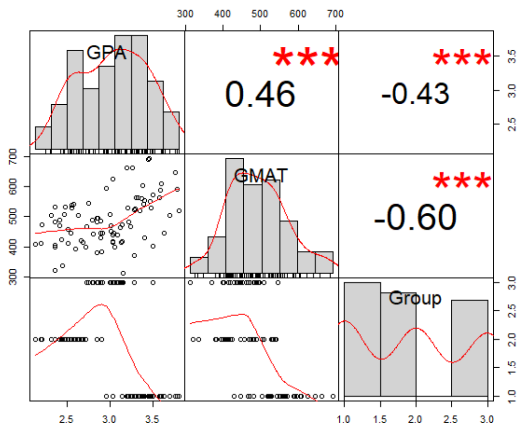
STAT 4360 Mini Project 3

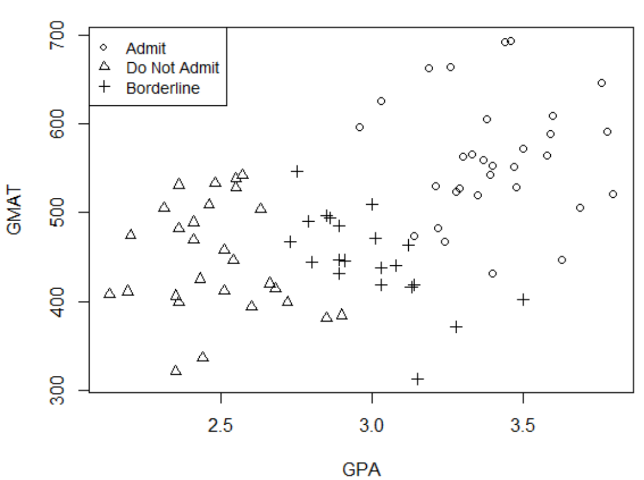
Name: Jaemin Lee

Section 1: Answers to the specific questions asked

1. a) Exploratory Data Analysis

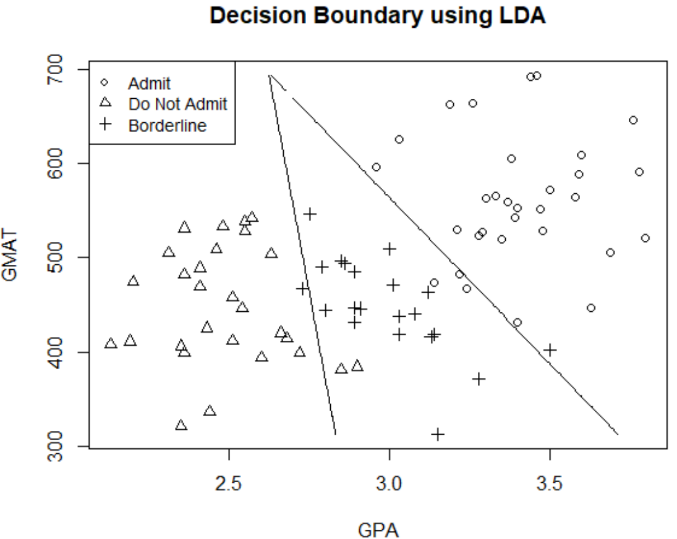
*Figure 1: Scatterplots/correlation matrix of the dataset*

The scatterplots and correlation matrix in Figure 1 represent that both GPA and GMAT have strong association with student’s graduate business school admission status.

