HW_06

Preet Dabhi

3/29/2022

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

```
## Warning: package 'randomForest' was built under R version 4.1.3
```

```
## randomForest 4.7-1
```

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

```
library(class)
library(C50)
```

```
## Warning: package 'C50' was built under R version 4.1.3
```

```
df=read.csv("D:/SEM 3/CS 513/HW_02/breast-cancer-wisconsin.csv",header = TRUE, sep = ',')

#sumary of Data Frame
summary(df)
```

```
##      Sample               F1               F2               F3
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000   Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134   Mean   : 3.207
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000   Max.   :10.000
##        F4               F5              F6                  F7
##  Min.   : 1.000   Min.   : 1.000   Length:699         Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000   Class :character   1st Qu.: 2.000
##  Median : 1.000   Median : 2.000   Mode  :character   Median : 3.000
##  Mean   : 2.807   Mean   : 3.216                      Mean   : 3.438
##  3rd Qu.: 4.000   3rd Qu.: 4.000                      3rd Qu.: 5.000
##  Max.   :10.000   Max.   :10.000                      Max.   :10.000
##        F8               F9             Class
##  Min.   : 1.000   Min.   : 1.000   Min.   :2.00
##  1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.:2.00
##  Median : 1.000   Median : 1.000   Median :2.00
##  Mean   : 2.867   Mean   : 1.589   Mean   :2.69
##  3rd Qu.: 4.000   3rd Qu.: 1.000   3rd Qu.:4.00
##  Max.   :10.000   Max.   :10.000   Max.   :4.00
```

```
# F6 is a type of character, need to convert into the number
df$F6<-as.numeric(as.character((df$F6)))
```

```
## Warning: NAs introduced by coercion
```

```
summary((df))
```

```
##      Sample               F1               F2               F3
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000   Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134   Mean   : 3.207
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000   Max.   :10.000
##
##        F4               F5               F6               F7
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 2.000
##  Median : 1.000   Median : 2.000   Median : 1.000   Median : 3.000
##  Mean   : 2.807   Mean   : 3.216   Mean   : 3.545   Mean   : 3.438
##  3rd Qu.: 4.000   3rd Qu.: 4.000   3rd Qu.: 6.000   3rd Qu.: 5.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##                                    NA’s   :16
##        F8               F9             Class
##  Min.   : 1.000   Min.   : 1.000   Min.   :2.00
##  1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.:2.00
##  Median : 1.000   Median : 1.000   Median :2.00
##  Mean   : 2.867   Mean   : 1.589   Mean   :2.69
##  3rd Qu.: 4.000   3rd Qu.: 1.000   3rd Qu.:4.00
##  Max.   :10.000   Max.   :10.000   Max.   :4.00
##
```

```
# count and remove NA's from the dataframe
sum(is.na(df))
```

```
## [1] 16
```

