HW2

Norman Hong

February 11, 2019

4.6

Suppose we collect data for a group of students in a statistics class with variables X_1 =hours studied, X_2 =undergrad GPA, and Y =received an A. We fit a logistic regression and produce estimated coefficient, $\hat{\beta}_0 = -6$, $\hat{\beta}_1 = .05$, $\hat{\beta}_2 = 1$.

(a)

Estimate the probability that a student who studies for 40 hours and has an undergraduate GPA of 3.5 gets an A in the class.

The following code returns a probability of .37, which is the probability that a student who studies for 40 hours and has a 3.5 GPA get an A in the class.

```
1/(1+exp(-(-6 + .05*40 + 1*3.5))) # probability
```

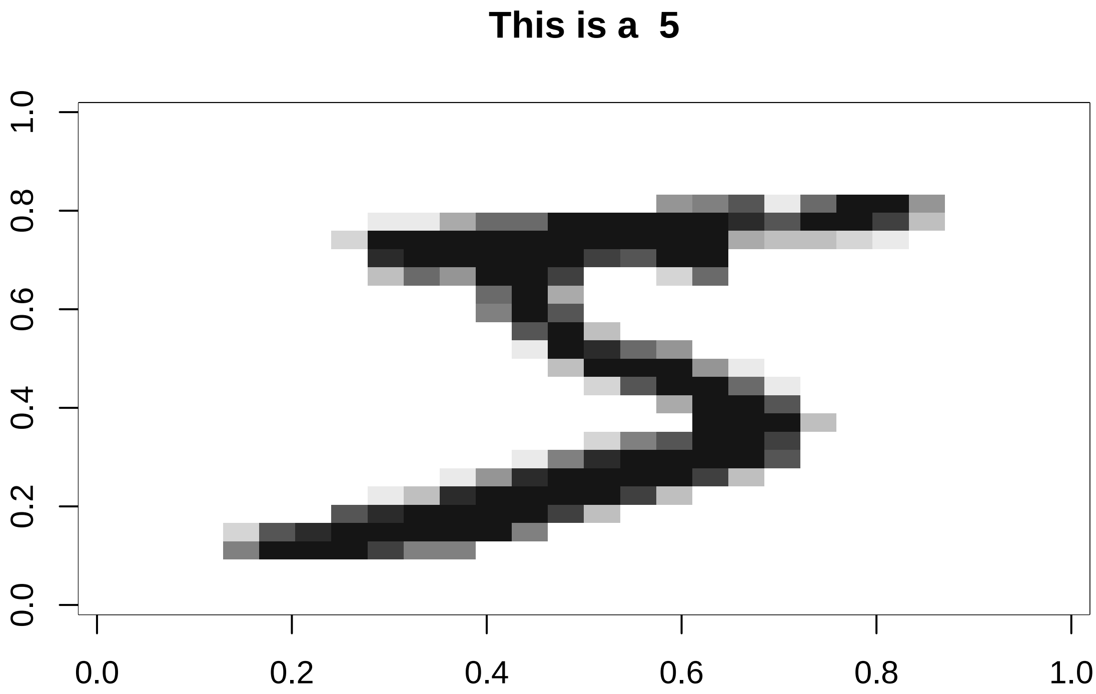
```
## [1] 0.3775407
```

(b)

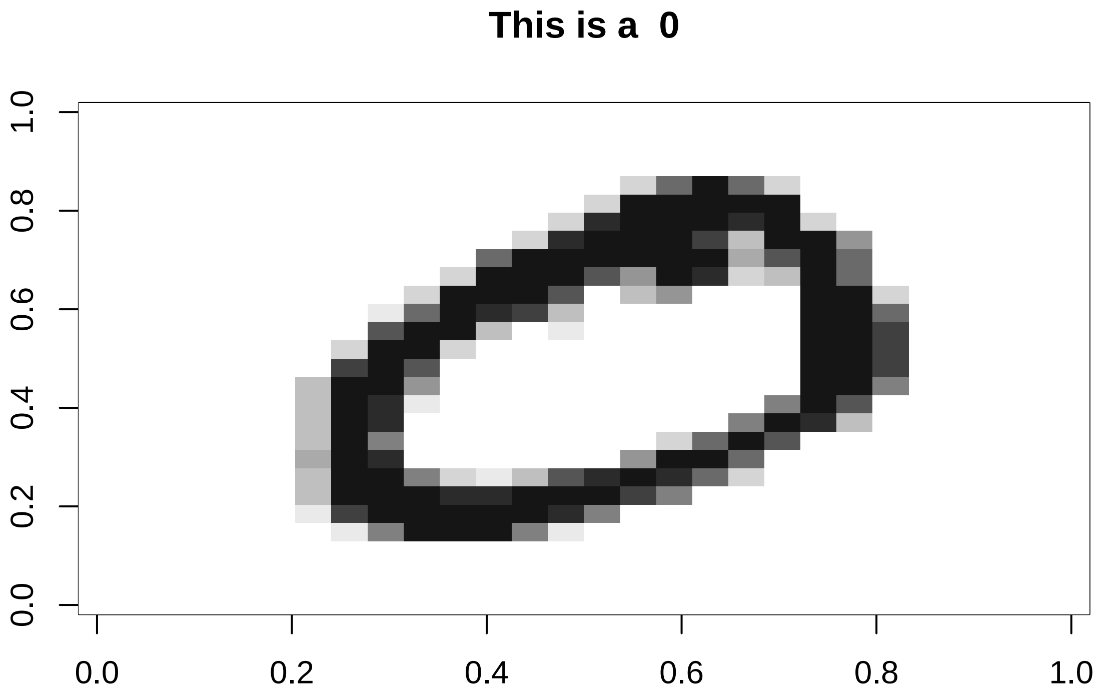
How many hours would the student in part (a) need to study to have a 50% chance of getting an A in the class? $.5 = 1/(1 + e^{-(-6+.05x+3.5)}) = .5 + .5e^{6-.05x-3.5} = 1 = e^{6-.05x-3.5} = 1 = lne^{6-.05x-3.5} = ln1$ $= 6 - .05x - 3.5 = 0 = x = 2.5/.05 = 50$

Exploring the mnist dataset.

