Decision Trees for Regression

Karen Mazidi

Try linear regression on Boston

We get a correlation of 0.9 and a rmse of 4.35.

```
library(tree)
library(MASS)
names(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

```
# divide into train and test
set.seed(1234)
i <- sample(nrow(Boston), 0.8*nrow(Boston), replace = FALSE)
train <- Boston[i,]
test <- Boston[-i,]
lm1 <- lm(medv~., data=train)
summary(lm1)
```

```
##
## Call:
## lm(formula = medv ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.537  -2.913  -0.546   1.848  24.915
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  44.900577   6.016980   7.462 5.59e-13 ***
## crim         -0.085000   0.049892  -1.704  0.08924 .
## zn            0.047219   0.015849   2.979  0.00307 **
## indus         0.038249   0.070942   0.539  0.59008
## chas          2.724575   0.966685   2.818  0.00507 **
## nox         -19.139048   4.382515  -4.367 1.62e-05 ***
## rm            2.949428   0.479982   6.145 1.98e-09 ***
## age          -0.007757   0.015670  -0.495  0.62087
## dis          -1.558391   0.224867  -6.930 1.75e-11 ***
## rad           0.302988   0.076673   3.952 9.21e-05 ***
## tax          -0.012284   0.004206  -2.920  0.00370 **
## ptratio      -1.008491   0.152951  -6.594 1.40e-10 ***
## black         0.008717   0.003345   2.606  0.00951 **
## lstat        -0.555420   0.056482  -9.834  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.884 on 390 degrees of freedom
## Multiple R-squared:  0.7203, Adjusted R-squared:  0.711
## F-statistic: 77.28 on 13 and 390 DF,  p-value: < 2.2e-16
```

```
pred <- predict(lm1, newdata=test)
cor(pred, test$medv)
```

```
## [1] 0.900081
```

```
rmse_lm <- sqrt(mean((pred-test$medv)^2))
```

Using tree

Correlation was 0.8433 and rmse was 5.14.

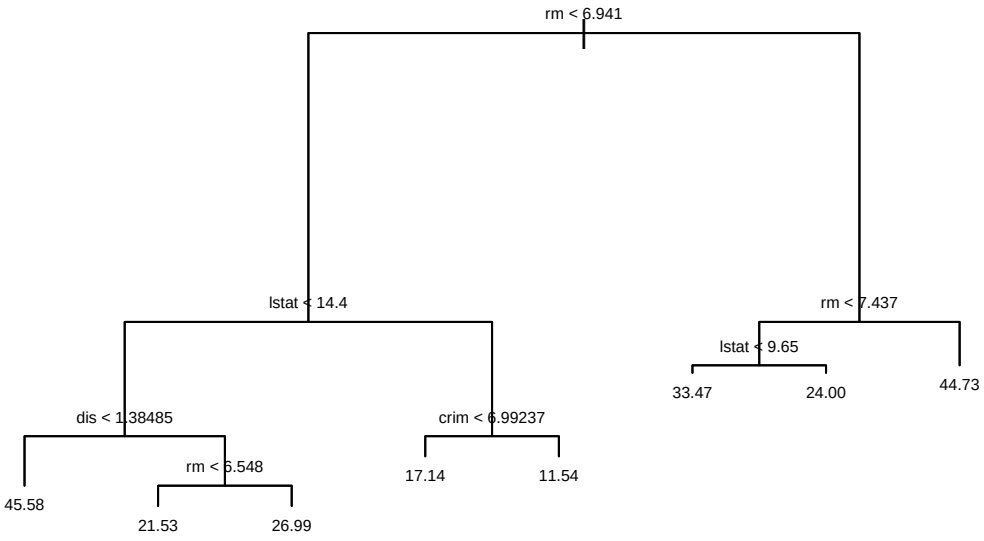
```
tree1 <- tree(medv~., data=train)
summary(tree1)
```

```
##
## Regression tree:
## tree(formula = medv ~ ., data = train)
## Variables actually used in tree construction:
## [1] "rm"    "lstat" "dis"   "crim"
## Number of terminal nodes:  8
## Residual mean deviance:  13.36 = 5292 / 396
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -22.830  -2.031   0.212   0.000   2.265  14.670
```