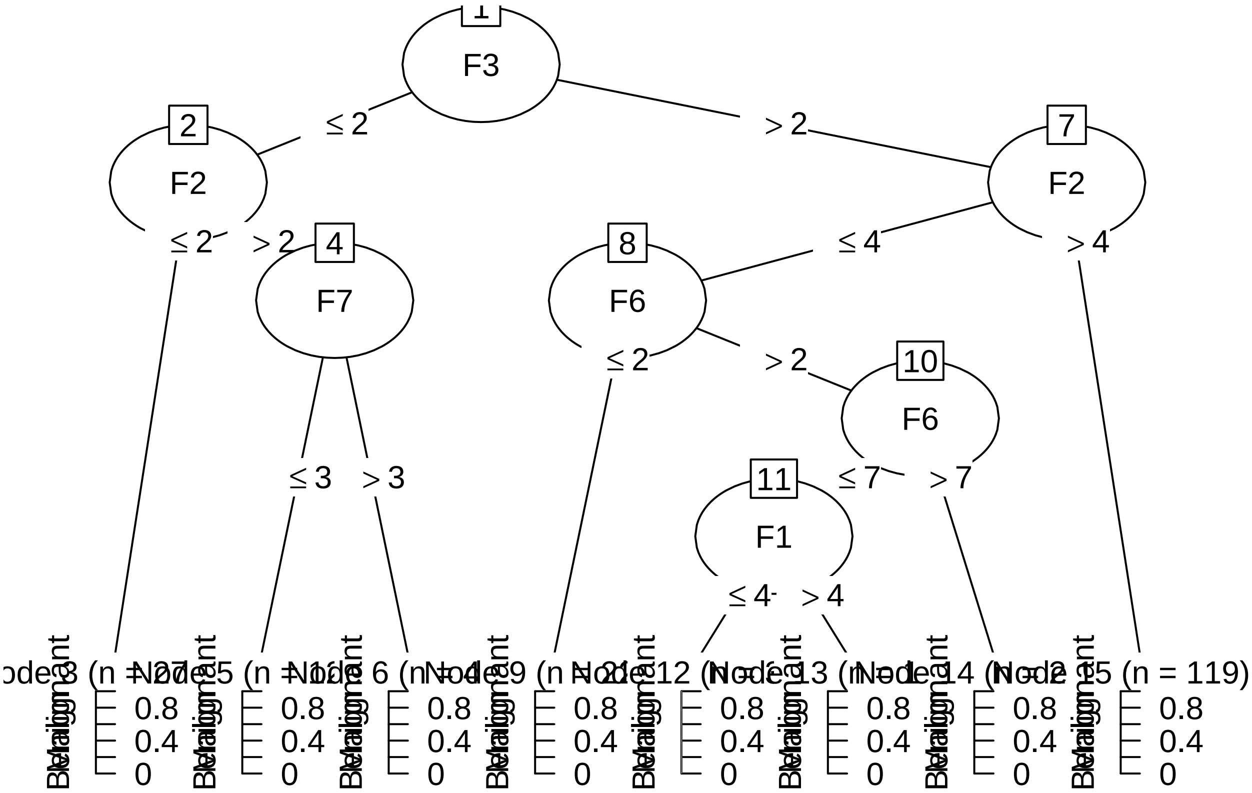
```
df<-na.omit(df)
sum(is.na(df))
```

```
## [1] 0
```

```
# convert Class into factor class
df$Class<-factor(df$Class, levels = c("2","4"), labels = c("Benign","Malignant"))
is.factor(df$Class)
```

```
## [1] TRUE
```

```r
# discard the sample/1st column from dataFrame

df<-df[2:11]
View(df)

# Split Train and Test data 70-30 ratio
split_size<-floor(0.70*nrow(df))


#set.seed(111)
random_sample<-sample(seq_len(nrow(df)), size = split_size)

train<-df[random_sample,]
test<-df[-random_sample,]

#Creating Accuracy function
accuracy<-function(x){
  sum(diag(x)/sum(rowSums(x)))*100
}

#Implementing C50
C50<-C5.0(Class~.,train)

plot(C50)
```



```
#Preddiction

pred_C50<-predict(C50,test,type = "class")
length(pred_C50)
```

```
## [1] 205
```

```
length(test)
```

```
## [1] 10
```

```
#confusionMatric
confMat_C50<-table(test$Class,pred_C50)
print(confMat_C50)
```

```
##            pred_C50
##             Benign Malignant
##   Benign       125         7
##   Malignant      5        68
```

