Problem Set 6

Noah Estrada-Rand

10/6/2019

a) Creating test and training sets

```
Boston$PriceyHome <- ifelse(Boston$medv > 40, 1, 0)

Boston$chas <- factor(Boston$chas)##creates a factor
set.seed(2019)
trainSize <- 0.75
train_idx <- sample(1:nrow(Boston), size = floor(nrow(Boston) * trainSize))
housing_train <- Boston[train_idx,]
housing_test <- Boston[-train_idx,]
```

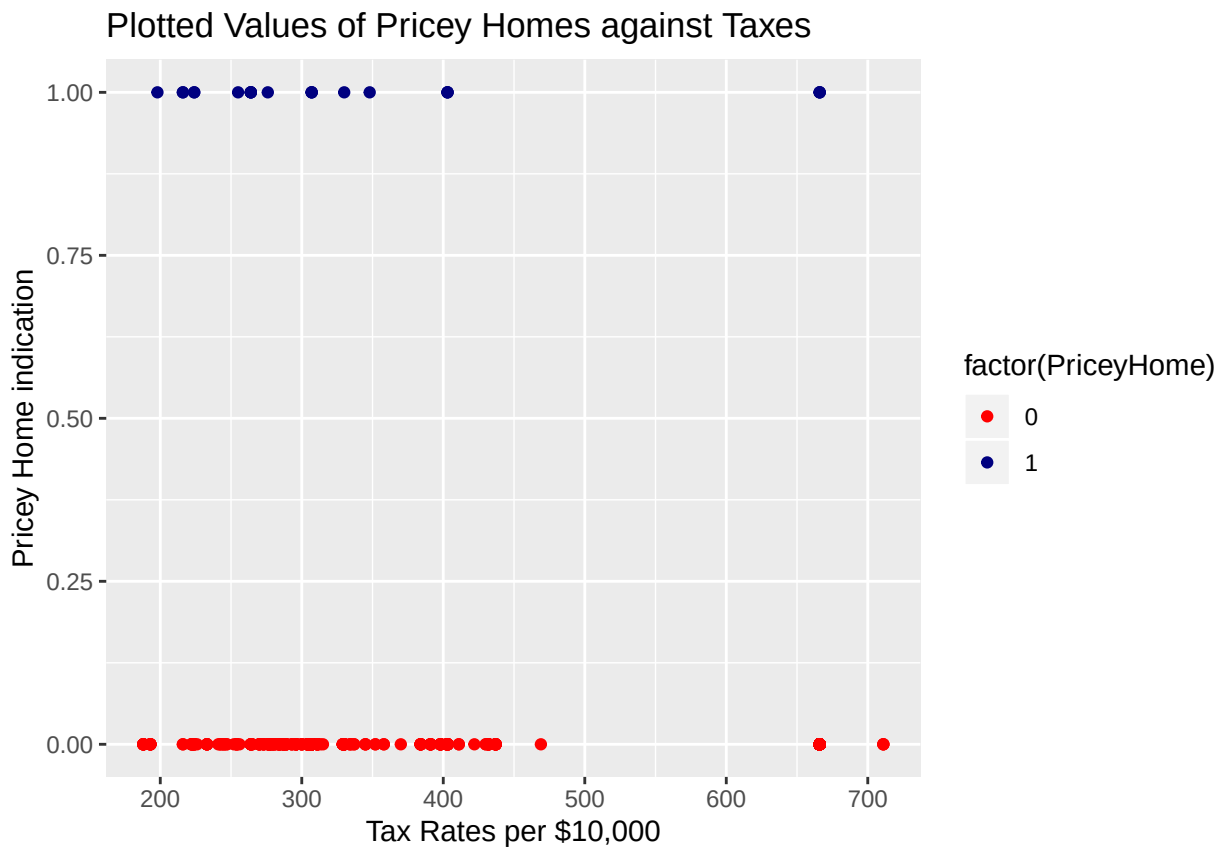
b) Mean Value of Variables Across Pricey And Non-Pricey Homes

```
summaryBy( . ~ PriceyHome,data = housing_train)
```

