35459 Multivariate Statistics

Week 11 (approx) **Seminar -**
Discrimination and Classification II

Classification using Logistic Regression

Wine Data

The wine data set contains data collected from three different cultivars of wine, measuring 13 different variables describing the concentration of chemicals that can be found in the wine as well as other properties of the wine. The variables are:

- Cult: Cultivar
- Alc: Alcohol
- MalAcid: Malic acid
- Ash: Ash
- AshAlk: Alcalinity of ash
- Mag: Magnesium
- TotPhen: Total phenols
- Flav: Flavanoids
- NonFlav: Nonflavanoid phenols
- Proant: Proanthocyanins
- Color: Color intensity
- Hue: Hue
- OD280OD315: OD280/OD315 of diluted wines
- Proline: Proline

Use multinomial logistic regression, CART and Neural Networks to develop rules that discriminate between the three cultivars.

```
library(VGAM)
library(rpart)
library(nnet)
```

```
wine_data <- read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data",sep=",")
colnames(wine_data)<-c("Cult","Alc","MalAcid","Ash","AshAlk","Mag","TotPhen","Flav",
                       "NonFlav","Proant","Color","Hue","OD280OD315","Proline")
```

```
attach(wine_data)
wine_data$Cult_1[wine_data$Cult==1]<-1
wine_data$Cult_1[wine_data$Cult!=1]<-0
wine_data$Cult_2[wine_data$Cult==2]<-1
wine_data$Cult_2[wine_data$Cult!=2]<-0
wine_data$Cult_3[wine_data$Cult==3]<-1
wine_data$Cult_3[wine_data$Cult!=3]<-0
detach(wine_data)
```

```
test_rows<-sample(1:nrow(wine_data), round(nrow(wine_data)*0.2) ,replace=FALSE)
wine_train<-wine_data[-test_rows,]
wine_test<-wine_data[test_rows,]
rownames(wine_test)<-NULL

wine_mnl<-vglm(formula = cbind(Cult_1,Cult_2,Cult_3) ~ Alc+MalAcid+Ash
               +AshAlk+Mag+TotPhen+Flav+NonFlav+Proant+Color+Hue
               +OD280OD315+Proline, family = multinomial, data = wine_train)

predictions<-predict(wine_mnl,newdata=wine_test,type="response")
wine_test$pred_mnl<-apply(predictions,1,function(i) which.max(i) )
print(table(wine_test$Cult,wine_test$pred_mnl))

##
##      1  2  3
##   1 12  0  0
##   2  0 16  0
##   3  0  0  8
```

Classification using CART

Classification trees are useful when the classification variable is categorical and regression trees are useful when the classification variable is numeric, and you would like to create a partition based around similar means.

Wine Data

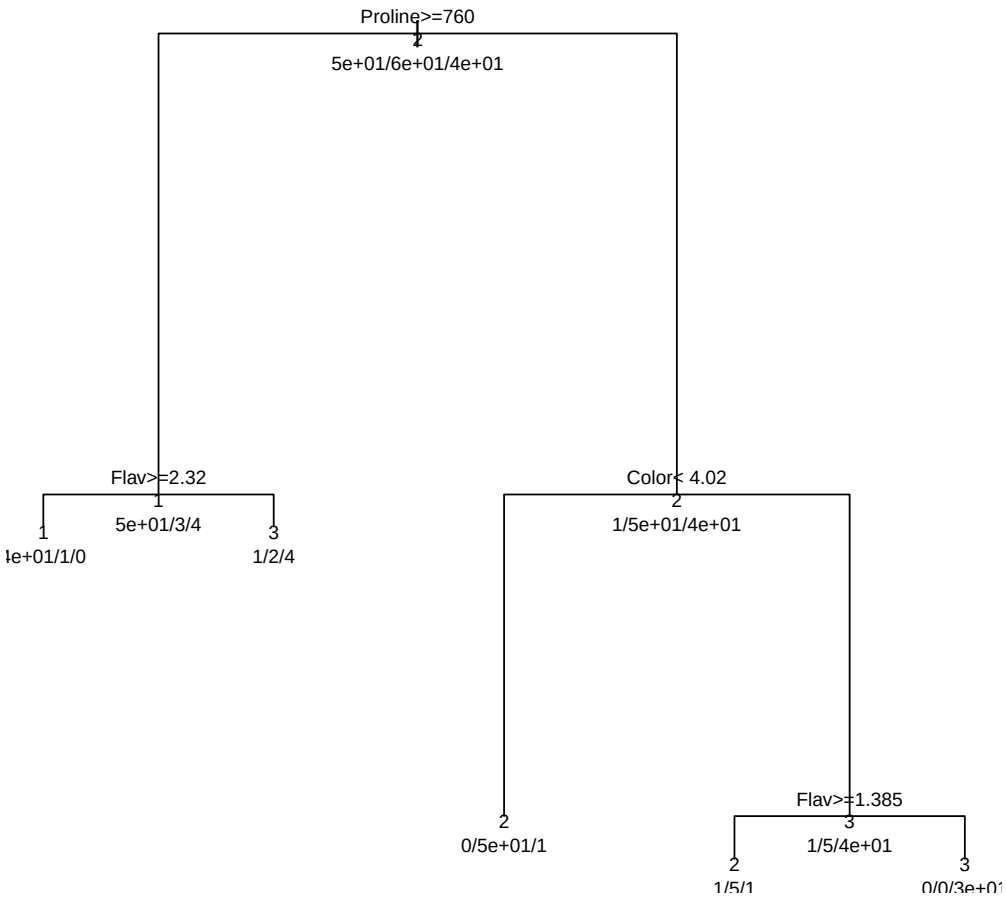
Classification Tree

```
wine_ct<-rpart(as.factor(Cult) ~ Alc+MalAcid+Ash+AshAlk+Mag+TotPhen+Flav
               +NonFlav+Proant+Color+Hue+OD280OD315+Proline,
```

