STAT462 Assignment 2: Classification

Graham Greig SN: 47022356

09/09/2021

Problem 1

Suppose we collect data for a group of students with variables $X1$ = hours spent studying per week, $X2$ = number of classes attended and Y = (1 if the student received a GPA value of 7 or better in in the class and 0 otherwise.) We fit a logistic regression model and find the estimated coefficients to be: $\hat{\beta}_0 = -16$; $\hat{\beta}_1 = 1.4$ and $\hat{\beta}_2 = 0.3$.

a.) Estimate the probability that a student gets a GPA value $\geqslant$ 7 if they study 5 or more hours per week and they attend all 36 classes.

The model for the logistic regression function for this classification problem is:

$$P(Y \geqslant 7|X1 = 5, X2 = 36) = p(x) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 5+\hat{\beta}_2 36}}{1 + e^{\hat{\beta}_0+\hat{\beta}_1 5+\hat{\beta}_2 36}}$$

```
#Start by defining the coefficients
beta_0 = -16
beta_1 = 1.4
beta_2 = 0.3

x1 = 5
x2 = 36

#Now evaluate the function
prob = exp(beta_0 + beta_1 * x1 + beta_2 * x2) / (1 + exp(beta_0 + beta_1 * x1 + beta_2 * x2))
```

The probability if found to be: **85.81%**.

b.) If a student attends 18 classes how many hours need to be studied to achieve a GPA of greater than or equal to 7 with a probability of 50%

This is best solved using the logit transform of the logistic regression function

$$log(\frac{p(x)}{1 - p(x)}) = \hat{\beta}_0 + \hat{\beta}_1 X1 + \hat{\beta}_2 X2$$

Plugging in $p(x) = 0.5$ and $X2 = 5$ and all $\hat{\beta}$ values gives:

$$log(1) = \hat{\beta}_0 + \hat{\beta}_1 X1 + \hat{\beta}_2 * 18$$

And so,

```r
X1 = (-beta_0 - beta_2*18)/beta_1
```

This finds that the student would need to study **7.57** hours per week to have a 50% chance of achieving a GPA of 7 or greater.

Problem 2

In this problem a logistic model will be fit to predict the probability that a banknote was forged using the banknote data set. This data has been divided into training and testing sets. (BankTrain.csv and BankTest.csv) The 5th column is the response variable where y = 1 indicates a forgery and y = 0 is a genuine note. Only X1 and X3 will be used as predictors.

a.) Perform Multiple Logistic Regression on the training data and comment on the model.

The model to fit will once again be $p(x) = \frac{e^{\hat{\beta_0}+\hat{\beta_1}X1+\hat{\beta_3}X3}}{1+e^{\hat{\beta_0}+\hat{\beta_1}X1+\hat{\beta_3}X3}}$

```r
# Load in the data
BankTrain=read.csv("BankTrain.csv",header=T,na.strings="?")
BankTest=read.csv("BankTest.csv",header=T,na.strings="?")

#Get the model
training_glm=glm(y~x1+x3, data=BankTrain, family=binomial)
summary(training_glm)
```

```
##
## Call:
## glm(formula = y ~ x1 + x3, family = binomial, data = BankTrain)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.83187  -0.28343  -0.06417   0.50032   1.99366
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.22041    0.11206   1.967   0.0492 *
## x1          -1.31489    0.08822 -14.905  < 2e-16 ***
## x3          -0.21738    0.02880  -7.548 4.42e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1322.01  on 959  degrees of freedom
## Residual deviance:  572.07  on 957  degrees of freedom
## AIC: 578.07
##
## Number of Fisher Scoring iterations: 6
```

From the summary it is clear that the P values for x1 and x3 are significant indicating that they should be considered in the model from a maximum likelihood estimation of Bernoulli trials. The intercept however is not very significant. From the p-values in the summary (Pr(>|Z|)) this is likely a good model. The beta values are found to be:

$$\hat{\beta}_0 = 0.22, \hat{\beta}_1 = -1.31, \hat{\beta}_3 = -0.22$$

b.) i.)Plot the training data and the decision boundary assuming a decision boundary of p(x) = 0.5

For this, the logit form of the GLM classiification is most useful. We get:

$$log(\frac{p(x)}{1-p(x)}) = \hat{\beta}_0 + \hat{\beta}_1 X1 + \hat{\beta}_3 X3$$

