Trees and Forests

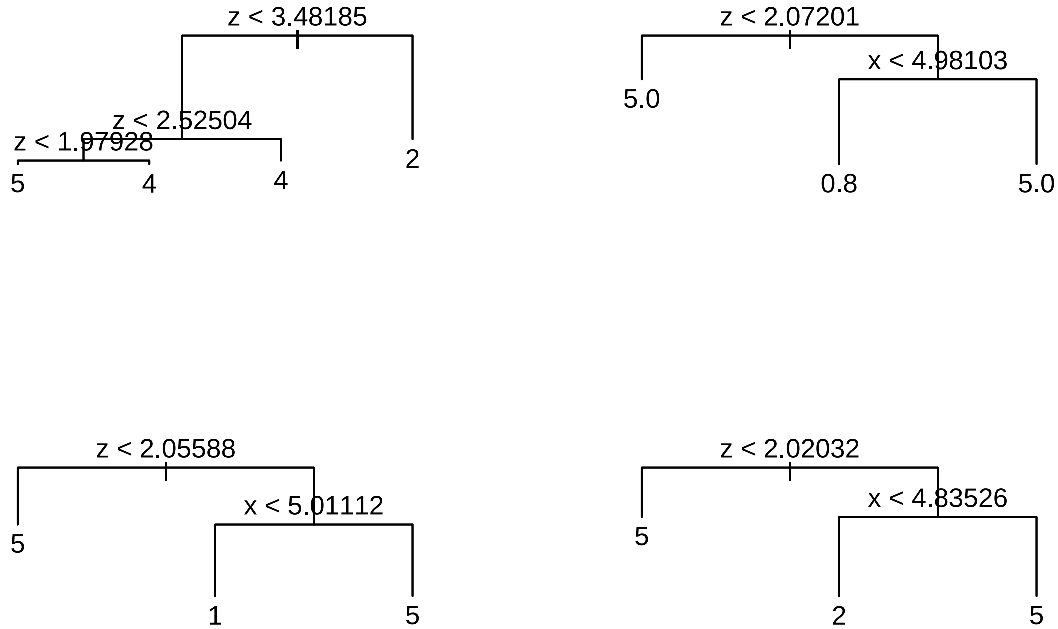
Munir Winkel

```
###   Using Trees and Random Forests in R
###   A Brief Tutorial
###   by Munir Winkel
###     10 / 31 / 2014


rm( list=( ls() ) )

### If packages are not installed, install them with this code:


  if (!is.element("tree", installed.packages()[, 1])) {
    install.packages("tree", repos = "http://cran.us.r-project.org")
        }
  library(tree)

  if (!is.element("randomForest", installed.packages()[, 1])) {
    install.packages("randomForest")
        }
 library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
if (!is.element("rgl", installed.packages()[, 1])) {
    install.packages("rgl")
        }
 library(rgl)
### The above code is based off of code by Dr. Hua Zhou




#### A Quick Example with Regression Trees

x <- rnorm(500,5,0.5)
z <- rexp(500)


### Seeing some variability in regression trees
### The ideal decision tree would say:
  #   if ( x > 5 or z < 2 ) then  a = 5
  #   otherwise a = 1

a <- ifelse(
   ( x > 5) |
   ( z < 2)
   , rnorm(500,5,.25) , rnorm(500,1,.75)
    )
```

```r
plot3d(a,z,x,size=6,col="blue")

##
## A function that creates training sets
### build = training set
### pure = test set

teachit <- function(x,percent=55){

    train <-sample(1:nrow(x), floor(dim(x)[1]  *percent/100))

    ### Pure has never been seen before
    pure <<- x[-train,]

    ### Build is what we'll use to build our model
    build <<- x[train,]
}




### Creating a Data Frame
frame <- data.frame(matrix(cbind(x,z,a),ncol=3))
colnames(frame) <- c("x","z","output")
par(mfrow=c(2,2))

### A simple loop to show that the trees change, depending on the data

set.seed(51)

for (i in 1:4){

  ### sampling data at __%
teachit(frame,percent=16)

test <- tree ( output ~ x + z , data=build)
plot(test)
text(test)
}
```




```
set.seed(0)
```

```
rm( list=( ls() ) )

## Writing that "Build Training Set" function again
teachit <- function(x,percent=55){

    train <-sample(1:nrow(x), floor(dim(x)[1]  *percent/100))

    ### Pure has never been seen before
    pure <<- x[-train,]

    ### Build is what we'll use to build our model
    build <<- x[train,]
}


### Beginning with birth name data from New York State
### Department of Health

setwd("~/Statistics Presentations")
x <- read.csv("longname.csv",header=TRUE)
# x <- read.csv("harder.csv",header=TRUE)

### Getting a look at the data
### When
```

```r
table(x[,1])
```

```
##
## 2007 2008 2009 2010 2011 2012
## 5814 6266 4691 2327 5237 3480
```

