STAT 318 Assignment 2

Student Name : Xiao Meng

Student ID : 88211337

1.

With the fitting logistic regression model, when a student get a GPA value ≥ 7,

$$ln(\frac{p(Y=1)}{1-p(Y=1)}) = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$$

(a)

If they study for 5 hours and attend 36 classes, which means $X_1 = 5, X_2 = 36$. With the given estimated coefficientes $\hat{\beta}_0 = -16, \hat{\beta}_1 = 1.4, \hat{\beta}_3 = 0.3$, we can get

$$p(Y=1) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X_1+\hat{\beta}_2 X_2}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X_1+\hat{\beta}_2 X_2}}$$

$$= \frac{e^{-16+1.4\times5+0.3\times36}}{1+e^{-16+1.4\times5+0.3\times36}}$$

$$= 0.8518(4dp)$$

(b)

As above, when $$p(Y = 1) = 0.5, X_2 = 18$$,

$$X_1 = \frac{ln(\frac{p(Y=1)}{1-p(Y=1)}) - \hat{\beta}_0 - \hat{\beta}_2 X_2}{\hat{\beta}_1}$$

$$= \frac{0-(-16)-0.3\times18}{1.4}$$

$$= 7.57(4dp)$$

2.

(a)

With the training data, the multiple logistic regression based on predicors X_1, X_3 is

$$ln(\frac{p(X)}{1-p(X)}) = 0.2204 - 1.3149X_1 - 0.2174X_3$$

As the p-value of variables are both much less than 0.001, which represents that both variables are statistically significant. Meanwhile, the coefficients of variables are negative shows that when value of predictors increases, logit value of this logistic regression model will decrease. It will more likely to be a genuine banknote.

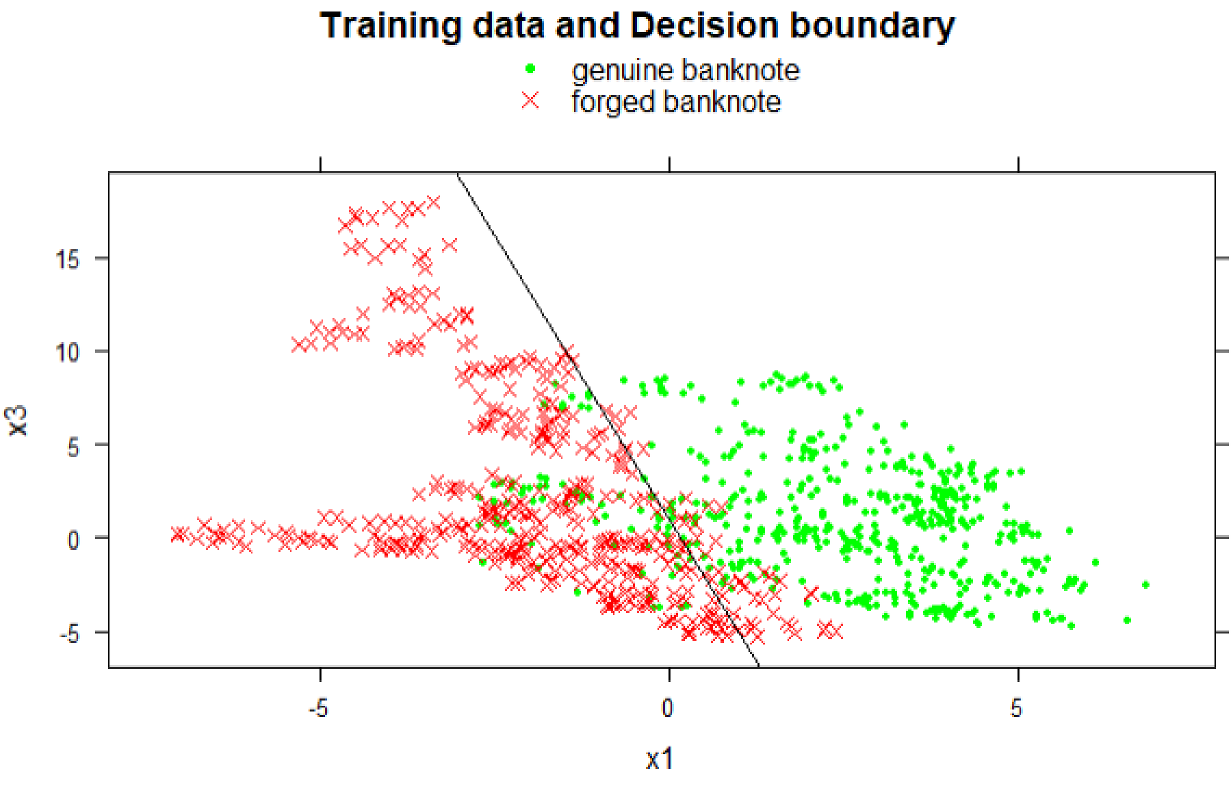
(b)

