DECISION TREE AND RANDOM FOREST: ECOSTAT TRAINING

Lumumba Wandera Victor

October 13, 2023

Contents

Load the Following Libraries

```
library(sjmisc)
library(sjPlot)
library(nnet)
library(wakefield)
library(dplyr)
library(nnet)
library(caTools)
library(ROCR)
library(stargazer)
library(dplyr)
library(nnet)
library(caTools)
library(ROCR)
library(stargazer)
library(ISLR)
library(ISLR2)
library(MASS)
library(caret)
library(splines)
library(splines2)
library(pROC)
library(ISLR)
library(ISLR2)
library(MASS)
library(caret)
library(splines)
library(splines2)
library(pROC)
library(randomForest)
library(rpart)
library(rpart.plot)
library(rattle)
library(ISLR2)
library(MASS)
library(caret)
library(splines)
library(pROC)
library(rattle)
library(rpart)
library(party)
library(partykit)
library(ggplot2)
```

APPLICATION OF RANDOM FOREST IN CLASSIFICATION MODEL

Load the data set

```
mydata4 <- read.csv("Cardiotocographic.csv", header = TRUE)
str(mydata4)
```

```
'data.frame':   2126 obs. of  22 variables:
 $ LB      : int  120 132 133 134 132 134 134 122 122 122 ...
 $ AC      : num  0 0.00638 0.00332 0.00256 0.00651 ...
 $ FM      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ UC      : num  0 0.00638 0.00831 0.00768 0.00814 ...
 $ DL      : num  0 0.00319 0.00332 0.00256 0 ...
 $ DS      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ DP      : num  0 0 0 0 0 ...
 $ ASTV    : int  73 17 16 16 16 26 29 83 84 86 ...
 $ MSTV    : num  0.5 2.1 2.1 2.4 2.4 5.9 6.3 0.5 0.5 0.3 ...
 $ ALTV    : int  43 0 0 0 0 0 0 6 5 6 ...
 $ MLTV    : num  2.4 10.4 13.4 23 19.9 0 0 15.6 13.6 10.6 ...
 $ Width   : int  64 130 130 117 117 150 150 68 68 68 ...
 $ Min     : int  62 68 68 53 53 50 50 62 62 62 ...
 $ Max     : int  126 198 198 170 170 200 200 130 130 130 ...
 $ Nmax    : int  2 6 5 11 9 5 6 0 0 1 ...
 $ Nzeros  : int  0 1 1 0 0 3 3 0 0 0 ...
 $ Mode    : int  120 141 141 137 137 76 71 122 122 122 ...
 $ Mean    : int  137 136 135 134 136 107 107 122 122 122 ...
 $ Median  : int  121 140 138 137 138 107 106 123 123 123 ...
 $ Variance: int  73 12 13 13 11 170 215 3 3 1 ...
 $ Tendency: int  1 0 0 1 1 0 0 1 1 1 ...
 $ NSP     : int  2 1 1 1 1 3 3 3 3 3 ...
```

```
attach(mydata4)
```

The data loaded has 2126 observations with 22 variables. This data is called is CTG. The data has the variable FHR, fetal heart rate and uterine contraction (UC) feature on cardiotocograms. 2126 fetal cardiotocograms (CTGs) automatically processed and diagnostic feature measured. CTG classified by three experrs obstetrician and consensus classification label as Normal, Suspect or Pathologic. The response variable is NSP (Normal, Suspect, Pathologic) and all the 21 variables are predictor variables. Remember NSP is an integer, we will have to convert it factor

```
mydata4$NSP <- factor(mydata4$NSP, levels = c(1,2,3),
                      labels = c("Normal", "Suspect", "Pathologic"))
str(mydata4)
```

```
'data.frame':   2126 obs. of  22 variables:
 $ LB      : int  120 132 133 134 132 134 134 122 122 122 ...
 $ AC      : num  0 0.00638 0.00332 0.00256 0.00651 ...
 $ FM      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ UC      : num  0 0.00638 0.00831 0.00768 0.00814 ...
 $ DL      : num  0 0.00319 0.00332 0.00256 0 ...
 $ DS      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ DP      : num  0 0 0 0 0 ...
 $ ASTV    : int  73 17 16 16 16 26 29 83 84 86 ...
 $ MSTV    : num  0.5 2.1 2.1 2.4 2.4 5.9 6.3 0.5 0.5 0.3 ...
 $ ALTV    : int  43 0 0 0 0 0 0 6 5 6 ...
 $ MLTV    : num  2.4 10.4 13.4 23 19.9 0 0 15.6 13.6 10.6 ...
 $ Width   : int  64 130 130 117 117 150 150 68 68 68 ...
 $ Min     : int  62 68 68 53 53 50 50 62 62 62 ...
 $ Max     : int  126 198 198 170 170 200 200 130 130 130 ...
 $ Nmax    : int  2 6 5 11 9 5 6 0 0 1 ...
 $ Nzeros  : int  0 1 1 0 0 3 3 0 0 0 ...
 $ Mode    : int  120 141 141 137 137 76 71 122 122 122 ...
 $ Mean    : int  137 136 135 134 136 107 107 122 122 122 ...
 $ Median  : int  121 140 138 137 138 107 106 123 123 123 ...
 $ Variance: int  73 12 13 13 11 170 215 3 3 1 ...
 $ Tendency: int  1 0 0 1 1 0 0 1 1 1 ...
 $ NSP     : Factor w/ 3 levels "Normal","Suspect",..: 2 1 1 1 1 3 3 3 3 3 ...
```

LB - FHR baseline (beats per minute) AC - # of accelerations per second FM - # of fetal movements per second UC - # of uterine contractions per second DL - # of light decelerations per second DS - # of severe decelerations per second DP - # of prolongued decelerations per second ASTV - percentage of time with abnormal short term variability MSTV - mean value of short term variability ALTV - percentage of time with abnormal long term variability MLTV - mean value of long term variability Width - width of FHR histogram Min - minimum of FHR histogram Max - Maximum of FHR histogram Nmax - # of histogram peaks Nzeros - # of histogram zeros Mode - histogram mode Mean - histogram mean Median - histogram median Variance - histogram variance Tendency - histogram tendency NSP - fetal state class code (N=normal; S=suspect; P=pathologic)

Now let us look at how many observations are present for each factor (Normal, Suspect, Pathologic)

```r
frq(mydata4, NSP)
```

```
NSP <categorical>
# total N=2126 valid N=2126 mean=1.30 sd=0.61

Value      |    N | Raw % | Valid % | Cum. %
--------------------------------------------
Normal     | 1655 | 77.85 |   77.85 |  77.85
Suspect    |  295 | 13.88 |   13.88 |  91.72
Pathologic |  176 |  8.28 |    8.28 | 100.00
<NA>       |    0 |  0.00 |    <NA> |   <NA>
```

From the frequency table above, majority of the respondents were found in normal category (1655), followed by those in suspect category (195) and finally those in pathologic category (176)

Data Partitioning

We shall start with random seed so that we can make this analysis reproducible

```r
set.seed(123)
ind <- sample(2, nrow(mydata4), replace = TRUE, prob = c(0.7, 0.3))
train_data <- mydata4[ind ==1,] ## 1495 observations
test_data <- mydata4[ind ==2,] ## 631 observations
```

Random Forest Model

- Random Forest developed by aggregating trees
- Can be used for classification or regression
- Avoids overfitting
- Can deal with large number of features
- Helps with feature selection based on importance
- Use-friendly: only 2 free parameters Trees - ntree, default 500 Variables randomly sampled as candidates at each split-mtry, default is sq.root(p) for classification & p/3 for regression

Random forest methodology is developed by aggregating several decision trees. Instead of using one decision tree, random forest uses several or hundreds of decision trees and aggregate the results from all the trees to come up with classification model. Remember, random forest can be used classification as well as regression. If the response variable is categorical variable, the algorithm will develop a classification model, and if the response variable is continuous, the algorithm will develop a regression model.

Steps

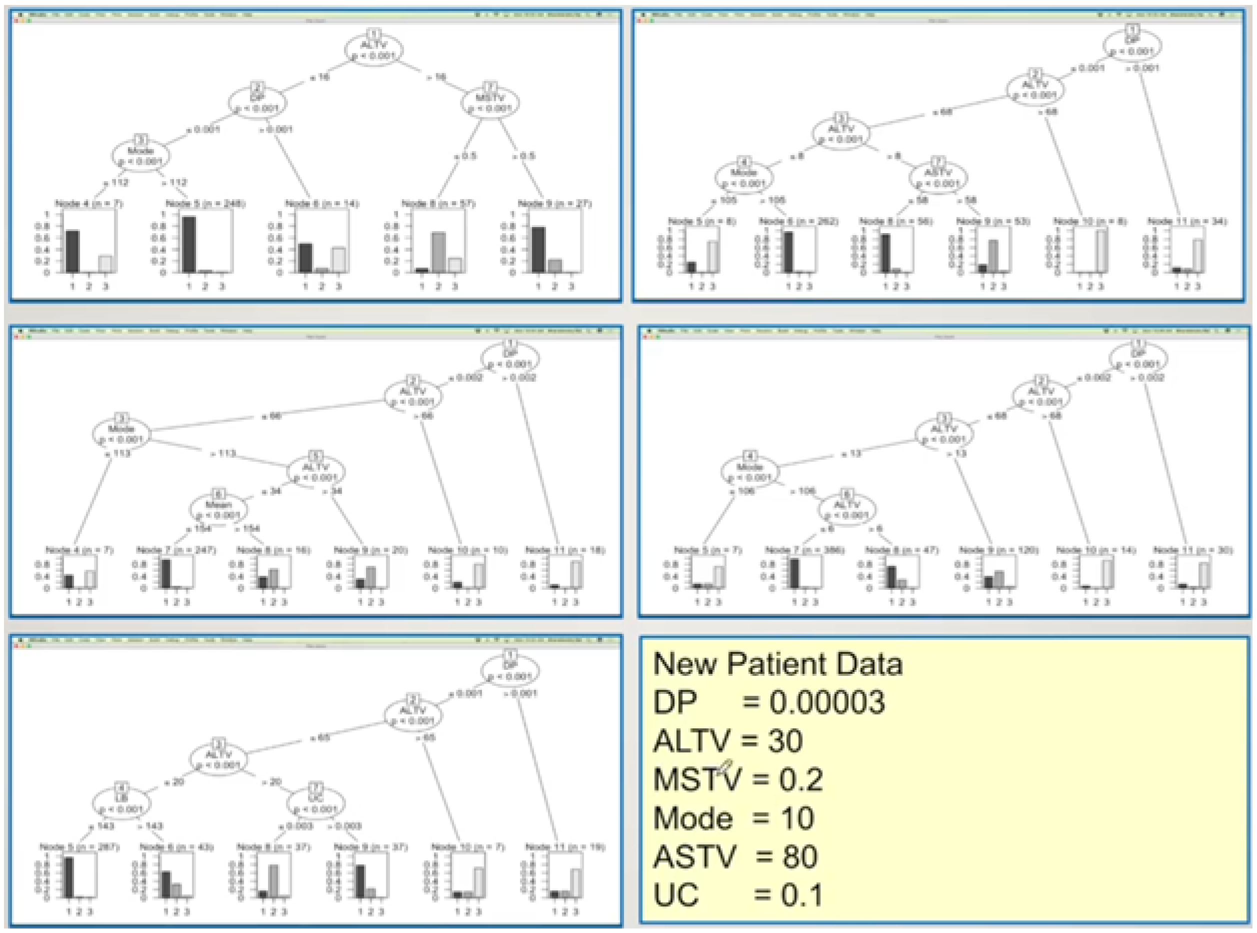
1. Draw ntree bootstrap samples

2. For each bootstrap sample, grow un-prunned tree by choosing best based on a random sample of mtry predictors at each node

3. Predict new data using the majority votes for classificatio and average for regression based on ntree trees

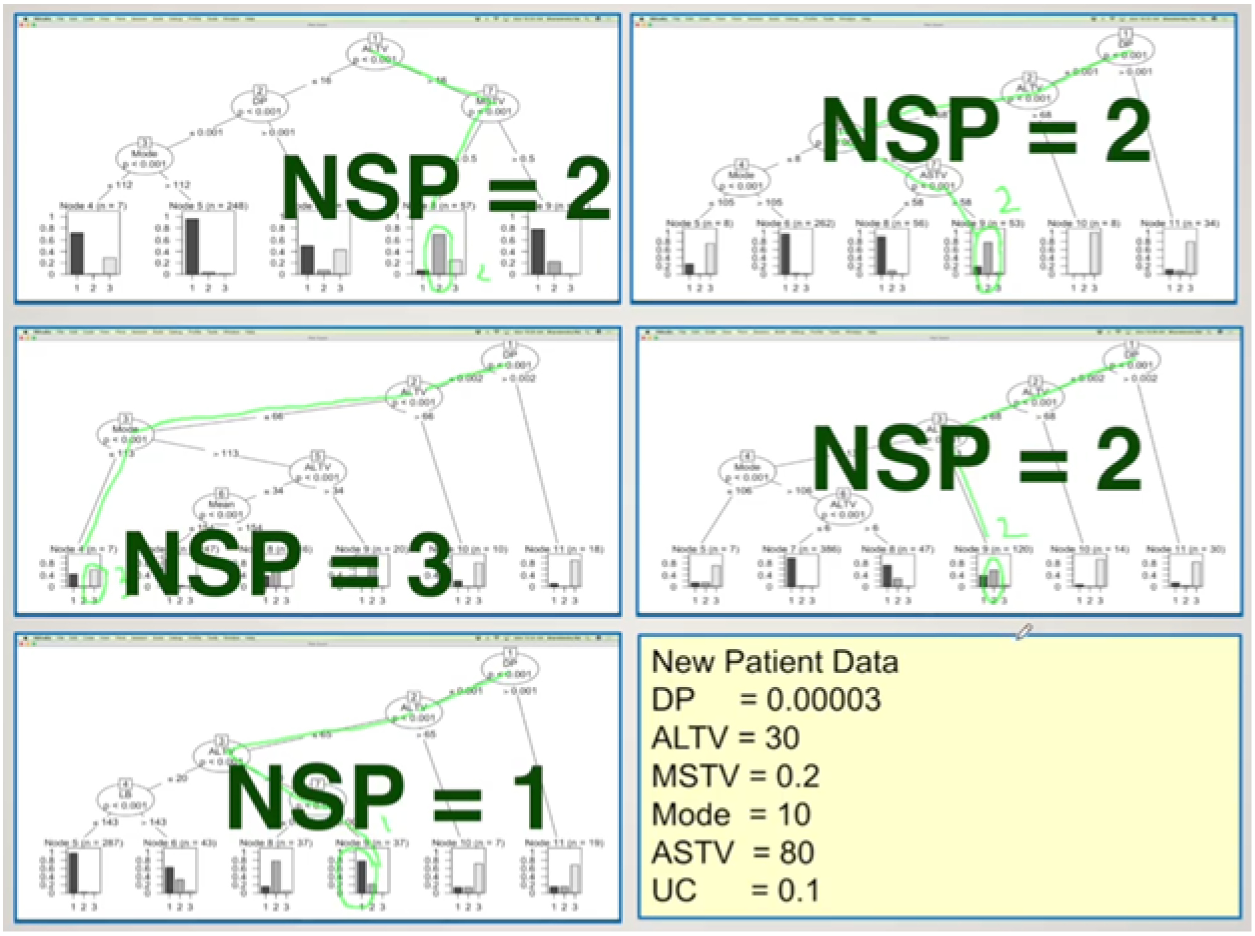
Consider an example below

```r
knitr::include_graphics("random.png")
```



New Patient Data
DP = 0.00003
ALTV = 30
MSTV = 0.2
Mode = 10
ASTV = 80
UC = 0.1

```r
knitr::include_graphics("random2.png")
```