```r
### What names?
table(x[,2])
```

```
##
##  ALEXIS   AMARI   ANGEL   ARIEL   AVERY   BLAKE CAMERON   CASEY   DYLAN
##    1943     175    2682     728    1311     393    1523     155    4735
## EMERSON  HAYDEN  JAIDEN  JAYDEN  JAYLIN  JORDAN  KAMARI   LOGAN  PARKER
##      96     274     127    2563      92    2432      18     703     102
##  PEYTON PHOENIX   QUINN   REESE   RILEY    RYAN    SHEA  SHILOH  SKYLER
##     674      22     132      49    1548    5033      17      23      11
##  TENZIN
##     254
```

```r
### What gender?
table(x[,2],x[,3])
```

```
##
##              F    M
##   ALEXIS  1466  477
##   AMARI     23  152
##   ANGEL    168 2514
##   ARIEL    334  394
##   AVERY   1118  193
##   BLAKE     23  370
##   CAMERON   72 1451
##   CASEY    107   48
##   DYLAN    132 4603
##   EMERSON   43   53
##   HAYDEN    74  200
##   JAIDEN     6  121
##   JAYDEN    27 2536
##   JAYLIN    62   30
##   JORDAN   209 2223
##   KAMARI     6   12
##   LOGAN      5  698
##   PARKER    11   91
##   PEYTON   562  112
##   PHOENIX   10   12
##   QUINN     71   61
##   REESE     44    5
##   RILEY   1148  400
##   RYAN      37 4996
##   SHEA       7   10
##   SHILOH    11   12
##   SKYLER     6    5
##   TENZIN   123  131
```

```r
### What do the data look like?
x[110:130,]
```

```
##     year firstname sex
## 110 2007    ALEXIS   M
## 111 2007    ALEXIS   M
## 112 2007    ALEXIS   M
## 113 2007    ALEXIS   M
## 114 2007    ALEXIS   M
## 115 2007    ALEXIS   M
## 116 2007    ALEXIS   M
## 117 2007    ALEXIS   F
## 118 2007    ALEXIS   F
## 119 2007    ALEXIS   F
## 120 2007    ALEXIS   F
## 121 2007    ALEXIS   F
## 122 2007    ALEXIS   F
## 123 2007    ALEXIS   F
## 124 2007    ALEXIS   F
## 125 2007    ALEXIS   F
## 126 2007    ALEXIS   F
## 127 2007    ALEXIS   F
## 128 2007    ALEXIS   F
## 129 2007    ALEXIS   F
## 130 2007    ALEXIS   F
```

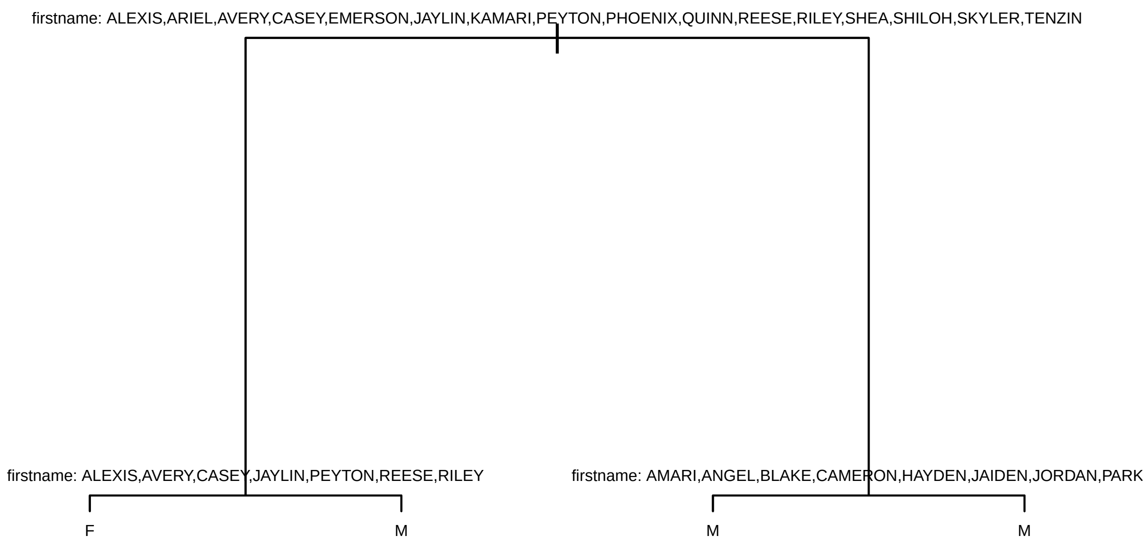
