Homework 3

Hyungkyu Lim

1) (10 points) Using the Boston data set (ISLR package), fit classification models in order to predict whether a given suburb has a crime rate above or below the median. Explore logistic regression, LDA and kNN models using various subsets of the predictors. Describe your findings.

```
DataBoston <- Boston

med_crime <- median(DataBoston$crim)

DataBoston$crim[DataBoston$crim < med_crime] <- 0

DataBoston$crim[DataBoston$crim > med_crime] <- 1

Boston.cor = cor(DataBoston)

set.seed(123)

train <- sample(1:nrow(DataBoston), .65*nrow(DataBoston))
crime_train <- DataBoston[train,]
crime_test <- DataBoston[-train,]

y_true_train <- as.numeric(crime_train$crim)
y_true_test <- as.numeric(crime_test$crim)

#Rogistic Regression
glm.fit<- glm(crim ~ indus+nox+rad+tax+dis+age, data = crime_train, family = "binomial")


# Predict
glm.probs.train <- predict(glm.fit, newdata = crime_train[,-1], type = "response")
glm.y_hat_train <- round(glm.probs.train)
glm.probs.test <- predict(glm.fit, newdata = crime_test[,-1], type = "response")
glm.y_hat_test <- round(glm.probs.test)

# Calculate the error rates
train_err <- sum(abs(glm.y_hat_train- y_true_train))/length(y_true_train)

train_err
```

```
## [1] 0.1219512
```

```
table(crime_test$crim,glm.y_hat_test)
```

```
##    glm.y_hat_test
##      0  1
##   0 77 13
##   1 15 73
```

```
round(mean(glm.y_hat_test!=crime_test$crim),3)
```

```
## [1] 0.157
```

```r
#LDA
Boston.lda.fit <- lda(crim~indus+nox+dis+rad+tax+age, data = crime_train)
Boston.pred.train <- predict(Boston.lda.fit, newdata = crime_train)
lda.y_hat_train <- as.numeric(Boston.pred.train$class)-1
Boston.pred.test <- predict(Boston.lda.fit, newdata = crime_test)
lda.y_hat_test <- as.numeric(Boston.pred.test$class)-1


# Compute the error
lda_train_error <- sum(abs(y_true_train - lda.y_hat_train))/length(y_true_train)

lda_train_error
```

```
## [1] 0.1402439
```

```r
table(crime_test$crim,lda.y_hat_test)
```

```
##    lda.y_hat_test
##      0  1
##   0 85  5
##   1 26 62
```

```r
round(mean(lda.y_hat_test!=crime_test$crim),3)
```

```
## [1] 0.174
```

```r
#Knn

subset.variable<-DataBoston[,c(1,3,5,7,8,9,10)]

scaled.variable<-scale(subset.variable[,-1])

set.seed(123)

subset_knn <- sample(nrow(subset.variable), nrow(subset.variable) * 0.65)
knn_train = subset.variable[subset_knn, ]
knn_test = subset.variable[-subset_knn, ]

set.seed(123)


knn_1<-knn(knn_train[,-1],knn_test[,-1],knn_train[,1],k=1)
table(knn_test[,1],knn_1)
```

```
##    knn_1
##      0  1
##   0 78 12
##   1  8 80
```

```r
round(mean(knn_1 != subset.variable[,1]),3)
```

```
## Warning in `!=.default`(knn_1, subset.variable[, 1]): longer object length
## is not a multiple of shorter object length
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```

```
## [1] 0.457
```

```r
round(mean(knn_1 != subset.variable[,1]),3)
```

```
## Warning in `!=.default`(knn_1, subset.variable[, 1]): longer object length
## is not a multiple of shorter object length

## Warning in `!=.default`(knn_1, subset.variable[, 1]): longer object length
## is not a multiple of shorter object length
```

```
## [1] 0.457
```

```r
knn_5<-knn(knn_train[,-1],knn_test[,-1],knn_train[,1],k=5)
table(knn_test[,1],knn_5)
```

```
##    knn_5
##      0  1
##   0 82  8
##   1  8 80
```

```r
round(mean(knn_5 != subset.variable[,1]),3)
```

```
## Warning in `!=.default`(knn_5, subset.variable[, 1]): longer object length
## is not a multiple of shorter object length

