question2

Michael Jones

06/02/2022

INSERT BELOW INTO MAIN—

a) Renaming Columns

```
mnist_test.df[,1] <- ifelse(mnist_test.df[,1] == 7, 1, 0)
names(mnist_test.df)[1] <- 'Y'

mnist_train.df[,1] <- ifelse(mnist_train.df[,1] == 7, 1, 0)
names(mnist_train.df)[1] <- 'Y'
```

b) Model Constructing

```
index.predictors = seq(15 * 28 + 1, length.out = 28, by = 1)

predictors.ch = names(mnist_train.df)[index.predictors]

#The code given fails, so I just did my own based on the columns given
mnist28.glm = glm(Y ~ `0...421` + `0...422` + `0...423` + `0...424` +
                    `0...425` + `0...426` + `0...427` + `0...428` +
                    `0...429` + `0...430`+ `0...431` + `0...432` +
                    `0...433` + `0...434` + `0...435` + `45` +
                    `186` + `253...438` + `253...439` + `150` +
                     `27` + `0...442` + `0...443` + `0...444` +
                    `0...445` + `0...446` + `0...447` + `0...448`,
                  family = binomial, data = mnist_train.df)


actualY <- mnist_train.df$Y
predictY0.5 <- ifelse(fitted.values(mnist28.glm) <= 0.5, 0, 1)

#Output Confusion Matrix
confusMatrix <- table(Actual = actualY, Predicted = predictY0.5)
confusMatrix
```

```
##        Predicted
## Actual     0     1
##      0 52474  1260
##      1  4334  1931
```

c) Esimated Prediction Error

Given $Pred.Error = \frac{FP+FN}{TP+TN+FP+FN}$

```
(confusMatrix[1,2] + confusMatrix[2,1])/sum(confusMatrix)
```

```
## [1] 0.09323489
```

Our estimated prediction error is approximately 0.093.

d) Estimated Sensitivity

Given $Sensitivity = \frac{TP}{TP+FN}$

```
confusMatrix[2,2]/sum(confusMatrix[2,])
```

```
## [1] 0.3082203
```

Our estimated sensitivity is approximately 0.308.

e) Estimated Specificity

Given $Specificity = \frac{TN}{TN+FP}$

```
confusMatrix[1,1]/sum(confusMatrix[1,])
```

```
## [1] 0.9765512
```

Our estimated specificity is approximately 0.977.

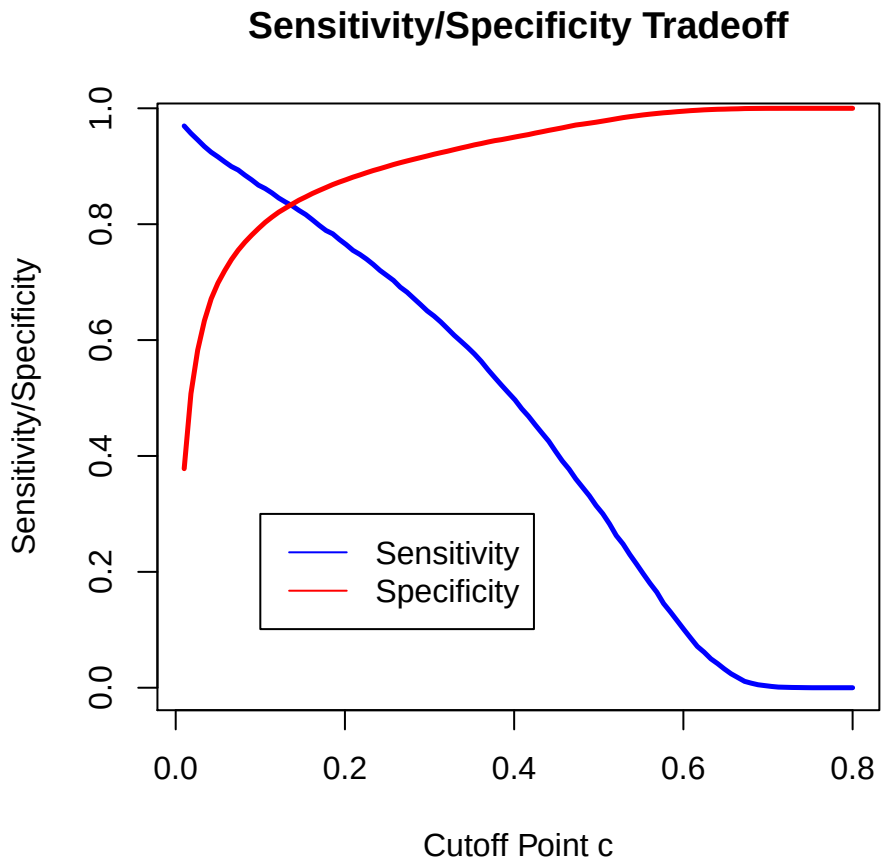
f) Specificity/Sensitivity Plot

```
n.plot = 100
c.vec = seq(0.01, 0.8, length.out = n.plot)

sensitivityData <- numeric(n.plot)
specificityData <- numeric(n.plot)
for(i in 1:100) {
  predictY <- ifelse(fitted.values(mnist28.glm) <= c.vec[i], 0, 1)
  confusMatrix2 <- table(Actual = actualY, Predicted = predictY)
  sensitivityData[i] <- confusMatrix2[2,2]/sum(confusMatrix2[2,])
  specificityData[i] <- confusMatrix2[1,1]/sum(confusMatrix2[1,])

}
```

```
plot(c.vec, sensitivityData, type = 'l', lwd=2.5, col = 'blue',
     main = 'Sensitivity/Specificity Tradeoff',
     xlab = 'Cutoff Point c',
     ylab = 'Sensitivity/Specificity')
lines(c.vec, specificityData, type = 'l', col = 'red', lwd=2.5)
legend(0.1,0.3, legend=c("Sensitivity", "Specificity"),
       col=c("blue", "red"), lty=c(1,1))
```



