Stat 5310 – “Show Me the Money”

December 11, 2016

# 1 Introduction

The 1996 Blockbuster hit “Jerry Maguire” said it best, “Show Me the Money”. The great empire that is the movie industry didn’t get that way by making flops. Movie studios invest 100’s of millions of dollars into movies. Some won’t make any money, but they do keep on making them in hopes of finding the ones that payoff big. Let’s see if perhaps a little “data magic” could shed more light on how to create the “movie magic” we all enjoy.

# 2 Data

The data was acquired from the Kaggle website which was scraped from the IMDB website. We collected data for over 5,000 movies from all over the world. As the title of the paper suggests, the focus of this research is predicting how much money a movie will make. Apart from the box office gross, the dataset includes several other variables. A list of those variable is below. Several of the variables are not very useful in their current state for regression. For example, “director\_name” is a factor comprised of 2,399 different levels. As good one particular director maybe, it would be hard pressed for one level, one director, to add enough to a model to be significant. Datamining techniques could be used to make engineered features but that is not the focus of the study. Consequently, variables of such nature were omitted from any regression models built. The strikethrough variables were the ignored ones and the highlighted one were the only variable used in our best model.

|  |  |
| --- | --- |
| **Variables in IMDB Movie Dataset** | |
| ***duration*** | \*\*\* gross \*\*\* |
| ***movie\_facebook\_likes*** | budget |
| ***actor\_3\_facebook\_likes*** | director\_facebook\_likes |
| ***actor\_1\_facebook\_likes*** | ~~movie\_imdb\_link~~ |
| ***num\_voted\_users*** | ~~actor\_1\_name~~ |
| ***title\_year*** | ~~language~~ |
| ***actor\_2\_facebook\_likes*** | ~~country~~ |
| ***imdb\_score*** | ~~movie\_title~~ |
| ***num\_critic\_for\_reviews*** | ~~genres~~ |
| ***color*** | ~~actor\_3\_name~~ |
| ***cast\_total\_facebook\_likes*** | ~~actor\_2\_name~~ |
| ***content\_rating*** | ~~director\_name~~ |
| facenumber\_in\_poster | ~~aspect\_ratio~~ |
| num\_user\_for\_reviews | ~~plot\_keywords~~ |
| \*\*\* Dependent Variable \*\*\* | |

# 3 Model Building

## Model1

Using R, we cleaned up the data collected from the Kaggle website by removing the unwanted variables and ensuring that only complete cases were in the dataset. Next, we created two subsets of the data. First, we created a training set which consist of 3,773 out of 3,873 observations that were left after cleaning the data. Second, we created a test set of 100 randomly selected movies after the cleaning. In our first model, we used all possible variables that were appropriate for regression analysis. A list of those can be found on page one. This model consisted all non-strikethrough variables. Also, the dependent variable “gross” was transformed with a log () function due to negative values being predicted. The linear equation from the first model is below. The R2 = 0.4371 and the adjusted R2 = 0.4335, which are not too pleasing but we’ll continue with testing the model.

**Y** = 113.4 + 0.007711\*num\_critic\_for\_reviews + 0.007485\*duration + -0.00001925\*director\_facebook\_likes + -0.0002035\*actor\_3\_facebook\_likes + -0.0001367\*actor\_1\_facebook\_likes + 0.000003157\*num\_voted\_users + 0.0001402\*cast\_total\_facebook\_likes + 0.00001997\*num\_user\_for\_reviews + -0.000000000152\*budget + -0.05125\*title\_year + -0.0001212\*actor\_2\_facebook\_likes + -0.2382\*imdb\_score + -0.00001291\*movie\_facebook\_likes + 0.784\*color Black and White + 1.469\*colorColor + 0.003085\*facenumber\_in\_poster + 1.714\*content\_ratingApproved + 4.515\*content\_ratingG + 3.222\*content\_ratingGP + 3.129\*content\_ratingM + 1.06\*content\_ratingNC-17 + -0.4985\*content\_ratingNot Rated + 0.7994\*content\_ratingPassed + 4.469\*content\_ratingPG + 3.985\*content\_ratingPG-13 + 2.993\*content\_ratingR + 0.6495\*content\_ratingUnrated + 2.418\*content\_ratingX

We will test the overall fit of the model.

1. H0 : B1 = B2 = … = B28 = 0 vs. H1 : Not all Bi = 0
2. p = 2.2e-16
3. Since p < a, we rejected the Ho
4. The model fits well and provides information

Let’s take a look at the diagnostic plots of the model



Figure - Diagnostic Plots for model 1

The statistical software R’s lm () provides four plots to evaluate regression models. The first plot, “Residuals vs Fitted”, is used to determining if there is some non-linear relation between the dependent variable and the predictor variables. According to plot, although it’s not perfect, it is reasonable to assume there is not a strong non-linear relation between variables. The distribution of the residuals is not quite normal. This was determined from the “Normal Q-Q” plot. Although most values lye on the dotted line, they fall of the line on the bottom left of the plot. The model shows homoscedasticity on the “Scale-Location” plot suggesting the residuals are spread equally along the ranges of predictors. Lastly, the “Residuals vs leverage” plot reveals there are no influential observations skewing the results. This model was “ok” but we wanted to see if needed all those variable or could we do without some. As a result, we ran the stepwise () function on model 1. We will discuss the model suggested by stepwise in the next section.

