DA410_Assignment5_MattGraham

This is our first project, analyzing air pollution, mortality rates, and relevant parameters.

```
library(nnspat)  # used for dist2full()
library("dplyr")  # used to select numeric datatypes
library("ggplot2")
library(reshape)  # used for melting matricies
library(klaR)
library(ggvis)
library(class)
library(gmodels)
library(MASS)
```

9.7 -

Do a classification analysis on the beetle data in Table 5.5

Setting up data

```
beetles <- read.table("C:/mattgraham93.github.io/school/22_3_DA410/data/T5_5_FBEETLES.DAT", h
eader=FALSE)
colnames(beetles) <- c("Experiment", 'Species', 'post_dist', 'l_elytra','len_sec_an', 'len_th
i_an')

oleracea <- beetles[beetles$Species==1, -c(1:2)]  # we assume that species 1 = oleracea
carduorum <- beetles[beetles$Species==2, -c(1:2)] # we assume that species 2 = carduorum

beetles
```

Experiment	Species	post_dist	l_elytra	len_sec_an	len_thi_an
<int>	<int>	<int>	<int>	<int>	<int>
1	1	189	245	137	163
2	1	192	260	132	217
3	1	217	276	141	192
4	1	221	299	142	213
5	1	171	239	128	158
6	1	192	262	147	173
7	1	213	278	136	201
8	1	192	255	128	185
9	1	170	244	128	192

Experiment <int>	Species <int>	post_dist <int>	l_elytra <int>	len_sec_an <int>	len_thi_an <int>
10	1	201	276	146	186

1-10 of 39 rows Previous **1** 2 3 4 Next

(a) Find the classification function z = (y1 — y~2)’Sp1^-1*y and the cutoff point 1/2(times)(ʹlı + z2).

```
cov.ol <- cov(oleracea)
cov.ca <- cov(carduorum)

n1 <- nrow(oleracea)
n2 <- nrow(carduorum)

df1 <- n1 - 1
df2 <- n2 - 1

s.pl <- ( (df1 * cov.ol) + (df2 * cov.ca) ) / (df1 + df2)

as.data.frame(s.pl)
```

	post_dist <dbl>	l_elytra <dbl>	len_sec_an <dbl>	len_thi_an <dbl>
post_dist	143.55910	151.8034	42.52660	71.99253
l_elytra	151.80341	367.7878	121.87653	106.24467
len_sec_an	42.52660	121.8765	118.31408	42.06401
len_thi_an	71.99253	106.2447	42.06401	208.07290

4 rows

Finding column means and z.bar

```
means.ol <- colMeans(oleracea)
means.ca <- colMeans(carduorum)

z.bar <- (means.ol - means.ca) %*% solve(s.pl)

z.ol <- z.bar - means.ol
z.ca <- z.bar - means.ca
```

z.bar

```
as.data.frame(z.bar)
```

post_dist <dbl>	l_elytra <dbl>	len_sec_an <dbl>	len_thi_an <dbl>
0.345249	-0.1303878	-0.1064338	-0.1433533

1 row

z.ol

```
as.data.frame(z.ol)
```

post_dist <dbl>	l_elytra <dbl>	len_sec_an <dbl>	len_thi_an <dbl>
-194.1284	-267.183	-137.4749	-186.0907

1 row

z.ca

```
as.data.frame(z.ca)
```

post_dist <dbl>	l_elytra <dbl>	len_sec_an <dbl>	len_thi_an <dbl>
-179.2048	-290.9304	-157.3064	-209.3934

1 row

Finding our cut-off point

```
cutoff <- (z.ol + z.ca) / 2
as.data.frame(cutoff)
```

post_dist <dbl>	l_elytra <dbl>	len_sec_an <dbl>	len_thi_an <dbl>
-186.6666	-279.0567	-147.3906	-197.742

1 row

When looking at our z-bars and associated cut-off point. We can see that all but the post_dist are what primarily distinguish carduorim from olercea beetles!

