Ch4-B Classification Problems Continued

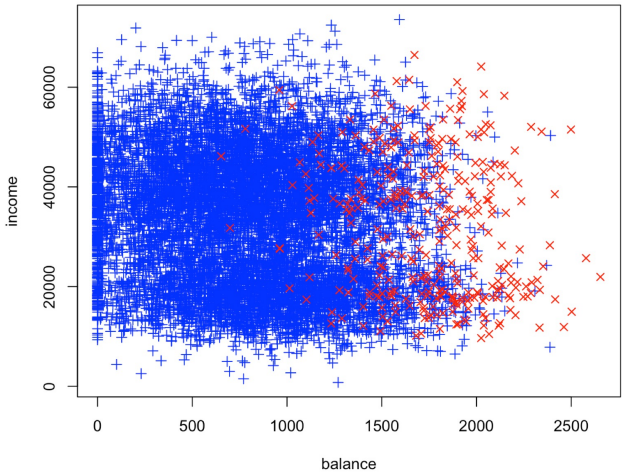
Contents

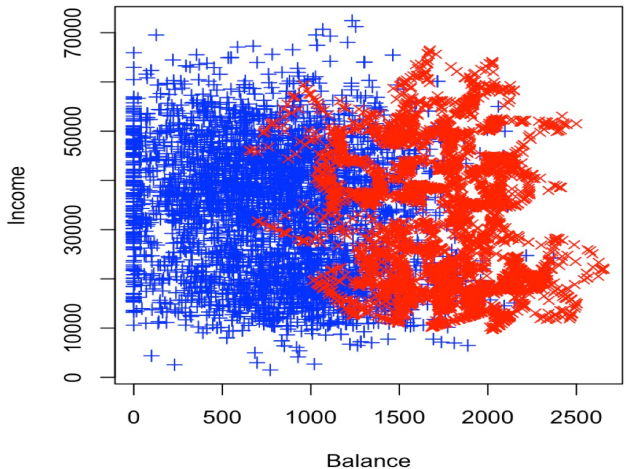
4B Subsection

[ToC]

B.1 Inbalance Problem

```
library(ISLR)
data(Default)
table(Default$default)
  No  Yes
9667  333
```


income
0
20000
40000
60000
0
500
1000
1500
2000
2500
balance


Income
0
10000
30000
50000
70000
0
500
1000
1500
2000
2500
Balance

B.2 What You can do

- Oversample - sample rows with "Yes" with replacement.
- Undersample - randomly select rows with "No" and suppress them.
- SMOTE - Synthetic Minority Over-sampling TEchnique

```
library(DMwR)
?SMOTE
```

- others

B.3 Heart Data

```
###--- 1. Preliminary
library(tidyverse)
Heart <- read_csv(file="http://faculty.marshall.usc.edu/gareth-james/ISL/Heart.csv")
Heart

http://faculty.marshall.usc.edu/gareth-james/ISL/

Heart
# A tibble: 303 x 15

Selected Columns:

   Age   Sex ChestPain   RestBP  Chol   Fbs RestECG MaxHR Oldpeak Slope    Ca Thal     AHD
 <dbl> <dbl> <chr>        <dbl> <dbl> <dbl>   <dbl> <dbl>   <dbl> <dbl> <dbl> <chr>    <chr>
```

63	1	typical	145	233	1	2	150	2.3	3	0	fixed	No
67	1	asymptomat	160	286	0	2	108	1.5	2	3	normal	Yes
67	1	asymptomat	120	229	0	2	129	2.6	2	2	reversa	Yes
37	1	nonanginal	130	250	0	0	187	3.5	3	0	normal	No
41	0	nontypical	130	204	0	2	172	1.4	1	0	normal	No
56	1	nontypical	120	236	0	0	178	0.8	1	0	normal	No
62	0	asymptomat	140	268	0	2	160	3.6	3	2	normal	Yes
57	0	asymptomat	120	354	0	0	163	0.6	1	0	normal	No
63	1	asymptomat	130	254	0	2	147	1.4	2	1	reversa	Yes
53	1	asymptomat	140	203	1	2	155	3.1	3	0	reversa	Yes

```
names(Heart)
#  0  "X1"          index
#  1  "Age"
#  2  "Sex"
#  3  "ChestPain"   (categorical)
#  4  "RestBP"
#  5  "Chol"        cholesterol measurement
#  6  "Fbs"
#  7  "RestECG"
#  8  "MaxHR"
#  9  "ExAng"
#  10 "Oldpeak"
#  11 "Slope"
#  12 "Ca"
#  13 "Thal"        Thalium stress test  (categorical)
#  14 "AHD"         Yes/No based on an angiographic test  Qualitative <-- Response
```

