Problem Set 7

Noah Estrada-Rand

10/16/2019

a) Create New Variables, Clean Data, and Split into Test and Train Sets

```
options(scipen = 50)
movies <- read.csv("movie_metadata.csv")
# removing missing values
movies <- movies[complete.cases(movies),]
# removing empty content rating or not rated
movies <- movies[(movies$content_rating != "" & movies$content_rating != "Not Rated"), ]
# removing movies with budget > 400M
movies <- movies[movies$budget < 400000000,]
# creating budget, gross, and profit columns in millions
movies$grossM <- movies$gross/1e+6
movies$budgetM <- movies$budget/1e+6
movies$profitM <- movies$grossM - movies$budgetM
# creating a column for main genre
movies$genre_main <- do.call('rbind',strsplit(as.character(movies$genres), '|', fixed=TRUE))[,1]
```

```
## Warning in rbind(c("Action", "Adventure", "Fantasy", "Sci-Fi"),
## c("Action", : number of columns of result is not a multiple of vector
## length (arg 2)
```

```
# creating a dummy for blockbuster movies
movies$blockbuster <- ifelse(movies$grossM > 200, 1, 0)
library(forcats)
movies$genre_main <- fct_lump(movies$genre_main,5)
movies$content_rating <- fct_lump(movies$content_rating,3)
movies$country <- fct_lump(movies$country,2)
movies$cast_total_facebook_likes000s <- movies$cast_total_facebook_likes / 1000
# top director
director_props <- data.frame(prop.table(table(movies$director_name)))
directors_indx <- order(director_props$Freq,decreasing = TRUE)
top_directors_indx <- directors_indx[1:floor(0.1*nrow(director_props))]
top_directors_names <- director_props[top_directors_indx, 1]
movies$top_director <- ifelse(movies$director_name %in% top_directors_names, 1, 0)
# train/test split
set.seed(1861)
train_idx <- sample(1:nrow(movies),size = floor(0.75*nrow(movies)))
movies_train <- movies[train_idx,]
movies_test <- movies[-train_idx,]
```

b) Comparing means

```
mean(movies_train$blockbuster)
```

```
## [1] 0.03935599
```

```
mean(movies_test$blockbuster)
```

```
## [1] 0.06008584
```

```
t.test(movies_train$blockbuster,movies_test$blockbuster)
```

```
##
##  Welch Two Sample t-test
##
## data:  movies_train$blockbuster and movies_test$blockbuster
## t = -2.4067, df = 1370, p-value = 0.01623
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.037626956 -0.003832732
## sample estimates:
##  mean of x  mean of y
## 0.03935599 0.06008584
```

Based on the above T-test it becomes clear that there is a statistically significant difference between the means of both the test and train set values for the blockbuster category. This is due to the fact that the t-test yielded a .01623 pvalue which is less than the standard .05 alpha value for significance.

c) Creating Logistic Model for Blockbuster Data

```
logit_1 <- glm(blockbuster ~ budgetM + top_director + cast_total_facebook_likes000s+
                 content_rating + genre_main, family = binomial,
               data = movies_train)
summary(logit_1)
```

```
##
## Call:
## glm(formula = blockbuster ~ budgetM + top_director + cast_total_facebook_likes000s +
##     content_rating + genre_main, family = binomial, data = movies_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3373  -0.1923  -0.1099  -0.0538   3.5763
##
## Coefficients:
##                                 Estimate Std. Error z value
## (Intercept)                    -4.818430   0.420363 -11.463
## budgetM                         0.022993   0.002165  10.622
## top_director                    0.595282   0.266037   2.238
## cast_total_facebook_likes000s   0.006687   0.002775   2.410
## content_ratingPG-13            -0.196937   0.310937  -0.633
## content_ratingR                -1.920051   0.526607  -3.646
## content_ratingOther             0.390799   0.502856   0.777
## genre_mainAdventure             0.434639   0.331964   1.309
## genre_mainComedy               -0.442454   0.451648  -0.980
## genre_mainCrime               -14.558908 737.031912  -0.020
## genre_mainDrama                -0.461686   0.518009  -0.891
## genre_mainOther                -0.065099   0.527714  -0.123
##                                           Pr(>|z|)
```

```
## (Intercept)                   < 0.0000000000000002 ***
## budgetM                       < 0.0000000000000002 ***
## top_director                                0.025248 *
## cast_total_facebook_likes000s               0.015971 *
## content_ratingPG-13                         0.526494
## content_ratingR                             0.000266 ***
## content_ratingOther                         0.437065
## genre_mainAdventure                         0.190434
## genre_mainComedy                            0.327262
## genre_mainCrime                             0.984240
## genre_mainDrama                             0.372785
## genre_mainOther                             0.901821
## ---
## 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: 556.41  on 2783  degrees of freedom
## AIC: 580.41
##
## Number of Fisher Scoring iterations: 18
```