(b) Find the classification table using the linear classification function in part (a).

```
beetle.lda <- lda(Species ~ post_dist + l_elytra + len_sec_an + len_thi_an, data=beetles)
beetle.lda
```

```
## Call:
## lda(Species ~ post_dist + l_elytra + len_sec_an + len_thi_an,
##     data = beetles)
##
## Prior probabilities of groups:
##         1         2
## 0.4871795 0.5128205
##
## Group means:
##   post_dist l_elytra len_sec_an len_thi_an
## 1  194.4737 267.0526   137.3684   185.9474
## 2  179.5500 290.8000   157.2000   209.2500
##
## Coefficients of linear discriminants:
##                    LD1
## post_dist  -0.09327642
## l_elytra    0.03522706
## len_sec_an  0.02875538
## len_thi_an  0.03872998
```

We can see our linear discriminant agree with the sentiment above. We have 3 primary factors that will inherently increase the likelihood that we assign a beetle to class 2, with post_dist being what levels the playing field to provide us with the likelihood of having of class 1.

```
beetle.predict <- predict(beetle.lda, beetles)$class
table(beetles$Species, beetle.predict, dnn = c('Actual group', 'Predicted group'))
```

```
##             Predicted group
## Actual group  1  2
##            1 19  0
##            2  1 19
```

Error rate:

```
err.rate <- 1 / (n1+n2)
err.rate
```

```
## [1] 0.02564103
```

Overall, we can conclude our model does a great job at predicting a beetle’s species within ~2.5%. That’s statistically significant, and can be improved!

(c) Find the classificaion table using the nearest neighbor method.

```
set.seed(1234)

beetle.ind <- sample(2, nrow(beetles), replace=TRUE, prob = c(0.67, 0.33))
beetle.training <- beetles[beetle.ind==1, 3:6]
beetle.test <- beetles[beetle.ind==2, 3:6]

beetle.trainingLabels <- beetles[beetle.ind==1,2]
beetle.testLabels <- beetles[beetle.ind==2,2]

beetle.knn.pred <- knn(train=beetle.training, test=beetle.test, cl=beetle.trainingLabels, k=
5)

beetleTestLabels <- data.frame(beetle.testLabels)

merge <- data.frame(beetle.knn.pred, beetleTestLabels)
merge
```

beetle.knn.pred	**beetle.testLabels**
<fct>	<int>
1	1
1	1
1	1
1	1
2	2
2	2
2	2
1	2
2	2

9 rows

Chi-square test

```
CrossTable(x=beetle.testLabels, y=beetle.knn.pred, prop.chisq=FALSE)
```

```
## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
## 
## 
## Total Observations in Table:  9 
## 
## 
##                   | beetle.knn.pred 
## beetle.testLabels |         1 |         2 | Row Total | 
## ------------------|-----------|-----------|-----------|
##                 1 |         4 |         0 |         4 | 
##                   |     1.000 |     0.000 |     0.444 | 
##                   |     0.800 |     0.000 |           | 
##                   |     0.444 |     0.000 |           | 
## ------------------|-----------|-----------|-----------|
##                 2 |         1 |         4 |         5 | 
##                   |     0.200 |     0.800 |     0.556 | 
##                   |     0.200 |     1.000 |           | 
##                   |     0.111 |     0.444 |           | 
## ------------------|-----------|-----------|-----------|
##      Column Total |         5 |         4 |         9 | 
##                   |     0.556 |     0.444 |           | 
## ------------------|-----------|-----------|-----------|
## 
## 
```

9.12 -

Do a classification analysis on the rootstock data of Table 6.2

```
root <- read.table("C:/mattgraham93.github.io/school/22_3_DA410/data/T6_2_ROOT.DAT", header=F
ALSE)
colnames(root) <- c('Rootstock', 'girth_4', 'growth_4', 'girth_15', 'weight_15')

k <- 6
p <- 4
n <- 8

root
```

Rootstock	girth_4	growth_4	girth_15	weight_15
<int>	<dbl>	<dbl>	<dbl>	<dbl>
1	1.11	2.569	3.58	0.760
1	1.19	2.928	3.75	0.821
1	1.09	2.865	3.93	0.928
1	1.25	3.844	3.94	1.009
1	1.11	3.027	3.60	0.766
1	1.08	2.336	3.51	0.726
1	1.11	3.211	3.98	1.209
1	1.16	3.037	3.62	0.750
2	1.05	2.074	4.09	1.036
2	1.17	2.885	4.06	1.094

1-10 of 48 rows Previous **1** 2 3 4 5 Next

(a) Find the linear classification functions.

