STAT462 ASSIGNMENT 2

(Chen Liang, Student ID 46275313)

Question1

(a) Known X_1 = 2h, X_2 = 30 and the fitting model is a logistics regression model, which means the log odds should be,

$$\ln\left(\frac{p(Y=1)}{1-\ p(Y=1)}\right) = \ \beta_0 +\ \beta_1 X_1 +\ \beta_2 X_2$$

therefore,

$$\begin{aligned}\mathrm{p}(Y=1) &= \frac{e^{\beta_0+\ \beta_1 X_1+\ \beta_2 X_2}}{1+\ e^{\beta_0+\ \beta_1 X_1+\ \beta_2 X_2}}\\ &= \frac{e^{-5+\ 1.3*2+\ 0.01*30}}{1+\ e^{-5+\ 1.3*2+\ 0.01*30}}\\ &= 0.1091\end{aligned}$$

(b) If p(Y = 1) =0.5 and X_2 = 30, then

$$\begin{aligned}X_1 &= \frac{\ln\left(\frac{p(Y=1)}{1-\ p(Y=1)}\right) -\ \beta_2 X_2 -\ \beta_0}{\beta_1}\\ &= \frac{ln(0.5/0.5) - 0.01*30+5}{1.3}\\ &= 3.62\ (h)\end{aligned}$$

Question 2

(a)

Use set.seed(1), sample(nrow(banknote), 0.7*nrow(banknote)) to get the randomly split training set, and let the rest of the dataset to be the testing set. The obtained training set contains 960 observations and the testing set contains 412 observations.

(b)

Fit the multiple logistic regression model based on training set and conduct summary() to check the coefficient and statistics of the model (shown in Table 1).

$$\mathrm{y} = 0.60357 - 1.11426x_1 - 0.27708x_2 \qquad \text{(Model 1)}$$

Table 1 Summary Statistics of Model 1

	Estimate	Std. Error	Z value	Pr(>|z|)	assessment
Intercept	0.60357	0.13325	4.53	5.91e-06	***
x_1	-1.11426	0.07754	-14.37	< 2e-16	***
x_2	-0.27708	0.02687	-10.31	< 2e-16	***
AIC	508.42				

From Table 1, it could be seen that both of the predictors which are x_1 and x_2 are significant (p value is less than 0.001). The negative coefficients of x_1 and x_2 mean that when their value is high, the banknote is less likely forged.

(c)

Firstly, build a function to get the coefficients of the model. Then set the number of replicates as 1000 to conduct the bootstrap and the result is as followed,

Table 2 Result of Bootstrap

	original	bias	Std. error
t_1^*	0.60357	0.01490	0.13441
t_2^* (β_1)	-1.11426	-0.01074	0.06694
t_3^*(β_2)	-0.27708	-0.00241	0.02503

Therefore, the standard error estimated using bootstrap is 0.06694 for β_1, and 0.02503 for β_2.

(d)

Firstly, on the decision boundary, pr(y = 1) is 0.5 which means the odds is 1. Thus, ln(pr(y = 1)/pr(y = 0)) = 0 and on the decision boundary we could know that $0.60357 - 1.11426x_1 - 0.27708x_2 = 0$.

Then, training dataset and testing dataset are plotted (taking x_1 as the x axis and x_2 as the y axis) and decision boundary is sketched on them which is displayed in Figure 1.

