Problem Set 6

MGSC 310, Fall 2019, Professor Hersh

Elmer Camargo + Nick Trella

Libraries Needed

```
library("tidyverse")
library("plotROC")
library("ROSE")
library("tidyverse")
library("magrittr")
```

Question 1 What Predicts Blockbusters

a - Preparing Data

```
options(scipen = 50)
set.seed(1861)
movies <- read.csv("data/movie_metadata.csv")
movies <- movies %>% filter(budget < 400000000) %>%
  filter(content_rating != "",
         content_rating != "Not Rated",
         !is.na(gross))
movies <- movies %>%
  mutate(genre_main = unlist(map(strsplit(as.character(movies$genres),"\\|"),
1)),
         grossM = gross / 1000000,
         budgetM = budget / 1000000,
         profitM = grossM - budgetM,
         blockbuster = ifelse(grossM > 200,1,0))
movies <- movies %>% mutate(genre_main = fct_lump(genre_main,5),
                            content_rating = fct_lump(content_rating,3),
                            country = fct_lump(country,2),
                            cast_total_facebook_likes000s =
                              cast_total_facebook_likes / 1000,) %>%
  drop_na()
top_director <- movies %>%
  group_by(director_name) %>%
  summarize(num_films = n()) %>%
  top_frac(.1) %>%
  mutate(top_director = 1) %>%
  select(-num_films)
movies <- movies %>%
  left_join(top_director, by = "director_name") %>%
  mutate(top_director = replace_na(top_director,0))
train_idx <- sample(1:nrow(movies),size = floor(0.75*nrow(movies)))
```

```
movies_train <- movies %>% slice(train_idx)
movies_test <- movies %>% slice(-train_idx)
```