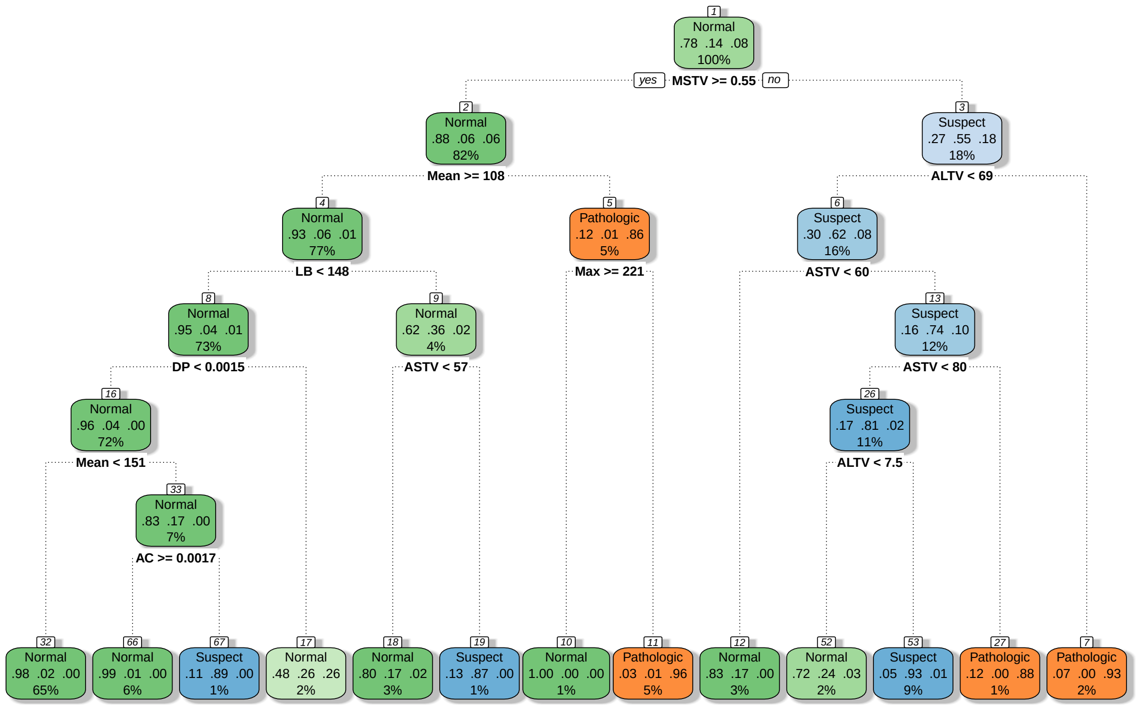


New Patient Data
DP = 0.00003
ALTV = 30
MSTV = 0.2
Mode = 10
ASTV = 80
UC = 0.1

According to the information given, the algorithm predicted that the patient belongs to suspect category. We are going to make use of random fores; however, let consider case of sample case of decision tree

Sample Decision Tree

```r
modFitA1 <- rpart(NSP ~ ., data=train_data, method="class")
fancyRpartPlot(modFitA1)
```



Rattle 2023−Oct−13 15:22:36 LUMUMBA

Prediction with Decision Tree

```r
mypred <- predict(modFitA1, newdata = train_data, type = "class")
head(mypred, 5)
```

```
         1          3          6          7          9 
   Suspect     Normal Pathologic Pathologic Pathologic 
Levels: Normal Suspect Pathologic
```

```r
head(train_data$NSP,5)
```

```
[1] Suspect    Normal     Pathologic Pathologic Pathologic
Levels: Normal Suspect Pathologic
```

```r
confusionMatrix(mypred, train_data$NSP)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   Normal Suspect Pathologic
```

```
  Normal       1143      53          8
  Suspect        11     159          2
  Pathologic      6       1        112

Overall Statistics

               Accuracy : 0.9458
                 95% CI : (0.9331, 0.9567)
    No Information Rate : 0.7759
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8462

 Mcnemar's Test P-Value : 3.327e-06

Statistics by Class:

                     Class: Normal Class: Suspect Class: Pathologic
Sensitivity                 0.9853         0.7465           0.91803
Specificity                 0.8179         0.9899           0.99490
Pos Pred Value              0.9493         0.9244           0.94118
Neg Pred Value              0.9416         0.9592           0.99273
Prevalence                  0.7759         0.1425           0.08161
Detection Rate              0.7645         0.1064           0.07492
Detection Prevalence        0.8054         0.1151           0.07960
Balanced Accuracy           0.9016         0.8682           0.95647
```

Make Prediction using the Test data

Prediction with Decision Tree

```r
mypred <- predict(modFitA1, newdata = test_data, type = "class")
head(mypred, 5)
```

```
         2          4          5          8         11
    Normal     Normal     Normal Pathologic    Suspect
Levels: Normal Suspect Pathologic
```

```r
head(test_data$NSP,5)
```

```
[1] Normal     Normal     Normal     Pathologic Suspect
Levels: Normal Suspect Pathologic
```

```r
confusionMatrix(mypred, test_data$NSP)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   Normal Suspect Pathologic
  Normal        477      31          9
  Suspect        15      50          1
  Pathologic      3       1         44

Overall Statistics

               Accuracy : 0.9049
                 95% CI : (0.8793, 0.9267)
    No Information Rate : 0.7845
    P-Value [Acc > NIR] : 6.958e-16

                  Kappa : 0.718

 Mcnemar's Test P-Value : 0.03567

Statistics by Class:

                     Class: Normal Class: Suspect Class: Pathologic
Sensitivity                 0.9636        0.60976           0.81481
Specificity                 0.7059        0.97086           0.99307
Pos Pred Value              0.9226        0.75758           0.91667
Neg Pred Value              0.8421        0.94336           0.98285
Prevalence                  0.7845        0.12995           0.08558
Detection Rate              0.7559        0.07924           0.06973
Detection Prevalence        0.8193        0.10460           0.07607
Balanced Accuracy           0.8348        0.79031           0.90394
```

DEVELOPE A RANDOM FOREST MODEL.

```
library(randomForest)
set.seed(222)
```

We will therefore estimate our classification model with NSP as our dependent variable and all the remaining variable as independent variables

```
rf <- randomForest(NSP~.,data = train_data)
```

Let us now look at the model using print function

```
print(rf)
```

```
Call:
 randomForest(formula = NSP ~ ., data = train_data)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 5.62%
Confusion matrix:
           Normal Suspect Pathologic class.error
Normal       1142      17          1  0.01551724
```

```
Suspect        47     164          2  0.23004695
Pathologic      6      11        105  0.13934426
```

The output is accompanied by the formular used, where we use NSP as DV, and all the other as IV. The data used in this algorithm is from the training data. This random forest is classification because the response variable is categorical/factor. The default number of trees is 500. The mtry, is the number of variable tried at each split is 4. Out of Bag estimate of error rate is approximately 5.75%. So we have about 94.25% accuracy, which is quite okay. The error when predicting Normal people is very low as compared to when predicting suspect or pathologic.

So what attributes does this model has, we can check using the command below;

```r
attributes(rf)
```

```
$names
 [1] "call"            "type"            "predicted"       "err.rate"
 [5] "confusion"       "votes"           "oob.times"       "classes"
 [9] "importance"      "importanceSD"    "localImportance" "proximity"
[13] "ntree"           "mtry"            "forest"          "y"
[17] "test"            "inbag"           "terms"

$class
[1] "randomForest.formula" "randomForest"
```

The model contains all the attributes listed above, for example, if we need the confusion matrix only, we can use the command below

```r
rf$confusion
```

```
           Normal Suspect Pathologic class.error
Normal       1142      17          1  0.01551724
Suspect        47     164          2  0.23004695
Pathologic      6      11        105  0.13934426
```

Prediction and Confusion Matrix

We will make prediction using the package caret

```r
library(caret)
```

We can view the prediction from the model vs the real data

```r
p1 <- predict(rf, train_data)
head(p1)
```

```
         1          3          6          7          9         10
   Suspect     Normal Pathologic Pathologic Pathologic Pathologic
Levels: Normal Suspect Pathologic
```

Observations from the real data

```r
head(train_data$NSP)
```

```
[1] Suspect    Normal     Pathologic Pathologic Pathologic Pathologic
Levels: Normal Suspect Pathologic
```

The first six prediction are all accurate, that is 100% accuracy for the six observations

```r
confusionMatrix(p1, train_data$NSP)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   Normal Suspect Pathologic
  Normal       1160       2          0
  Suspect         0     211          0
  Pathologic      0       0        122

Overall Statistics

               Accuracy : 0.9987
                 95% CI : (0.9952, 0.9998)
    No Information Rate : 0.7759
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9964

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Normal Class: Suspect Class: Pathologic
Sensitivity                 1.0000         0.9906           1.00000
Specificity                 0.9940         1.0000           1.00000
Pos Pred Value              0.9983         1.0000           1.00000
Neg Pred Value              1.0000         0.9984           1.00000
Prevalence                  0.7759         0.1425           0.08161
Detection Rate              0.7759         0.1411           0.08161
Detection Prevalence        0.7773         0.1411           0.08161
Balanced Accuracy           0.9970         0.9953           1.00000
```

From the confusion matrix, the model predicted that 1160 people belong in grouped and were alos observed to belong in group 1. The model predicted that 211 people belong in group 2 and the were observed to belong to group 2 and lastly the model predicted that 122 people belong in groups and indeed were observed to belong in group 3. However, we two people we misclassified to belong to group while they were observed to belong in group 2

Sensitity statistics tell us how often classes were correctly classified. In other words, sensitity for class 1 is 1.00, that is, we have 100 accuracy that patient in class 1 were correctly classified to belong in class 1.

Out of Baf Error

For each bootstrap iteration and related tree prediction error using data not in bootstrap sample (also called out of bag of OOB data) is estimated * Classification: Accuracy * Regression: R-Sq & SMSE

Predicting with Test Data Now that all the data points in the train data set are seen by the model, we can try predicting using the test data which is not seen by the model. We can view the prediction from the model vs the real data

```r
p2 <- predict(rf, test_data)
head(p2)
```

```
         2          4          5          8         11         16 
    Normal     Normal     Normal Pathologic    Suspect     Normal 
Levels: Normal Suspect Pathologic
```

Observations from the real data

```r
head(test_data$NSP)
```

```
[1] Normal     Normal     Normal     Pathologic Suspect    Normal    
Levels: Normal Suspect Pathologic
```

Create the Confusion Matrix

```r
confusionMatrix(p2, test_data$NSP)
```

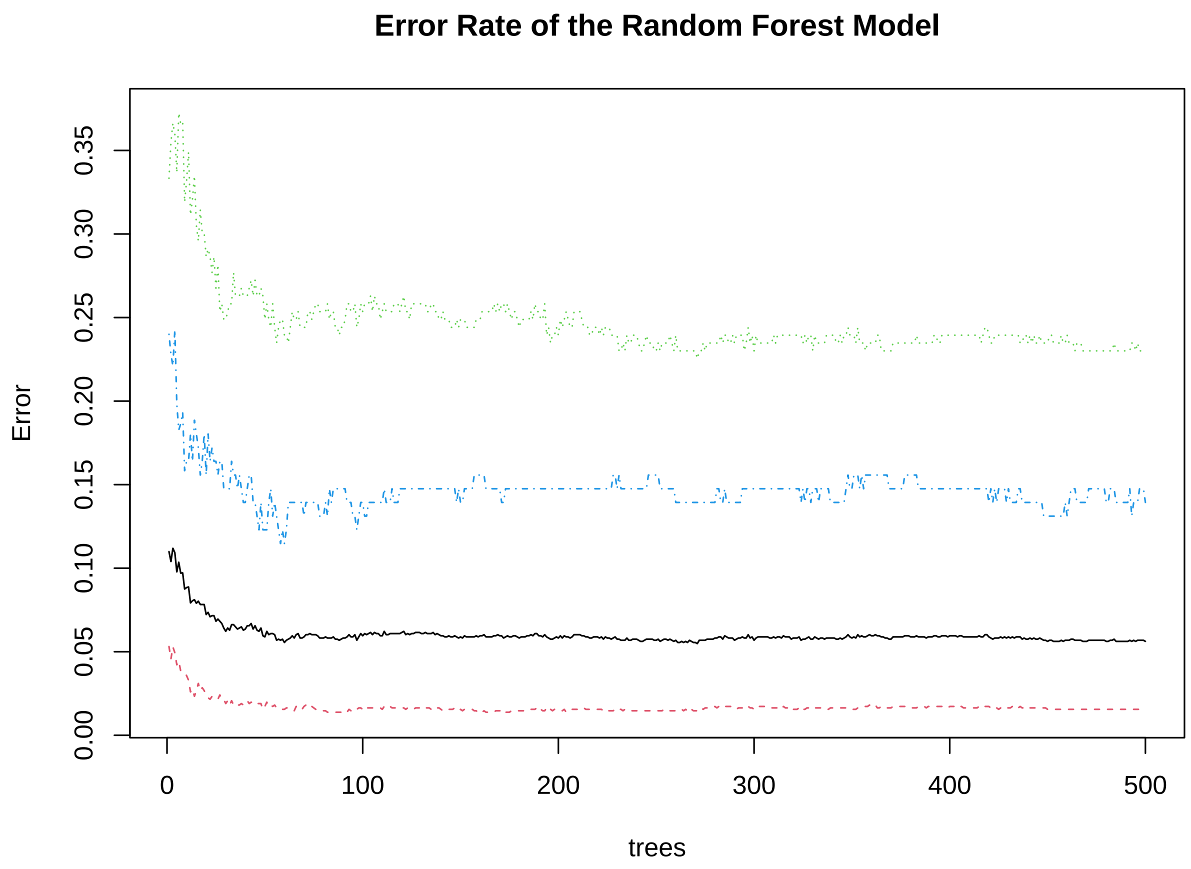
```
Confusion Matrix and Statistics

            Reference
Prediction   Normal Suspect Pathologic
  Normal        482      17          4
  Suspect        11      61          4
  Pathologic      2       4         46

Overall Statistics
                                          
               Accuracy : 0.9334          
                 95% CI : (0.9111, 0.9516)
    No Information Rate : 0.7845          
    P-Value [Acc > NIR] : <2e-16          
                                          
                  Kappa : 0.8109          
                                          
 Mcnemar's Test P-Value : 0.5823          

Statistics by Class:

                     Class: Normal Class: Suspect Class: Pathologic
Sensitivity                 0.9737        0.74390           0.85185
Specificity                 0.8456        0.97268           0.98960
Pos Pred Value              0.9583        0.80263           0.88462
Neg Pred Value              0.8984        0.96216           0.98618
Prevalence                  0.7845        0.12995           0.08558
Detection Rate              0.7639        0.09667           0.07290
Detection Prevalence        0.7971        0.12044           0.08241
Balanced Accuracy           0.9097        0.85829           0.92073
```

The test data has not been seen by the random forest model and the accuracy has come down to 93.34%. The 95% CI is still good, ranging between 91% to 95%. There is slightly higher misclassification from the test data as compared to the train data. Besides, sensitivity for preicting class 1 is still higher as compared to class 2 and 3.

Error Rate of Random Forest Model

We will make a plot using the command below to see error rate in our model

```
plot(rf, main ="Error Rate of the Random Forest Model")
```



As the number of trees grows, the out of bag error comes down to later remain constant. From the plot, we cannot improve the error after about 300 trees.

Tune the Random Forest Model

The code below is used to optimize and fine-tune parameters for a random forest model. Here's a breakdown of the function's arguments and their purposes:

train_data[,-22]: This represents the training data where all columns except the 22nd column are used as predictor variables (independent variables) for model training.

train_data[,22]: This represents the 22nd column of the training data, which is typically used as the target variable (dependent variable) that the model will aim to predict.

stepFactor = 0.5: stepFactor is a parameter that controls the size of steps to be taken when exploring the parameter space. In this case, it's set to 0.5, which means that the function will explore parameter values in increments of 0.5.

plot = TRUE: When plot is set to TRUE, it means that the function will generate plots to visualize the results of tuning.

ntreeTry = 300: ntreeTry specifies the number of trees to try when tuning the random forest model. In this case, it will try different numbers of trees in the range specified to optimize the model.