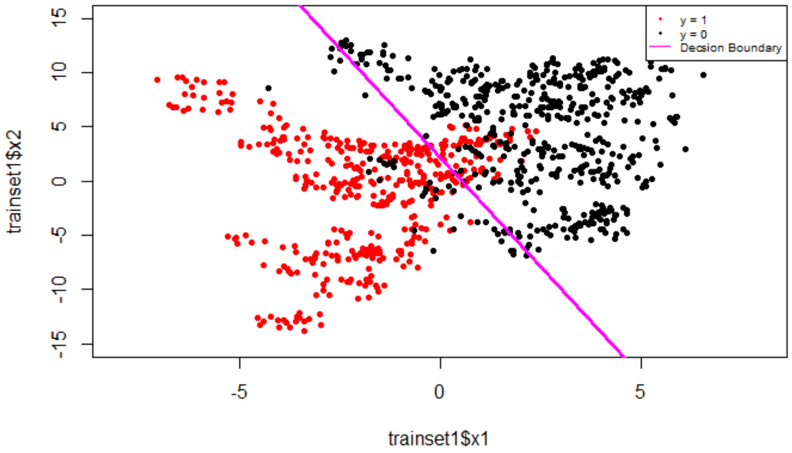


Figure 1 Decision Boundary of y = 1 and y = 0

(e)

Table 3 Confusion Matrix

Bankmodel.pred	Genuine banknote	Forged banknote
Forged banknote	22	163
Genuine banknote	199	28
Accuracy rate	87.86%	

Through the confusion matrix, it can be checked that the prediction accuracy rate of the model based on testing set is 87.86%. In addition, for the forged banknote, the error rate of prediction is 28/(28+163) = 14.66%. This rate is very important for the actual situation, because if we could not identify forged banknotes, it will cause some serious consequences.

Question 3

(a)

Using bankmodel_lda<-lda(y~x_1+x_2,trainset) to fit the LDA model based on the training dataset and calculating the training and testing error is 11.67% and 12.14% respectively.

(b)

Using bankmodel_qda<-qda(y~x_1+x_2,trainset) to fit the QDA model based on the training dataset and calculating the training and testing error is 10.31% and 11.17% respectively.

(c)

The training error and testing error of the three methods is shown as below,

Table 4 Training and Testing Error of the three Methods

	Logistic regression	lda	qda
Training error	11.14%	11.67%	10.31%
Testing error	12.14%	12.14%	11.17%

It can be seen from the Table 4 that the highest prediction accuracy is produced by the model fitted through QDA, followed by the logistic regression model, and the worst is the model fitted using LDA.

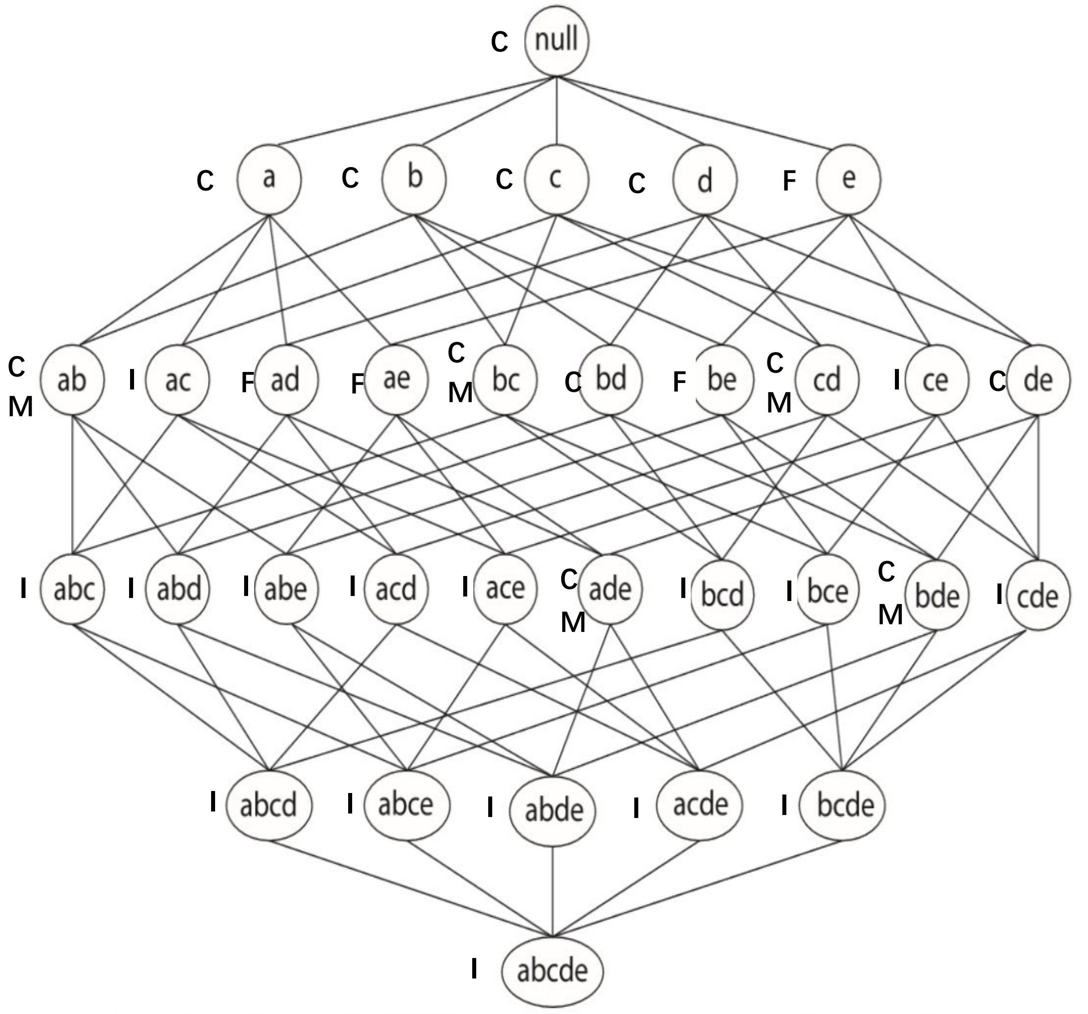
There are two possible reasons that lda perform not well enough. Firstly, from Figure1, it can be seen that the two categories of data will overlap after projection, which will affect the accuracy of LDA. On the other hand, since LDA requires all the categories are normal distributed and have the same covariance, so the accuracy of the model will be affected if the data do not match the assumption.