Finding s.pl

```
root.s.pl <- matrix(0, nrow = p, ncol = p)

for(i in 1:k) {
  root.s.pl <- root.s.pl + cov(root[root$Rootstock == i, 2:5]) / k
}

as.data.frame(root.s.pl)
```

	girth_4	growth_4	girth_15	weight_15
	<dbl>	<dbl>	<dbl>	<dbl>
girth_4	0.00761875	0.04039437	0.01319256	0.00517000
growth_4	0.04039437	0.28911405	0.10389554	0.05024318
girth_15	0.01319256	0.10389554	0.10216220	0.05908705
weight_15	0.00517000	0.05024318	0.05908705	0.04101249

4 rows

Finding ybar

```
ybar <- matrix(nrow = p, ncol = k)
rownames(ybar) <- colnames(root[, 2:5])
for(i in 1:k) {
  ybar[,i] <- colMeans(root[root$Rootstock == i, 2:5])
}
as.data.frame(ybar)
```

	V1	V2	V3	V4	V5	V6
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
girth_4	1.137500	1.157500	1.107500	1.09750	1.08000	1.036250
growth_4	2.977125	3.109125	2.815250	2.87975	2.55725	2.214625
girth_15	3.738750	4.515000	4.455000	3.90625	4.31250	3.596250
weight_15	0.871125	1.280500	1.391375	1.03900	1.18100	0.735000

4 rows

Completing classification function

```
classification <- matrix(nrow = p + 1, ncol = k)
rownames(classification) <- c('c0','c1','c2','c3','c4')

for(i in 1:k) {
  classification[1, i] <- ybar[,i]%*%solve(root.s.pl)%*%as.matrix(ybar[,i]/2)

  classification[2:(p+1), i] <-ybar[,i]%*%solve(root.s.pl)
}
as.data.frame(classification)
```

	V1	V2	V3	V4	V5	V6
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
c0	299.98047	353.15985	328.49312	291.8181	347.54494	315.79619
c1	314.63998	317.11988	324.58920	307.2601	316.76668	311.30097
c2	-59.41719	-63.98083	-65.15229	-59.3731	-65.82568	-63.06005
c3	149.61051	168.16143	154.91041	147.6523	168.22072	160.62208
c4	-161.17769	-172.64428	-150.35643	-153.3870	-172.85120	-175.47759

5 rows

Above is the complete classification analysis across all rootstocks and measurements. We can see some interesting variability across c3.

(b) Find the classification table using the linear classification functions in part (a) (assuming Σ1 = Σ2 = Σ3).

```
roots.lda <- lda(Rootstock ~ girth_4 + growth_4 + girth_15 + weight_15, data=root)
roots.lda
```

```
## Call:
## lda(Rootstock ~ girth_4 + growth_4 + girth_15 + weight_15, data = root)
## 
## Prior probabilities of groups:
##         1         2         3         4         5         6 
## 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 
## 
## Group means:
##   girth_4 growth_4 girth_15 weight_15
## 1 1.13750 2.977125  3.73875  0.871125
## 2 1.15750 3.109125  4.51500  1.280500
## 3 1.10750 2.815250  4.45500  1.391375
## 4 1.09750 2.879750  3.90625  1.039000
## 5 1.08000 2.557250  4.31250  1.181000
## 6 1.03625 2.214625  3.59625  0.735000
## 
## Coefficients of linear discriminants:
##                  LD1        LD2       LD3       LD4
## girth_4    3.0479952  -1.140083 -1.002448 23.419063
## growth_4  -1.7025953  -1.215888  1.672714 -3.076804
## girth_15   4.2332645   7.166403  3.045553 -2.011416
## weight_15 -0.4785144 -11.520302 -5.506192  3.101660
## 
## Proportion of trace:
##    LD1    LD2    LD3    LD4 
## 0.6421 0.2707 0.0784 0.0089
```

Observing the diagonals, we can see that our main determining factor is growth at a 4 year period which will decrease the classification determination by a fairly significant factor. This can infer that growth at 4 years is a fairly significant determining factor of how we classify our rootstock.