B.4 Logistic Regression

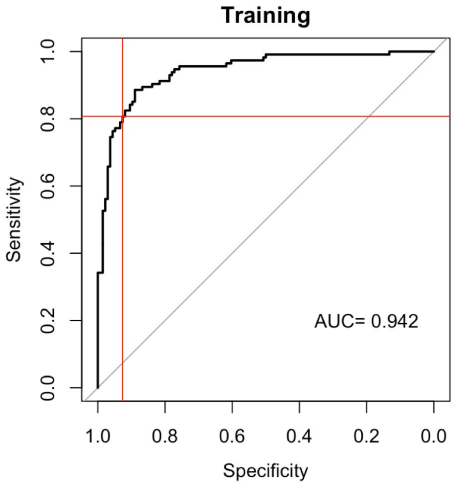
```
Fit02 <- glm(AHD ~ . ,  family=binomial, data=Train.set )
summary(Fit02)

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)        -2.813794   3.175042  -0.886 0.375498
Age                -0.028923   0.027463  -1.053 0.292264
Sex                 1.424383   0.579761   2.457 0.014016 *
ChestPainnonanginal -2.080367   0.578583  -3.596 0.000324 ***
ChestPainnontypical -0.412606   0.628551  -0.656 0.511540
ChestPaintypical    -1.739736   0.725582  -2.398 0.016498 *
RestBP              0.027497   0.012538   2.193 0.028298 *
Chol                0.002791   0.004496   0.621 0.534777
Fbs                -0.309673   0.641538  -0.483 0.629306
RestECG             0.384676   0.214779   1.791 0.073288 .
MaxHR              -0.028656   0.012480  -2.296 0.021664 *
ExAng               0.637727   0.496910   1.283 0.199357
Oldpeak             0.537647   0.264349   2.034 0.041966 *
Slope               0.795655   0.409614   1.942 0.052083 .
Ca                  1.502259   0.315446   4.762 1.91e-06 ***
Thalnormal          0.167320   0.831943   0.201 0.840605
Thalreversable      1.999820   0.832240   2.403 0.016264 *
```

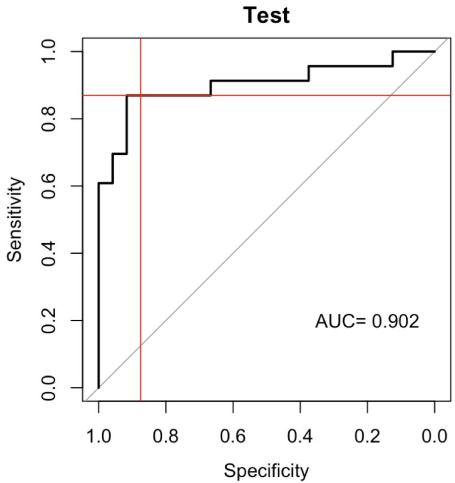
```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 344.64  on 249  degrees of freedom
Residual deviance: 156.92  on 233  degrees of freedom
AIC: 190.92
```

- dummy variable

```
levels(Heart$ChestPain)
  asymptomatic nonanginal nontypical typical
```