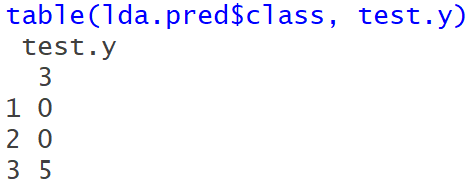


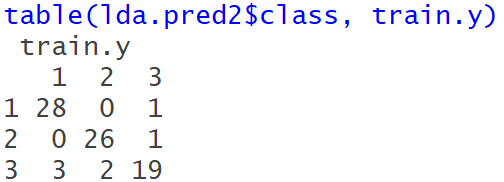
*Figure 2: Scatterplot of GPA and GMAT*

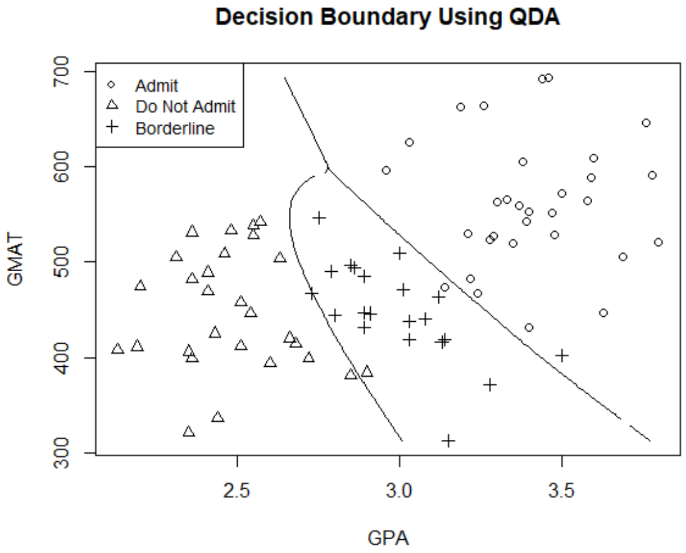
Figure 2 shows how students’ GPA and GMAT score affect their admission status. It shows that the higher the GPA (3.0 or above) and GMAT (450 or above), the more likely they get admitted.

b)

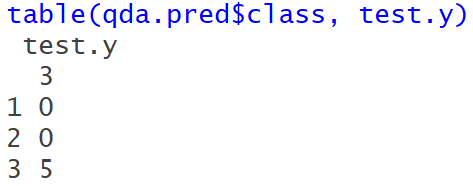
The plot above shows the decision boundary using Linear Discriminant Analysis (LDA). Two decision boundaries that separates three groups seem sensible as they well capture each group fairly accurately.



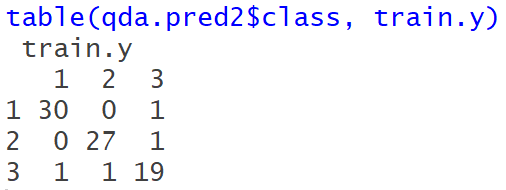
Above is the confusion matrix for test data using LDA. There is no misclassification on test data. Thus, the misclassification rate is 0 %.

Above is the confusion matrix for training data using LDA. It shows that 7 students are misclassified. 28 students are correctly classified as group 1 (admitted), 26 students are correctly classified as group 2 (borderline), and 19 students are correctly classified as group 3 (not admitted). The misclassification rate turned out to be 8.75%.

c)

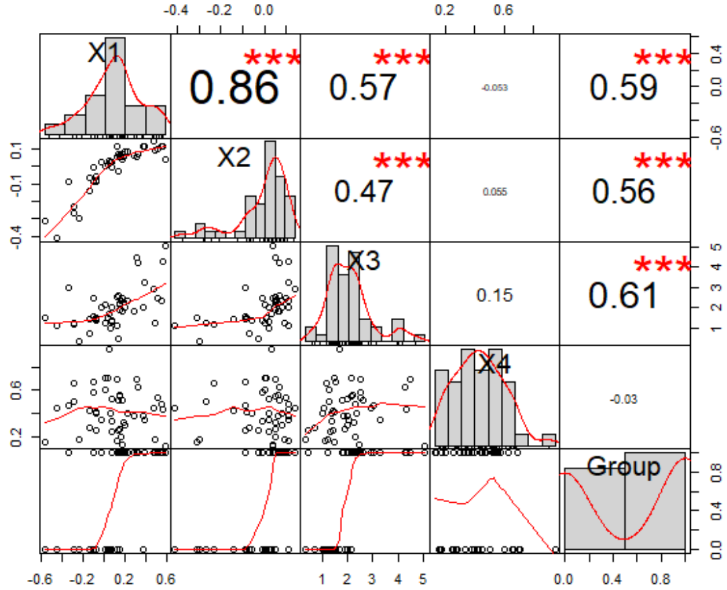
The plot above shows the decision boundary using Quadratic Discriminant Analysis (QDA). Two decision boundaries that separate three groups seem sensible as they well capture each group fairly accurately.

