Breast Tumour Classification using Random Forest

Sumanth

4/13/2020

Cancer Classification (benign or malignant)

The Data is loaded from the mlbench library. This data frame has 699 observations and 11 variables, one being a character variable, 9 being ordered or nominal, and 1 target class.

Import Libraries

```r
library(ggplot2)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.6.2
```

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

```r
library(naniar)  #For visual representation of missing values
```

```
## Warning: package 'naniar' was built under R version 3.6.3
```

```r
library(mlbench) #For Breast Cancer Dataset
print('Libraries Imported!')
```

```
## [1] "Libraries Imported!"
```

Exploratory Data Analysis

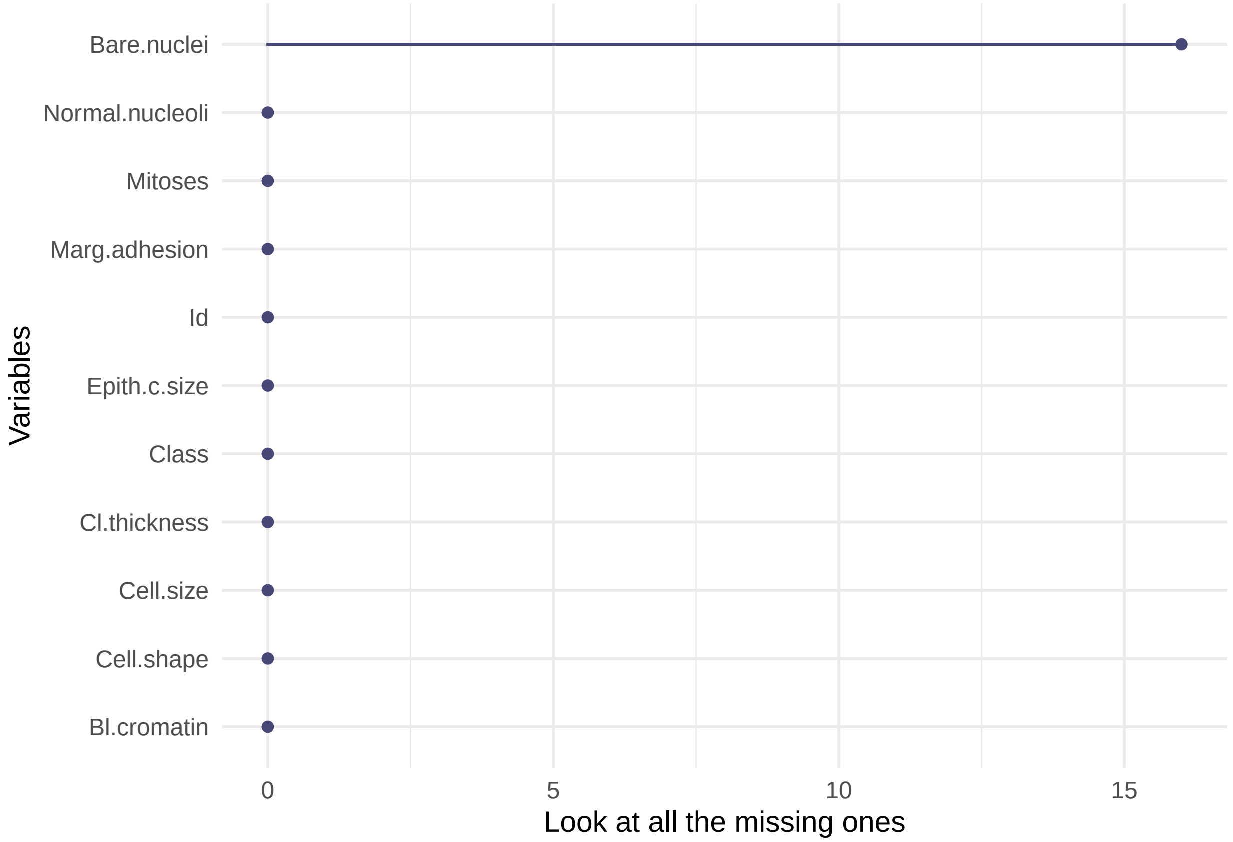
```r
# Importing and observing the structure of the data
data("BreastCancer")
summary(BreastCancer)
```

```
##       Id             Cl.thickness   Cell.size     Cell.shape  Marg.adhesion
##  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
##
##
##
##
##
```

```
#compare and visualize missing values
table(complete.cases(BreastCancer))
```

```
##
## FALSE  TRUE
##    16   683
```

```
gg_miss_var(BreastCancer) + labs(y = "Look at all the missing ones")
```



