# Demonstrate step by step procedure of Random Forest using R

Harini G

```
#Name: Harini G


#install.packages("randomForest")
library("randomForest")

## Warning: package 'randomForest' was built under R version 4.0.4

## randomForest 4.6-14

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

#install.packages("reuire")
library(caTools)

## Warning: package 'caTools' was built under R version 4.0.4

#importing the dataset
data=read.csv("D:/Harini(christ unniversity)/2nd sem subjects/R/heart
disease.csv")
#dimension of the dataset
dim(data)#dataset contains 14 columns and 303 observations(rows)

## [1] 303  14

names(data)#printing the names of the columns

##  [1] "age"      "sex"      "cp"       "trestbps" "chol"     "fbs"
##  [7] "restecg"  "thalach"  "exang"    "oldpeak"  "slope"    "ca"
## [13] "thal"     "target"

head(data)

##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  63   1  3      145  233   1       0     150     0     2.3     0  0    1
## 2  37   1  2      130  250   0       1     187     0     3.5     0  0    2
## 3  41   0  1      130  204   0       0     172     0     1.4     2  0    2
## 4  56   1  1      120  236   0       1     178     0     0.8     2  0    2
## 5  57   0  0      120  354   0       1     163     1     0.6     2  0    2
## 6  57   1  0      140  192   0       1     148     0     0.4     1  0    1
##   target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

```
#data$target[data$target>1]
summary(data)
```

```
##       age             sex               cp           trestbps
##  Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0
##  1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0
##  Median :55.00   Median :1.0000   Median :1.000   Median :130.0
##  Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0
##       chol            fbs             restecg          thalach
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5
##  Median :240.0   Median :0.0000   Median :1.0000   Median :153.0
##  Mean   :246.3   Mean   :0.1485   Mean   :0.5281   Mean   :149.6
##  3rd Qu.:274.5   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
##      exang           oldpeak         slope             ca
##  Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.80   Median :1.000   Median :0.0000
##  Mean   :0.3267   Mean   :1.04   Mean   :1.399   Mean   :0.7294
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.20   Max.   :2.000   Max.   :4.0000
##       thal           target
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:2.000   1st Qu.:0.0000
##  Median :2.000   Median :1.0000
##  Mean   :2.314   Mean   :0.5446
##  3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :3.000   Max.   :1.0000
```

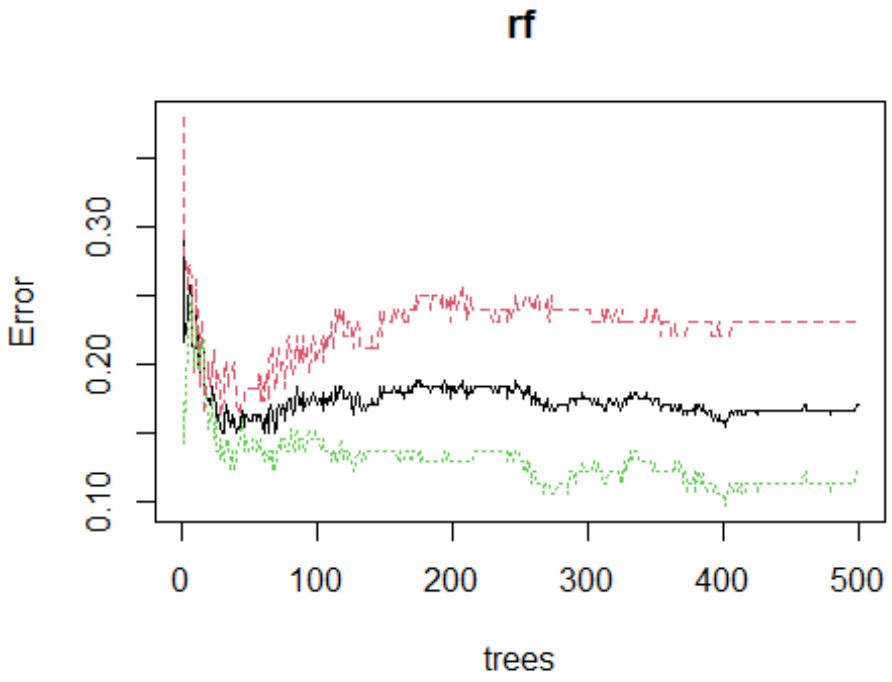
```
sapply(data, class)#displaying the datatype of each column
```