## Model 2

The stepwise () function revealed a “better” model could be constructed from the variables used in model1. A list of the variables used could be found on page 1. The ***bold and italic*** and highlighted variables are the ones in model2. The linear equation for model2 is below.

**Y =** 4.418908\*content\_ratingG + 4.376438\*content\_ratingPG + 3.877552\*content\_ratingPG-13 + 3.14243\*content\_ratingGP + 3.051225\*content\_ratingM + 2.883832\*content\_ratingR + 2.332755\*content\_ratingX + 1.53309\*content\_ratingApproved + 1.462782\*colorColor + 0.9714281\*content\_ratingNC-17 + 0.7480404\*content\_ratingPassed + 0.7357954\*color Black and White + 0.5587502\*content\_ratingUnrated + 0.00766669\*num\_critic\_for\_reviews + 0.006950608\*duration + 0.0001622229\*cast\_total\_facebook\_likes + 0.000003154153\*num\_voted\_users + -0.0000127232\*movie\_facebook\_likes + -0.0001456395\*actor\_2\_facebook\_likes + -0.0001587485\*actor\_1\_facebook\_likes + -0.0002290609\*actor\_3\_facebook\_likes + -0.05105337\*title\_year + -0.2320718\*imdb\_score + -0.5879809\*content\_ratingNot Rated +

Comparatively, model2 did not perform considerably better than model1. However, model2 is able to gain relatively the same amount of with less variables and less computing power. The R2 = 0.4346 and the adjusted R2 = 0.4310. We will choose to use model2 in predicting the gross for movies.

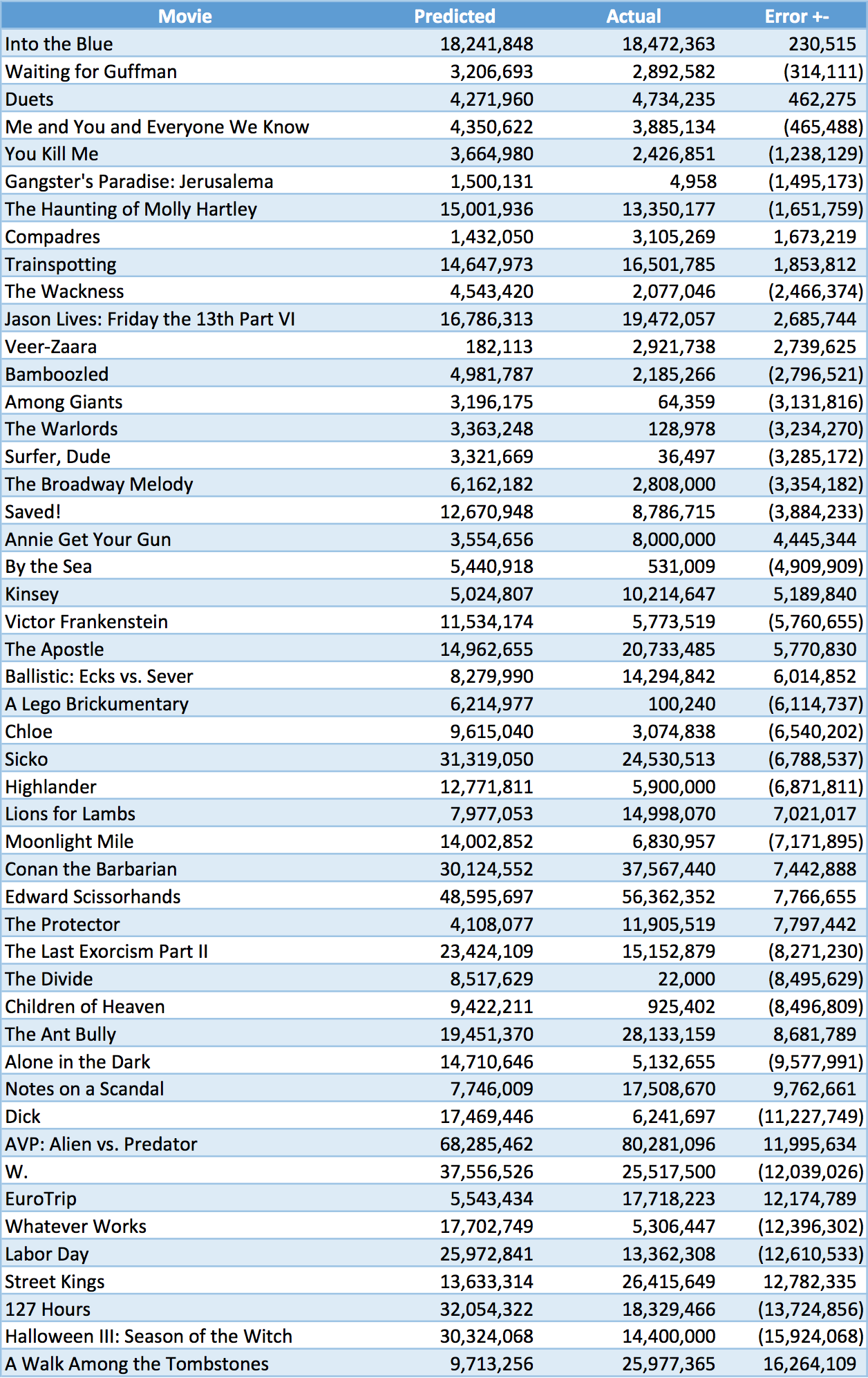
We will test the overall fit of the model.