d) Interpretation of Coefficients

```
exp(logit_1$coefficients)
```

```
##                   (Intercept)                       budgetM
##               0.0080794633498               1.0232589268520
##                  top_director cast_total_facebook_likes000s
##               1.8135424882354               1.0067094885772
##           content_ratingPG-13               content_ratingR
##               0.8212421783181               0.1465995112896
##           content_ratingOther           genre_mainAdventure
##               1.4781614419006               1.5444052754299
##              genre_mainComedy               genre_mainCrime
##               0.6424576441243               0.0000004754959
##               genre_mainDrama               genre_mainOther
##               0.6302204930683               0.9369742427844
```

Looking at the coefficient for content_ratingR it appears that R-rated movies are 85.34% less likely to be a blockbuster than G rated movies, when all other variables are controlled. Furthermore, when considering the coefficient for genre_mainAdventure, it also becomes apparent that adventure themed movies are 54.44% more likely to be a blockbuster than action movies. And lastly, when observing the coefficient for top_directory, we find that a movie that is directed by what is considered a "top director" is 81.35% more likely to be a blockbuster than a movie without a top director.

e) Creating Training and Test Predictions

```
preds_train <- data.frame(movies_train,
                          predictions = predict(logit_1,type = "response"))
preds_test <- data.frame(movies_test,
```

```
                          predictions = predict(logit_1,newdata = movies_test,type = "response"))
```

