MT 3100 (Brandt)
Statistical Learning

Assignment 8

In this assignment, we continue to study the Auto data set. We will test our logistic model for FromUS, and we will also look at some LDA models.

Note on setseed()
The setseed() command allows us to control the output of random processes. In some of the exercises below, we will randomly choose training and test sets for our models. If we all use the same seeds, we should all get the same answers.

Note on naming your models
I suggest you give each of your models a different name. When I worked the exercise below, I stuck with the book's notation and naming conventions as best I could. To distinguish my models, I used glm1.fits, glm1.probs, glm1.pred, and so on, and then updated to glm2.fits, glm2.probs, etc.

1. **Most experts go with a training/test ratio somewhere between 70/30 and 90/10. Let's train a logistic model for FromUS using a training set of 300 (and thus a test set of 92). This gives us a training/test ratio of 77/23. You can randomly set up your test and training sets as follows:**

```
set.seed(0) # Choose a seed
train=sample.int(392, 300) # Randomly choose 300 of the cars.
Auto.train=Auto[train,] # The rows of Auto for the training cars.
Auto.test=Auto[-train,] # The rows of Auto for the test cars.
FromUS.test=Auto$FromUS[-train] # FromUS values for the test cars.
```

   a. **Using seeds 0, 3, and 4, find a logistic regression model for FromUS as a function of cylinders, displacement, weight, and horsepower. You can use this command**

```
glm.fits=glm(FromUS~cylinders+displacement+weight+horsepower,
data=Auto.train,family=binomial)
```

The command above is a variation of the command given at the bottom of page 175. We don't need to say what the subset is because we have set data to be the training set.

You will need to set up commands similar to those at the top of page 176 to give the confusion matrix for your models. Be patient. It may take a few tries and revisions to get your commands to work. Run your commands for each seed and complete the following table:

CODE:

```
Auto = read.csv(file="Auto.csv",head=TRUE,sep=",",stringsAsFactors=FALSE,
na.strings="?")
Auto = na.omit(Auto)
```

```
displacement=Auto$displacement
cylinders=Auto$cylinders
horsepower=Auto$horsepower
weight=Auto$weight
mpg=Auto$mpg
FromUS=rep(0,392) #Set up a vector of 392 zeroes
FromUS[Auto$origin==1]=1 #Switch value fro US cars
Auto=data.frame(Auto,FromUS) #Add new variable to Auto
set.seed(0) # Choose a seed
train=sample.int(392, 300) # Randomly choose 300 of the cars.
Auto.train=Auto[train,] # The rows of Auto for the training cars.
Auto.test=Auto[-train,] # The rows of Auto for the test cars.
FromUS.test=Auto$FromUS[-train] # FromUS values for the test cars.

glm.fits=glm(FromUS~cylinders+displacement+weight+horsepower,
data=Auto.train,family=binomial)
#seed(0)
glm.fit.seed0 <- glm(FromUS ~ cylinders + displacement + weight + horsepower, data =
Auto.train, family = binomial)
summary(glm.fit.seed0)
glm.probs.seed0 <- predict(glm.fit.seed0, Auto.test, type = "response")
glm.pred.seed0 <- rep(0,92)
glm.pred.seed0[glm.probs.seed0 > 0.5] = 1
confusion.matrix.seed0 <- table(glm.pred.seed0, FromUS.test)
confusion.matrix.seed0
#seed(3)
set.seed(3)
train <- sample.int(392, 300)
Auto.train <- Auto[train,]
Auto.test <- Auto[-train,]
FromUS.test <- Auto$FromUS[-train]
glm.fit.seed3 <- glm(FromUS ~ cylinders + displacement + weight + horsepower, data =
Auto.train, family = binomial)
summary(glm.fit.seed3)
glm.probs.seed3 <- predict(glm.fit.seed3, Auto.test, type = "response")
glm.pred.seed3 <- rep(0,92)
glm.pred.seed3[glm.probs.seed3 > 0.5] = 1
confusion.matrix.seed3 <- table(glm.pred.seed3, FromUS.test)
confusion.matrix.seed3
#seed(4)
set.seed(4)
train <- sample.int(392, 300)
Auto.train <- Auto[train,]
Auto.test <- Auto[-train,]
FromUS.test <- Auto$FromUS[-train]
glm.fit.seed4 <- glm(FromUS ~ cylinders + displacement + weight + horsepower, data =
Auto.train, family = binomial)
summary(glm.fit.seed4)
glm.probs.seed4 <- predict(glm.fit.seed4, Auto.test, type = "response")
glm.pred.seed4 <- rep(0,92)
glm.pred.seed4[glm.probs.seed4 > 0.5] <- 1
```