```
##       age       sex        cp  trestbps      chol       fbs   restecg
thalach
## "integer" "integer" "integer" "integer" "integer" "integer" "integer"
"integer"
##     exang   oldpeak     slope        ca      thal    target
## "integer" "numeric" "integer" "integer" "integer" "integer"
```

```
#chaning the datatype for few columns
data=transform(data,
sex=as.factor(sex),cp=as.factor(cp),fbs=as.factor(fbs),restecg=as.factor(rest
ecg),exang=as.factor(exang),slope=as.factor(slope),ca=as.factor(ca),thal=as.f
actor(thal),target=as.factor(target))
sapply(data, class)#displaying the datatype of each column
```

```
##       age       sex        cp  trestbps      chol       fbs   restecg
thalach
## "integer"  "factor"  "factor" "integer" "integer"  "factor"  "factor"
```

```
"integer"
##     exang   oldpeak     slope        ca      thal    target
##  "factor" "numeric"  "factor"  "factor"  "factor"  "factor"

summary(data)

##       age        sex     cp         trestbps          chol       fbs
##  Min.   :29.00   0: 96   0:143   Min.   : 94.0   Min.   :126.0   0:258
##  1st Qu.:47.50   1:207   1: 50   1st Qu.:120.0   1st Qu.:211.0   1: 45
##  Median :55.00           2: 87   Median :130.0   Median :240.0
##  Mean   :54.37           3: 23   Mean   :131.6   Mean   :246.3
##  3rd Qu.:61.00                   3rd Qu.:140.0   3rd Qu.:274.5
##  Max.   :77.00                   Max.   :200.0   Max.   :564.0
##  restecg    thalach      exang      oldpeak     slope   ca      thal
target
##  0:147   Min.   : 71.0   0:204   Min.   :0.00   0: 21   0:175   0:  2
0:138
##  1:152   1st Qu.:133.5   1: 99   1st Qu.:0.00   1:140   1: 65   1: 18
1:165
##  2:  4   Median :153.0           Median :0.80   2:142   2: 38   2:166
##          Mean   :149.6           Mean   :1.04           3: 20   3:117
##          3rd Qu.:166.0           3rd Qu.:1.60           4:  5
##          Max.   :202.0           Max.   :6.20

colSums(is.na(data))#checking if their are any null values

##      age      sex       cp trestbps     chol      fbs  restecg  thalach
##        0        0        0        0        0        0        0        0
##    exang  oldpeak    slope       ca     thal   target
##        0        0        0        0        0        0

#splitting the dataset into training and testing
sample=sample.split(data$target,SplitRatio=0.75)
train=subset(data,sample==TRUE)
test=subset(data,sample==FALSE)
dim(train)#dimesion of train data

## [1] 228  14

dim(test)#dimension of test data

## [1] 75 14

#building the random forest model
rf=randomForest(target~.,data=train)
#rf
plot(rf)
```




```
#Red line represents MCR of class not having heart diseases,
#green line represents MCR of class having heart diseases and
#black line represents overall MCR or OOB error.
#Overall error rate is what we are interested in which seems considerably
good.
#rf$confusion[, 'class.error']
varImpPlot(rf,sort = T,main = "Variable Importance",n.var = 5)
var.imp <- data.frame(importance(rf,type = 2))
#important variables for prediction are cp,thal,ca,thalach,oldpeak
var.imp$Variables <- row.names(var.imp)
var.imp[order(var.imp$MeanDecreaseGini, decreasing = T),]
```