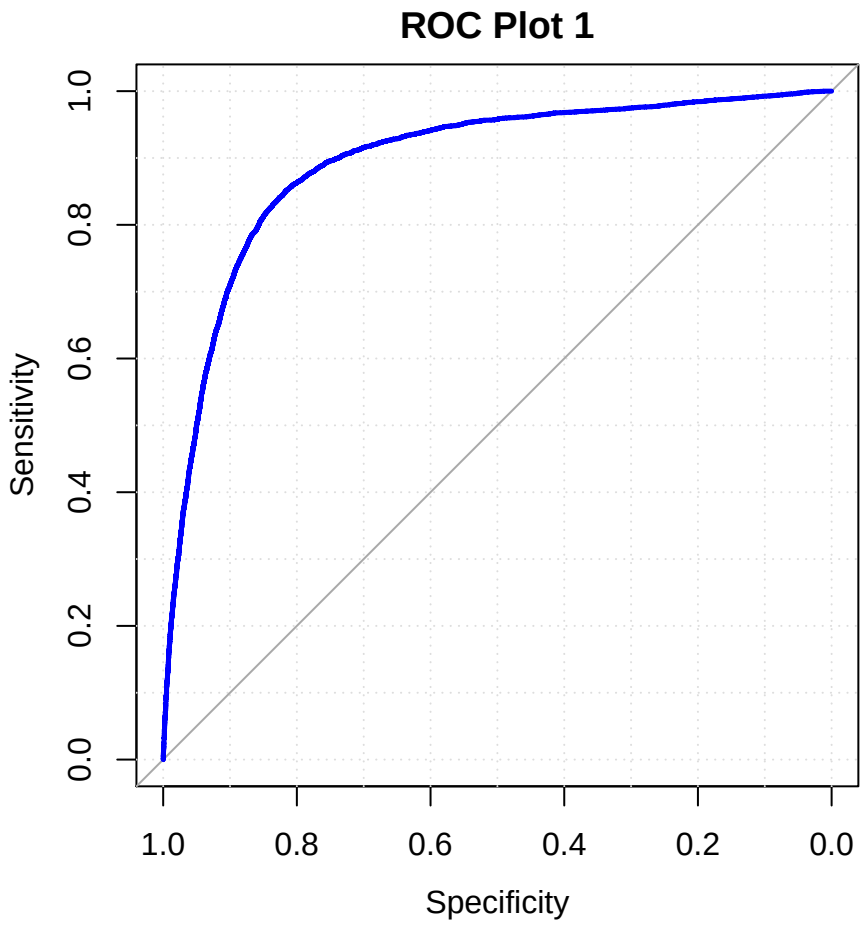
g) ROC Plots

```
rocPlot <- roc(response = mnist_train.df$Y,
               predictor = fitted.values(mnist28.glm))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(rocPlot, col = "blue", grid = TRUE, lwd=2.5, main = "ROC Plot 1")
```

