# HW6.R

davidfu

2021-11-14

Company : Stevens Project : HW6 Purpose : HW6 First Name : David Last Name : Fu CWID : 10471854 Date : November 15, 2021

Homework 6

```
rm(list=ls())
library('C50')
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
set.seed(513)
```

Data pre-processing

```
cancerData=read.csv("/Users/davidfu/Downloads/breast-cancer-wisconsin.csv", header=TRUE)
cancerData$Class <- as.factor(cancerData$Class)


index<-sort(sample(nrow(cancerData),round(.30*nrow(cancerData))))
training<-cancerData[-index,]
test<-cancerData[index,]
```

6.1

```
C50_class <- C5.0( Class~.,data=training )

summary(C50_class)
```

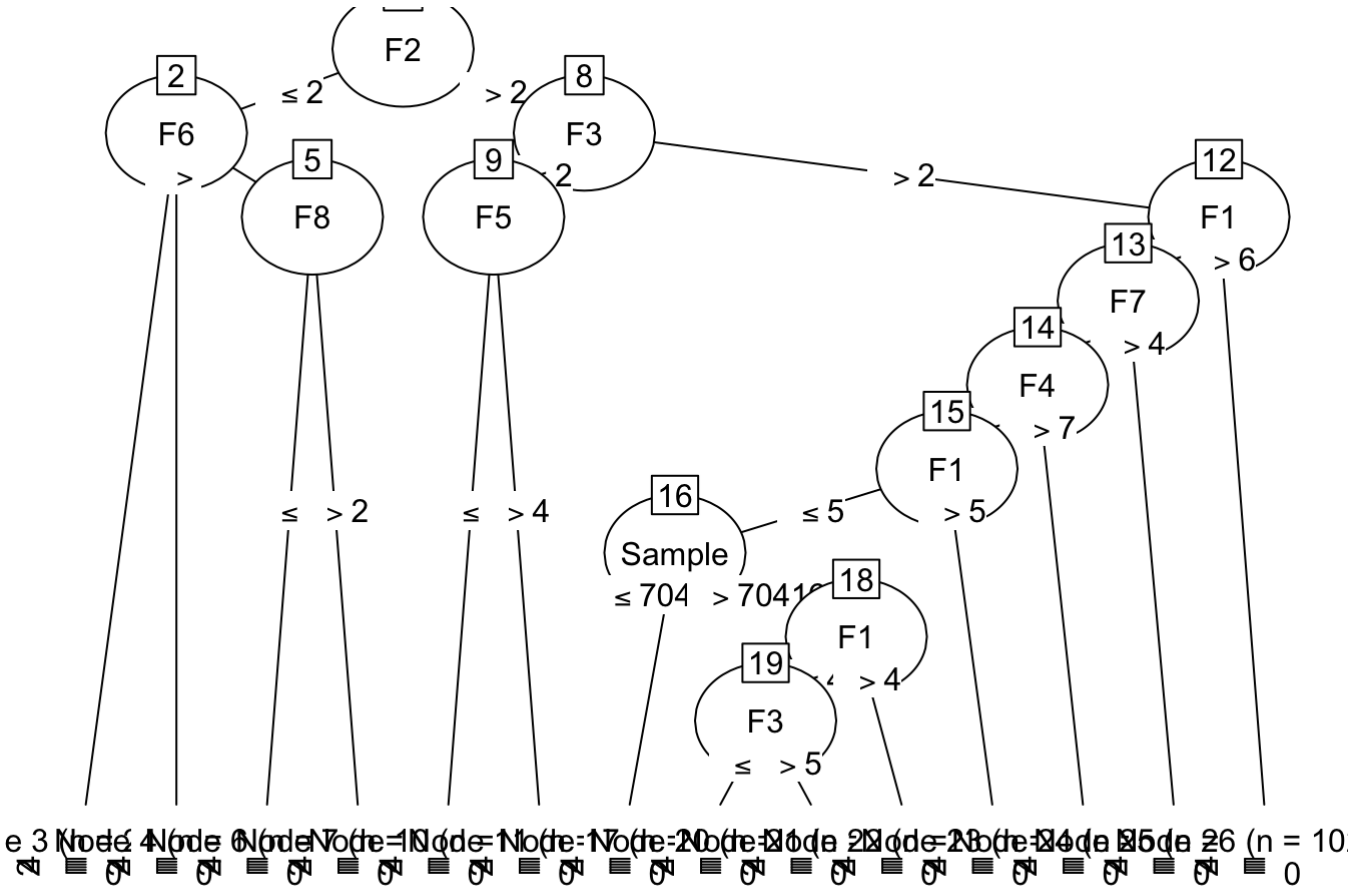
```
##
## Call:
## C5.0.formula(formula = Class ~ ., data = training)
##
##
## C5.0 [Release 2.07 GPL Edition]      Sun Nov 14 15:30:14 2021
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 489 cases (11 attributes) from undefined.data
##
## Decision tree:
##
## F2 <= 2:
## :...F6 in {1,7,?,9,2}: 2 (277.5)
## :   F6 in {10,8,6}: 4 (7.1/1.1)
## :   F6 in {4,3,5}:
## :   :...F8 <= 2: 2 (18.4/2)
## :       F8 > 2: 4 (2)
## F2 > 2:
## :...F3 <= 2:
##     :...F5 <= 4: 2 (15/1)
##     :   F5 > 4: 4 (2)
##     F3 > 2:
##     :...F1 > 6: 4 (102/2)
##         F1 <= 6:
##         :...F7 > 4: 4 (37/2)
##             F7 <= 4:
##             :...F4 > 7: 4 (6)
##                 F4 <= 7:
##                 :...F1 > 5: 2 (3)
##                     