b - T Test and Hypothesis Testing

```
summary(movies_train)
##               color                director_name
##                   :   1   Steven Spielberg:  20
##   Black and White:  91   Woody Allen     :  16
##  Color           :2703   Clint Eastwood  :  13
##                          Renny Harlin    :  12
##                          Ridley Scott    :  12
##                          Spike Lee       :  12
##                          (Other)         :2710
##  num_critic_for_reviews    duration     director_facebook_likes
##  Min.   :  1.0          Min.   : 37.0   Min.   :    0.0
##  1st Qu.: 75.0          1st Qu.: 96.0   1st Qu.:   10.5
##  Median :134.0          Median :106.0   Median :   60.0
##  Mean   :164.5          Mean   :110.1   Mean   :  765.2
##  3rd Qu.:219.0          3rd Qu.:119.0   3rd Qu.:  226.0
##  Max.   :775.0          Max.   :330.0   Max.   :22000.0
##
##  actor_3_facebook_likes         actor_2_name  actor_1_facebook_likes
##  Min.   :    0.0        Morgan Freeman :  16   Min.   :     0
##  1st Qu.:  191.0        Brad Pitt      :  11   1st Qu.:   745
##  Median :  440.0        Charlize Theron:  10   Median :  1000
##  Mean   :  766.5        Bruce Willis   :   9   Mean   :  7902
##  3rd Qu.:  690.0        Adam Sandler   :   8   3rd Qu.: 13000
##  Max.   :23000.0        James Franco   :   7   Max.   :640000
##                         (Other)        :2734
##      gross                                genres
##  Min.   :      703   Comedy|Drama         : 106
##  1st Qu.:  7876514   Comedy               : 102
##  Median : 28751715   Comedy|Drama|Romance: 102
##  Mean   : 50701852   Comedy|Romance       :  98
##  3rd Qu.: 64075567   Drama                :  89
##  Max.   :760505847   Drama|Romance        :  88
##                      (Other)              :2210
##             actor_1_name           movie_title   num_voted_users
##  Robert De Niro   :  31   HalloweenÂ     :   3   Min.   :     22
##  Denzel Washington:  26   HomeÂ          :   3   1st Qu.:  18777
##  Johnny Depp      :  26   PanÂ           :   3   Median :  52029
##  Nicolas Cage     :  26   BrothersÂ      :   2   Mean   : 104047
##  Bruce Willis     :  24   Casino RoyaleÂ :   2   3rd Qu.: 124765
##  Robert Downey Jr.:  22   CrashÂ         :   2   Max.   :1689764
##  (Other)          :2640   (Other)        :2780
##  cast_total_facebook_likes         actor_3_name  facenumber_in_poster
##  Min.   :     0            Steve Coogan  :   8   Min.   : 0.000
##  1st Qu.:  1899            Anne Hathaway :   6   1st Qu.: 0.000
##  Median :  4050            Ben Mendelsohn:   6   Median : 1.000
##  Mean   : 11652            Robert Duvall :   6   Mean   : 1.418
```

```
##  3rd Qu.: 16236            Thomas Lennon :   6   3rd Qu.: 2.000
##  Max.   :656730            Bruce McGill  :   5   Max.   :43.000
##                            (Other)       :2758
##
plot_keywords
##
:  12
##  1940s|child hero|fantasy world|orphan|reference to peter pan
:   3
##  alien friendship|alien invasion|australia|flying car|mother daughter rela
tionship:   3
##  halloween|masked killer|michael myers|slasher|trick or treat
:   3
##  18 wheeler|mutant|ninja|sewer|turtle
:   2
##  1988 winter olympics|coach|ski jumper|winter|winter olympics
:   2
##  (Other)
:2770
