Lab 12

Daniel Tshiani

2025-06-13

```
library(tree)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

1

```
load("../data/Auto-3.rda")
attach(Auto)
```

```
## The following object is masked from package:lubridate:
##
##     origin
```

```
## The following object is masked from package:ggplot2:
##
##     mpg
```

a

```
Auto$ECO <- ifelse(mpg > median(mpg), "Economy", "Consuming")
table(Auto$ECO)
```

```
##
## Consuming   Economy
##       196       196
```

```
glimpse(Auto)
```

```
## Rows: 392
## Columns: 10
## $ mpg          <dbl> 18, 15, 18, 16, 17, 15, 14, 14, 14, 15, 15, 14, 15, 14, 2~
## $ cylinders    <dbl> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 6, 6, 6, 4, ~
## $ displacement <dbl> 307, 350, 318, 304, 302, 429, 454, 440, 455, 390, 383, 34~
```

```
## $ horsepower   <dbl> 130, 165, 150, 150, 140, 198, 220, 215, 225, 190, 170, 16~
## $ weight       <dbl> 3504, 3693, 3436, 3433, 3449, 4341, 4354, 4312, 4425, 385~
## $ acceleration <dbl> 12.0, 11.5, 11.0, 12.0, 10.5, 10.0, 9.0, 8.5, 10.0, 8.5, ~
## $ year         <dbl> 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 7~
## $ origin       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 3, ~
## $ name         <fct> chevrolet chevelle malibu, buick skylark 320, plymouth sa~
## $ ECO          <chr> "Consuming", "Consuming", "Consuming", "Consuming", "Cons~
```

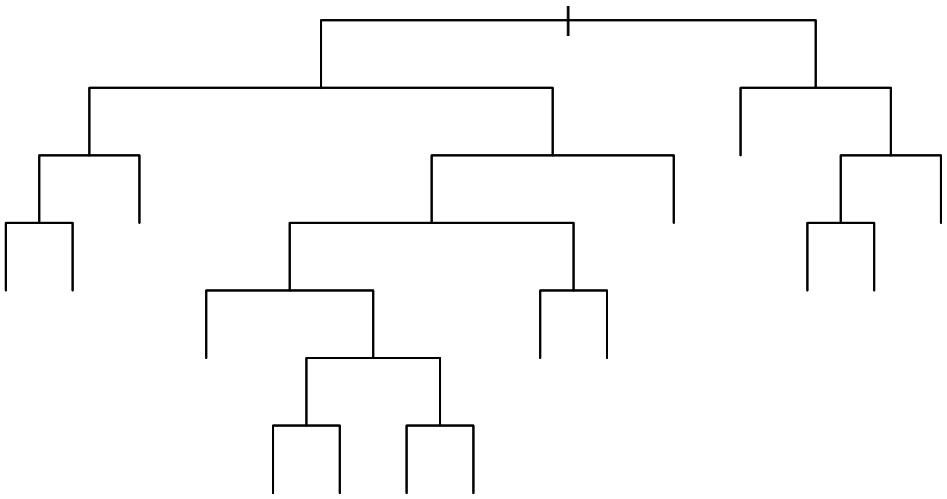
```
Auto$ECO <- as.factor(Auto$ECO)
```

```
tree(ECO ~ .-name, Auto)
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
## 1) root 392 543.4 Consuming ( 0.5 0.5 )
##   2) mpg < 22.75 196   0.0 Consuming ( 1.0 0.0 ) *
##   3) mpg > 22.75 196   0.0 Economy ( 0.0 1.0 ) *
```

b

```
tree_model <- tree(ECO ~ .-name -mpg, Auto)
plot(tree_model, type = "uniform")
```



```
summary(tree_model)
```

```
##
## Classification tree:
## tree(formula = ECO ~ . - name - mpg, data = Auto)
## Variables actually used in tree construction:
## [1] "displacement" "horsepower"   "year"         "weight"       "acceleration"
## Number of terminal nodes:  15
## Residual mean deviance:  0.16 = 60.3 / 377
## Misclassification error rate: 0.04592 = 18 / 392
```

the misclassification rate is about 4%

c

```
set.seed(123)
train_indices <- sample(1:nrow(Auto), nrow(Auto)/2)
train_data <- Auto[train_indices, ]
test_data <- Auto[-train_indices, ]
```

