732A95tenta

Mitt tentainlogg

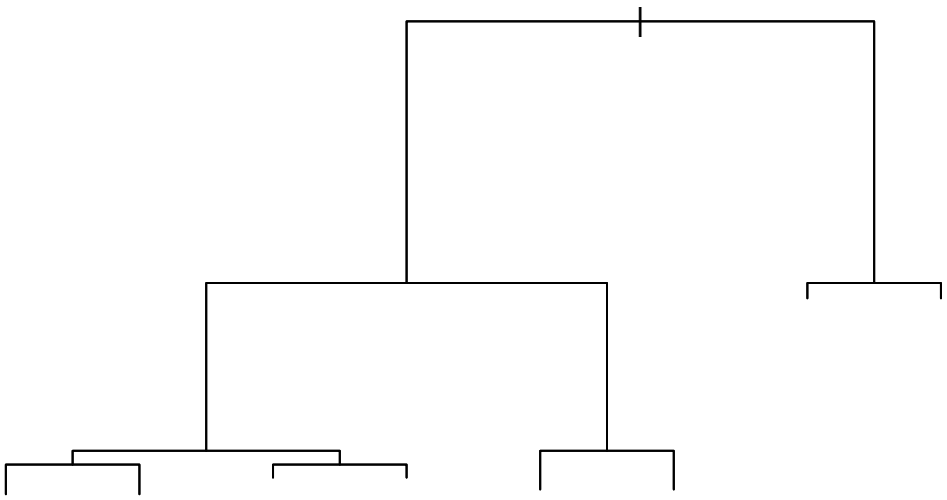
4 January 2017

Assignment 1

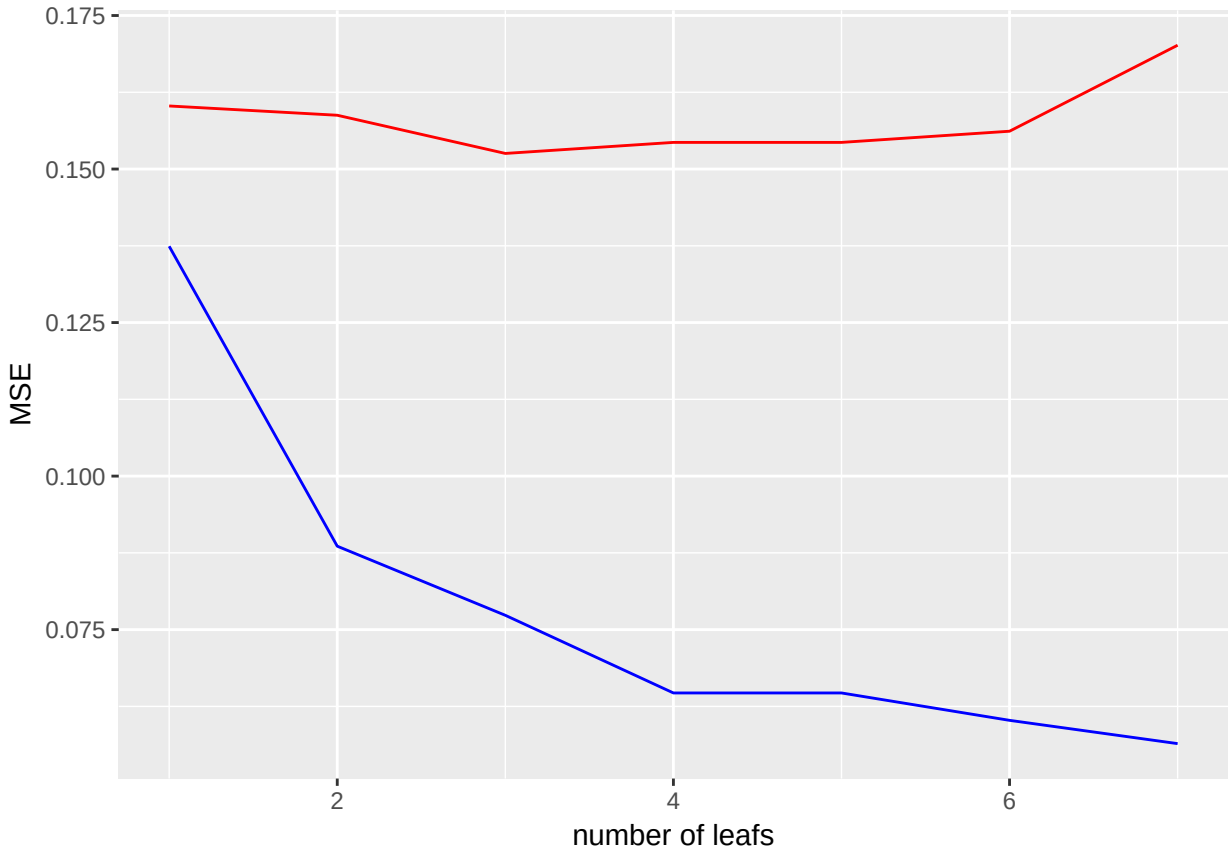
1.1

```
glass <- read.csv2("glass.csv")

set.seed(12345)
glass <- glass[sample(nrow(glass), replace = FALSE),]

train <- glass[1:107,]
test  <- glass[108:(108+53),]
valid  <- glass[(108+54):214,]

library(tree)


glass.tree <- tree(formula = Al ~ ., data = train, split = "deviance" )
plot(glass.tree)
no.leafs <- data.frame(trainS=1,testS = 1)
rad <- 1

for (i in 2:8) {
  pruned.glass    <-  prune.tree(glass.tree, best = i)
   no.leafs[rad,1] <-  mean( (train$Al - predict(pruned.glass))^2 )
   no.leafs[rad,2] <-   mean((valid$Al - predict( pruned.glass , newdata = valid, type = "vector"))^2)
  #
   # no.leafs[rad,1] <-  deviance( pruned.glass )
   # no.leafs[rad,2] <-  2 * deviance(predict( pruned.glass , newdata = valid, type = "tree"))
  rad <- rad + 1