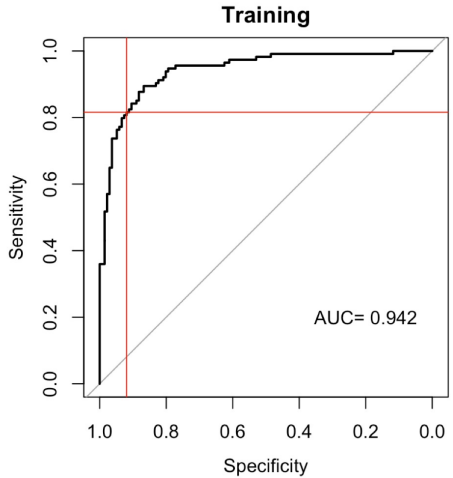
- Variable selection

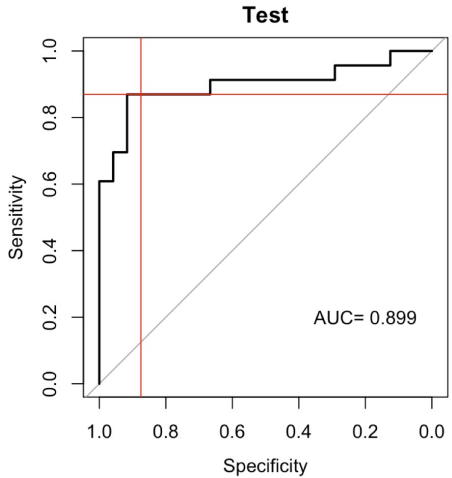

Training
Sensitivity
Specificity
AUC= 0.942
1.0
0.8
0.6
0.4
0.2
0.0


Test
Sensitivity
Specificity
AUC= 0.902
1.0
0.8
0.6
0.4
0.2
0.0

```
Fit02 <- glm(AHD ~ . -Chol ,  family=binomial, data=Train.set )
summary(Fit02)

Call:
glm(formula = AHD ~ . - Chol, family = binomial, data = Train.set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7821  -0.5429  -0.1251   0.2864   3.0508

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         -2.26277    3.04600  -0.743 0.457564
Age                 -0.02713    0.02716  -0.999 0.317733
Sex                  1.32587    0.55271   2.399 0.016447 *
ChestPainnonanginal -2.07941    0.58005  -3.585 0.000337 ***
ChestPainnontypical -0.38972    0.62695  -0.622 0.534198
ChestPaintypical    -1.74576    0.72412  -2.411 0.015915 *
RestBP               0.02751    0.01253   2.195 0.028187 *
Fbs                 -0.29597    0.63686  -0.465 0.642118
RestECG              0.40523    0.21184   1.913 0.055762 .
MaxHR               -0.02834    0.01251  -2.265 0.023530 *
ExAng                0.63825    0.49428   1.291 0.196610
Oldpeak              0.54752    0.26368   2.076 0.037852 *
Slope                0.79240    0.40977   1.934 0.053139 .
Ca                   1.49686    0.31563   4.742 2.11e-06 ***
```

```
Thalnormal           0.19158    0.82813   0.231 0.817049
Thalreversable       2.04787    0.82443   2.484 0.012992 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 344.64  on 249  degrees of freedom
Residual deviance: 157.29  on 234  degrees of freedom
AIC: 189.29
```


Training
Sensitivity
Specificity
AUC= 0.942
1.0
0.8
0.6
0.4
0.2
0.0


Test
Sensitivity
Specificity
AUC= 0.899
1.0
0.8
0.6
0.4
0.2
0.0

B.5 Some method of variable selection

- Stepwise selection - Forward/Backward
- All subset
- Chi-square test for association
- Many others

B.6 Stepwise Selection

- Forward - start with 1 variable model
- Backward - start with all variable model
- Only add 1 variable at a time.
- Use measure of fit within models with same p. (R^2, SSE, ER)
- Then use global measure of fit to decide on best p. (AIC, AUC)

B.7 All Subset Selection

- Use measure of fit that can be used across models with different p. (AIC, AUC)
- 14 variables - $2^{14} = 16384$ models to compare.

B.8 LDA and QDA

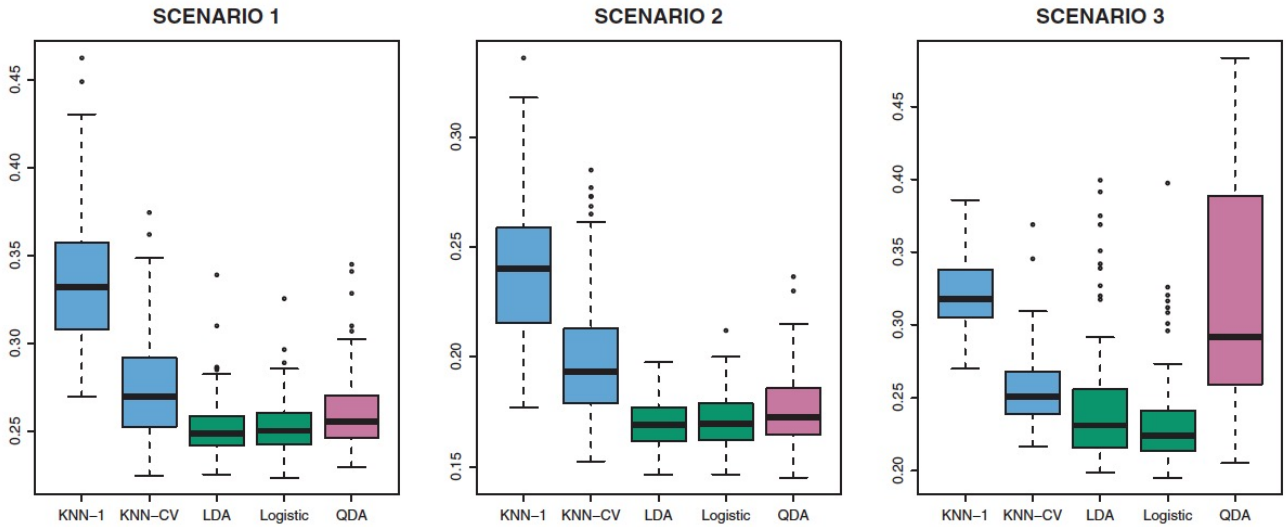
B.9 Classification Method Comparison

- Scenario 1: There were 20 training observations with two predictor variables in each of two classes. The observations within each class were uncorrelated random normal variables with a different mean in each class.

- Scenario 2: As in Scenario 1, except that within each class, the two predictors had a correlation of -0.5.