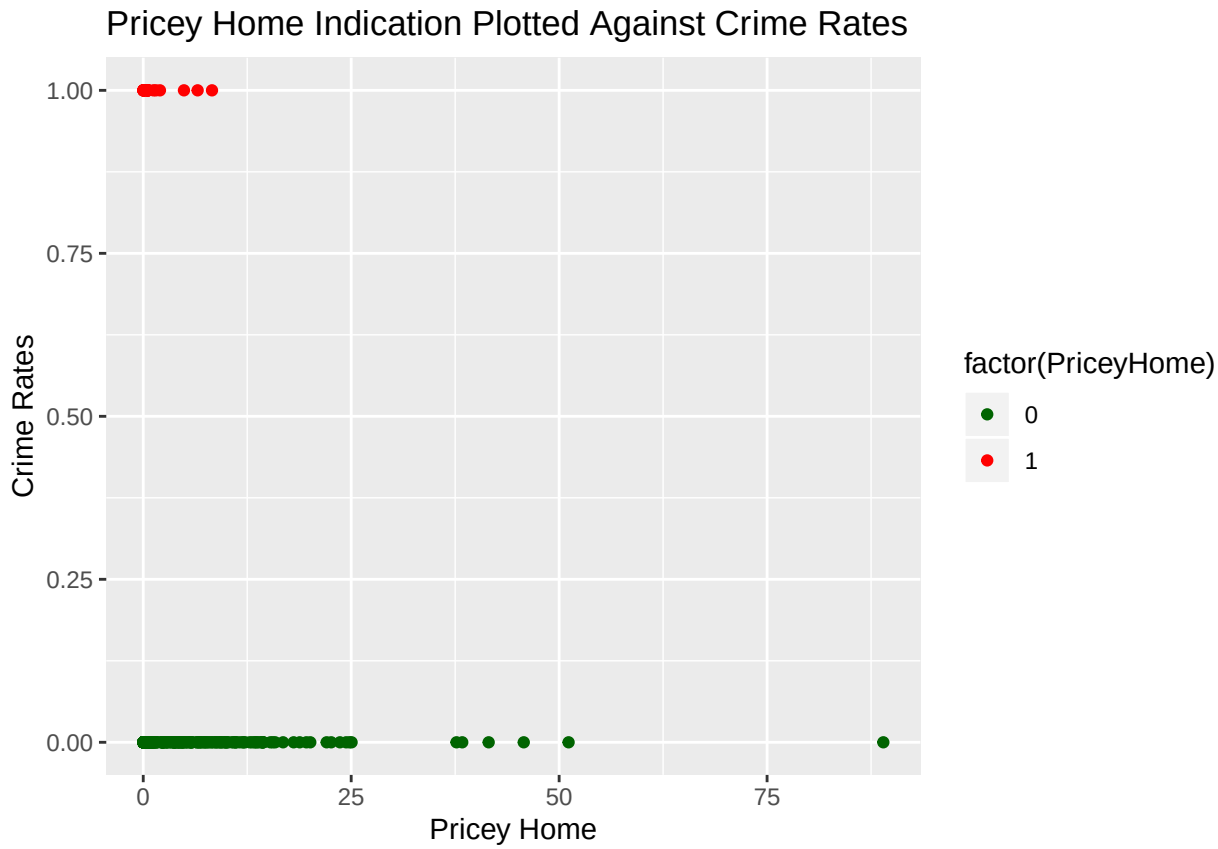
```
##   PriceyHome crim.mean  zn.mean indus.mean  nox.mean  rm.mean age.mean
## 1          0  3.631687 11.13025  11.321653 0.5563297 6.206986 68.43361
## 2          1  1.308569 26.56818   7.817727 0.5277727 7.735227 65.14545
##   dis.mean rad.mean tax.mean ptratio.mean black.mean lstat.mean medv.mean
## 1 3.847307 9.831933 412.9804     18.57451   354.9478  13.252493  21.11064
## 2 3.596964 7.500000 339.5909     15.81364   384.3741   4.240909  47.36364
```

From the above results it becomes apparent that pricey neighborhoods in Boston deviate from non-pricey neighborhoods on mulitple dimensions. In terms of crime rate, the mean rate of crime in pricey neighborhoods is much lower than that of less pricey neighborhoods. Moreover, the amount of zoning for residential land over 25,000 square feet is almost double the proportion zoned in non-pricey neighborhoods. Interstingly, the property tax rate per $10,000 is also lower in pricier neighborhoods. Lastly, the lower status of the population percent is lower in pricey neighborhoods than less pricey neighborhoods, while the median value of owner occupied homes is much higher in pricier neighborhoods. # c)