Getting predictions

```
roots.lda.pred <- predict(roots.lda)$class
table(root$Rootstock, roots.lda.pred, dnn=c("Actual group", "Predicted group"))
```

```
##             Predicted group
## Actual group 1 2 3 4 5 6
##            1 5 0 0 1 0 2
##            2 0 3 2 1 2 0
##            3 0 0 6 1 1 0
##            4 3 0 1 4 0 0
##            5 0 3 1 0 3 1
##            6 2 0 0 0 2 4
```

Defining accuracy

```
correct <- sum(roots.lda.pred==root$Rootstock)
total <- nrow(root)

root.pred.perc <- correct / total
root.pred.err <- 1 - root.pred.perc

root.pred.err
```

```
## [1] 0.4791667
```

We did not produce a very strong model using linear discriminant analysis. This would not be something to use in real life.

(c) Find the classification table using quadratic classification functions (assuming population covariance matrices are not equal).

QDA

```
roots.qda <- qda(Rootstock ~ girth_4 + growth_4 + girth_15 + weight_15, data=root)
roots.qda
```

```
## Call:
## qda(Rootstock ~ girth_4 + growth_4 + girth_15 + weight_15, data = root)
## 
## Prior probabilities of groups:
##         1         2         3         4         5         6 
## 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 
## 
## Group means:
##   girth_4 growth_4 girth_15 weight_15
## 1 1.13750 2.977125  3.73875  0.871125
## 2 1.15750 3.109125  4.51500  1.280500
## 3 1.10750 2.815250  4.45500  1.391375
## 4 1.09750 2.879750  3.90625  1.039000
## 5 1.08000 2.557250  4.31250  1.181000
## 6 1.03625 2.214625  3.59625  0.735000
```

QDA predictions

```
roots.qda.pred <- predict(roots.qda)$class

table(root$Rootstock, roots.qda.pred, dnn = c("Actual group", 'Predicted group'))
```

```
##              Predicted group
## Actual group 1 2 3 4 5 6
##            1 8 0 0 0 0 0
##            2 0 7 0 1 0 0
##            3 1 0 6 0 1 0
##            4 0 0 1 7 0 0
##            5 0 3 0 0 4 1
##            6 2 0 0 0 1 5
```

Defining accuracy

```
correct <- sum(roots.qda.pred==root$Rootstock)
total <- nrow(root)

root.pred.perc <- correct / total
root.pred.err <- 1 - root.pred.perc

root.pred.err
```

```
## [1] 0.2291667
```

Here, we can see a 50% reduction in error, which is significant, but not significant enough.

(d) Find the classification table using the nearest neighbor method.

```
set.seed(1234)

roots.ind <- sample(2, nrow(root), replace=TRUE, prob = c(0.67, 0.33))
roots.training <- root[roots.ind==1, 2:5]
roots.test <- root[roots.ind==2, 2:5]

roots.trainingLabels <- root[roots.ind==1,2]
roots.testLabels <- root[roots.ind==2,2]

roots.knn.pred <- knn(train=roots.training, test=roots.test, cl=roots.trainingLabels, k=5)

rootsTestLabels <- data.frame(roots.testLabels)

roots.merge <- data.frame(roots.knn.pred, rootsTestLabels)
roots.merge
```

roots.knn.pred	roots.testLabels
<fct>	<dbl>
1.19	1.11
1.2	1.11
1.17	1.15

roots.knn.pred <fct>	roots.testLabels <dbl>
1.11	1.19
1.05	1.03
1.05	1.01
1.05	0.99
1.15	1.05
1.02	1.05
1.2	1.13