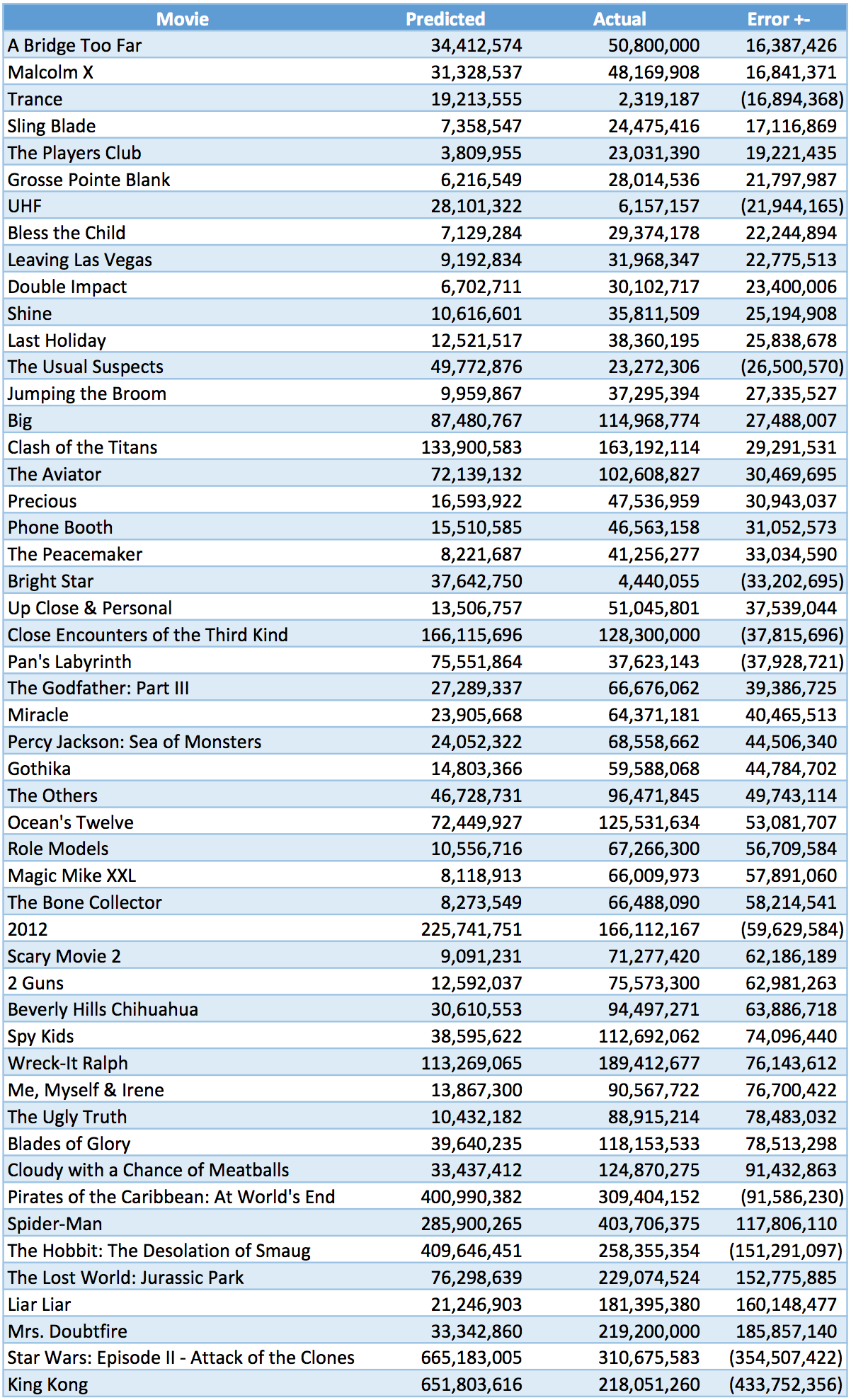
1. H0 : B1 = B2 = … = B24 = 0 vs. H1 : Not all Bi = 0
2. p = 2.2e-16
3. Since p < a, we rejected the Ho
4. The model fits well and provides information

Let’s take a look at the diagnostic plots of the model

# Predictions for Model2

Lastly, on the next page is a table of the predicted value and the actual value of the 100 test movies we separated from the dataset earlier.





In Conclusion, we chose model2. It provided more information on predicting movie grosses. Additionally, it uses less variables and computing power then model1.

# References

R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. [www.r-project.org](http://www.r-project.org).

John Fox (2016). RcmdrMisc: R Commander Miscellaeous Functions. R

Package Version 1.0-5. https://CRAN.R-project.org/package=RcmdrMisc

Kaggle - www.kaggle.com

https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset/version/1

# Appendix

# bring in movie data and libraries

library(RcmdrMisc)

movie <- read.csv("~/movie\_metadata.csv", header=TRUE)

movie2 <- movie[,c(3,4,5,6,8,13,14,19,23,24,25,26,28,1,16,22,9)]

movie2 <- movie2[complete.cases(movie2),]

test.num <- sample(1:3873, 100, replace=F)

movie.test <- movie2[test.num,-17]

test.gross <- movie2[test.num,17]

movie.train <- movie2[-test.num,]

movie.train$gross <- log(movie.train$gross)

# See structure of data

str(movie)

'data.frame': 5043 obs. of 28 variables:

$ color : Factor w/ 3 levels ""," Black and White",..: 3 3 3 3 1 3 3 3 3 3 ...

