Assignment 2

Predicting the type of a Breast Tumor (benign or malignant) Using Random-Forest Model

Mohamed Megahed

4/12/2020

The data is loaded using the mlbench library, data(BreastCancer) A data frame with 699 observations on 11 variables, one being a character variable, 9 being ordered or nominal, and 1 target class.

```
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ggplot2)
```

*** Understanding Data structure

```
summary(BreastCancer)

##       Id             Cl.thickness   Cell.size     Cell.shape  Marg.adhesio
n
##  Length:699         1      :145   1      :384   1      :353   1      :407
##  Class :character   5      :130   10     : 67   2      : 59   2      : 58
##  Mode  :character   3      :108   3      : 52   10     : 58   3      : 58
##                     4      : 80   2      : 45   3      : 56   10     : 55
##                     10     : 69   4      : 40   4      : 44   4      : 33
##                     2      : 50   5      : 30   5      : 34   8      : 25
##                     (Other):117   (Other): 81   (Other): 95   (Other): 63
##   Epith.c.size  Bare.nuclei   Bl.cromatin  Normal.nucleoli    Mitoses
```

```
##  2       :386   1       :402   2       :166   1       :443    1       :579
##  3       : 72   10      :132   3       :165   10      : 61    2       : 35
##  4       : 48   2       : 30   1       :152   3       : 44    3       : 33
##  1       : 47   5       : 30   7       : 73   2       : 36    10      : 14
##  6       : 41   3       : 28   4       : 40   8       : 24    4       : 12
##  5       : 39   (Other): 61   5       : 34   6       : 22    7       :  9
##  (Other): 66   NA's    : 16   (Other): 69   (Other): 69    (Other): 17
##        Class
##  benign   :458
##  malignant:241
##
##
##
##
##
```

```r
str(BreastCancer)
```

```
## 'data.frame':    699 obs. of  11 variables:
##  $ Id             : chr  "1000025" "1002945" "1015425" "1016277" ...
##  $ Cl.thickness   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 5 5 3 6 4
8 1 2 2 4 ...
##  $ Cell.size      : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1
10 1 1 1 2 ...
##  $ Cell.shape     : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1
10 1 2 1 1 ...
##  $ Marg.adhesion  : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 5 1 1 3
8 1 1 1 1 ...
##  $ Epith.c.size   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 2 7 2 3 2
7 2 2 2 2 ...
##  $ Bare.nuclei    : Factor w/ 10 levels "1","2","3","4",..: 1 10 2 4 1 10
10 1 1 1 ...
##  $ Bl.cromatin    : Factor w/ 10 levels "1","2","3","4",..: 3 3 3 3 3 9 3
3 1 2 ...
##  $ Normal.nucleoli: Factor w/ 10 levels "1","2","3","4",..: 1 2 1 7 1 7 1
1 1 1 ...
##  $ Mitoses        : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1
5 1 ...
##  $ Class          : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1
1 1 1 ...
```

```r
levels(BreastCancer$Class)
```

```
## [1] "benign"    "malignant"
```

Let’s calculate the number and percent of missing data and plot them ** checking if there any missing data using “Amelia package”

```r
# Check if there are any missing values:
anyNA(BreastCancer)
```

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

```r
sum(is.na(BreastCancer))
```

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

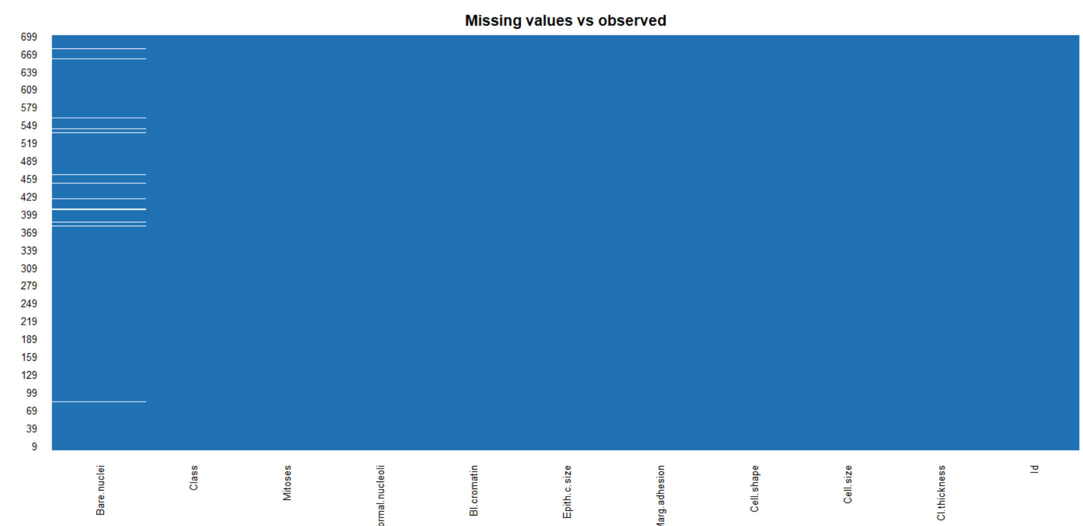
Ploting the missing and observed data values

```r
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.6, built: 2019-11-24)
## ## Copyright (C) 2005-2020 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```r
missmap(BreastCancer, main = "Missing values vs observed",legend = FALSE)
```




```r
mean(is.na(BreastCancer))
```

```
## [1] 0.002080895
```