```r
dim(x)
```

```
## [1] 27815     3
```

```r
#### LEARNING MOMENT
#### WHY IS THIS BAD ?

badtree <- tree(sex ~ firstname + year , data=x)
summary(badtree)
```

```
##
## Classification tree:
## tree(formula = sex ~ firstname + year, data = x)
## Variables actually used in tree construction:
## [1] "firstname"
## Number of terminal nodes:  4
## Residual mean deviance:  0.507 = 14100 / 27800
## Misclassification error rate: 0.0957 = 2663 / 27815
```

```r
plot(badtree)

text(badtree,pretty=0,cex=0.5)
```




```
teachit(x,percent=35)


## What does it look like?
head(build)
```

```
##       year firstname sex
## 24942 2008      RYAN   M
## 7385  2007   CAMERON   M
## 10350 2008     DYLAN   M
## 15933 2008    JAYDEN   M
## 25259 2009      RYAN   M
## 5609  2007     AVERY   F
```

```
dim(build)
```

```
## [1] 9735    3
```

```
table(build[,2],build[,3])
```

```
##
##             F   M
##   ALEXIS  531 174
##   AMARI     7  60
```

```
##   ANGEL     63  857
##   ARIEL    122  148
##   AVERY    377   68
##   BLAKE      4  146
##   CAMERON   23  499
##   CASEY     42   13
##   DYLAN     53 1638
##   EMERSON   18   25
##   HAYDEN    19   76
##   JAIDEN     4   39
##   JAYDEN     5  847
##   JAYLIN    25   12
##   JORDAN    69  752
##   KAMARI     2    5
##   LOGAN      3  255
##   PARKER     3   37
##   PEYTON   201   38
##   PHOENIX    5    2
##   QUINN     27   23
##   REESE      8    4
##   RILEY    422  138
##   RYAN      12 1738
##   SHEA       3    4
##   SHILOH     4    1
##   SKYLER     1    3
##   TENZIN    35   45
```

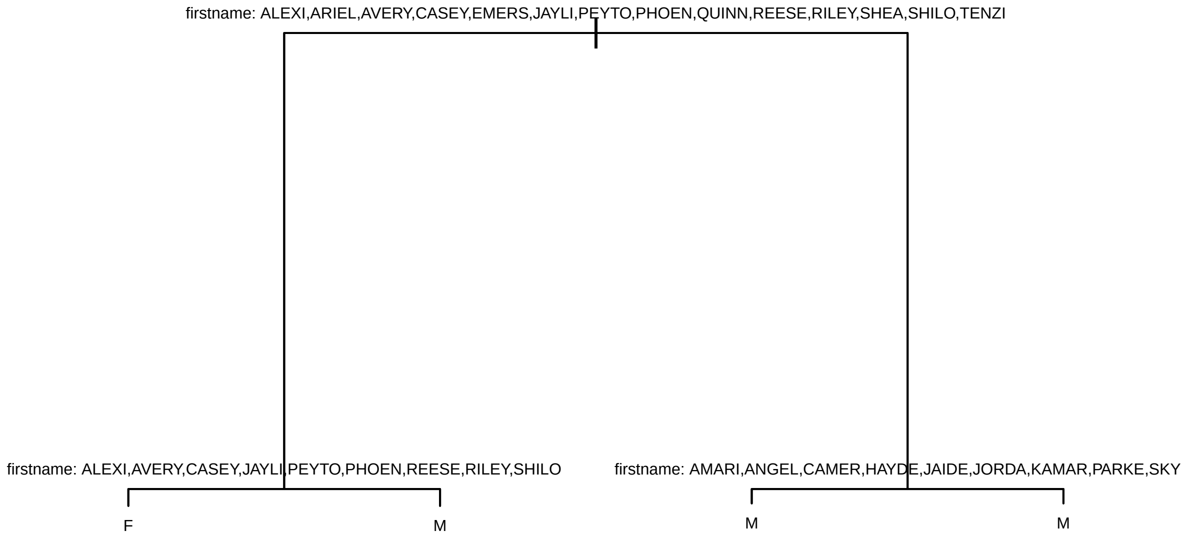
```
### Fitting a Tree to the Training Set


stree <- tree(sex ~ firstname , data = build)
summary(stree)
```

```
##
## Classification tree:
## tree(formula = sex ~ firstname, data = build)
## Number of terminal nodes:  4
## Residual mean deviance:  0.506 = 4920 / 9730
## Misclassification error rate: 0.0948 = 923 / 9735
```