Above is the confusion matrix for test data using QDA. There is no misclassification just like LDA’s. Thus, the misclassification rate is 0%.



Above is the confusion matrix for training data using QDA. It shows that 4 students are misclassified. 30 students were correctly classified as group 1, 27 students were correctly classified as group 2, and 19 students are correctly classified as group 3. The misclassification rate turned out to be 5%.

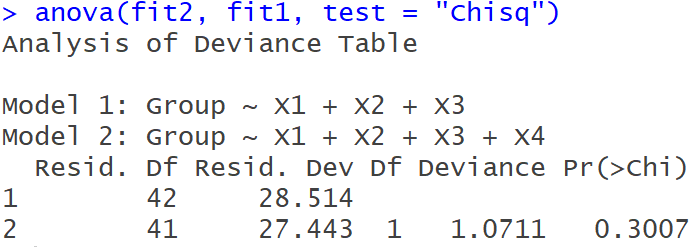
d) Notice the number of misclassifications on LDA was 7, but it went down to 4 on QDA. Also, notice the difference in misclassification rate between LDA and QDA. LDA’s was 8.75%, whereas QDA’s was 5%. This tells us that QDA performs better than LDA and it is a more reliable model. The decision boundary of QDA is slightly better than that of LDA as well. Therefore, I recommend QDA rather than LDA.

2. a) Exploratory Data Analysis

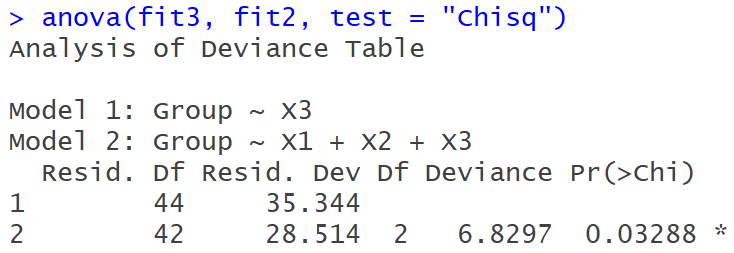
*Figure 3: Scatterplots/Correlation Matrix of the dataset*

Figure 3 represents that predictors X1, X2, and X3 are significant in determining whether the firms are bankrupt or not. Column 6 and 7 in the dataset which represent predictors X and X.1 are removed as they were null values.

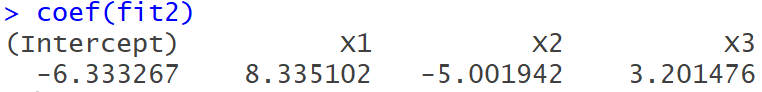
b) Fit1 represents the logistic regression model with all 4 variables (X1, X2, X3, and X4). P-value for X3 is 0.00455 which is statistically significant. Fit2 still shows that p-value for X3 is statistically significant. I did Chi2 test to compare fit1 and fit2 using ANOVA.



P-value for model 2 is 0.30 which means that it’s not significant. Thus, this tells us that X4 can be dropped. This also goes with the observation from the scatter plots in part (a). However, according to the summary of fit2, X1 and X2 weren’t statistically significant. To fully investigate whether they are significant predictors or not, I compared fit2 to fit3, which only contains X3 as a predictor.

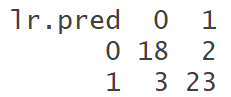


Interestingly, this shows that predictors X1 and X2 are actually significant as the p-value for model 2 is 0.033. Therefore, my final model is fit2.

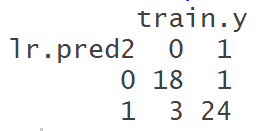


Above shows the estimated coefficients for final model. A positive coefficient indicates that as the value of the independent variable increases, the mean of the dependent variable also tends to increase. A negative coefficient suggests that as the independent variable increases, the mean dependent variable tends to decrease. Therefore, according to fit2, if X1 increases by 1 unit, the average rate of being classified as bankrupt increases by the factor of 8.335. If X2 increases by 1 unit, then the average rate of being classified as bankrupt decreases by the factor of 5.0019. If X3 increases by 1 unit, then the average rate of being classified as bankrupt increases by the factor of 3.2.