$ director\_name : Factor w/ 2399 levels "","A. Raven Cruz",..: 927 801 2027 377 603 106 2030 1652 1228 551 ...

$ num\_critic\_for\_reviews : int 723 302 602 813 NA 462 392 324 635 375 ...

$ duration : int 178 169 148 164 NA 132 156 100 141 153 ...

$ director\_facebook\_likes : int 0 563 0 22000 131 475 0 15 0 282 ...

$ actor\_3\_facebook\_likes : int 855 1000 161 23000 NA 530 4000 284 19000 10000 ...

$ actor\_2\_name : Factor w/ 3033 levels "","50 Cent","A. Michael Baldwin",..: 1407 2218 2488 534 2432 2549 1227 801 2439 653 ...

$ actor\_1\_facebook\_likes : int 1000 40000 11000 27000 131 640 24000 799 26000 25000 ...

$ gross : int 760505847 309404152 200074175 448130642 NA 73058679 336530303 200807262 458991599 301956980 ...

$ genres : Factor w/ 914 levels "Action","Action|Adventure",..: 107 101 128 288 754 126 120 308 126 447 ...

$ actor\_1\_name : Factor w/ 2098 levels "","50 Cent","A.J. Buckley",..: 302 979 353 1968 526 440 785 221 336 32 ...

$ movie\_title : Factor w/ 4917 levels "[Rec] ","[Rec] 2 ",..: 398 2731 3279 3708 3332 1961 3291 3459 399 1631 ...

$ num\_voted\_users : int 886204 471220 275868 1144337 8 212204 383056 294810 462669 321795 ...

$ cast\_total\_facebook\_likes: int 4834 48350 11700 106759 143 1873 46055 2036 92000 58753 ...

$ actor\_3\_name : Factor w/ 3522 levels "","50 Cent","A.J. Buckley",..: 3442 1392 3134 1769 1 2714 1969 2162 3018 2941 ...

$ facenumber\_in\_poster : int 0 0 1 0 0 1 0 1 4 3 ...

$ plot\_keywords : Factor w/ 4761 levels "","10 year old|dog|florida|girl|supermarket",..: 1320 4283 2076 3484 1 651 4745 29 1142 2005 ...

$ movie\_imdb\_link : Factor w/ 4919 levels "http://www.imdb.com/title/tt0006864/?ref\_=fn\_tt\_tt\_1",..: 2965 2721 4533 3756 4918 2476 2526 2458 4546 2551 ...

$ num\_user\_for\_reviews : int 3054 1238 994 2701 NA 738 1902 387 1117 973 ...

$ language : Factor w/ 48 levels "","Aboriginal",..: 13 13 13 13 1 13 13 13 13 13 ...

$ country : Factor w/ 66 levels "","Afghanistan",..: 65 65 63 65 1 65 65 65 65 63 ...

$ content\_rating : Factor w/ 19 levels "","Approved",..: 10 10 10 10 1 10 10 9 10 9 ...

$ budget : num 2.37e+08 3.00e+08 2.45e+08 2.50e+08 NA ...

$ title\_year : int 2009 2007 2015 2012 NA 2012 2007 2010 2015 2009 ...

$ actor\_2\_facebook\_likes : int 936 5000 393 23000 12 632 11000 553 21000 11000 ...

$ imdb\_score : num 7.9 7.1 6.8 8.5 7.1 6.6 6.2 7.8 7.5 7.5 ...

$ aspect\_ratio : num 1.78 2.35 2.35 2.35 NA 2.35 2.35 1.85 2.35 2.35 ...

$ movie\_facebook\_likes : int 33000 0 85000 164000 0 24000 0 29000 118000 10000 ...

# Run baseline model - Simple regression using all variables

movie.lm <- lm(gross~.,data = movie.train)

summary(movie.lm)

Call:

lm(formula = gross ~ ., data = movie.train)

Residuals:

Min 1Q Median 3Q Max

-9.4549 -0.6753 0.3212 1.0863 4.6606

Coefficients:

Estimate Std. Error t value Pr(|t|)

