chapter 4

Classification

06/10/2024

#chp 4(10)

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.3.3
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

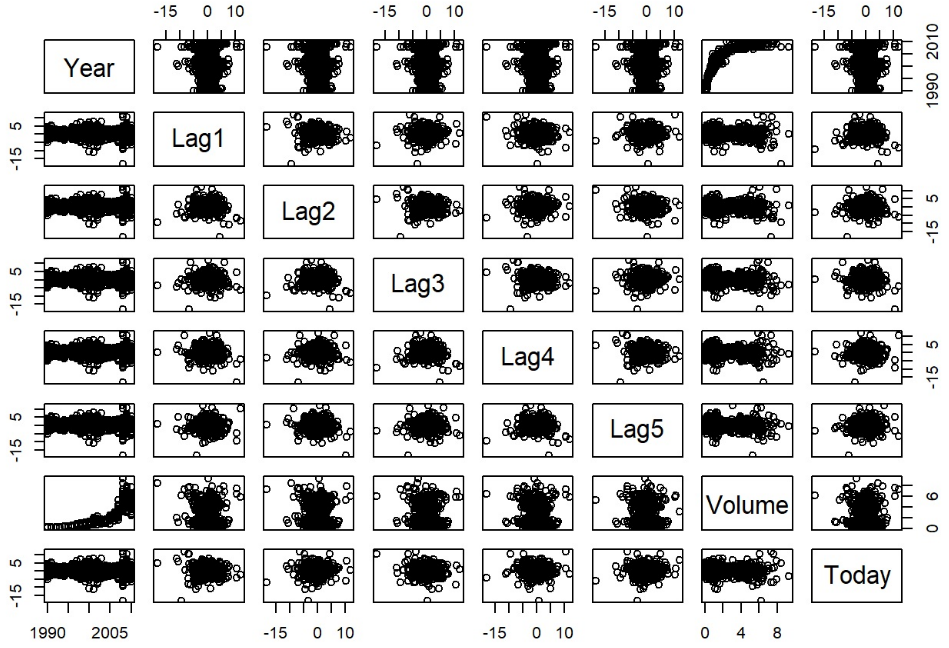
```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

10. This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

a. Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?

```
pairs(Weekly[ ,-9])
```



```
prop.table(table(Weekly$Direction))
```

```
##
##      Down        Up
## 0.4444444 0.5555556
```

b. Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

```
fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data = Weekly, family = binomial)
summary(fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

INTERPRETATION #Lag2 is statistically significant (p-value = 0.0296), with a positive coefficient (0.05844) #the probability of predicting an “Up” direction also increases #Lag1, negatively correlated with market direction, is not statistically significant (p-value = 0.1181)
(c) Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
library(class)
library(MASS)
```

```
pred = predict(fit, type = "response")
pred_d = ifelse(pred > 0.5, "Up", "Down")
```

```
conf_matrix = table(Predicted = pred_d, Actual = Weekly$Direction)
accuracy <- mean(pred_d == Weekly$Direction)
```

```
conf_matrix
```

```
##          Actual
## Predicted Down  Up
##      Down   54  48
##      Up    430 557
```

```
accuracy
```

```
## [1] 0.5610652
```

INTERPRETATION #The confusion matrix shows that the model correctly predicted the outcome about 56.1% #True Negatives (TN): 54 (Predicted “Down” and actual “Down”) #False Positives (FP): 48 (Predicted “Up” but actual “Down”) #False Negatives (FN): 430 (Predicted “Down” but actual “Up”) #True Positives (TP): 557 (Predicted “Up” and actual “Up”)

d. Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the heldout data (that is, the data from 2009 and 2010).

```
train = Weekly[Weekly$Year <= 2008, ]
test =Weekly[Weekly$Year > 2008, ]
```

```
model = glm(Direction ~ Lag2, data = train, family = binomial)
```

```
pred = ifelse(predict(model, test, type = "response") > 0.5, "Up", "Down")
```

```
conf_matrix = table(pred, test$Direction)
accuracy = mean(pred == test$Direction)
```

```
conf_matrix
```

```
##
## pred   Down Up
##   Down    9  5
##   Up     34 56
```

```
accuracy
```

```
## [1] 0.625
```

INTERPRETATION #The model has a moderate accuracy of 62.5%. #True Negatives (TN): 9 (Predicted “Down” and actual “Down”) #False Positives (FP): 5 (Predicted “Up” but actual “Down”) #False Negatives (FN): 34 (Predicted “Down” but actual “Up”) #True Positives (TP): 56 (Predicted “Up” and actual “Up”)

```
library(MASS)
```

e. Repeat (d) using LDA.