- Scenario 3: $X1$ and $X2$ were generated from the t-distribution, with 50 observations per class.

Test Error Rates

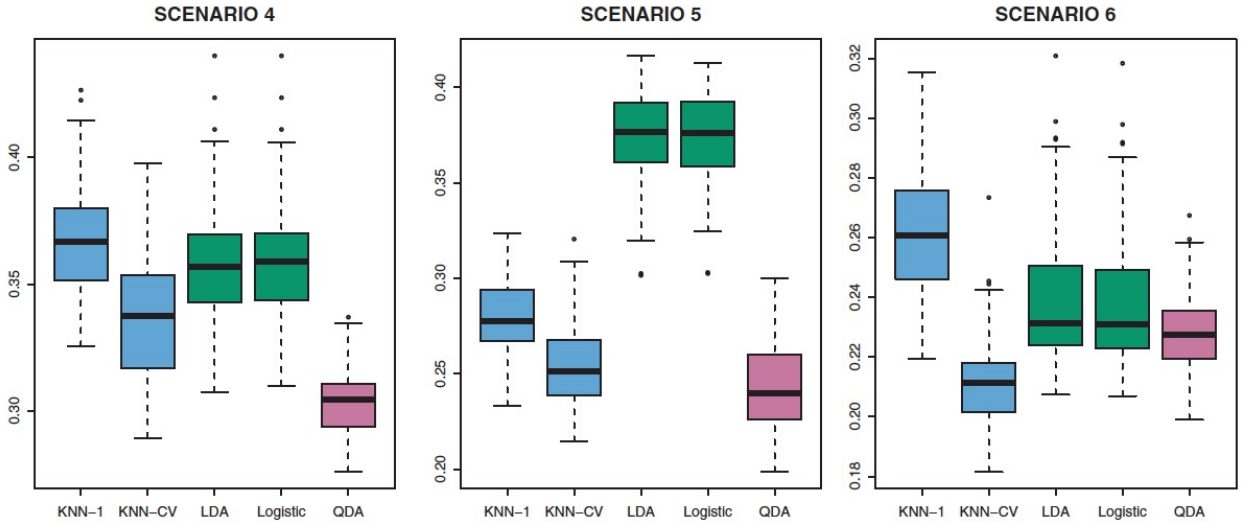


- Scenario 4: The data were generated from a normal distribution, with a correlation of 0.5 between the predictors in the first class, and correlation of -0.5 between the predictors in the second class. This setup corresponded to the QDA assumption, and resulted in quadratic decision boundaries. The left-hand panel of Figure 4.11 shows that QDA outperformed all of the other approaches.

- Scenario 5: Within each class, the observations were generated from a normal distribution with uncorrelated predictors. However, the responses were sampled from the logistic function using $X1^2, X2^2$ and $X1 \times X2$ as predictors. Consequently, there is a quadratic decision boundary.

- Scenario 6: Details are as in Scenario 5, but the responses were sampled from a more complicated non-linear function. As a result, even the quadratic decision boundaries of QDA could not adequately model the data.

Test Error Rates



Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani

B.10 Linear Discriminat Analysis

1. Logistic Regression Modeled

$$P(Y = y|X = x)$$

2. LDA models

$$f_X(x|Y = y) = P(X = x|Y = y)$$

3. Then uses Bayse theorem to flip into $P(Y = y|X = x)$

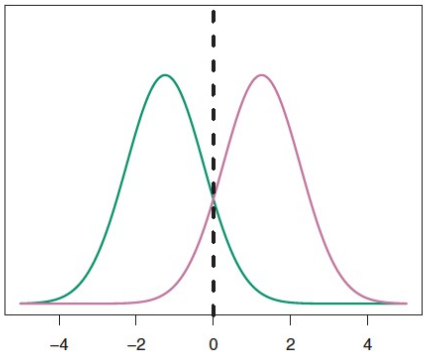
B.11 Reasons for another method

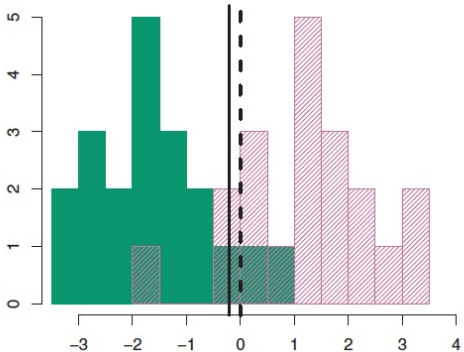
- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. Linear discriminant analysis does not suffer from this problem.
- If n is small and the distribution of the predictors X is approximately normal in each of the classes, the linear discriminant model is again more stable than the logistic regression model.
- LDA is popular when we have more than two response classes.