##                                                movie_imdb_link
##  http://www.imdb.com/title/tt0077651/?ref_=fn_tt_tt_1:   3
##  http://www.imdb.com/title/tt2224026/?ref_=fn_tt_tt_1:   3
##  http://www.imdb.com/title/tt3332064/?ref_=fn_tt_tt_1:   3
##  http://www.imdb.com/title/tt0072271/?ref_=fn_tt_tt_1:   2
##  http://www.imdb.com/title/tt0080749/?ref_=fn_tt_tt_1:   2
##  http://www.imdb.com/title/tt0087277/?ref_=fn_tt_tt_1:   2
##  (Other)                                                :2780
##  num_user_for_reviews     language     country     content_rating
##  Min.   :   1.0       English :2685   UK   : 237   PG   : 422
##  1st Qu.: 108.0       French  :  26   USA  :2222   PG-13: 980
##  Median : 208.0       Spanish :  17   Other: 336   R    :1279
##  Mean   : 329.3       Mandarin:  10                Other: 114
##  3rd Qu.: 390.5       German  :   8
##  Max.   :5060.0       Italian :   5
##                       (Other) :  44
##      budget            title_year   actor_2_facebook_likes
##  Min.   :      218   Min.   :1929   Min.   :     0.0
##  1st Qu.: 10000000   1st Qu.:1999   1st Qu.:   392.5
##  Median : 25000000   Median :2004   Median :   690.0
##  Mean   : 37986863   Mean   :2003   Mean   :  2022.2
##  3rd Qu.: 50000000   3rd Qu.:2010   3rd Qu.:   979.0
##  Max.   :390000000   Max.   :2016   Max.   :137000.0
##
##    imdb_score     aspect_ratio   movie_facebook_likes     genre_main
##  Min.   :1.600   Min.   : 1.18   Min.   :     0       Action   :709
##  1st Qu.:5.800   1st Qu.: 1.85   1st Qu.:     0       Adventure:284
##  Median :6.500   Median : 2.35   Median :   215       Comedy   :744
##  Mean   :6.448   Mean   : 2.11   Mean   :  9122       Crime    :182
##  3rd Qu.:7.200   3rd Qu.: 2.35   3rd Qu.: 11000       Drama    :492
##  Max.   :9.300   Max.   :16.00   Max.   :349000       Other    :384
```

```
##
##      grossM            budgetM            profitM
##  Min.   :  0.0007   Min.   :  0.0002   Min.   :-375.869
##  1st Qu.:  7.8765   1st Qu.: 10.0000   1st Qu.: -10.559
##  Median : 28.7517   Median : 25.0000   Median :   1.177
##  Mean   : 50.7019   Mean   : 37.9869   Mean   :  12.715
##  3rd Qu.: 64.0756   3rd Qu.: 50.0000   3rd Qu.:  24.400
##  Max.   :760.5058   Max.   :390.0000   Max.   : 523.506
##
##   blockbuster      cast_total_facebook_likes000s  top_director
##  Min.   :0.00000   Min.   :  0.000               Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:  1.899               1st Qu.:0.0000
##  Median :0.00000   Median :  4.050               Median :0.0000
##  Mean   :0.03936   Mean   : 11.652               Mean   :0.3628
##  3rd Qu.:0.00000   3rd Qu.: 16.236               3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :656.730               Max.   :1.0000
##
```

```
mean_train <- mean(movies_train$blockbuster)
mean_test <- mean(movies_test$blockbuster)
mean_difference <- abs(mean_train - mean_test)


p_value <-  t.test(x = movies_train$blockbuster, y = movies_test$blockbuster,
mu = mean_difference)

p_value
##
##  Welch Two Sample t-test
##
## data:  movies_train$blockbuster and movies_test$blockbuster
## t = -4.8133, df = 1370, p-value = 0.000001649
## alternative hypothesis: true difference in means is not equal to 0.0207298
4
## 95 percent confidence interval:
##  -0.037626956 -0.003832732
## sample estimates:
##  mean of x  mean of y
## 0.03935599 0.06008584
```