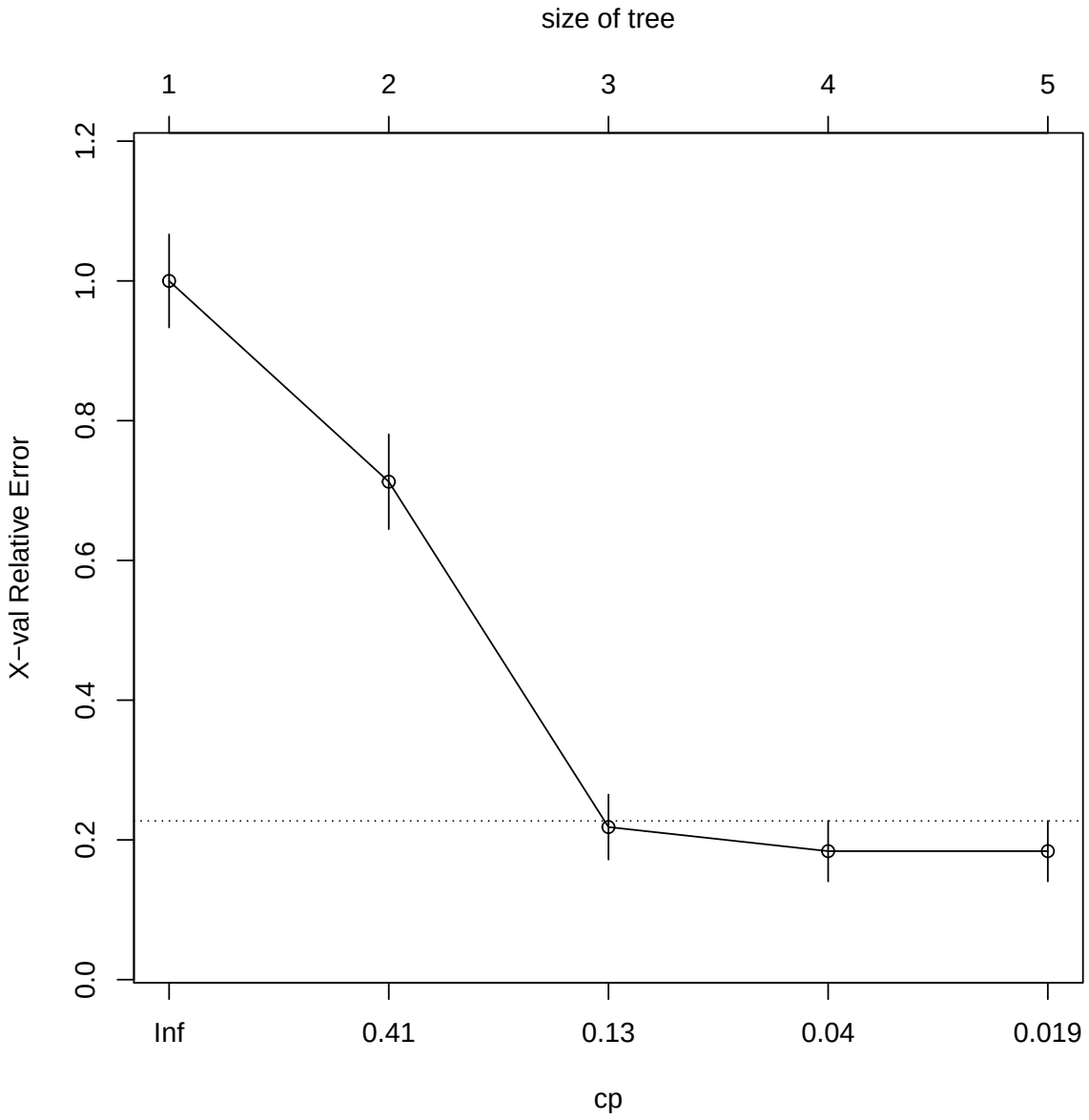
```
               data = wine_train, method="class")

summary(wine_ct)

## Call:
## rpart(formula = as.factor(Cult) ~ Alc + MalAcid + Ash + AshAlk +
##     Mag + TotPhen + Flav + NonFlav + Proant + Color + Hue + OD280OD315 +
##     Proline, data = wine_train, method = "class")
##   n= 142
##
##         CP nsplit rel error xerror    xstd
## 1 0.49425      0   1.00000 1.0000 0.06672
## 2 0.34483      1   0.50575 0.7126 0.06793
## 3 0.04598      2   0.16092 0.2184 0.04663
## 4 0.03448      3   0.11494 0.1839 0.04331
## 5 0.01000      4   0.08046 0.1839 0.04331
##
## Variable importance
##       Flav OD280OD315        Alc    Proline      Color        Hue
##         19         14         13         12         11          9
##    TotPhen    MalAcid     AshAlk        Ash    NonFlav
##          7          7          6          1          1
##
## Node number 1: 142 observations,    complexity param=0.4943
##   predicted class=2  expected loss=0.6127  P(node) =1
##     class counts:    47    55    40
##    probabilities: 0.331 0.387 0.282
##   left son=2 (53 obs) right son=3 (89 obs)
##   Primary splits:
##       Proline    < 760   to the right, improve=37.22, (0 missing)
##       Color      < 3.46  to the right, improve=35.98, (0 missing)
##       Flav       < 1.315 to the right, improve=33.64, (0 missing)
##       OD280OD315 < 2.19  to the right, improve=33.02, (0 missing)
##       Alc        < 12.76 to the right, improve=30.37, (0 missing)
##   Surrogate splits:
##       Flav       < 2.33  to the right, agree=0.852, adj=0.604, (0 split)
##       TotPhen    < 2.335 to the right, agree=0.796, adj=0.453, (0 split)
##       Alc        < 12.98 to the right, agree=0.775, adj=0.396, (0 split)
##       AshAlk     < 17.45 to the left,  agree=0.768, adj=0.377, (0 split)
##       OD280OD315 < 2.845 to the right, agree=0.739, adj=0.302, (0 split)
##
## Node number 2: 53 observations,    complexity param=0.03448
##   predicted class=1  expected loss=0.1321  P(node) =0.3732
##     class counts:    46     3     4
```

```
##    probabilities: 0.868 0.057 0.075
##   left son=4 (46 obs) right son=5 (7 obs)
##   Primary splits:
##       Flav       < 2.32  to the right, improve=6.647, (0 missing)
##       OD280OD315 < 2.67  to the right, improve=4.778, (0 missing)
##       TotPhen    < 2.405 to the right, improve=3.567, (0 missing)
##       NonFlav    < 0.425 to the left,  improve=3.567, (0 missing)
##       Hue        < 0.885 to the right, improve=3.567, (0 missing)
##   Surrogate splits:
##       TotPhen    < 2.125 to the right, agree=0.943, adj=0.571, (0 split)
##       Hue        < 0.77  to the right, agree=0.943, adj=0.571, (0 split)
##       OD280OD315 < 2.12  to the right, agree=0.943, adj=0.571, (0 split)
##       Alc        < 12.66 to the right, agree=0.925, adj=0.429, (0 split)
##       NonFlav    < 0.515 to the left,  agree=0.906, adj=0.286, (0 split)
##
## Node number 3: 89 observations,    complexity param=0.3448
##   predicted class=2  expected loss=0.4157  P(node) =0.6268