confusion.matrix.seed4 <- table(glm.pred.seed4, FromUS.test)
confusion.matrix.seed

Seed	Confusion Matrix	Significance for hp (enter the stars)	Success Rate
0	FromUS.test glm.pred.seed0 0 1 0 36 4 1 5 47	*	0.9021 = 90.21%
3	FromUS.test glm.pred.seed3 0 1 0 32 10 1 2 48	*	0.8695 = 86.95%
4	FromUS.test glm.pred.seed4 0 1 0 30 11 1 3 48	*	0.8478 = 84.78%

LR Model 1: FromUS~cylinders+displacement+weight+horsepower

Now we see something very important. Changing the seed changes the test set and thus changes the model and its performance! Yes, there is a lot of uncertainty in this business. But these three runs suggest that horsepower is not as significant as the other variables.

b. Remove horsepower. Using seeds 0, 3, and 4, find a logistic regression model for FromUS as a function of cylinders, displacement, and weight. From your output, complete the table below. Do the results suggest that removing horsepower was a good idea?

CODE:

```
Auto = read.csv(file="Auto.csv",head=TRUE,sep=",",stringsAsFactors=FALSE, na.strings="?")
Auto = na.omit(Auto)
displacement=Auto$displacement
cylinders=Auto$cylinders
horsepower=Auto$horsepower
weight=Auto$weight
mpg=Auto$mpg
FromUS=rep(0,392) #Set up a vector of 392 zeroes
FromUS[Auto$origin==1]=1 #Switch value fro US cars
Auto=data.frame(Auto,FromUS) #Add new variable to Auto
```

```
set.seed(0) # Choose a seed
train=sample.int(392, 300) # Randomly choose 300 of the cars.
Auto.train=Auto[train,] # The rows of Auto for the training cars.
Auto.test=Auto[-train,] # The rows of Auto for the test cars.
FromUS.test=Auto$FromUS[-train] # FromUS values for the test cars.

glm.fits=glm(FromUS~cylinders+displacement+weight+horsepower,
data=Auto.train,family=binomial)
#seed(0)
glm.fit.seed0 <- glm(FromUS ~ cylinders + displacement + weight, data = Auto.train, family =
binomial)
summary(glm.fit.seed0)
glm.probs.seed0 <- predict(glm.fit.seed0, Auto.test, type = "response")
glm.pred.seed0 <- rep(0,92)
glm.pred.seed0[glm.probs.seed0 > 0.5] = 1
confusion.matrix.seed0 <- table(glm.pred.seed0, FromUS.test)
confusion.matrix.seed0
#seed(3)
set.seed(3)
train <- sample.int(392, 300)
Auto.train <- Auto[train,]
Auto.test <- Auto[-train,]
FromUS.test <- Auto$FromUS[-train]
glm.fit.seed3 <- glm(FromUS ~ cylinders + displacement + weight, data = Auto.train, family =
binomial)
summary(glm.fit.seed3)
glm.probs.seed3 <- predict(glm.fit.seed3, Auto.test, type = "response")
glm.pred.seed3 <- rep(0,92)
glm.pred.seed3[glm.probs.seed3 > 0.5] = 1
confusion.matrix.seed3 <- table(glm.pred.seed3, FromUS.test)
confusion.matrix.seed3
#seed(4)
set.seed(4)
train <- sample.int(392, 300)
Auto.train <- Auto[train,]
Auto.test <- Auto[-train,]
FromUS.test <- Auto$FromUS[-train]
glm.fit.seed4 <- glm(FromUS ~ cylinders + displacement + weight, data = Auto.train, family =
binomial)
summary(glm.fit.seed4)
glm.probs.seed4 <- predict(glm.fit.seed4, Auto.test, type = "response")
glm.pred.seed4 <- rep(0,92)
glm.pred.seed4[glm.probs.seed4 > 0.5] <- 1
confusion.matrix.seed4 <- table(glm.pred.seed4, FromUS.test)
confusion.matrix.seed4
```