```
lda_m= lda(Direction~Lag2, data=train)
pred1=predict(lda_m,test)
table(test$Direction,pred1$class)
```

```
##
##        Down Up
##   Down    9 34
##   Up      5 56
```

```
names(pred)
```

```
##   [1] "986"  "987"  "988"  "989"  "990"  "991"  "992"  "993"  "994"  "995" 
##  [11] "996"  "997"  "998"  "999"  "1000" "1001" "1002" "1003" "1004" "1005"
##  [21] "1006" "1007" "1008" "1009" "1010" "1011" "1012" "1013" "1014" "1015"
##  [31] "1016" "1017" "1018" "1019" "1020" "1021" "1022" "1023" "1024" "1025"
##  [41] "1026" "1027" "1028" "1029" "1030" "1031" "1032" "1033" "1034" "1035"
##  [51] "1036" "1037" "1038" "1039" "1040" "1041" "1042" "1043" "1044" "1045"
##  [61] "1046" "1047" "1048" "1049" "1050" "1051" "1052" "1053" "1054" "1055"
##  [71] "1056" "1057" "1058" "1059" "1060" "1061" "1062" "1063" "1064" "1065"
##  [81] "1066" "1067" "1068" "1069" "1070" "1071" "1072" "1073" "1074" "1075"
##  [91] "1076" "1077" "1078" "1079" "1080" "1081" "1082" "1083" "1084" "1085"
## [101] "1086" "1087" "1088" "1089"
```

INTERPRETATION #The model is highly effective at predicting “Up” #it struggles with predicting “Down” correctly, with only 9 correct “Down” predictions out of 43 actual “Down” cases

f. Repeat (d) using QDA.

```
qda_m= qda(Direction~Lag2, data=train)
pred2=predict(qda_m,test)
table(test$Direction,pred2$class)
```

```
##
##        Down Up
##   Down    0 43
##   Up      0 61
```

INTERPRETATION #he overall accuracy is 58.6%, and the precision for “Up” is also 58.6% (g) Repeat (d) using KNN with K =1.

```
library(class)
```

```
tr=Weekly$Year  <= 2008
x_train=Weekly[tr,-9]
y_train=Weekly[tr,9]
x_test=Weekly[!tr,-9]
y_test=Weekly[!tr,9]
pred=knn(x_train,x_test,y_train,k=1)
table(y_test,pred)
```

```
##       pred
## y_test Down Up
##   Down   37  6
##   Up     15 46
```

INTERPRETATION #The model has an overall accuracy of 79.8%. It performs well in predicting both “Up” and “Down”

h. Which of these methods appears to provide the best results on this data?

#interpertation# Taking the target metric as the accuracy of the classifier: LDA & Logistic Regression get the same test accuracy of 0.625, so these two are tied.

i. Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier

```
tr=Smarket$Year<2005
tr_data=Smarket[tr,]
te_data=Smarket[!tr,]
fit=glm(Direction~.,data = tr_data,family = binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
probs=predict(fit, te_data,type="response")
pred=rep("Down",nrow(te_data))
pred[probs>0.5]="Up"
table(pred)
```

```
## pred
## Down   Up
##  110  142
```

```
table(pred,te_data$Direction)
```

```
##
## pred   Down  Up
##   Down  110   0
##   Up      1 141
```

INTERPRETATION #This model is highly effective and accurate #his model performs an overall accuracy of 99.6%

#chp 4(11)

11. In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set.

```
?Auto
```

```
## starting httpd help server ... done
```

a. Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median. You can compute the median using the median() function. Note you may find it helpful to use the data.frame() function to create a single data set containing both mpg01 and the other Auto variables.

```
median(Auto$mpg)
```

```
## [1] 22.75
```

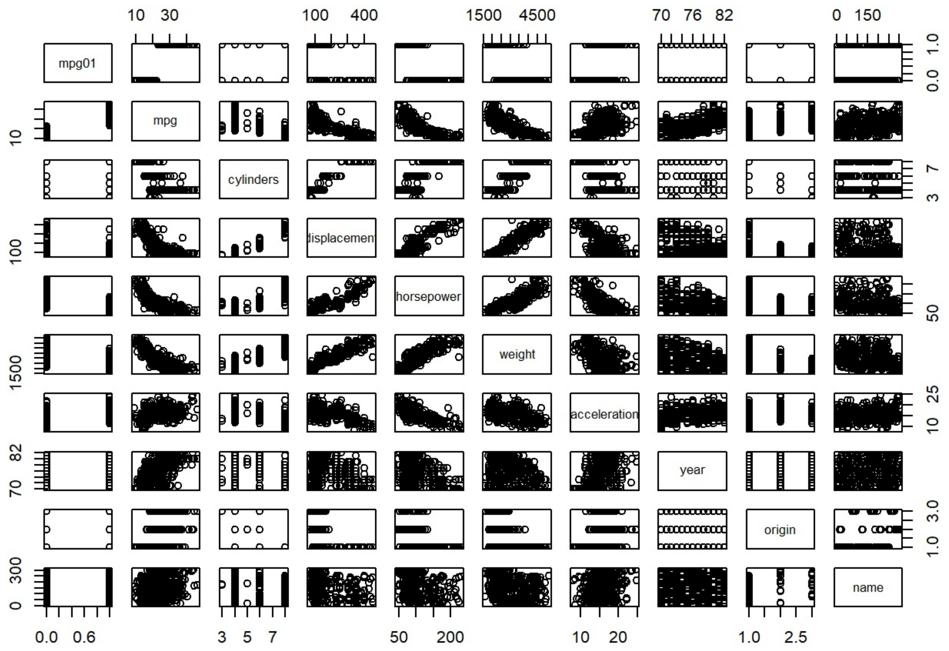
```
mpg01=rep(0,nrow(Auto))
mpg01[Auto$mpg>median(Auto$mpg)]=1
Auto_mpg01 = data.frame(mpg01,Auto)
head(Auto_mpg01)
```