##     class counts:     1    52    36
##    probabilities: 0.011 0.584 0.404
##   left son=6 (48 obs) right son=7 (41 obs)
##   Primary splits:
##       Color      < 4.02  to the left,  improve=31.60, (0 missing)
##       Flav       < 1.315 to the right, improve=29.61, (0 missing)
##       OD280OD315 < 2.19  to the right, improve=28.51, (0 missing)
##       Hue        < 0.785 to the right, improve=24.61, (0 missing)
##       MalAcid    < 2.455 to the left,  improve=18.06, (0 missing)
##   Surrogate splits:
##       OD280OD315 < 2.055 to the right, agree=0.876, adj=0.732, (0 split)
##       Alc        < 12.52 to the left,  agree=0.854, adj=0.683, (0 split)
##       Flav       < 1.235 to the right, agree=0.843, adj=0.659, (0 split)
##       Hue        < 0.785 to the right, agree=0.809, adj=0.585, (0 split)
##       MalAcid    < 2.455 to the left,  agree=0.787, adj=0.537, (0 split)
##
## Node number 4: 46 observations
##   predicted class=1  expected loss=0.02174  P(node) =0.3239
##     class counts:    45     1     0
##    probabilities: 0.978 0.022 0.000
##
## Node number 5: 7 observations
##   predicted class=3  expected loss=0.4286  P(node) =0.0493
##     class counts:     1     2     4
##    probabilities: 0.143 0.286 0.571
##
## Node number 6: 48 observations
```

```
##   predicted class=2  expected loss=0.02083  P(node) =0.338
##     class counts:     0    47     1
##    probabilities: 0.000 0.979 0.021
##
## Node number 7: 41 observations,    complexity param=0.04598
##   predicted class=3  expected loss=0.1463  P(node) =0.2887
##     class counts:     1     5    35
##    probabilities: 0.024 0.122 0.854
##   left son=14 (7 obs) right son=15 (34 obs)
##   Primary splits:
##       Flav    < 1.385 to the right, improve=7.345, (0 missing)
##       Hue     < 0.9   to the right, improve=4.547, (0 missing)
##       Proline < 517.5 to the left,  improve=3.552, (0 missing)
##       MalAcid < 1.515 to the left,  improve=3.236, (0 missing)
##       Ash     < 2.215 to the left,  improve=3.236, (0 missing)
##   Surrogate splits:
##       Hue        < 0.97  to the right, agree=0.951, adj=0.714, (0 split)
##       OD280OD315 < 2.395 to the right, agree=0.927, adj=0.571, (0 split)
##       MalAcid    < 1.185 to the left,  agree=0.902, adj=0.429, (0 split)
##       Ash        < 2.06  to the left,  agree=0.902, adj=0.429, (0 split)
##       AshAlk     < 17.15 to the left,  agree=0.902, adj=0.429, (0 split)
##
## Node number 14: 7 observations
##   predicted class=2  expected loss=0.2857  P(node) =0.0493
##     class counts:     1     5     1
##    probabilities: 0.143 0.714 0.143
##
## Node number 15: 34 observations
##   predicted class=3  expected loss=0  P(node) =0.2394
##     class counts:     0     0    34
##    probabilities: 0.000 0.000 1.000

plot(wine_ct)
text(wine_ct, use.n=TRUE, all=TRUE, cex=.7)
```




```
plotcp(wine_ct)
```




```
wine_test$pred_ct<-predict(wine_ct,wine_test,type="vector")
table(wine_test$Cult,wine_test$pred_ct)

##
##      1  2  3
##   1 11  1  0
##   2  0 15  1
##   3  0  1  7
```

