Assignment 6

Siddharth Harsukh Pansuria | CWID: 20005837

3/26/2022

Clearing environmental variables.

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

Loading the breast cancer dataset

```
data <- read.csv('breast-cancer-wisconsin.csv')
```

Displaying top 10 rows of the data loaded

```
head(data, n = 10)
```

```
##     Sample F1 F2 F3 F4 F5 F6 F7 F8 F9 Class
## 1  1000025  5  1  1  1  2  1  3  1  1     2
## 2  1002945  5  4  4  5  7 10  3  2  1     2
## 3  1015425  3  1  1  1  2  2  3  1  1     2
## 4  1016277  6  8  8  1  3  4  3  7  1     2
## 5  1017023  4  1  1  3  2  1  3  1  1     2
## 6  1017122  8 10 10  8  7 10  9  7  1     4
## 7  1018099  1  1  1  1  2 10  3  1  1     2
## 8  1018561  2  1  2  1  2  1  3  1  1     2
## 9  1033078  2  1  1  1  2  1  1  1  5     2
## 10 1033078  4  2  1  1  2  1  2  1  1     2
```

Removing first Column since it only contains ID

```
data <- data[-1]
```

Setting column 'Class' as a factor of benign and malignant

```
data$Class <- factor(data$Class, levels = c(2,4), labels = c("benign", "malignant"))
```

Displaying top 10 rows of the data loaded

```
head(data, n = 10)
```

```
##    F1 F2 F3 F4 F5 F6 F7 F8 F9     Class
## 1   5  1  1  1  2  1  3  1  1    benign
## 2   5  4  4  5  7 10  3  2  1    benign
## 3   3  1  1  1  2  2  3  1  1    benign
## 4   6  8  8  1  3  4  3  7  1    benign
## 5   4  1  1  3  2  1  3  1  1    benign
## 6   8 10 10  8  7 10  9  7  1 malignant
## 7   1  1  1  1  2 10  3  1  1    benign
## 8   2  1  2  1  2  1  3  1  1    benign
## 9   2  1  1  1  2  1  1  1  5    benign
## 10  4  2  1  1  2  1  2  1  1    benign
```

Splitting the dataset into training and test data

```
id <- sort(sample(nrow(data), as.integer(.70 * nrow(data))))
train_data <- data[id,]
test_data <- data[-id,]
```

Installing package and implementing C5.0

```
# install.packages("C50",repos="http://cran.us.r-project.org")
library('C50')
```

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

```
C50 <- C5.0(Class~., data = train_data)
```

Predicting using C5.0

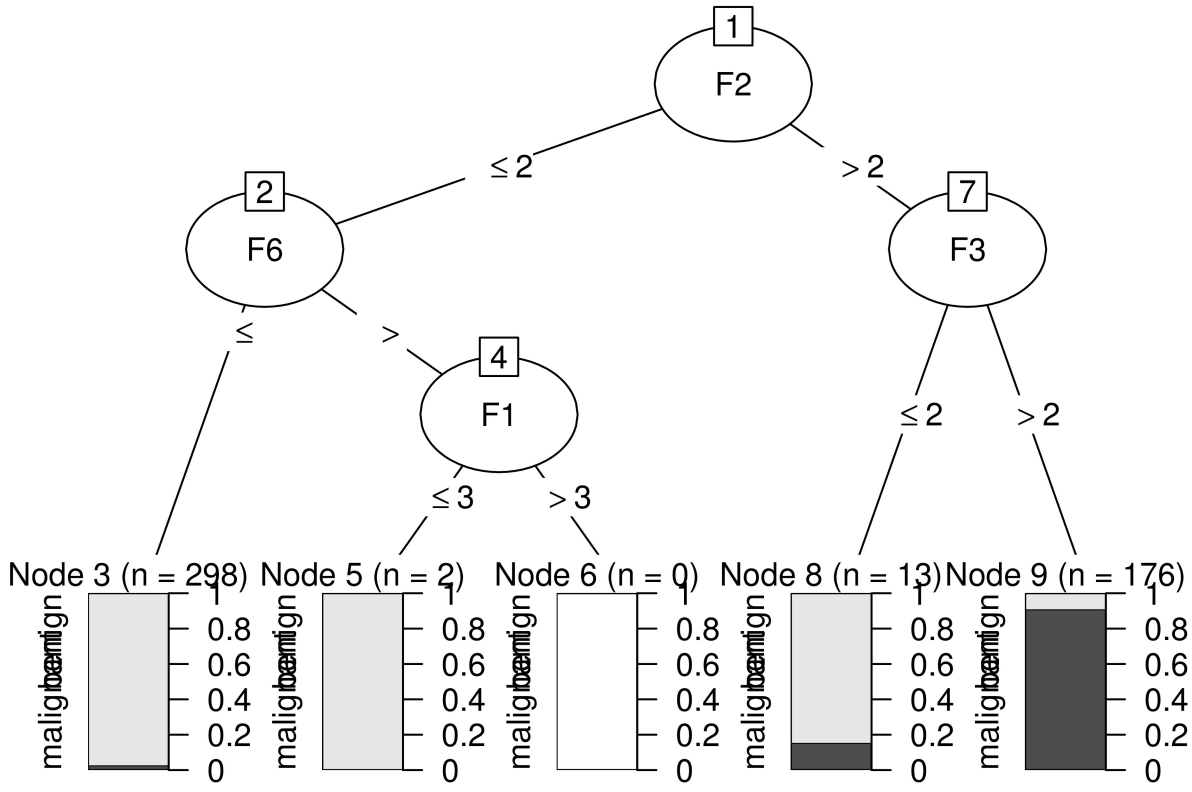
```
predict_C50 <- predict(C50, test_data, type = "class" )
```