Logistic regression which uses the maximum likelihood to fit the coefficient is therefore affected less by the overlap and the model will be more appropriate. However, in Figure 1, we can see that the actual boundary tends to be quadratic. QDA regression which assumes a quadratic form and has no assumption about the covariance is more suitable and should be recommended in this case.

Question 4

(a)

The type of the itemset is marked on the left hand.



(b)

the confidence and lift for the rule {d, e} → {a} is,

$$c(\{\mathrm{d}, \mathrm{e}\} \rightarrow \ \{a\}) \ = \ \frac{\sigma(\{\mathrm{d}, \mathrm{e}\} \cup \{a\})}{\sigma(\{\mathrm{d}, \mathrm{e}\})} = \frac{4}{6} = \frac{2}{3}$$

$$\mathrm{lift}(\{\mathrm{d}, \mathrm{e}\}, \{a\}) = \frac{s(\{\mathrm{d}, \mathrm{e}\} \cup \{a\})}{s(\{\mathrm{d}, \mathrm{e}\})S(\{a\})} = \frac{c(\{\mathrm{d}, \mathrm{e}\} \rightarrow \ \{a\})}{s(Y)} = \frac{{}^{2}/_{3}}{{}^{1}/_{2}} = \frac{4}{3}$$

The value of confidence of the rule represents that there is 66.67% possibility for ‘a’ to appear in transactions that contain {d, e}, which means that when given {b, e}, the probability of {a} is estimated by association rule is 66.67%. The value of lift is greater than 1 stands that {b, e} and {a} are positively correlated.