Using $p(x) = 0.5$ gives:

$$X1 = -\frac{\hat{\beta}_3}{\hat{\beta}_1}X3 - \frac{\hat{\beta}_0}{\hat{\beta}_1}$$

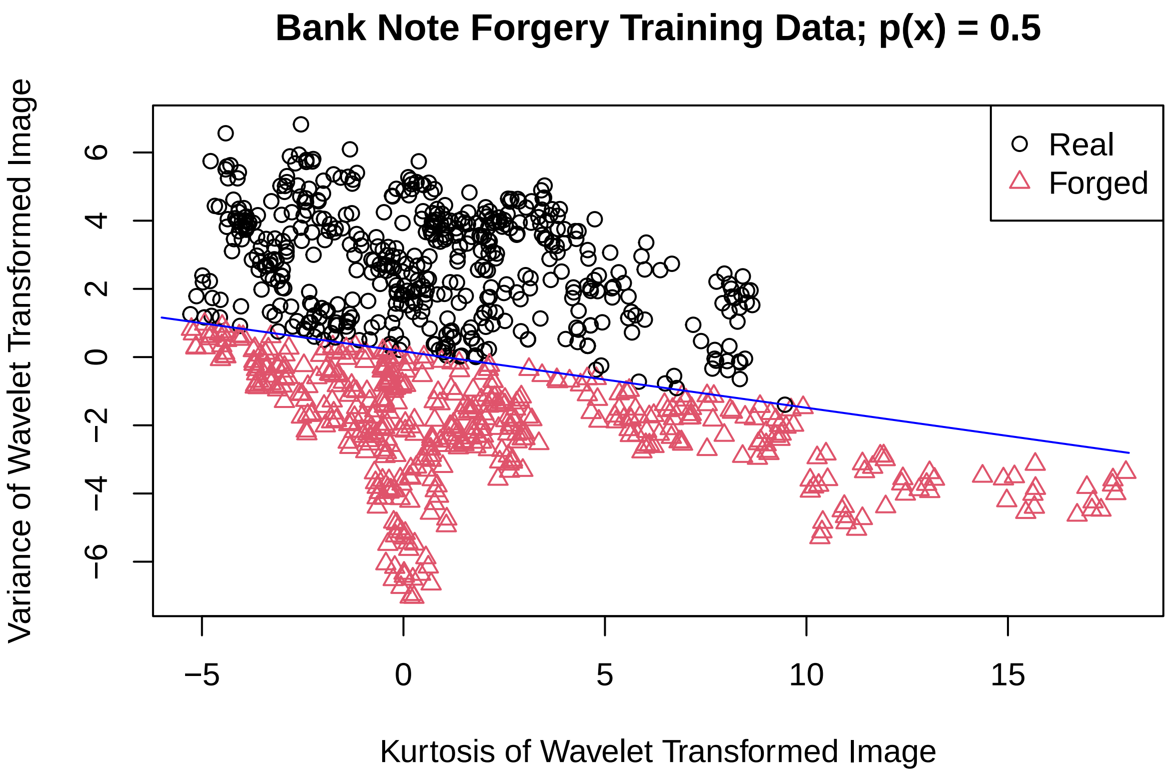
This is a model which can have its decision boundary plotted with slope $-\frac{\hat{\beta}_3}{\hat{\beta}_1}$ and intercept $-\frac{\hat{\beta}_0}{\hat{\beta}_1}$.

```
glm_probs = predict(training_glm, type="response")
glm_pred=rep("Real Banknote",960)
glm_pred[glm_probs>.5]="Forged Banknote"
glm_pred = factor(glm_pred, levels = c("Real Banknote", "Forged Banknote"))

plot(BankTrain$x3, BankTrain$x1, pch = as.integer(glm_pred), col = as.integer(glm_pred), main = "Bank No
legend("topright",legend = c("Real", "Forged"), pch = c(1,2),col = c(1,2),text.col = "black",horiz = FAL

x = seq(from=-6,to=18,0.01)
beta = summary(training_glm)$coef[,1]
y = -beta[3]/beta[2]*x -beta[1]/beta[2]

lines(x,y,col="blue")
```



ii) Compute the confusion matrix for the Testing data set and comment on the output

