932	0.057	1972	0.174	2012	0.957		
1813	0.067	1853	0.032	1893	0.037	1933	0.054	1973	0.185	2013	0.971		

Revised June 9, 2017, using final 2016 CPI (CPI = 2.40007), from the Bureau of Labor Statistics, http://www.bls.gov/cpi/data.htm, "All Urban Consumers (Current Series)," January 2017. Note: The early 2016 average inflation estimate for 2016 by CBO and OMB was 1.40 percent. The actual (final) was 1.01 percent. INFLATION ASSUMPTIONS: Inflation conversion factors for 2017 and later assume 2.50% inflation in 2017, 2.30% in 2018 and 2019, 2.25% in 2020, and 2.35% each year 2021 through 2027. These are averages of OMB and CBO inflation estimates as of January (CBO) and May (OMB) 2017.Inflation assumptions: Inflation conversion factors for 2017 and later assume 2.50% inflation in 2017, 2.30% in 2018 and 2019, 2.25% in 2020, and 2.35% each year 2021 through 2027. These are averages of OMB and CBO inflation estimates as of January (CBO) and May (OMB) 2017.

CPI is CPI-U, the broader measure for all urban consumers, year-to-year average (not December to December).

Conversion factors for years before 1913 are re-based from data from the *Historical Statistics of the United States Millennial Edition* (Cambridge University Press, 2006). Calculation starting 1913 uses the CPI-U as the base, from the US Bureau of Labor Statistics. Monthly and annual CPI data are available at the BLS web site: http://stats.bls.gov/cpi/home.htm#data (CPI-U = all urban consumers).

CF denominated in years 1995 to estimated 2017 in Excel and pdf formats for dollars for years 1774 to estimated 2027 are posted at the online address indicated below.

Prior to the 2008 revision, a different data base was used for the period starting 1665 and ending 1913. See the main inflation conversion factor page for details.

The address of the inflation conversion factor web page is http://liberalarts.oregonstate.edu/spp/polisci/research/inflation-conversion-factors.

cv2016
Rev 06/12/2017

e-mail: Robert.Sahr@oregonstate.edu; home page: http://liberalarts.oregonstate.edu/spp/polisci/robert-sahr

Appendix B1 Linear Regression

Linear Regression

```
## Call:
## lm(formula = train_data$gross ~ ., data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0717 -0.5951  0.2050  0.9130  6.4521
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -2.639980   0.606948  -4.350 1.42e-05 ***
## duration                0.396882   0.059444   6.677 3.02e-11 ***
## facenumber_in_poster   -0.031355   0.017020  -1.842 0.065561 .
## budget                  0.697450   0.030562  22.821  < 2e-16 ***
## director_name_rating    0.180245   0.050828   3.546 0.000398 ***
## average_actor_rating   -0.048544   0.069517  -0.698 0.485054
## G                       2.280262   0.337462   6.757 1.76e-11 ***
## NC.17                   2.005458   0.506569   3.959 7.75e-05 ***
## Passed                  2.969486   0.575702   5.158 2.70e-07 ***
## PG                      2.027819   0.262989   7.711 1.82e-14 ***
## PG.13                   1.807260   0.242631   7.449 1.31e-13 ***
## R                       1.494478   0.235004   6.359 2.41e-10 ***
## Color                  -0.215075   0.169672  -1.268 0.205065
## Language                0.990801   0.191398   5.177 2.45e-07 ***
## Asia                   -1.263002   0.266618  -4.737 2.29e-06 ***
## Europe                 -0.832890   0.094533  -8.811  < 2e-16 ***
## Oceania                -0.629946   0.212658  -2.962 0.003084 **
## S_America              -0.481822   0.795669  -0.606 0.544866
## Adventure               0.055060   0.101393   0.543 0.587156
## Animation               0.009583   0.188880   0.051 0.959539
## Biography              -0.076765   0.144183  -0.532 0.594489
## Comedy                  0.230909   0.085581   2.698 0.007021 **
## Crime                  -0.223349   0.095446  -2.340 0.019362 *
## Documentary             0.773235   0.324953   2.380 0.017412 *
## Drama                  -0.332581   0.083046  -4.005 6.40e-05 ***
## Family                  0.180036   0.161965   1.112 0.266432
## Fantasy                -0.121487   0.106185  -1.144 0.252693
## History                -0.047172   0.179215  -0.263 0.792406
## Horror                  0.385969   0.130386   2.960 0.003104 **
## Music                   0.209292   0.170824   1.225 0.220624
## Musical                -0.716899   0.276837  -2.590 0.009666 **
## Mystery                 0.082939   0.115549   0.718 0.472959
## Romance                 0.074973   0.081458   0.920 0.357462
## SciFi                  -0.289042   0.106607  -2.711 0.006750 **
## Sport                  -0.213016   0.163298  -1.304 0.192201
## Thriller                0.188807   0.091523   2.063 0.039225 *
## War                    -0.693088   0.181379  -3.821 0.000136 ***
## Western                -0.447982   0.259649  -1.725 0.084595 .
## Action                  0.111610   0.094969   1.175 0.240022
## Short                   2.517698   1.606092   1.568 0.117107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.561 on 2415 degrees of freedom
## Multiple R-squared:  0.4541, Adjusted R-squared:  0.4453
## F-statistic: 51.51 on 39 and 2415 DF,  p-value: < 2.2e-16
```