(c)

$$\mathrm{Odds}(X, Y) = \frac{s(X \cup Y)s(\neg X \cup \neg Y)}{s(X \cup \neg Y)s(\neg X \cup Y)}$$

$$= \frac{\frac{\sigma(X \cup Y)}{N} * \frac{\sigma(\neg X \cup \neg Y)}{N}}{\frac{\sigma(X \cup \neg Y)}{N} * \frac{\sigma(\neg X \cup Y)}{N}} = \frac{\sigma(X \cup Y)\sigma(\neg X \cup \neg Y)}{\sigma(X \cup \neg Y)\sigma(\neg X \cup Y)}$$

The itemset containing X but not Y ($X \cup \neg Y$) could be considered as extracting the itemset containing both X and Y from the itemset containing just X, which means,

$$\sigma(X \cup \neg Y) = \sigma(X) - \sigma(X \cup Y)$$

Similarly,

$$\sigma(\neg X \cup Y) = \sigma(Y) - \sigma(X \cup Y)$$

Therefore, when adding some null transactions to change N, since null transactions do not contain both X and Y, $\sigma(X \cup Y)$, $\sigma(X \cup \neg Y)$ and $\sigma(\neg X \cup Y)$ will not change. At the same time, $\sigma(\neg X \cup \neg Y) = N - \sigma(X \cup Y) - \sigma(X \cup \neg Y) - \sigma(\neg X \cup Y)$ will change as the increase or decrease of N. In sum, odds will then be different as adding or dropping null transactions, which means it is not null invariant.

Question 5

Firstly, calculating the Bayes Boundary,

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

$$\pi_1 \frac{1}{\sqrt{2\pi}\sigma_1} exp\left(-\frac{1}{2\sigma_1^2}(x - \mu_1)^2\right) = \pi_0 \frac{1}{\sqrt{2\pi}\sigma_0} exp\left(-\frac{1}{2\sigma_0^2}(x - \mu_0)^2\right)$$

$$0.7 * \frac{1}{\sqrt{2\pi} * 2} exp\left(-\frac{1}{2 * 2^2}(x - 2)^2\right) = 0.3 * \frac{1}{\sqrt{2\pi} * 2} exp\left(-\frac{1}{2 * 2^2}(x - 0)^2\right)$$

$$\ln\frac{7}{3} = -\frac{1}{2 * 2^2}(x - 0)^2 + \frac{1}{2 * 2^2}(x - 2)^2$$

$$x \quad = -0.6946$$

Then, the error rate should be the part in 0 but is put into 1 and the part in 1 but is put into 0 under the density curve.

$$\int_{-\infty}^{-0.6946} 0.7 * \frac{1}{\sqrt{2\pi} * 2} exp\left(-\frac{1}{2 * 2^2}(x - 2)^2\right) dx + \int_{-0.6946}^{+\infty} 0.3 * \frac{1}{\sqrt{2\pi} * 2} exp\left(-\frac{1}{2 * 2^2}(x - 0)^2\right) dx$$

$$= 0.062259 + 0.190745 = 25.3\%$$

Therefore, the error rate is 25.3%.