```
##          MeanDecreaseGini Variables
## cp             15.8642569        cp
## thalach        14.5984343   thalach
## oldpeak        13.4412336   oldpeak
## ca             10.8031750        ca
## thal           10.7807712      thal
## age             8.9324925       age
## chol            8.3240870      chol
## trestbps        8.2845954  trestbps
## exang           7.1224983     exang
## slope           6.8924741     slope
## sex             3.2736212       sex
## restecg         2.3319608   restecg
## fbs             0.8498982       fbs
```

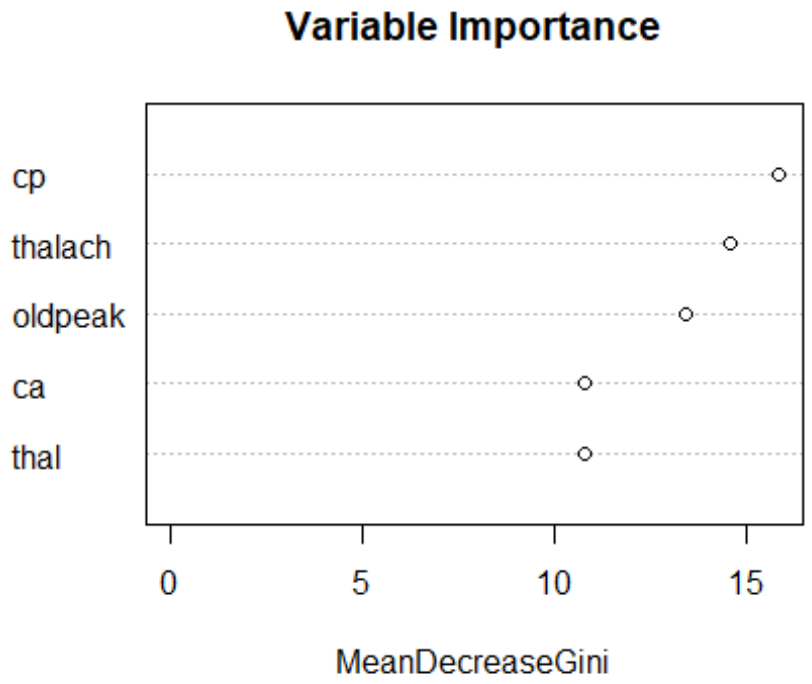
```
#higher descrease in Gini means that a particular predictor variable
#plays a greater role in partitioning the data into the defined classes.
train$predicted.response <- predict(rf, train)#training the data
library(e1071)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```




```
#printing the confusion matrix for taining data
confusionMatrix(data = train$predicted.response,reference = train$target)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   0   1
##          0 104   0
##          1   0 124
##
##                Accuracy : 1
```

```
##                  95% CI : (0.984, 1)
##     No Information Rate : 0.5439
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.4561
##          Detection Rate : 0.4561
##    Detection Prevalence : 0.4561
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

```
#the accuracy we have got for taining is 100%
test$predicted.response <- predict(rf, test)#testing
#printing the confusion matrix for test data
confusionMatrix(data = test$predicted.response,reference = test$target)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 29  9
##          1  5 32
##
##                Accuracy : 0.8133
##                  95% CI : (0.7067, 0.894)
##     No Information Rate : 0.5467
##     P-Value [Acc > NIR] : 1.183e-06
##
##                   Kappa : 0.6271
##
##  Mcnemar's Test P-Value : 0.4227
##
##             Sensitivity : 0.8529
##             Specificity : 0.7805
##          Pos Pred Value : 0.7632
##          Neg Pred Value : 0.8649
##              Prevalence : 0.4533
##          Detection Rate : 0.3867
##    Detection Prevalence : 0.5067
##       Balanced Accuracy : 0.8167
##
```

```
##         'Positive' Class : 0
##

#the accuracy we hae got for testing data is is more than 80%
```