The confusion matrix is computed as follows:

```
test_probs = predict(training_glm,BankTest,type = "response")
test_glm_pred=rep("Real Banknote",412)
test_glm_pred[test_probs>.5]="Forged Banknote"
test_glm_pred = factor(test_glm_pred, levels = c("Real Banknote", "Forged Banknote"))
table(test_glm_pred,BankTest$y)
```

```
##
## test_glm_pred       0   1
##   Real Banknote   204  24
##   Forged Banknote  32 152
```

The accuracy of this model is found by:

```
(204 + 152)/412
```

```
## [1] 0.8640777
```

So the model is 86.4% accurate on the test data. This is an alright estimate of forgeries but almost 15% will not be caught by this model. Also, there are almost equal false forgeries (32) and false real notes (24)

iii.) Using $p(x) = 0.3$ and $p(x) = 0.6$, compute the confusion matrices. Comment when $p(x) = 0.3$ could be desirable.

Starting with 0.6:

```r
test_probs = predict(training_glm,BankTest,type = "response")
test_glm_pred=rep("Real Banknote",412)
test_glm_pred[test_probs>.6]="Forged Banknote"
test_glm_pred = factor(test_glm_pred, levels = c("Real Banknote", "Forged Banknote"))
table(test_glm_pred,BankTest$y)
```

```
##
## test_glm_pred       0   1
##   Real Banknote   210  35
##   Forged Banknote  26 141
```

```r
(210 + 141)/412
```

```
## [1] 0.8519417
```

Moving to $p(x) = 0.6$ has slightly decreased the accuracy of the model. This has also flipped the proportion of false forgeries and false real notes.

Now with $p(x) = 0.3$

```r
test_probs = predict(training_glm,BankTest,type = "response")
test_glm_pred=rep("Real Banknote",412)
test_glm_pred[test_probs>.3]="Forged Banknote"
test_glm_pred = factor(test_glm_pred, levels = c("Real Banknote", "Forged Banknote"))
table(test_glm_pred,BankTest$y)
```

```
##
## test_glm_pred       0   1
##   Real Banknote   183   5
##   Forged Banknote  53 171
```

```r
(183 + 171)/(412)
```

```
## [1] 0.8592233
```

This model is also slightly less accurate however, only 5 real banknotes were classified as forgeries and so this model would be useful when trying to keep as many real banknotes in circulation as possible while having a number of forgeries stay in circulation.

Problem 3

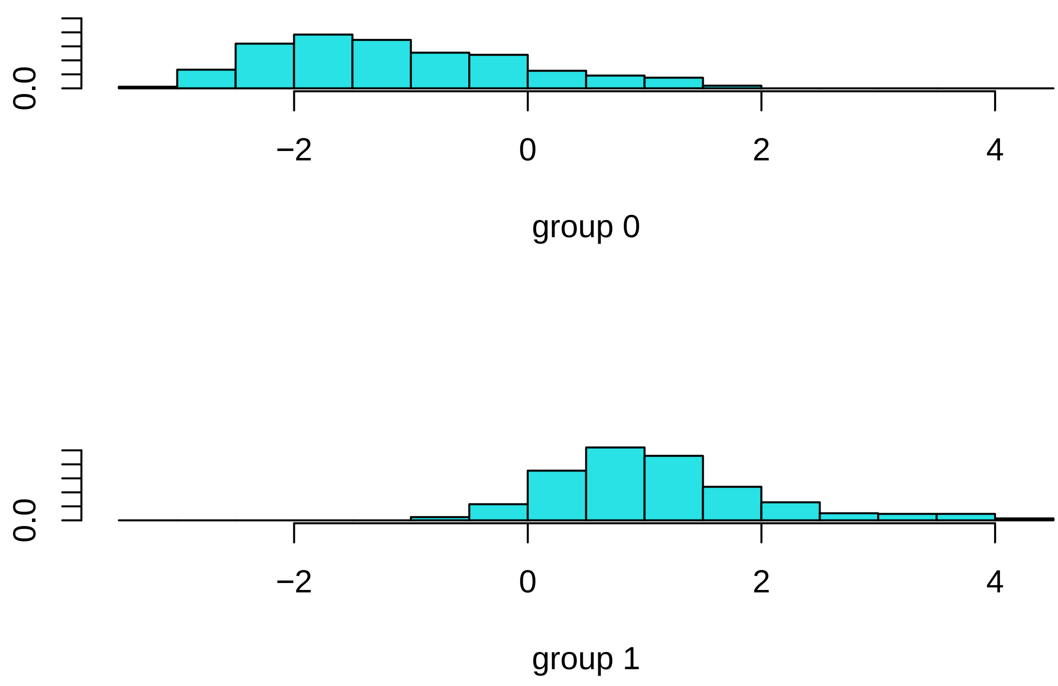
In this problem linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) models will be fit to the training data set.