}


library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```



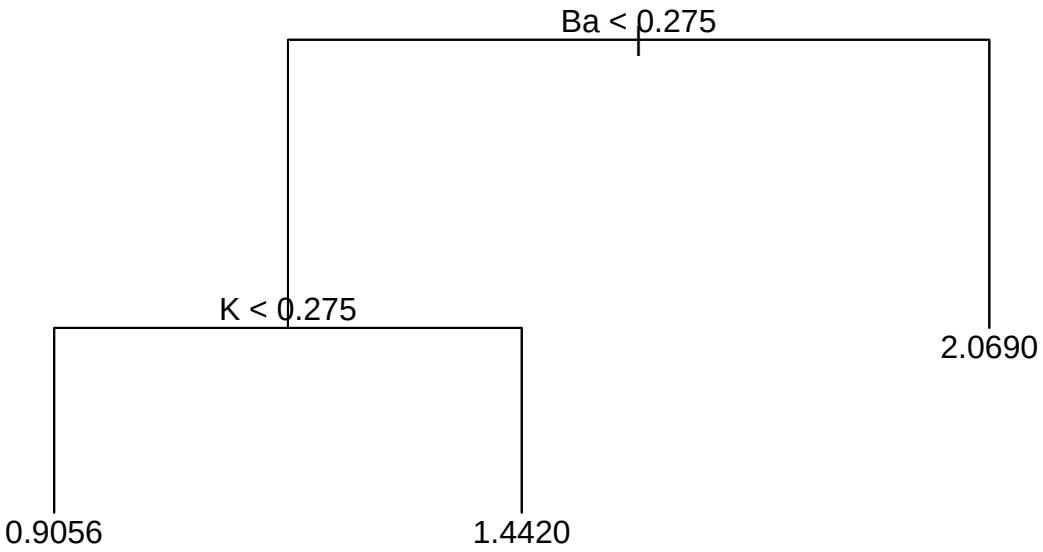
```
ggplot(data = no.leafs) + geom_line(aes(x = 1:7, y = trainS),color = "blue") +
  geom_line(aes(x = 1:7, y = testS),color = "red") + labs(x="number of leafs" ,y = "MSE")
```



The optimal number of leafs in the regression tree is three since it gives the lowest Validation error (red line). The blue line represents the MSE for the training data set and the line seems to decrese very slowly when the number of leafs are increased. Meanwhile the validation error decreses in the begining but then increases. This is the bias -variance trade off because we continue to try to minimize the MSE for the training set the model becomes overfitted and loses its predictive power since it becomes less general and more specific to the data.

1.2