Using the Bootstrap and 1000 replicates, the estimated standard errors for $\hat{\beta}_1$ is 0.0894(4dp), $\hat{\beta}_2$ is 0.0267(4dp).

	original	bias	std.error
t1*	0.2204101	0.002915229	0.12143542
t2*	-1.3148902	-0.010448191	0.08944161
t3*	-0.2173841	-0.001463668	0.02672804

(c) i

Plot the training data and the decision boundary for $\theta = 0.5$.

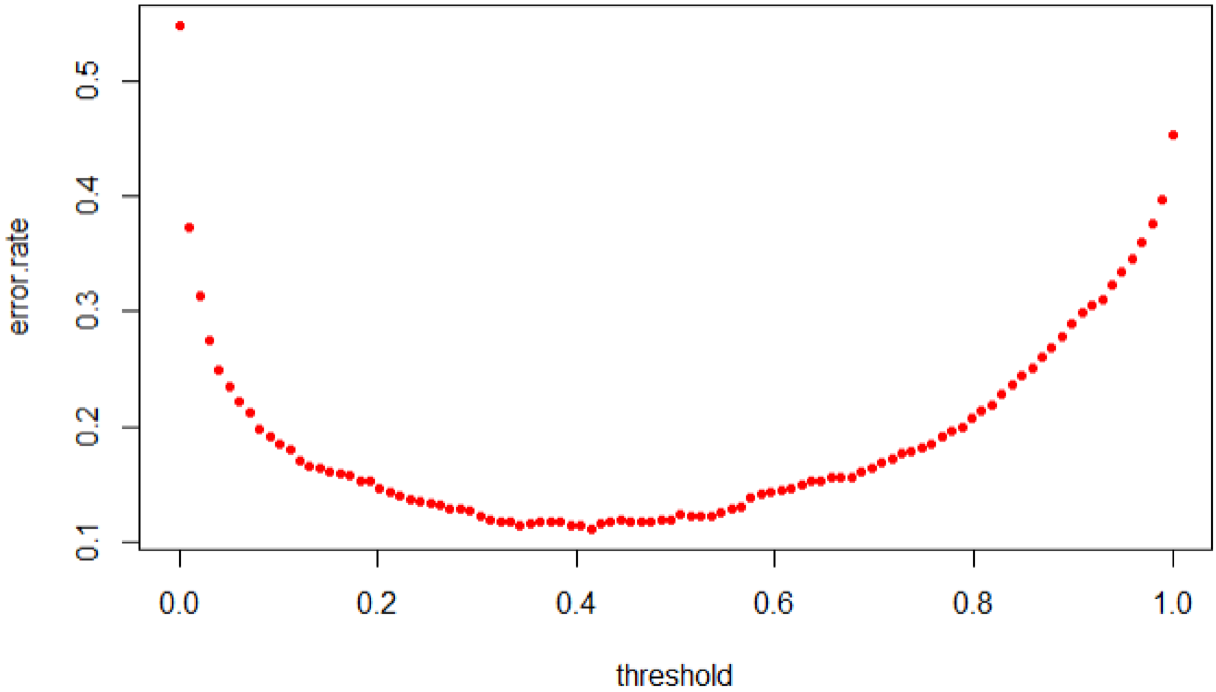


(c) ii

glm.pred	genuine banknote	forged banknote
genuine banknote	204	24
forged banknote	32	152

The accurancy rate is 86.41%($\frac{204+152}{412}$). From the confusion matrix we can see, there are 24 forged banknotes incorrectly assigned to genuine banknotes, while this error rate is 13.64%($\frac{24}{24+152}$). For the actual situation, this error rate is vary important, which might lead some serious consequences.