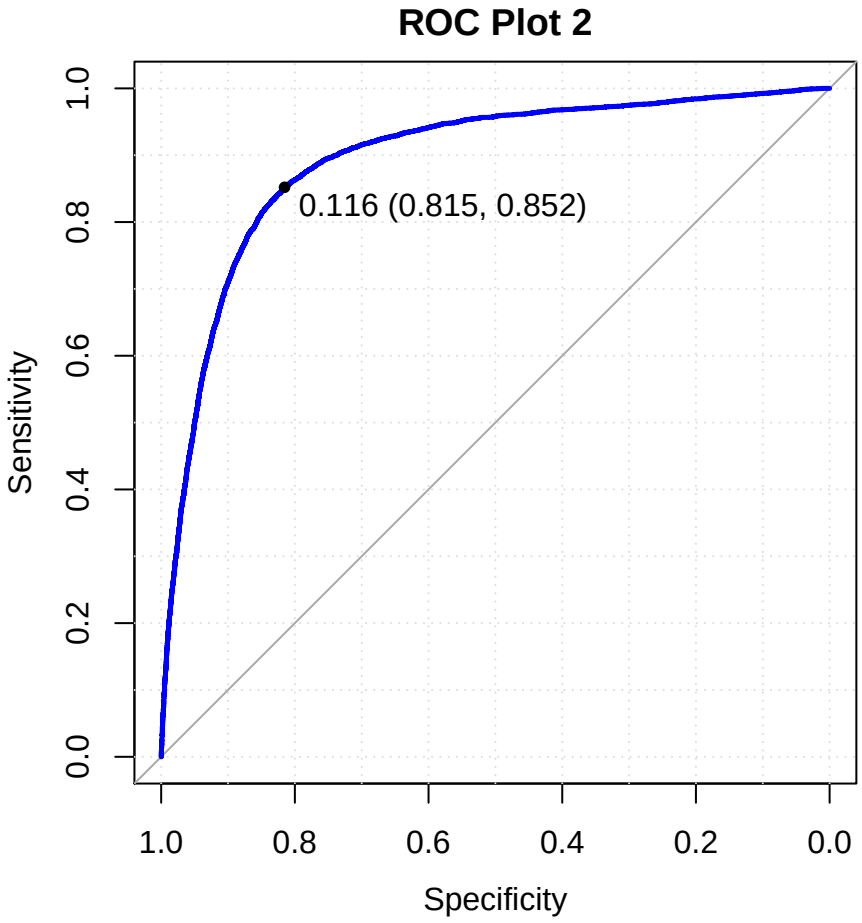


h) Area under ROC Curve

The area under the ROC curve is called AUC (Area Under Curve) and its area pesents our models predictive ability.

i) Max Specificity/Sensitivty

```
plot(rocPlot, col = "blue", grid = TRUE, lwd=2.5, main = "ROC Plot 2",
     print.thres = "best")
```

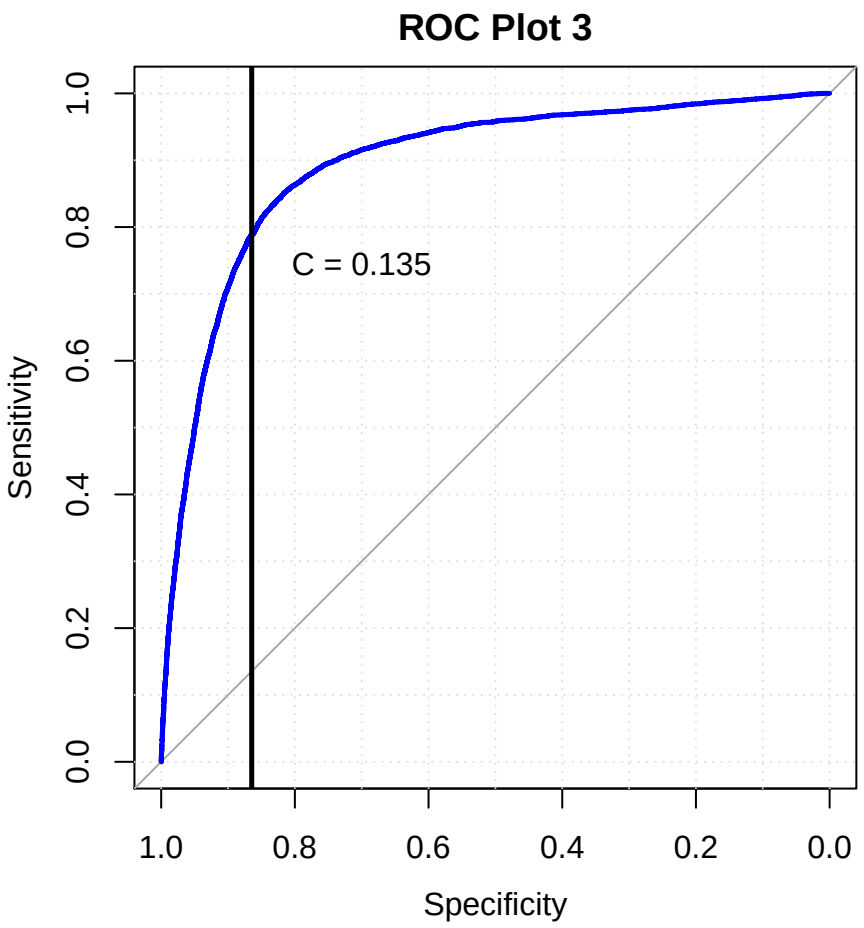


The above plot shows are optimal (given we wish to maximize the sum of sensitivity and specificity) is 0.116.

j) Maximizing the Minimum of Sensitivty and Specificity

```
ind2 <- with(rocPlot, which.min(abs(sensitivities - specificities)))

plot(rocPlot, col = "blue", grid = TRUE, lwd=2.5, main = "ROC Plot 3")
abline(v = 1-rocPlot$thresholds[ind2], col = 'black', lwd = 2.5)
text(0.7,0.7,"C = 0.135",srt=0.2,pos=3)
```


ROC Plot 3
C = 0.135
Sensitivity
Specificity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.8
0.6
0.4
0.2
0.0