** we have 16 missing values in our dataset.

Cleaning missing data

```r
Breast <- na.omit(BreastCancer)[,c(2:11)]
```

```r
library(randomForest)
```

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

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

```
##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
set.seed(123)
intrain <- createDataPartition(y = Breast$Class, p= 0.7, list = FALSE)
training <- Breast[intrain,]
testing <- Breast[-intrain,]
```

```
set.seed(123)
rf.model<-train(Class~.,data=training,method='rf')
rf.model
```

```
## Random Forest
##
## 479 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 479, 479, 479, 479, 479, 479, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9554657  0.9033081
##   41    0.9499567  0.8908242
##   80    0.9455779  0.8813470
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

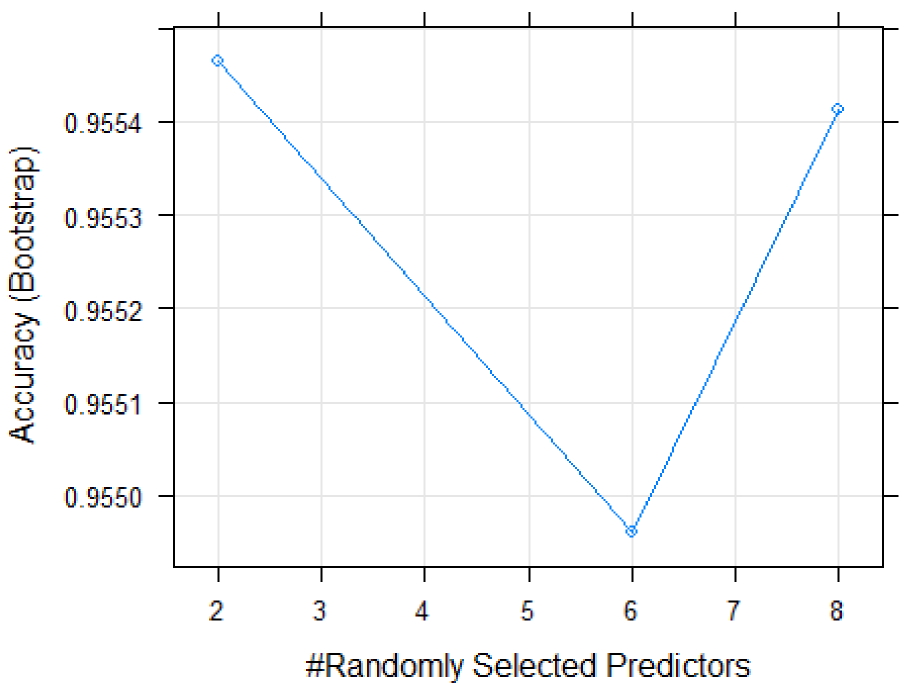
Grid search with Bootstrapped Resampling

```
set.seed(123)
Grid_Serach <- expand.grid(.mtry=c(2,6,8))
#Building a random forest model
RF_Grid_Boot<-train(Class~.,
                 data=training,
                 method='rf',
                 tuneGrid=Grid_Serach)
print(RF_Grid_Boot)
```

```
## Random Forest
##
```

```
## 479 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 479, 479, 479, 479, 479, 479, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.9554657  0.9033081
##   6     0.9549596  0.9021462
##   8     0.9554124  0.9031210
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```
plot(RF_Grid_Boot)
```




```
preds_rf_boot <- predict(RF_Grid_Boot, testing[1:9])

confusionMatrix(table(preds_rf_boot, testing$Class))
```

```
## Confusion Matrix and Statistics
##
##
## preds_rf_boot benign malignant
```