Car data

The car data set (Chambers, Cleveland, Kleiner and Tukey, 1983) consists of 13 variables measured for 74 car types. The abbreviations in Table B.3 are as follows:

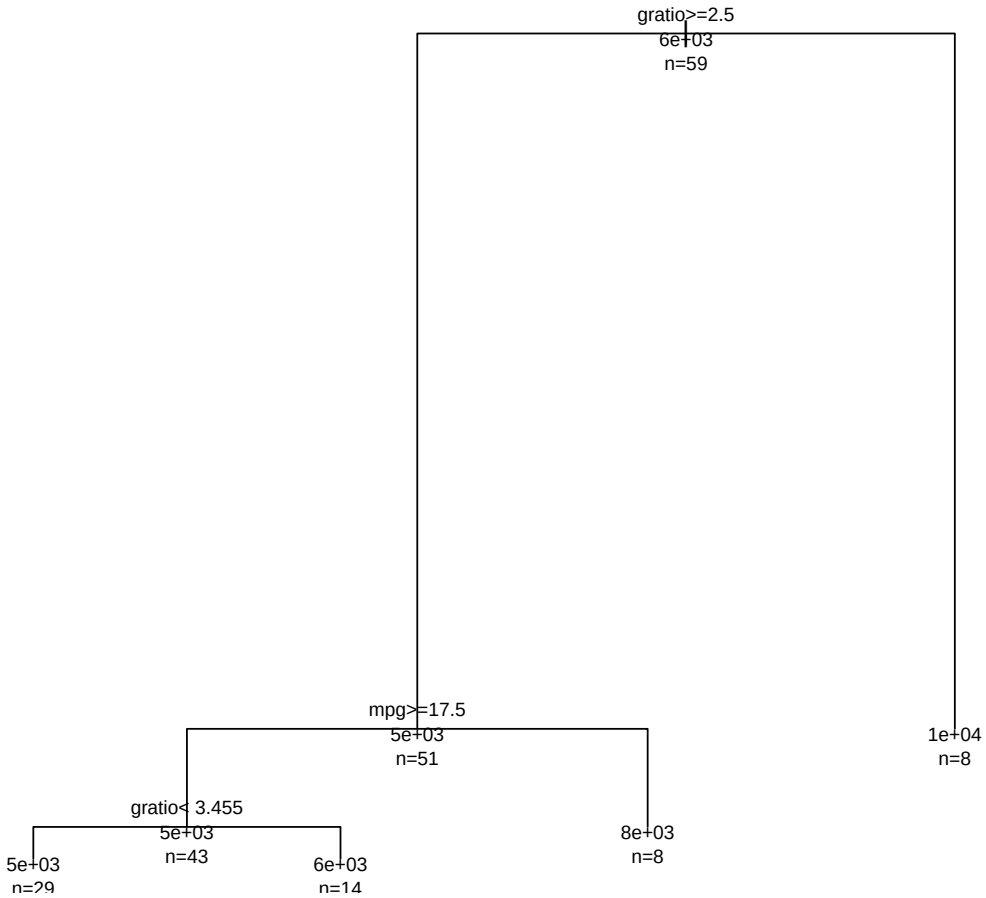
- X1: Price
- X2: Mileage (in miles per gallone)
- X3: Repair record 1978 (rated on a 5-point scale; 5 best, 1 worst)
- X4: Repair record 1977 (scale as before)
- X5: Headroom (in inches)
- X6: Rear seat clearance (distance from front seat back to rear seat, in inches)
- X7: Trunk space (in cubic feet)
- X8: Weight (in pound)
- X9: Length (in inches)
- X10: Turning diameter (clearance required to make a U-turn, in feet)
- X11: Displacement (in cubic inches)
- X12: Gear ratio for high gear
- X13: Company headquarter (1 for U.S., 2 for Japan, 3 for Europe)

Use a regression tree to develop classification rules to predict price.

