**SECTION 1 Problem 3**

1. Sensitivity, Specificity and overall misclassification rate were calculated using the confusion matrix for LDA as follows.

> table(LDA.pred$class, Default)

Default

**0 1**

**0 615 138**

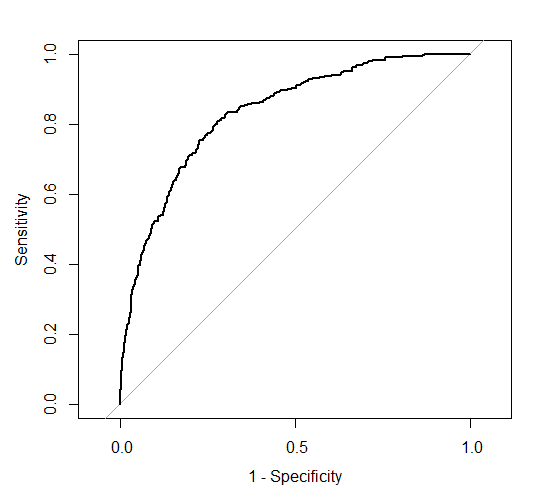
**1 85 162**

sensitivity = 162/(138+162) = 162/300 = **0.54**

specificity = 615/(615+85) = 615/700 = **0.8785714**

mc.rate = (85+138)/(615+85+138+162) = 223/1000 = **0.223**

We observe that specificity is much higher than sensitivity and sensitivity value is close to 0.5. Which suggests that LDA performed well classifying class 1 but did poorly on class 0. ROC curve lie in between the 45-degree line and the inverted L shape we are looking for.



1. Sensitivity, Specificity and overall misclassification rate were calculated using the confusion matrix for QDA as follows.

> table(QDA.pred$class, Default)

Default

**0 1**

**0 593 70**

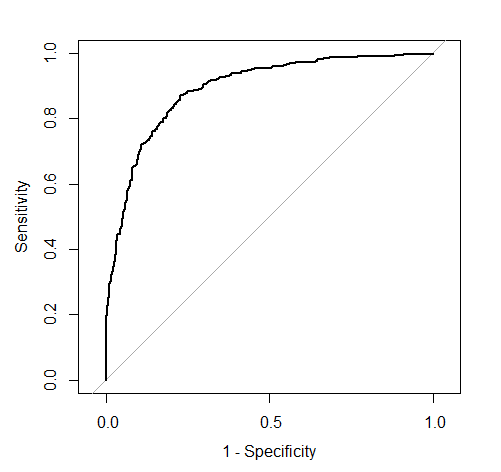
**1 107 230**

sensitivity = 230/(70+230) = 230/300 = **0.7666667**

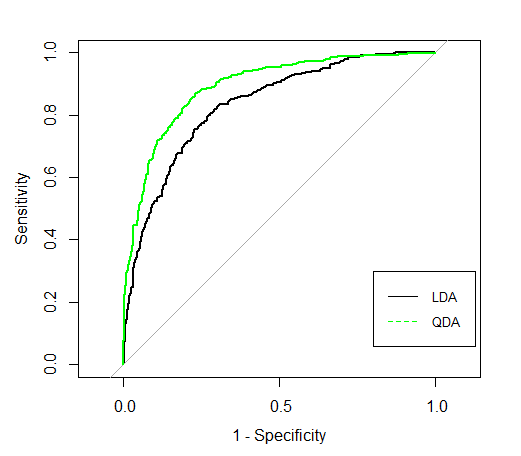
specificity = 593/(593+107) = 593/700 = **0.8471429**

mc.rate = (107+70)/(593+107+70+230) = 177/1000 = **0.177**

We observe that both the specificity and sensitivity are quite high and the overall misclassification rate is low. The model seems to be doing a good job at classification. This is further evident by the ROC curve. The shape of the ROC curve is quite close to an inverted L shape, which is the shape that we would expected under an ideal classifier.



1. When we compare the two methods, we see that QDA was a good classifier for this data set since the misclassification rate was low for QDA and even the ROC curves show that QDA curve shape is much more close to an inverted L shape. Thus, I recommend QDA classifier for this data set.



**SECTION 2 problem 3**

# Import data

credit.data <- read.csv(file.choose(), header = T)

attach(credit.data)

# Seperate data into respond and prodictors

train.y <- credit.data[,1]

train.x <- credit.data[,2:21]

# a)

library(MASS) # Need for LDA and QDA

# Fit LDA for the training data

LDA.fit <- lda(Default ~. , data = credit.data)

# Predictions for training data

LDA.pred <- predict(LDA.fit, credit.data)

# Confusion matrix for LDA

table(LDA.pred$class, Default)

# sensitivity = 162/(138+162) = 162/300

# specificity = 615/(615+85) = 615/700

# mc.rate = (85+138)/(615+85+138+162) = 223/1000

# install.packages("pROC")

library(pROC) # Need for roc

# Calculating and Plotting ROC curve for LDA

roc.LDA <- roc(train.y, LDA.pred$posterior[, 1], levels = c(0, 1))

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

# b)

# Fit QDA for the training data

QDA.fit <- qda(Default ~. , data = credit.data)

# Predictions for traning data

QDA.pred <- predict(QDA.fit, credit.data)

# Confusion matrix for QDA

table(QDA.pred$class, Default)

# sensitivity = 230/(70+230) = 230/300

# specificity = 593/(593+107) = 593/700

# mc.rate = (107+70)/(593+107+70+230) = 177/1000

# Calculating and Plotting ROC curve for QDA

roc.QDA <- roc(train.y, QDA.pred$posterior[, 1], levels = c(0, 1))

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

# c)

# Plotting ROC curve for LDA and QDA

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

plot(roc.QDA, add = T, col = "green")

legend(0.2, 0.3, legend=c("LDA", "QDA"), col=c("black", "green"), lty=1:2, cex=0.8)