f) Using LOOCV to make Predictions

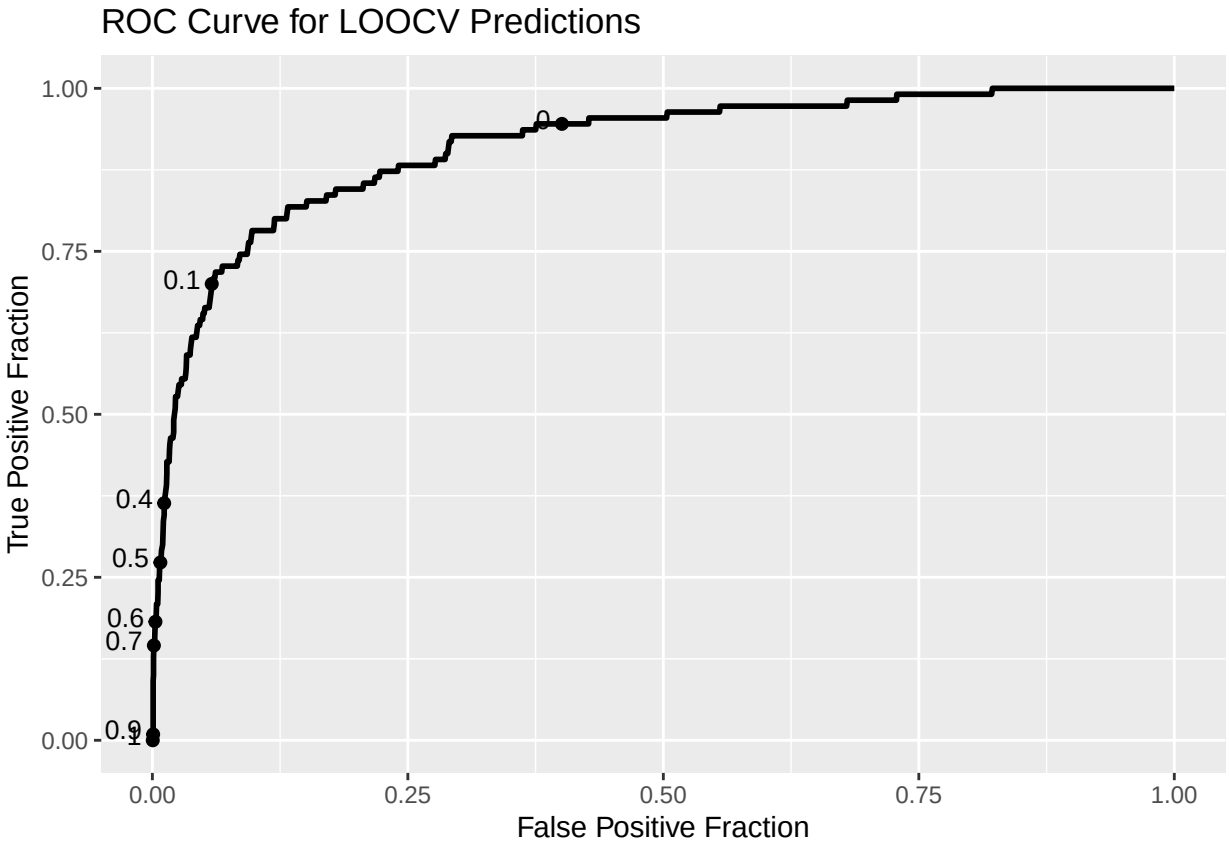
```
preds_LOOCV = NULL;
for(i in 1:nrow(movies_train)){
  mod <- glm(blockbuster ~ budgetM + top_director + cast_total_facebook_likes000s+
               content_rating +genre_main, family = binomial,
             data = movies_train[-i,])
  preds_LOOCV[i] <- predict(mod,newdata = movies_train[i,],type = "response")
}
head(preds_LOOCV)
```

```
## [1] 0.000765992082658 0.000000005332603 0.009077669902434 0.020670757289352
## [5] 0.054385033102088 0.051142956397321
```