Plotting the tree

```
plot(C50)
```

```
## 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
```



Frequency table for C5.0

```
table(C5.0 = predict_C50, Class = test_data$Class)
```

```
##            Class
## C5.0        benign malignant
##   benign       130         6
##   malignant      9        65
```

Calculating the error rate

Number of wrong predictions

```
wrong_preds_C50 <- sum(predict_C50 != test_data$Class)
```

Error Rate in C5.0 predictions

```
error_rate_C50 <- wrong_preds_C50/length(predict_C50)
print(paste("Error Rate:" , error_rate_C50))
```

```
## [1] "Error Rate: 0.0714285714285714"
```

Installing package and implementing Random Forest

```
# install.packages('randomForest', repos = "http://cran.us.r-project.org")
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.
```

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

Implementing Random Forest methodology

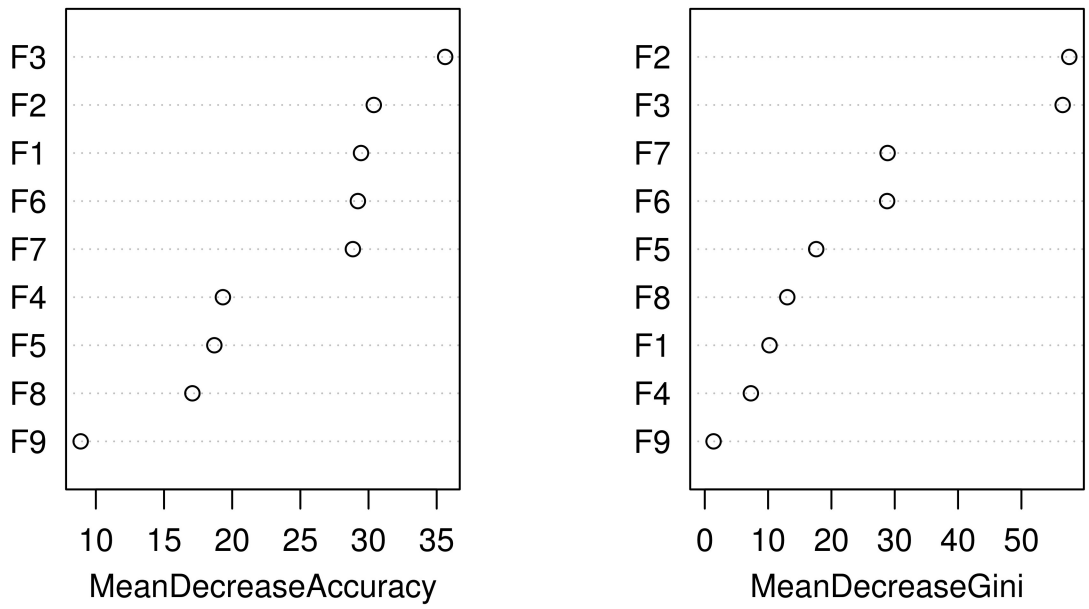
```
predict_randomF <- predict(randomF, test_data)
```

Identifying and plotting only the important features

```
importance(randomF)
```

```
##       benign malignant MeanDecreaseAccuracy MeanDecreaseGini
## F1 15.213383  32.69868            29.448054        10.221785
## F2 17.272388  26.42049            30.386660        57.563772
## F3 13.033279  33.86263            35.621568        56.517382
## F4 11.135793  16.34492            19.319152         7.281076
## F5 16.085272  10.75669            18.690244        17.600240
## F6 16.406302  24.45834            29.219098        28.798351
## F7 11.365189  26.04944            28.852033        28.869995
## F8 14.211406  10.70825            17.080466        13.027445
## F9  7.568183   4.64243             8.884604         1.387994
```

```
varImpPlot(randomF, main = "Chart of variable importance")
```



Frequency table for Random Forest

```
table(RandomForest = predict_randomF, Class = test_data$Class)
```

```
##             Class
## RandomForest benign malignant
##    benign       135         5
##    malignant      4        66
```

Calculating the error rate

Number of wrong predictions

```
wrong_preds_RF <- sum(predict_randomF != test_data$Class)
```

Error Rate in Random Forest predictions

```
error_rate_RF <- wrong_preds_RF/length(predict_randomF)
print(paste("Error Rate:" , error_rate_RF))
```

```
## [1] "Error Rate: 0.0428571428571429"
```