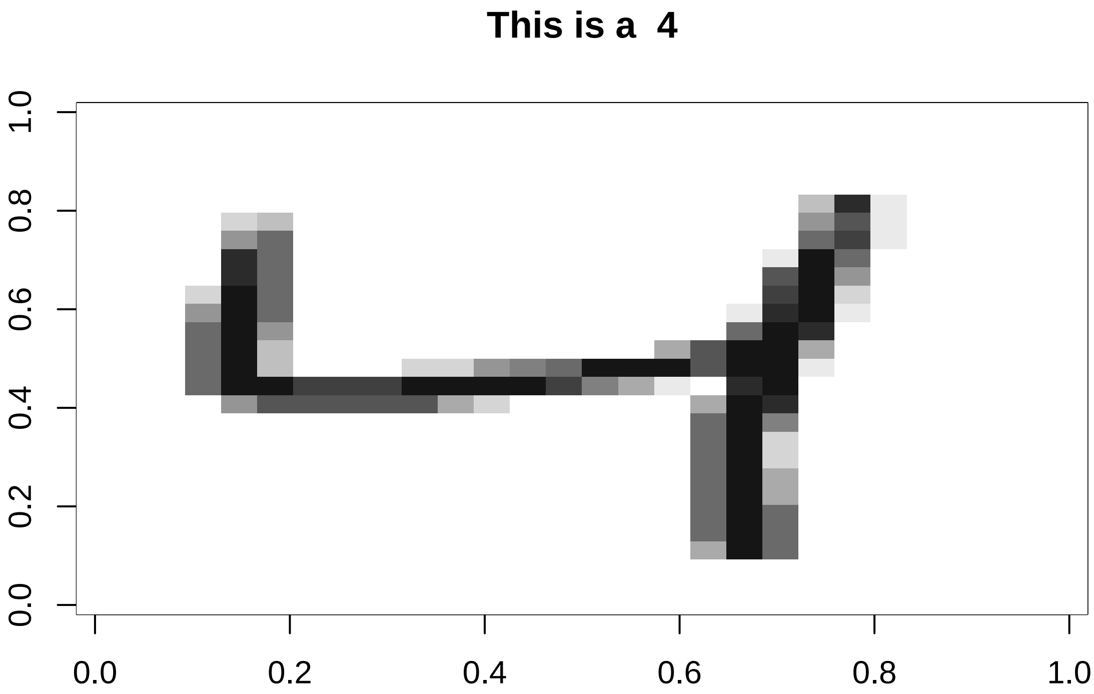
```
plot_digit <- function(j){
  arr784 <- as.numeric(train$x[j,])
  col=gray(12:1/12) # creating a vector from 12 ... 1 and divide by 12
  image(matrix(arr784, nrow=28)[,28:1], col=col,
        main=paste("This is a ", train$y[j]))
}
plot_digit(1)
```



```
plot_digit(2)
```



```r
plot_digit(3)
```



```r
# determine if all values in a column are 0's for training data in
# mnist data set.
det0 <- function(df){
  a <- c()
  for(i in 1:784){
    if(all(df$x[,i] == 0)){
      a <- c(a,i)
    }
  }
  cat(a)
}
det0(train)
```

```
## 1 2 3 4 5 6 7 8 9 10 11 12 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 53 54 55 56 57 58 83 84 85
```

```r
# Creating smaller data set that only contains class 3 and class 5.
trainX35 <- train$x[train$y == 3 | train$y == 5,]
trainY35 <- train$y[train$y == 3 | train$y == 5]
# number 5 is class 1; number 3 is class 0
trainY35 <- as.numeric(trainY35 == 5)
testX35 <- test$x[test$y == 3 | test$y == 5,]
testY35 <- test$y[test$y == 3 | test$y == 5]
testY35 <- as.numeric(testY35 == 5)
dataTrain <- data.frame(X=trainX35, Y=trainY35)
dataTest <- data.frame(X=testX35, Y=testY35)
```

```
# Determining predictor with highest variance
vars <- apply(trainX35, MARGIN=2, var)
sortedHighVar <- sort(vars, decreasing=TRUE, index.return=TRUE)
sortedHighVar$ix[1:30]
```

```
##  [1] 353 325 180 187 216 324 403 382 243 208 352 181 607 298 188 495 523
## [18] 522 215 155 154 550 439 271 347 244 597 156 381 375
```

```
# Determining predictor with lowest variance
sortedLowVar <- sort(vars, decreasing=FALSE, index.return=TRUE)
sortedLowVar$ix[1:100]
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
##  [18]  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
##  [35]  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51
##  [52]  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  81  82
##  [69]  83  84  85  86  87 112 113 140 141 142 169 197 225 253 281 309 337
##  [86] 364 365 366 367 392 419 420 448 449 476 477 504 505 532 533
```

```
# confusion matrix.  Used to approximate best threshold
confusion_mat <- function(df, threshold){
  table(df$Y, df$probabilities > threshold)
}
```

Extra 25

Build a classifier using only 1 variable (pixel). This variable should have large variation. Give the summary of the model and write out the logistic regression equation that has been obtained. Determine the fraction of true positives on the test set if the fraction of false positives on the training set is kept to .1.

$log(p(X)/(1-p(X))) = .8089 - .00972X_1$

```
# Fitting logistic model.
trainModel <- glm(Y~X.353,data=dataTrain, family=binomial)
summary(trainModel)
```

```
## 
## Call:
## glm(formula = Y ~ X.353, family = binomial, data = dataTrain)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.6617  -0.7264  -0.6847   0.7610   1.7737  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  1.0910605  0.0322571   33.82   <2e-16 ***
## X.353       -0.0095361  0.0001943  -49.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 15971  on 11551  degrees of freedom
## Residual deviance: 13141  on 11550  degrees of freedom
## AIC: 13145
```

```
##
## Number of Fisher Scoring iterations: 4
```

```
# the response is the probability an observation belongs to class 1 (number 5)
dataTrain$probabilities <- predict(trainModel, dataTrain, type="response")

rocTrain <- roc(dataTrain$Y, dataTrain$probabilities)
rocTrainDf <- data.frame(rocTrain$sensitivities, rocTrain$specificities, rocTrain$thresholds)
rocTrainDf[rocTrainDf$rocTrain.specificities > .87,]
```

```
##     rocTrain.sensitivities rocTrain.specificities rocTrain.thresholds
## 249              0.5724036              0.8703311           0.7348819
## 250              0.5688987              0.8726146           0.7367356
## 251              0.5663162              0.8744087           0.7385810
## 252              0.5652094              0.8755505           0.7404180
## 253              0.5617045              0.8775077           0.7422467
## 254              0.5600443              0.8789757           0.7440669
## 255              0.5570928              0.8804436           0.7458786
## 256              0.5556170              0.8815854           0.7476819
## 257              0.0000000              1.0000000                 Inf
```

The output shows that at threshold .74, the training FPR is approximately .12, which is closest I can get to .10. Recall that $FPR = 1 - Specificity$ and True positive rate (sensitivity) is $sensitivity = TP/(TP + FN)$. The sensitivity on the test data is .54.