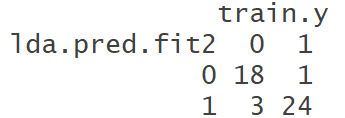
3. a) Equation for the decision boundary: Beta0 + (Beta1 \* X1) + (Beta2 \* X2) + (Beta3 \* X3) = 0, implying that -6.333267 + (8.335102 \* X1) + (-5.001942 \* X2) + (3.201476 \* X3) = 0.



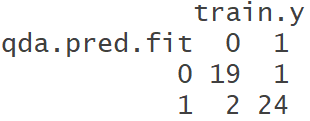
Above is the confusion matrix of logistic regression model (fit2) with X1, X2, and X3. The misclassification rate is 5/46 = 0.108 (10.8%). Sensitivity = P(predicted response = 0| true response = 0), which is 18/21 = 0.857. Sepcificity = P(predicted response = 1| true response = 1) which is 23/25 = 0.92. AUC of ROC curve is 0.9352. (ROC curve plot will be posted with other models in part (e))

b) Equation for decision boundary: -5.319513 + (7.137804 \* X1) + (-3.703330 \* X2) + (3.414834 \* X3) + (-2.968390 \* X4) = 0

Above is the confusion matrix of logistic regression model (fit1) with X1, X2, X3, and X4. The misclassification rate is 4/ 46 = 008695 (8.7%). Sensitivity is 18/21 = 0.857. Specificity is 24/25 = 0.96. AUC of ROC curve is 0.941. (ROC curve plot will be posted with other models in part (e))

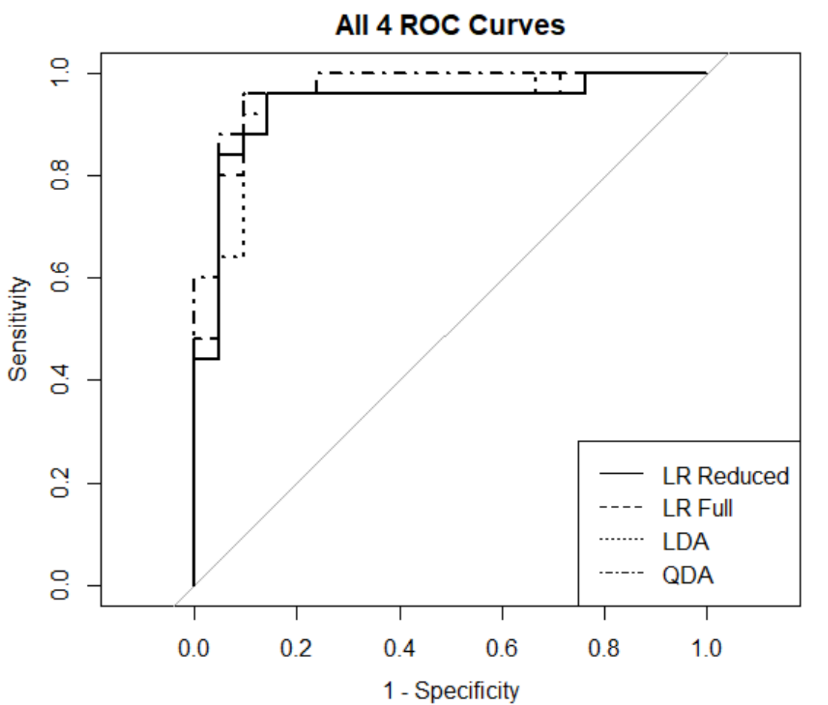
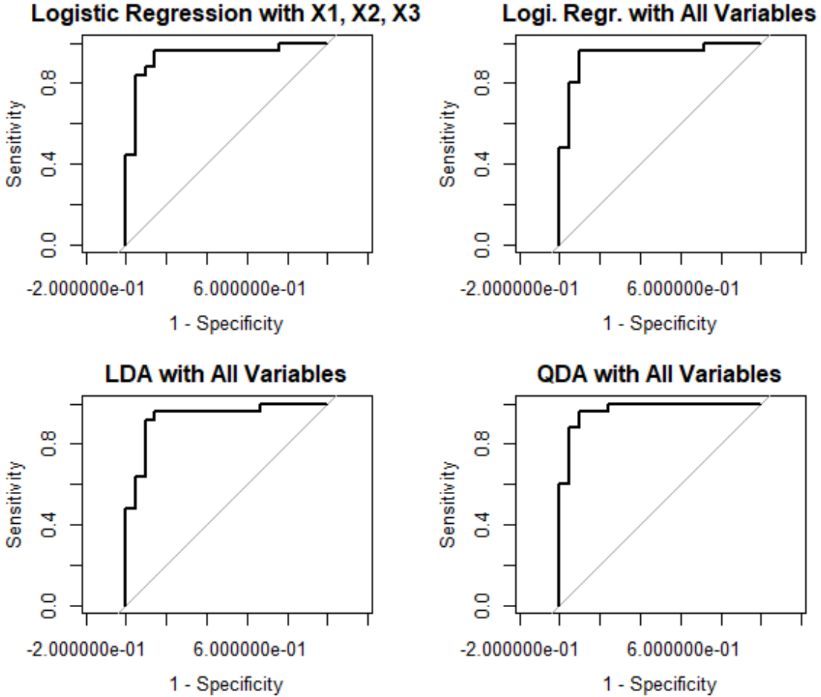
c) Equation for decision boundary: -5.319513 + (7.137804 \* X1) + (-3.703330 \* X2) + (3.414834 \* X3) + (-2.968390 \* X4) = 0 (this is the same as logistic regression, because LDA produces the same results as logistic regression)