B.12 Bayes

$$P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{i=1}^{K} \pi_i f_i(x)}$$

Assume f_X is gaussian with same σ





- Assign $X = x$ to k where $P(Y = k|X = x)$ is largest
- Equivalent to assigning obs to the class k with largest value of $\delta_k(x)$ where

$$\delta_k(x) = x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

- If $K = 2$ and $\pi_1 = \pi_2 = .5$, the Bayes decision boundary is at

$$x = \frac{\mu_1^2 - \mu_2^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}$$

- In the real analysis, we woudn't know π_k, μ_k and σ . They are estimated using

$$\hat{\mu}_k = \frac{1}{n_k}\sum_{i:y_i=k} x_i \qquad \hat{\sigma}^2 = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:y_i=k}(x_i - \hat{\mu}_k)^2 \qquad \hat{\pi}_k = \frac{n_k}{n}$$

- Can be extended to case $\sigma_1 \neq \sigma_2 \neq \cdots \neq \sigma_k$ (QDA)
- Default threshold is 50%, but can be adjusted.

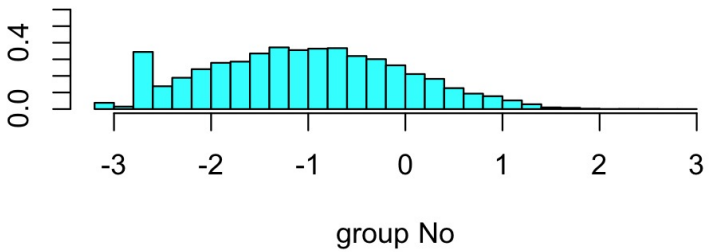
B.13 Default Data

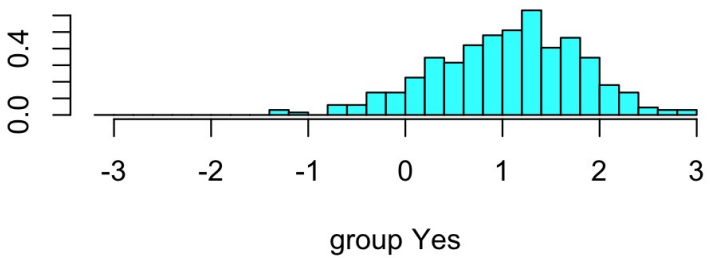
```
library(ISLR)          # Load ISLR package
attach(Default)
names(Default)
#  "default" "student" "balance" "income"


###--- Linear Discriminant Analysis on all data

library(MASS)
lda.fit=lda(default ~ student + balance,  data=Default)
lda.fit
plot(lda.fit)          # same as hist(lda.pred$x[default=="No"], 30, xlim=c(-3,3))
                       #         hist(lda.pred$x[default=="Yes"], 30, xlim=c(-3,3))
```

```
###--- Get fitted Response
lda.pred=predict(lda.fit)

names(lda.pred)           # see what's inside the prediction
#  lda.pred$class         # Yes/No already assigned using threshold of .5
#  lda.pred$posterior     # posterior prob of being "No" and "Yes"

library(caret) # for confusionMatrix
CM <- confusionMatrix(factor(lda.pred$class), factor(default), positive="Yes")
CM

#          Reference
#Prediction   No  Yes
#       No  9644  252
#       Yes   23   81
#
#               Accuracy : 0.9725
```

```
#                 95% CI : (0.9691, 0.9756)
#
#            Sensitivity : 0.2432
#            Specificity : 0.9976
#         Pos Pred Value : 0.7788
#         Neg Pred Value : 0.9745
#
#       'Positive' Class : Yes



#-- Assign Yes/No using threshold of .2
lda.pred2 <- ifelse(lda.pred$posterior[,2]>.2, "Yes", "No")

CM <- confusionMatrix(factor(lda.pred2), factor(default), positive="Yes")
CM

#          Reference
```

```
#Prediction   No  Yes
#       No  9432  138
#       Yes  235  195
#
#               Accuracy : 0.9627
#                 95% CI : (0.9588, 0.9663)
#
#            Sensitivity : 0.5856        # % of Yes caught
#            Specificity : 0.9757        # % of No  caught
#         Pos Pred Value : 0.4535        # % of Yes.pred correct
#         Neg Pred Value : 0.9856        # % of No.pred  correct
```