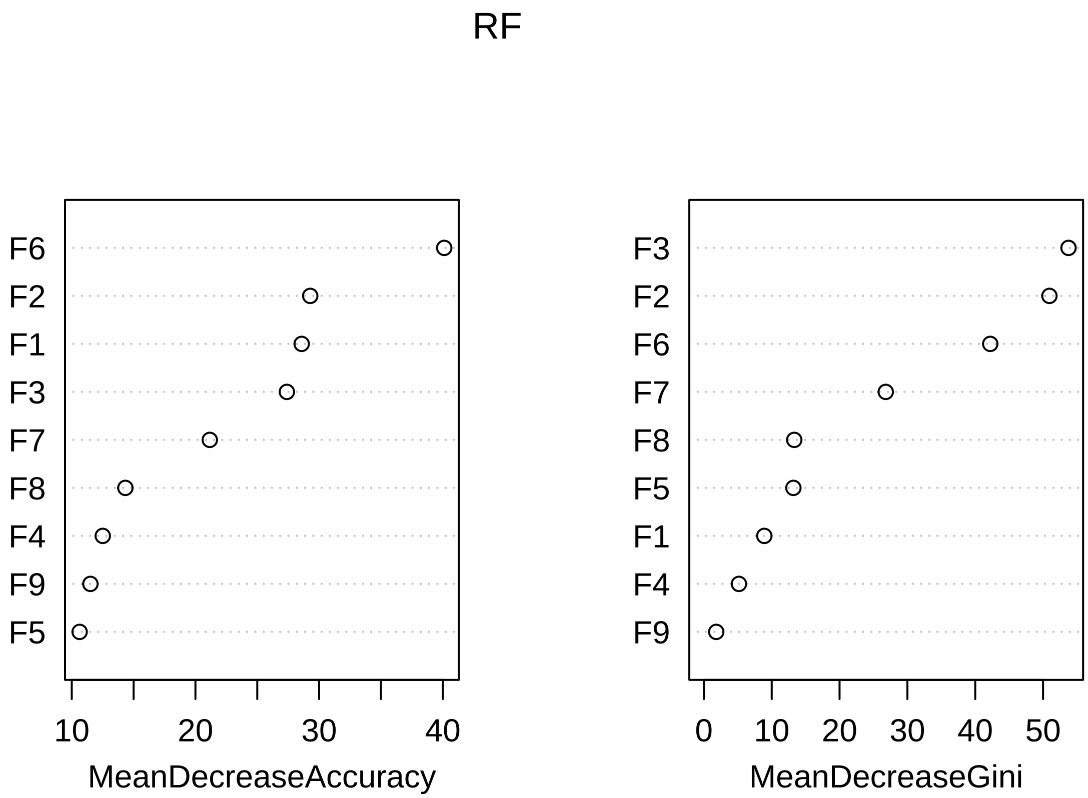
```
#Accuracy of C50
accuracy(confMat_C50)
```

```
## [1] 94.14634
```

```
####  Implementing Random Forest ####
RF<-randomForest(Class~.,train, importance=TRUE, ntree=1000)
importance(RF)
```

```
##       Benign Malignant MeanDecreaseAccuracy MeanDecreaseGini
## F1 24.547736 22.183537             28.58962         8.904472
## F2 23.861352 19.016152             29.28132        50.938590
## F3 14.131078 24.035614             27.39173        53.753539
## F4 10.094520  9.724453             12.50686         5.164124
## F5  9.007901  5.612302             10.63511        13.193618
## F6 31.682740 35.323108             40.11537        42.217665
## F7 11.758942 17.405684             21.15972        26.810524
## F8 12.176783 10.252445             14.33583        13.320896
## F9 10.508075  5.424261             11.50710         1.820905
```

```
varImpPlot(RF)
```



```
# Prediction for Random Forest

pred_RF<-predict(RF,test,type = "class")
length(pred_RF)
```

```
## [1] 205
```

```
length(test)
```

```
## [1] 10
```

```
#confusionMatric
confMat_RF<-table(test$Class,pred_RF)
print(confMat_RF)
```

```
##              pred_RF
##               Benign Malignant
##   Benign         126         6
##   Malignant        3        70
```

```
# Accuracy
accuracy(confMat_RF)
```

```
## [1] 95.60976
```