```
##     benign       129         3
##     malignant      4        68
##
##                Accuracy : 0.9657
##                  95% CI : (0.9306, 0.9861)
##     No Information Rate : 0.652
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9246
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9699
##             Specificity : 0.9577
##          Pos Pred Value : 0.9773
##          Neg Pred Value : 0.9444
##              Prevalence : 0.6520
##          Detection Rate : 0.6324
##    Detection Prevalence : 0.6471
##       Balanced Accuracy : 0.9638
##
##        'Positive' Class : benign
##
```

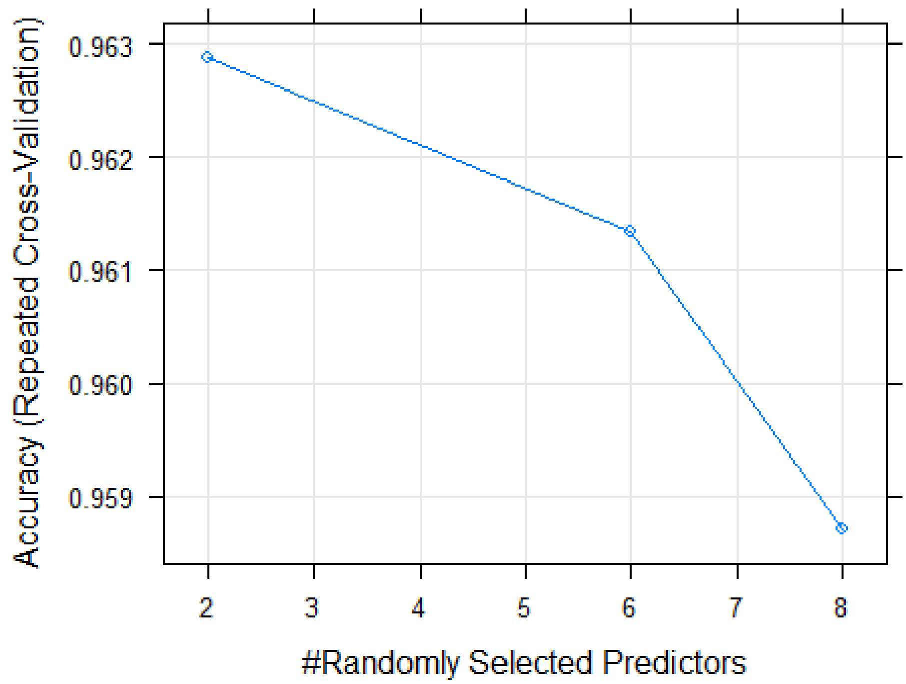
Grid Search with Cross-Validation (10 fold, repeated 4 times)

```
set.seed(123)
control <- trainControl(method="repeatedcv", number=10, repeats=4, search="gr
id")
Grid_Serach <- expand.grid(.mtry=c(2,6,8))
# Random forest Model Building
RF_Grid_CV<-train(Class~.,
                 data=training,
                 method='rf',
                 tuneGrid=Grid_Serach,
                 trControl=control
               )
print(RF_Grid_CV)
```

```
## Random Forest
##
## 479 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 4 times)
## Summary of sample sizes: 430, 431, 431, 431, 432, 431, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
```

```
##   2      0.9628856  0.9192825
##   6      0.9613342  0.9163904
##   8      0.9586968  0.9103331
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```
plot(RF_Grid_CV)
```




```
#Prediction using test data
preds_rf_cv <- predict(RF_Grid_CV, testing[1:9])
confusionMatrix(table(preds_rf_cv, testing$Class))
```

```
## Confusion Matrix and Statistics
##
##
## preds_rf_cv benign malignant
##   benign       129         3
##   malignant      4        68
##
##                 Accuracy : 0.9657
##                   95% CI : (0.9306, 0.9861)
##      No Information Rate : 0.652
##      P-Value [Acc > NIR] : <2e-16
##
##                    Kappa : 0.9246
##
```

```
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9699
##             Specificity : 0.9577
##          Pos Pred Value : 0.9773
##          Neg Pred Value : 0.9444
##              Prevalence : 0.6520
##          Detection Rate : 0.6324
##    Detection Prevalence : 0.6471
##       Balanced Accuracy : 0.9638
##
##        'Positive' Class : benign
##
```

From the above analysis we can notice that The 10-fold cross validation has a better accuracy than bootstrapped resampling.