0.0
0.4
-3
-2
-1
0
1
2
3
group No


0.0
0.4
-3
-2
-1
0
1
2
3
group Yes

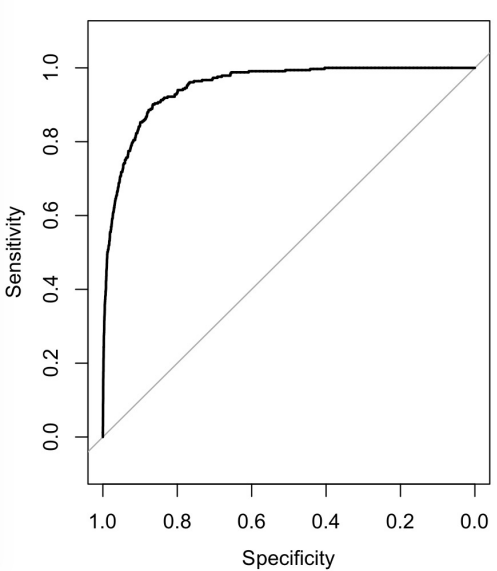
B.14 Changing Threshold and AUC

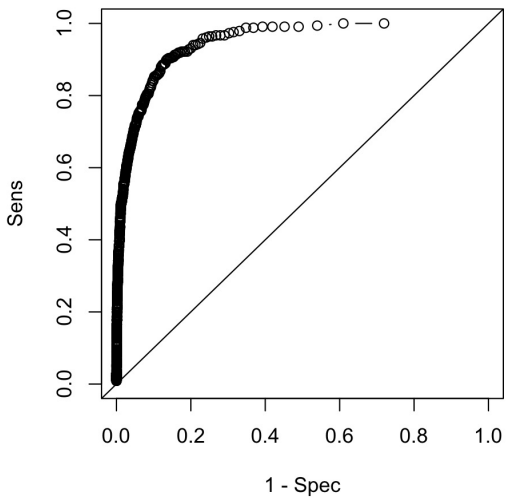
```
###--- Linear Discriminant Analysis on all data
library(MASS)
lda.fit=lda(default ~ student + balance,  data=Default)
lda.fit
plot(lda.fit)

###--- Get fitted Response
lda.pred=predict(lda.fit)

###--- Assign Yes/No to fitted response using threshold of .2
lda.pred2 <- ifelse(lda.pred$posterior[,2]>.2, "Yes", "No")
CM <- confusionMatrix(factor(lda.pred2), factor(default), positive="Yes")
CM
```

```
###--- Use roc function to see ROC and AUC
library(pROC)

plot.roc(default, lda.pred$posterior[,2], levels=c("No", "Yes"))
abline(v=.9757, h=.5856, col="red")   #  point out threshold=.2 case

auc(default, lda.pred$posterior[,2], levels=c("No", "Yes"))

#  Area under the curve: 0.9496
```

```
###--- Do the same by hand to check (Optional)
#     Calculate results for many threshold at once

Cutoff <- seq(.001, .9, .001)     # <- don't go too far, like (0 to 1)
Sens <- Spec <- 0
for (i in 1:length(Cutoff)) {

      # True/False prediction for each cutoff point
      pred01 = ifelse( lda.pred$posterior[,2]>Cutoff[i], "Yes", "No" )

      CM <- confusionMatrix(factor(pred01), factor(default), positive="Yes")
      Sens[i] <- CM$byClass["Sensitivity"]
      Spec[i] <- CM$byClass["Specificity"]
}

plot(1-Spec, Sens, type="b", xlim=c(0,1), ylim=c(0,1))
abline(a=0, b=1)
```

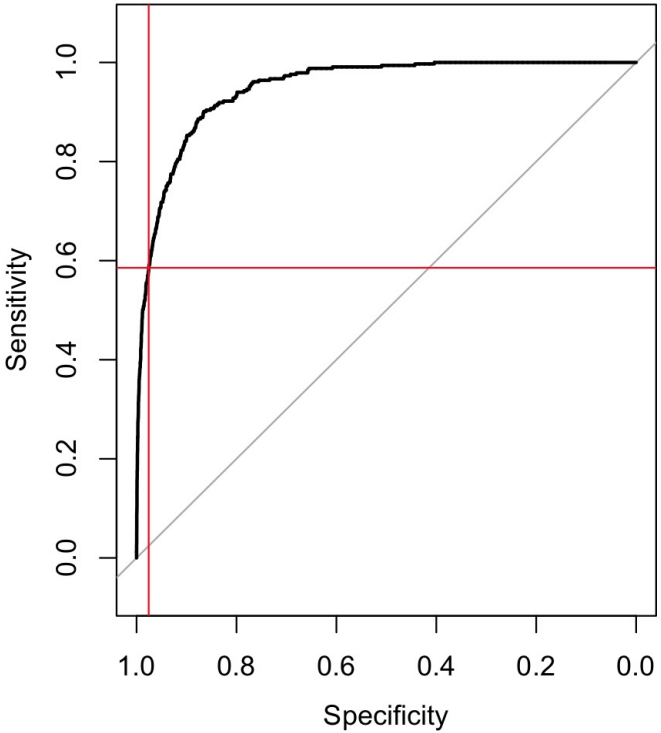
```
abline(v=1-.9757, h=.5856, col="red")   #  point out threshold=.2 case
```