(Intercept) 1.119e+02 7.646e+00 14.640 < 2e-16 \*\*\*

num\_critic\_for\_reviews 7.658e-03 3.962e-04 19.327 < 2e-16 \*\*\*

duration 7.454e-03 1.385e-03 5.383 7.80e-08 \*\*\*

director\_facebook\_likes -1.880e-05 9.544e-06 -1.969 0.048985 \*

actor\_3\_facebook\_likes -2.092e-04 4.188e-05 -4.996 6.13e-07 \*\*\*

actor\_1\_facebook\_likes -1.400e-04 2.518e-05 -5.560 2.89e-08 \*\*\*

num\_voted\_users 3.075e-06 3.483e-07 8.828 < 2e-16 \*\*\*

cast\_total\_facebook\_likes 1.434e-04 2.510e-05 5.714 1.19e-08 \*\*\*

num\_user\_for\_reviews 4.793e-05 1.182e-04 0.406 0.685007

budget -1.514e-10 1.214e-10 -1.247 0.212295

title\_year -5.054e-02 3.679e-03 -13.738 < 2e-16 \*\*\*

actor\_2\_facebook\_likes -1.245e-04 2.662e-05 -4.677 3.01e-06 \*\*\*

imdb\_score -2.336e-01 3.251e-02 -7.184 8.11e-13 \*\*\*

movie\_facebook\_likes -1.273e-05 1.891e-06 -6.734 1.91e-11 \*\*\*

color Black and White 7.473e-01 1.675e+00 0.446 0.655601

colorColor 1.469e+00 1.668e+00 0.881 0.378436

facenumber\_in\_poster 3.714e-03 1.359e-02 0.273 0.784663

content\_ratingApproved 1.757e+00 5.046e-01 3.482 0.000503 \*\*\*

content\_ratingG 4.588e+00 3.101e-01 14.795 < 2e-16 \*\*\*

content\_ratingGP 3.251e+00 1.690e+00 1.923 0.054535 .

content\_ratingM 3.157e+00 1.213e+00 2.604 0.009261 \*\*

content\_ratingNC-17 1.080e+00 7.271e-01 1.485 0.137673

content\_ratingNot Rated -4.823e-01 3.645e-01 -1.323 0.185890

content\_ratingPassed 8.494e-01 1.708e+00 0.497 0.618977

content\_ratingPG 4.486e+00 2.621e-01 17.115 < 2e-16 \*\*\*

content\_ratingPG-13 3.989e+00 2.580e-01 15.460 < 2e-16 \*\*\*

content\_ratingR 2.995e+00 2.558e-01 11.709 < 2e-16 \*\*\*

content\_ratingUnrated 6.633e-01 4.240e-01 1.565 0.117780

content\_ratingX 2.445e+00 5.923e-01 4.128 3.74e-05 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.667 on 3744 degrees of freedom

Multiple R-squared: 0.4379, Adjusted R-squared: 0.4337

F-statistic: 104.2 on 28 and 3744 DF, p-value: < 2.2e-16

par(mfrow=c(2,2))

plot(movie.lm)

Warning messages:

1: not plotting observations with leverage one:

2964, 3093, 3369

2: not plotting observations with leverage one:

2964, 3093, 3369

# Predict test

pred1 <- predict(movie.lm,movie.test)

pred1 <- exp(pred1)

final1 <- cbind(round(pred1,digits = 0),test.gross)

colnames(final1) <- c("Predicted","Actual")

rownames(final1) <- seq(100)

final1 <- as.data.frame(final1)

final1$Error\_sqr <- sqrt((final1$Predicted - final1$Actual)^2)

final1$Error <- final1$Predicted - final1$Actual

final1.1 <- as.data.frame(final1.1)

Error in as.data.frame(final1.1) : object 'final1.1' not found

final1.1

Error: object 'final1.1' not found

# Run stepwise and build best model

stepwise(movie.lm)