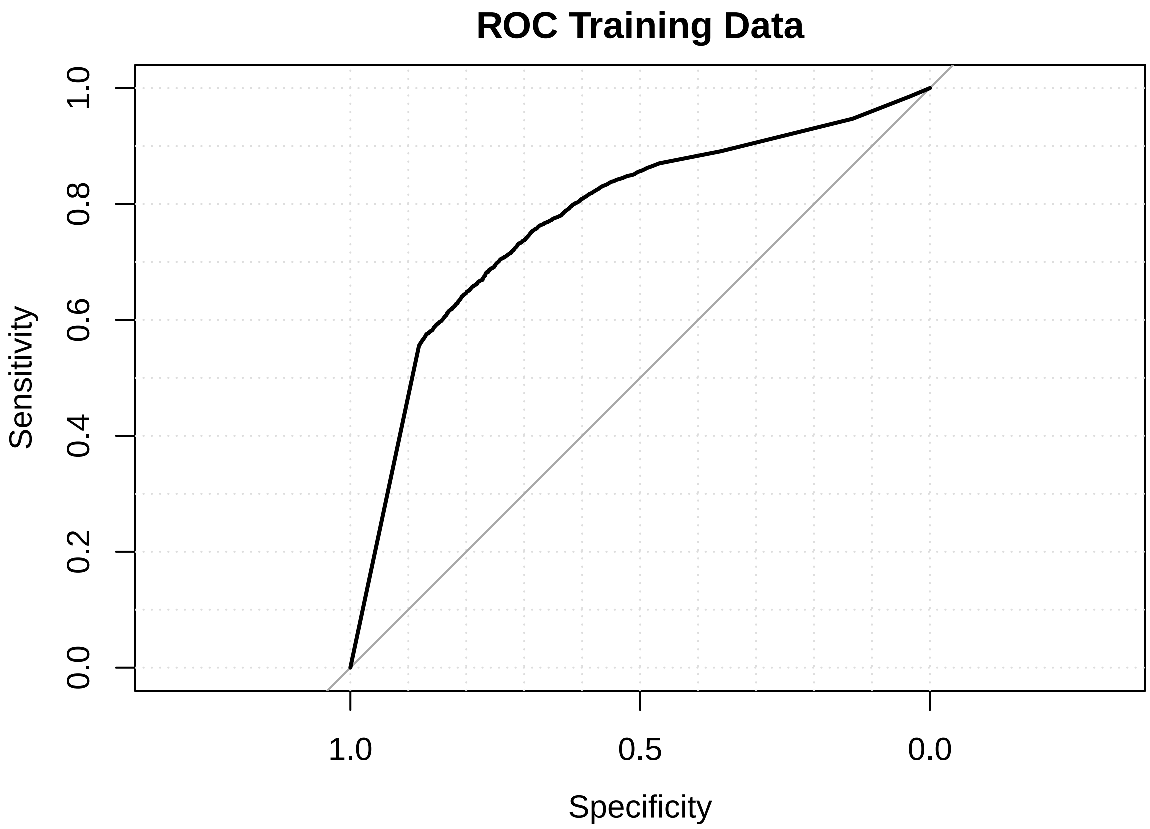
```
dataTest$probabilities <- predict(trainModel, dataTest, type="response")
table(dataTest$Y, dataTest$probabilities > .7476819)
```

```
##
##     FALSE TRUE
##   0   876  134
##   1   402  490
```

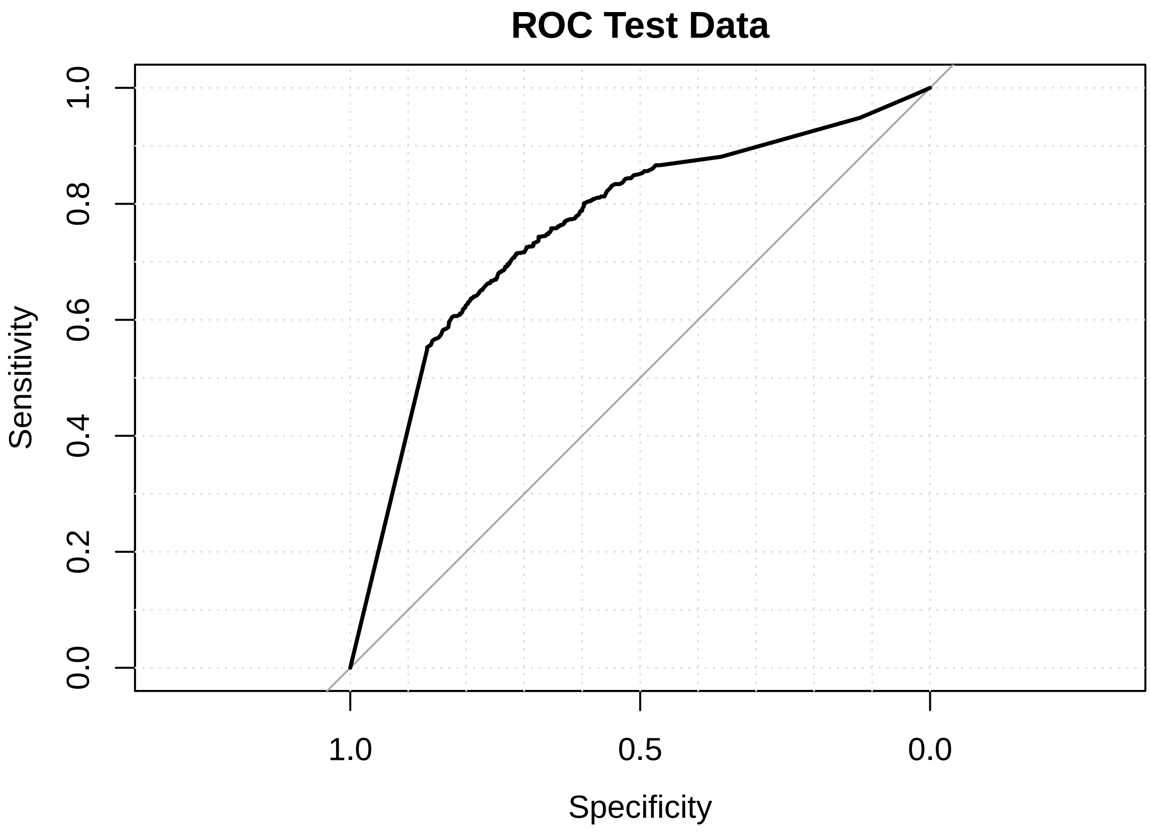
```
# Sensitivity
490/(402+490)
```

```
## [1] 0.5493274
```

```
plot(roc(dataTrain$Y, dataTrain$probabilities), grid=TRUE, main="ROC Training Data")
```



```
plot(roc(dataTest$Y, dataTest$probabilities), grid=TRUE, main="ROC Test Data")
```



```
cat("AUC for Training data:",auc(dataTrain$Y, dataTrain$probabilities))
```

```
## AUC for Training data: 0.7732375
```

```
cat("\nAUC for Test data:", auc(dataTest$Y, dataTest$probabilities))
```

```
##
## AUC for Test data: 0.7598016
```

Extra 26

(variables refer to predictor variables) Choose two variables that have small correlation and large variation. Find the area under the ROC curve (auc) using the training data and the test data. Make a scatter plot of the two variables, colored by the type of digit, and use this to explain the performance of the classifier.

```
# Determining predictor with highest variance
sortedHighVar$ix[1:10]
```

```
##  [1] 353 325 180 187 216 324 403 382 243 208
```