```
best.glass <- prune.tree(glass.tree, best = 3)
plot(best.glass)
text(best.glass)
```




```
testerror <-  mean( ( predict(pruned.glass,newdata = test) - test$Al )^2 )
print(paste("The test error was calculated to:",testerror))
```

```
## [1] "The test error was calculated to: 0.149578325316183"
```

The chosen variables are K and Ca, im to bad at chemistry to remember what they represent but

1.3

a)

```
library(pls)
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
myplsr <- plsr(formula = Al ~ ., data = train, validation  = c("CV") )

summary(myplsr)
```

```
## Data:    X dimension: 107 7
##  Y dimension: 107 1
## Fit method: kernelpls
## Number of components considered: 7
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
```

```
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.4665   0.4105   0.3912   0.3651   0.3205   0.2501   0.1290
## adjCV       0.4665   0.4091   0.3909   0.3614   0.3200   0.2433   0.1278
##        7 comps
## CV      0.1299
## adjCV   0.1287
##
## TRAINING: % variance explained
##     1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X     35.86    72.19    91.13    97.80    99.09    99.87   100.00
## Al    31.98    43.73    56.75    64.93    86.86    94.41    94.43
```

One would need 3 components to explain over 90 % of the variation in the feature-space ### b)

c)

```
myplsr$validation$PRESS
```

```
##     1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## Al 18.03282 16.37788 14.26178 10.99356 6.692535 1.781594 1.806053
```

All 7 components are needed according to the PRESS since 7 comps has the lowest value.

d)

e)

```
myplsr$coefficients
```

```
## , , 1 comps
##
##              Al
## Na  0.008835006
## Mg -0.164090892
## Si  0.031871940
## K   0.021049512
## Ca -0.099759609
## Ba  0.070091664
## Fe -0.002373527
##
## , , 2 comps
##
##              Al
## Na -0.105885813
## Mg -0.182492794
## Si  0.002343184
## K   0.072212317
## Ca -0.217019847
## Ba  0.099860617
## Fe -0.002280949
##
## , , 3 comps
##
```

```
##              Al
## Na -0.319668176
## Mg -0.291170650
## Si -0.089021787
## K   0.130519805
## Ca -0.235569000
## Ba  0.121716799
## Fe -0.002092571
##
## , , 4 comps
##
##              Al
## Na -0.35680800
## Mg -0.39452422
## Si -0.37771865
## K   0.10196171
## Ca -0.32887585
## Ba  0.04919988
## Fe -0.01411537
##
## , , 5 comps
##
##              Al
## Na -0.67225793
## Mg -0.84586015
## Si -0.79251966
## K  -0.32352053
## Ca -0.76055484
## Ba -0.84334600
## Fe -0.06466132
##
## , , 6 comps
##
##            Al
## Na -0.9174895
## Mg -0.9381058
## Si -0.9508007
## K  -0.9950250
## Ca -0.9371439
## Ba -0.8615871
## Fe -0.1062298
##
## , , 7 comps
##
##            Al
## Na -0.9189207
## Mg -0.9370586
## Si -0.9492225
## K  -0.9928497
## Ca -0.9359308
## Ba -0.8611291
## Fe -0.1963429
```

f)

```
mean((test$Al - predict(myplsr, newdata = test))^2)
```

```
## [1] 0.09611379
```

1.4

The PLSR have a lower test-error and therefor have a better predictive power. Because we want to estimate the prediction power of the model one observation is a very small sample and is probably similar to the training set so that it would have a high variance.

Assignment 2