F1 <= 5:
##                     :...Sample <= 704168: 2 (4)
##                         Sample > 704168:
##                         :...F1 > 4: 4 (10/1)
##                             F1 <= 4:
##                             :...F3 <= 5: 2 (3)
##                                 F3 > 5: 4 (2)
##
##
## Evaluation on training data (489 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##      14    9( 1.8%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     318     6    (a): class 2
```

```
##       3   162    (b): class 4
##
##
##  Attribute usage:
##
##  100.00% F2
##   61.15% F6
##   37.63% F3
##   34.15% F1
##   13.29% F7
##    5.73% F4
##    5.32% F8
##    3.89% Sample
##    3.48% F5
##
##
## Time: 0.0 secs
```

```
plot(C50_class)
```

```
## Warning in partysplit(varid = as.integer(i), breaks = as.numeric(j[1]), : NAs
## introduced by coercion
```

```
## Warning in .bincode(as.numeric(x), breaks = unique(c(-Inf,
## breaks_split(split), : NAs introduced by coercion
```




```
C50_predict<-predict( C50_class ,test , type="class" )
table(actual=test[,4],C50=C50_predict)
```

```
##       C50
## actual   2   4
##      1  103   1
##      2   15   1
##      3    8  10
##      4    5   9
##      5    1   7
##      6    0   8
##      7    0  10
##      8    0  12
##      9    0   1
##      10   1  18
```

```
wrong<- (test[,4]!=C50_predict)
c50_rate<-sum(wrong)/length(test[,4])
c50_rate
```