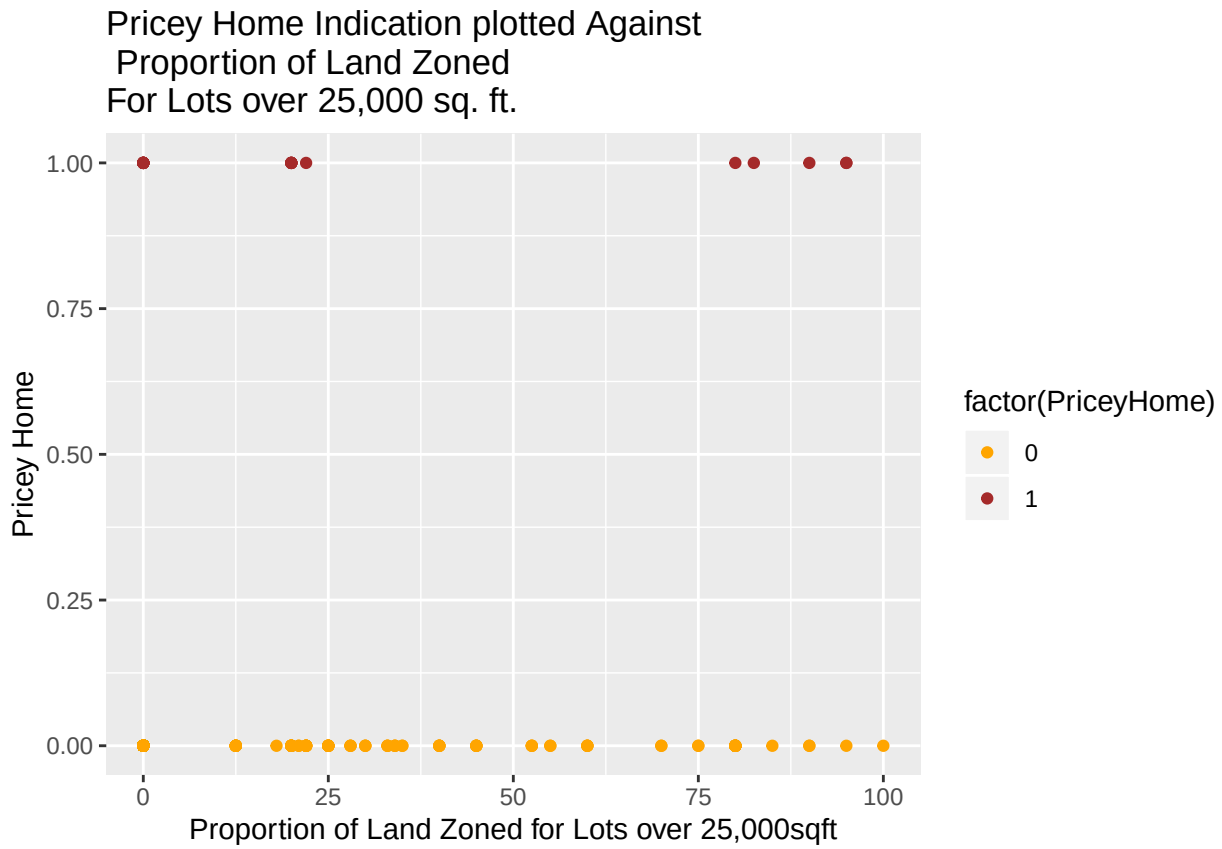
```
ggplot(housing_train,aes(x = tax,y = PriceyHome,color = factor(PriceyHome))) + geom_point() +
  scale_color_manual(values = c("red","navy blue")) +
  labs(title = "Plotted Values of Pricey Homes against Taxes",
       y = "Pricey Home indication",
       x = "Tax Rates per $10,000")
```




```
ggplot(housing_train,aes(x =crim,y = PriceyHome,color = factor(PriceyHome))) +geom_point() +
  scale_color_manual(values = c("dark green","red")) +
  labs(title = "Pricey Home Indication Plotted Against Crime Rates",
       y = "Crime Rates",
       x = "Pricey Home")
```




```
ggplot(housing_train,aes(x = zn,y = PriceyHome,color = factor(PriceyHome))) +geom_point()+
  scale_color_manual(values =c("orange","brown")) +
  labs(title = "Pricey Home Indication plotted Against\n Proportion of Land Zoned\nFor Lots over 25,000
       x = "Proportion of Land Zoned for Lots over 25,000sqft",
       y = "Pricey Home")
```