```
train_model <- tree(ECO ~ . -name - mpg, data = train_data)

test_preds <- predict(train_model, newdata= test_data, type = "class")

conf_mat <- table(Predicted = test_preds, Actual = test_data$ECO)
conf_mat
```

```
##            Actual
## Predicted   Consuming Economy
##   Consuming        84       7
##   Economy          15      90
```

```
accuracy <- sum(diag(conf_mat)) / sum(conf_mat)
accuracy
```

```
## [1] 0.8877551
```

the accuracy is about 88% which is less than the previous accuracy. the previous accuracy was about 96%.

d

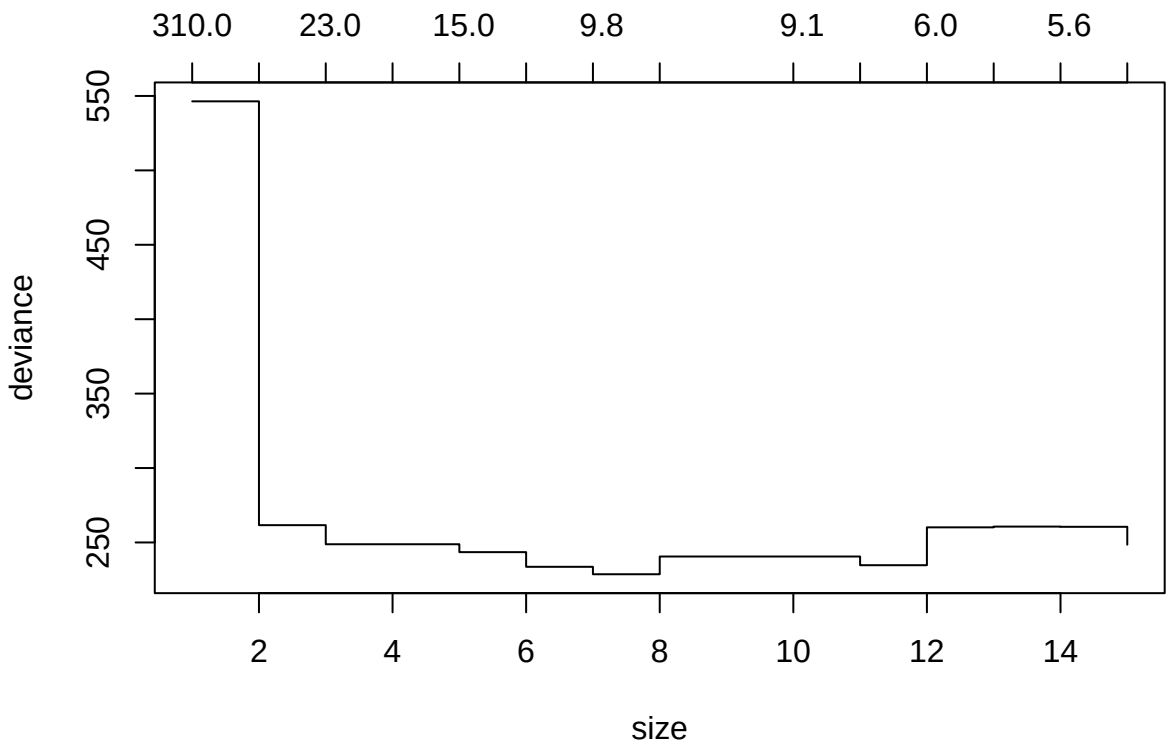
```
cv <- cv.tree(tree_model)
cv
```

```
## $size
##  [1] 15 14 13 12 11 10  8  7  6  5  4  3  2  1
##
## $dev
##  [1] 248.4622 260.5094 260.6479 260.1427 234.7036 240.5529 240.5529 228.6400
##  [9] 233.6362 243.4912 248.7987 248.7987 261.6184 546.3731
##
## $k
##  [1]       -Inf   5.594483   5.827839   6.011820   7.896077   9.113858
##  [7]   9.161949   9.831019  12.605665  14.720942  22.928434  23.373834
## [13]  38.496957 308.400711
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
cv$size[which.min(cv$dev)]
```