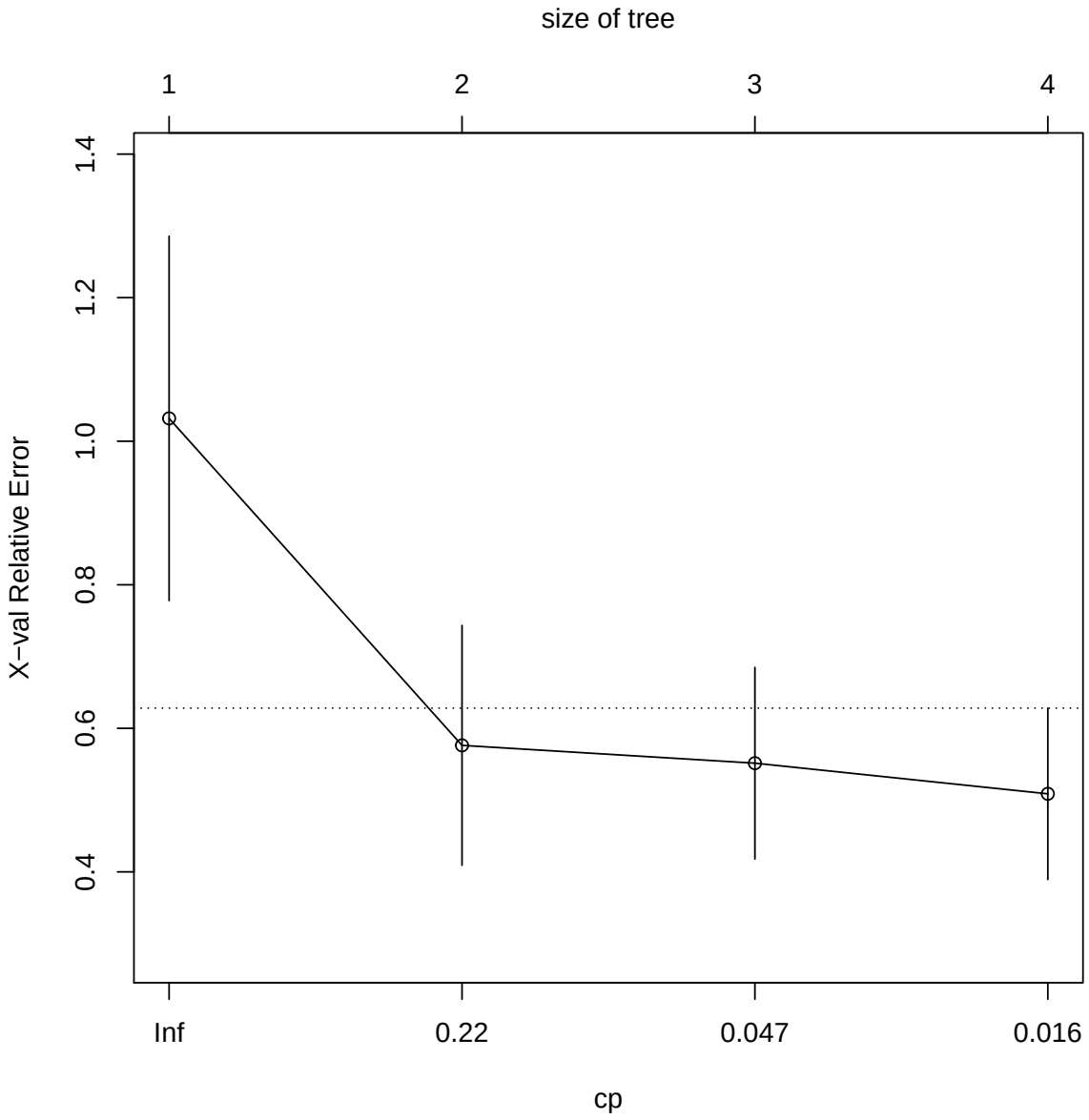
```
car_data <- read.csv("C:/Documents/car.csv") test_rows<-
sample(1:nrow(car_data), round(nrow(car_data)*0.2) ,replace=FALSE)
car_data_train<-car_data[-test_rows,]
car_data_test<-car_data[test_rows,]

car_rt<-rpart(Price ~ mpg+hroom+rseat+trunk+weight+length+turn+displa+gratio
              +Origin, data = car_data_train, method="anova")

summary(car_rt)

## Call:
## rpart(formula = Price ~ mpg + hroom + rseat + trunk + weight +
##     length + turn + displa + gratio + Origin, data = car_data_train,
##     method = "anova")
##   n= 59
```

```
##
##        CP nsplit rel error xerror   xstd
## 1 0.58808      0    1.0000 1.0318 0.2538
## 2 0.08293      1    0.4119 0.5763 0.1671
## 3 0.02714      2    0.3290 0.5514 0.1335
## 4 0.01000      3    0.3018 0.5088 0.1194
##
## Variable importance
## gratio displa weight    mpg length  rseat   turn Origin
##     26     23     18     16     11      6      1      1
##
## Node number 1: 59 observations,    complexity param=0.5881
##   mean=6355, MSE=8.876e+06
##   left son=2 (51 obs) right son=3 (8 obs)
##   Primary splits:
##       gratio < 2.5   to the right, improve=0.5881, (0 missing)
##       weight < 3785  to the left,  improve=0.4732, (0 missing)
##       displa < 334   to the left,  improve=0.4717, (0 missing)
##       mpg    < 17.5  to the right, improve=0.3464, (0 missing)
##       rseat  < 29.25 to the left,  improve=0.2946, (0 missing)
##   Surrogate splits:
##       displa < 334   to the left,  agree=0.983, adj=0.875, (0 split)
##       weight < 3890  to the left,  agree=0.949, adj=0.625, (0 split)
##       mpg    < 15.5  to the right, agree=0.932, adj=0.500, (0 split)
##       length < 219   to the left,  agree=0.915, adj=0.375, (0 split)
##       rseat  < 29.75 to the left,  agree=0.898, adj=0.250, (0 split)
##
## Node number 2: 51 observations,    complexity param=0.08293
##   mean=5450, MSE=3.166e+06
##   left son=4 (43 obs) right son=5 (8 obs)
##   Primary splits:
##       mpg    < 17.5  to the right, improve=0.2690, (0 missing)
##       Origin splits as  LRL,       improve=0.1848, (0 missing)
##       length < 171   to the left,  improve=0.1359, (0 missing)
##       rseat  < 28.75 to the left,  improve=0.1287, (0 missing)
##       trunk  < 11.5  to the left,  improve=0.1212, (0 missing)
##   Surrogate splits:
##       weight < 3715  to the left,  agree=0.941, adj=0.625, (0 split)
##       length < 205   to the left,  agree=0.902, adj=0.375, (0 split)
##       turn   < 44.5  to the left,  agree=0.902, adj=0.375, (0 split)
##       displa < 254   to the left,  agree=0.902, adj=0.375, (0 split)
##       gratio < 2.72  to the right, agree=0.863, adj=0.125, (0 split)
##
## Node number 3: 8 observations
```

```
##   mean=1.212e+04, MSE=6.784e+06
##
## Node number 4: 43 observations,    complexity param=0.02714
##   mean=5052, MSE=1.567e+06
##   left son=8 (29 obs) right son=9 (14 obs)
##   Primary splits:
##       gratio < 3.455 to the left,  improve=0.21090, (0 missing)
##       Origin splits as  LRL,       improve=0.14180, (0 missing)
##       length < 169.5 to the left,  improve=0.12790, (0 missing)
##       mpg    < 27    to the right, improve=0.11570, (0 missing)
##       hroom  < 2.75  to the right, improve=0.08827, (0 missing)
##   Surrogate splits:
##       displa < 127.5 to the right, agree=0.837, adj=0.500, (0 split)
##       Origin splits as  LRR,       agree=0.837, adj=0.500, (0 split)
##       weight < 2090  to the right, agree=0.791, adj=0.357, (0 split)
##       mpg    < 22.5  to the left,  agree=0.744, adj=0.214, (0 split)
##       length < 156.5 to the right, agree=0.744, adj=0.214, (0 split)
##
## Node number 5: 8 observations
##   mean=7590, MSE=6.329e+06
##
## Node number 8: 29 observations
##   mean=4653, MSE=4.931e+05
##
## Node number 9: 14 observations
##   mean=5880, MSE=2.777e+06

plot(car_rt)
text(car_rt, use.n=TRUE, all=TRUE, cex=.7)
```




```
plotcp(car_rt)
```




```
car_rt$cptable
```