1-10 of 11 rows Previous **1** 2 Next

Determining accuracy

```
CrossTable(x=roots.testLabels, y=roots.knn.pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  11
##
##
##                  | roots.knn.pred
## roots.testLabels |      1.02 |      1.05 |      1.09 |      1.11 |      1.15 |      1.17 |
1.19 |       1.2 | Row Total |
##
-----------------|-----------|-----------|-----------|-----------|-----------|-----------|---
--------|-----------|-----------|
##             0.99 |         0 |         1 |         0 |         0 |         0 |         0 |
0 |         0 |         1 |
##                  |     0.000 |     1.000 |     0.000 |     0.000 |     0.000 |     0.000 |
0.000 |     0.000 |     0.091 |
##                  |     0.000 |     0.333 |     0.000 |     0.000 |     0.000 |     0.000 |
0.000 |     0.000 |           |
##                  |     0.000 |     0.091 |     0.000 |     0.000 |     0.000 |     0.000 |
0.000 |     0.000 |           |
##
-----------------|-----------|-----------|-----------|-----------|-----------|-----------|---
--------|-----------|-----------|
##             1.01 |         0 |         1 |         0 |         0 |         0 |         0 |
0 |         0 |         1 |
##                  |     0.000 |     1.000 |     0.000 |     0.000 |     0.000 |     0.000 |
0.000 |     0.000 |     0.091 |
##                  |     0.000 |     0.333 |     0.000 |     0.000 |     0.000 |     0.000 |
0.000 |     0.000 |           |
##                  |     0.000 |     0.091 |     0.000 |     0.000 |     0.000 |     0.000 |
0.000 |     0.000 |           |
##
-----------------|-----------|-----------|-----------|-----------|-----------|-----------|---
--------|-----------|-----------|
##             1.03 |         0 |         1 |         0 |         0 |         0 |         0 |
0 |         0 |         1 |
##                  |     0.000 |     1.000 |     0.000 |     0.000 |     0.000 |     0.000 |
0.000 |     0.000 |     0.091 |
##                  |     0.000 |     0.333 |     0.000 |     0.000 |     0.000 |     0.000 |
0.000 |     0.000 |           |
##                  |     0.000 |     0.091 |     0.000 |     0.000 |     0.000 |     0.000 |
0.000 |     0.000 |           |
##
```