Above is the confusion matrix for LDA. Notice it produces the same confusion matrix as logistic regression. This follows what we discussed in class. Therefore, the misclassification rate, sensitivity and specificity are the same as logistic regression’s (misclassification rate = 8.69%, Sensitivity = 18/21, and specificity = 24/25). However, strangely, AUC was off by a little bit. It turned out to be 0.9333, whereas logistic regression’s was 0.94. I did an intense research on why this is happening, but unfortunately, couldn’t figure out the cause of slight discrepancy. (ROC curve plot will be posted with other models in part (e))

d) Getting an equation of decision boundary for QDA is harder as R doesn’t provide coefficients for the fit unlike other methods. However, one can find the decision boundary by using the brute force approach just like we did in question 1.

Above is the confusion matrix for QDA. The misclassification rate is 3/46 = 0.65 (6.5%). Sensitivity is 19/21 = 0.90. Specificity is 24/25 = 0.96. AUC of ROC curve is 0.9695. (ROC curve plot will be posted with other models in part (e))

e) Comparing all the methods used in (a) – (e), it looks like QDA has the least misclassification rate (6.5%) and its AUC is the highest (0.9695). AUC curve represents the probability that the ‘+’ subject is more likely to be classified as ‘+’ than the ‘-’ subject. Thus, the closer to 1, the better. Therefore, I would recommend using QDA for this particular data. And below are the ROC curve plots that were required in part (a) – (d). The one on the left are 2 by 2 graphs and the one on the right is a plot that combines all 4 ROC curves.



Section 2: R Code

########## Question 1 (a) Exploratory Analysis ##########

admission = read.csv("C:/Users/jaemi/Desktop/STAT 4360/Projects/Project 3/admission.csv")

head(admission);str(admission) # 85 observations and 3 variables

# 1-80 are training data and 81-85 are test data

adm = data.frame(admission)

chart.Correlation(adm) # both GPA and GMAT have strong association in determining student's admission status

# training data (row 1-80 from col 1,2,3)

train = adm[1:80, ]

train.x = adm[1:80, 1:2]

train.y = adm[1:80, 3]

train.id <- logical(85) # creates a logical vector of the specified length.

