Lab 12

Exercise 1 - Classification Trees

For this lab we will use the `Auto.rda` dataset. In addition we should use `library(tree)` and thus we have to install the `tree` package.

a) Define a categorical variable `ECO` by using `ECO = ifelse( mpg > median(mpg), "Economy", "Consuming" )`. Then include `ECO` into the dataset and check how many observations belong to the `Economy` as well as `Consuming` class. Use the main `tree` command and classify `ECO` based on `mpg`. Hint: The `R` code for this task is `tree( ECO ~ .-name, Cars )`.

b) Classifying `ECO` based on mpg is trivial! The tree picks this obvious split immediately. Thus exclude `mpg` and predict `ECO` based on the car's technical characteristics. Visualize the tree by using `plot(tree.fit, type="uniform")`. In addition, by using `summary`, display the misclassification rate. What do you observe?

c) Estimate the correct classification rate by cross-validation. For simplicity, use 50% for training and 50% for testing. What do you observe in terms of the accuracy? How does this compare to the previous question?

d) Use cross-validation to determine the optimal complexity of a tree and the number of terminal nodes that minimizes the deviance. The built-in cross-validation function in order to determine the optimal complexity of a tree is `cv.tree`.

e) Instead of optimizing by the smallest deviance, optimize the complexity and the number of terminal nodes by the smallest mis-classification error.

f) Prune the tree to the optimal size obtain by e).

Exercise 2 - Regression Trees (This question is just for demonstration)

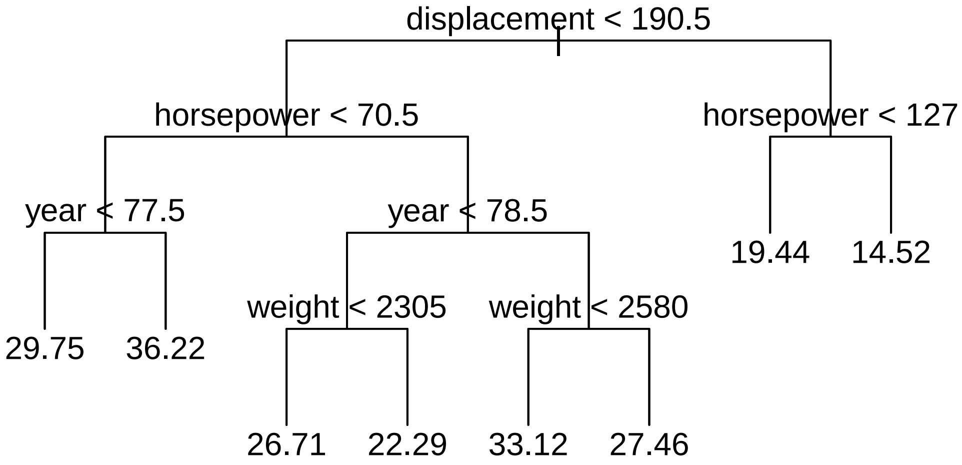
```
load("Auto.rda")
attach(Auto)
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.0.5
```

```
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
```