i.)Perform LDA analysis on the data set

```r
library(MASS)
lda_fit=lda(y~x1+x3, data=BankTrain)
lda_fit
```

```
## Call:
## lda(y ~ x1 + x3, data = BankTrain)
##
## Prior probabilities of groups:
##         0         1
## 0.5479167 0.4520833
##
## Group means:
##          x1       x3
## 0  2.322977 0.938296
## 1 -1.870594 2.114927
##
## Coefficients of linear discriminants:
##             LD1
## x1 -0.55425154
## x3 -0.07209638
```

```
plot(lda_fit)
```



From the prior probabilities we can see that ~54.8% of the training data are real banknotes and ~45.2% are fakes. We see that the data is roughly normal in nature and so this could be a good method of analysis. To determine this, lets look at the confusion matrix.

```
lda_probs = predict(lda_fit,BankTest,type = "response")
lda_class=lda_probs$class
table(lda_class,BankTest$y)
```

```
##
## lda_class   0   1
##         0 203  22
##         1  33 154
```

```
mean(lda_class==BankTest$y)
```

```
## [1] 0.8665049
```

This model is 86.7% accurate.

ii)Repeat using QDA

```
qda_fit=qda(y~x1+x3, data=BankTrain)
qda_fit
```

```
## Call:
## qda(y ~ x1 + x3, data = BankTrain)
##
## Prior probabilities of groups:
##         0         1
## 0.5479167 0.4520833
##
## Group means:
##          x1       x3
## 0  2.322977 0.938296
## 1 -1.870594 2.114927
```

```
qda_class=predict(qda_fit,BankTest)$class
table(qda_class,BankTest$y)
```

```
##
## qda_class   0   1
##         0 208  18
##         1  28 158
```

```
mean(qda_class==BankTest$y)
```

```
## [1] 0.8883495
```

From the confusion matrix we see that the QDA model is slightly more accurate at 88.8%.

iii.) Compare with the logistic regression for $p(x) = 0.5$

Comparing the accuracy of the models we have: QDA: 88.8%, LDA: 86.7% and GLM:86.4%. The LDA and GLM models have pretty similar false positive and false negative rates while the QDA model does slightly better on both which increases its accuracy. Because of this, I would choose to use the QDA model over the other two for its slightly improved accuracy at separating forgeries and real banknotes.

Problem 4

Consider a binary classifcation problem $Y \in \{1, 0\}$; 1g with one predictor X. Assume that X is normally distributed in each class with $X : N(0, 4) = f_0(x)$ in class 0 and $X : N(2, 4) = f_1(x)$ in class 1. Calculate Bayes error rate when the prior probability of being in class 0 is $\pi_0 = 0.4$.

Since $\pi_0 = 0.4$, $\pi_1 = 0.6$

To find the Bayes error rate, the decision boundary of this classifier must be found first. This is found at:

$$\pi_0 f_0(X) = \pi_1 f_1(x)$$

Since the variance is constant in both, this can be solved with linear discriminant analysis so,