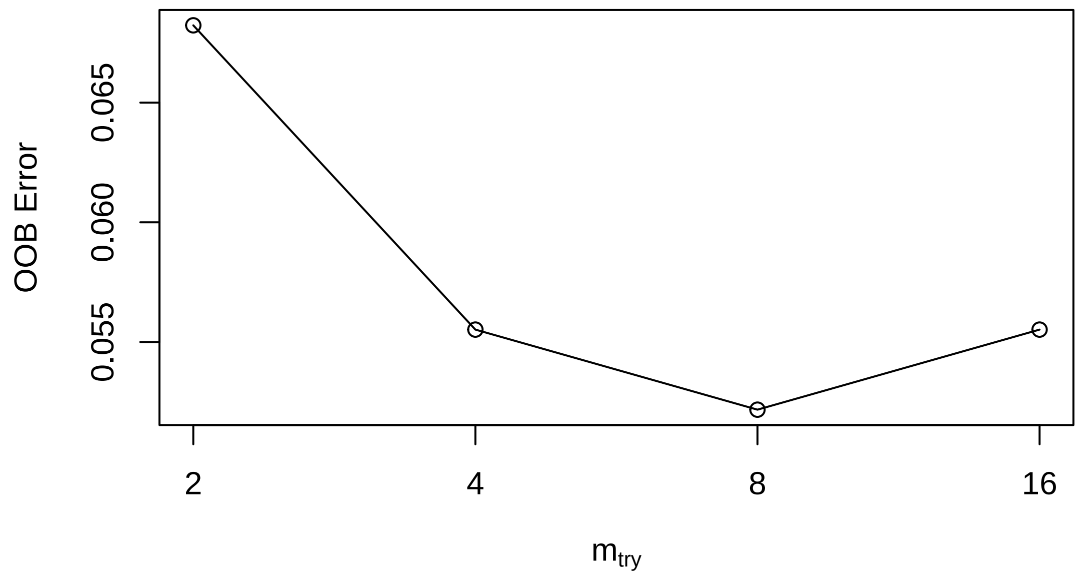
trace = TRUE: When trace is set to TRUE, it provides information on the progress of the tuning process.

improve = 0.05: improve sets the threshold for improvement. The tuning process will continue until an improvement in the model's performance (measured by the out-of-bag error) is less than or equal to 0.05.

The tuneRF function is used to find the optimal number of trees and other hyperparameters for a random forest model. It does this by conducting a search over different values for the number of trees and other hyperparameters, evaluating the model's performance, and then providing recommendations for the best combination of hyperparameters. The results are typically visualized in a plot to help you choose the best model configuration.

```r
t<- tuneRF(train_data[,-22], train_data[,22],
       stepFactor = 0.5,
       plot = TRUE,
       ntreeTry = 300,
       trace = TRUE,
       improve = 0.05)
```

```
mtry = 4  OOB error = 5.55%
Searching left ...
mtry = 8    OOB error = 5.22%
0.06024096 0.05
mtry = 16   OOB error = 5.55%
-0.06410256 0.05
Searching right ...
mtry = 2    OOB error = 6.82%
-0.3076923 0.05
```



The out of bag error is hight when mtry =2 but comes down as mtry increases to 4 and comes further down when mtry = 8 and rises when mtry = 16. This gives us an idea of what mtry value we should choose. We can now go back to our random forest model and change a few things to tune our model

```
rf2 <- randomForest(NSP~.,data = train_data,
                    ntree = 300,
                    mtry = 8,
                    importance = TRUE)
```

Print the model

```
print(rf2)
```

```
Call:
 randomForest(formula = NSP ~ ., data = train_data, ntree = 300,      mtry = 8, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 300
No. of variables tried at each split: 8

        OOB estimate of  error rate: 5.02%
Confusion matrix:
           Normal Suspect Pathologic class.error
Normal       1138      16          6  0.01896552
Suspect        41     171          1  0.19718310
Pathologic      7       4        111  0.09016393
```

Earlier, the out of bag error was 5.62% and this one, the OOB has come down to 5.28%. Besides, classification for predicting class 2 and 3 groups have improved as compared to the previous model. However, the classification error for predicting class 1 worsened. Let us check the accuracy level of our model

```r
p_1 <- predict(rf2, train_data)
head(p_1)
```

```
         1          3          6          7          9         10
   Suspect     Normal Pathologic Pathologic Pathologic Pathologic
Levels: Normal Suspect Pathologic
```

```r
head(train_data$NSP)
```

```
[1] Suspect    Normal     Pathologic Pathologic Pathologic Pathologic
Levels: Normal Suspect Pathologic
```

```r
confusionMatrix(p_1, train_data$NSP)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   Normal Suspect Pathologic
  Normal       1159       1          0
  Suspect         1     212          0
  Pathologic      0       0        122

Overall Statistics

               Accuracy : 0.9987
                 95% CI : (0.9952, 0.9998)
    No Information Rate : 0.7759
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9964

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Normal Class: Suspect Class: Pathologic
Sensitivity                 0.9991         0.9953           1.00000
Specificity                 0.9970         0.9992           1.00000
Pos Pred Value              0.9991         0.9953           1.00000
Neg Pred Value              0.9970         0.9992           1.00000
Prevalence                  0.7759         0.1425           0.08161
Detection Rate              0.7753         0.1418           0.08161
Detection Prevalence        0.7759         0.1425           0.08161
Balanced Accuracy           0.9981         0.9973           1.00000
```

The real test to see how our model is performing will be based on the test data as shown below

```r
p_2 <- predict(rf2, test_data)
head(p_2)
```

```
         2          4          5          8         11         16 
    Normal     Normal     Normal Pathologic    Suspect     Normal 
Levels: Normal Suspect Pathologic
```

```r
head(test_data$NSP)
```

```
[1] Normal     Normal     Normal     Pathologic Suspect    Normal    
Levels: Normal Suspect Pathologic
```

```r
confusionMatrix(p_2, test_data$NSP)
```

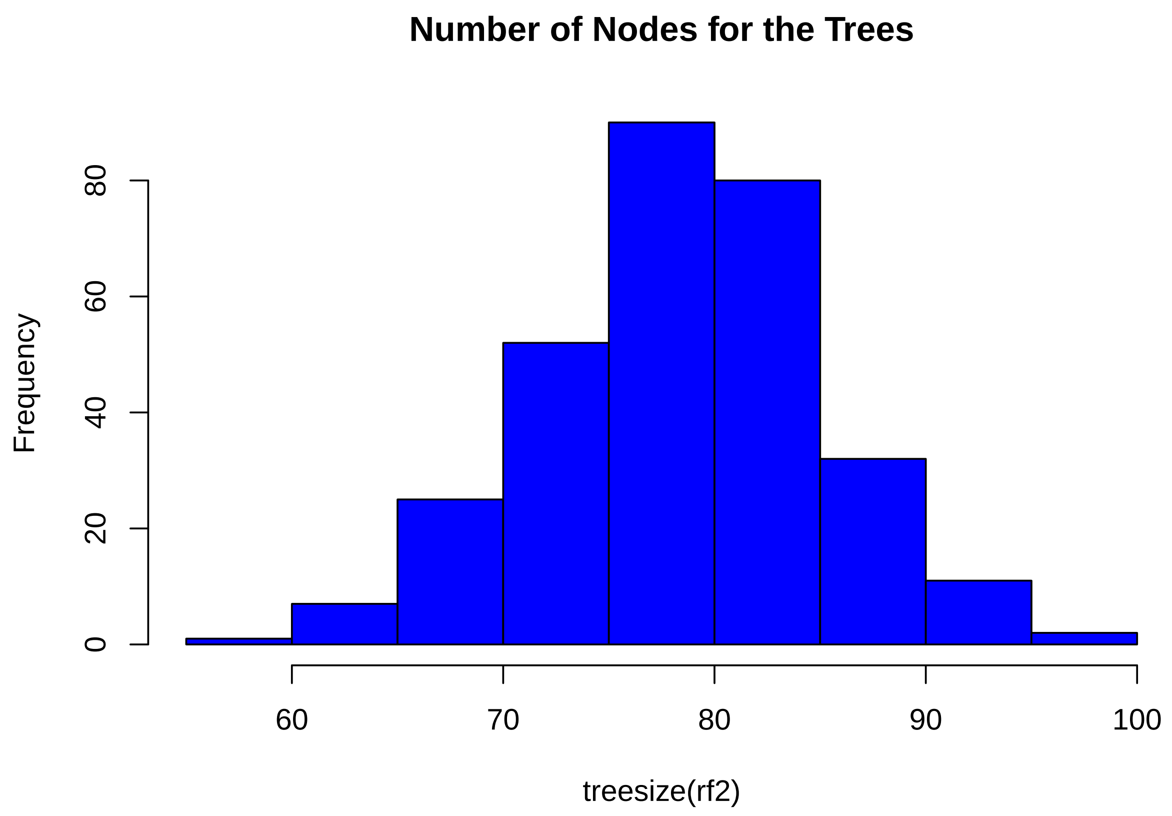
```
Confusion Matrix and Statistics

            Reference
Prediction   Normal Suspect Pathologic
  Normal        482      16          3
  Suspect        11      61          2
  Pathologic      2       5         49

Overall Statistics
                                          
               Accuracy : 0.9382          
                 95% CI : (0.9165, 0.9557)
    No Information Rate : 0.7845          
    P-Value [Acc > NIR] : <2e-16          
                                          
                  Kappa : 0.8256          
                                          
 Mcnemar's Test P-Value : 0.4915          

Statistics by Class:

                     Class: Normal Class: Suspect Class: Pathologic
Sensitivity                 0.9737        0.74390           0.90741
Specificity                 0.8603        0.97632           0.98787
Pos Pred Value              0.9621        0.82432           0.87500
Neg Pred Value              0.9000        0.96230           0.99130
Prevalence                  0.7845        0.12995           0.08558
Detection Rate              0.7639        0.09667           0.07765
Detection Prevalence        0.7940        0.11727           0.08875
Balanced Accuracy           0.9170        0.86011           0.94764
```

No much changes in our model but there are some improvements

Number of Nodes for Trees

Remember we have 300 trees in this model

```r
hist(treesize(rf2),
     main = "Number of Nodes for the Trees",
     col = "blue")
```

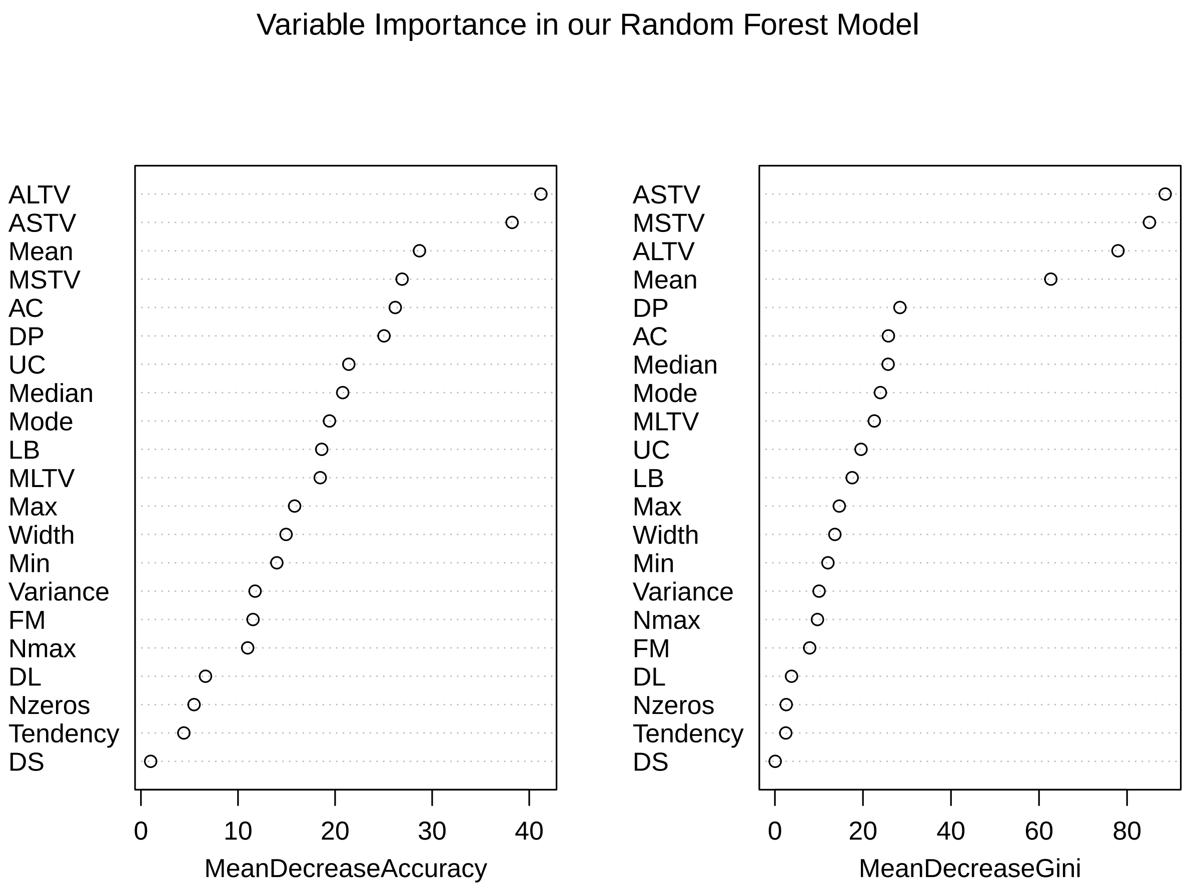


The histogram shows the distribution of the number of nodes in each of those 300 trees. From the histogram above, there are about 80 trees with approximately 80 nodes in each. We also have few trees with sixty nodes and also few trees with more than 100 nodes. Majority of trees have an average of 80 nodes.

Variable Importance

We can also find which variable played an important role in the model using the command below

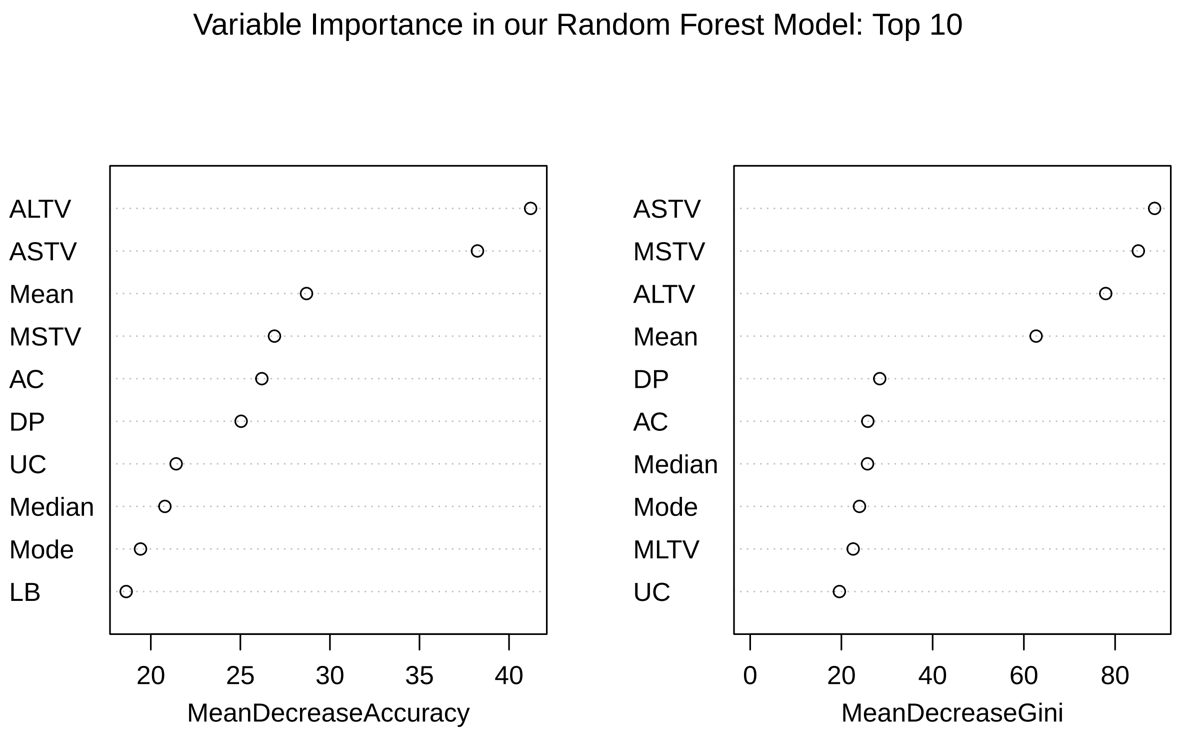
```r
varImpPlot(rf2, main = "Variable Importance in our Random Forest Model")
```



The chart shows how worst the model would perform if we remove each variable. In other words, the first chart shows the mean decrease accuracy when a variable is removed. Some variable have a higher contribution to our model as compared to other. For example, DS variable has almost zero contribution while variable such as ASTV abd ALTV have higher contribution to our model. The second chart measure how pure the nodes are at the end of the tree without each variable. From the chart, the first four stands out as the most significant predictors, where if we remove them from model, mean Gini decreases significantly.