The low p-value means we are confident that the observed value did not happen by chance. We reject the null hypotehesis

c - Model Summary

```
logit_train <- glm(blockbuster ~ budgetM + top_director + cast_total_facebook
_likes000s + content_rating + genre_main,
                   family = binomial,
                   data = movies_train)
```

```
logit_test <- glm(blockbuster ~ budgetM + top_director + cast_total_facebook_
likes000s + content_rating + genre_main,
                  family = binomial,
                  data = movies_test)
summary(logit_train)
##
## Call:
## glm(formula = blockbuster ~ budgetM + top_director + cast_total_facebook_l
ikes000s +
##     content_rating + genre_main, family = binomial, data = movies_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3617  -0.1909  -0.1111  -0.0534   3.5660
##
## Coefficients:
##                                Estimate Std. Error z value
## (Intercept)                   -4.784644   0.415845 -11.506
## budgetM                        0.023097   0.002158  10.702
## top_director                   0.607554   0.248837   2.442
## cast_total_facebook_likes000s  0.006694   0.002772   2.415
## content_ratingPG-13           -0.184195   0.309130  -0.596
## content_ratingR               -1.918355   0.526983  -3.640
## content_ratingOther            0.402269   0.504138   0.798
## genre_mainAdventure            0.419475   0.331818   1.264
## genre_mainComedy              -0.458585   0.452521  -1.013
## genre_mainCrime              -14.592267 734.472604  -0.020
## genre_mainDrama               -0.482782   0.519183  -0.930
## genre_mainOther               -0.087916   0.527822  -0.167
##                                          Pr(>|z|)
## (Intercept)                   < 0.0000000000000002 ***
## budgetM                       < 0.0000000000000002 ***
## top_director                                0.014623 *
## cast_total_facebook_likes000s               0.015734 *
## content_ratingPG-13                         0.551275
## content_ratingR                             0.000272 ***
## content_ratingOther                         0.424909
## genre_mainAdventure                         0.206168
## genre_mainComedy                            0.310869
## genre_mainCrime                             0.984149
## genre_mainDrama                             0.352429
## genre_mainOther                             0.867713
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 927.34  on 2794  degrees of freedom
## Residual deviance: 555.53  on 2783  degrees of freedom
## AIC: 579.53
```

```
##
## Number of Fisher Scoring iterations: 18
```

d - Interpretting Coeficcients

```
train_exp_coef <- exp(logit_train$coefficients)
train_exp_coef
##                   (Intercept)                       budgetM
##                  8.357101e-03                  1.023366e+00
##                  top_director cast_total_facebook_likes000s
##                  1.835935e+00                  1.006717e+00
##           content_ratingPG-13               content_ratingR
##                  8.317735e-01                  1.468483e-01
##           content_ratingOther           genre_mainAdventure
##                  1.495213e+00                  1.521163e+00
##              genre_mainComedy               genre_mainCrime
##                  6.321779e-01                  4.598951e-07
##               genre_mainDrama               genre_mainOther
##                  6.170641e-01                  9.158378e-01



exp(-1.918355)
## [1] 0.1468483
exp(0.419475)
## [1] 1.521163
exp(0.607554)
## [1] 1.835935
```