(c) iii



From the figure above we can see when the threshold is 0.42, there will be the minimal training error rate 11.15%.

glm.pred	genuine banknote	forged banknote
genuine banknote	200	13
forged banknote	36	163

Applying this best threshold to testing data, the accurancy rate increases to 88.11%. Meanwhile the incorrectly predicted forged banknotes decreases to 13 and this error rate fails to 7.39%. The new threshold gives a good results.

3.

Model	Training error	Testing error
Logistic Regresion	12.08%	13.59%
LDA	12.08%	13.35%
QDA	11.46%	11.17%

(a)

Fitting an LDA model with training data, the training error is 12.08% and testing error is 13.35%.

(b)

Fitting an QDA model with training data, the training error is 11.46% and testing error is 11.17%.

(c)

From the table above we can see, among these three models, QDA gives the best performance which has both the lowest training error and testing error. While the errors of Logistic Regreesion and LDA are quite similiar.

As LDA regression is based on the assumption that each classification shares the same variance. However the variance of these two classification is 8.17 and 18.92, which is quite different. This reason might make LDA perform worse in this data set.

From figure of training data and the decision boundary, we know these two classifications have some overlapping points, which might make the boundary not to be linear. Therefore, the Logistic regression model has a higher error rate than QDA. Besides, the QDA does not make assumptions about the same variance of each classification. So in this data set, I recommand the QDA which performs the best.

4.

As $\pi_0 = 0.4, \pi_1 = 1 - \pi_0 = 0.6$, the decision boundary is

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

$$0.4 \times \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{8}x^2} = 0.6 \times \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{8}(x-2)^2}$$

$$e^{-\frac{1}{8}x^2 + \frac{1}{8}(x-2)^2} = \frac{3}{2}$$

$$e^{\frac{1-x}{2}} = \frac{3}{2}$$

$$x = 1 - 2ln(\frac{3}{2})$$
$$x = 0.1891(4dp)$$

The Bayes error rate is

$$1 - (\int_{-\infty}^{0.1891} \pi_0 f_0(x) + \int_{0.1891}^{\infty} \pi_1 f_1(x))$$

$$= 1 - (\int_{-\infty}^{0.1891} 0.4 \times \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{8}x^2} + \int_{0.1891}^{\infty} 0.6 \times \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{8}(x-2)^2})$$