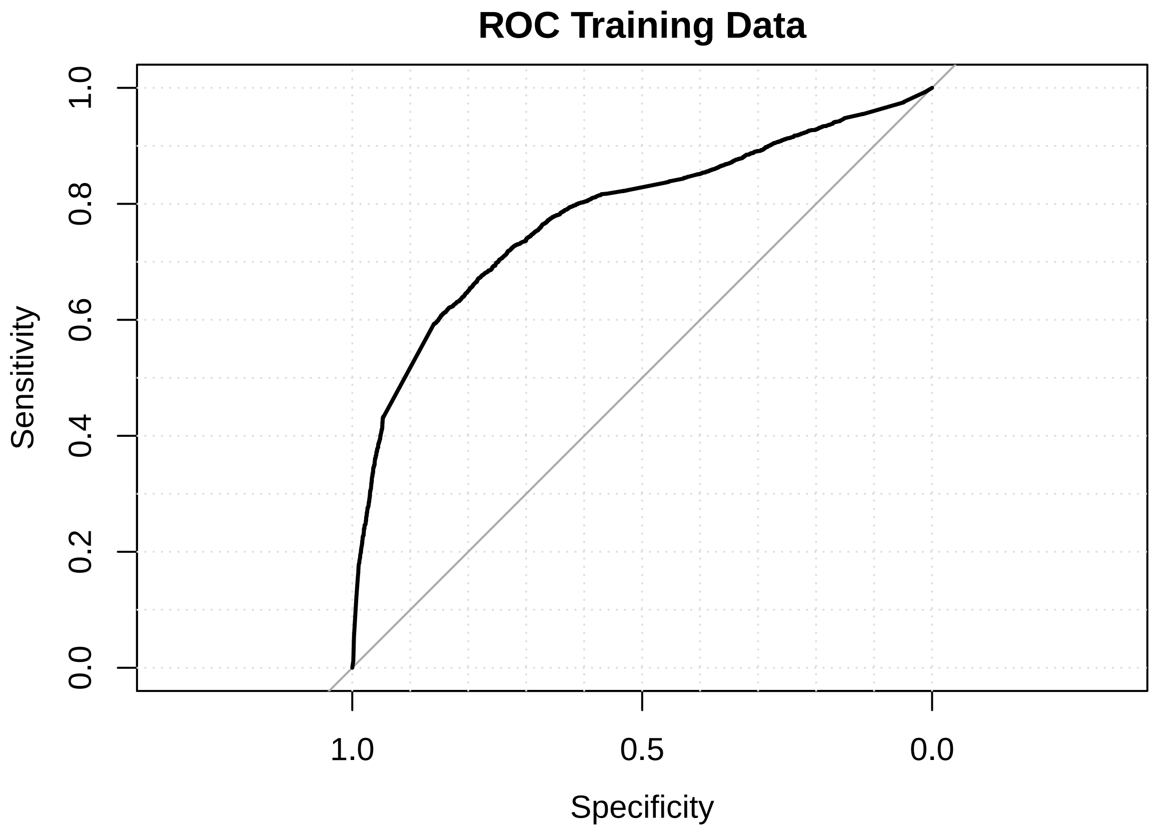
```
cor(trainX35[,sortedHighVar$ix[1:10]])
```

```
##              [,1]        [,2]        [,3]       [,4]        [,5]
## [1,] 1.00000000  0.67433063  0.32862190 0.17594525  0.04152201
## [2,] 0.67433063  1.00000000  0.28205480 0.06493588 -0.01854714
## [3,] 0.32862190  0.28205480  1.00000000 0.03480532 -0.14015201
## [4,] 0.17594525  0.06493588  0.03480532 1.00000000  0.60304113
## [5,] 0.04152201 -0.01854714 -0.14015201 0.60304113  1.00000000
##  [6,]  0.51090991  0.81430104  0.27394926 -0.03385620 -0.12601314
##  [7,] -0.12223485 -0.08737165 -0.03653595  0.03987426  0.10993468
##  [8,]  0.67416520  0.24572250  0.26222209  0.19816498  0.06357363
##  [9,]  0.08871213  0.17271574 -0.05812846  0.18202981  0.54325901
## [10,]  0.23869961  0.14611869  0.61107332  0.18034143  0.03832636
##              [,6]        [,7]         [,8]         [,9]        [,10]
##  [1,]  0.51090991 -0.12223485  0.674165203  0.088712128  0.238699613
##  [2,]  0.81430104 -0.08737165  0.245722501  0.172715735  0.146118687
##  [3,]  0.27394926 -0.03653595  0.262222087 -0.058128457  0.611073323
##  [4,] -0.03385620  0.03987426  0.198164982  0.182029811  0.180341434
##  [5,] -0.12601314  0.10993468  0.063573625  0.543259011  0.038326359
##  [6,]  1.00000000 -0.10640203  0.210813814  0.074773648  0.082316173
##  [7,] -0.10640203  1.00000000 -0.071963287  0.111242817  0.046907595
##  [8,]  0.21081381 -0.07196329  1.000000000  0.004836824  0.230278010
##  [9,]  0.07477365  0.11124282  0.004836824  1.000000000 -0.003640214
## [10,]  0.08231617  0.04690759  0.230278010 -0.003640214  1.000000000
```

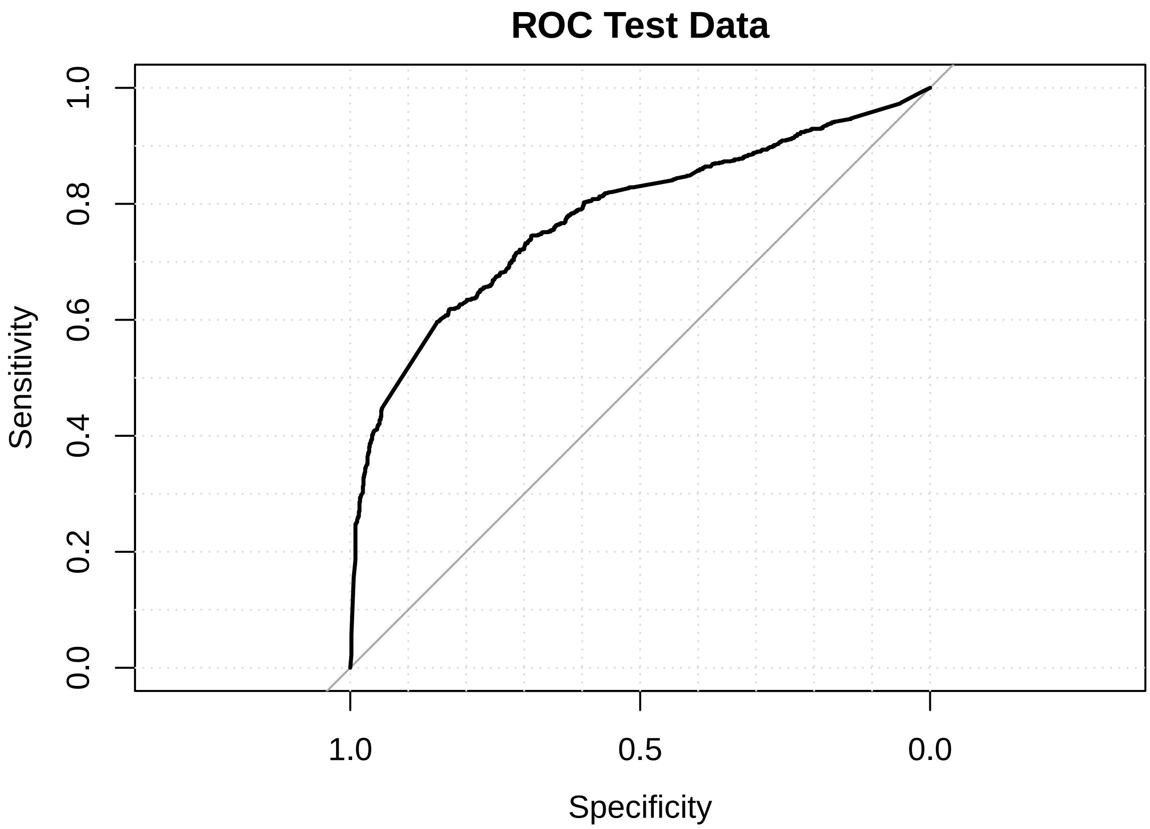
```
# 1st and 5th in the sorted list has low correlation.
# Corresponds to 353 and 216
```

```
glm.fit <- glm(Y~X.353+X.216, data=dataTrain, family=binomial)
dataTrain$probabilities <- predict(glm.fit, dataTrain, type = "response")
dataTrain$pred <- as.numeric(dataTrain$probabilities > .38)
dataTest$probabilities <- predict(glm.fit, dataTest, type = "response")
dataTest$pred <- as.numeric(dataTest$probabilities > .38)
```

```
# Roc function and auc function
# use roc function to determine corresponding TP when given FP.
plot(roc(dataTrain$Y, dataTrain$probabilities), grid=TRUE, main="ROC Training Data")
```



```r
plot(roc(dataTest$Y, dataTest$probabilities), grid=TRUE, main="ROC Test Data")
```