## Warning in `!=.default`(knn_5, subset.variable[, 1]): longer object length
## is not a multiple of shorter object length
```

```
## [1] 0.464
```

```r
knn_10<-knn(knn_train[,-1],knn_test[,-1],knn_train[,1],k=10)
table(knn_test[,1],knn_10)
```

```
##    knn_10
##      0  1
##   0 79 11
##   1  8 80
```

```r
round(mean(knn_10 != subset.variable[,1]),3)
```

```
## Warning in `!=.default`(knn_10, subset.variable[, 1]): longer object length
## is not a multiple of shorter object length

## Warning in `!=.default`(knn_10, subset.variable[, 1]): longer object length
## is not a multiple of shorter object length
```

```
## [1] 0.462
```

```r
knn_20<-knn(knn_train[,-1],knn_test[,-1],knn_train[,1],k=20)
table(knn_test[,1],knn_20)
```

```
##    knn_20
##      0  1
##   0 77 13
##   1 12 76
```

```r
round(mean(knn_20 != subset.variable[,1]),3)
```

```
## Warning in `!=.default`(knn_20, subset.variable[, 1]): longer object length
## is not a multiple of shorter object length

## Warning in `!=.default`(knn_20, subset.variable[, 1]): longer object length
```

```
## is not a multiple of shorter object length

## [1] 0.472
```

2) (10 points) Download the diabetes data set (http://astro.temple.edu/~alan/DiabetesAndrews36_1.txt). Disregard the first three columns. The fourth column is the observation number, and the next five columns are the variables (glucose.area, insulin.area, SSPG, relative.weight, and fasting.plasma.glucose). The final column is the class number. Assume the population prior probabilities are estimated using the relative frequencies of the classes in the data. (Note: this data can also be found in the MMST library)

(a) Produce pairwise scatterplots for all five variables, with different symbols or colors representing the three different classes. Do you see any evidence that the classes may have difference covariance matrices? That they may not be multivariate normal?

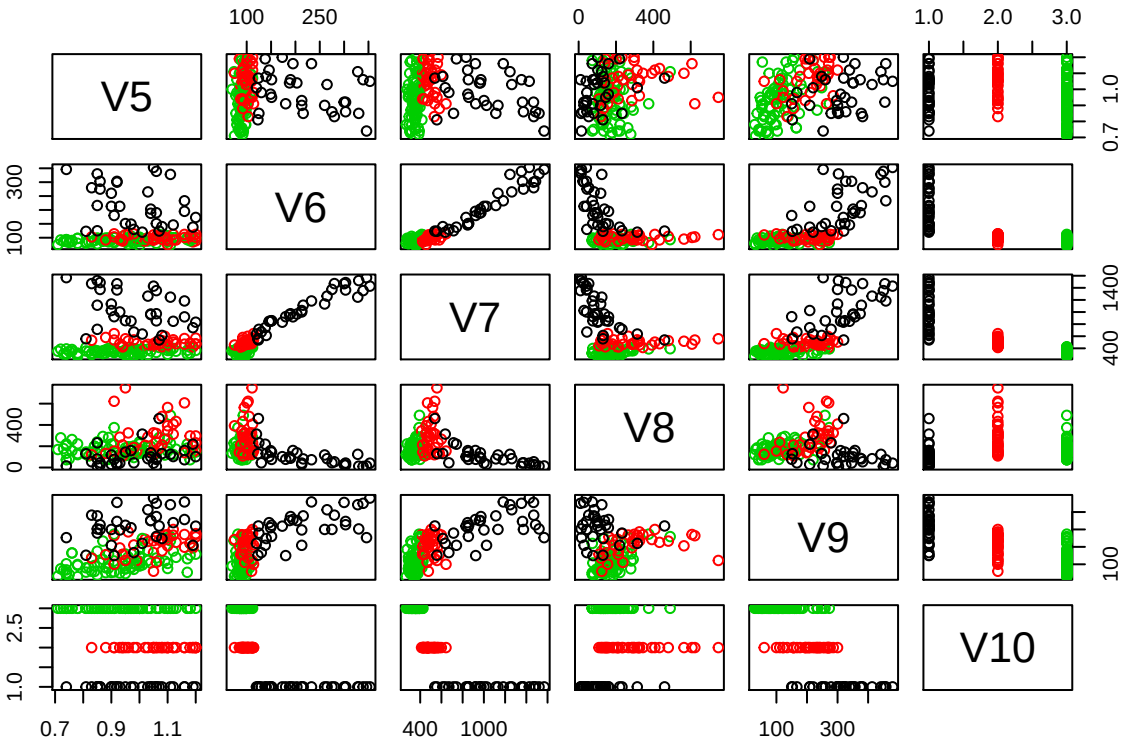
(b) Apply linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA). How does the performance of QDA compare to that of LDA in this case?

(c) Suppose an individual has (glucose area = 0.98, insulin area =122, SSPG =

544. Relative weight = 186, fasting plasma glucose = 184). To which class does LDA assign this individual? To which class does QDA?