```
stree
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
## 1) root 9735 10000 M ( 0.21 0.79 )
##   2) firstname: ALEXIS,ARIEL,AVERY,CASEY,EMERSON,JAYLIN,PEYTON,PHOENIX,QUINN,REESE,RILEY,SHEA,SHILOH
##     4) firstname: ALEXIS,AVERY,CASEY,JAYLIN,PEYTON,PHOENIX,REESE,RILEY,SHILOH 2065  2000 F ( 0.78 0.
##     5) firstname: ARIEL,EMERSON,QUINN,SHEA,TENZIN 450   600 M ( 0.46 0.54 ) *
##   3) firstname: AMARI,ANGEL,BLAKE,CAMERON,DYLAN,HAYDEN,JAIDEN,JAYDEN,JORDAN,KAMARI,LOGAN,PARKER,RYAN
##     6) firstname: AMARI,ANGEL,CAMERON,HAYDEN,JAIDEN,JORDAN,KAMARI,PARKER,SKYLER 2519  1000 M ( 0.08 (
##     7) firstname: BLAKE,DYLAN,JAYDEN,LOGAN,RYAN 4701   800 M ( 0.02 0.98 ) *
```

```r
plot(stree)
text(stree,pretty=5,cex=0.5)
```




```r
### Predicting using this tree

predtree <- predict(stree,pure,type="class")


### How well did we do? Note: uses "pure" data
tab <- table(predtree,pure$sex); tab
```

```
##
## predtree     F     M
##        F  2913   839
##        M   904 13424
```

```r
### Success
tab1 <- round(sum(diag(tab))/sum(tab),4)
tab1
```

```
## [1] 0.9036
```

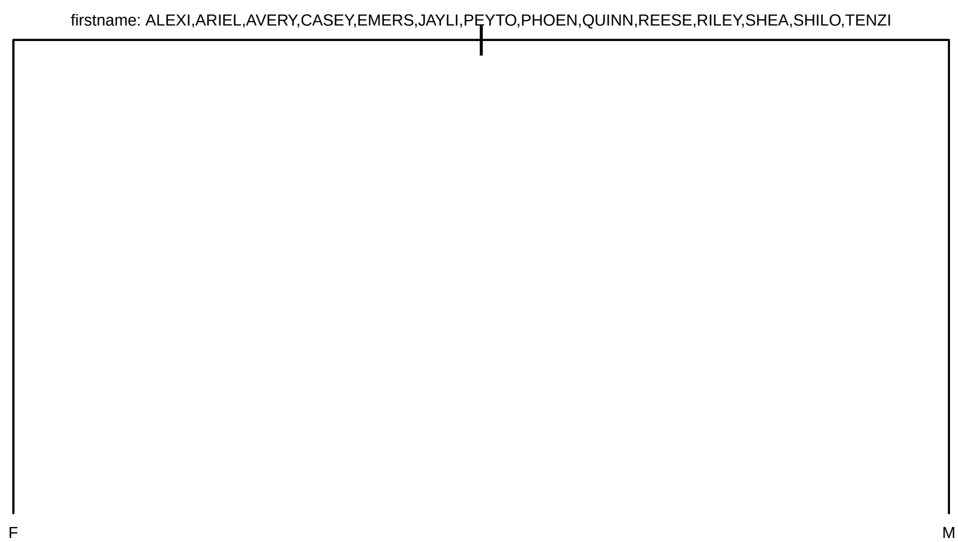
```
### Pruning the tree
pruned <- prune.misclass(stree , best=2)
pruned
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
## 1) root 9735 10000 M ( 0.21 0.79 )
##   2) firstname: ALEXIS,ARIEL,AVERY,CASEY,EMERSON,JAYLIN,PEYTON,PHOENIX,QUINN,REESE,RILEY,SHEA,SHILOH
##   3) firstname: AMARI,ANGEL,BLAKE,CAMERON,DYLAN,HAYDEN,JAIDEN,JAYDEN,JORDAN,KAMARI,LOGAN,PARKER,RYAN
```

```
plot(pruned)
print(pruned)
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
## 1) root 9735 10000 M ( 0.21 0.79 )
##   2) firstname: ALEXIS,ARIEL,AVERY,CASEY,EMERSON,JAYLIN,PEYTON,PHOENIX,QUINN,REESE,RILEY,SHEA,SHILOH
##   3) firstname: AMARI,ANGEL,BLAKE,CAMERON,DYLAN,HAYDEN,JAIDEN,JAYDEN,JORDAN,KAMARI,LOGAN,PARKER,RYAN
```

```
text(pruned, pretty=5 , cex=0.5)