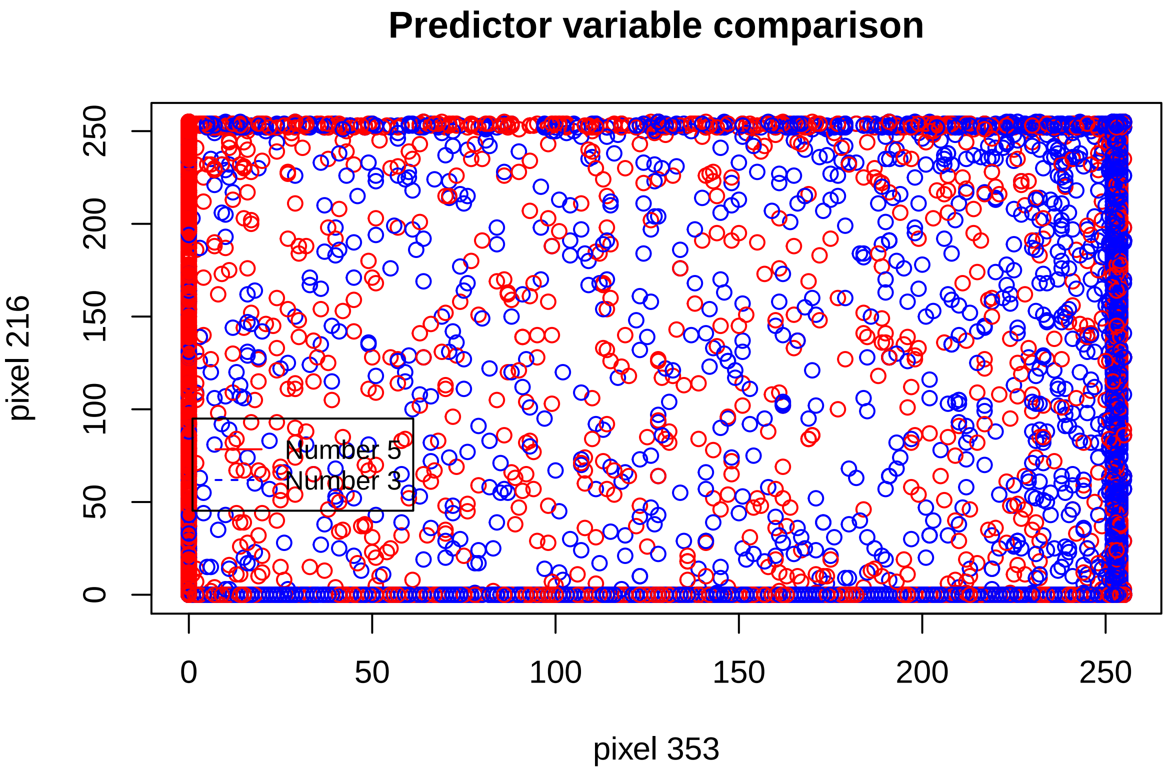
```
cat("AUC for Training data:",auc(dataTrain$Y, dataTrain$probabilities))
```

```
## AUC for Training data: 0.7794323
```

```
cat("\nAUC for Test data:", auc(dataTest$Y, dataTest$probabilities))
```

```
##
## AUC for Test data: 0.7773404
```

```
plot(x=dataTrain$X.353, y=dataTrain$X.216, xlab='pixel 353', ylab='pixel 216', main='Predictor variable
legend(1, 95, legend=c("Number 5", "Number 3"), col=c("red", "blue"), lty=1:2, cex=0.8)
```



The scatter plot of the two predictor variables show that there is no linearly separable line. No matter the decision boundary used, there is no way to perfectly separate the two classes.

4.10

This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the smarket data from this chapter's lab, except that it contains 1089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

(a)

Produce some numerical and graphical summaries of the Weekly data. Does there appear to be any patterns.

There is a non-linear correlation between volume and year. Also, the rest of the pair-wise combinations of variables appears to be uncorrelated. This was determined from the scatterplot matrix and the correlation matrix.

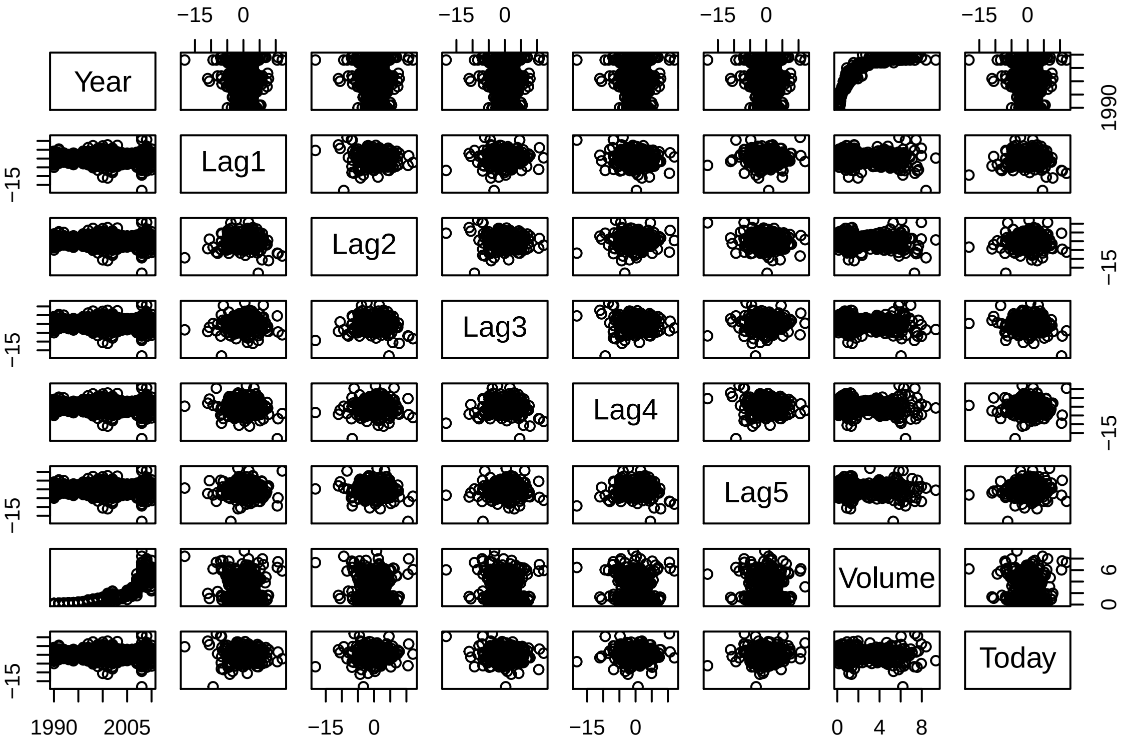
```r
# numerical summary
summary(Weekly)
```

```
##       Year           Lag1               Lag2               Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
##       Lag4               Lag5              Volume
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747  
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202  
##  Median :  0.2380   Median :  0.2340   Median :1.00268  
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462  
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373  
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821  
##      Today          Direction 
##  Min.   :-18.1950   Down:484  
##  1st Qu.: -1.1540   Up  :605  
##  Median :  0.2410             
##  Mean   :  0.1499             
##  3rd Qu.:  1.4050             
##  Max.   : 12.0260             
```

```r
cor(Weekly[-9])
```

```
##               Year         Lag1        Lag2        Lag3         Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
##                Lag5      Volume        Today
## Year   -0.030519101  0.84194162 -0.032459894
## Lag1   -0.008183096 -0.06495131 -0.075031842
## Lag2   -0.072499482 -0.08551314  0.059166717
## Lag3    0.060657175 -0.06928771 -0.071243639
## Lag4   -0.075675027 -0.06107462 -0.007825873
## Lag5    1.000000000 -0.05851741  0.011012698
## Volume -0.058517414  1.00000000 -0.033077783
## Today   0.011012698 -0.03307778  1.000000000
```

```r
# graphical EDA
pairs(Weekly[-9])
```



