STAT318/462 Assignment 2

Question 1 (2 marks, 1 mark for each correct answer)

```r
# (a)
b0 = -16; b1 = 1.4; b2 = 0.3;
exp(b0 + b1*5 + b2*36)/(1+exp(b0 + b1*5 + b2*36))
```

```
## [1] 0.8581489
```

```r
# (b)
(-b0 - b2*18)/b1
```

```
## [1] 7.571429
```

Question 2 (10 marks)

(a) (3 marks: 2 for fitting the model and 1 for comments)

```r
Train = read.csv('BankTrain.csv')
Test = read.csv('BankTest.csv')
glm.fit <- glm(y~x1+x3,
               Train,
               family=binomial)
summary(glm.fit)
```

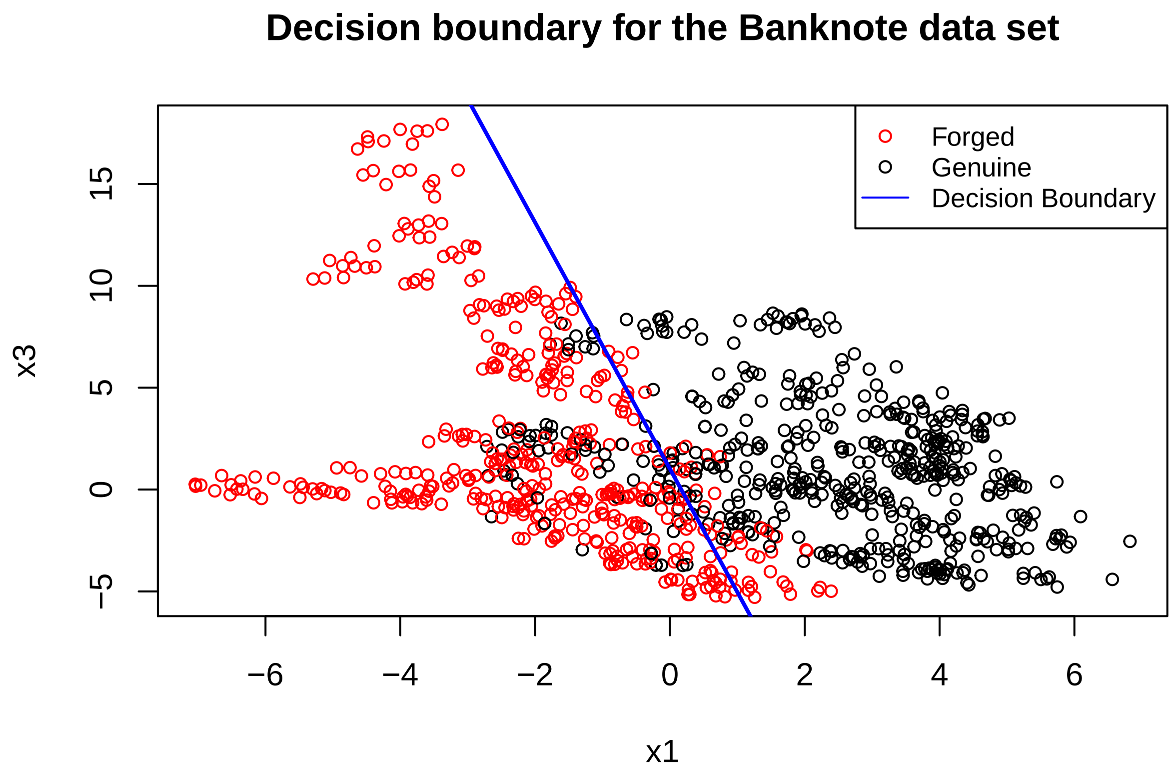
```
##
## Call:
## glm(formula = y ~ x1 + x3, family = binomial, data = Train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.83187  -0.28343  -0.06417   0.50032   1.99366
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.22041    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
```

They must say that the regression coefficients are significant. Other comments include the coefficients are negative, an increase in x_1 corresponds to a decrease in the log odds, etc. **The mark should be given if a reasonable effort was made.**

(bi) (2 marks: 1 for plot, 1 for labelling)

```
beta = coef(glm.fit)
plot(Train$x1,
     Train$x3,
     col=Train$y + 1,
     pch=21,
     cex=0.8,
     xlab="x1",
     ylab="x3",
     main="Decision boundary for the Banknote data set")
i.val <- c(-15,20)
b.val <- (-beta[1] -beta[3]*i.val)/beta[2]
points(b.val,
       i.val,
       col="blue",
       type="l",
       lwd=2)
legend("topright",
       legend = c("Forged", "Genuine","Decision Boundary"),
       col = c("red","black","blue"),
       pch= c(21,21,NA),
       lty=c(NA,NA,1),
       cex = 0.8,
       text.col = "black",
       horiz = FALSE)
```



(b ii) (2 marks: 1 for confusion matrix, 1 for comments)

```
glm.probs <- predict(glm.fit,
                     Test,
                     type="response")
glm.pred <- rep(0,nrow(Test))
glm.pred[glm.probs>0.5]=1
table(glm.pred,
      Test$y)
```

```
##
## glm.pred   0   1
##        0 204  24
##        1  32 152
```

```
testMSE=mean(glm.pred != Test$y)
testMSE
```

```
## [1] 0.1359223
```

At least two observations must be made (0.5 marks for each). the testing error (0.14), the sensitivity (0.86), the specificity (0.86), etc.