Direction: backward/forward

Criterion: BIC

Start: AIC=4063.86

gross ~ num\_critic\_for\_reviews + duration + director\_facebook\_likes +

actor\_3\_facebook\_likes + actor\_1\_facebook\_likes + num\_voted\_users +

cast\_total\_facebook\_likes + num\_user\_for\_reviews + budget +

title\_year + actor\_2\_facebook\_likes + imdb\_score + movie\_facebook\_likes +

color + facenumber\_in\_poster + content\_rating

Df Sum of Sq RSS AIC

- facenumber\_in\_poster 1 0.21 10399 4055.7

- num\_user\_for\_reviews 1 0.46 10399 4055.8

- budget 1 4.32 10403 4057.2

- director\_facebook\_likes 1 10.77 10409 4059.5

<none 10398 4063.9

- color 2 62.13 10461 4069.9

- actor\_2\_facebook\_likes 1 60.76 10459 4077.6

- actor\_3\_facebook\_likes 1 69.31 10468 4080.7

- duration 1 80.46 10479 4084.7

- actor\_1\_facebook\_likes 1 85.85 10484 4086.6

- cast\_total\_facebook\_likes 1 90.70 10489 4088.4

- movie\_facebook\_likes 1 125.93 10524 4101.0

- imdb\_score 1 143.35 10542 4107.3

- num\_voted\_users 1 216.47 10615 4133.4

- title\_year 1 524.15 10923 4241.2

- num\_critic\_for\_reviews 1 1037.42 11436 4414.4

- content\_rating 12 2664.32 13063 4825.7

Step: AIC=4055.7

gross ~ num\_critic\_for\_reviews + duration + director\_facebook\_likes +

actor\_3\_facebook\_likes + actor\_1\_facebook\_likes + num\_voted\_users +

cast\_total\_facebook\_likes + num\_user\_for\_reviews + budget +

title\_year + actor\_2\_facebook\_likes + imdb\_score + movie\_facebook\_likes +

color + content\_rating

Df Sum of Sq RSS AIC

- num\_user\_for\_reviews 1 0.41 10399 4047.6

- budget 1 4.37 10403 4049.0

- director\_facebook\_likes 1 10.92 10410 4051.4

<none 10399 4055.7

- color 2 62.04 10461 4061.7

+ facenumber\_in\_poster 1 0.21 10398 4063.9

- actor\_2\_facebook\_likes 1 60.80 10460 4069.5

- actor\_3\_facebook\_likes 1 69.17 10468 4072.5

- duration 1 81.51 10480 4076.9

- actor\_1\_facebook\_likes 1 85.91 10485 4078.5

- cast\_total\_facebook\_likes 1 90.79 10490 4080.3

- movie\_facebook\_likes 1 125.81 10524 4092.8

- imdb\_score 1 144.27 10543 4099.5

- num\_voted\_users 1 217.87 10617 4125.7

- title\_year 1 525.68 10924 4233.5

- num\_critic\_for\_reviews 1 1038.64 11437 4406.7

- content\_rating 12 2664.70 13063 4817.6

Step: AIC=4047.61

gross ~ num\_critic\_for\_reviews + duration + director\_facebook\_likes +

actor\_3\_facebook\_likes + actor\_1\_facebook\_likes + num\_voted\_users +

cast\_total\_facebook\_likes + budget + title\_year + actor\_2\_facebook\_likes +

imdb\_score + movie\_facebook\_likes + color + content\_rating

Df Sum of Sq RSS AIC

- budget 1 4.37 10404 4041.0

- director\_facebook\_likes 1 11.09 10410 4043.4

<none 10399 4047.6

- color 2 61.64 10461 4053.4

+ num\_user\_for\_reviews 1 0.41 10399 4055.7

+ facenumber\_in\_poster 1 0.16 10399 4055.8

- actor\_2\_facebook\_likes 1 60.89 10460 4061.4

- actor\_3\_facebook\_likes 1 69.23 10468 4064.4

- duration 1 85.46 10485 4070.3

- actor\_1\_facebook\_likes 1 85.98 10485 4070.4

- cast\_total\_facebook\_likes 1 90.83 10490 4072.2

- movie\_facebook\_likes 1 134.90 10534 4088.0

- imdb\_score 1 151.03 10550 4093.8

- num\_voted\_users 1 398.52 10798 4181.3

- title\_year 1 536.50 10936 4229.2

- num\_critic\_for\_reviews 1 1171.01 11570 4442.0

- content\_rating 12 2664.29 13063 4809.4

Step: AIC=4040.96

gross ~ num\_critic\_for\_reviews + duration + director\_facebook\_likes +

actor\_3\_facebook\_likes + actor\_1\_facebook\_likes + num\_voted\_users +

cast\_total\_facebook\_likes + title\_year + actor\_2\_facebook\_likes +

imdb\_score + movie\_facebook\_likes + color + content\_rating

Df Sum of Sq RSS AIC

- director\_facebook\_likes 1 11.04 10414 4036.7

<none 10404 4041.0

- color 2 61.23 10465 4046.6

+ budget 1 4.37 10399 4047.6

+ num\_user\_for\_reviews 1 0.41 10403 4049.0

+ facenumber\_in\_poster 1 0.20 10403 4049.1

- actor\_2\_facebook\_likes 1 60.87 10464 4054.7

- actor\_3\_facebook\_likes 1 69.31 10473 4057.8

- duration 1 83.73 10487 4063.0

- actor\_1\_facebook\_likes 1 85.89 10489 4063.7

- cast\_total\_facebook\_likes 1 90.76 10494 4065.5

- movie\_facebook\_likes 1 133.26 10537 4080.7

- imdb\_score 1 150.55 10554 4086.9

- num\_voted\_users 1 398.55 10802 4174.6

- title\_year 1 536.93 10940 4222.6

- num\_critic\_for\_reviews 1 1166.99 11570 4433.9

- content\_rating 12 2660.00 13064 4801.2

Step: AIC=4036.73

gross ~ num\_critic\_for\_reviews + duration + actor\_3\_facebook\_likes +

actor\_1\_facebook\_likes + num\_voted\_users + cast\_total\_facebook\_likes +

title\_year + actor\_2\_facebook\_likes + imdb\_score + movie\_facebook\_likes +

color + content\_rating

Df Sum of Sq RSS AIC

<none 10414 4036.7

+ director\_facebook\_likes 1 11.04 10404 4041.0

- color 2 63.12 10478 4043.1

+ budget 1 4.32 10410 4043.4

+ num\_user\_for\_reviews 1 0.58 10414 4044.8

+ facenumber\_in\_poster 1 0.33 10414 4044.8

- actor\_2\_facebook\_likes 1 61.96 10476 4050.9

- actor\_3\_facebook\_likes 1 70.72 10485 4054.0

- duration 1 80.35 10495 4057.5

- actor\_1\_facebook\_likes 1 86.98 10502 4059.9

- cast\_total\_facebook\_likes 1 91.78 10506 4061.6

- movie\_facebook\_likes 1 134.28 10549 4076.8

- imdb\_score 1 152.83 10567 4083.5

- num\_voted\_users 1 387.52 10802 4166.3

- title\_year 1 530.38 10945 4215.9

- num\_critic\_for\_reviews 1 1165.82 11580 4428.8

- content\_rating 12 2656.01 13070 4795.0

Call:

lm(formula = gross ~ num\_critic\_for\_reviews + duration + actor\_3\_facebook\_likes +

actor\_1\_facebook\_likes + num\_voted\_users + cast\_total\_facebook\_likes +

title\_year + actor\_2\_facebook\_likes + imdb\_score + movie\_facebook\_likes +

color + content\_rating, data = movie.train)

Coefficients:

(Intercept) num\_critic\_for\_reviews duration actor\_3\_facebook\_likes

1.114e+02 7.660e-03 7.312e-03 -2.112e-04

actor\_1\_facebook\_likes num\_voted\_users cast\_total\_facebook\_likes title\_year

-1.409e-04 3.076e-06 1.443e-04 -5.027e-02

actor\_2\_facebook\_likes imdb\_score movie\_facebook\_likes color Black and White

-1.257e-04 -2.374e-01 -1.284e-05 7.511e-01

colorColor content\_ratingApproved content\_ratingG content\_ratingGP

1.477e+00 1.786e+00 4.586e+00 3.276e+00

content\_ratingM content\_ratingNC-17 content\_ratingNot Rated content\_ratingPassed

3.190e+00 1.085e+00 -4.790e-01 9.054e-01

content\_ratingPG content\_ratingPG-13 content\_ratingR content\_ratingUnrated

4.476e+00 3.988e+00 2.993e+00 6.713e-01

content\_ratingX

2.465e+00

movie2.lm <- lm(formula = gross ~ actor\_1\_facebook\_likes +

+ actor\_2\_facebook\_likes +

+ actor\_3\_facebook\_likes +

+ cast\_total\_facebook\_likes +

+ color +

+ content\_rating +

+ duration +

+ imdb\_score +

+ movie\_facebook\_likes +

+ num\_critic\_for\_reviews +

+ num\_voted\_users +

+ title\_year ,

+ data = movie.train)

summary(movie2.lm)

Call:

lm(formula = gross ~ actor\_1\_facebook\_likes + actor\_2\_facebook\_likes +

actor\_3\_facebook\_likes + cast\_total\_facebook\_likes + color +

content\_rating + duration + imdb\_score + movie\_facebook\_likes +

num\_critic\_for\_reviews + num\_voted\_users + title\_year, data = movie.train)

Residuals:

Min 1Q Median 3Q Max

-9.4453 -0.6683 0.3297 1.0882 4.6462

Coefficients:

Estimate Std. Error t value Pr(|t|)

(Intercept) 1.114e+02 7.567e+00 14.725 < 2e-16 \*\*\*

actor\_1\_facebook\_likes -1.409e-04 2.518e-05 -5.595 2.36e-08 \*\*\*

actor\_2\_facebook\_likes -1.257e-04 2.662e-05 -4.722 2.42e-06 \*\*\*

actor\_3\_facebook\_likes -2.112e-04 4.187e-05 -5.045 4.76e-07 \*\*\*

cast\_total\_facebook\_likes 1.443e-04 2.510e-05 5.747 9.80e-09 \*\*\*

color Black and White 7.511e-01 1.675e+00 0.448 0.653981

colorColor 1.477e+00 1.668e+00 0.885 0.376123

content\_ratingApproved 1.786e+00 5.038e-01 3.546 0.000397 \*\*\*

content\_ratingG 4.586e+00 3.096e-01 14.812 < 2e-16 \*\*\*

content\_ratingGP 3.276e+00 1.691e+00 1.938 0.052758 .

content\_ratingM 3.190e+00 1.213e+00 2.630 0.008566 \*\*

content\_ratingNC-17 1.085e+00 7.271e-01 1.492 0.135820

content\_ratingNot Rated -4.790e-01 3.644e-01 -1.314 0.188846

content\_ratingPassed 9.054e-01 1.707e+00 0.531 0.595788

content\_ratingPG 4.476e+00 2.620e-01 17.082 < 2e-16 \*\*\*

content\_ratingPG-13 3.988e+00 2.581e-01 15.451 < 2e-16 \*\*\*

content\_ratingR 2.993e+00 2.558e-01 11.701 < 2e-16 \*\*\*

content\_ratingUnrated 6.713e-01 4.240e-01 1.583 0.113464

content\_ratingX 2.465e+00 5.923e-01 4.161 3.24e-05 \*\*\*

duration 7.312e-03 1.360e-03 5.377 8.02e-08 \*\*\*

imdb\_score -2.374e-01 3.201e-02 -7.416 1.48e-13 \*\*\*

movie\_facebook\_likes -1.284e-05 1.847e-06 -6.952 4.24e-12 \*\*\*

num\_critic\_for\_reviews 7.660e-03 3.740e-04 20.483 < 2e-16 \*\*\*

num\_voted\_users 3.076e-06 2.605e-07 11.809 < 2e-16 \*\*\*

title\_year -5.027e-02 3.639e-03 -13.816 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.667 on 3748 degrees of freedom

Multiple R-squared: 0.4371, Adjusted R-squared: 0.4335

F-statistic: 121.3 on 24 and 3748 DF, p-value: < 2.2e-16

plot(movie2.lm)

Warning messages:

1: not plotting observations with leverage one:

2964, 3093

2: not plotting observations with leverage one:

2964, 3093

3: In sqrt(crit \* p \* (1 - hh)/hh) : NaNs produced

4: In sqrt(crit \* p \* (1 - hh)/hh) : NaNs produced

# Predict test 2

pred2 <- predict(movie2.lm,movie.test)

pred2 <- exp(pred2)

final2 <- cbind(round(pred2,digits = 0),round(test.gross,digits = 0))

colnames(final2) <- c("Predicted","Actual")

rownames(final2) <- seq(100)

final2 <- as.data.frame(final2)

final2$Error <- final2$Predicted - final2$Actual

final2.1 <- format(final2, big.mark = ",",big.interval = 3)

final2.1 <- as.data.frame(final2.1)

final2.1

Predicted Actual Error

1 68,285,462 80,281,096 -11,995,634

2 7,129,284 29,374,178 -22,244,894

3 182,113 2,921,738 -2,739,625

4 12,521,517 38,360,195 -25,838,678

5 4,543,420 2,077,046 2,466,374

6 21,246,903 181,395,380 -160,148,477

7 48,595,697 56,362,352 -7,766,655

8 37,556,526 25,517,500 12,039,026

9 1,500,131 4,958 1,495,173

10 37,642,750 4,440,055 33,202,695

11 14,803,366 59,588,068 -44,784,702

12 12,592,037 75,573,300 -62,981,263

13 9,959,867 37,295,394 -27,335,527

14 651,803,616 218,051,260 433,752,356

15 8,221,687 41,256,277 -33,034,590

16 400,990,382 309,404,152 91,586,230

17 10,616,601 35,811,509 -25,194,908

18 23,905,668 64,371,181 -40,465,513

19 39,640,235 118,153,533 -78,513,298

20 76,298,639 229,074,524 -152,775,885

21 31,328,537 48,169,908 -16,841,371

22 49,772,876 23,272,306 26,500,570

23 75,551,864 37,623,143 37,928,721

24 33,342,860 219,200,000 -185,857,140

25 3,363,248 128,978 3,234,270

26 17,469,446 6,241,697 11,227,749

27 72,449,927 125,531,634 -53,081,707

28 12,771,811 5,900,000 6,871,811

29 30,610,553 94,497,271 -63,886,718

30 4,271,960 4,734,235 -462,275

31 10,556,716 67,266,300 -56,709,584

32 7,746,009 17,508,670 -9,762,661

33 9,091,231 71,277,420 -62,186,189

34 9,422,211 925,402 8,496,809

35 4,350,622 3,885,134 465,488

36 11,534,174 5,773,519 5,760,655

37 14,710,646 5,132,655 9,577,991

38 3,196,175 64,359 3,131,816

39 16,786,313 19,472,057 -2,685,744

40 15,001,936 13,350,177 1,651,759

41 27,289,337 66,676,062 -39,386,725

42 14,002,852 6,830,957 7,171,895

43 17,702,749 5,306,447 12,396,302

44 15,510,585 46,563,158 -31,052,573

45 1,432,050 3,105,269 -1,673,219

46 4,981,787 2,185,266 2,796,521

47 225,741,751 166,112,167 59,629,584

48 38,595,622 112,692,062 -74,096,440

49 16,593,922 47,536,959 -30,943,037

50 8,517,629 22,000 8,495,629

51 9,713,256 25,977,365 -16,264,109

52 5,543,434 17,718,223 -12,174,789

53 8,118,913 66,009,973 -57,891,060

54 5,024,807 10,214,647 -5,189,840

55 46,728,731 96,471,845 -49,743,114

56 285,900,265 403,706,375 -117,806,110

57 4,108,077 11,905,519 -7,797,442

58 13,867,300 90,567,722 -76,700,422

59 7,977,053 14,998,070 -7,021,017

60 30,324,068 14,400,000 15,924,068

61 32,054,322 18,329,466 13,724,856

62 6,702,711 30,102,717 -23,400,006

63 3,664,980 2,426,851 1,238,129

64 14,962,655 20,733,485 -5,770,830

65 7,358,547 24,475,416 -17,116,869

66 409,646,451 258,355,354 151,291,097

67 3,206,693 2,892,582 314,111

68 87,480,767 114,968,774 -27,488,007

69 133,900,583 163,192,114 -29,291,531

70 72,139,132 102,608,827 -30,469,695

71 9,615,040 3,074,838 6,540,202

72 19,213,555 2,319,187 16,894,368

73 19,451,370 28,133,159 -8,681,789

74 6,216,549 28,014,536 -21,797,987

75 166,115,696 128,300,000 37,815,696

76 6,162,182 2,808,000 3,354,182

77 23,424,109 15,152,879 8,271,230

78 28,101,322 6,157,157 21,944,165

79 25,972,841 13,362,308 12,610,533

80 31,319,050 24,530,513 6,788,537

81 8,279,990 14,294,842 -6,014,852

82 665,183,005 310,675,583 354,507,422

83 3,321,669 36,497 3,285,172

84 12,670,948 8,786,715 3,884,233

85 8,273,549 66,488,090 -58,214,541

86 13,506,757 51,045,801 -37,539,044

87 24,052,322 68,558,662 -44,506,340

88 18,241,848 18,472,363 -230,515

89 30,124,552 37,567,440 -7,442,888

90 5,440,918 531,009 4,909,909

91 14,647,973 16,501,785 -1,853,812

92 33,437,412 124,870,275 -91,432,863

93 34,412,574 50,800,000 -16,387,426

94 6,214,977 100,240 6,114,737

95 9,192,834 31,968,347 -22,775,513

96 10,432,182 88,915,214 -78,483,032

97 13,633,314 26,415,649 -12,782,335

98 3,809,955 23,031,390 -19,221,435

99 113,269,065 189,412,677 -76,143,612

100 3,554,656 8,000,000 -4,445,344