(b)

Use the full data set to perform a logistic regression with Direction as the response variable and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

The intercept is statistically significant at alpha level of .01. The coefficient for predictor variable Lag2 is statistically significant at alpha of .05.

```
weeklyCopy = Weekly
# 1 = Down and 0 = Up
weeklyCopy$Direction = as.numeric(weeklyCopy$Direction == 'Down')
regWeekly <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data=Weekly, family=binomial)
summary(regWeekly)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

(c)

Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

The accuracy is 43.8%. The confusion matrix shows that most of the errors are False positives. There is about 557 observations that were predicted to be in class down but were actually up.

```
confusion_mat <- function(df, threshold){
  myTab <- table(df$Direction , df$prob > threshold)
  return(myTab)
}
```

```
weeklyCopy$prob <- predict(regWeekly, weeklyCopy, type = "response")
table <- confusion_mat(weeklyCopy, .50)
table
```

```
##
##     FALSE TRUE
##   0    48  557
##   1    54  430
```

```
# Computing Accuracy (TP + TN)/sum(observations)
(table[1] + table[4])/sum(table)
```

```
## [1] 0.4389348
```

(d)

Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010). (Question saying to use training model to predict the test data, 2009 and 2010)

The accuracy is 0.375.

```
# 1 = Down and 0 = Up
# Fitting logistic regression on 1990 to 2008 data.
weeklyEx910 <- weeklyCopy[weeklyCopy$Year <= 2008,]
regEx910 <- glm(Direction ~ Lag2,data=weeklyEx910, family=binomial)
summary(regEx910)
```

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = binomial, data = weeklyEx910)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.368  -1.091  -1.021   1.264   1.536
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.20326    0.06428  -3.162  0.00157 **
## Lag2        -0.05810    0.02870  -2.024  0.04298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
## AIC: 1354.5
##
## Number of Fisher Scoring iterations: 4
```

```
# Compute confusion matrix and accuracy rate for 2009 and 2010.
weekly910 <- Weekly[Weekly$Year > 2008,]
weekly910$prob <- predict(regEx910, weekly910, type = "response")
table <- confusion_mat(weekly910, .50)
table
```

```
##
##        FALSE TRUE
##   Down    34    9
##   Up      56    5
```

```
(table[1] + table[4])/sum(table)
```

```
## [1] 0.375
```

Extra 23

Replace the factor variable Purchase with a new numerical variable purchase01 which equals 1 if a customer bought Minute Maid orange juice (MM) and equals 0 if she bought Citrus Hill orange juice (CH)

(a)

Fit a logistic model to predict purchase01 from all predictors. Call this model fit.22a. There are several predictors for which a coefficient estimate is not available. Give a reason for each such predictor why this happens. Look for simple arithmetic relations between some of the predictors.

This happens when the variable can be expressed as a transformation of another variable. This causes perfect collinearity, which inflates the variance of the coefficients and causes the coefficients to be indeterminant. For instance, STORE is equivalent to STOREID, but with the the category 0 turned into 7. The variables SalePriceMM and SalePriceCH are a transformed version of PriceCH and PriceMM.

```
OJ$purchase01 = as.numeric(OJ$Purchase == 'MM')
fit.22a <- glm(purchase01 ~ . - Purchase, data=OJ, family=binomial)
```

```
summary(fit.22a)
```

```
##
## Call:
## glm(formula = purchase01 ~ . - Purchase, family = binomial, data = OJ)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7811  -0.5426  -0.2327   0.5304   2.7894
##
## Coefficients: (5 not defined because of singularities)
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      5.15806    2.02648   2.545  0.01092 *
## WeekofPurchase  -0.01181    0.01080  -1.093  0.27442
## StoreID         -0.17089    0.13847  -1.234  0.21716
## PriceCH          4.58650    1.81386   2.529  0.01145 *
## PriceMM         -3.62495    0.90259  -4.016 5.92e-05 ***
## DiscCH          10.79673   18.60661   0.580  0.56174
## DiscMM          26.46155    9.08497   2.913  0.00358 **
## SpecialCH        0.26723    0.34207   0.781  0.43468
## SpecialMM        0.31693    0.27307   1.161  0.24579
## LoyalCH         -6.30227    0.39834 -15.821  < 2e-16 ***
## SalePriceMM           NA         NA      NA       NA
## SalePriceCH           NA         NA      NA       NA
## PriceDiff             NA         NA      NA       NA
## Store7Yes        0.31128    0.71681   0.434  0.66411
## PctDiscMM      -50.69763   19.01208  -2.667  0.00766 **
## PctDiscCH      -27.33993   35.17272  -0.777  0.43698
## ListPriceDiff         NA         NA      NA       NA
## STORE                 NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1430.9  on 1069  degrees of freedom
## Residual deviance:  816.6  on 1057  degrees of freedom
## AIC: 842.6
##
## Number of Fisher Scoring iterations: 5
```

(b)

Remove all predictors for which a coefficient estimate is not available and fit a new model. Call this model fit.22b. What are the differences between fit.22a and fit.22b, if any?