```
DataDiabetes <- read.table("diabetes.txt")
Diabetes <- as.data.frame(DataDiabetes[,5:10])

#a
pairs(Diabetes, col=Diabetes$V10)
```




```
set.seed(123)
diabetes.train <- sample(1:nrow(Diabetes), .65*nrow(Diabetes))

diabetes_train <- Diabetes[diabetes.train,]
diabetes_test <- Diabetes[-diabetes.train,]
```

```
diabetes_true_train <- diabetes_train$V10
diabetes_true_test <-  diabetes_test$V10


#b
#LDA

diabetes.lda.fit <- lda(V10~., data = diabetes_train)
diabetes.pred.train <- predict(diabetes.lda.fit, newdata = diabetes_train)
diabetes_y_hat_train <- as.numeric(diabetes.pred.train$class)
diabetes.pred.test <- predict(diabetes.lda.fit, newdata = diabetes_test)
diabetes_y_hat_test <- as.numeric(diabetes.pred.test$class)

#Compute the error,LDA
diabetes_train_error <- sum(abs(diabetes_true_train - diabetes_y_hat_train))/length(diabetes_true_train

diabetes_train_error
```

```
## [1] 0.09574468
```

```
table(diabetes_y_hat_test,diabetes_true_test)
```

```
##                    diabetes_true_test
## diabetes_y_hat_test  1  2  3
##                   1  6  0  0
##                   2  1 12  1
##                   3  1  3 27
```

```
round(mean(diabetes_y_hat_test!= diabetes_test$V10),3)
```

```
## [1] 0.118
```

```
#QDA

diabetes.qda.fit <- qda(V10 ~., data = diabetes_train)
diabetes.qda.pred.train = predict(diabetes.qda.fit, newdata = diabetes_train)
diabetes_qda_y_hat_train <-as.numeric(diabetes.qda.pred.train$class)
diabetes.qda.pred.test = predict(diabetes.qda.fit, newdata = diabetes_test)
diabetes_qda_y_hat_test <- as.numeric(diabetes.qda.pred.test$class)

# Compute the error,QDA
diabetes_qda_train_error <- sum(abs(diabetes_true_train-diabetes_qda_y_hat_train))/length(diabetes_true_

diabetes_qda_train_error
```

```
## [1] 0.03191489
```

```
table(diabetes_true_test,diabetes_qda_y_hat_test)
```

```
##                    diabetes_qda_y_hat_test
## diabetes_true_test  1  2  3
##                   1  7  1  0
##                   2  2 10  3
##                   3  0  1 27
```

```
round(mean(diabetes_qda_y_hat_test!= diabetes_test$V10),3)
```

```
## [1] 0.137
```

```
#c

newData <- as.data.frame(t(c(0.98,122,544,186,184)))

names(newData) <- names(diabetes_train[,-6])

new.lda.pred <- predict(diabetes.lda.fit, newdata = newData)
new.lda.pred.value <- as.numeric(new.lda.pred$class)
new.lda.pred.value
```

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

```
new.qda.pred <- predict(diabetes.qda.fit, newdata = newData)
new.qda.pred.value <- as.numeric(new.qda.pred$class)
new.qda.pred.value
```

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