- As we can observe, There are 16 missing values in the data. I have considered removing rows with missing values instead of imputing them.

```
# Data without missing values and ID column
bc<-na.omit(BreastCancer)[,c(2:11)]
table(complete.cases(bc))
```

```
##
## TRUE
##  683
```

Now, it is confirmed that there are no missing values. we will proceed to split the data

```
#spliting the data
set.seed(20)
intrain <- createDataPartition(y = bc$Class, p= 0.7, list = FALSE)
training <- bc[intrain,]
testing <- bc[-intrain,]
```

Model withouth grid search

```r
set.seed(20)
rf.model<-train(Class~.,data=training,method='rf')
print(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    0.9605913  0.9145190
##   41    0.9515329  0.8944200
##   80    0.9423910  0.8739099
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

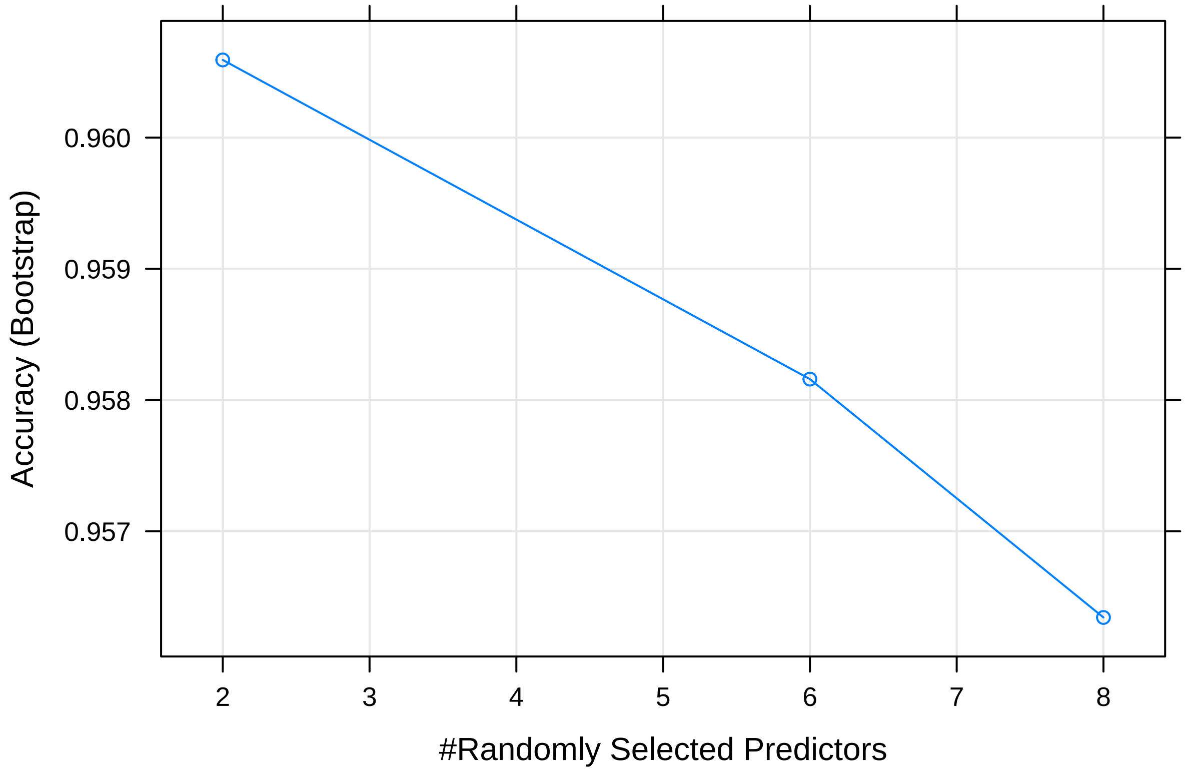
Grid search with Bootstrapped Resampling

```r
set.seed(20)
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     0.9605913  0.9145190
##   6     0.9581598  0.9094454
##   8     0.9563436  0.9052808
##
```