No coefficient estimate, standard error or t value changed.

```
fit.22b <- glm(purchase01 ~ . - Purchase - ListPriceDiff - STORE - SalePriceMM - SalePriceCH - PriceDif
summary(fit.22b)
```

```
##
## Call:
## glm(formula = purchase01 ~ . - Purchase - ListPriceDiff - STORE -
##     SalePriceMM - SalePriceCH - PriceDiff, family = binomial,
##     data = OJ)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7811  -0.5426  -0.2327   0.5304   2.7894
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      5.15806    2.02648   2.545  0.01092 *
## WeekofPurchase  -0.01181    0.01080  -1.093  0.27442
## StoreID         -0.17089    0.13847  -1.234  0.21716
## PriceCH          4.58650    1.81386   2.529  0.01145 *
## PriceMM         -3.62495    0.90259  -4.016 5.92e-05 ***
## DiscCH          10.79673   18.60661   0.580  0.56174
## DiscMM          26.46155    9.08497   2.913  0.00358 **
## SpecialCH        0.26723    0.34207   0.781  0.43468
## SpecialMM        0.31693    0.27307   1.161  0.24579
## LoyalCH         -6.30227    0.39834 -15.821  < 2e-16 ***
## Store7Yes        0.31128    0.71681   0.434  0.66411
## PctDiscMM      -50.69763   19.01208  -2.667  0.00766 **
## PctDiscCH      -27.33993   35.17272  -0.777  0.43698
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1430.9  on 1069  degrees of freedom
## Residual deviance:  816.6  on 1057  degrees of freedom
## AIC: 842.6
##
## Number of Fisher Scoring iterations: 5
```

(c)

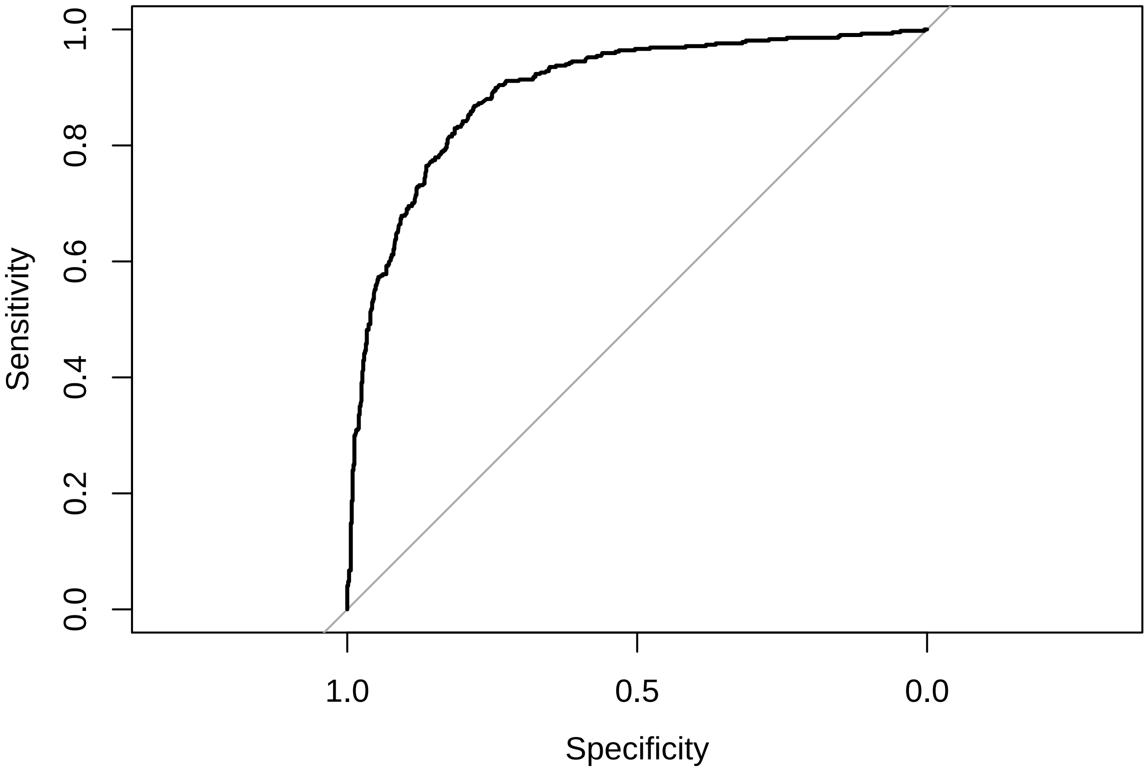
Which predictors are significant for fit.22b? Make a new model with only those predictors and call it fit.22c. Plot the ROC curve and show that the area under the curve is approximately .89.

PriceCH, DiscMM, and PctDiscMM are statistically significant at alpha level of .01. PriceMM and LoyalCH are statistically significant at alpha of .001.

```r
fit.22c <- glm(purchase01 ~ PriceCH + DiscMM + PctDiscMM + PriceMM + LoyalCH, data=OJ, family=binomial)
summary(fit.22c)
```

```
##
## Call:
## glm(formula = purchase01 ~ PriceCH + DiscMM + PctDiscMM + PriceMM +
##     LoyalCH, family = binomial, data = OJ)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6394  -0.5800  -0.2564   0.5634   2.8592
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   5.5773     1.7465   3.194  0.00141 **
## PriceCH       2.7315     1.1199   2.439  0.01472 *
## DiscMM       25.1929     8.3831   3.005  0.00265 **
## PctDiscMM   -47.9810    17.5153  -2.739  0.00616 **
## PriceMM      -3.8818     0.8313  -4.669 3.02e-06 ***
## LoyalCH      -6.3725     0.3814 -16.706  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1430.85  on 1069  degrees of freedom
## Residual deviance:  855.19  on 1064  degrees of freedom
## AIC: 867.19
##
## Number of Fisher Scoring iterations: 5
```

```
OJ$pred <- predict(fit.22c, OJ, type='response')
plot(roc(OJ$purchase01, OJ$pred))
```



```
cat("Area under ROC Training data:", auc(OJ$purchase01, OJ$pred))
```

```
## Area under ROC Training data: 0.8940841
```

(d)

Consider now the predicted odds that a customer purchased Minute Maid. How do these odds change if the price of minute maid is decreased by .01? How do thse odds change if the price of Citrus Hill is increased by .01? How do thse odds change if the discount offered for minute maid is increased by .01? Note that this is essentially the same as dropping the price for minute maid, but the predicted effect on the odds is very different.

$log(p(X)/(1-p(X))) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 ...$ According to the logistic regression model from part c, if the price of minute maid (PriceMM) decreased by .01, the log odds has a correlated increase of .00388. If the price of Citrus Hill (PriceCH) is increased by .01, the log odds has a correlated increase of .0273. If the discount offered for minute maid is increased by .01, the log odds has a correlated increase of .251, which is very different from when the price of minute maid decreased by .01.

Extra 27

Build a classifier that uses the 10 variables with the largest variances. Make ROC curves for training and test data and comment on the performance of the classifier. Is this a good way to select 10 predictors for classification? Can you think of other ways of selecting 10 predictors for classification?

Another way to select 10 best predictors is to select variables with lowest correlations or use lasso.

