ML2: Exam Preparation

Code

Denis Baskan

10 March 2019

Exercise: Tree Models

In the following code, a Regression tree and Classification tree will be applied. Some parts are from the book James et. al. Lab 8.3.2.

Load Packages

If packages are not installed, then they will be installed.

Hide

```
#load needed packages; if not installed, then do so
required.packages =  c('MASS')#c('forecast', 'tseries', 'ggplot2', 'gsubfn','Metrics')#,'prophet
load.packages <- function(packages){

  for (pckg in packages){
    if (!(pckg %in% installed.packages()[,"Package"])){
      install.packages(pckg)
    }

    library(pckg, character.only = TRUE)
  }
}
load.packages(required.packages)
```

Show Boston data set and some scatterplots

The data set contains data on housing values and other information about Boston suburbs.

Hide

```
#The data set contains data on housing values and other information about Boston suburbs.
?Boston
head(Boston)
```

Hide

```
cat("Number of rows: ", nrow(Boston))
```

```
Number of rows:  506
```

Hide

```
cat("Number of columns: ", ncol(Boston))
```

```
Number of columns:  14
```

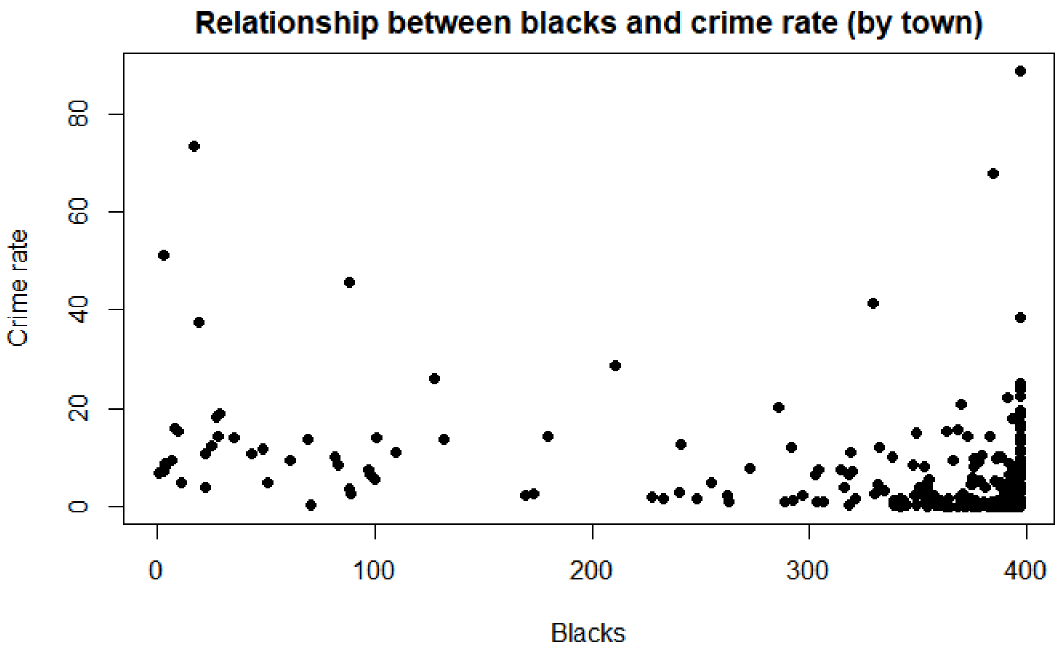
Hide

```
cat("Column names: ", colnames(Boston))
```

```
Column names:  crim zn indus chas nox rm age dis rad tax ptratio black lstat medv
```