$$\delta_0(x) = \delta_1(x)$$

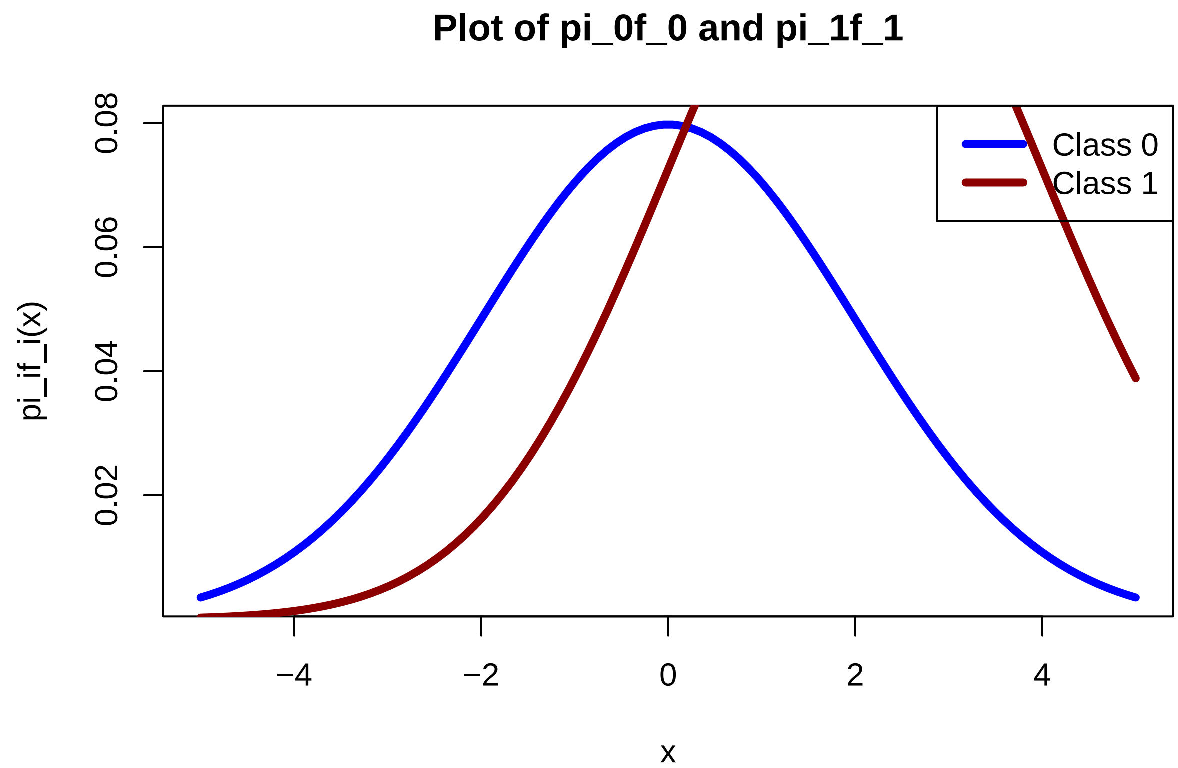
$$\frac{\mu_0 x}{\sigma^2} - \frac{\mu_0^2}{2\sigma^2} + ln(\pi_0) = \frac{\mu_1 x}{\sigma^2} - \frac{\mu_1^2}{2\sigma^2} + ln(\pi_1)$$

$$x(\frac{\mu_0 - \mu_1}{\sigma^2}) = \frac{\mu_0^2 - \mu_1^2}{2\sigma^2} + ln(\frac{\pi_1}{\pi_0})$$

```
#Plot out the gaussians to see where they should cross (check of math)
x = seq(-5,5,length = 100)
pi_0 = 0.4
pi_1 = 1 - pi_0
mu_0 = 0
mu_1 = 2
sigma_2 = 4

#Define the functions
f_0 = pi_0*dnorm(x,mu_0,sqrt(sigma_2))
f_1 = pi_1*dnorm(x,mu_1,sqrt(sigma_2))

#Plot the curve out.
plot(x,f_0,col = "blue", lwd = 4 ,type  ='l', main = "Plot of pi_0f_0 and pi_1f_1",
     xlab = "x", ylab = "pi_if_i(x)")
#Plot the second curve.
points(x, f_1, col="dark red", lwd = 4, type = 'l')
legend("topright",legend = c("Class 0", "Class 1"),
        col = c("blue","dark red"),lwd = 4,
        text.col = "black",
        horiz = FALSE)
```




```
numerator = (mu_0^2 - mu_1^2)/(2*sigma_2) + log(pi_1/pi_0)
denominator = (mu_0 - mu_1)/sigma_2
X = numerator/denominator
X
```

```
## [1] 0.1890698
```

This finds the boundary to be 0.1890698. Now, the Bayes error rate is computed as:

$$\pi_0 P(X > 0.1890698|Y = 0) + \pi_1 P(X < 0.1890698|Y = 1)$$

```
#Given the boundary value compute the integral

int_1 = pnorm(X,mu_1,sigma_2)
int_0 = 1 - pnorm(X,mu_0,sigma_2)
error_rate = pi_1*int_1 + pi_0*int_0
error_rate
```

```
## [1] 0.3876824
```

So, the LDA analysis has a Bayes Error Rate of about 38.9%.