Let us view the fisrt top ten variables

```r
varImpPlot(rf2,
           sort = TRUE,
           n.var = 10,
           main = "Variable Importance in our Random Forest Model: Top 10")
```



We can as well get the quantitative value to evaluate the variable importance.

```r
rf2$importance
```

```
               Normal      Suspect    Pathologic MeanDecreaseAccuracy
LB       1.553953e-02 0.0258200214  1.071221e-02         1.657837e-02
AC       2.943162e-02 0.0871365675  4.320153e-02         3.856144e-02
FM       2.486324e-03 0.0114239468  3.655310e-03         3.824810e-03
UC       5.177781e-03 0.0477066403  3.462117e-02         1.362509e-02
DL       1.390607e-03 0.0011991263  1.702274e-02         2.630076e-03
DS       7.698229e-06 0.0000000000  0.000000e+00         6.006006e-06
DP       1.127090e-02 0.0061564706  5.551498e-02         1.410008e-02
ASTV     2.836333e-02 0.2595434435  1.826003e-01         7.354497e-02
MSTV     2.988848e-02 0.1897893962  2.050543e-01         6.687412e-02
ALTV     3.806223e-02 0.1688399845  1.836021e-01         6.840553e-02
MLTV     6.206474e-03 0.0319690528  3.022055e-02         1.179525e-02
Width    1.034185e-02 0.0101418426  3.082483e-02         1.190649e-02
Min      9.490114e-03 0.0115299398  3.972243e-02         1.219832e-02
Max      1.356609e-02 0.0105077479  8.713479e-03         1.272368e-02
Nmax     3.965936e-03 0.0029536825  1.343104e-02         4.631204e-03
Nzeros   7.601632e-04 0.0015629388 -1.894073e-05         8.093686e-04
Mode     2.117570e-02 0.0233126568  3.929863e-02         2.287848e-02
Mean     3.464728e-02 0.0451425315  2.300968e-01         5.206579e-02
Median   2.396184e-02 0.0295561776  5.979045e-02         2.760608e-02
Variance 9.396708e-03 0.0087543808  2.225253e-02         1.037441e-02
Tendency 6.445705e-04 0.0004500091  2.162051e-03         7.167475e-04
         MeanDecreaseGini
LB            17.54785188
AC            25.78439515
```

```
FM               7.89618149
UC              19.55790339
DL               3.77273295
DS               0.06106956
DP              28.40131621
ASTV            88.66220697
MSTV            85.09145239
ALTV            77.94261740
MLTV            22.57844113
Width           13.62028859
Min             12.04096306
Max             14.62494129
Nmax             9.66268459
Nzeros           2.56566805
Mode            23.96025039
Mean            62.67833384
Median          25.72019197
Variance        10.03170156
Tendency         2.46446976
```

Let us now get which predictor variables are actually used in the random forest

```r
varUsed(rf2)
```

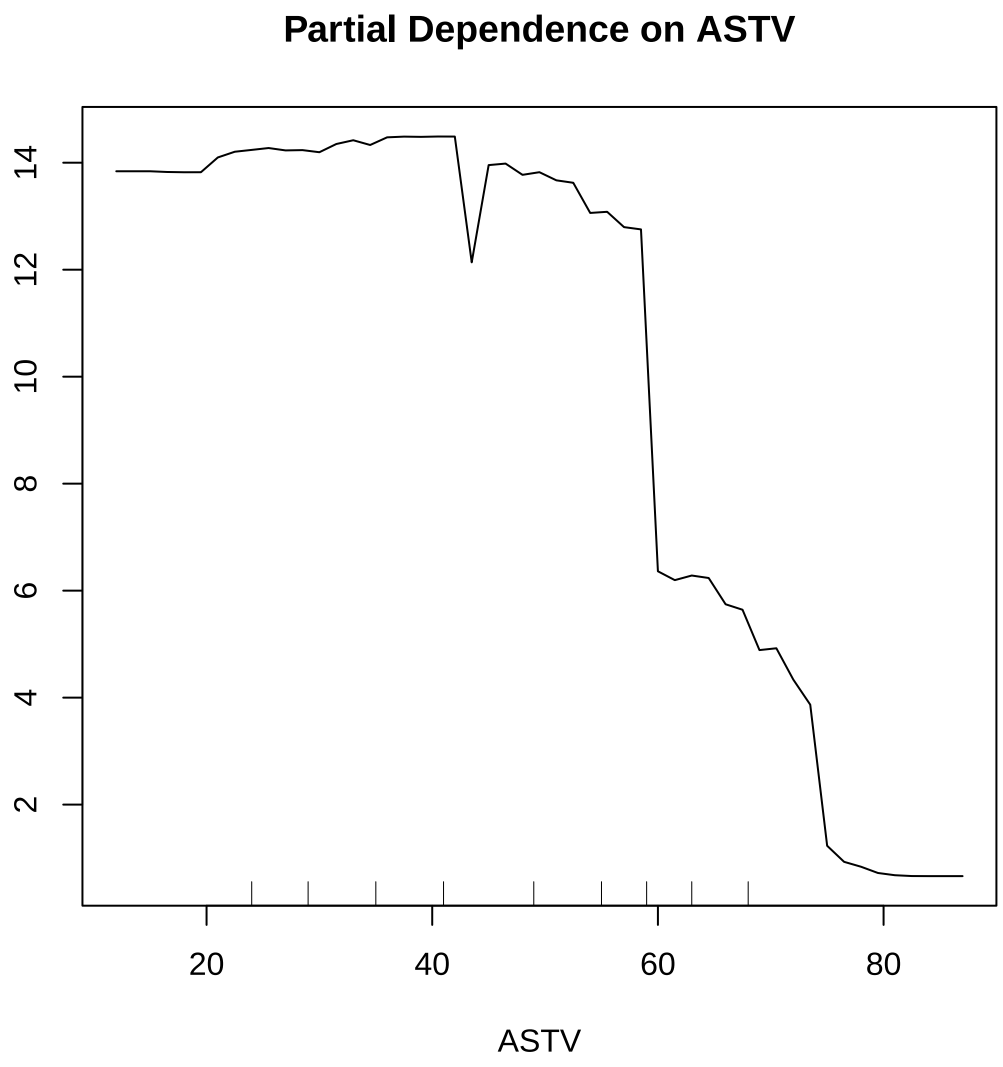
```
 [1] 1299  996  897 1605  375    5  724 2156 1278 2240 1553 1247 1143 1352  982
[16]  327 1227 1504 1296  881  322
```

The results above shows how many time each variable occurred/was used in the model. DS was only used twice in the model.

Partial Dependence Plot

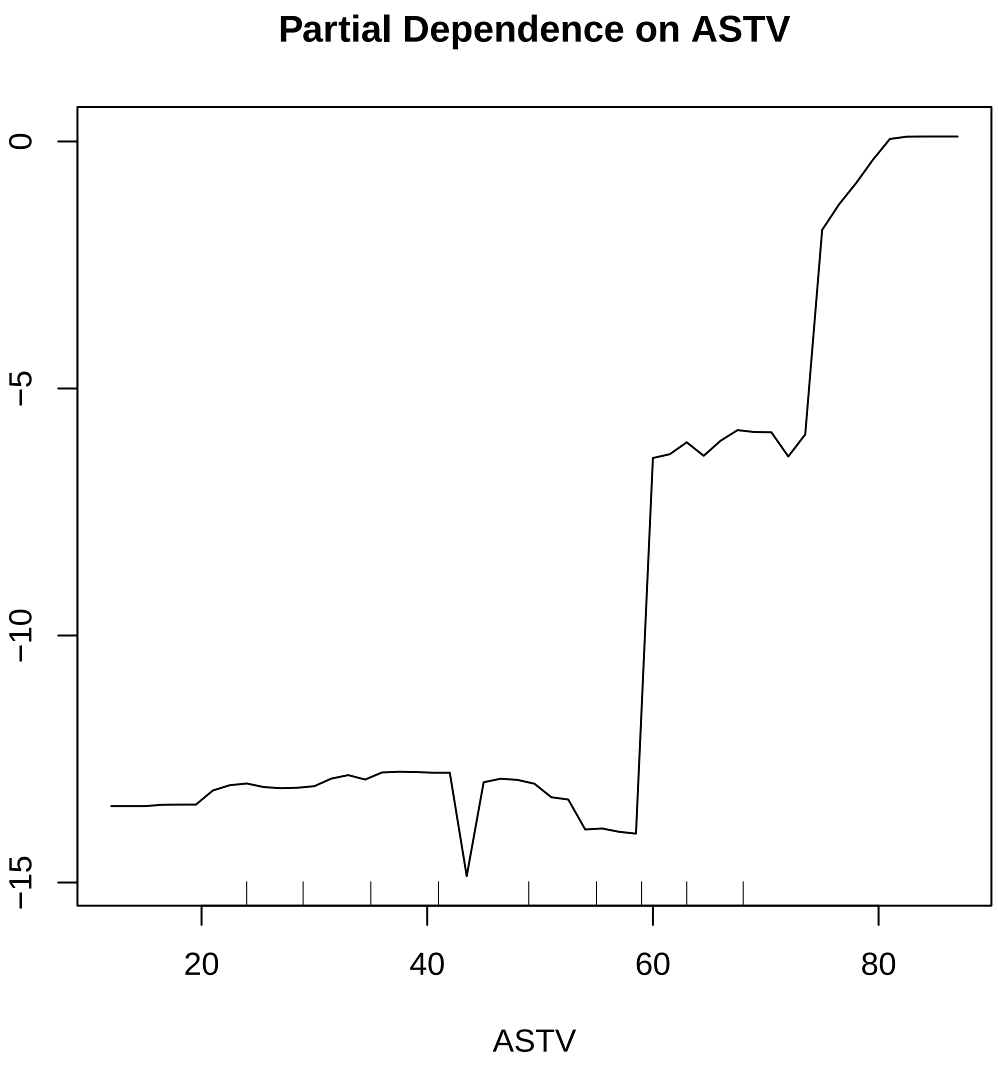
partial dependence plot give a good graphical depiction of marginal effect of a variable on the class probability (classification) or response (regression)

```r
partialPlot(rf2, train_data, ASTV, "Normal")
```



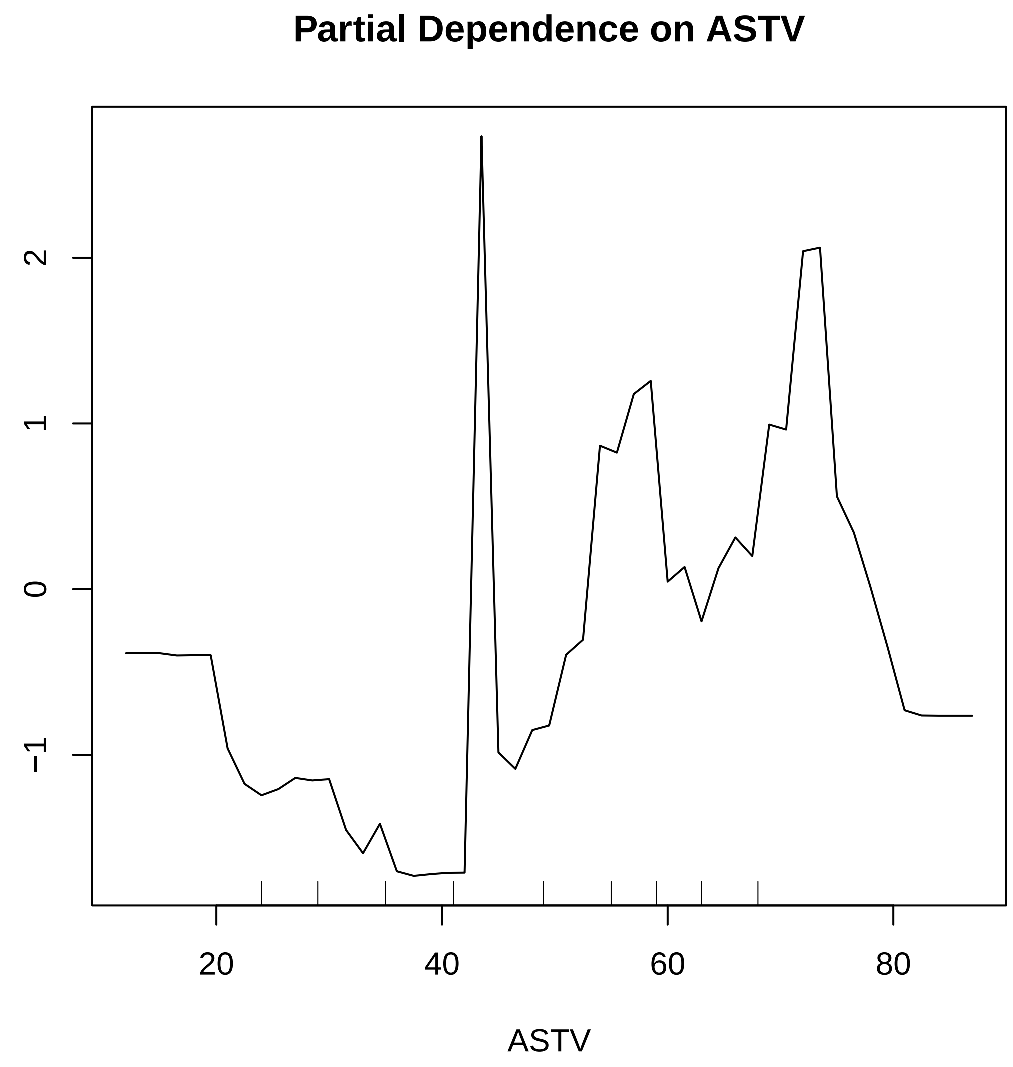
When ASTV is less than 60, the model predicts the patient to belong in class 1 as compared to when ASTV is more than 60. Similarly we can look at the plot for class 3

```r
partialPlot(rf2, train_data, ASTV, "Pathologic")
```



From the plot, when ASTV is more than 60, the model predict class 3 more than when ASTV is less than 60. From the previous results we saw that misclassification was generally high in class two. Lets make a plot to ASTV for class 2

```r
partialPlot(rf2, train_data, ASTV, "Suspect")
```



We can see there is more confusion with class 2.