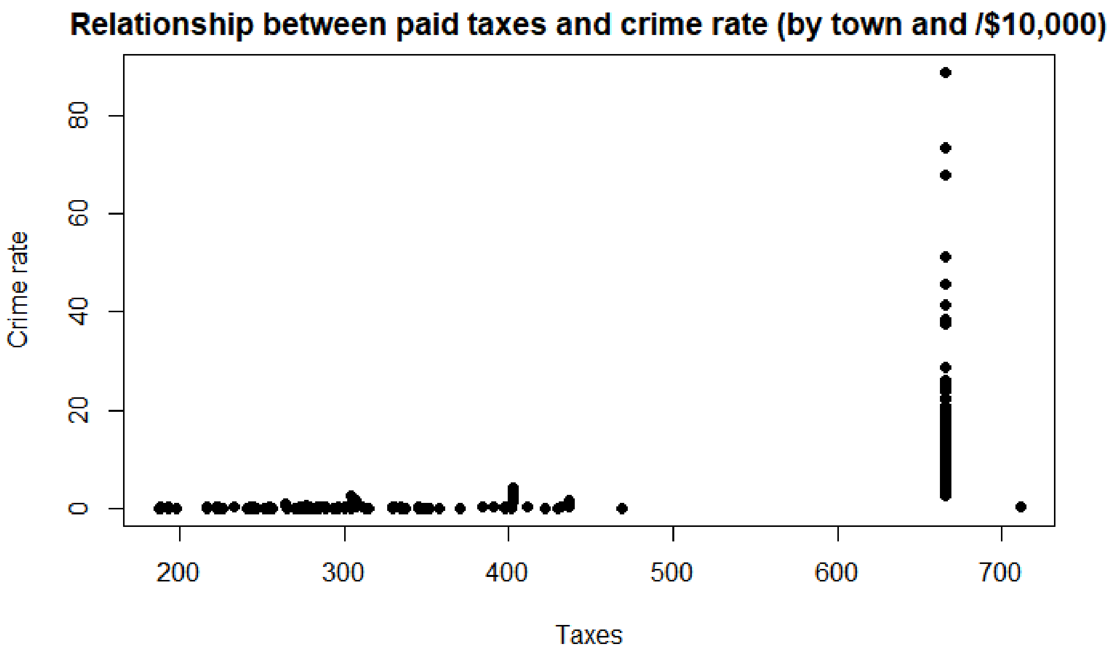
Hide

```
plot(black,crim,main="Relationship between blacks and crime rate (by town)", xlab="Blacks", ylab="Crime rate
", pch=19)
```



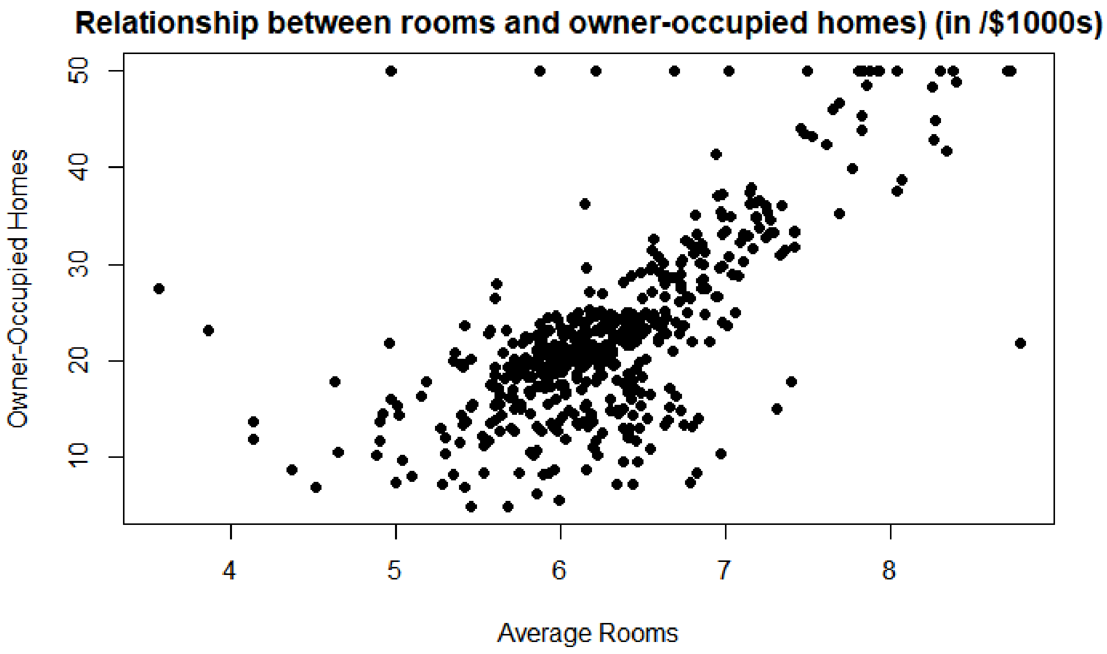
Hide

```
plot(tax,crim,main="Relationship between paid taxes and crime rate (by town and /$10,000)", xlab="Taxes", yl
ab="Crime rate", pch=19)
```