```
##   mpg01 mpg cylinders displacement horsepower weight acceleration year origin
## 1     0  18         8          307        130   3504         12.0   70      1
## 2     0  15         8          350        165   3693         11.5   70      1
## 3     0  18         8          318        150   3436         11.0   70      1
## 4     0  16         8          304        150   3433         12.0   70      1
## 5     0  17         8          302        140   3449         10.5   70      1
## 6     0  15         8          429        198   4341         10.0   70      1
##                        name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

INTERPRETATION #Cars that have fuel efficiency (mpg) greater than the median mpg in the dataset #cars with mpg greater than the median, the corresponding entries in mpg01 are updated to 1

b. Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01?Scat terplots and boxplots may be useful tools to answer this question. Describe your findings.

```
pairs(mpg01~.,data = Auto)
```



INTERPRETATION #separations between the clusters of mpg01 = 0 (less fuel-efficient cars) and mpg01 = 1 (more fuel-efficient cars)

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

```
set.seed(123)
tr = sample(nrow(Auto_mpg01),nrow(Auto_mpg01)*.7)
train12 = Auto_mpg01[tr,]
test12 = Auto_mpg01[-tr,]
```

. (d) Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
library(MASS)
```

```
set.seed(123)
fit = lda(mpg01~cylinders+displacement+horsepower+weight, data = train12)
pred = predict(fit,test12)
summary(pred)
```

```
##           Length Class  Mode
## class     118    factor numeric
## posterior 236    -none- numeric
## x         118    -none- numeric
```

INTERPRETATION #a factor, this means the predictions are categorical #This component contains the posterior probabilities associated with the predictions. A length of 236

```
table(test12$mpg01,pred$class)
```

```
##
##      0  1
##   0 50 10
##   1  3 55
```

e. Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained? (

```
fit1 = qda(mpg01~cylinders+displacement+horsepower+weight,data = train12)
pred = predict(fit1,test12)
summary(pred)
```

```
##           Length Class  Mode
## class     118    factor numeric
## posterior 236    -none- numeric
```

INTERPRETATION #the model has made predictions for 118 observations #The posterior component consists of probabilities associated with the predictions for each class

```
table(test12$mpg01,pred$class)
```

```
##
##      0  1
##   0 53  7
##   1  5 53
```

INTERPRETATION #the confusion matrix indicates that the model performs well overall, with an accuracy of approximately 89.83%

f. Perform logistic regression on the training data in order to pre dict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
fit2 = glm(mpg01~ +cylinders+displacement+weight+horsepower, family = 'binomial', data = train12)
probs = predict(fit2, test12, type = "response")
pred = rep(0,nrow(test12))
pred[probs > 0.5] = 1
table(pred,test12$mpg01)
```

```
##
## pred  0  1
##    0 53  6
##    1  7 52
```

INTERPRETATION #This matrix suggests a balanced performance with good accuracy, high true positive and true negative rates, and relatively low false positives and negatives. #The model correctly predicted 0 when the true value was 0. #The model incorrectly predicted 1 when the true value was 0. #The model incorrectly predicted 0 when the true value was 1. #The model correctly predicted 1 when the true value was 1.

g. Perform KNN on the training data, with several values of K, in order to predict mpg01. Use only the variables that seemed most associated with mpg01 in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?

```
library(class)
```

```
set.seed(123)
names(Auto_mpg01)
```

```
##  [1] "mpg01"        "mpg"          "cylinders"    "displacement" "horsepower"
##  [6] "weight"       "acceleration" "year"         "origin"       "name"
```

```
tra = sample(nrow(Auto_mpg01),nrow(Auto_mpg01)*0.7)
x_train = Auto_mpg01[tra,-c(1,10,11)]
y_train = Auto_mpg01[tra,1]
x_test = Auto_mpg01[-tra,-c(1,10,11)]
y_test = Auto_mpg01[-tra,1]
```

```
fit4 = knn(x_train,x_test,y_train,k=1)
table(y_test,fit4)
```

```
##       fit4
## y_test  0  1
##      0 52  8
##      1 11 47
```

INTERPRETATION #the model correctly predicted 0 (negative class) when the true value was also 0. #the model incorrectly predicted 0 when the true value was 1. #the model incorrectly predicted 1 (positive class) when the true value was 0.