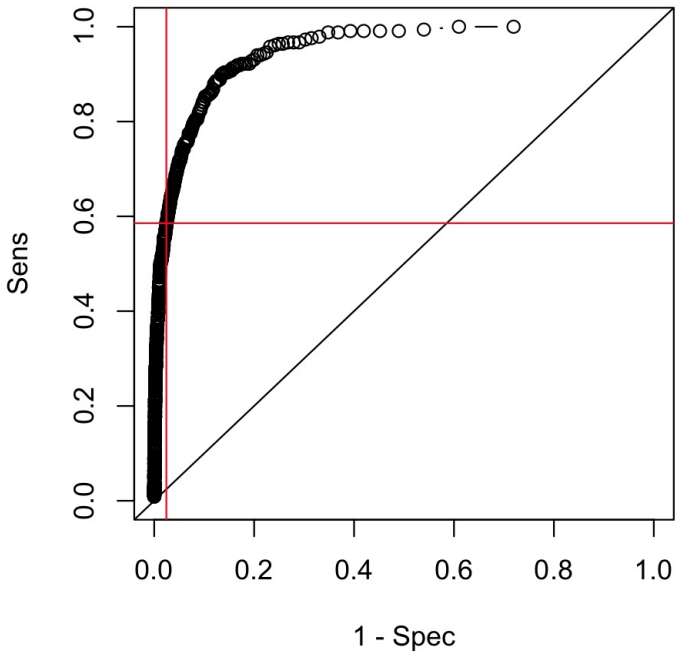




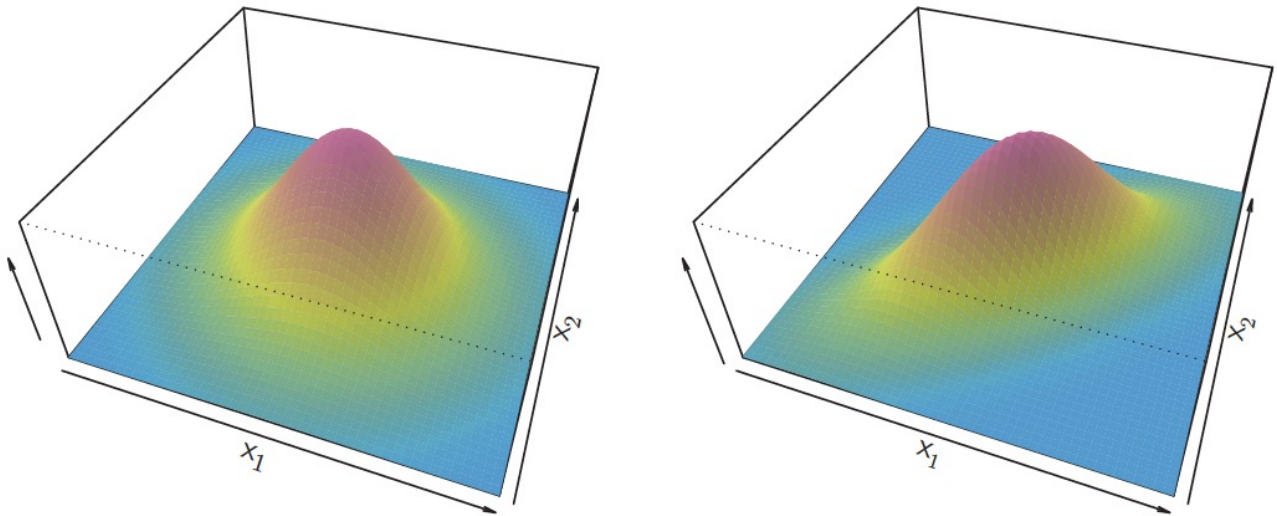

```
Area under the curve: 0.9496
    Sensitivity :    % of Yes caught           Pos Pred Value : % of Yes.pred correct
    Specificity :    % of No  caught           Neg Pred Value : % of No.pred  correct
```

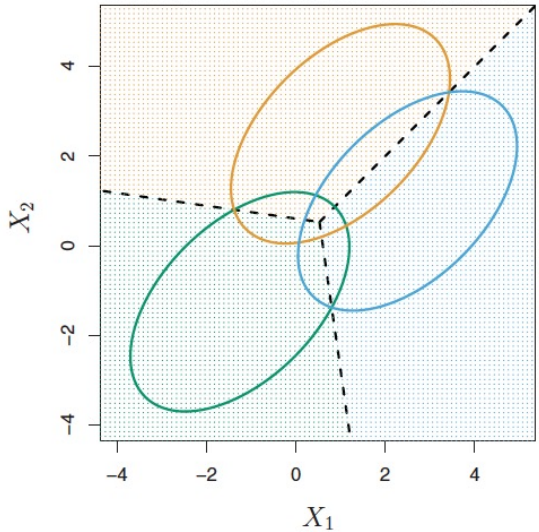
Corrected Plot
Area under the curve: 0.9496
    Sensitivity :   % of Yes caught   Pos Pred Value : % of Yes.pred correct
    Specificity :   % of No caught   Neg Pred Value : % of No.pred correct