```



```
### How well did we do? Note: uses "pure" data
predprune <- predict(pruned , pure, type="class")
ptab <- table(predprune, pure$sex); ptab
```

```
##
## predprune     F     M
##         F  3198  1157
##         M   619 13106
```

```
### Success
ptab1 <- round(sum(diag(ptab))/sum(ptab),4)
ptab1
```

```
## [1] 0.9018
```

```
#### Random Forest Approach

forest <- randomForest(sex ~ year + firstname,
                     data=build,
                     ntree=75,
                    importance = TRUE)


pforest <- predict( forest , newdata=pure )

## Looking at Results
ftab <- table( pforest  ,  pure$sex )
ftab
```

```
##
## pforest     F     M
##       F  3045   957
##       M   772 13306
```

```
ftab1 <- round( sum( diag( ftab ))/sum( ftab ) , 4 )
ftab1
```

```
## [1] 0.9044
```

```
summary(forest)
```

```
##                 Length Class  Mode
## call                5  -none- call
## type                1  -none- character
## predicted        9735  factor numeric
## err.rate          225  -none- numeric
## confusion           6  -none- numeric
## votes           19470  matrix numeric
## oob.times        9735  -none- numeric
## classes             2  -none- character
```

```
## importance          8  -none- numeric
## importanceSD        6  -none- numeric
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             14  -none- list
## y                9735  factor numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
## terms               3  terms  call
```

```
print(forest)
```

```
##
## Call:
##  randomForest(formula = sex ~ year + firstname, data = build,      ntree = 75, importance = TRUE)
##               Type of random forest: classification
##                     Number of trees: 75
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 9.48%
## Confusion matrix:
##      F    M class.error
## F 1677  411     0.19684
## M  512 7135     0.06695
```