Seed	Confusion Matrix	Success Rate
0	FromUS.test glm.pred.seed0 0 1 0 38 4 1 3 47	0.9239 = 92.39%
3	FromUS.test glm.pred.seed3 0 1 0 33 9 1 1 49	0.8913 = 89.13%
4	FromUS.test glm.pred.seed4 0 1 0 32 11 1 1 48	0.8695 = 86.95%

LR Model 2: FromUS~cylinders+displacement+weight

c. Do you agree that horsepower should be removed from the model?

I agree with the fact that the variable "horsepower" should be remove from the model because the success rate in every seed increased showing that our model better predicts the output.

2. Let's find some LDA models. Set up your training and test sets as you did in Question 1. Then follow the commands on page 179. a. Find an LDA model for FromUS as a function of cylinders, displacement, weight, and horsepower. Complete the table:

a. **Find an LDA model for FromUS as a function of cylinders, displacement, weight, and horsepower. Complete the table:**

CODE:

```
Auto = read.csv(file="Auto.csv",head=TRUE,sep=",",stringsAsFactors=FALSE, na.strings="?")
Auto = na.omit(Auto)
displacement=Auto$displacement
cylinders=Auto$cylinders
horsepower=Auto$horsepower
weight=Auto$weight
mpg=Auto$mpg
FromUS=rep(0,392) #Set up a vector of 392 zeroes
FromUS[Auto$origin==1]=1 #Switch value fro US cars
Auto=data.frame(Auto,FromUS) #Add new variable to Auto
set.seed(0) # Choose a seed
```

```
train=sample.int(392, 300) # Randomly choose 300 of the cars.
Auto.train=Auto[train,] # The rows of Auto for the training cars.
Auto.test=Auto[-train,] # The rows of Auto for the test cars.
FromUS.test=Auto$FromUS[-train] # FromUS values for the test cars.

library(MASS)
ldaMod = lda(FromUS~cylinders+displacement+weight+horsepower)
ldaMod
ldaMod.pred=predict(ldaMod,Auto.test)
ldaMod.class=ldaMod.pred$class
table(ldaMod.class,FromUS.test)

set.seed(3) # Choose a seed
train=sample.int(392, 300) # Randomly choose 300 of the cars.
Auto.train=Auto[train,] # The rows of Auto for the training cars.
Auto.test=Auto[-train,] # The rows of Auto for the test cars.
FromUS.test=Auto$FromUS[-train] # FromUS values for the test cars.

library(MASS)
ldaMod = lda(FromUS~cylinders+displacement+weight+horsepower)
ldaMod
ldaMod.pred=predict(ldaMod,Auto.test)
ldaMod.class=ldaMod.pred$class
table(ldaMod.class,FromUS.test)


set.seed(4) # Choose a seed
train=sample.int(392, 300) # Randomly choose 300 of the cars.
Auto.train=Auto[train,] # The rows of Auto for the training cars.
Auto.test=Auto[-train,] # The rows of Auto for the test cars.
FromUS.test=Auto$FromUS[-train] # FromUS values for the test cars.

library(MASS)
ldaMod = lda(FromUS~cylinders+displacement+weight+horsepower)
ldaMod
ldaMod.pred=predict(ldaMod,Auto.test)
ldaMod.class=ldaMod.pred$class
table(ldaMod.class,FromUS.test)
```