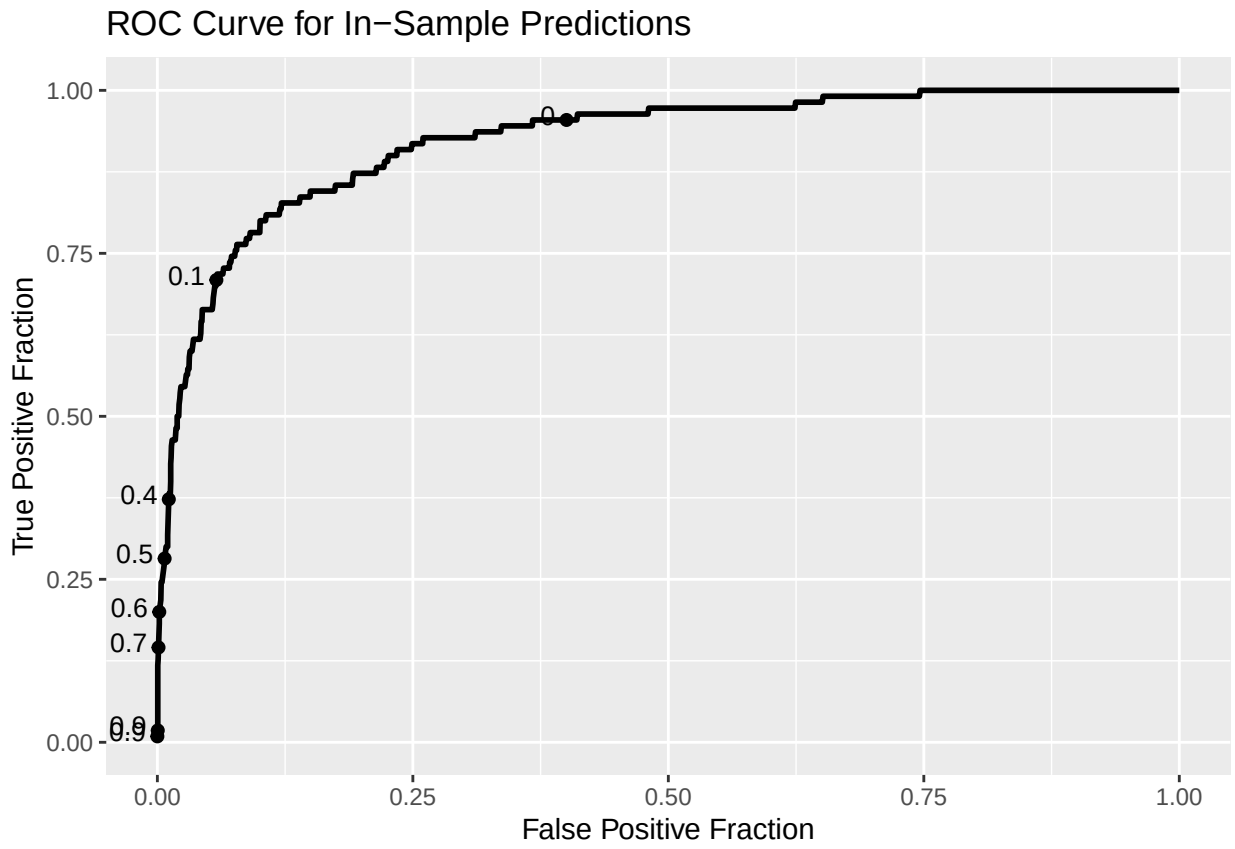
```
preds_train <- data.frame(preds_train,loocvPreds = preds_LOOCV)
```