# each element of the vector is equal to FALSE

train.id[1:80] <- TRUE # set 1:80 (train data) to be true

# test data (row 81-85 from col 1, 2, 3)

test.x = adm[81:85, 1:2]

test.y = adm[81:85, 3]

# plot GPA vs GMAT using training data

plot(adm[1:80, 1], adm[1:80, 2], xlab = "GPA", ylab = "GMAT", pch = admission$Group)

legend("topleft", legend = c("Admit", "Do Not Admit", "Borderline"), pch = c(1, 2, 3), cex = 0.9)

# strong association between GPA and GMAT (higher the GPA, higher the GMAT)

# also, notice that GPA and GMAT have association with students' admission status

######### Question 1 (b) #######

## lda using the training data

library(MASS)

lda.train <- lda(Group ~ GPA + GMAT, data = adm, subset = train.id)

lda.train

# get predictions for test data

lda.pred = predict(lda.train, adm[!train.id, ])

# confusion matrix for test data

table(lda.pred$class, test.y) # there is no misclassification

# error rate for test data

mean(lda.pred$class != test.y) # 0 %

# get predictions for train data

lda.pred2 = predict(lda.train, adm[train.id, ])

length(train.y)

# confusion matrix for train data

table(lda.pred2$class, train.y)

# correctly identified 28 students for group 1

# " ----------------" 26 students for group 2

# " ----------------" 19 students for group 3

# 7 students are misclassified

# error rate for train data

mean(lda.pred2$class != train.y) # 8.75 %

## Decision boundary

# Set up a dense grid and compute posterior prob on the grid

n.grid = 100

x1.grid = seq(f = min(train.x[, 1]), t = max(train.x[, 1]), l = n.grid)

x2.grid = seq(f = min(train.x[, 2]), t = max(train.x[, 2]), l = n.grid)

grid = expand.grid(x1.grid, x2.grid)

colnames(grid) = colnames(train.x)

pred.grid = predict(lda.train, grid)

# p\*(x) for class boundaries

p1star = pred.grid$posterior[,1] - pmax(pred.grid$posterior[,2], pred.grid$posterior[,3])

p2star = pred.grid$posterior[,2] - pmax(pred.grid$posterior[,1], pred.grid$posterior[,3])

prob1 = matrix(p1star, nrow = n.grid, ncol = n.grid, byrow = F)

prob2 = matrix(p2star, nrow = n.grid, ncol = n.grid, byrow = F)

plot(train.x, pch = train.y, main = "Decision Boundary using LDA")

contour(x1.grid, x2.grid, prob1, levels = 0, labels = "", xlab = "", ylab = "", main = "", add = T)

contour(x1.grid, x2.grid, prob2, levels = 0, labels = "", xlab = "", ylab = "", main = "", add = T)

legend("topleft", legend = c("Admit", "Do Not Admit", "Borderline"), pch = c(1, 2, 3), cex = 0.9)

############### Question 1 (C) #############################

# fit qda

qda.train = qda(Group ~ GPA + GMAT, data = adm, subset = train.id)

qda.train

# get predictions for test data

qda.pred = predict(qda.train, adm[!train.id,])

test.y

# confusion matrix for test data

table(qda.pred$class, test.y) # no missclassification

# error rate for test data

mean(qda.pred$class != test.y) # 0 %

# get predictions for train data

qda.pred2 = predict(qda.train, adm[train.id,])

# confusion matrix for train data

table(qda.pred2$class, train.y)

# 4 misclassifications. Notice the difference from LDA (7 misclassifications)

# accuracy rate for train data

mean(qda.pred2$class != train.y) # 5 % -> lower than LDA (QDA performs better)

## Decision boundary

# set up a dense grid and compute posterior probability on the grid

n.grid = 100

x1.grid = seq(f = min(train.x[, 1]), t = max(train.x[, 1]), l = n.grid)

x2.grid = seq(f = min(train.x[, 2]), t = max(train.x[, 2]), l = n.grid)

grid = expand.grid(x1.grid, x2.grid)

colnames(grid) = colnames(train.x)

qda.pred.grid = predict(qda.train, grid)

