HW6 Question 6

Philip Lee

10/11/2019

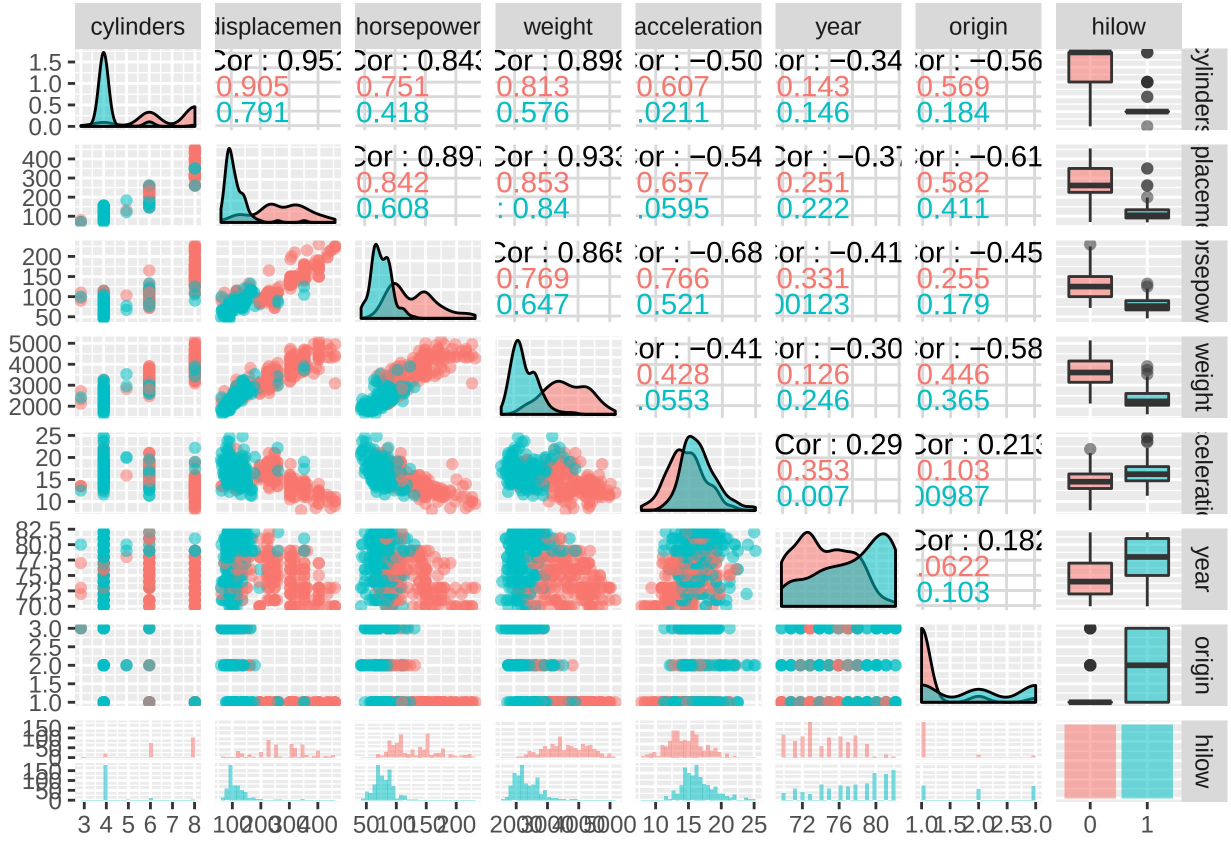
##6

#a

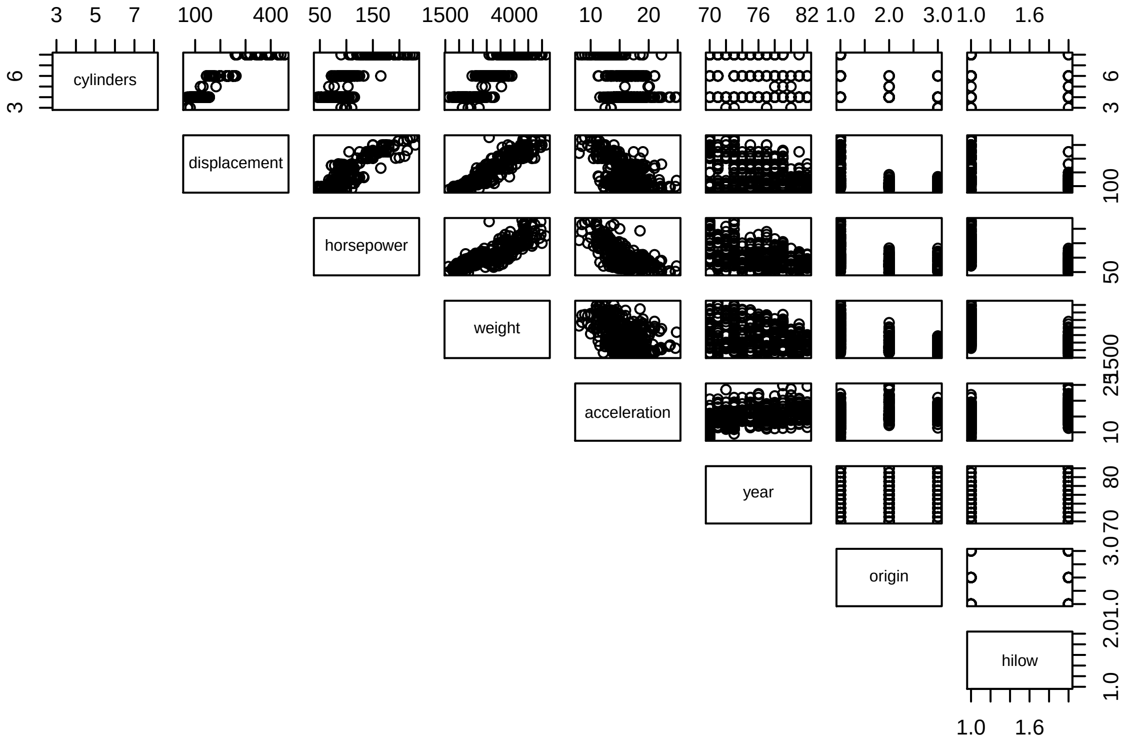
```
library(ISLR)  ##loads ISLR
Auto<-Auto  ##stores Auto data into Audo
attach(Auto)     ##attaches Auto
medmpg<-median(mpg)  ##finds the median mpg
hilow<-ifelse(Auto$mpg > medmpg,1,0) ##makes binary variable hilow
Auto$hilow<-as.factor(hilow)  ##converts hilow as a factor and creates a new column called hilow
#in the data
```

#b

```
library(GGally) ##loads a package called GGally that has a function called ggpairs
ggpairs(Auto[c(-1,-9)], aes(colour = hilow, alpha = 0.4))  ##used to get the boxplots between hilow
```



```
#and other variables
pairs(Auto[c(-1,-9)],lower.panel = NULL)  ##used to get the scatterplots between hilow
```



```
#and other variables
```

#c

Predictors: cylinders displacement horsepower weight acceleration year will be chosen as the predictors. Variable origin is left out as it is a categorical variable, which we do not use for classification. Variable mpg is also not used since it has a perfect correlation with hilow. Although cylinders in the scatterplot looks very discrete, almost to the point of being a categorical variable, the boxplot between hilow and cylinders indicate that there is a noticeable different between the low and high gas mileage vehicles. Hence, we will use cylinders as one of our predictors. Other variables will be used as predictors as they are continuous variables.