$$= 1 - (0.2151 + 0.4904)$$

$$= 0.2945(4dp)$$

Appendix R Code

```r
#=====================
#------- 1(a) --------
#=====================
beta0 = -16
beta1 = 1.4
beta2 = 0.3
x1 = 5
x2 = 36
y = (exp(beta0 + beta1 * x1 + beta2 * x2))/(1 + exp(beta0 + beta1 * x1 + beta2 * x2))
y


#=====================
#------- 1(b) --------
#=====================
y = 0.5
x2 = 18
(log(y / (1 - y)) - beta0 - beta2 * x2)/beta1


#=====================
#------- 2(a) --------
#=====================
#fit the training datas from BankTrain.csv with logistic model
#Banktrain = read.csv("/Users/mac/Desktop/Computer Science/Data Mining/Assignment/A2/BankTrain.csv")

Banktrain = read.csv("E:/文档/UC/318/Assignment/A2/BankTrain.csv")
glm.fit = glm(y ~ x1 + x3 , data = Banktrain, family =  binomial)
summary(glm.fit)


#=====================
#------- 2(b) --------
#=====================
library(boot)
set.seed(2)
#create a new function to fit the model with a single row
boot.fn = function(data, index){
  return (coef(glm(y ~ x1 + x3, data = Banktrain, family = binomial, subset = index)))
}

#estimate the standard errors for coefficients using the bootstrap
boot(data = Banktrain, statistic = boot.fn, R = 1000)
```

```r

#compare to the standard errors calculating by the model
summary(glm(y ~ x1 + x3, data = Banktrain, family = binomial))


#=====================
#------- 2(c) --------
#=====================
library(lattice)
# parameters of boundary when threshold = 0.5
slope = -coef(glm.fit)[2] / coef(glm.fit)[3]
intercept = -coef(glm.fit)[1] / coef(glm.fit)[3]

xyplot(x3 ~ x1, data = Banktrain, groups = y, pch = c(20, 4), col = c("green", "red"), main = 'Training data and Decision boundary',
       key=list(space='top',
                points = list(pch = c(20, 4), col = c('green', 'red')),
                text = list(lab = c('genuine banknote', 'forged banknote'))),
       panel = function(...){
         panel.xyplot(...)
         panel.abline(intercept, slope)})


#=====================
#------- 2(d) --------
#=====================
#predict probabilities with testing datas from BankTest.csv
Banktest = read.csv("E:/文档/UC/318/Assignment/A2/BankTest.csv")

glm.probs = predict(glm.fit, Banktest, type = "response")

#create a vector with the results of predictions from logistic model
glm.pred = rep("0", 412)
glm.pred[glm.probs > .5] = "1"

#createa a matrix to classify how the predictions perform
table(glm.pred, Banktest$y)
mean(glm.pred == Banktest$y)


#=====================
#------- 2(e) --------
#=====================
#predict probabilities with testing datas from BankTest.csv

glm.probs.train = predict(glm.fit, Banktrain, type = "response")

#create a vector with the results of predictions from logistic model
threshold = seq(0, 1, length = 100)
error.rate = list()

for(x in threshold){
  glm.pred.train = rep("0", 960)
```

```r
  glm.pred.train[glm.probs.train > x] = "1"
  error.traing = mean(glm.pred.train != Banktrain$y)
  error.rate = c(error.rate, list(error.traing))
}
min(unlist(error.rate))
match(c(min(unlist(error.rate))), error.rate)

plot(threshold, error.rate, pch = 20, col = 'red')

glm.probs.test = predict(glm.fit, Banktest, type = "response")
glm.pred.test = rep("0", 412)
glm.pred.test[glm.probs.test > .42] = "1"

#createa a matrix to classify how the predictions perform
table(glm.pred.test, Banktest$y)
mean(glm.pred.test == Banktest$y)



#=====================
#------- 3(a) --------
#=====================
#LDA model
library(MASS)
lda.fit = lda(y ~ x1 + x3, data = Banktrain)

#predict classification of training data and training error
lda.class.train = predict(lda.fit, Banktrain)$class
mean(lda.class.train != Banktrain$y)


#predict classification of test data and test error
lda.class.test = predict(lda.fit, Banktest)$class
mean(lda.class.test != Banktest$y)


#=====================
#------- 3(b) --------
#=====================
#ODA model
qda.fit = qda(y ~ x1 + x3, data = Banktrain)

#predict classification of training data and training error
qda.class.train = predict(qda.fit, Banktrain)$class
mean(qda.class.train != Banktrain$y)

#predict classification of test data and test error
qda.class.test = predict(qda.fit, Banktest)$class
mean(qda.class.test != Banktest$y)


#=====================
#------- 3(c) --------
```

```r
#=====================
#Compute training  error and test error of logistic model

#train error
glm.probs.train1 = predict(glm.fit, Banktrain, type = "response")
glm.pred.train1 = rep("0", 960)
glm.pred.train1[glm.probs.train1 > .5] = "1"
mean(glm.pred.train1 != Banktrain$y)

#test error
glm.probs.test1 = predict(glm.fit, Banktest, type = "response")
glm.pred.test1 = rep("0", 412)
glm.pred.test1[glm.probs.test1 > .5] = "1"
mean(glm.pred.test1 != Banktest$y)

sd(Banktrain$x1)^2
sd(Banktrain$x3)^2


#=====================
#-------   4   -------
#=====================
fx = function(x, pi0, pi1){return (pi0 * dnorm(x, 0, 2) - pi1 * dnorm(x, 2, 2))}
root = uniroot(fx, c(-5, 5), pi0 = 0.4, pi1 = 0.6, tol = 0.0001)
boundary = root$root
boundary
pi0 = 0.4
pi1 = 1 - pi0
fx0 = function(x) pi0 * dnorm(x, 0, 2)
fx1 = function(x) pi1 * dnorm(x, 2, 2)
1 - (integrate(fx0, -Inf, boundary)$value + integrate(fx1, boundary, Inf)$value)
```