rated r [the exp of r says that movies rated R are 85% less likely to be a 'blockbuster'] genremain (the coefficient says adventure movies are 52% more likely to be a 'blockbuster') top_director (if a movie has a 'top director' it is roughly 83% more likely to be a)

e - Predictions

```
preds_LOOCV <- NULL

#this for loop takes a minute, do not stop
for(i in 1:nrow(movies_train))
  {
  mod = glm(blockbuster ~ budgetM + top_director + cast_total_facebook_likes0
00s + content_rating + genre_main,
            data = movies_train %>% slice(-i),family = binomial)
  preds_LOOCV[i] <- predict(mod, newdata =
                              slice(movies_train,i))
  if(i%% 300 ==0){print(i)}
}
## [1] 300
## [1] 600
## [1] 900
## [1] 1200
## [1] 1500
```

```
## [1] 1800
## [1] 2100
## [1] 2400
## [1] 2700

head(preds_LOOCV)
## [1]  -7.159121 -19.033854  -4.667644  -4.425174  -2.817222  -2.874187

preds_LOOCV_DF <- data.frame(
  scores_LOOCV_train = preds_LOOCV,
  movies_train
)
```

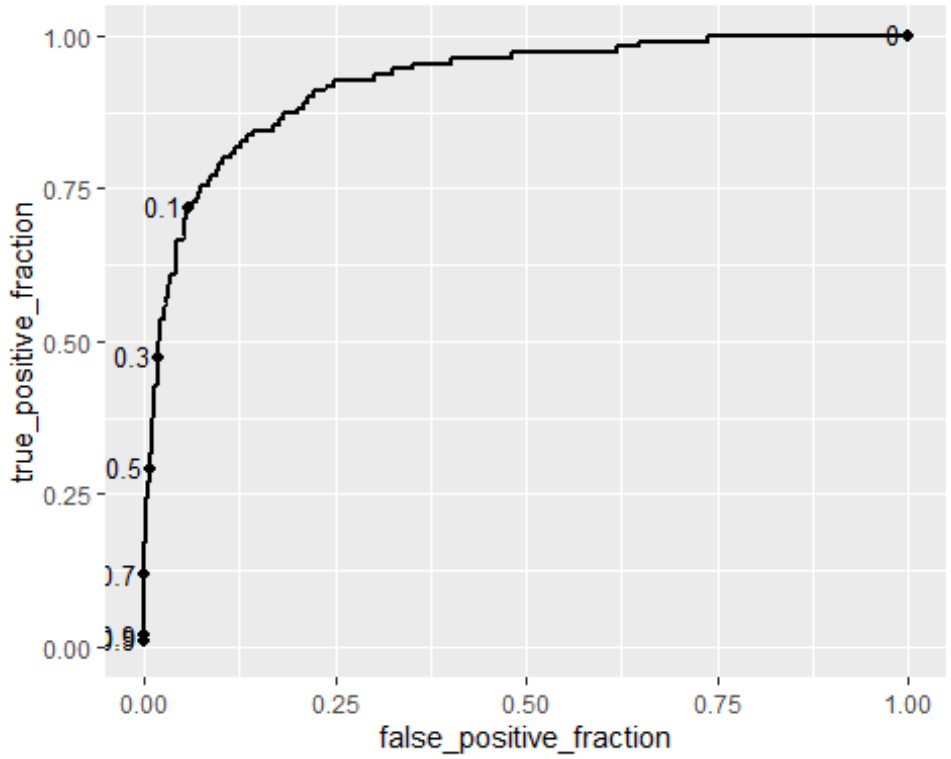
f - Fitted Models

```
preds_train_DF <- data.frame(
  scores_train = predict(logit_train,
                         newdata = movies_train,
                         type = "response"),
  movies_train
)

preds_test_DF <- data.frame(
  scores_test = predict(logit_train,
                        newdata = movies_test,
                         type = "response"),
  movies_test
)
```