#d

```r
Auto<-Auto[c(-1,-9)]
set.seed(199) ##sets seed
sample.data<-sample.int(nrow(Auto), floor(.50*nrow(Auto)), replace = F)  ##splits the data in half
train<-Auto[sample.data, ]     ##gets the training set
test<-Auto[-sample.data, ]  ##gets the test set
library(MASS) ##for lda function

##lda predictions on training data
lda.auto <- lda(hilow ~ cylinders+displacement+weight+acceleration+year+horsepower, train)##performs
##LDA on the training data
lda.train <- predict(lda.auto) ##predicts if vehicles in the training set have high or low gas mileage
trainlda<-table(train$hilow,lda.train$class) ##Produces the confusion matrix
OERldaTrain<-sum(trainlda[c(2,3)])/sum(trainlda)  ##calculates overall error rate by summing up the
##false positive and false negative numbers and dividing them by the total sample size

##lda predictions on test data.
lda.test <- predict(lda.auto,test) ## predicts if vehicles in the test set have high or low gas mileage
testlda<-table(test$hilow,lda.test$class) ##produces confusion matrix
OERldaTest<-sum(testlda[c(2,3)])/sum(testlda) ##gets the overall error rate

##qda predicitons on training data, same thing as LDA but with QDA function instead of LDA
qda.auto<-qda(hilow~cylinders+displacement+weight+acceleration+year+horsepower,train)
qda.train<-predict(qda.auto)
trainqda<-table(train$hilow,qda.train$class)
OERtrainQda<-sum(trainqda[c(2,3)])/sum(trainqda)

##lda predictions on test data
qda.test<- predict(qda.auto,test)
testqda<-table(test$hilow,qda.test$class)
OERtestQda<-sum(testqda[c(2,3)])/sum(testqda)

trainlda ##confusion matrix for training LDA
```

```
##
##      0  1
##   0 94 13
##   1  3 86
```

```r
OERldaTrain ##overall error rate for LDA on training set
```

```
## [1] 0.08163265
```

```r
testlda ##confusion matrix for test LDA
```

```
##
##       0   1
##   0  75  14
##   1   5 102
```

```r
OERldaTest ##overall error rate for LDA on test set
```

```
## [1] 0.09693878
```

```r
trainqda ##confusion matrix for training QDA
```

```
##
```

```
##      0  1
##   0 98  9
##   1  6 83
```

```
OERtrainQda ##overall error rate for QDA on training set
```

```
## [1] 0.07653061
```

```
testqda ##confusion matrix for test QDA
```

```
##
##      0  1
##   0 75 14
##   1 10 97
```

```
OERtestQda ##overall error rate for QDA on test set
```

```
## [1] 0.122449
```

#e Overall error rates were calculated by summing up the false positive and false negative numbers in the confusion matrix and dividing the sum by the total sample size The overall error rate for both LDA and QDA are very similar in value. For the training set, the QDA has the lower overall error rate compared to LDA's (0.0765 for QDA, 0.0816 for LDA). When looking soely at the training set, then QDA is preferable over LDA due to its lower overall error rate. For the test set, the LDA has the lower overall error rate compared to QDA's (0.0969 for LDA, 0.1224 for QDA). Because LDA has lower overall error rate for the test set, LDA is preferable overQDA in the test set. This indicates that QDA is overfitting for this analysis due to its flexibility. Since the OER of the test set is higher for the QDA, this indicates that QDA is not adequate for this data split for classification and LDA performs better for the split data.

#f

```
set.seed(199)
library(MASS) ##for lda
LDAtrainOER<-NULL ##makes an empty vector to store all 1000 overall error rates for each model
LDAtestOER<-NULL
QDAtrainOER<-NULL
QDAtestOER<-NULL
for (i in 1:1000){ ##does part d 1000 times
  sample.data<-sample.int(nrow(Auto), floor(.50*nrow(Auto)), replace = F)
  train<-Auto[sample.data, ]
  test<-Auto[-sample.data, ]
  lda.auto <- lda(hilow ~ cylinders+displacement+weight+acceleration+year+horsepower, train)
  lda.train <- predict(lda.auto)
  trainlda<-table(train$hilow,lda.train$class)
  LDAtrainOER[i]<-sum(trainlda[c(2,3)])/sum(trainlda)
  lda.test <- predict(lda.auto,test)
  testlda<-table(test$hilow,lda.test$class)
  LDAtestOER[i]<-sum(testlda[c(2,3)])/sum(testlda)
  qda.auto<-qda(hilow~cylinders+displacement+weight+acceleration+year+horsepower,train)
  qda.train<-predict(qda.auto)
  trainqda<-table(train$hilow,qda.train$class)
  QDAtrainOER[i]<-sum(trainqda[c(2,3)])/sum(trainqda)
  qda.test<- predict(qda.auto,test)
  testqda<-table(test$hilow,qda.test$class)
  QDAtestOER[i]<-sum(testqda[c(2,3)])/sum(testqda)
}
mean(LDAtrainOER)  ##gets the avg OER from the loop
```

```
## [1] 0.08679082
```

```r
mean(LDAtestOER)
```

```
## [1] 0.09698469
```

```r
mean(QDAtrainOER)
```

```
## [1] 0.08562245
```

```r
mean(QDAtestOER)
```

```
## [1] 0.1012908
```

#g

Similar to the result from part d, the OER for QDA on the training set is lower than the LDA's (0.0868 for LDA and 0.0856 for QDA) and the OER for the LDA on the test set is lower than the QDA's (0.0970 for LDA and 0.1013 for QDA). Hence, we can claim that QDA performs better on the training set while LDA performs better on the test set as lower overall error rate indicates a better performance in classification.