Hide

```
plot(rm,medv,main="Relationship between rooms and owner-occupied homes) (in /$1000s)", xlab="Average Rooms",
ylab="Owner-Occupied Homes", pch=19)
```



Any suburbs with particularly high crime rate? Tax rates? Pupil-teacher ratios?

Hide

```
#x-axis has no meaning

plot(Boston[order(Boston$crim),]$crim,main="Crime Rate", xlab="Suburb", ylab="Crime rate (by town)", pch=19,
type='l')
plot(Boston[order(Boston$tax),]$tax,main="Taxes", xlab="Suburb", ylab="Taxes", pch=19,type='l')
plot(Boston[order(Boston$ptratio),]$ptratio,main="Pupil-Teacher ratio", xlab="Suburb", ylab="Pupil-Teacher r
atio", pch=19,type='l')
```

One can observe suburbs with really high values and almost exponential slope for the column crime rate.

Some Statistics

Hide

```
cat("Number of suburbs bound the Charles river: ",sum(Boston$chas==1))
```

```
Number of suburbs bound the Charles river:  35
```

Hide

```
cat("Median value of pupil-teacher ratio : ",median(Boston$ptratio))
```

```
Median value of pupil-teacher ratio :  19.05
```

Hide

```
cat("Lowest median value of owner-occupied homes : ",min(Boston$medv))
```

```
Lowest median value of owner-occupied homes :  5
```

Hide

```
cat("Number of suburbs with more than 7 rooms per dwelling: ",sum(Boston$rm > 7))
```

```
Number of suburbs with more than 7 rooms per dwelling:  64
```

Hide

```
cat("Number of suburbs with more than 8 rooms per dwelling: ",sum(Boston$rm > 8))
```

```
Number of suburbs with more than 8 rooms per dwelling:  13
```

Hide

```
#compare some data
Boston[Boston$medv==min(Boston$medv),]
```

Hide

```
print("These 2 suburbs have high values for the predictors crime, black, tax, pt-ratio.")
```

```
[1] "These 2 suburbs have high values for the predictors crime, black, tax, pt-ratio."
```

The data with more than 8 rooms have a low crime rate and high values for age, tax, black for example. The 2 suburbs with the lowest medv values have high values for the predictors crime, black, tax, pt-ratio. One can conclude that populations with a lower status and cheap houses are at increased risk of being a crime victim.