APPENDIX: R CODE

```
banknote<-read.csv('D:/2018 first semester/datamining/Banknote.csv',header = TRU
E)
dim(banknote)
```

```
## [1] 1372    5
```

```
y1<-banknote$y
banknote$y<-factor(y1,levels = c(0,1),labels = c('genuine banknote','forged bank
note'))
head(banknote)
```

```
##        x1      x2      x3       x4                y
## 1 3.62160  8.6661 -2.8073 -0.44699 genuine banknote
## 2 4.54590  8.1674 -2.4586 -1.46210 genuine banknote
## 3 3.86600 -2.6383  1.9242  0.10645 genuine banknote
## 4 3.45660  9.5228 -4.0112 -3.59440 genuine banknote
## 5 0.32924 -4.4552  4.5718 -0.98880 genuine banknote
## 6 4.36840  9.6718 -3.9606 -3.16250 genuine banknote
```

```
banknote<-banknote[,-3]
banknote<-banknote[,-3]

head(banknote)
```

```
##        x1      x2                y
## 1 3.62160  8.6661 genuine banknote
## 2 4.54590  8.1674 genuine banknote
## 3 3.86600 -2.6383 genuine banknote
## 4 3.45660  9.5228 genuine banknote
## 5 0.32924 -4.4552 genuine banknote
## 6 4.36840  9.6718 genuine banknote
```

```
summary(banknote)
```

```
##        x1                 x2                           y
##  Min.   :-7.0421   Min.   :-13.773   genuine banknote:762
##  1st Qu.:-1.7730   1st Qu.: -1.708   forged banknote :610
##  Median : 0.4962   Median :  2.320
##  Mean   : 0.4337   Mean   :  1.922
##  3rd Qu.: 2.8215   3rd Qu.:  6.815
##  Max.   : 6.8248   Max.   : 12.952
```

```
set.seed(1)
train_sub <- sample(nrow(banknote), 0.7*nrow(banknote))
trainset<-banknote[train_sub,]
testset<-banknote[-train_sub,]
head(trainset)
```

```
##           x1      x2                y
## 365   5.7823  5.5788 genuine banknote
## 511   3.5770  2.4004 genuine banknote
## 785  -3.4083  4.8587  forged banknote
## 1244 -5.0676 -5.1877  forged banknote
## 276   3.4312  6.2637 genuine banknote
## 1229 -1.3414 -1.9162  forged banknote
```

```r
head(testset)
```

```
##         x1      x2                y
## 3   3.8660 -2.6383 genuine banknote
## 8   2.0922 -6.8100 genuine banknote
## 12  3.9899 -2.7066 genuine banknote
## 13  1.8993  7.6625 genuine banknote
## 14 -1.5768 10.8430 genuine banknote
## 15  3.4040  8.7261 genuine banknote
```

```r
dim(trainset)
```

```
## [1] 960   3
```

```r
dim(testset)
```

```
## [1] 412   3
```

```r
#fit multiple logistics model
bankmodel1<-glm(y~x1 + x2, data = trainset, family=binomial)
summary(bankmodel1)
```

```
## 
## Call:
## glm(formula = y ~ x1 + x2, family = binomial, data = trainset)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -2.47599  -0.34480  -0.06152   0.23524   2.59203  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  0.60357    0.13325    4.53 5.91e-06 ***
## x1          -1.11426    0.07754  -14.37  < 2e-16 ***
## x2          -0.27708    0.02687  -10.31  < 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: 1315.30  on 959  degrees of freedom
## Residual deviance:  502.42  on 957  degrees of freedom
## AIC: 508.42
## 
## Number of Fisher Scoring iterations: 6
```

```r
#(c)calculating the std. error
set.seed(2)
library(boot)
```