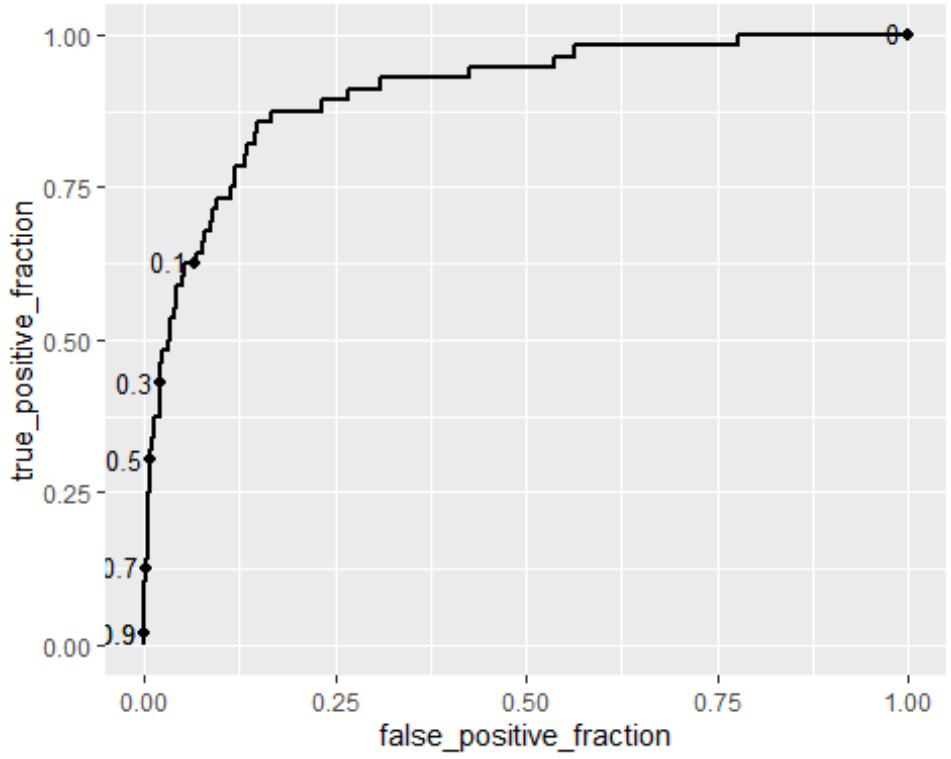
g - Plot! Plot! Plot!

```
trainROC <- ggplot(data = preds_train_DF,
                   aes(m = scores_train,
                       d = blockbuster)) +
  geom_roc(labelsize = 3.5,
           cutoffs.at = c(.99,.9,.7,.5,.3,.1,0))

testROC <- ggplot(data = preds_test_DF,
                  aes(m = scores_test,
                      d = blockbuster)) +
  geom_roc(labelsize = 3.5,
           cutoffs.at = c(.99,.9,.7,.5,.3,.1,0))



LOOCV_ROC <- ggplot(data = preds_LOOCV_DF,
                  aes(m = scores_LOOCV_train,
                      d = blockbuster)) +
  geom_roc(labelsize = 3.5,
           cutoffs.at = c(.99,.9,.7,.5,.3,.1,0))
```

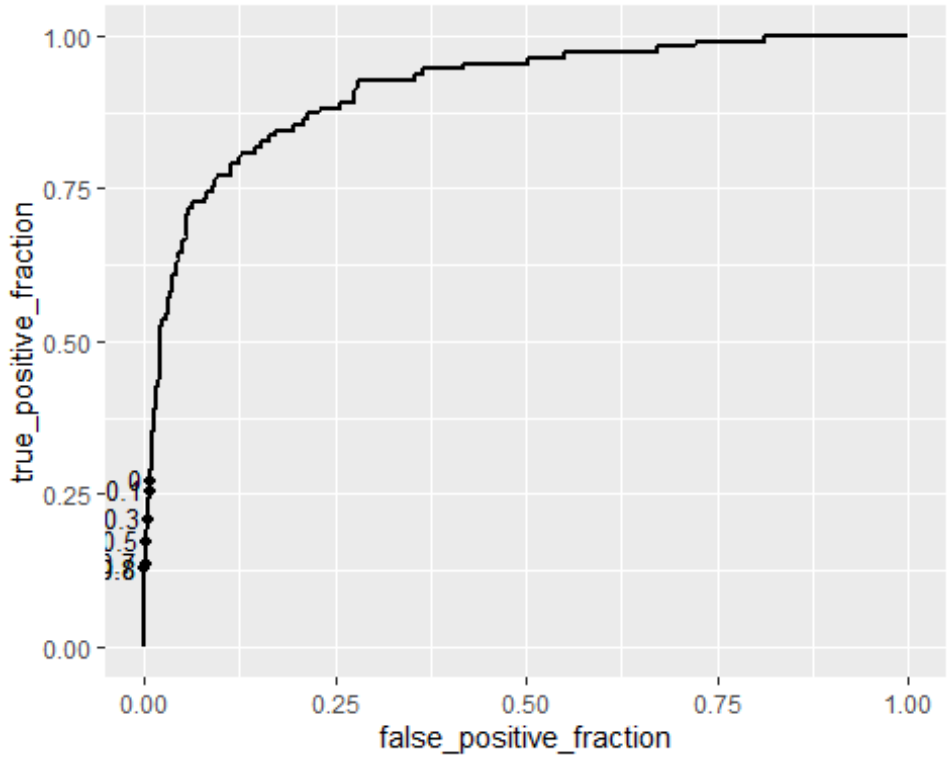
```
plot(trainROC)
```




```
plot(testROC)
```




```
plot(LOOCV_ROC)
```



