Lab 8 STAT 627

David Saff

a) Load the data Auto.rda and install the package ISLR.

```
load("~/AU R Studio/Auto.rda")
#install.packages("ISLR")
library(ISLR)
attach(Auto)
```

Here we are loading the data and attaching the ISLR library.

b) Create a fuel consumption rating variable named Economy that will be treated as categorical based on the following info. For mpg<17 mark as Heavy. For mpg>=17 & mpg<22.75 mark as OK. For mpg>=22.75 & mpg<29 mark as Eco. For mpg>=29 mark as Excellent.

```
Economy = rep("Gas consumption", length(Auto$mpg))

Economy[Auto$mpg<17] = "Heavy"
Economy[Auto$mpg>=17 & Auto$mpg<22.75] = "OK"
Economy[Auto$mpg>=22.75 & Auto$mpg<29] = "Eco"
Economy[Auto$mpg>=29] = "Excellent"
table(Economy)
```

```
## Economy
##       Eco Excellent     Heavy        OK
##        93       103        92       104
```

From the table above, we see the frequencies of where the data falls into these separate categories. As we see, the numbers are pretty evenly spread across the four categories.

c) Perform LDA using the lda function and using all the available data. Interpret the output based on the theory discussed in class. Use 'library(MASS)'.

```
library(MASS)
lda(Economy ~ acceleration + year + horsepower + weight)
```

```
## Call:
## lda(Economy ~ acceleration + year + horsepower + weight)
##
## Prior probabilities of groups:
##       Eco Excellent     Heavy        OK
## 0.2372449 0.2627551 0.2346939 0.2653061
##
## Group means:
##           acceleration     year horsepower   weight
```

```
## Eco           16.33011 76.04301   87.82796 2537.387
## Excellent     16.64757 78.93204   70.69903 2151.816
## Heavy         13.23043 73.29348  158.20652 4151.380
## OK            15.78462 75.37500  105.25962 3150.692
##
## Coefficients of linear discriminants:
##                       LD1          LD2          LD3
## acceleration -0.011123931  0.031857342 -0.249711185
## year         -0.193137397 -0.233122185  0.153228971
## horsepower    0.009199232 -0.044693477 -0.050634817
## weight        0.002222240  0.001371949  0.002151756
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.9814 0.0128 0.0058
```

Above, we receive sample proportions of the 4 groups from our data. This comes from the section called "Prior probabilities of groups." For example we have a .265 probability that a random observation comes from the "ok" class.

We also see the group means for the other variables sorted by their groups. We see that the "Excellent group as an average HP of 70.69 and a mean weight of 2151.816.

Then, the coefficients of linear discriminants shows us the coefficients to find the lines to discriminate between the four groups. As we see there are only three columns rather than four. This is because the number of deltas is given by the minimum between k-1 and p. In our data k = 4 and p = 4. So k-1 = 3, which is the minimum. The first column separates the first class from the second, third and fourth. The second distinguishes the second class from the third and the fourt. Lastly, delta 3 distinguishes the third class from the fourth.

The last section captures the differences between the group variances. LD1 captures 98.14% of the difference, LD2 adds 1.28% and the LD3 only adds .58%. Obviously, the first one is the most important.

d) Perform cross-validation using the CV=TRUE option. Construct the confusion matrix as well as the proportion of correctly classified counts. Option CV=TRUE is used for a leave one out cross-validation; for each sampling unit, it gives its class assignment without the current observation. This is a method of estimating the testing classifications rate instead of the training rate.

```
lda.fit = lda(Economy ~ acceleration + year + horsepower + weight, CV = TRUE)
table(Economy, lda.fit$class)
```

```
##
## Economy     Eco Excellent Heavy OK
##   Eco        61        20     0 12
##   Excellent  15        86     0  2
##   Heavy       0         0    78 14
##   OK         22         1     8 73
```

The correctly classified counts are the ones on the main diagonal: 61, 86, 78, 73. The CV = TRuE line lets us estimate the testing classification rate. This procedure is repeated to produce less bias and avoid overestimating the test error. Lastly, the lda.fit$class gives the class to each observation.

```
mean(Economy == lda.fit$class)
```

```
## [1] 0.7602041
```

This shows the classification rate (76%).

e) Specify our own prior distribution; c(0.25,0.25,0,25,0.25) lists prior probabilities in the same order the classes are listed. Construct the confusion matrix as well as the proportion of correctly classified counts. What do you observe on the results?

```
lda.fit = lda(Economy ~ acceleration + year + horsepower + weight, prior = c(.25,.25,.25,.25), CV = TRU
table(Economy, lda.fit$class)
```

```
##
## Economy     Eco Excellent Heavy OK
##   Eco        68        14     0 11
##   Excellent  16        86     0  1
##   Heavy       0         0    79 13
##   OK         22         1     8 73
```

```
mean(Economy ==lda.fit$class)
```

```
## [1] 0.7806122
```

Here we see that priors made an impact and the classifcation rate increased to 78%.

f) For this part use the priors c(0.4,0.3,0.2,0.1). What do you observe on the results?

```
lda.fit = lda(Economy ~ acceleration + year + horsepower + weight, prior = c(.4,.3,.2,.1),CV = TRUE)
table(Economy, lda.fit$class)
```

```
##
## Economy     Eco Excellent Heavy OK
##   Eco        77        12     0  4
##   Excellent  19        84     0  0
##   Heavy       0         0    80 12
##   OK         51         0    11 42
```

```
mean(Economy ==lda.fit$class)
```

```
## [1] 0.7219388
```

With this set of priors we get a lower classification rate. This is probably due to the uneven distributions in the prior.