```
## [1] 7
```

```
plot(cv)
```



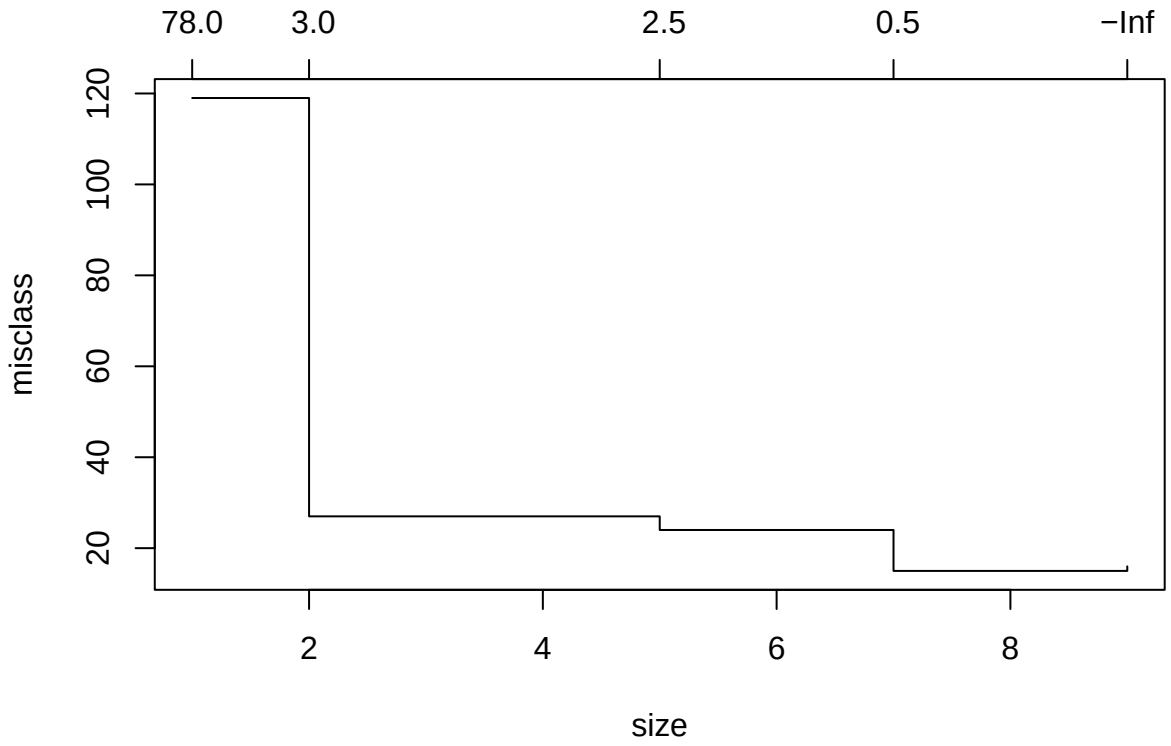
optimal complexity of a tree is 15.

e

```
cv_m <- cv.tree(train_model, FUN = prune.misclass)
cv_m
```

```
## $size
## [1] 9 7 5 2 1
##
## $dev
## [1]  16  15  24  27 119
##
## $k
## [1] -Inf  0.5  2.5  3.0 78.0
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv_m)
```



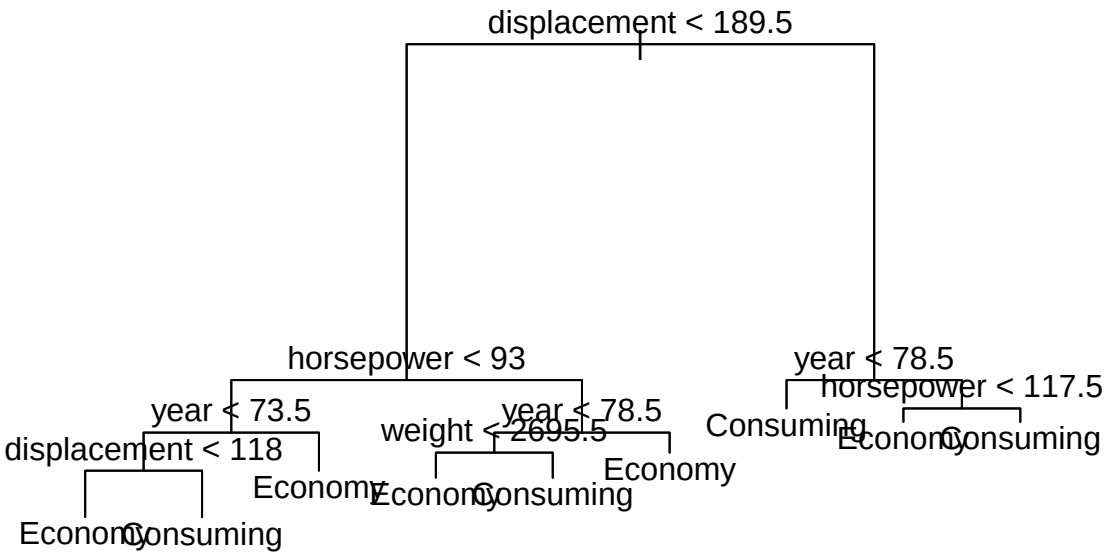