```
## 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         2
##     malignant      4        69
##
##                Accuracy : 0.9706
##                  95% CI : (0.9371, 0.9891)
##     No Information Rate : 0.652
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9356
##
##  Mcnemar's Test P-Value : 0.6831
##
```

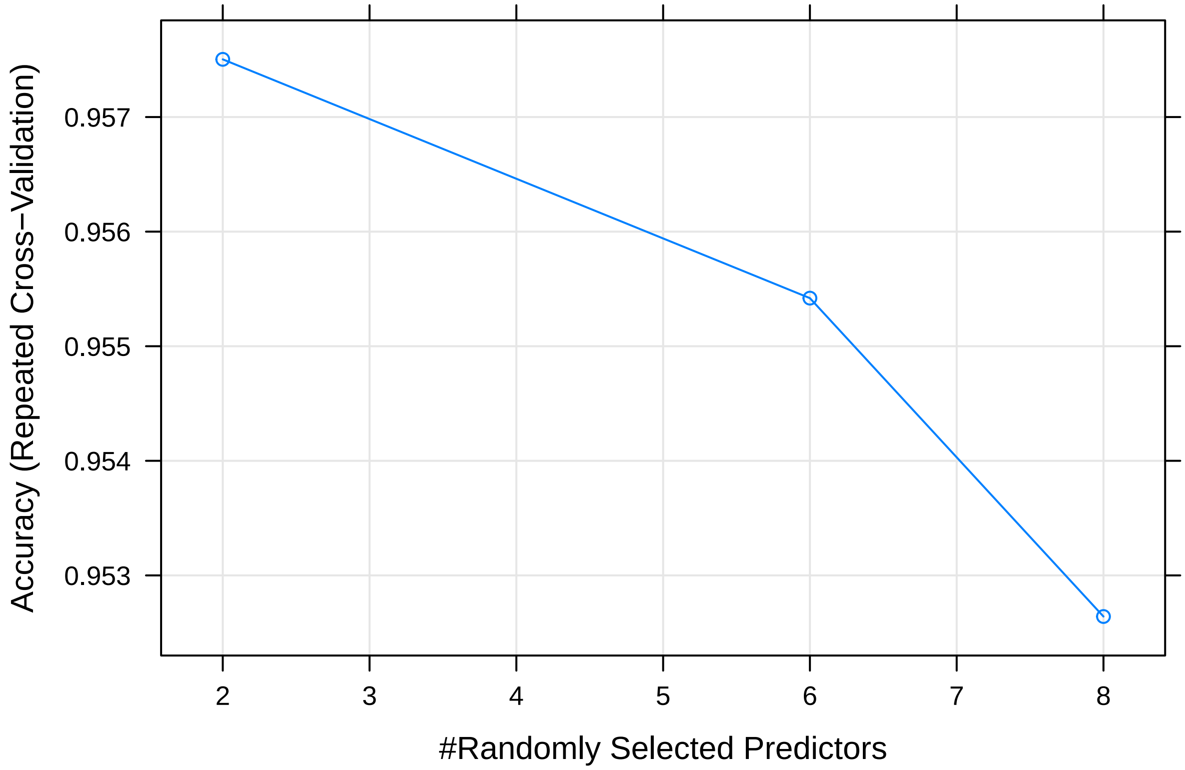
```
##             Sensitivity : 0.9699
##             Specificity : 0.9718
##          Pos Pred Value : 0.9847
##          Neg Pred Value : 0.9452
##              Prevalence : 0.6520
##          Detection Rate : 0.6324
##    Detection Prevalence : 0.6422
##       Balanced Accuracy : 0.9709
##
##        'Positive' Class : benign
##
```

Grid Search with Cross-Validation (10 fold, repeated 3 times)

```
set.seed(20)
control <- trainControl(method="repeatedcv", number=10, repeats=3, search="grid")
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 3 times)
## Summary of sample sizes: 431, 431, 431, 431, 431, 432, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.9575035  0.9069478
##   6     0.9554196  0.9025849
##   8     0.9526412  0.8962723
##
## 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)
```



```r
#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       130         2
##   malignant      3        69
##
##                Accuracy : 0.9755
##                  95% CI : (0.9437, 0.992)
##     No Information Rate : 0.652
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9462
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9774
##             Specificity : 0.9718
##          Pos Pred Value : 0.9848
##          Neg Pred Value : 0.9583
##              Prevalence : 0.6520
##          Detection Rate : 0.6373
##    Detection Prevalence : 0.6471
##       Balanced Accuracy : 0.9746
##
##        'Positive' Class : benign
##
```

Observations & Conclusions :

- Removing the missing data instead of Imputing it yielded a better accuracy
- The Data Partitioning is Highly crutial part of model building.
  - For 70% data split, it's been observed that there is **low Bias and Low variance** when compared with data split = 80%
- The 10-fold cross validation yields a better accuracy than bootstrapped resampling