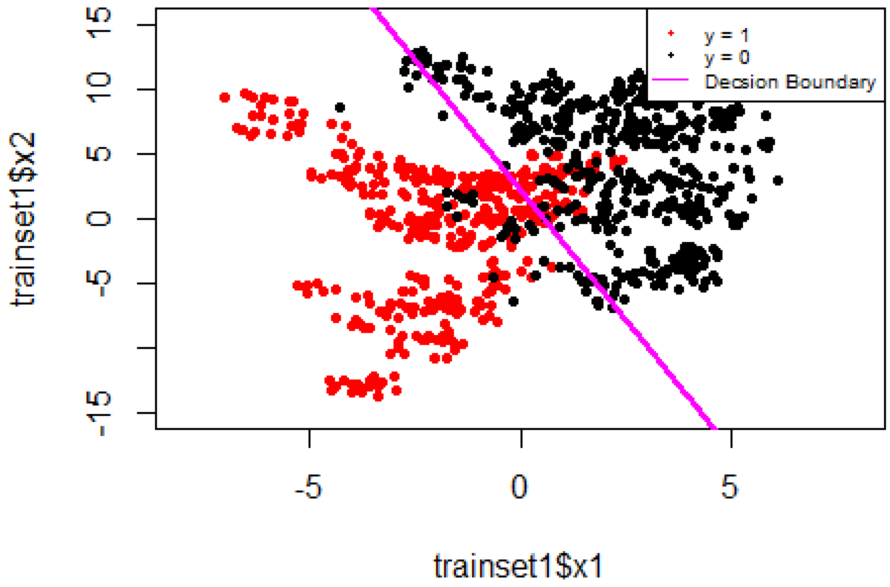
```
## Warning: package 'boot' was built under R version 3.4.4
```

```
boot.fn=function(data,index){
  return(coef(glm(y~(x1+x2),data = trainset,family = binomial(),subset = inde
x)))
}
boot(trainset,boot.fn,R = 1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = trainset, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##       original       bias    std. error
## t1*  0.6035730  0.014901245  0.13441231
## t2* -1.1142595 -0.010743933  0.06693579
## t3* -0.2770824 -0.002407812  0.02502768
```

```
#(d)plot the trainset and decision boudary
trainset1<-subset(trainset,y=='forged banknote',select = c(1,2))
plot(trainset1$x1,trainset1$x2,xlim=c(-8,8),pch = 20, ylim=c(-15,15),col='red')
trainset2<-subset(trainset,y=='genuine banknote',select = c(1,2))

points(trainset2$x1,trainset2$x2, pch = 20)
x1_1<-seq(-8,8,0.001)
points(x1_1,((0.60357-1.11426*x1_1)/0.27708),pch=20,cex=0.4, col='magenta')
text.legend=c('y = 1','y = 0','Decsion Boundary')
legend('topright',pch = c(20,20,-1),lty=c(-1,-1,1),cex=0.6,legend=text.legend,co
l=c('red','black','magenta'))
```



```
#(e)confusion matrix
probs=predict(bankmodel1,testset,type = 'response')
bankmodel.pred=rep('genuine banknote',412)
bankmodel.pred[probs>0.5]='forged banknote'
table(bankmodel.pred,testset$y)

##
## bankmodel.pred     genuine banknote forged banknote
##   forged banknote                22             163
##   genuine banknote              199              28

mean(bankmodel.pred==testset$y)

## [1] 0.8786408

#Question3
#(a)
library(MASS)
bankmodel_lda<-lda(y~x1+x2,trainset)
bankmodel_lda

## Call:
## lda(y ~ x1 + x2, data = trainset)
##
## Prior probabilities of groups:
## genuine banknote  forged banknote
##        0.5635417        0.4364583
##
## Group means:
##                         x1         x2
## genuine banknote  2.282338  4.3066708
## forged banknote  -1.852982 -0.8856809
##
## Coefficients of linear discriminants:
##            LD1
## x1 -0.46997531
## x2 -0.09657778

bankmodel_lda_trainpred<-predict(bankmodel_lda,trainset)
bankmodel_lda_testpred<-predict(bankmodel_lda,testset)
mean(bankmodel_lda_trainpred$class!=trainset$y)

## [1] 0.1166667

mean(bankmodel_lda_testpred$class!=testset$y)

## [1] 0.1213592

#(b)
bankmodel_qda<-qda(y~x1+x2,trainset)
bankmodel_qda

## Call:
## qda(y ~ x1 + x2, data = trainset)
##
## Prior probabilities of groups:
```

```
## genuine banknote  forged banknote
##        0.5635417        0.4364583
##
## Group means:
##                         x1         x2
## genuine banknote  2.282338  4.3066708
## forged banknote  -1.852982 -0.8856809
```

```
bankmodel_qda_trainpred<-predict(bankmodel_qda,trainset)
bankmodel_qda_testpred<-predict(bankmodel_qda,testset)
mean(bankmodel_qda_trainpred$class!=trainset$y)
```

```
## [1] 0.103125
```

```
mean(bankmodel_qda_testpred$class!=testset$y)
```

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

```
bankmodel_glm_trainpred<-predict(bankmodel1,trainset,type = 'response')
bankmodel.pred_train=rep('genuine banknote',960)
bankmodel.pred_train[bankmodel_glm_trainpred>0.5]='forged banknote'
mean(bankmodel.pred_train!=trainset$y)
```

```
## [1] 0.1114583
```

```
bankmodel_glm_testpred<-predict(bankmodel1,testset,type = 'response')
bankmodel.pred_test=rep('genuine banknote',412)
bankmodel.pred_test[bankmodel_glm_testpred>0.5]='forged banknote'
mean(bankmodel.pred_test!=testset$y)
```

```
## [1] 0.1213592
```