```
cv_m$size[which.min(cv$dev)]
```

```
## [1] NA
```

optimal complexity would be at 9 nodes

f

```
opt_m <- prune.misclass(train_model, best = 9)
plot(opt_m)
text(opt_m)
```



2

```
load("../data/Auto-3.rda")
attach(Auto)
```

```
## The following objects are masked from Auto (pos = 3):
##
##     acceleration, cylinders, displacement, horsepower, mpg, name,
##     origin, weight, year
```

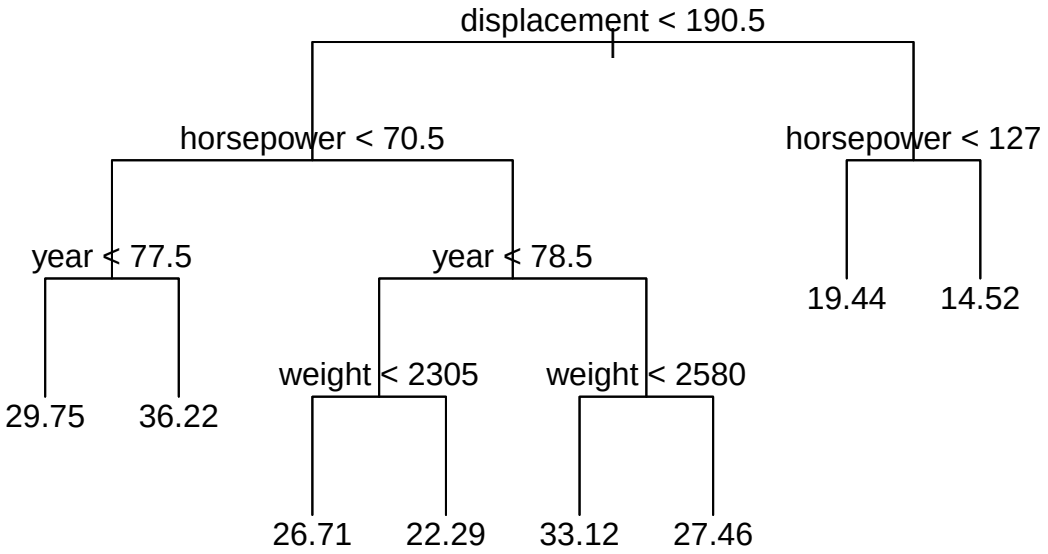
```
## The following object is masked from package:lubridate:
##
##     origin
```

```
## The following object is masked from package:ggplot2:
##
##     mpg
```

```
tree.mpg <- tree(mpg ~ .-name-origin + as.factor(origin), Auto)
tree.mpg
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 392 23820.0 23.45
##    2) displacement < 190.5 222  7786.0 28.64
##      4) horsepower < 70.5 71  1804.0 33.67
##        8) year < 77.5 28   280.2 29.75 *
##        9) year > 77.5 43   814.5 36.22 *
##      5) horsepower > 70.5 151  3348.0 26.28
##       10) year < 78.5 94  1222.0 24.12
##         20) weight < 2305 39   362.2 26.71 *
##         21) weight > 2305 55   413.7 22.29 *
##       11) year > 78.5 57   963.7 29.84
##         22) weight < 2580 24   294.2 33.12 *
##         23) weight > 2580 33   225.0 27.46 *
##    3) displacement > 190.5 170  2210.0 16.66
##      6) horsepower < 127 74   742.0 19.44 *
##      7) horsepower > 127 96   457.1 14.52 *
```

```
plot(tree.mpg, type = "uniform"); text(tree.mpg)
```




```
summary(tree.mpg)
```

```
##
## Regression tree:
## tree(formula = mpg ~ . - name - origin + as.factor(origin), data = Auto)
## Variables actually used in tree construction:
## [1] "displacement" "horsepower"   "year"         "weight"
## Number of terminal nodes:  8
## Residual mean deviance:  9.346 = 3589 / 384
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -9.4170 -1.5190 -0.2855  0.0000  1.7150 18.5600
```