Fit a Regression tree using column medv as outcome variable

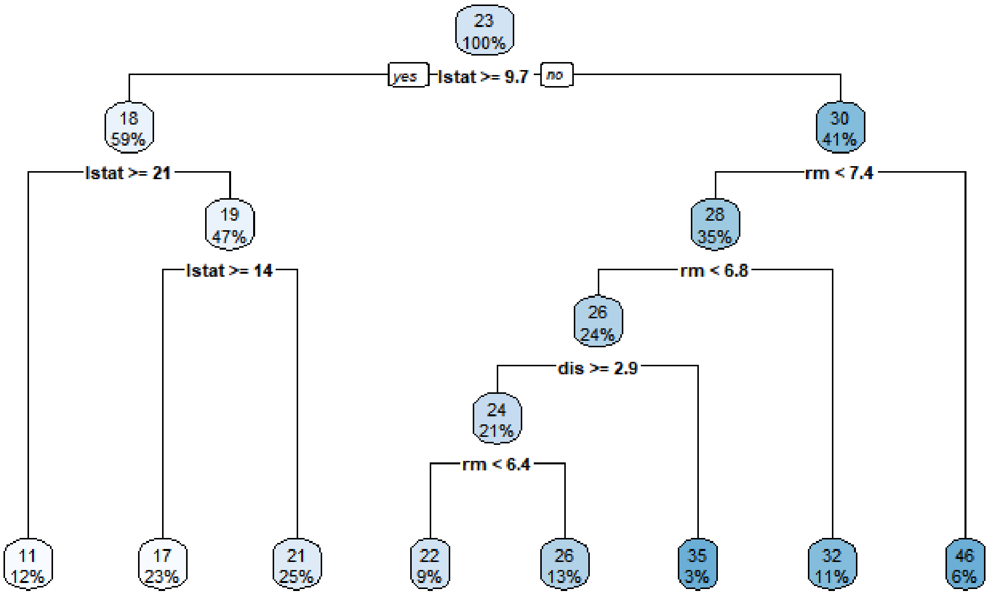
Hide

```
#Note: outcome variable should be continuous. Exercise follows Lab 8.3.2 in James et al.
set.seed(1) #set a fix random generator to reproduce the same results next time
train = sample(1:nrow(Boston), nrow(Boston)/2)   #split data randomly
tree.boston = rpart(medv ~.,Boston,subset=train) # create a tree
print(tree.boston) #only 3 variables were used
```

```
n= 253

node), split, n, deviance, yval
      * denotes terminal node

 1) root 253 20894.66000 22.67312
   2) lstat>=9.715 150  3464.71500 17.55133
     4) lstat>=21.49 30   311.88970 11.10333 *
     5) lstat< 21.49 120  1593.69900 19.16333
      10) lstat>=14.48 58   743.28220 17.15690 *
      11) lstat< 14.48 62   398.48920 21.04032 *
   3) lstat< 9.715 103  7764.58400 30.13204
     6) rm< 7.437 89  3310.16000 27.57640
      12) rm< 6.7815 61  1994.62200 25.52131
        24) dis>=2.85155 54   544.00830 24.32778
          48) rm< 6.36 22    47.17864 21.82273 *
          49) rm>=6.36 32   263.86000 26.05000 *
        25) dis< 2.85155 7   780.27430 34.72857 *
      13) rm>=6.7815 28   496.64960 32.05357 *
     7) rm>=7.437 14   177.84360 46.37857 *
```

Hide

```
rpart.plot(tree.boston)
```



The tree indicates that a higher socioeconomic status leads in buying more expensive houses. A median house price of $46,000 can be observed when lstat is lower than 9.7% and number of rooms are higher than 7.4 on average.

Can pruning the tree improve our model?

Rule: Choose the smalles number of nodes (largest cp value) which lies within 1 std. dev. of the smallest deviance, i.e. lies below the dotted line.