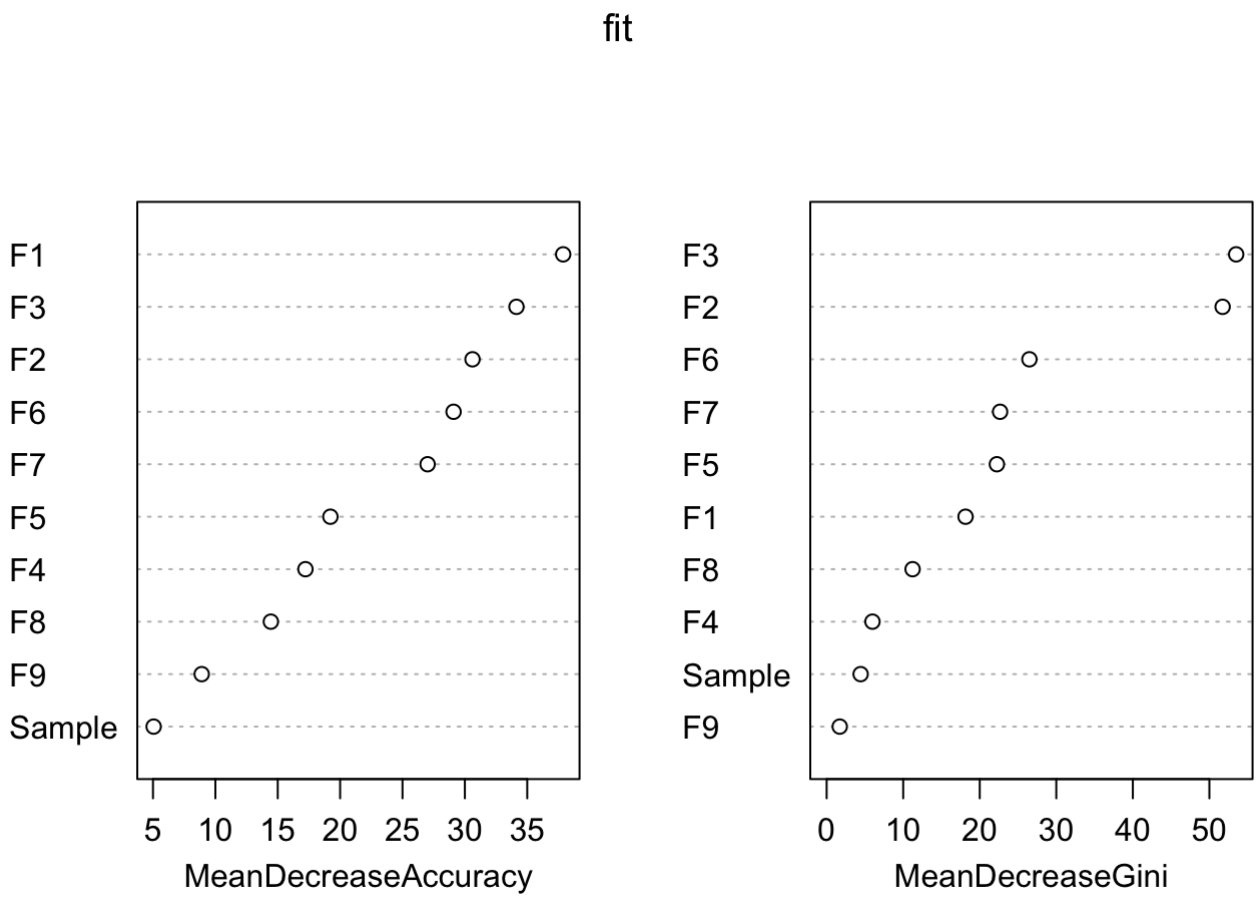
```
## [1] 0.8857143
```

6.2

```
fit <- randomForest( Class~., data=training, importance=TRUE, ntree=1000)
importance(fit)
```

```
##                2         4 MeanDecreaseAccuracy MeanDecreaseGini
## Sample  4.171793  3.187875             5.052912         4.440148
## F1     23.990906 36.313817            37.880432        18.144831
## F2     17.959975 24.680850            30.612491        51.730927
## F3     18.590482 31.038749            34.134890        53.500722
## F4     11.440263 13.894096            17.220041         5.985733
## F5     14.091233 13.596551            19.220985        22.237658
## F6     18.598561 22.772378            29.093227        26.488386
## F7     14.823079 24.409926            27.011632        22.657640
## F8     12.208263  8.583037            14.448329        11.227498
## F9      7.742260  4.612385             8.905665         1.712505
```

```
varImpPlot(fit)
```




```
Prediction <- predict(fit, test)
table(actual=test[,4],Prediction)
```

```
##       Prediction
## actual   2   4
##      1  104   0
##      2   14   2
##      3   10   8
##      4    2  12
##      5    0   8
##      6    1   7
##      7    0  10
##      8    0  12
##      9    0   1
##      10   0  19
```

```
wrong<- (test[,4]!=Prediction )
error_rate<-sum(wrong)/length(wrong)
error_rate
```

```
## [1] 0.8761905
```

Memory Clean up

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