Extracting A Single Tree

Let us get the first tree

```
getTree(rf2, 1, labelVar = TRUE)
```

```
    left daughter right daughter split var  split point status prediction
1               2              3      ALTV 7.500000e+00      1       <NA>
2               4              5      Mean 1.080000e+02      1       <NA>
3               6              7      MSTV 4.500000e-01      1       <NA>
4               8              9        DP 5.181345e-04      1       <NA>
5              10             11      ASTV 7.400000e+01      1       <NA>
6              12             13      ALTV 6.850000e+01      1       <NA>
```

```
7              14             15       Max 1.850000e+02      1        <NA>
8              16             17       Max 1.775000e+02      1        <NA>
9              18             19     Width 7.250000e+01      1        <NA>
10             20             21        DP 1.489997e-03      1        <NA>
11             22             23    Median 1.520000e+02      1        <NA>
12             24             25       Min 1.255000e+02      1        <NA>
13              0              0      <NA> 0.000000e+00     -1 Pathologic
14             26             27        UC 3.834495e-03      1        <NA>
15             28             29      ASTV 5.050000e+01      1        <NA>
16              0              0      <NA> 0.000000e+00     -1 Pathologic
17              0              0      <NA> 0.000000e+00     -1     Normal
18              0              0      <NA> 0.000000e+00     -1    Suspect
19              0              0      <NA> 0.000000e+00     -1 Pathologic
20             30             31      Mode 1.040000e+02      1        <NA>
21             32             33      Mode 1.145000e+02      1        <NA>
22             34             35      MSTV 5.500000e-01      1        <NA>
23             36             37      Mode 1.530000e+02      1        <NA>
24             38             39        UC 5.547580e-03      1        <NA>
25             40             41       Max 1.425000e+02      1        <NA>
26             42             43        LB 1.365000e+02      1        <NA>
27             44             45      Mean 1.585000e+02      1        <NA>
28              0              0      <NA> 0.000000e+00     -1     Normal
29             46             47      MSTV 9.500000e-01      1        <NA>
30             48             49        DP 8.785115e-04      1        <NA>
31             50             51        LB 1.425000e+02      1        <NA>
32             52             53       Min 6.900000e+01      1        <NA>
33             54             55      ASTV 6.350000e+01      1        <NA>
34              0              0      <NA> 0.000000e+00     -1 Pathologic
35              0              0      <NA> 0.000000e+00     -1    Suspect
36              0              0      <NA> 0.000000e+00     -1     Normal
37              0              0      <NA> 0.000000e+00     -1    Suspect
38             56             57       Min 1.235000e+02      1        <NA>
39             58             59      Nmax 5.000000e-01      1        <NA>
40              0              0      <NA> 0.000000e+00     -1 Pathologic
41             60             61      ASTV 8.000000e+01      1        <NA>
42             62             63     Width 1.025000e+02      1        <NA>
43             64             65      MSTV 5.500000e-01      1        <NA>
44             66             67     Width 1.195000e+02      1        <NA>
45              0              0      <NA> 0.000000e+00     -1    Suspect
46             68             69     Width 1.320000e+02      1        <NA>
47              0              0      <NA> 0.000000e+00     -1    Suspect
48              0              0      <NA> 0.000000e+00     -1    Suspect
49              0              0      <NA> 0.000000e+00     -1     Normal
50             70             71      MSTV 4.500000e-01      1        <NA>
51             72             73      ASTV 4.200000e+01      1        <NA>
52              0              0      <NA> 0.000000e+00     -1    Suspect
53              0              0      <NA> 0.000000e+00     -1     Normal
54             74             75        FM 1.870177e-01      1        <NA>
55              0              0      <NA> 0.000000e+00     -1 Pathologic
56             76             77        FM 2.219280e-03      1        <NA>
57              0              0      <NA> 0.000000e+00     -1     Normal
58              0              0      <NA> 0.000000e+00     -1    Suspect
59              0              0      <NA> 0.000000e+00     -1     Normal
60             78             79      MLTV 9.100000e+00      1        <NA>
```

```
61              0              0      <NA> 0.000000e+00     -1 Pathologic
62             80             81    Median 1.415000e+02      1       <NA>
63              0              0      <NA> 0.000000e+00     -1    Suspect
64             82             83        LB 1.550000e+02      1       <NA>
65             84             85        AC 4.170140e-04      1       <NA>
66             86             87        FM 9.012281e-03      1       <NA>
67              0              0      <NA> 0.000000e+00     -1    Suspect
68              0              0      <NA> 0.000000e+00     -1    Suspect
69              0              0      <NA> 0.000000e+00     -1     Normal
70             88             89     Width 2.200000e+01      1       <NA>
71             90             91     Width 1.750000e+01      1       <NA>
72              0              0      <NA> 0.000000e+00     -1     Normal
73             92             93     Width 2.350000e+01      1       <NA>
74              0              0      <NA> 0.000000e+00     -1     Normal
75             94             95      MSTV 1.850000e+00      1       <NA>
76             96             97        FM 1.703756e-03      1       <NA>
77              0              0      <NA> 0.000000e+00     -1    Suspect
78             98             99        UC 8.173059e-03      1       <NA>
79              0              0      <NA> 0.000000e+00     -1     Normal
80              0              0      <NA> 0.000000e+00     -1     Normal
81            100            101     Width 5.100000e+01      1       <NA>
82            102            103    Median 1.595000e+02      1       <NA>
83              0              0      <NA> 0.000000e+00     -1     Normal
84            104            105        LB 1.455000e+02      1       <NA>
85            106            107        UC 3.772002e-03      1       <NA>
86            108            109      Nmax 6.500000e+00      1       <NA>
87              0              0      <NA> 0.000000e+00     -1    Suspect
88              0              0      <NA> 0.000000e+00     -1    Suspect
89              0              0      <NA> 0.000000e+00     -1     Normal
90            110            111      ASTV 5.600000e+01      1       <NA>
91            112            113        DP 8.138395e-04      1       <NA>
92            114            115  Tendency 5.000000e-01      1       <NA>
93            116            117      MLTV 8.450000e+00      1       <NA>
94              0              0      <NA> 0.000000e+00     -1     Normal
95            118            119  Variance 4.650000e+01      1       <NA>
96            120            121      ALTV 1.500000e+01      1       <NA>
97              0              0      <NA> 0.000000e+00     -1 Pathologic
98            122            123      Mode 1.355000e+02      1       <NA>
99              0              0      <NA> 0.000000e+00     -1     Normal
100             0              0      <NA> 0.000000e+00     -1     Normal
101             0              0      <NA> 0.000000e+00     -1    Suspect
102             0              0      <NA> 0.000000e+00     -1    Suspect
103             0              0      <NA> 0.000000e+00     -1     Normal
104           124            125      ASTV 4.350000e+01      1       <NA>
105             0              0      <NA> 0.000000e+00     -1     Normal
106             0              0      <NA> 0.000000e+00     -1     Normal
107             0              0      <NA> 0.000000e+00     -1    Suspect
108           126            127      Mode 1.620000e+02      1       <NA>
109           128            129  Variance 6.000000e+00      1       <NA>
110             0              0      <NA> 0.000000e+00     -1     Normal
111             0              0      <NA> 0.000000e+00     -1    Suspect
112           130            131    Median 1.475000e+02      1       <NA>
113           132            133        FM 2.242762e-01      1       <NA>
114             0              0      <NA> 0.000000e+00     -1    Suspect
```

```
115             0              0    <NA> 0.000000e+00      -1     Normal
116             0              0    <NA> 0.000000e+00      -1     Normal
117           134            135     Min 1.345000e+02       1       <NA>
118             0              0    <NA> 0.000000e+00      -1    Suspect
119             0              0    <NA> 0.000000e+00      -1 Pathologic
120           136            137     Max 1.500000e+02       1       <NA>
121             0              0    <NA> 0.000000e+00      -1    Suspect
122           138            139    Nmax 1.500000e+00       1       <NA>
123           140            141    ASTV 5.850000e+01       1       <NA>
124             0              0    <NA> 0.000000e+00      -1     Normal
125           142            143    ALTV 9.500000e+00       1       <NA>
126           144            145    ALTV 6.050000e+01       1       <NA>
127             0              0    <NA> 0.000000e+00      -1    Suspect
128           146            147     Min 1.015000e+02       1       <NA>
129             0              0    <NA> 0.000000e+00      -1     Normal
130             0              0    <NA> 0.000000e+00      -1     Normal
131           148            149    MLTV 9.350000e+00       1       <NA>
132             0              0    <NA> 0.000000e+00      -1     Normal
133           150            151    Mode 1.380000e+02       1       <NA>
134           152            153    ASTV 4.350000e+01       1       <NA>
135           154            155    MSTV 7.500000e-01       1       <NA>
136             0              0    <NA> 0.000000e+00      -1 Pathologic
137             0              0    <NA> 0.000000e+00      -1     Normal
138             0              0    <NA> 0.000000e+00      -1    Suspect
139             0              0    <NA> 0.000000e+00      -1 Pathologic
140             0              0    <NA> 0.000000e+00      -1     Normal
141             0              0    <NA> 0.000000e+00      -1    Suspect
142             0              0    <NA> 0.000000e+00      -1     Normal
143           156            157     Max 1.470000e+02       1       <NA>
144             0              0    <NA> 0.000000e+00      -1     Normal
145           158            159     Min 1.160000e+02       1       <NA>
146             0              0    <NA> 0.000000e+00      -1     Normal
147             0              0    <NA> 0.000000e+00      -1    Suspect
148             0              0    <NA> 0.000000e+00      -1     Normal
149           160            161      AC 1.562341e-03       1       <NA>
150             0              0    <NA> 0.000000e+00      -1     Normal
151             0              0    <NA> 0.000000e+00      -1    Suspect
152           162            163     Min 8.950000e+01       1       <NA>
153           164            165      DL 3.508204e-03       1       <NA>
154             0              0    <NA> 0.000000e+00      -1     Normal
155             0              0    <NA> 0.000000e+00      -1    Suspect
156             0              0    <NA> 0.000000e+00      -1     Normal
157             0              0    <NA> 0.000000e+00      -1    Suspect
158             0              0    <NA> 0.000000e+00      -1    Suspect
159             0              0    <NA> 0.000000e+00      -1     Normal
160           166            167     Min 1.225000e+02       1       <NA>
161             0              0    <NA> 0.000000e+00      -1     Normal
162             0              0    <NA> 0.000000e+00      -1     Normal
163             0              0    <NA> 0.000000e+00      -1    Suspect
164           168            169    ALTV 6.500000e+00       1       <NA>
165             0              0    <NA> 0.000000e+00      -1    Suspect
166             0              0    <NA> 0.000000e+00      -1    Suspect
167             0              0    <NA> 0.000000e+00      -1     Normal
168             0              0    <NA> 0.000000e+00      -1     Normal
```

```
169             0              0      <NA> 0.000000e+00     -1    Suspect
```

From the results, whenever it says, the status is -1, it means, this node is a terminal node and the classification based on this terminal node is the patient NSP value is 2 or the patient is suspect. Similarly at various terminals where status is negative the model predict patients' NSP as 3 and 1 as well.

Testing the Model by classifying patients with following information.

```
patients <- read.csv("patients.csv")
patients$NSP <- factor(patients$NSP, levels = c(1,2,3),
                       labels = c("Normal","Suspect","Pathologic"))
str(patients)
```

```
'data.frame':   6 obs. of  22 variables:
 $ LB      : int  134 122 122 122 151 131
 $ AC      : num  0.0014 0 0 0 0 ...
 $ FM      : num  0 0 0 0 0 ...
 $ UC      : num  0.012623 0 0.001517 0.002967 0.000834 ...
 $ DL      : num  0.008415 0 0 0 0.000834 ...
 $ DS      : int  0 0 0 0 0 0
 $ DP      : num  0.00281 0 0 0 0 ...
 $ ASTV    : int  29 83 84 86 64 28
 $ MSTV    : num  6.3 0.5 0.5 0.3 1.9 1.5
 $ ALTV    : int  0 6 5 6 9 0
 $ MLTV    : num  0 15.6 13.6 10.6 27.6 5.4
 $ Width   : int  150 68 68 68 130 87
 $ Min     : int  50 62 62 62 56 71
 $ Max     : int  200 130 130 130 186 158
 $ Nmax    : int  6 0 0 1 2 2
 $ Nzeros  : int  3 0 0 0 0 0
 $ Mode    : int  71 122 122 122 150 141
 $ Mean    : int  107 122 122 122 148 137
 $ Median  : int  106 123 123 123 151 141
 $ Variance: int  215 3 3 1 9 10
 $ Tendency: int  0 1 1 1 1 1
 $ NSP     : Factor w/ 3 levels "Normal","Suspect",..: 3 3 3 3 2 1
```

```
head(patients,5)
```

```
   LB          AC FM          UC          DL DS          DP ASTV MSTV ALTV MLTV
1 134 0.001402525  0 0.012622721 0.008415147  0 0.002805049   29  6.3    0  0.0
2 122 0.000000000  0 0.000000000 0.000000000  0 0.000000000   83  0.5    6 15.6
3 122 0.000000000  0 0.001517451 0.000000000  0 0.000000000   84  0.5    5 13.6
4 122 0.000000000  0 0.002967359 0.000000000  0 0.000000000   86  0.3    6 10.6
5 151 0.000000000  0 0.000834028 0.000834028  0 0.000000000   64  1.9    9 27.6
  Width Min Max Nmax Nzeros Mode Mean Median Variance Tendency        NSP
1   150  50 200    6      3   71  107    106      215        0 Pathologic
2    68  62 130    0      0  122  122    123        3        1 Pathologic
3    68  62 130    0      0  122  122    123        3        1 Pathologic
4    68  62 130    1      0  122  122    123        1        1 Pathologic
5   130  56 186    2      0  150  148    151        9        1    Suspect
```

```r
# Use the trained random forest model to classify the patient
predicted_class <- predict(rf2, patients)
```

```r
# The 'predicted_class' variable now contains the predicted class (1, 2, or 3)
# You can print or use it as needed
print(predicted_class)
```

```
         1          2          3          4          5          6
Pathologic Pathologic Pathologic Pathologic    Suspect     Normal
Levels: Normal Suspect Pathologic
```

```r
### Actual data
head(patients$NSP)
```

```
[1] Pathologic Pathologic Pathologic Pathologic Suspect    Normal
Levels: Normal Suspect Pathologic
```

Accuracy of the Model

```r
confusionMatrix(predicted_class, patients$NSP)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   Normal Suspect Pathologic
  Normal          1       0          0
  Suspect         0       1          0
  Pathologic      0       0          4

Overall Statistics

               Accuracy : 1
                 95% CI : (0.5407, 1)
    No Information Rate : 0.6667
    P-Value [Acc > NIR] : 0.08779

                  Kappa : 1

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Normal Class: Suspect Class: Pathologic
Sensitivity                 1.0000         1.0000            1.0000
Specificity                 1.0000         1.0000            1.0000
Pos Pred Value              1.0000         1.0000            1.0000
Neg Pred Value              1.0000         1.0000            1.0000
Prevalence                  0.1667         0.1667            0.6667
Detection Rate              0.1667         0.1667            0.6667
Detection Prevalence        0.1667         0.1667            0.6667
Balanced Accuracy           1.0000         1.0000            1.0000
```