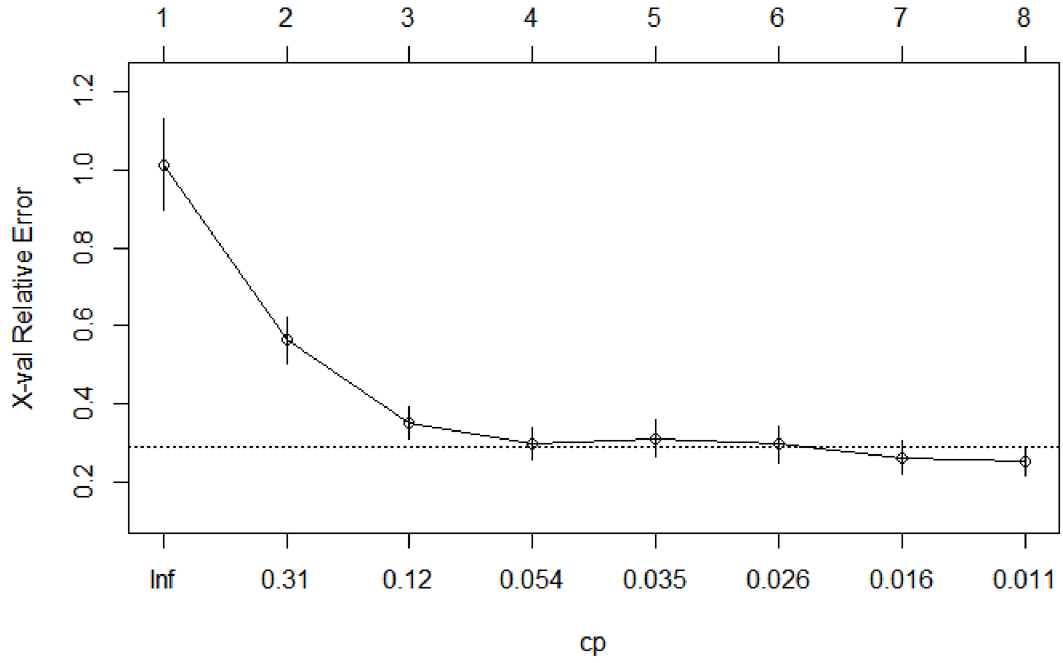
Hide

```
printcp(tree.boston)
```

```
Regression tree:
rpart(formula = medv ~ ., data = Boston, subset = train)

Variables actually used in tree construction:
[1] dis   lstat rm

Root node error: 20895/253 = 82.588

n= 253

        CP nsplit rel error  xerror     xstd
1 0.462576      0   1.00000 1.01273 0.117728
2 0.204673      1   0.53742 0.56339 0.059699
3 0.074618      2   0.33275 0.34949 0.039945
4 0.039191      3   0.25813 0.29608 0.039976
5 0.032082      4   0.21894 0.31072 0.048076
6 0.021629      5   0.18686 0.29508 0.048053
7 0.011150      6   0.16523 0.26146 0.042604
8 0.010000      7   0.15408 0.25032 0.037014
```

Hide

```
plotcp(tree.boston)
```



cp=0.016 lies below the dotted line and could improve our model

Prune the tree and compare models

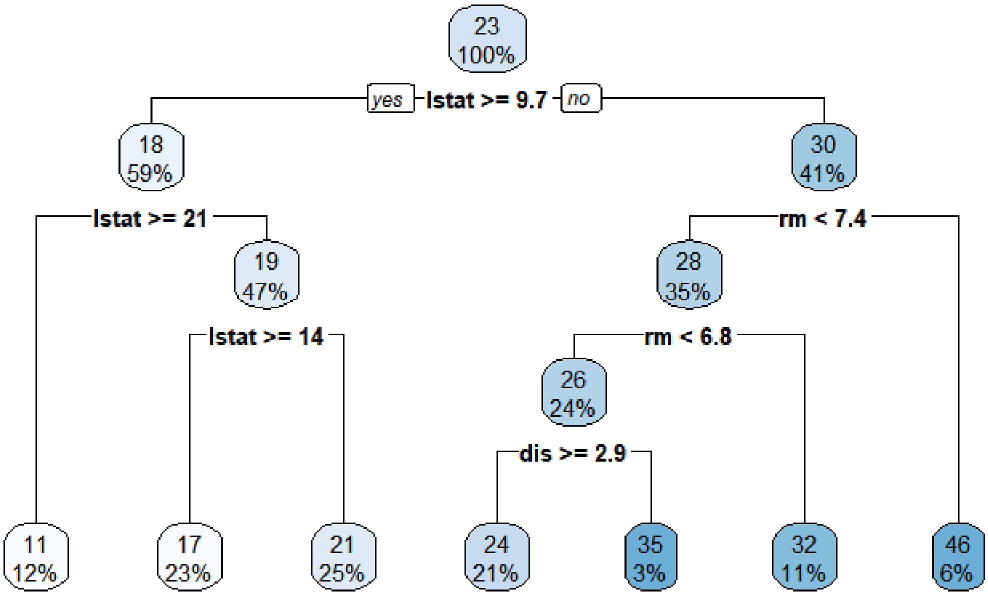
Hide

```
prune.boston = prune(tree.boston,cp=0.016)
prune.boston
```

```
n= 253

node), split, n, deviance, yval
      * denotes terminal node

 1) root 253 20894.6600 22.67312
   2) lstat>=9.715 150  3464.7150 17.55133
     4) lstat>=21.49 30   311.8897 11.10333 *
     5) lstat< 21.49 120  1593.6990 19.16333
      10) lstat>=14.48 58   743.2822 17.15690 *
      11) lstat< 14.48 62   398.4892 21.04032 *
   3) lstat< 9.715 103  7764.5840 30.13204
     6) rm< 7.437 89  3310.1600 27.57640
      12) rm< 6.7815 61  1994.6220 25.52131
        24) dis>=2.85155 54   544.0083 24.32778 *
        25) dis< 2.85155 7   780.2743 34.72857 *
      13) rm>=6.7815 28   496.6496 32.05357 *
     7) rm>=7.437 14   177.8436 46.37857 *
```