From the above plots it becomes apparent that neighborhoods labeled as "pricey" have lower taxes rates, in general, than do non pricey neighborhoods. Moreover, we are able to observe that all "pricey" neighborhoods are under 20% crime rate while the range of crime rates for non pricey neighborhoods is much broader and reaches higher. Lastly, when evaluating the proportion of land zoned for residential lots over 25,000 sq.ft. results are not as stark, wit pricey neighborhoods existing at the extremes of land zoned while non pricey neighborhoods have proportions at all levels.

d) Logit Model of PriceyHome by Charles River Location

```
mod1 <- glm(PriceyHome~chas,data = housing_train,family = binomial)
summary(mod1)
```

```
##
## Call:
## glm(formula = PriceyHome ~ chas, family = binomial, data = housing_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7002  -0.3128  -0.3128  -0.3128   2.4665
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.9928     0.2485 -12.042  < 2e-16 ***
## chas1         1.7119     0.5633   3.039  0.00237 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 167.94  on 378  degrees of freedom
## Residual deviance: 160.68  on 377  degrees of freedom
## AIC: 164.68
##
## Number of Fisher Scoring iterations: 5
```

```
exp(mod1$coefficients)
```

```
## (Intercept)       chas1
##  0.05014749  5.53921562
```

Based on the coefficients above, it appears that a neighborhood located adjacent the Charles River leads to that neighborhood being 453.92% more likely to being placed in the "pricey" category. In other words, when compared to neighborhoods not adjacent to the Charles river, the probability of having median home values above $40,000 is four and a half times higher than that of non pricey neighborhoods.

e)

```
mod2 <-glm(PriceyHome ~ chas + crim + lstat + ptratio + zn + rm + tax + rad + nox,
           family = binomial,
           data = housing_train)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(mod2)
```

```
##
## Call:
## glm(formula = PriceyHome ~ chas + crim + lstat + ptratio + zn +
##     rm + tax + rad + nox, family = binomial, data = housing_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4310  -0.0330  -0.0031  -0.0001   2.7393
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.599874   9.209524  -0.174 0.862086
## chas1        0.756749   1.775917   0.426 0.670022
## crim         0.103148   0.075421   1.368 0.171429
## lstat       -1.193118   0.352360  -3.386 0.000709 ***
## ptratio     -0.722816   0.298585  -2.421 0.015486 *
## zn          -0.013093   0.017917  -0.731 0.464930
## rm           1.885059   0.672090   2.805 0.005035 **
## tax         -0.002658   0.006963  -0.382 0.702611
## rad          0.313567   0.167422   1.873 0.061080 .
## nox          6.928135   7.016511   0.987 0.323444
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 167.943  on 378  degrees of freedom
## Residual deviance:  45.662  on 369  degrees of freedom
## AIC: 65.662
##
## Number of Fisher Scoring iterations: 10
```

```
exp(mod2$coefficients)
```

```
##  (Intercept)        chas1         crim        lstat      ptratio
##    0.2019220    2.1313356    1.1086552    0.3032743    0.4853837
##           zn           rm          tax          rad          nox
##    0.9869924    6.5867450    0.9973452    1.3682976 1020.5890271
```

From the above coefficients, it appears that being located along the Charles River does not have a statistically significant impact on predicting whether or not a neighborhood is in the pricey category. With the addition of other variables from the Boston dataset, it becomes apparent that the location of a neighborhood relative to the Charles River is a confounding variable. This is due to the fact that its effect changes significantly when considered with other variables, indicating that it changes systematically with other variables, making it an unfavorable predictor.

f) Generated Class Predictions

```
housing_train$priceyProb <- predict(mod2,type = "response")
housing_test$priceyProb <- predict(mod2,type = "response",newdata = housing_test)

housing_train$PriceyPred <- ifelse(housing_train$priceyProb > .5,1,0)
housing_test$PriceyPred <- ifelse(housing_test$priceyProb >.5,1,0)
head(housing_train$PriceyPred)
```

```
## [1] 1 0 0 0 0 0
```

```
head(housing_test$PriceyPred)
```

```
## [1] 0 0 0 0 0 0
```

g)