g) Three ROC Curves

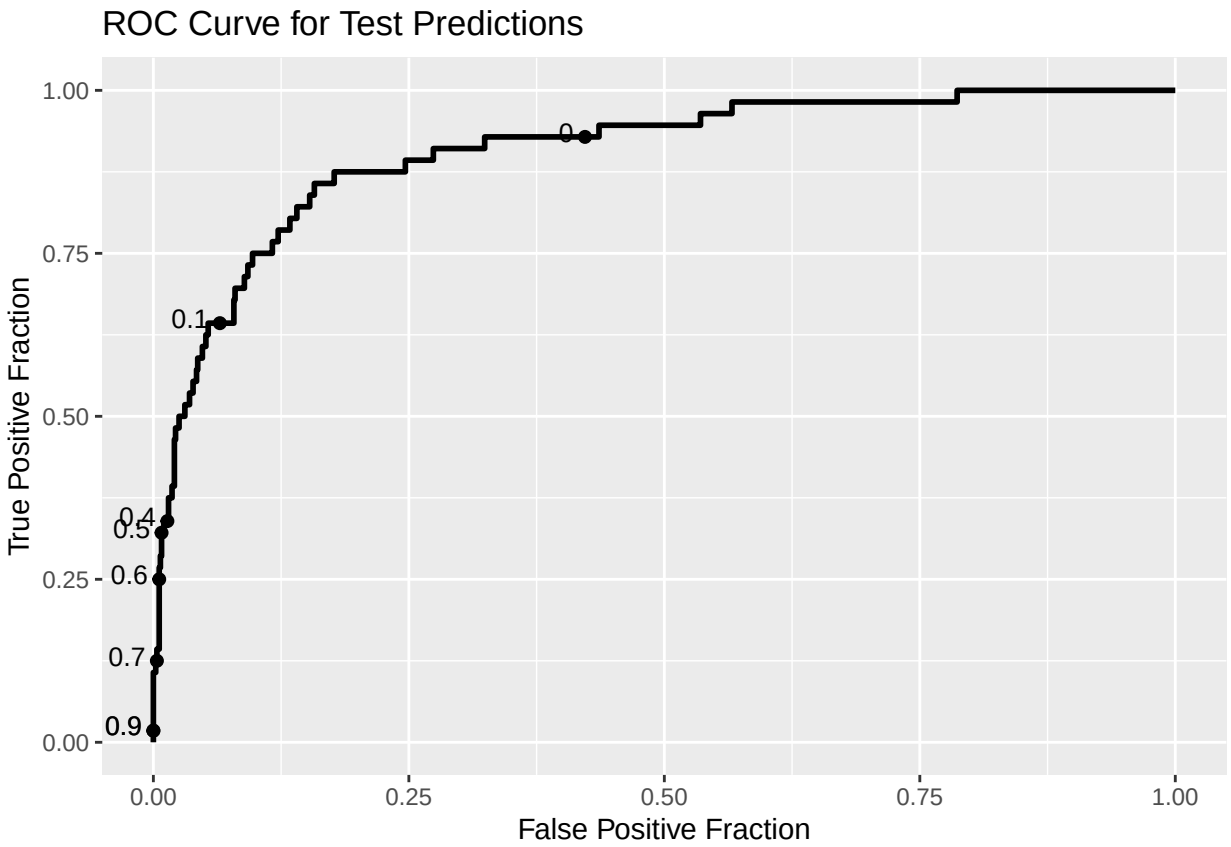
```
loocvROC <- ggplot(preds_train,aes(m = loocvPreds,
                                   d = blockbuster)) +
  geom_roc(labelsize = 3.5,
           cutoffs.at = c(.99,.9,.7,.6,.5,.4,.1,.01)) +
  labs(title = "ROC Curve for LOOCV Predictions",x = "False Positive Fraction",
       y= "True Positive Fraction")
inSampleROC <- ggplot(preds_train,aes(m = predictions,
                                      d = blockbuster)) +
  geom_roc(labelsize = 3.5,
           cutoffs.at = c(.99,.9,.7,.6,.5,.4,.1,.01)) +
  labs(title = "ROC Curve for In-Sample Predictions",x = "False Positive Fraction",
       y= "True Positive Fraction")
testROC <- ggplot(preds_test,aes(m = predictions,
                                      d = blockbuster)) +
  geom_roc(labelsize = 3.5,
           cutoffs.at = c(.99,.9,.7,.6,.5,.4,.1,.01)) +
  labs(title = "ROC Curve for Test Predictions",x = "False Positive Fraction",
       y= "True Positive Fraction")
loocvROC
```




```
inSampleROC
```


ROC Curve for In−Sample Predictions
True Positive Fraction
False Positive Fraction
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
0
0.1
0.4
0.5
0.6
0.7
0.9

```
testROC
```



In relation to one another, the curves are very similar. However, upon inspection it appears that the ROC curve for test predictions has much less difference between the .7, .6, .5, and .4 cutoffs in terms of true positive fraction. Both the sample and loocv ROC curves had larger differences between the same values for true positive fraction. Yet, overall, the loocv ROC curve appears more leftward and upward than the other two graphs, indicating that the model for the given cutoff points yielded higher true positive and lower false positive amounts than the in sample and test predictions.

h)

```
calc_auc(loocvROC)
```

```
##   PANEL group       AUC
## 1     1    -1 0.9109548
```

```
calc_auc(inSampleROC)
```

```
##   PANEL group       AUC
## 1     1    -1 0.9224581
```

```
calc_auc(testROC)
```

```
##   PANEL group       AUC
## 1     1    -1 0.9056385
```

In order of importance, we place the in sample predic as the highest importance, the loocv predictions as next highest then the test as third highest. We observe this order given the fact that the predictions for the sample data were trained on the sample data, which typtically leads to a better ability to distinguish between blockbusters and movies that are not. Moreover, the fact that the test data comes in at the model of lowest

importance is conerning, as this sugggests that there is significant variance in the data not accounted for by the initial model. And lastly, while the loocv ROC only trails the in sample ROC by .01, this may be due to the fact that overall the data has high variance, leading to less accurate predictions of blockbuster status.