Hide

```
rpart.plot(prune.boston)
```



Hide

```
#compare models by calculating MSE (Mean Squarred Error)
pred.train<-predict(tree.boston,newdata=Boston[train,])
mean((Boston$medv[train]-pred.train)^2)
```

```
[1] 12.72517
```

Hide

```
pred.train.prune<-predict(prune.boston,newdata=Boston[train,])
mean((Boston$medv[train]-pred.train.prune)^2)
```

```
[1] 13.646
```

Calculate the MSE for the test set

Hide

```
pred.test<-predict(tree.boston,newdata=Boston[-train,])
mean((Boston$medv[-train]-pred.test)^2)
```

```
[1] 25.35825
```

Hide

```
pred.test<-predict(prune.boston,newdata=Boston[-train,])
mean((Boston$medv[-train]-pred.test)^2)
```

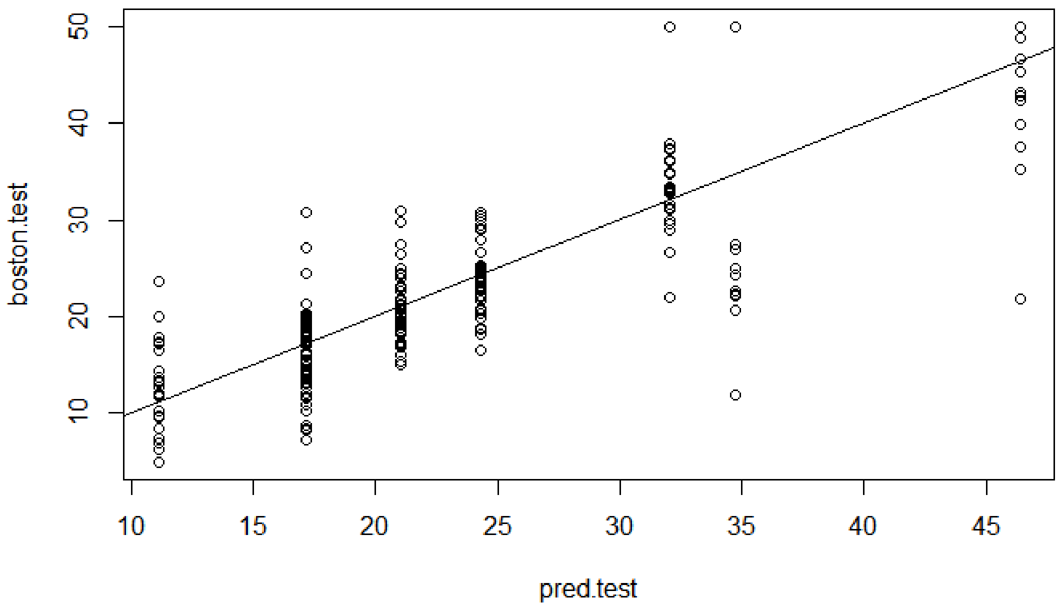
```
[1] 25.82207
```

Pruned tree performes slightly worse applied on train and test set, but we gained a simpler model. Taking the square root of the test set MSE gives $5,000 rounded. That’s the range where test prediction lay in.

Plot observed median values medv agains predictions

Hide

```
boston.test=Boston[-train,"medv"]
plot(pred.test,boston.test)
abline(c(0,1))
```



boston.test
50
40
30
20
10
10
15
20
25
30
35
40
45
pred.test