Prediction with another set of data

Testing the Model by classifying patients with following information.

```
patients1 <- read.csv("cardio.csv")
patients1$NSP <- factor(patients1$NSP, levels = c(1,2,3),
                        labels = c("Normal","Suspect","Pathologic"))
str(patients1)
```

```
'data.frame':   19 obs. of  22 variables:
 $ LB      : int  145 145 145 145 145 145 145 145 145 145 ...
 $ AC      : num  0.00713 0.00298 0.00489 0 0.00153 ...
 $ FM      : num  0.00891 0.00298 0.00977 0.00199 0.00767 ...
 $ UC      : num  0.00178 0.00149 0.00489 0.00199 0.00307 ...
 $ DL      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ DS      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ DP      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ASTV    : int  36 34 35 34 40 41 43 41 40 41 ...
 $ MSTV    : num  1.4 1.7 1.9 1.7 1.4 1.2 1.5 1.1 1.1 1.8 ...
 $ ALTV    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ MLTV    : num  13.6 22.3 21.3 25.9 18.4 15.8 0 11.4 1.2 23.4 ...
 $ Width   : int  119 117 140 109 140 136 109 125 113 117 ...
 $ Min     : int  57 57 56 57 56 60 63 62 73 50 ...
 $ Max     : int  176 174 196 166 196 196 172 187 186 167 ...
 $ Nmax    : int  3 6 5 5 9 8 5 4 4 4 ...
 $ Nzeros  : int  1 1 0 1 1 1 0 0 0 0 ...
 $ Mode    : int  148 150 148 150 148 145 156 167 165 154 ...
 $ Mean    : int  150 147 150 147 148 148 156 164 163 151 ...
 $ Median  : int  150 150 151 150 149 149 158 166 165 154 ...
 $ Variance: int  12 11 12 10 7 6 11 11 11 16 ...
 $ Tendency: int  1 1 1 1 0 0 1 1 1 1 ...
 $ NSP     : Factor w/ 3 levels "Normal","Suspect",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
head(patients1,5)
```

```
   LB          AC          FM          UC DL DS DP ASTV MSTV ALTV MLTV Width
1 145 0.007130125 0.008912656 0.001782531  0  0  0   36  1.4    0 13.6   119
2 145 0.002976190 0.002976190 0.001488095  0  0  0   34  1.7    0 22.3   117
3 145 0.004885993 0.009771987 0.004885993  0  0  0   35  1.9    0 21.3   140
4 145 0.000000000 0.001988072 0.001988072  0  0  0   34  1.7    0 25.9   109
5 145 0.001533742 0.007668712 0.003067485  0  0  0   40  1.4    0 18.4   140
  Min Max Nmax Nzeros Mode Mean Median Variance Tendency    NSP
1  57 176    3      1  148  150    150       12        1 Normal
2  57 174    6      1  150  147    150       11        1 Normal
3  56 196    5      0  148  150    151       12        1 Normal
4  57 166    5      1  150  147    150       10        1 Normal
5  56 196    9      1  148  148    149        7        0 Normal
```

```
# Use the trained random forest model to classify the patient
predicted_class <- predict(rf2, patients1)
```

```r
# The 'predicted_class' variable now contains the predicted class (1, 2, or 3)
# You can print or use it as needed
print(predicted_class)
```

```
         1          2          3          4          5          6          7 
    Normal     Normal     Normal     Normal     Normal     Normal     Normal 
         8          9         10         11         12         13         14 
    Normal     Normal     Normal     Normal    Suspect    Suspect    Suspect 
        15         16         17         18         19 
   Suspect    Suspect Pathologic Pathologic     Normal 
Levels: Normal Suspect Pathologic
```

```r
### Actual data
head(patients1$NSP)
```

```
[1] Normal Normal Normal Normal Normal Normal
Levels: Normal Suspect Pathologic
```

Accuracy of the Model

```r
confusionMatrix(predicted_class, patients1$NSP)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   Normal Suspect Pathologic
  Normal         12       0          0
  Suspect         1       4          0
  Pathologic      0       0          2

Overall Statistics
                                          
               Accuracy : 0.9474          
                 95% CI : (0.7397, 0.9987)
    No Information Rate : 0.6842          
    P-Value [Acc > NIR] : 0.007219        
                                          
                  Kappa : 0.895           
                                          
 Mcnemar's Test P-Value : NA              

Statistics by Class:

                     Class: Normal Class: Suspect Class: Pathologic
Sensitivity                 0.9231         1.0000            1.0000
Specificity                 1.0000         0.9333            1.0000
Pos Pred Value              1.0000         0.8000            1.0000
Neg Pred Value              0.8571         1.0000            1.0000
Prevalence                  0.6842         0.2105            0.1053
Detection Rate              0.6316         0.2105            0.1053
Detection Prevalence        0.6316         0.2632            0.1053
Balanced Accuracy           0.9615         0.9667            1.0000
```

Additional Machine Learning Algorithm to Predict Heart Failure

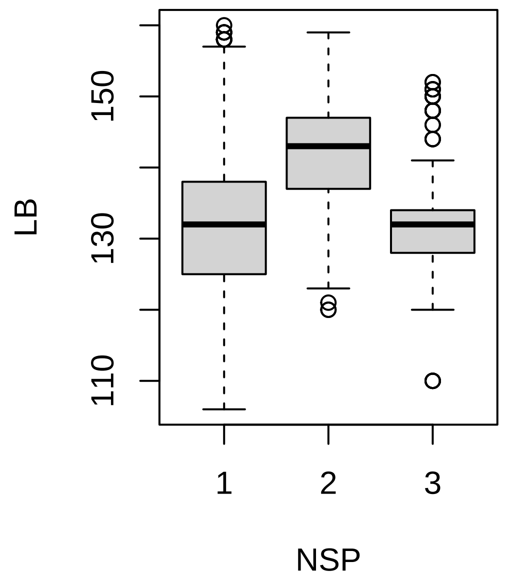
```r
mydata4 <- read.csv("Cardiotocographic.csv", header = TRUE)
mydata4$NSP <- as.factor(mydata4$NSP)
str(mydata4)
```

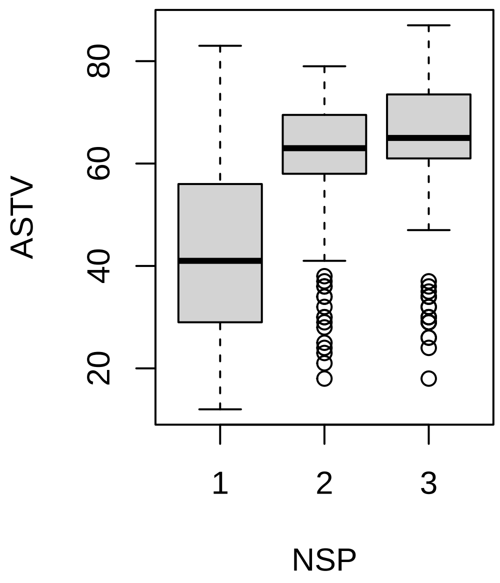
```
'data.frame':   2126 obs. of  22 variables:
 $ LB      : int  120 132 133 134 132 134 134 122 122 122 ...
 $ AC      : num  0 0.00638 0.00332 0.00256 0.00651 ...
 $ FM      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ UC      : num  0 0.00638 0.00831 0.00768 0.00814 ...
 $ DL      : num  0 0.00319 0.00332 0.00256 0 ...
 $ DS      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ DP      : num  0 0 0 0 0 ...
 $ ASTV    : int  73 17 16 16 16 26 29 83 84 86 ...
 $ MSTV    : num  0.5 2.1 2.1 2.4 2.4 5.9 6.3 0.5 0.5 0.3 ...
 $ ALTV    : int  43 0 0 0 0 0 0 6 5 6 ...
 $ MLTV    : num  2.4 10.4 13.4 23 19.9 0 0 15.6 13.6 10.6 ...
 $ Width   : int  64 130 130 117 117 150 150 68 68 68 ...
 $ Min     : int  62 68 68 53 53 50 50 62 62 62 ...
 $ Max     : int  126 198 198 170 170 200 200 130 130 130 ...
 $ Nmax    : int  2 6 5 11 9 5 6 0 0 1 ...
 $ Nzeros  : int  0 1 1 0 0 3 3 0 0 0 ...
 $ Mode    : int  120 141 141 137 137 76 71 122 122 122 ...
 $ Mean    : int  137 136 135 134 136 107 107 122 122 122 ...
 $ Median  : int  121 140 138 137 138 107 106 123 123 123 ...
 $ Variance: int  73 12 13 13 11 170 215 3 3 1 ...
 $ Tendency: int  1 0 0 1 1 0 0 1 1 1 ...
 $ NSP     : Factor w/ 3 levels "1","2","3": 2 1 1 1 1 3 3 3 3 3 ...
```

```r
attach(mydata4)
```

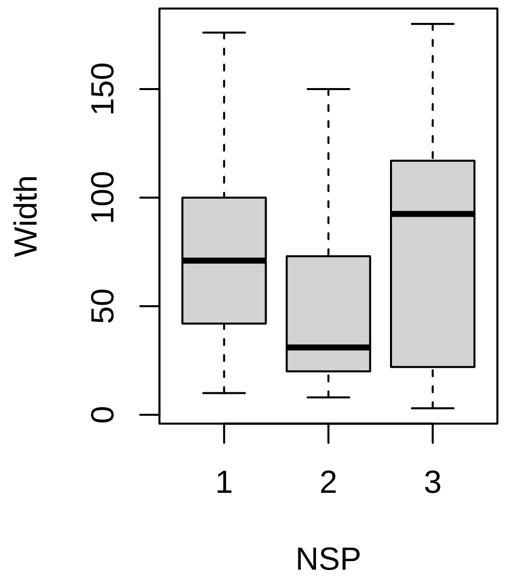
Make Some Plots

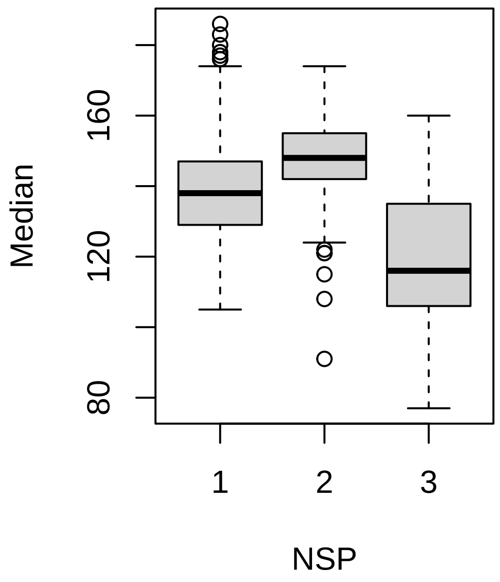
```r
par(mfrow=c(1,2))

boxplot(LB ~ NSP)
boxplot(ASTV ~ NSP)
```





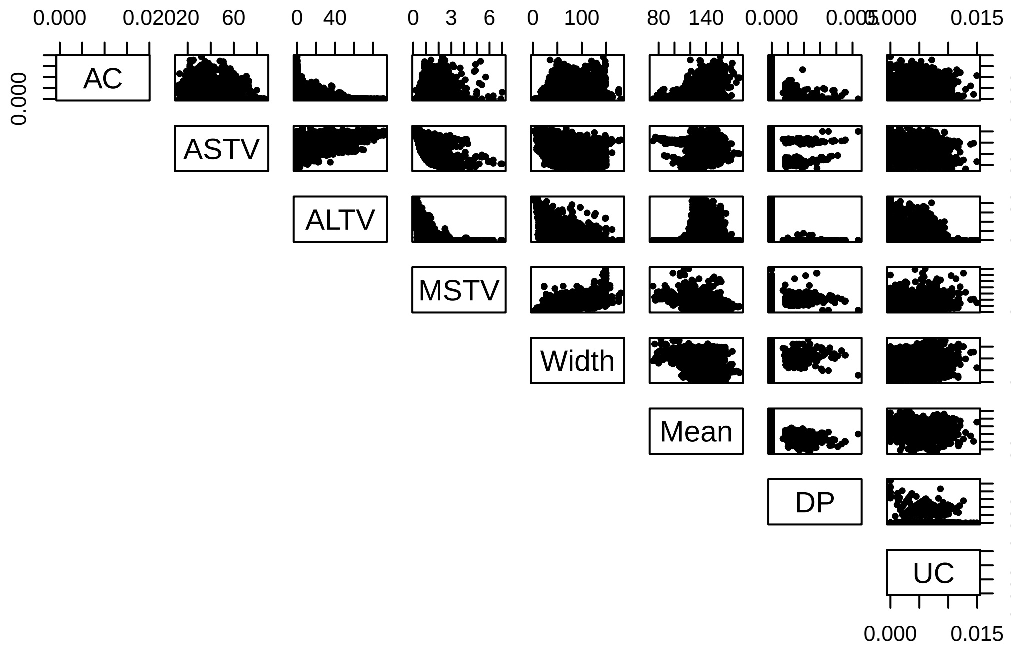
```r
boxplot(Width ~ NSP)
boxplot(Median ~ NSP)
```





Correlation Matrix

```r
par(mfrow=c(1,1))
pairs( cbind(AC, ASTV, ALTV, MSTV, Width, Mean, DP, UC), pch=19, lower.panel=NULL, cex=.5)
```



looking at classification based on p.hat = .5 cutoff

10-fold CV, repeated 5 times

```r
train_model <- trainControl(method = "repeatedcv", number = 5, repeats=10)


model.cart <- train( NSP ~.,
  data = mydata4,
  method = "rpart",
  trControl = train_model)

model.cart
```

```
CART

2126 samples
  21 predictor
   3 classes: '1', '2', '3'
```

```
No pre-processing
Resampling: Cross-Validated (5 fold, repeated 10 times)
Summary of sample sizes: 1701, 1701, 1701, 1701, 1700, 1701, ...
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.08917197  0.8742240  0.6550994
  0.17622081  0.8401331  0.5268462
  0.20806794  0.8002354  0.2604269

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.08917197.
```

```
model.cart$finalModel
```

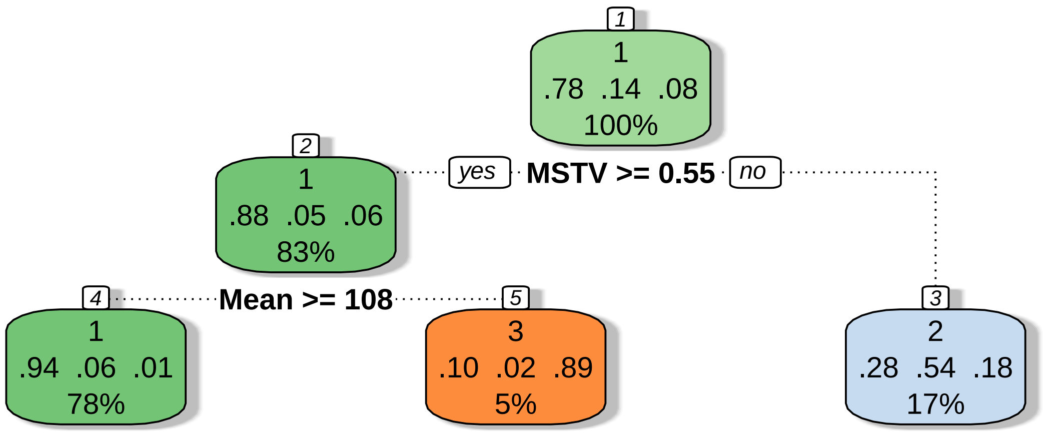
```
n= 2126

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 2126 471 1 (0.778457197 0.138758231 0.082784572)
  2) MSTV>=0.55 1754 202 1 (0.884834664 0.053591790 0.061573546)
    4) Mean>=107.5 1649 107 1 (0.935112189 0.055791389 0.009096422) *
    5) Mean< 107.5 105  12 3 (0.095238095 0.019047619 0.885714286) *
  3) MSTV< 0.55 372 171 2 (0.276881720 0.540322581 0.182795699) *
```

```
confusionMatrix(predict(model.cart, mydata4),
                reference=mydata4$NSP, positive="1")
```

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3
         1 1542   92   15
         2  103  201   68
         3   10    2   93

Overall Statistics

               Accuracy : 0.8636
                 95% CI : (0.8483, 0.8779)
    No Information Rate : 0.7785
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6292

 Mcnemar's Test P-Value : 8.841e-14

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            0.9317  0.68136  0.52841
Specificity            0.7728  0.90661  0.99385
```