```
pred <- predict(tree1, newdata=test)
cor(pred, test$medv)
```

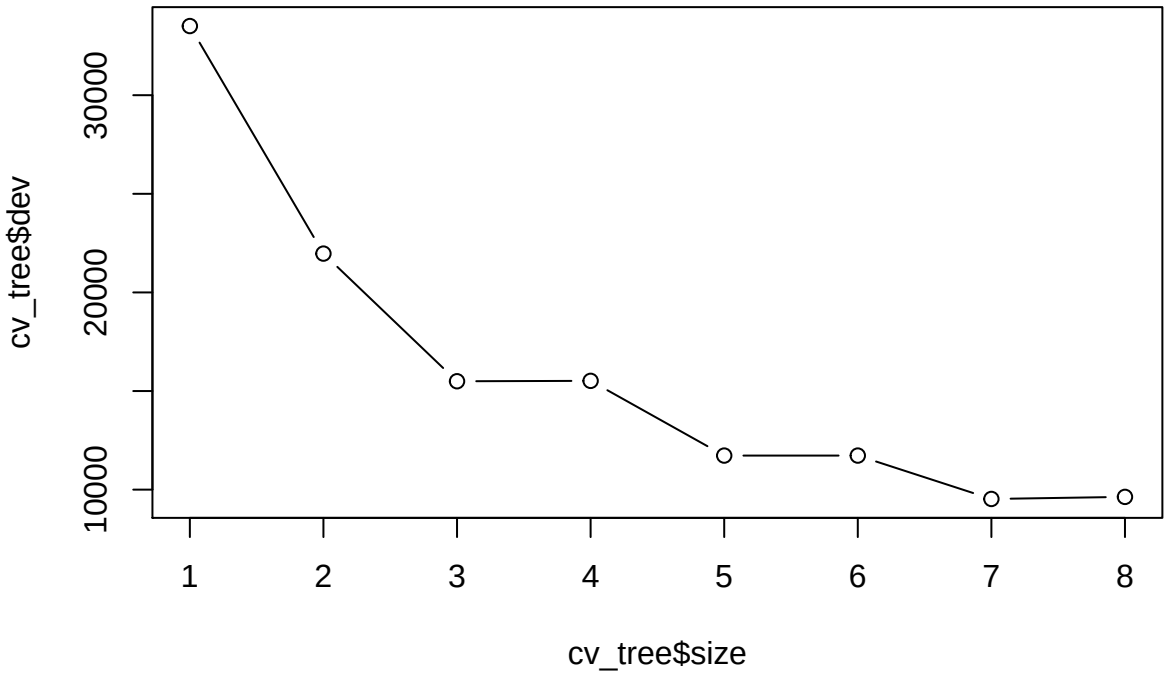
```
## [1] 0.8913526
```

```
rmse_tree <- sqrt(mean((pred-test$medv)^2))
plot(tree1)
text(tree1, cex=0.5, pretty=0)
```



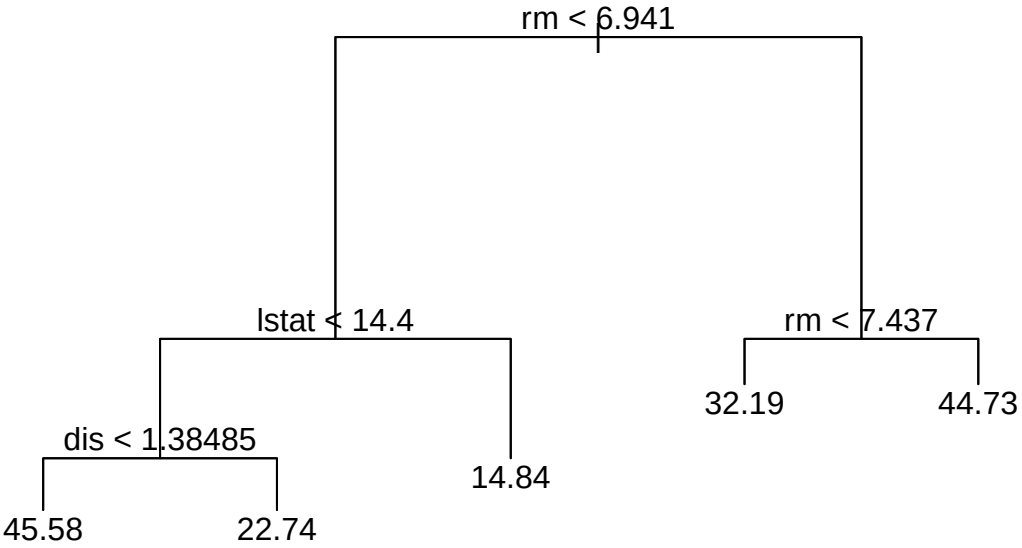
cross validation

```
cv_tree <- cv.tree(tree1)
plot(cv_tree$size, cv_tree$dev, type='b')
```



prune the tree

```
tree_pruned <- prune.tree(tree1, best=5)
plot(tree_pruned)
text(tree_pruned, pretty=0)
```



test on the pruned tree

The cor is now 0.845, very slightly above the unpruned tree but still lower than linear regression. The rmse is 5.18, very similar to the unpruned tree but higher than linear regression.

In this case pruning did not improve results on the test data but the tree is simpler and easier to interpret.

```
pred_pruned <- predict(tree_pruned, newdata=test)
cor(pred_pruned, test$medv)
```

```
## [1] 0.8456787
```

```
rmse_pruned <- sqrt(mean((pred_pruned-test$medv)^2))
```

bagging

The importance=TRUE argument tells the algorithm to consider the importance of predictors. This effectively is the same as bagging.

```
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(1234)
tree_bagged <- randomForest(medv~., data=train, importance=TRUE)
tree_bagged
```

```
##
## Call:
##  randomForest(formula = medv ~ ., data = train, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 10.84628
##                     % Var explained: 86.83
```

predict on the bagged tree

Now the correlation is much higher than even linear regression and the rmse is almost half.

```
pred_bag <- predict(tree_bagged, newdata=test)
cor(pred_bag, test$medv)
```

```
## [1] 0.9619739
```

```
rmse_bag <- sqrt(mean((pred_bag-test$medv)^2))
```

random forest

Removing argument importance=TRUE will result in a random forest.

```
tree_forest <- randomForest(medv~., data=train)
tree_forest
```

```
##
## Call:
##  randomForest(formula = medv ~ ., data = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 10.73234
##                     % Var explained: 86.97
```

predict

Our results for the random forest were slightly lower than for the bagging.

```
pred_forest <- predict(tree_forest, newdata=test)
cor(pred_forest, test$medv)
```

```
## [1] 0.9607446
```

```
rmse_rforest <- sqrt(mean((pred_forest-test$medv)^2))
```