Chapter 4 Problem 11

Andira Putri

September 19, 2018

a. Create a binary variable `mpg01` that contains a 1 if `mpg` contains a value above its median.

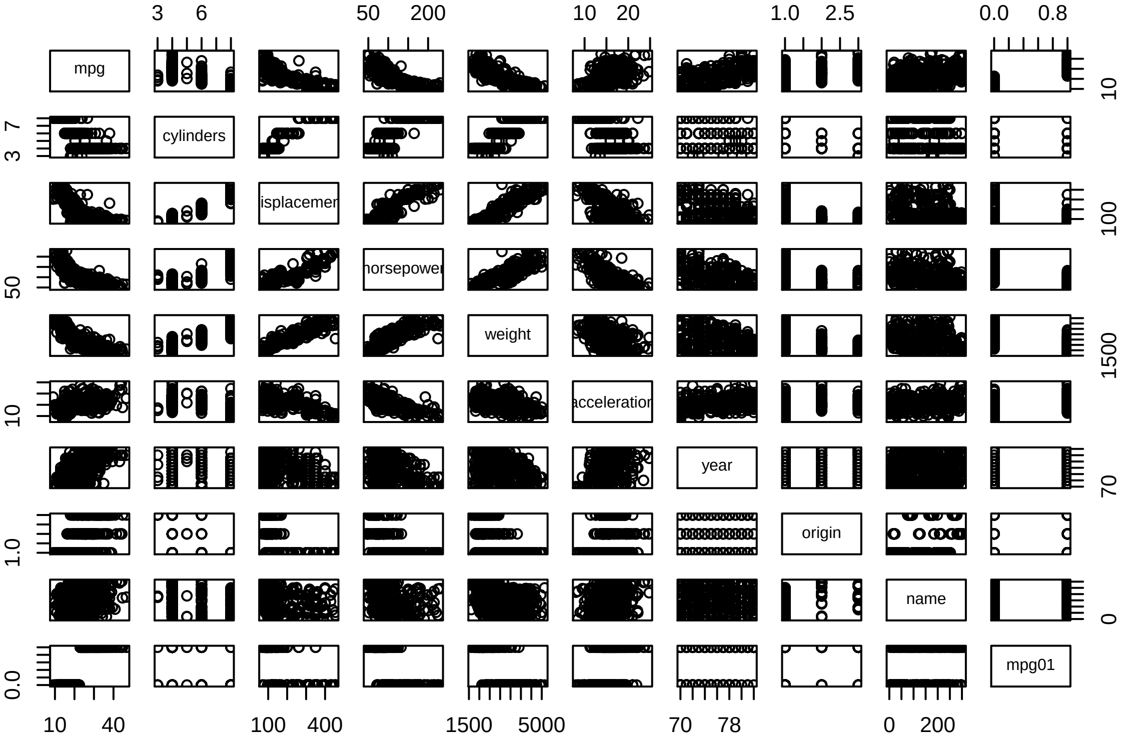
```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.5.1
```

```
data(Auto)
med=median(Auto$mpg)
mpg01=ifelse(Auto$mpg > med, 1, 0)
auto2=data.frame(Auto,mpg01)
```

b. Explore the data graphically to investigate associations between `mpg01` and other features. What features might be helpful in predicting `mpg01`?

```
pairs(auto2)
```



The features that are likely related to `mpg01` are `displacement`,`horsepower`,`weight`, and `acceleration`. These quantitative features also happen to show strong relationships with `mpg`.

c. Split the data into training set and a test set.

```
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 3.5.1
```

```
set.seed(2) #for reproducibility of results
split=sample.split(auto2,SplitRatio=0.7) #70% training
#creates vector with 70% TRUE and 30% FALSE
#vector acts like a new column in auto2 set
training=subset(auto2, split==TRUE)
test=subset(auto2, split==FALSE)
```

d. Perform LDA and give test error.

```
library(MASS)
ldafit=lda(mpg01~displacement+horsepower+weight+acceleration,data=training)
ldapred=predict(ldafit,test)
names(ldapred)
```

```
## [1] "class"     "posterior" "x"
```

```
ldaclass=ldapred$class
table(ldaclass,test$mpg01)
```

```
##
## ldaclass  0  1
##        0 50  1
##        1 11 55
```

The test error is (1+11)/117 = 0.1026.

e. Perform QDA and give test error.

```
qda.fit=qda(mpg01~displacement+horsepower+weight+acceleration,data=training)
qda.pred=predict(qda.fit,test)
names(qda.pred)
```

```
## [1] "class"     "posterior"
```

```
qda.class=qda.pred$class
table(qda.class,test$mpg01)
```

```
##
## qda.class  0  1
##         0 56  5
##         1  5 51
```

The test error is (5+5)/117 = 0.0855.

f. Perform logistic regression and give test error.

```
logreg=glm(mpg01~displacement+horsepower+weight+acceleration,data=training,family=binomial)
logprob=predict(logreg,test,type="response")
logpred=rep("0",nrow(test)) #default pred is 0
logpred[logprob>0.5]="1" #change to 1 if prob > 0.5
table(logpred,test$mpg01)
```

```
##
## logpred  0  1
##       0 54  5
##       1  7 51
```

The test error is (5+7)/117 = 0.1026.

g. Perform KNN with different values of K and give test errors. Which value seems to perform the best?

```r
library(class)
```

```
## Warning: package 'class' was built under R version 3.5.1
```

```r
train.X <- as.matrix(training$displacement,training$horsepower,training$weight,training$acceleration)
test.X <- as.matrix(test$displacement, test$horsepower,test$weight,test$acceleration)
set.seed(2)
knn.pred=knn(train.X,test.X,training$mpg01,k=1)
mean(knn.pred != test$mpg01) #calculates error rate
```

```
## [1] 0.1452991
```

```r
knn.pred2=knn(train.X,test.X,training$mpg01,k=5)
mean(knn.pred2 != test$mpg01)
```

```
## [1] 0.05982906
```

```r
knn.pred3=knn(train.X,test.X,training$mpg01,k=10)
mean(knn.pred3 != test$mpg01)
```

```
## [1] 0.09401709
```

```r
knn.pred4=knn(train.X,test.X,training$mpg01,k=20)
mean(knn.pred4 != test$mpg01)
```

```
## [1] 0.08547009
```

```r
knn.pred5=knn(train.X,test.X,training$mpg01,k=40)
mean(knn.pred5 != test$mpg01)
```

```
## [1] 0.07692308
```

```r
knn.pred6=knn(train.X,test.X,training$mpg01,k=100)
mean(knn.pred6 != test$mpg01)
```

```
## [1] 0.08547009
```

KNN performed the best when K=5.