Seed	Confusion Matrix	Success Rate

0	FromUS.test ldaMod.class 0 1 0 40 9 1 1 42	0.8913 = 89.13%
3	FromUS.test ldaMod.class 0 1 0 33 15 1 1 43 Call:	0.8260 = 82.60 %
4	FromUS.test ldaMod.class 0 1 0 32 16 1 1 43	0.8152 = 81.52%

LDA Model 1: FromUS~cylinders+displacement+weight+horsepower

b. **Examine the output for set.seed(4) and explain the meaning of the following quantities:**
   - **The value of 0.38 listed under Prior probabilities of groups.**
   - **The values of 4.13 and 249.33 under Group means.**

   The value 0.38 is telling us the percentage of the car that lands in the category "0", 38% percent of the car that are not FromUS.
   The value 4.13 in the group means table is telling us that car that are not from the US have 4.13 cylinders has a mean and the value 249.33 is telling us that car from the US have a mean displacement of 249.33

c. **Remove horsepower from the model and complete the table**

CODE:

```
Auto = read.csv(file="Auto.csv",head=TRUE,sep=",",stringsAsFactors=FALSE, na.strings="?")
Auto = na.omit(Auto)
displacement=Auto$displacement
cylinders=Auto$cylinders
horsepower=Auto$horsepower
weight=Auto$weight
mpg=Auto$mpg
FromUS=rep(0,392) #Set up a vector of 392 zeroes
FromUS[Auto$origin==1]=1 #Switch value fro US cars
Auto=data.frame(Auto,FromUS) #Add new variable to Auto
set.seed(0) # Choose a seed
train=sample.int(392, 300) # Randomly choose 300 of the cars.
Auto.train=Auto[train,] # The rows of Auto for the training cars.
Auto.test=Auto[-train,] # The rows of Auto for the test cars.
FromUS.test=Auto$FromUS[-train] # FromUS values for the test cars.

library(MASS)
```

```
ldaMod = lda(FromUS~cylinders+displacement+weight)
ldaMod
ldaMod.pred=predict(ldaMod,Auto.test)
ldaMod.class=ldaMod.pred$class
table(ldaMod.class,FromUS.test)

set.seed(3) # Choose a seed
train=sample.int(392, 300) # Randomly choose 300 of the cars.
Auto.train=Auto[train,] # The rows of Auto for the training cars.
Auto.test=Auto[-train,] # The rows of Auto for the test cars.
FromUS.test=Auto$FromUS[-train] # FromUS values for the test cars.

library(MASS)
ldaMod = lda(FromUS~cylinders+displacement+weight)
ldaMod
ldaMod.pred=predict(ldaMod,Auto.test)
ldaMod.class=ldaMod.pred$class
table(ldaMod.class,FromUS.test)


set.seed(4) # Choose a seed
train=sample.int(392, 300) # Randomly choose 300 of the cars.
Auto.train=Auto[train,] # The rows of Auto for the training cars.
Auto.test=Auto[-train,] # The rows of Auto for the test cars.
FromUS.test=Auto$FromUS[-train] # FromUS values for the test cars.

library(MASS)
ldaMod = lda(FromUS~cylinders+displacement+weight)
ldaMod
ldaMod.pred=predict(ldaMod,Auto.test)
ldaMod.class=ldaMod.pred$class
table(ldaMod.class,FromUS.test)
```

Seed	Confusion Matrix	Success Rate
0	FromUS.test ldaMod.class 0 1 0 41 8 1 0 43	0.9130 = 91.30 %
3	FromUS.test ldaMod.class 0 1 0 34 16 1 0 42	0.8260 = 82.60%

4	FromUS.test ldaMod.class 0 1 0 33 19 1 0 40	0.7934 = 79.34%

LDA Model 2: FromUS~cylinders+displacement+weight

d. Based on the numbers in the tables above, do you think horsepower should be removed from the model?