```
Pos Pred Value          0.9351  0.54032  0.88571
Neg Pred Value          0.7631  0.94641  0.95893
Prevalence              0.7785  0.13876  0.08278
Detection Rate          0.7253  0.09454  0.04374
Detection Prevalence    0.7756  0.17498  0.04939
Balanced Accuracy       0.8523  0.79398  0.76113
```

```
fancyRpartPlot(model.cart$finalModel)
```



Rattle 2023−Oct−13 15:23:00 LUMUMBA

```
model.rf <- train(
  NSP ~.,
  data = mydata4,
  method = "rf",
  trControl = train_model)
model.rf
```

```
Random Forest

2126 samples
  21 predictor
   3 classes: '1', '2', '3'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 10 times)
Summary of sample sizes: 1701, 1700, 1701, 1701, 1701, 1701, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
```

```
   2    0.9318432  0.8044846
  11    0.9442125  0.8433624
  21    0.9394626  0.8303969

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 11.
```

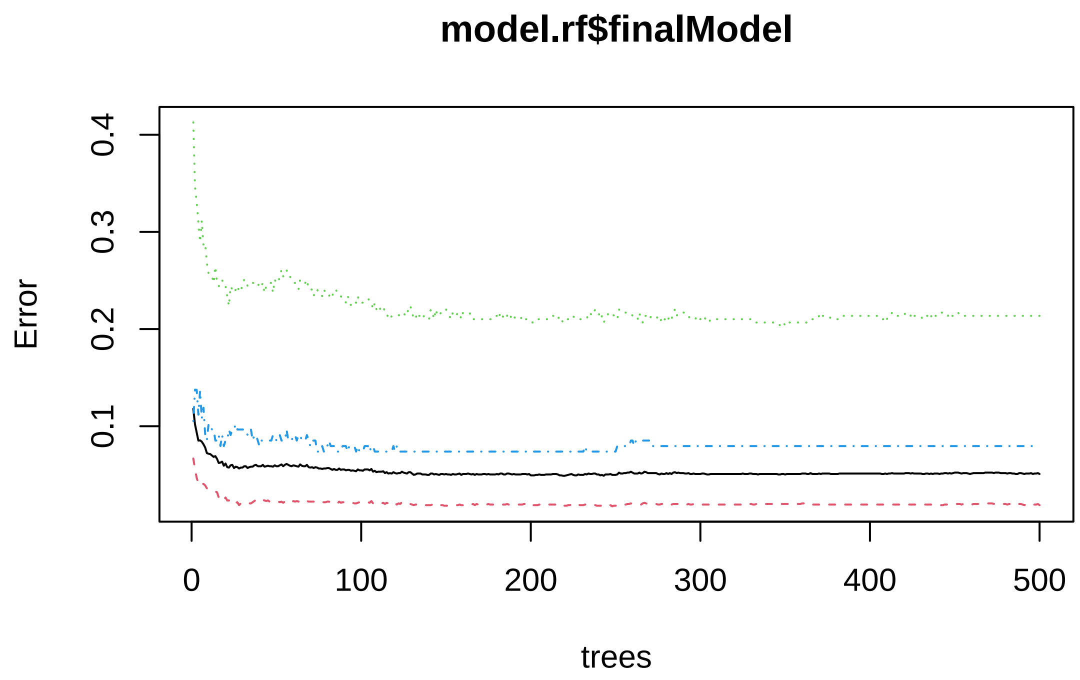
```
summary(model.rf$finalModel)
```

```
                Length Class      Mode     
call               4   -none-     call     
type               1   -none-     character
predicted       2126   factor     numeric  
err.rate        2000   -none-     numeric  
confusion         12   -none-     numeric  
votes           6378   matrix     numeric  
oob.times       2126   -none-     numeric  
classes            3   -none-     character
importance        21   -none-     numeric  
importanceSD       0   -none-     NULL     
localImportance    0   -none-     NULL     
proximity          0   -none-     NULL     
ntree              1   -none-     numeric  
mtry               1   -none-     numeric  
forest            14   -none-     list     
y               2126   factor     numeric  
test               0   -none-     NULL     
inbag              0   -none-     NULL     
xNames            21   -none-     character
problemType        1   -none-     character
tuneValue          1   data.frame list     
obsLevels          3   -none-     character
param              0   -none-     list     
```

```
model.rf$finalModel
```

```
Call:
 randomForest(x = x, y = y, mtry = param$mtry) 
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 11

        OOB estimate of  error rate: 5.08%
Confusion matrix:
     1   2   3 class.error
1 1624  23   8  0.01873112
2   57 232   6  0.21355932
3    8   6 162  0.07954545
```

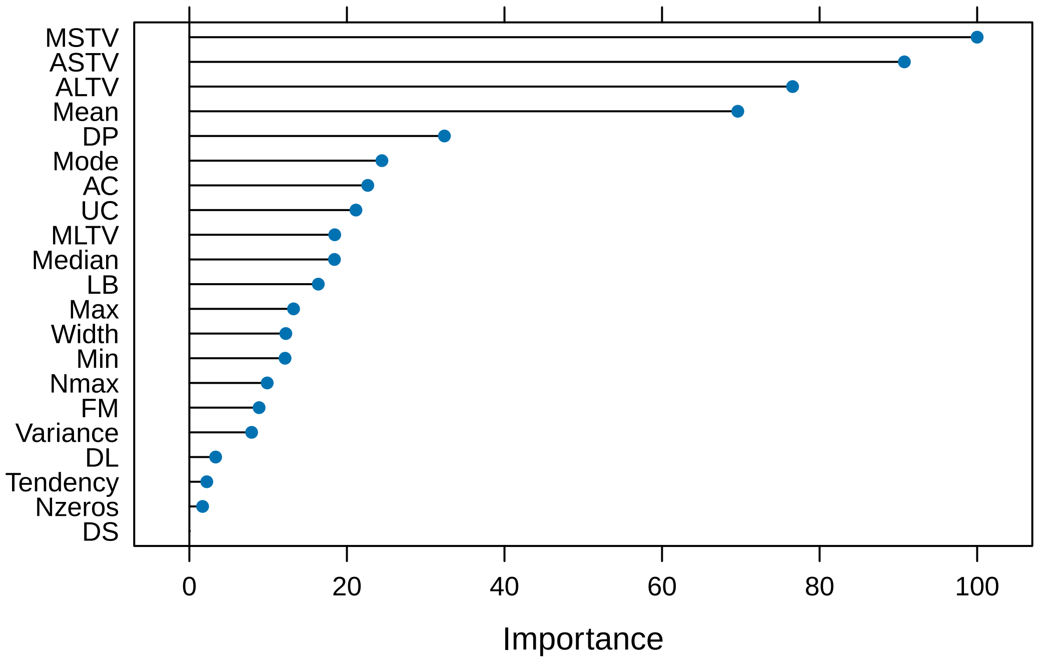
```
plot(model.rf$finalModel)
```



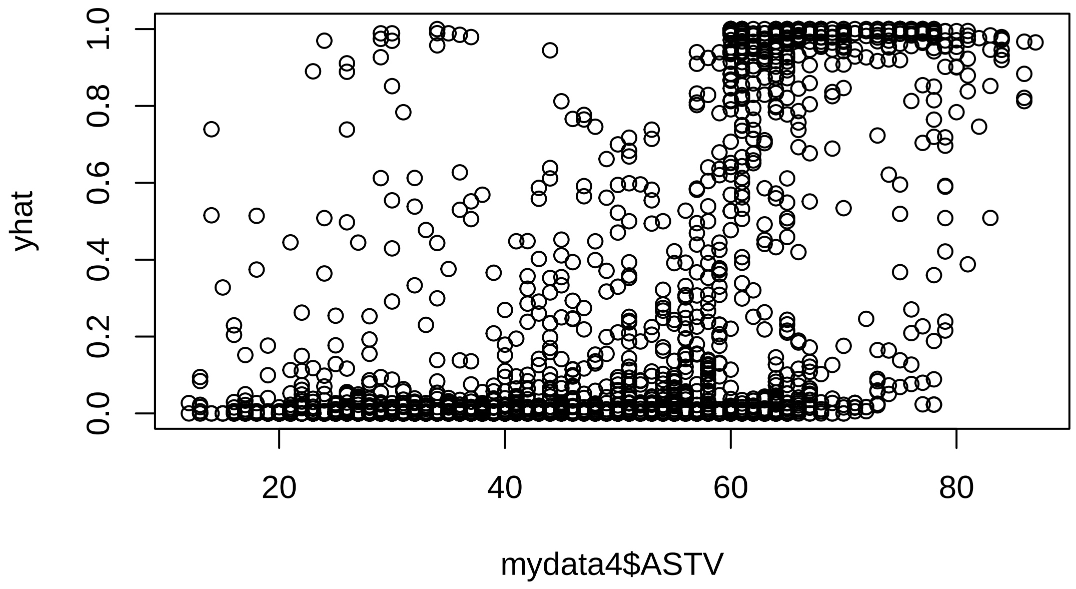
```
varImp(model.rf$finalModel)
```