The three ROC curves show how each model performed in terms of the true to false positive tradeoff. Comparing them all at the same cutoff (say 0.1) we can compare each model's relative accuracy in terms of the true and false positive tradeoffs.

The training model (graph 1) shows in getting almost 75% of the true positives we have an approximately 8% false positive rate.

Our test model (graph 2) only gets us about 62% of the true positives with roughly the same false positive rate.

The LOOCV model at a cutoff of 0.1 indicates that although we are predicting close to 0 false positives we are only able to predict roughly 25% of true positives.

h - ROC and AUC

```
calc_auc(trainROC)
##   PANEL group       AUC
## 1     1    -1 0.9239733
calc_auc(LOOCV_ROC)
##   PANEL group       AUC
## 1     1    -1 0.9125546
calc_auc(testROC)
##   PANEL group       AUC
## 1     1    -1 0.9060054
```

We suppose the models are ordered the way they are becuase of possible model overfitting from the training data or having picked a poor sample for the test set. LOOCV_ROC being the second highest seems weird considering the weird cutoff situation before. There may be an error in the code somewhere or something else at play.

i - Upsampling and Downsampling with ROSE

```
rose_down <- ROSE(blockbuster ~budgetM + top_director + cast_total_facebook_l
ikes000s + content_rating + genre_main,
                  data = movies_train,
                  N = 5000, p = 1/2)
rose_up <- ROSE(blockbuster ~budgetM + top_director + cast_total_facebook_lik
es000s + content_rating + genre_main,
                  data = movies_train,
                  N = 220, p = 1/2)


logit_down <- glm(blockbuster ~ budgetM + top_director + cast_total_facebook_
likes000s + content_rating + genre_main,
                  data =rose_down$data,
                  family = "binomial")
summary(logit_down)
##
## Call:
## glm(formula = blockbuster ~ budgetM + top_director + cast_total_facebook_l
ikes000s +
```

```
##     content_rating + genre_main, family = "binomial", data = rose_down$dat
a)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5321  -0.4832  -0.0001   0.4694   3.0908
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -1.255e+00  1.315e-01  -9.546  < 2e-16
## budgetM                        2.124e-02  9.102e-04  23.335  < 2e-16
## top_director                   1.000e+00  8.500e-02  11.768  < 2e-16
## cast_total_facebook_likes000s  1.058e-02  2.235e-03   4.734 2.20e-06
## content_ratingPG-13           -7.294e-01  1.117e-01  -6.528 6.68e-11
## content_ratingR               -2.023e+00  1.420e-01 -14.242  < 2e-16
## content_ratingOther           -2.311e-01  1.918e-01  -1.205   0.2283
## genre_mainAdventure            3.235e-01  1.276e-01   2.536   0.0112
## genre_mainComedy              -8.596e-01  1.281e-01  -6.711 1.94e-11
## genre_mainCrime               -1.595e+01  2.764e+02  -0.058   0.9540
## genre_mainDrama               -1.162e+00  1.646e-01  -7.059 1.68e-12
## genre_mainOther               -3.530e-01  1.502e-01  -2.350   0.0188
##
## (Intercept)                   ***
## budgetM                       ***
## top_director                  ***
## cast_total_facebook_likes000s ***
## content_ratingPG-13           ***
## content_ratingR               ***
## content_ratingOther
## genre_mainAdventure           *
## genre_mainComedy              ***
## genre_mainCrime
## genre_mainDrama               ***
## genre_mainOther               *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6931.4  on 4999  degrees of freedom
## Residual deviance: 3461.8  on 4988  degrees of freedom
## AIC: 3485.8
##
## Number of Fisher Scoring iterations: 16
```

```
logit_up <- glm(blockbuster ~ budgetM + top_director + cast_total_facebook_li
kes000s + content_rating + genre_main,
                  data =rose_up$data,
                  family = "binomial")
```

```
summary(logit_up)
##
## Call:
## glm(formula = blockbuster ~ budgetM + top_director + cast_total_facebook_l
ikes000s +
##     content_rating + genre_main, family = "binomial", data = rose_up$data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7983  -0.3364   0.0227   0.2781   3.2618
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -1.977e+00  7.910e-01  -2.500   0.0124
## budgetM                        3.574e-02  6.281e-03   5.690 1.27e-08
## top_director                   6.527e-01  4.952e-01   1.318   0.1874
## cast_total_facebook_likes000s  2.886e-02  1.357e-02   2.128   0.0334
## content_ratingPG-13           -1.382e+00  6.177e-01  -2.238   0.0252
## content_ratingR               -1.571e+00  7.169e-01  -2.191   0.0285
## content_ratingOther            5.132e-01  9.757e-01   0.526   0.5989
## genre_mainAdventure           -4.325e-01  7.115e-01  -0.608   0.5433
## genre_mainComedy              -1.054e+00  8.085e-01  -1.304   0.1923
## genre_mainCrime               -1.599e+01  1.364e+03  -0.012   0.9907
## genre_mainDrama               -7.658e-01  7.952e-01  -0.963   0.3356
## genre_mainOther               -9.365e-01  7.469e-01  -1.254   0.2099
##
## (Intercept)                   *
## budgetM                       ***
## top_director
## cast_total_facebook_likes000s *
## content_ratingPG-13           *
## content_ratingR               *
## content_ratingOther
## genre_mainAdventure
## genre_mainComedy
## genre_mainCrime
## genre_mainDrama
## genre_mainOther
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 304.69  on 219  degrees of freedom
## Residual deviance: 118.85  on 208  degrees of freedom
## AIC: 142.85
##
## Number of Fisher Scoring iterations: 16
```

j - Sensitivity and Specificity

```
J_DF_down <- data.frame(
  scores_J = predict(logit_down,newdata= movies_test,
                     type = "response"), movies_test)
J_DF_up <- data.frame(
  scores_J1 = predict(logit_up,newdata=movies_test,
                      type = "response"), movies_test)
J_DF_c <- data.frame(
  scores_J2 = predict(logit_train,newdata=movies_test,
                      type = "response"), movies_test)

J_DF_down %<>% mutate(class_pred05=ifelse(scores_J>0.5,1,0))
J_DF_up %<>% mutate(class_pred06=ifelse(scores_J1>0.5,1,0))
J_DF_c %<>% mutate(class_pred07=ifelse(scores_J2>0.5,1,0))

table(movies_test$blockbuster,J_DF_down$class_pred05)
##
##       0   1
##   0 749 127
##   1   9  47
table(movies_test$blockbuster,J_DF_up$class_pred06)
##
##       0   1
##   0 759 117
##   1  13  43
table(movies_test$blockbuster,J_DF_c$class_pred07)
##
##       0   1
##   0 869   7
##   1  40  16
```

Sensitivity for the upsampled model is 47/(47+127) = .27

Sensitivity for the downsampled model is 43/(43+117) = .268

Sensitivity for the logistic model is 16/(16+7) = .69

Specificity for the upsampled model is 749/749+9 = .98

Specificity for the downsampled model is 759/759+13 = .98

Specificity for the logistic model is 869/(869+40) = .95

The up sampling and the down sampling model provides little benefit in the specificity because it reduces sensitivity by about 40%. To predict blockbuster movies, we should use the logit model as a reference because it gives us the best sensitivity/specificity

combination. The logit model is also preferable because we are able to better predict whether a movie will be a blockbuster or not.