Overall, I do not think horsepower should be removed from the model because the success rate of two out of three seeds decreased.

2. Recall that logistic regression is mostly used for classification situations where there are just 2 classes.

One advantage of LDA is that is that it can be applied to situations with any number of classes. Set up your training and test sets as you did in Questions 1 and 2.

a. Find a 3rd LDA model that predicts origin (1=US, 2=Europe, 3=Japan) as a function of cylinders, displacement, weight, and horsepower. Complete the table:

CODE:

```
Auto = read.csv(file="Auto.csv",head=TRUE,sep=",",stringsAsFactors=FALSE, na.strings="?")
Auto = na.omit(Auto)
displacement=Auto$displacement
cylinders=Auto$cylinders
horsepower=Auto$horsepower
weight=Auto$weight
mpg=Auto$mpg
origin=Auto$origin

#FromUS=rep(0,392) #Set up a vector of 392 zeroes
#FromUS[Auto$origin==1]=1 #Switch value fro US cars
#Auto=data.frame(Auto,FromUS) #Add new variable to Auto

set.seed(0) # Choose a seed
train=sample.int(392, 300) # Randomly choose 300 of the cars.
Auto.train=Auto[train,] # The rows of Auto for the training cars.
Auto.test=Auto[-train,] # The rows of Auto for the test cars.
origin.test=Auto$origin[-train] # FromUS values for the test cars.

library(MASS)
ldaMod = lda(origin~cylinders+displacement+weight+horsepower)
ldaMod
ldaMod.pred=predict(ldaMod,Auto.test)
```

```
ldaMod.class=ldaMod.pred$class
table(ldaMod.class,origin.test)

set.seed(3) # Choose a seed
train=sample.int(392, 300) # Randomly choose 300 of the cars.
Auto.train=Auto[train,] # The rows of Auto for the training cars.
Auto.test=Auto[-train,] # The rows of Auto for the test cars.
origin.test=Auto$origin[-train] # FromUS values for the test cars.

library(MASS)
ldaMod = lda(origin~cylinders+displacement+weight+horsepower)
ldaMod
ldaMod.pred=predict(ldaMod,Auto.test)
ldaMod.class=ldaMod.pred$class
table(ldaMod.class,origin.test)


set.seed(4) # Choose a seed
train=sample.int(392, 300) # Randomly choose 300 of the cars.
Auto.train=Auto[train,] # The rows of Auto for the training cars.
Auto.test=Auto[-train,] # The rows of Auto for the test cars.
origin.test=Auto$origin[-train] # FromUS values for the test cars.

library(MASS)
ldaMod = lda(origin~cylinders+displacement+weight+horsepower)
ldaMod
ldaMod.pred=predict(ldaMod,Auto.test)
ldaMod.class=ldaMod.pred$class
table(ldaMod.class,origin.test)
```

Seed	Confusion Matrix	Success Rate
0	origin.test ldaMod.class 1 2 3 1 45 2 1 2 0 7 2 3 6 10 19	0.7717 = 77.17%
3	origin.test ldaMod.class 1 2 3 1 48 3 0 2 1 9 2 3 9 5 15	0.7826 = 78.26%

4	origin.test ldaMod.class 1 2 3 1 49 6 0 2 2 5 2 3 8 5 15	0.75 = 75%

LDA Model 3: origin~cylinders+displacement+weight

b. Examine the confusion matrix obtained above for setseed(4). Explain the meaning of the values 2 and 6 in the matrix.

Starting off with analyzing the value 6 in the matrix, it is telling us that 6 of the 16 European cars were incorrectly predicted to be from USA.
Then we can notice two values "2".
The first value reported in the first column of the matrix is telling us that 2 of the 59 USA cars were incorrectly predicted to be from Europe.
The second value found in the last column of the matrix is telling us that 2 of the 17 Japanese cars were incorrectly predicted to be from Europe.