# p\*(x) for class boundaries

p1star.qda = qda.pred.grid$posterior[,1] - pmax(qda.pred.grid$posterior[,2], qda.pred.grid$posterior[,3])

p2star.qda = qda.pred.grid$posterior[,2] - pmax(qda.pred.grid$posterior[,1], qda.pred.grid$posterior[,3])

prob1.qda = matrix(p1star.qda, nrow = n.grid, ncol = n.grid, byrow = F)

prob2.qda = matrix(p2star.qda, nrow = n.grid, ncol = n.grid, byrow = F)

plot(train.x, pch = train.y, main = 'Decision Boundary Using QDA')

contour(x1.grid, x2.grid, prob1.qda, levels = 0, labels = "", xlab = "", ylab = "", main = "", add = T)

contour(x1.grid, x2.grid, prob2.qda, levels = 0, labels = "", xlab = "", ylab = "", main = "", add = T)

legend("topleft", legend = c("Admit", "Do Not Admit", "Borderline"), pch = c(1, 2, 3), cex = 0.9)

######## Question 1 (D) ########

library(pROC)

#### Question 2

########### (A) exploratory analysis ##############

bank = read.csv("C:/Users/jaemi/Desktop/STAT 4360/Projects/Project 3/bankruptcy.csv")

head(bank);str(bank) # 47 observations and 7 variables

summary(bank)

# remove columns that have null values (column 6 and 7)

bank = bank[, 1:5]

head(bank)

library(PerformanceAnalytics)

chart.Correlation(bank)

# looks like x1, x2 and x3 are significant variables in determining whether the person is bankrupt or not

# train data including the 4th variable

train.x = bank[,1:4]

# train data excluding the 4th variable

train.x2 = bank[, 1:3]

train.y = as.factor(bank[,5])

# relevel 0 and 1 as 0 is '+' response and 1 is '-' response

train.y = relevel(train.y, ref = '1')

train$Group = as.factor(train$Group)

train$Group = relevel(train$Group, ref = '1')

################ Question 2 (B) ################

# logistic regression

fit1 = glm(Group ~ X1 + X2 + X3 + X4, family = binomial, data = bank)

summary(fit1)

fit2 = glm(Group ~ X1 + X2 + X3, family = binomial, data = bank)

summary(fit2)

anova(fit2, fit1, test = "Chisq")

# p-value is 0.30 > 0.05 which means that X4 can be dropped as it is not significant

fit3 = glm(Group ~ X3, family = binomial, data = bank)

summary(fit3)

anova(fit3, fit2, test = "Chisq")

# p-value is 0.032 which is less than 0.05. This tells us that those dropped variables (X1, X2) are

# actaully significant. Therefore, our final model will contain X1, X2, and X3 as predictors

# final logistic regression model is fit2

# interpretation of estimated regression coefficients

coef(fit2)

############ Question 3 (A) ###############

# equation for the decision boundary

# B0 + (B1 \* X1) + (B2 \* X2) + (B3 \* X3) = 0

# implying that X3 = -(B0/B3) -(B1\*x1)/B3 - (B2\*X2)/B3 is the eq of the line

# confusion matrix

# estimated probabilties for train data

lr.prob = predict(fit2, data = bank, type = "response")

# predicted classes (using 0.5 cutoff)

# '+' = 0 (bankrupt) & '-' = 1 (non-bankrupt)

lr.pred = ifelse(lr.prob >= 0.5, 1, 0)

# train error rate

1 - mean(lr.pred == train.y) # 10.86 %

# confusion matrix and (sensitivity, specificity)

table(lr.pred, bank[, 5]) # 5 misclassifications

# sensitivity = P(predicted response = 0| true response = 0)

sensitivity = 18/21

# 0.8571429

# sepcificity = P(predicted response = 1| true response = 1)

specificity = 23/25

# 0.92

# ROC curve

# case = '1' , control = '0'

roc.lr = roc(train.y, lr.prob, levels = c(0, 1))

roc.lr

# AUC = 0.9352

# plot the ROC curve

plot(roc.lr, legacy.axes = T)

############ Question 3 (B) ###############

# decision boundary equation

coef(fit1)

# -5.319513 + (7.137804 \* X1) + (-3.703330 \* X2) + (3.414834 \* X3) + (-2.968390 \* X4) = 0

# estimated probabilties for train data

lr.prob2 = predict(fit1, data = bank, type = "response")

lr.prob2

# predicted classes (using 0.5 cutoff)

# '+' = 1 & '-' = 0

lr.pred2 = ifelse(lr.prob2 >= 0.5, 1, 0)