Appendix B2 Lasso

```
best_lambda_lasso_before
## [1] 0.0007831699
predict(cv.lasso_before, s = best_lambda_lasso_before, type = "coefficients")
## 41 x 1 sparse Matrix of class "dgCMatrix"
##                                    1
## (Intercept)           -2.573128619
## (Intercept)            .
## duration               0.393440798
## facenumber_in_poster  -0.030969991
## budget                 0.699131079
## director_name_rating   0.177906951
## average_actor_rating  -0.046851076
## G                      2.187692540
## NC.17                  1.915980261
## Passed                 2.872628210
## PG                     1.938725232
## PG.13                  1.720096973
## R                      1.410342042
## Color                 -0.204951269
## Language               1.002070193
## Asia                  -1.264367851
## Europe                -0.830734552
## Oceania               -0.626493565
## S_America             -0.474405522
## Adventure              0.054290507
## Animation              0.007868113
## Biography             -0.074135811
## Comedy                 0.228059998
## Crime                 -0.220069524
## Documentary            0.763277810
## Drama                 -0.332429598
## Family                 0.177915613
## Fantasy               -0.115436881
## History               -0.042718723
## Horror                 0.379425589
## Music                  0.201411401
## Musical               -0.705082234
## Mystery                0.081990799
## Romance                0.073053889
## SciFi                 -0.283018446
## Sport                 -0.205730787
## Thriller               0.186182100
## War                   -0.689659515
## Western               -0.438619994
## Action                 0.107855343
## Short                  2.401739449
finallasso_before = glmnet(X,Y,alpha = 1, lambda = best_lambda_lasso_before)
mse_lasso_before_inter_min
<- cv.lasso_before$cvm[which(cv.lasso_before$lambda ==cv.lasso_before$lambda.min)]
sqrt(mse_lasso_before_inter_min)
## [1] 1.582405
```

Appendix B3 Best Subset Selection

```
> ave_mean
 [1] 1.704354 1.650217 1.630705 1.620439 1.612917 1.605965 1.599763 1.594492 1.590491 1.586020 1.579860 1.574713 1.569988 1.566670 1.563931 1.561646 1.559842 1.558205 1.556750 1.555390
[21] 1.554039 1.552792 1.551808 1.550972 1.550243 1.549599 1.549004 1.548528 1.548143 1.547832 1.547580 1.547376 1.547206 1.547085 1.546997 1.546941 1.546920
> best_size=which.min(ave_mean)
> best.subset_ = regsubsets(gross ~ ., data = dt,nvmax = 38,force.in = FALSE)
> best.subset_$nbest
[1] 1
> summ<-summary(best.subset_)
> summ$adjr2
 [1] 0.3531979 0.3942067 0.4088591 0.4143500 0.4193471 0.4245878 0.4288405 0.4326707 0.4348606 0.4378734 0.4409664 0.4445818 0.4471821 0.4493926 0.4514342 0.4530859 0.4543389 0.4552504
[19] 0.4561707 0.4571316 0.4577185 0.4582606 0.4586442 0.4590182 0.4593677 0.4596447 0.4599389 0.4599994 0.4600159 0.4599910 0.4598990 0.4598027 0.4597015 0.4595911 0.4594318 0.4592395
[37] 0.4590058 0.4587566
> which.min(summ$adjr2)
[1] 1
> which.max(summ$adjr2)
[1] 29
>
```

Appendix C

Random Forest Output: Importance Ranking of Variables

budget	duration	Family	R
3228.21683147	593.61458021	255.86602740	213.91834032
Adventure	director_name_rating	Drama	Action
195.63210701	185.31895962	178.39421897	172.94430647
PG.13	Europe	Thriller	average_actor_rating
142.32928389	132.18464050	111.92327237	110.11167619
Comedy	Romance	War	facenumber_in_poster
58.51868481	53.72629200	41.92222937	41.65144729
Passed	Asia	History	Animation
23.40163298	17.25243926	16.75497333	14.72249824
Color	Crime	Biography	Music
11.87091760	11.12212045	8.05834338	5.46232117
Western	Short	S_America	Mystery
0.02887024	0.00000000	-0.03555298	-0.07413624
G	Musical	Sport	Oceania
-4.19946052	-7.58373907	-8.11379924	-14.72847183