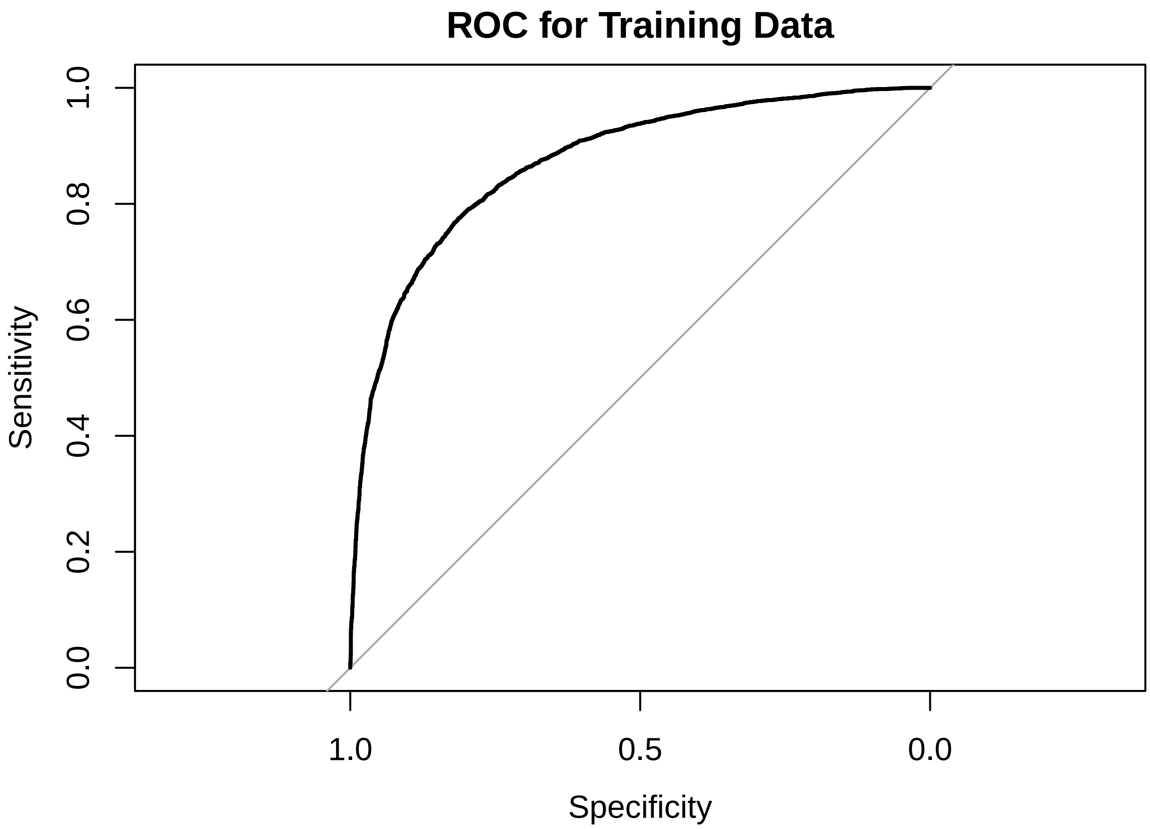
```r
# Determine highest variance variables.
sortedHighVar$ix[1:10]
```

```
##  [1] 353 325 180 187 216 324 403 382 243 208
```

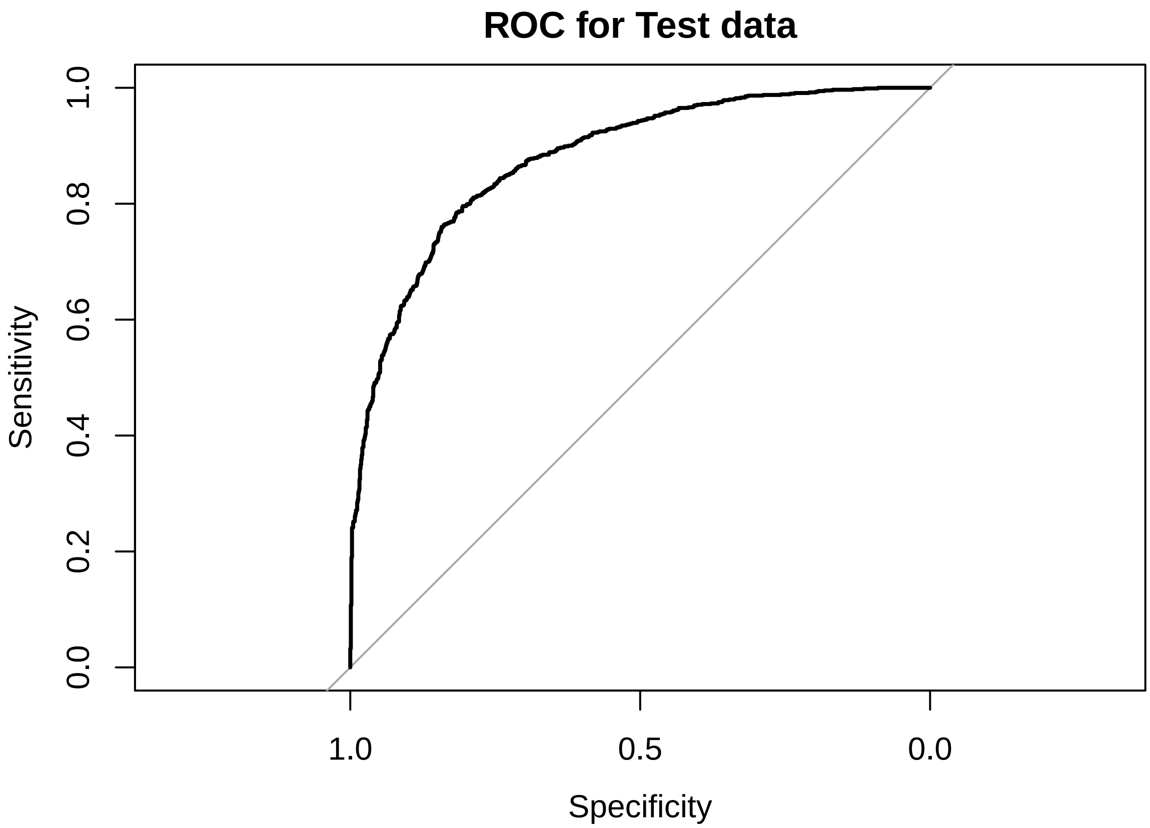
```r
regTrain <- glm(Y~X.353+X.325+X.180+X.187+X.216+X.324+X.403+X.382+X.243+X.208, data=dataTrain, family=bi

dataTrain$probabilities <- predict(regTrain, dataTrain, type='response')
dataTrain$pred <- as.numeric(dataTrain$probabilities > .5)
dataTest$probabilities <- predict(regTrain, dataTest, type='response')
dataTest$pred <- as.numeric(dataTest$probabilities > .5)

plot(roc(dataTrain$Y, dataTrain$probabilities), main='ROC for Training Data')
```



```r
plot(roc(dataTest$Y, dataTest$probabilities), main='ROC for Test data')
```



```
cat("AUC for Training data:",auc(dataTrain$Y, dataTrain$probabilities))
```

```
## AUC for Training data: 0.8725303
```

```
cat("\nAUC for Test data:", auc(dataTest$Y, dataTest$probabilities))
```

```
##
## AUC for Test data: 0.8781335
```