```
### What About Logistic Regression?

logist <- glm(sex ~ firstname + year,
              family = binomial( logit ),
              data = build)

### Predicted Coefficients
coefficients(logist)
```

```
##     (Intercept)   firstnameAMARI   firstnameANGEL   firstnameARIEL
##       8.2205814        3.2641187        3.7275486        1.3110890
##   firstnameAVERY   firstnameBLAKE firstnameCAMERON   firstnameCASEY
##      -0.5920531        4.7244763        4.1950483       -0.0551732
##   firstnameDYLAN firstnameEMERSON  firstnameHAYDEN  firstnameJAIDEN
##       4.5483281        1.4479583        2.5011830        3.3927181
##  firstnameJAYDEN  firstnameJAYLIN  firstnameJORDAN  firstnameKAMARI
##       6.2408327        0.3785875        3.5053017        2.0270944
##   firstnameLOGAN  firstnamePARKER  firstnamePEYTON firstnamePHOENIX
##       5.5673963        3.6370507       -0.5461477        0.2131014
##   firstnameQUINN   firstnameREESE   firstnameRILEY    firstnameRYAN
##       0.9648661        0.4176565        0.0001647        6.0892109
##    firstnameSHEA  firstnameSHILOH  firstnameSKYLER  firstnameTENZIN
##       1.3984858       -0.2754906        2.2280044        1.3701379
##            year
##      -0.0046471
```

```
### What is it modeleing? Probability of "Male"
log2 <- (glm(sex ~ NULL,
              family=binomial( logit ),
              data=build))
coefficients(log2)
```

```
## (Intercept)
##       1.298
```

```
log(sum(build$sex == "M")/sum(build$sex == "F"))
```

```
## [1] 1.298
```

```
### Creating Predictor Variables
lpred <- predict(logist, newdata=pure, type= "response" )

### Setting a Cut-Off value at 50%
lpred50 <- ifelse(lpred >= .50 , "M", "F")

ltab <- table( lpred50 , pure$sex ) ; ltab
```

```
##
## lpred50     F     M
##       F  2957   877
##       M   860 13386
```

```
ltab1 <- round( sum( diag( ltab ) )/ sum( ltab ) , 4 ) ; ltab1
```

```
## [1] 0.9039
```

```
### Comparing All Five

data.frame("Unpruned",tab1,"Pruned",ptab1  ,
           "Forest",ftab1 , "Logistic", ltab1)
```

```
##   X.Unpruned.   tab1 X.Pruned.  ptab1 X.Forest.  ftab1 X.Logistic.  ltab1
## 1    Unpruned 0.9036    Pruned 0.9018    Forest 0.9044    Logistic 0.9039
```

```
#Unpruned
list(tab,ptab,ftab,ltab)
```

```
## [[1]]
##
## predtree     F     M
##        F  2913   839
##        M   904 13424
##
## [[2]]
##
```

```
## predprune     F     M
##         F  3198  1157
##         M   619 13106
##
## [[3]]
##
## pforest     F     M
##       F  3045   957
##       M   772 13306
##
## [[4]]
##
## lpred50     F     M
##       F  2957   877
##       M   860 13386
```

```
## Cleanup
rm(list=(ls()))
```