Training Data Confusion Matrix

```
name_of_rows <- c("Not Pricey","Pricey","Sum")
name_of_cols <- c("Predicted Not","Predicted Pricey","Sum")
train_confusion <- table(housing_train$PriceyHome,housing_train$PriceyPred)
train_confusion <- addmargins(train_confusion)
rownames(train_confusion) <- name_of_rows
colnames(train_confusion) <- name_of_cols
print(train_confusion)
```

```
##
##              Predicted Not Predicted Pricey Sum
##   Not Pricey           355                2 357
##   Pricey                 6               16  22
##   Sum                  361               18 379
```

Test Data Confusion Matrix

```
test_confusion <- table(housing_test$PriceyHome,housing_test$PriceyPred)
test_confusion <- addmargins(test_confusion)
rownames(test_confusion) <- name_of_rows
colnames(test_confusion) <- name_of_cols
print(test_confusion)
```

```
##
##              Predicted Not Predicted Pricey Sum
##   Not Pricey           117                1 118
##   Pricey                 3                6   9
##   Sum                  120                7 127
```

Training and Test Accuracy Based on Logit Model

```
train_accuracy <- (sum(housing_train$PriceyHome == 1 & housing_train$PriceyPred == 1)+
                     sum(housing_train$PriceyHome == 0 & housing_train$PriceyPred == 0))/length(housing_

test_accuracy <- (sum(housing_test$PriceyHome == 1 & housing_test$PriceyPred == 1)+
                     sum(housing_test$PriceyHome == 0 & housing_test$PriceyPred == 0))/length(housing_tes
print(train_accuracy)
```

```
## [1] 0.9788918
```

```
print(test_accuracy)
```

```
## [1] 0.9685039
```

Training and Test Sensitivity/True Positive Rate

```
train_sensitivity <- sum(housing_train$PriceyHome == 1 & housing_train$PriceyPred == 1)/
  (sum(housing_train$PriceyHome == 1))

test_sensitivity <-sum(housing_test$PriceyHome == 1 & housing_test$PriceyPred == 1)/
  sum(housing_test$PriceyHome == 1)
print(train_sensitivity)
```

```
## [1] 0.7272727
```

```
print(test_sensitivity)
```

```
## [1] 0.6666667
```

Training and Test Specificity/True Negative Rate

```
train_specificity <- sum(housing_train$PriceyHome == 0 & housing_train$PriceyPred == 0)/
  sum(housing_train$PriceyHome == 0)

test_specificity <- sum(housing_test$PriceyHome == 0 & housing_test$PriceyPred == 0)/
  sum(housing_test$PriceyHome == 0)
print(train_specificity)
```

```
## [1] 0.9943978
```

```
print(test_specificity)
```

```
## [1] 0.9915254
```

Training and Test False Positives

```
train_false_positive <- sum(housing_train$PriceyHome == 0 & housing_train$PriceyPred == 1)

test_false_positive <- sum(housing_test$PriceyHome == 0 & housing_test$PriceyPred == 1)

print(train_false_positive)
```

```
## [1] 2
```

```
print(test_false_positive)
```

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

Training and Test False Negatives

```
train_false_negative <- sum(housing_train$PriceyHome == 1 & housing_train$PriceyPred == 0)

test_false_negative <- sum(housing_test$PriceyHome == 1 & housing_test$PriceyPred == 0)

print(train_false_negative)
```

```
## [1] 6
```

```
print(test_false_negative)
```