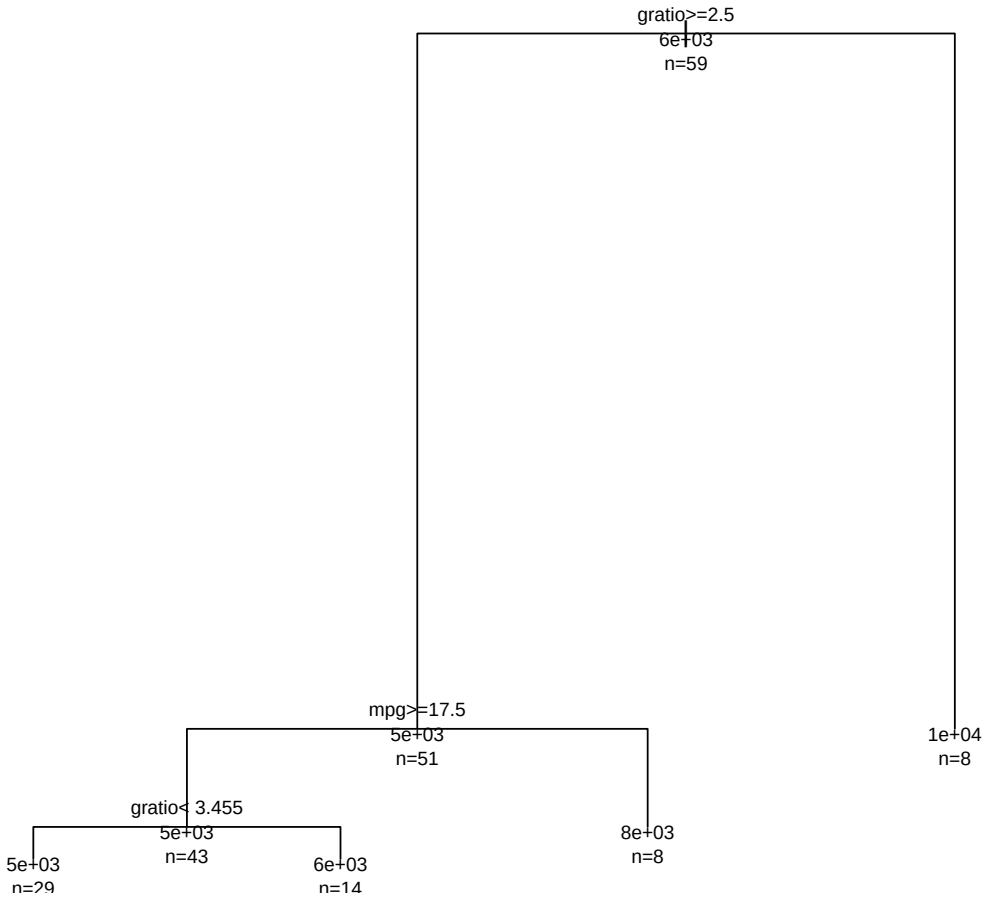
```
##         CP nsplit rel error xerror   xstd
## 1 0.58808      0    1.0000 1.0318 0.2538
## 2 0.08293      1    0.4119 0.5763 0.1671
## 3 0.02714      2    0.3290 0.5514 0.1335
## 4 0.01000      3    0.3018 0.5088 0.1194
```

```
car_pfit<- prune(car_rt, cp=car_rt$cptable[which.min(car_rt$cptable[,"xerror"])
                                           ,"CP"])
summary(car_pfit)
```