# train error rate

1 - mean(lr.pred2 == train.y) # 8.69 %

# confusion matrix and (sensitivity, specificity)

table(lr.pred2, train.y) # 4 misclassifications

# classification is better when predicting 1

# however predicting 0 didn't change

# sensitivity = P(predicted response = 0| true response = 0)

sensitivity = 18/21

# 0.8571429

# sepcificity = P(predicted response = 1| true response = 1)

specificity = 24/25

# 0.96

#### note: using all predictors, sensitivity went up by 4 %, whereas specificity didn't change at all

# ROC curve

# case = '1' , control = '0'

roc.lr2 = roc(train.y, lr.prob2, levels = c(0, 1))

roc.lr2

# AUC = 0.941

# plot the ROC curve

plot(roc.lr2, legacy.axes = T)

############ Question 3 (C) ###############

lda.fit2 = lda(train.y ~ ., data = train.x)

# decision boundary equation (same as Logistic Regression)

# -5.319513 + (7.137804 \* X1) + (-3.703330 \* X2) + (3.414834 \* X3) + (-2.968390 \* X4) = 0

# estimated probabilties for train data

lda.prob.fit2 = predict(lda.fit2, data = bank)

# predicted classes (using 0.5 cutoff)

lda.pred.fit2 = ifelse(lda.prob.fit2$posterior[,2] >= 0.5, 1,0)

lda.prob.fit2$posterior

# train error rate

1 - mean(lda.pred.fit2 == train.y) # 8.69 % - notice it is the same as the logistic regression error rate

# confusion matrix and (sensitivity, specificity)

table(lda.pred.fit2, train.y) # 4 misclassifications

# misclassification is also the same as logistic regression

# which follows what we learned in class

# sensitivity = P(predicted response = 0| true response = 0)

sensitivity = 18/21

# 0.8571429

# sepcificity = P(predicted response = 1| true response = 1)

specificity = 24/25

# 0.96

# ROC curve

# case = '1' , control = '0'

roc.lda = roc(train.y, lda.prob.fit2$posterior[,2], levels = c(0,1))

roc.lda

# AUC = 0.941

# plot the ROC curve

plot(roc.lda, legacy.axes = T)

############ Question 3 (D) ###############

qda.fit = qda(train.y ~ ., data = train.x)

# estimated probabilties for train data

qda.prob.fit = predict(qda.fit, data = bank)

# predicted classes (using 0.5 cutoff)

qda.pred.fit = ifelse(qda.prob.fit$posterior[,2] >= 0.5, 1,0)

# train error rate

1 - mean(qda.pred.fit == train.y) # 6.52 % - notice it is the same as the logistic regression error rate

# confusion matrix and (sensitivity, specificity)

table(qda.pred.fit, train.y) # 3 misclassifications

# sensitivity = P(predicted response = 1| true response = 1)

sensitivity = 19/21

# 0.9047619

# sepcificity = P(predicted response = 0| true response = 0)

specificity = 24/25

# 0.96

# ROC curve

# case = '1' , control = '0'

roc.qda = roc(train.y, qda.prob.fit$posterior[,2], levels = c(0, 1))

roc.qda

# AUC = 0.9695

# plot the ROC curve

plot(roc.qda, legacy.axes = T)

############ Queestion 3 (E) #############

# compare the misclassification rates of all 4

par(mfrow = c(2,2))

plot(roc.lr, legacy.axes = T, main = "Logistic Regression with X1, X2, X3")

plot(roc.lr2, legacy.axes = T, main = "Logi. Regr. with All Variables")

plot(roc.lda, legacy.axes = T, main = "LDA with All Variables")

plot(roc.qda, legacy.axes = T, main = "QDA with All Variables")

par(mfrow = c(1,1))

plot(roc.lr, legacy.axes = T, main = "All 4 ROC Curves")

plot(roc.lr2, add = T, lty = 2)

plot(roc.lda, add = T, lty = 3)

plot(roc.qda, add = T, lty = 4)

legend("bottomright",legend = c("LR Reduced", "LR Full", "LDA", "QDA"),lty = c(1, 2, 3, 4))