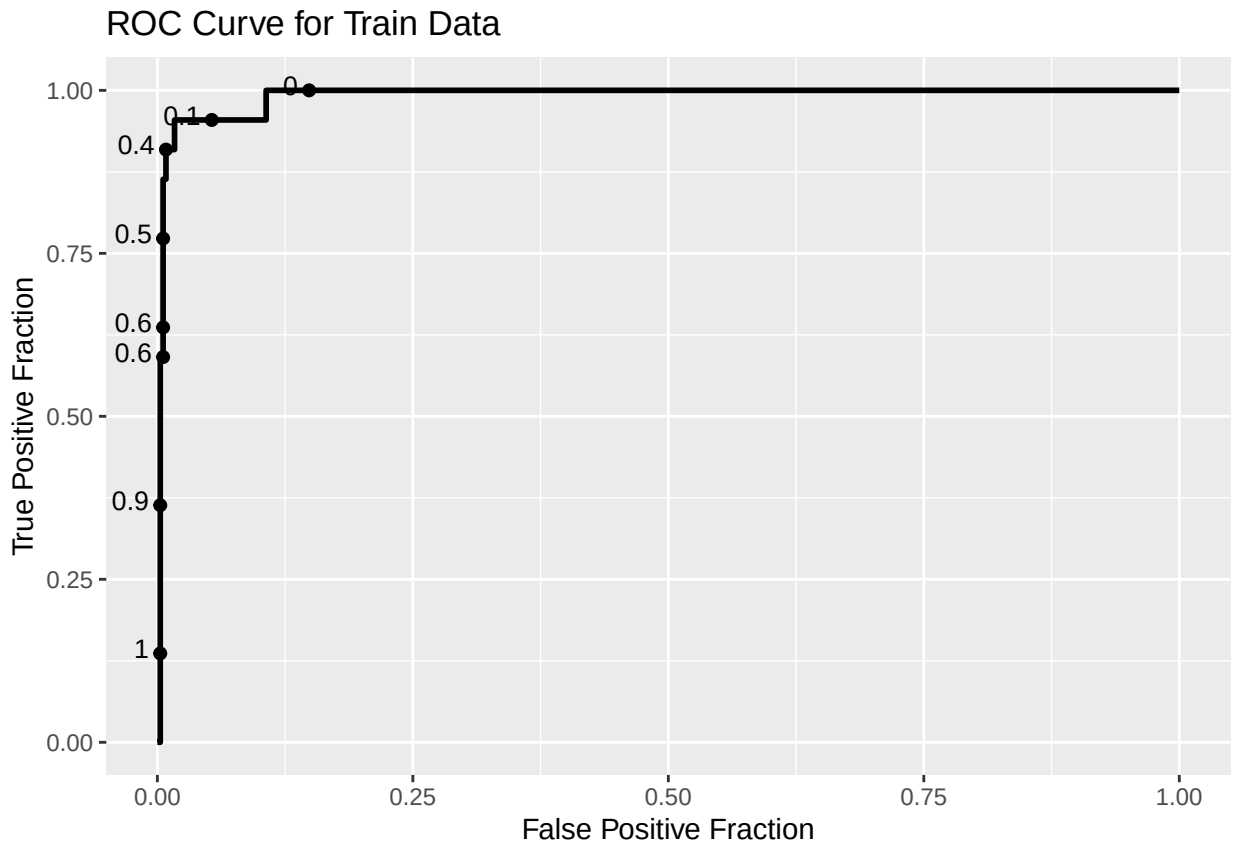
```
## [1] 3
```

h) Adjusting the Cutoff

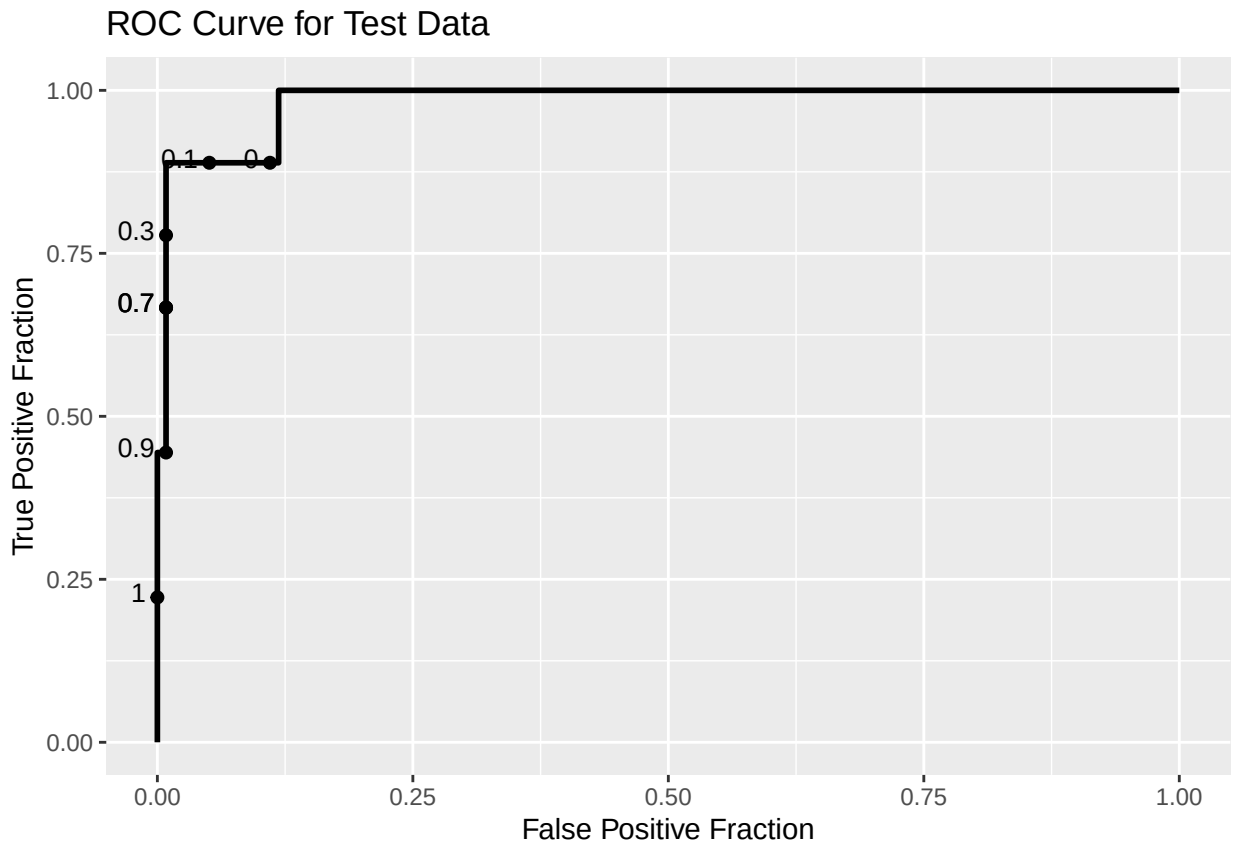
Based on the aforementioned accuracy metrics, I believe we should adjust the cutoff to lower than .5 given the fact that the rate of false positives in both the test and training sets are higher than the false negatives. However, we also need to consider the ramifications of this shift. Given the inverse tradeoff that exists between false positives and false negatives, we need to evaluate if having more false negatives than false positives and vice versa presents any significant implications. In this case, it does not necessarily determine anything life or death related, and only implies a rating of "pricey" or not. Thus we can adjust as needed to maximize the accuracy and reduce false positives and negatives as much as possible.

i) ROC Plots

```
train_ROC <- ggplot(housing_train,aes(m = priceyProb,
                    d = PriceyHome)) +
  geom_roc(labelsize = 3.5,
           cutoffs.at = c(.99,.9,.7,.6,.5,.4,.1,.01)) +
  labs(title = "ROC Curve for Train Data",x = "False Positive Fraction",
       y= "True Positive Fraction")
test_ROC <- ggplot(housing_test,aes(m = priceyProb,
                         d = PriceyHome)) +
  geom_roc(labelsize = 3.5,
           cutoffs.at = c(.99,.9,.7,.6,.5,.4,.1,.01)) +
  labs(title = "ROC Curve for Test Data",x = "False Positive Fraction",
       y= "True Positive Fraction")
train_ROC
```


ROC Curve for Train Data
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
0.6
0.9
1

```
test_ROC
```



j) AUC Calculations

```
train_AUC <- calc_auc(train_ROC)
test_AUC <- calc_auc(test_ROC)
print(train_AUC)
```

```
##   PANEL group       AUC
## 1     1    -1 0.9908327
```

```
print(test_AUC)
```

```
##   PANEL group       AUC
## 1     1    -1 0.9830508
```

Regarding the model, I would say that the model is slightly overfit due to the fact that the test error is slightly higher than the train data. This might be indicative of overfitting as the model has become too fit to the train data with the AUC value .01 away from encompassing the entire area under the curve. While the model's performance on the test data is only .02 away, we must observe that the model does not capture as much of the data in the test set as it does in training. Thus we need to adjust the probability cutoff point perhaps to give the model more flexibility when encountering new data in hopes of having more area under the curve for test data than train data.