```
              Overall
LB         22.9384967
AC         31.5312698
FM         12.6508906
UC         29.4776634
DL          5.1060651
DS          0.5361335
DP         44.8461939
ASTV      124.7502475
MSTV      137.3937671
ALTV      105.3323077
MLTV       25.7922601
Width      17.2978347
Min        17.1618847
Max        18.6372755
Nmax       14.0623581
Nzeros      2.8341471
Mode       33.9885301
Mean       95.8131455
Median     25.7409495
Variance   11.3537206
Tendency    3.5667884
```

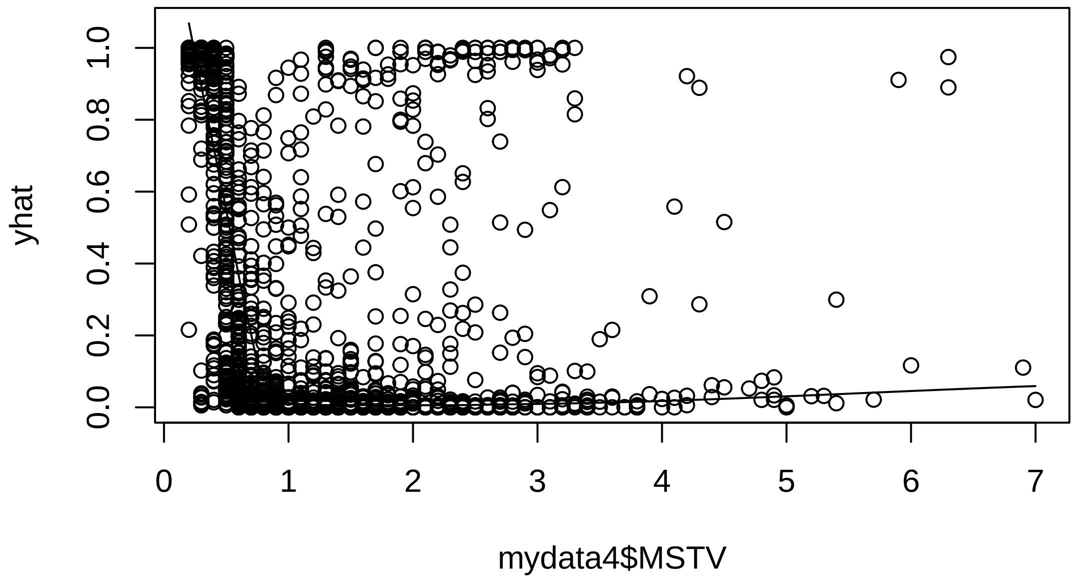
```
plot( varImp(model.rf) )
```



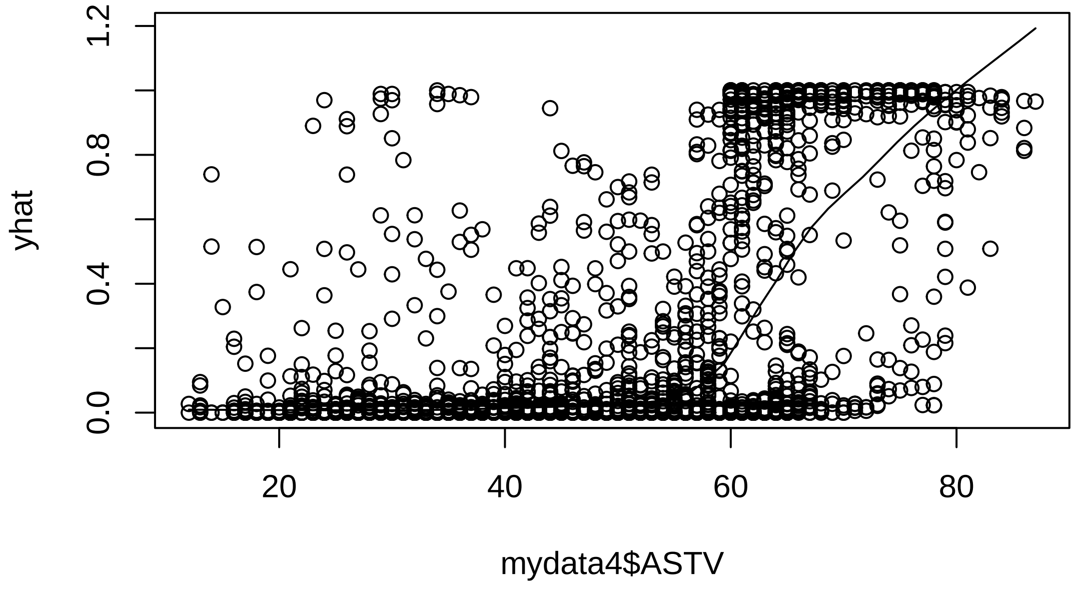
```r
yhat = 1-predict(model.rf$finalModel, type="prob")[,1]
plot(mydata4$ASTV, yhat)
```



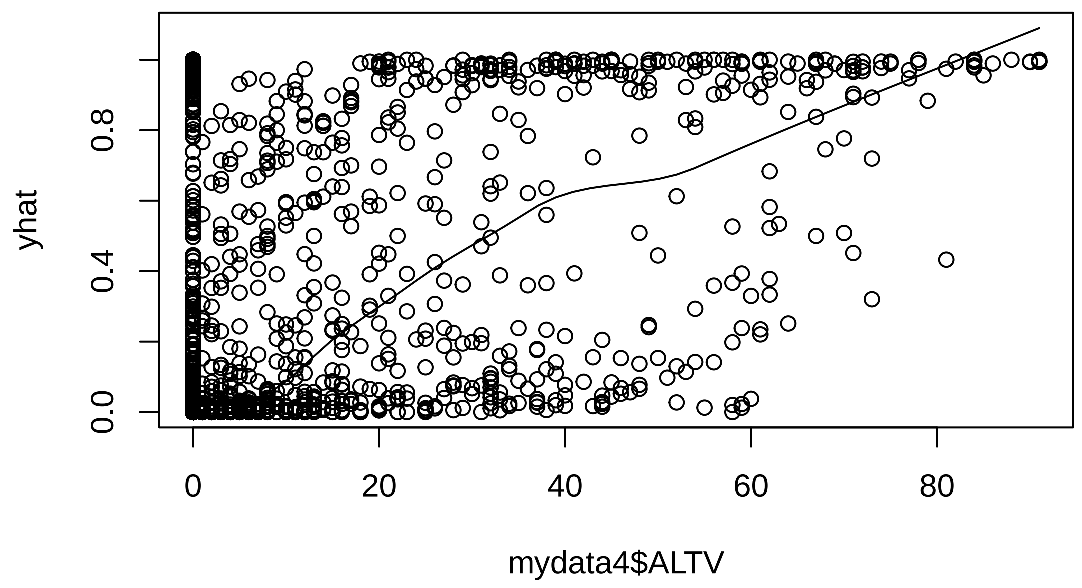
```
scatter.smooth(mydata4$MSTV, yhat, span=.4)
```



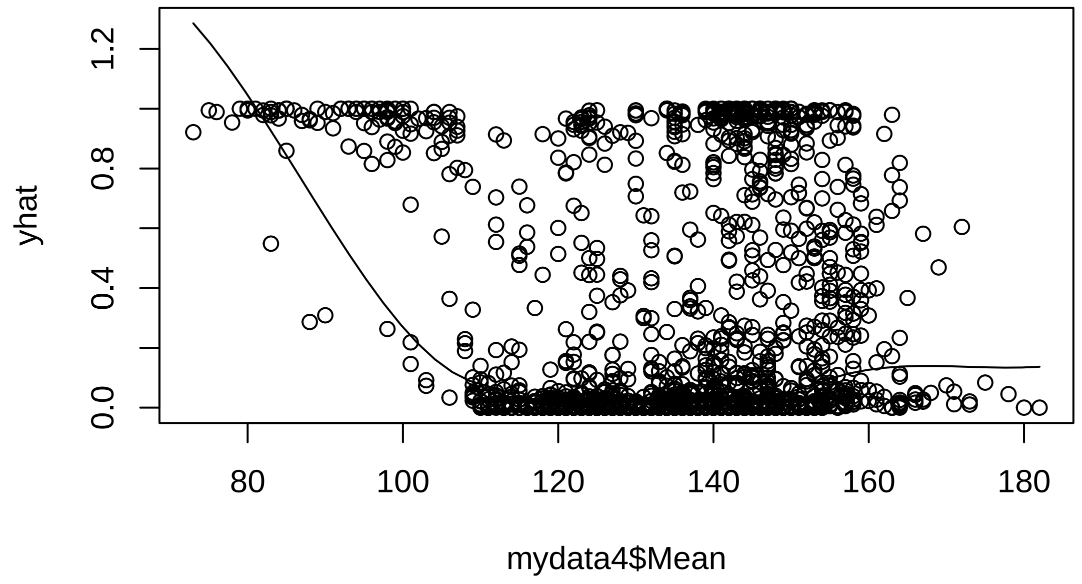
```
scatter.smooth(mydata4$ASTV, yhat, span=.4)
```



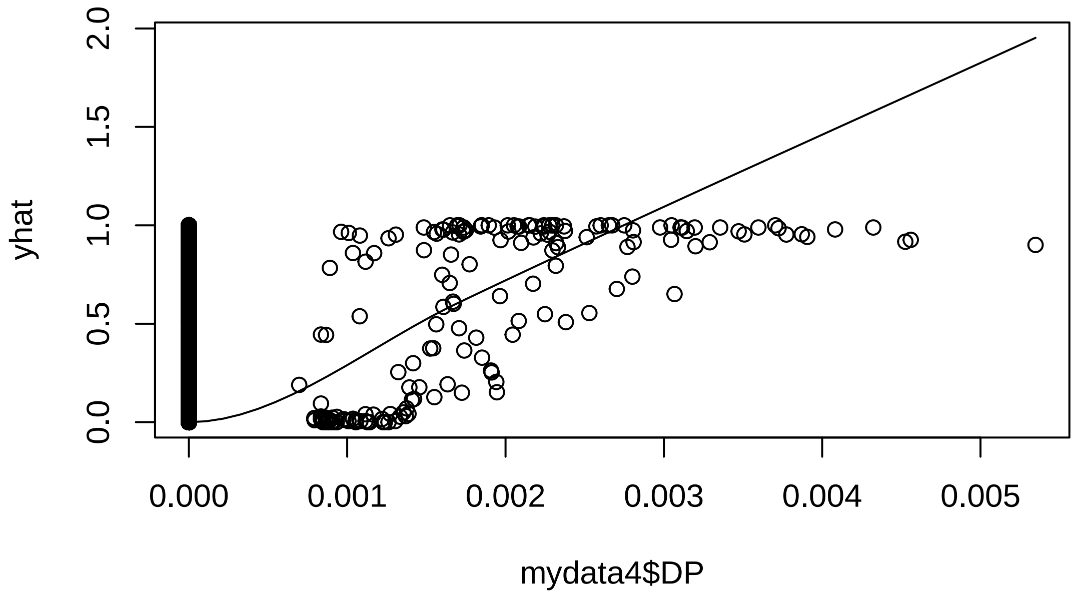
```
scatter.smooth(mydata4$ALTV, yhat, span=.4)
```



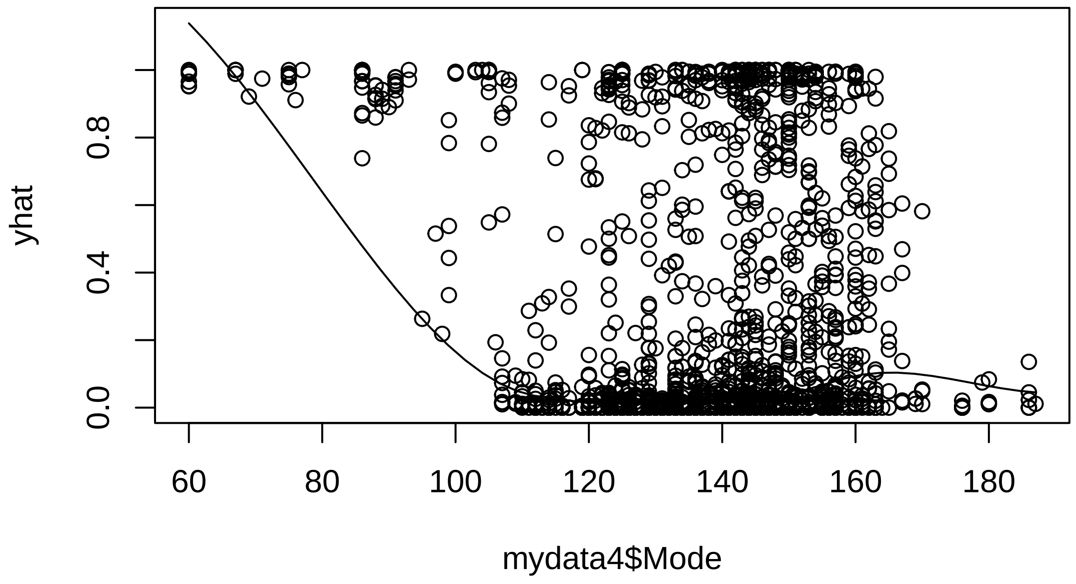
```
scatter.smooth(mydata4$Mean, yhat, span=.4)
```



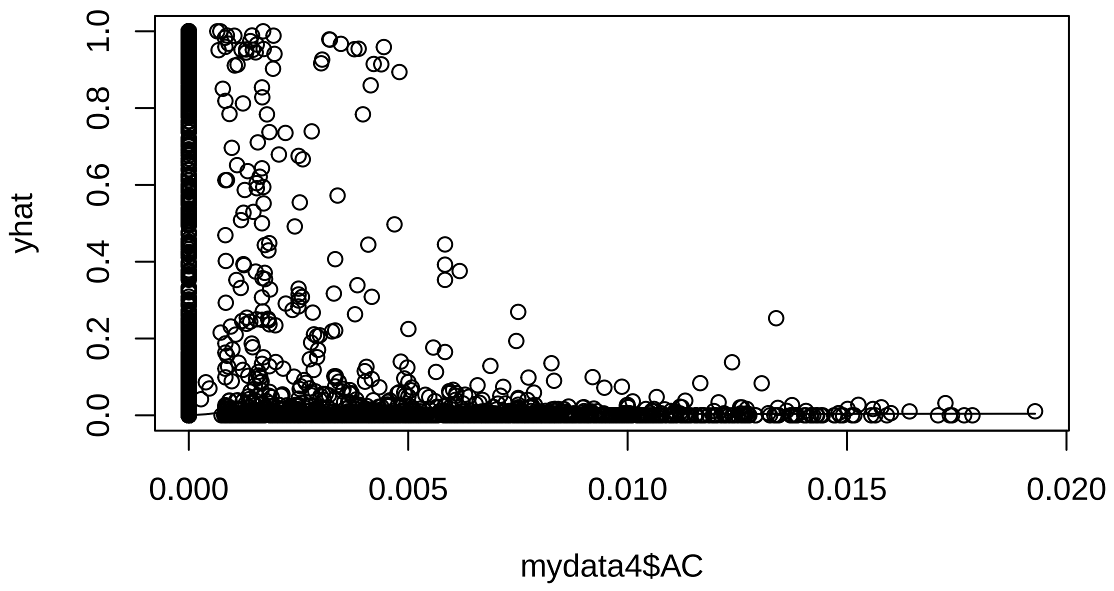
```
scatter.smooth(mydata4$DP, yhat, span=.4)
```



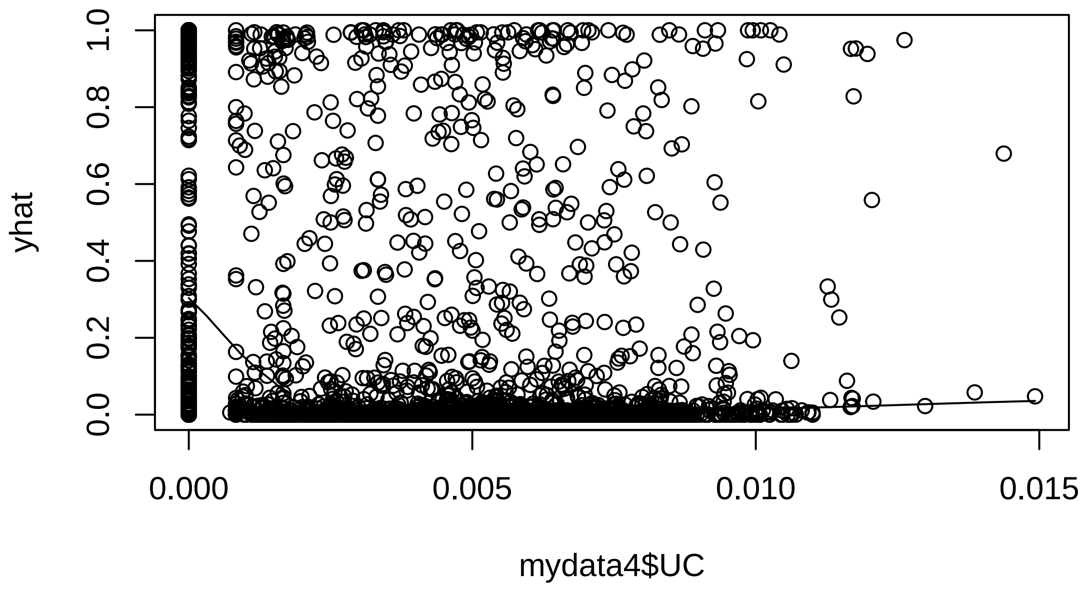
```
scatter.smooth(mydata4$Mode, yhat, span=.4)
```



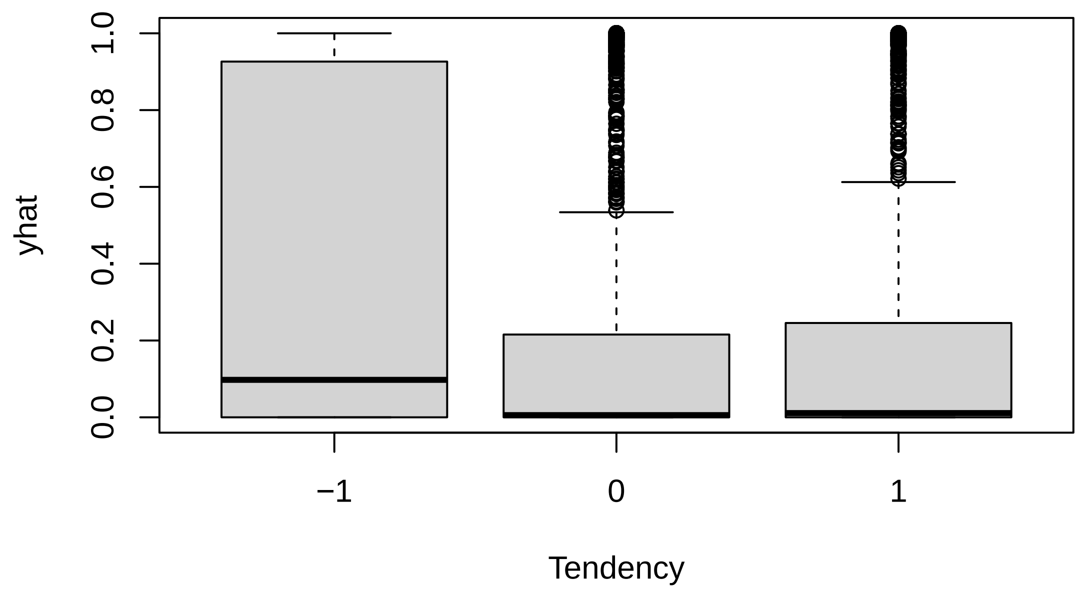
```
scatter.smooth(mydata4$AC, yhat, span=.4)
```



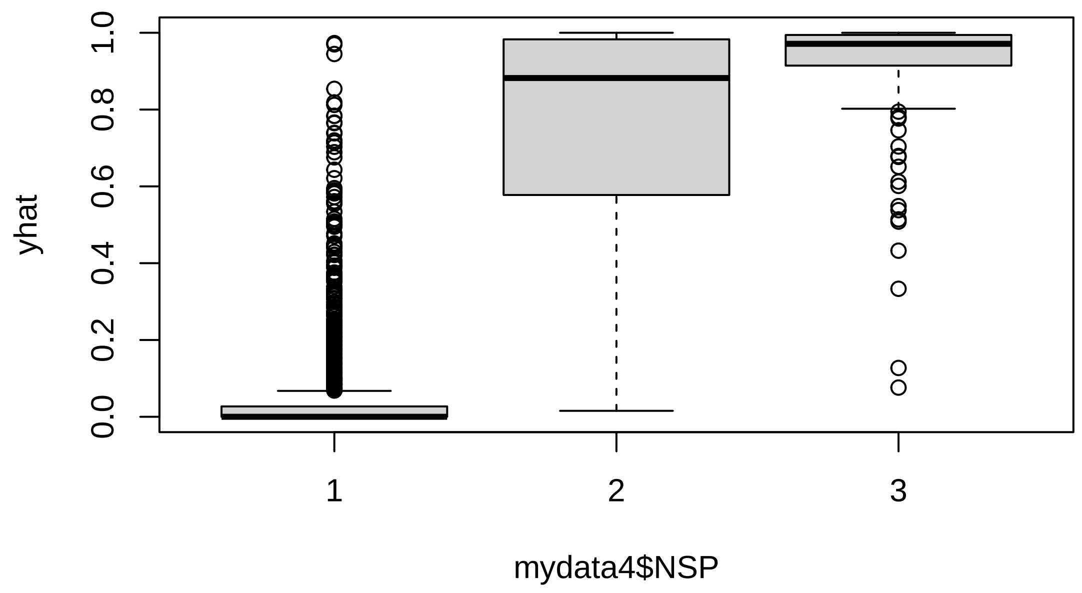
```
scatter.smooth(mydata4$UC, yhat, span=.4)
```



```r
boxplot(yhat ~ Tendency)
```



```r
boxplot(yhat ~ mydata4$NSP)
```



```r
confusionMatrix(predict(model.rf, mydata4),
                reference=mydata4$NSP, positive="1")
```

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3
         1 1655    2    0
         2    0  293    0
         3    0    0  176

Overall Statistics

               Accuracy : 0.9991
                 95% CI : (0.9966, 0.9999)
    No Information Rate : 0.7785
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9974

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            1.0000   0.9932  1.00000
Specificity            0.9958   1.0000  1.00000
Pos Pred Value         0.9988   1.0000  1.00000
Neg Pred Value         1.0000   0.9989  1.00000
Prevalence             0.7785   0.1388  0.08278
Detection Rate         0.7785   0.1378  0.08278
Detection Prevalence   0.7794   0.1378  0.08278
Balanced Accuracy      0.9979   0.9966  1.00000
```