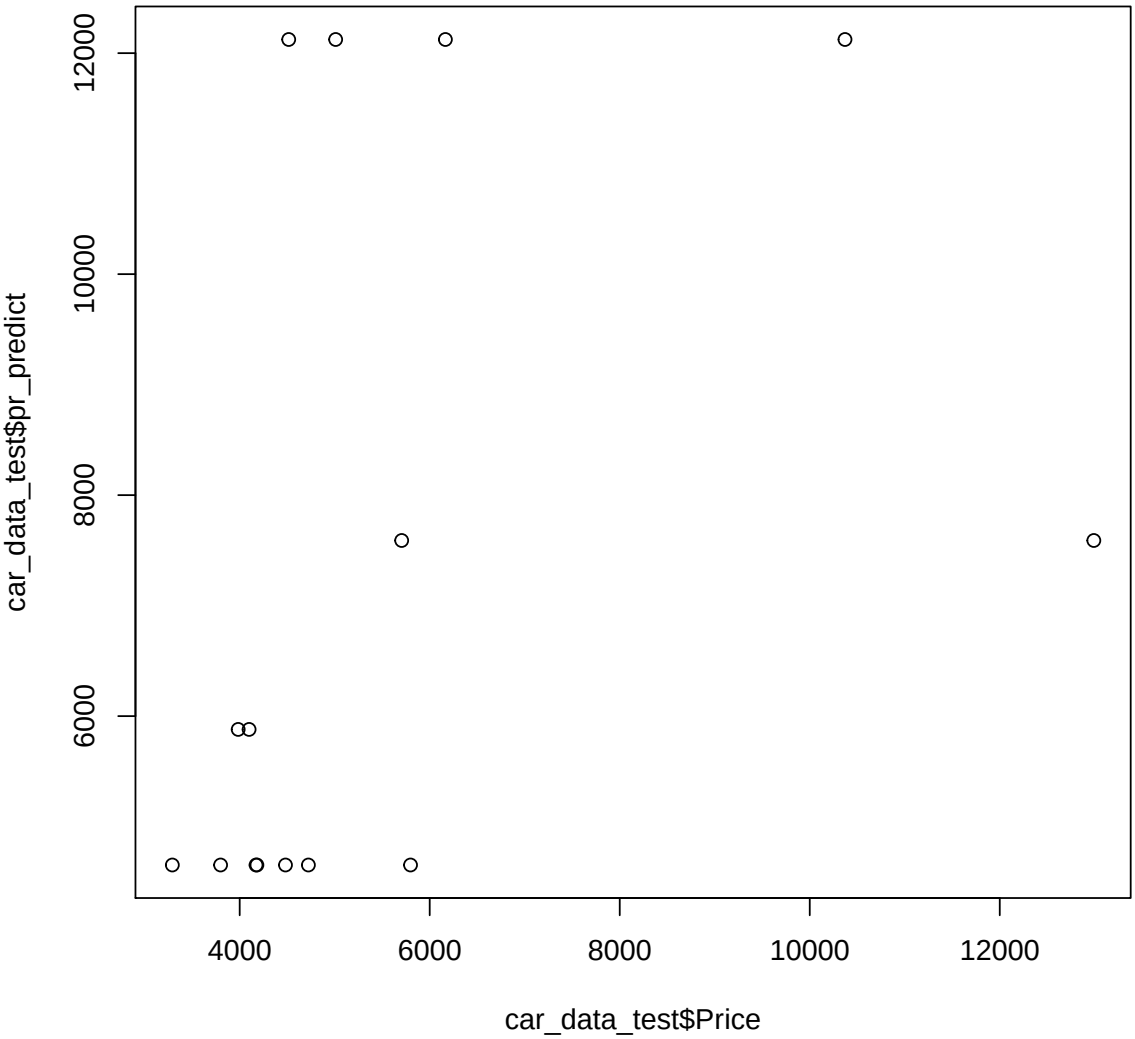
```
## Call:
```

```
## rpart(formula = Price ~ mpg + hroom + rseat + trunk + weight +
##     length + turn + displa + gratio + Origin, data = car_data_train,
##     method = "anova")
##   n= 59
##
##         CP nsplit rel error xerror   xstd
## 1 0.58808      0    1.0000 1.0318 0.2538
## 2 0.08293      1    0.4119 0.5763 0.1671
## 3 0.02714      2    0.3290 0.5514 0.1335
## 4 0.01000      3    0.3018 0.5088 0.1194
##
## Variable importance
## gratio displa weight    mpg length  rseat   turn Origin
##     26     23     18     16     11      6      1      1
##
## Node number 1: 59 observations,    complexity param=0.5881
##   mean=6355, MSE=8.876e+06
##   left son=2 (51 obs) right son=3 (8 obs)
##   Primary splits:
##       gratio < 2.5   to the right, improve=0.5881, (0 missing)
##       weight < 3785  to the left,  improve=0.4732, (0 missing)
##       displa < 334   to the left,  improve=0.4717, (0 missing)
##       mpg    < 17.5  to the right, improve=0.3464, (0 missing)
##       rseat  < 29.25 to the left,  improve=0.2946, (0 missing)
##   Surrogate splits:
##       displa < 334   to the left,  agree=0.983, adj=0.875, (0 split)
##       weight < 3890  to the left,  agree=0.949, adj=0.625, (0 split)
##       mpg    < 15.5  to the right, agree=0.932, adj=0.500, (0 split)
##       length < 219   to the left,  agree=0.915, adj=0.375, (0 split)
##       rseat  < 29.75 to the left,  agree=0.898, adj=0.250, (0 split)
##
## Node number 2: 51 observations,    complexity param=0.08293
##   mean=5450, MSE=3.166e+06
##   left son=4 (43 obs) right son=5 (8 obs)
##   Primary splits:
##       mpg    < 17.5  to the right, improve=0.2690, (0 missing)
##       Origin splits as  LRL,       improve=0.1848, (0 missing)
##       length < 171   to the left,  improve=0.1359, (0 missing)
##       rseat  < 28.75 to the left,  improve=0.1287, (0 missing)
##       trunk  < 11.5  to the left,  improve=0.1212, (0 missing)
##   Surrogate splits:
##       weight < 3715  to the left,  agree=0.941, adj=0.625, (0 split)
##       length < 205   to the left,  agree=0.902, adj=0.375, (0 split)
##       turn   < 44.5  to the left,  agree=0.902, adj=0.375, (0 split)
```