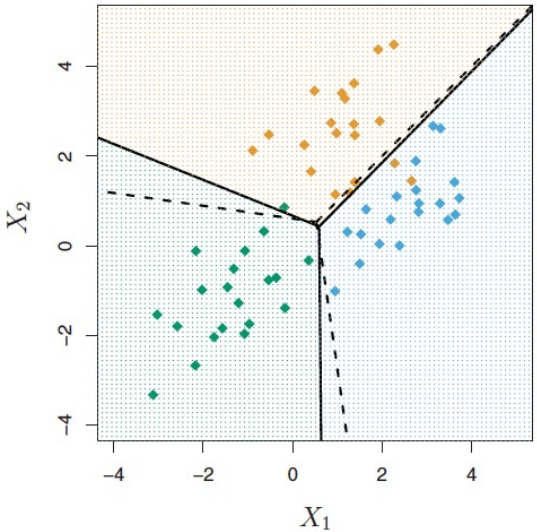

Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.8
0.6
0.4
0.2
0.0
Specificity


Sens
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
1 - Spec

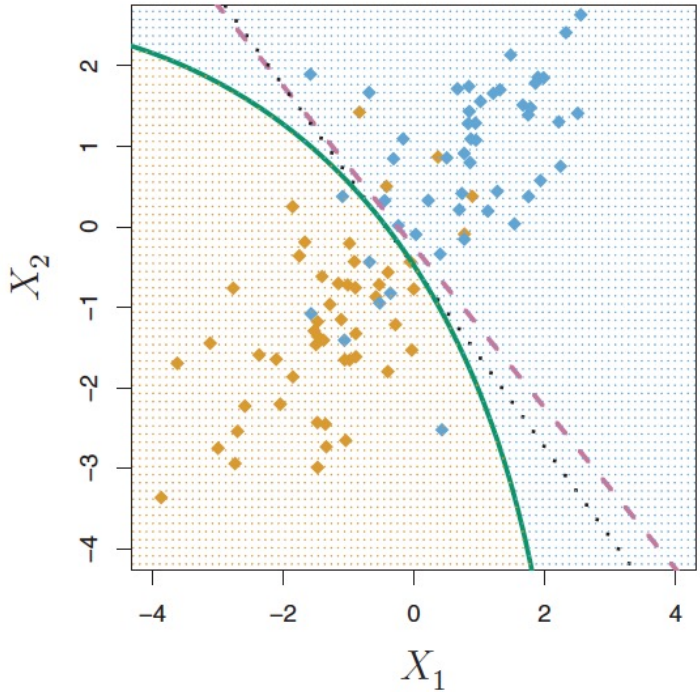
B.15 When there is more than 1 predictor

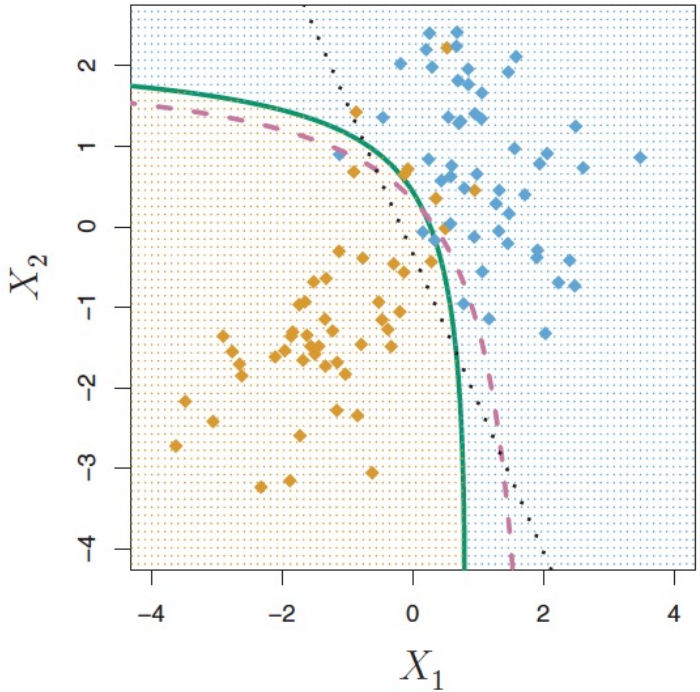



X_2
X_1
4
2
0
−2
−4
−4
−2
0
2
4


X_2
X_1
4
2
0
−2
−4
−4
−2
0
2
4

B.16 Quadratic Discriminant Analysis


X_1
X_2
−4
−2
0
2
4
−4
−3
−2
−1
0
1
2


X_1
X_2
−4
−2
0
2
4
−4
−3
−2
−1
0
1
2