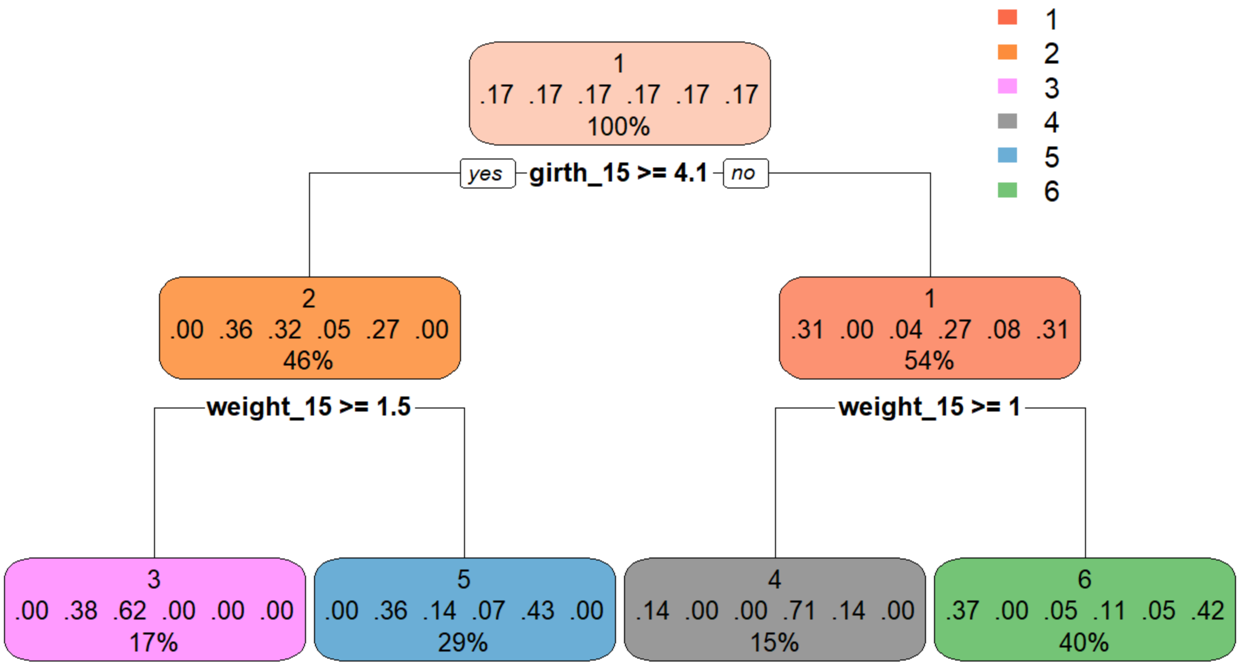
```
-----------------|-----------|-----------|-----------|-----------|-----------|-----------|---
--------|-----------|-----------|
##             1.05 |         1 |         0 |         0 |         0 |         1 |         0 |
0 |         0 |         2 |
##                  |     0.500 |     0.000 |     0.000 |     0.000 |     0.500 |     0.000 |
0.000 |     0.000 |     0.182 |
##                  |     1.000 |     0.000 |     0.000 |     0.000 |     1.000 |     0.000 |
0.000 |     0.000 |           |
##                  |     0.091 |     0.000 |     0.000 |     0.000 |     0.091 |     0.000 |
0.000 |     0.000 |           |
##
-----------------|-----------|-----------|-----------|-----------|-----------|-----------|---
--------|-----------|-----------|
##             1.11 |         0 |         0 |         0 |         0 |         0 |         0 |
1 |         1 |         2 |
##                  |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |
0.500 |     0.500 |     0.182 |
##                  |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |
1.000 |     0.500 |           |
##                  |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |
0.091 |     0.091 |           |
##
-----------------|-----------|-----------|-----------|-----------|-----------|-----------|---
--------|-----------|-----------|
##             1.13 |         0 |         0 |         1 |         0 |         0 |         0 |
0 |         1 |         2 |
##                  |     0.000 |     0.000 |     0.500 |     0.000 |     0.000 |     0.000 |
0.000 |     0.500 |     0.182 |
##                  |     0.000 |     0.000 |     1.000 |     0.000 |     0.000 |     0.000 |
0.000 |     0.500 |           |
##                  |     0.000 |     0.000 |     0.091 |     0.000 |     0.000 |     0.000 |
0.000 |     0.091 |           |
##
-----------------|-----------|-----------|-----------|-----------|-----------|-----------|---
--------|-----------|-----------|
##             1.15 |         0 |         0 |         0 |         0 |         0 |         1 |
0 |         0 |         1 |
##                  |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     1.000 |
0.000 |     0.000 |     0.091 |
##                  |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     1.000 |
0.000 |     0.000 |           |
##                  |     0.000 |     0.000 |     0.000 |     0.000 |     0.000 |     0.091 |
0.000 |     0.000 |           |
##
-----------------|-----------|-----------|-----------|-----------|-----------|-----------|---
--------|-----------|-----------|
##             1.19 |         0 |         0 |         0 |         1 |         0 |         0 |
0 |         0 |         1 |
##                  |     0.000 |     0.000 |     0.000 |     1.000 |     0.000 |     0.000 |
0.000 |     0.000 |     0.091 |
##                  |     0.000 |     0.000 |     0.000 |     1.000 |     0.000 |     0.000 |
0.000 |     0.000 |           |
##                 |     0.000 |     0.000 |     0.000 |     0.091 |     0.000 |     0.000 |
0.000 |     0.000 |           |
##
-----------------|-----------|-----------|-----------|-----------|-----------|-----------|---
--------|-----------|-----------|
##    Column Total |         1 |         3 |         1 |         1 |         1 |         1 |
1 |         2 |        11 |
##                 |     0.091 |     0.273 |     0.091 |     0.091 |     0.091 |     0.091 |
0.091 |     0.182 |           |
##
-----------------|-----------|-----------|-----------|-----------|-----------|-----------|---
--------|-----------|-----------|
##
##
```

I don’t really know how to analyze this table for accuracy. Ultimately, it looks like we were only able to determine one classification correctly. I must have done something wrong.

Decision Tree example

```
library(rpart)
library(rpart.plot)

fit <- rpart(Rootstock ~ girth_4 + growth_4 + girth_15 + weight_15, data=root, method='class
')
rpart.plot(fit)
```


1
.17 .17 .17 .17 .17 .17
100%
yes
girth_15 >= 4.1
no
2
.00 .36 .32 .05 .27 .00
46%
1
.31 .00 .04 .27 .08 .31
54%
weight_15 >= 1.5
weight_15 >= 1
3
.00 .38 .62 .00 .00 .00
17%
5
.00 .36 .14 .07 .43 .00
29%
4
.14 .00 .00 .71 .14 .00
15%
6
.37 .00 .05 .11 .05 .42
40%
1
2
3
4
5
6