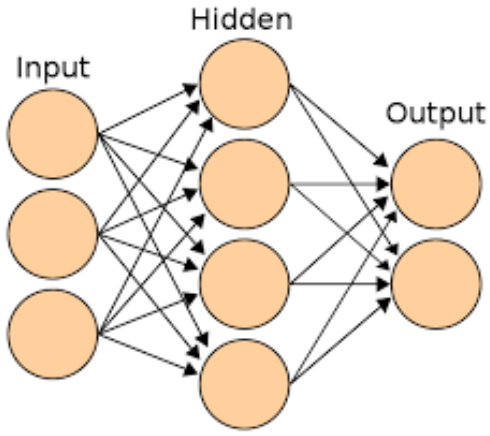
```
##       displa < 254   to the left,  agree=0.902, adj=0.375, (0 split)
##       gratio < 2.72  to the right, agree=0.863, adj=0.125, (0 split)
##
## Node number 3: 8 observations
##   mean=1.212e+04, MSE=6.784e+06
##
## Node number 4: 43 observations,    complexity param=0.02714
##   mean=5052, MSE=1.567e+06
##   left son=8 (29 obs) right son=9 (14 obs)
##   Primary splits:
##       gratio < 3.455 to the left,  improve=0.21090, (0 missing)
##       Origin splits as  LRL,       improve=0.14180, (0 missing)
##       length < 169.5 to the left,  improve=0.12790, (0 missing)
##       mpg    < 27    to the right, improve=0.11570, (0 missing)
##       hroom  < 2.75  to the right, improve=0.08827, (0 missing)
##   Surrogate splits:
##       displa < 127.5 to the right, agree=0.837, adj=0.500, (0 split)
##       Origin splits as  LRR,       agree=0.837, adj=0.500, (0 split)
##       weight < 2090  to the right, agree=0.791, adj=0.357, (0 split)
##       mpg    < 22.5  to the left,  agree=0.744, adj=0.214, (0 split)
##       length < 156.5 to the right, agree=0.744, adj=0.214, (0 split)
##
## Node number 5: 8 observations
##   mean=7590, MSE=6.329e+06
##
## Node number 8: 29 observations
##   mean=4653, MSE=4.931e+05
##
## Node number 9: 14 observations
##   mean=5880, MSE=2.777e+06

plot(car_pfit)
text(car_pfit, use.n=TRUE, all=TRUE, cex=.7)
```




```
car_data_test$pr_predict<-predict(car_pfit,car_data_test,type="vector")
plot(car_data_test$Price,car_data_test$pr_predict)
```


car_data_test$pr_predict
12000
10000
8000
6000
4000
6000
8000
10000
12000
car_data_test$Price

Classification using Neural Networks



Wine Data

```
wine_train$Cult<-as.factor(wine_train$Cult)
wine_nnet<-nnet(Cult ~ Alc+MalAcid+Ash+AshAlk+Mag+TotPhen+Flav
                +NonFlav+Proant+Color+Hue+OD280OD315+Proline,
                data = wine_train,size=5,decay=0.1)

## # weights:  88
## initial  value 165.653483
## iter  10 value 154.816037
## iter  20 value 143.415144
## iter  30 value 130.957549
## iter  40 value 126.164260
## iter  50 value 79.456330
## iter  60 value 59.199426
## iter  70 value 26.793134
## iter  80 value 22.177088
## iter  90 value 20.855061
## iter 100 value 20.555531
## final  value 20.555531
## stopped after 100 iterations

wine_test$pred_nnet<-predict(wine_nnet,wine_test,type="class")
table(wine_test$Cult,wine_test$pred_nnet)

##
##      1  2  3
##   1 11  1  0
##   2  0 15  1
```

```
##   3  0  0  8
```

In-class exercise: Swiss Bank Notes

The data in Notes.csv contain various characteristics of 100 genuine and 100 counterfeit Swiss bank notes. The characteristics include:

- Length of the bank note
- Height of the bank note, measured on the left
- Height of the bank note, measured on the right
- Distance of inner frame to the lower border
- Distance of inner frame to the upper border
- Length of the diagonal

Observations 1-100 are the genuine bank notes and the other 100 observations are the counterfeit bank notes. Use logistic regression, classification trees and Neural Networks to develop rules that discriminate between the notes.

In-class exercise: Credit Card Scoring

The data in creditcard.csv contain information obtained from credit card applications. For the purpose of confidentiality, the names and values of the attributes of the data set have been coded. These dat have been obtained from here: `https://archive.ics.uci.edu/ml/datasets/Statlog+(Australian+Credit+Approval)`

The variables are as follows:

- A1: Categorical variable (0,1)
- A2: Continuous variable
- A3: Continuous variable
- A4: Categorical variable (1,2,3)
- A5: Categorical variable (1,2,3,4,5,6,7,8,9,10,11,12,13,14)
- A6: Categorical variable (1,2,3,4,5,6,7,8,9)
- A7: Continuous variable

- A8: Categorical variable (0,1)
- A9: Categorical variable (0,1)
- A10: Continuous variable
- A11: Categorical variable (0,1)
- A12: Categorical variable (1,2,3)
- A13: Continuous variable
- A14: Continuous variable
- A15: Prediction variable (1,2)

Use variables A1–A14 to develop a classification rule for predicting A15.

Exercises

Use logistic regression, classification trees and Neural Networks to develop classification rules for

- the bankrupcy data in Table 11.4
- the iris data in Table 11.5
- the admission data in Table 11.6

of Johnson and Wichern. Experiment with different proportions of training and test data set sizes (as a percentage of observations in each group), and determine the ability of the models built on the training data to predict for the test data.