(b iii) (3 marks: 1 mark for the confusion matrices, 1 mark for comments and 1 mark for a relevant situation.)

```
glm.probs <- predict(glm.fit,
                     Test,
                     type="response")
glm.pred <- rep(0,nrow(Test))
glm.pred[glm.probs>0.3]=1
table(glm.pred,
      Test$y)
```

```
##
## glm.pred   0   1
##        0 183   5
##        1  53 171
```

```
testMSE=mean(glm.pred != Test$y)
testMSE
```

```
## [1] 0.1407767
```

```
glm.probs <- predict(glm.fit,
                     Test,
                     type="response")
glm.pred <- rep(0,nrow(Test))
glm.pred[glm.probs>0.6]=1
table(glm.pred,
      Test$y)
```

```
##
## glm.pred   0   1
##        0 210  35
##        1  26 141
```

```
testMSE=mean(glm.pred != Test$y)
testMSE
```

```
## [1] 0.1480583
```

At least two observations must be made. the testing error increased in both cases, the sensitivity increased for theta=0.3 (0.97) and the specificity decreased (0.78), the sensitivity decreased for theta=0.6 (0.80) and the specificity increased (0.89) etc. The theta=0.3 threshold could be useful if there were a cost differential between a low-cost type I error (false positive) and a higher cost type II error (false negative).

Question 3 (6 marks)

(a) (2 marks: 1 for test error (can stated in part(c)), 1 for confusion matrix)

```
library(MASS)
lda.fit=lda(y~x1+x3,
            data=Train)
lda.pred=predict(lda.fit,
                 Test)
table(lda.pred$class,
      Test$y)
```

```
##
```

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

```r
testMSE=mean(lda.pred$class != Test$y)
testMSE
```

```
## [1] 0.1334951
```

(b) (2 marks: 1 for test (can stated in part(c)), 1 for confusion)

```r
qda.fit=qda(y~x1+x3,
            data=Train)
qda.pred=predict(qda.fit,
                 Test)
table(qda.pred$class,
      Test$y)
```

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

```r
testMSE=mean(qda.pred$class != Test$y)
testMSE
```

```
## [1] 0.1116505
```

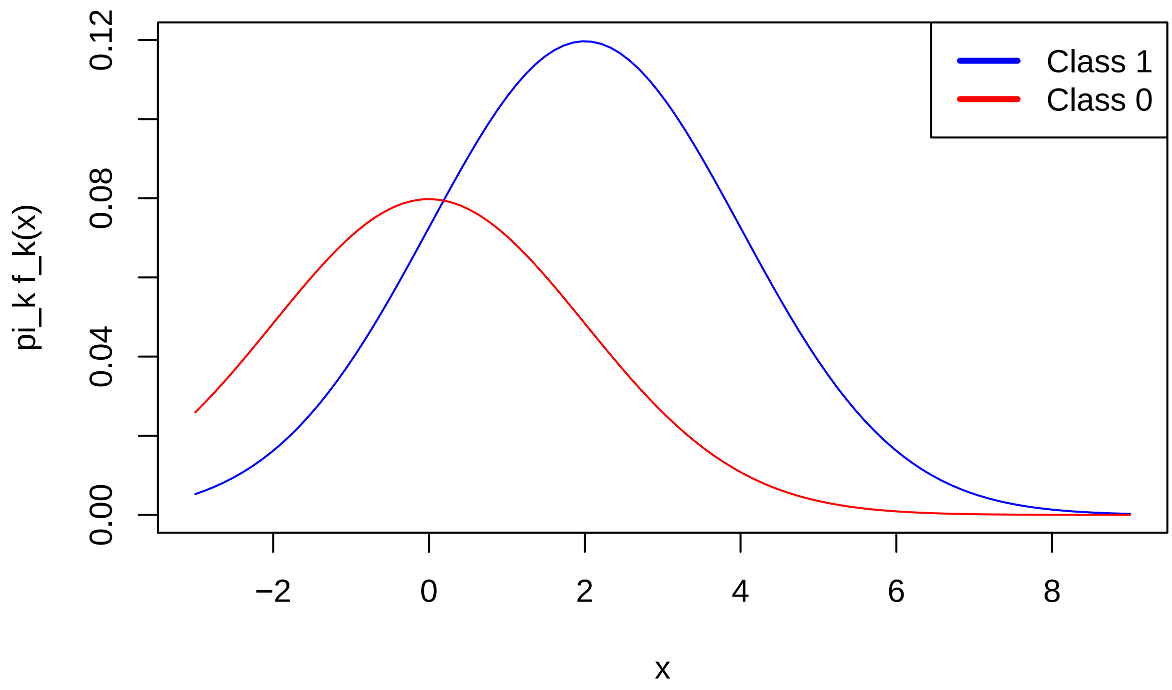
(c) (2 marks: 1 for comments, 1 for making a reasonable recommendation)

Possible comments: QDA had a lower training MSE than LDA, QDA had a lower testing MSE than LDA, LDA was too simple to capture the non-linear boundary, etc. The best model was QDA because it had the lowest testing MSE (lower than LDA and logistic regression).

Question 4 (2 marks: 1 mark for the correct boundary and one mark for the correct error. The plot is not required.)

```r
x = seq(-3,9,length=100)
plot(x,
     0.6*dnorm(x,2,2),
     pch=21,
     col="blue",
     cex=0.6,
     type="l",
     xlab="x",
     ylab="pi_k f_k(x)")
points(x,
       0.4*dnorm(x,0,2),
       pch=21,
       col="red",
       cex=0.6,
       type="l")
legend("topright",
       legend = c("Class 1", "Class 0"),
```

```
       col = c("blue","red"),
       lwd = 3,
       text.col = "black",
       horiz = FALSE)
```




```
b = 2*(log(0.4/0.6) + 0.5)
b
```

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

```
BayesError = 0.4*(1 - pnorm(b, mean = 0, sd = 2))+ 0.6*pnorm(b, mean = 2, sd = 2)
BayesError
```

```
## [1] 0.2945026
```