Codes

Akshay

2023-11-29

Installing necessary packages

```
# install.packages("dplyr")
# install.packages("rpart.plot")
# install.packages("caret")
# install.packages("magrittr")
# install.packages("e1071")
# install.packages("Metrics")
# install.packages("vctrs")
# install.packages("devtools")
# devtools::install_github("dongyuanwu/RSBID")
```

1) Reading the data set Link to Dataset : https://archive.ics.uci.edu/dataset/225/ilpd+indian+liver+patient+dataset

```
df <- read.csv("…/Indian Liver Patient Dataset (ILPD).csv", stringsAsFactors
= TRUE)
```

2) Understanding the data set

```
#View(df)
dim(df)

## [1] 583  11

summary(df)

##       Age           Gender          TB                DB        
##  Min.   : 4.00   Female:142   Min.   : 0.400   Min.   : 0.100  
##  1st Qu.:33.00   Male  :441   1st Qu.: 0.800   1st Qu.: 0.200  
##  Median :45.00                Median : 1.000   Median : 0.300  
##  Mean   :44.75                Mean   : 3.299   Mean   : 1.486  
##  3rd Qu.:58.00                3rd Qu.: 2.600   3rd Qu.: 1.300  
##  Max.   :90.00                Max.   :75.000   Max.   :19.700  
##                                                                
##     Alkphos            Sgpt              Sgot              TP       
##  Min.   :  63.0   Min.   :  10.00   Min.   :  10.0   Min.   :2.700  
##  1st Qu.: 175.5   1st Qu.:  23.00   1st Qu.:  25.0   1st Qu.:5.800  
##  Median : 208.0   Median :  35.00   Median :  42.0   Median :6.600  
##  Mean   : 290.6   Mean   :  80.71   Mean   : 109.9   Mean   :6.483  
##  3rd Qu.: 298.0   3rd Qu.:  60.50   3rd Qu.:  87.0   3rd Qu.:7.200  
##  Max.   :2110.0   Max.   :2000.00   Max.   :4929.0   Max.   :9.600  
##                                                                
##       ALB          A.G.Ratio         Selector
```

```
##  Min.   :0.900   Min.   :0.3000   Min.   :1.000  
##  1st Qu.:2.600   1st Qu.:0.7000   1st Qu.:1.000  
##  Median :3.100   Median :0.9300   Median :1.000  
##  Mean   :3.142   Mean   :0.9471   Mean   :1.286  
##  3rd Qu.:3.800   3rd Qu.:1.1000   3rd Qu.:2.000  
##  Max.   :5.500   Max.   :2.8000   Max.   :2.000  
##                  NA's   :4
```

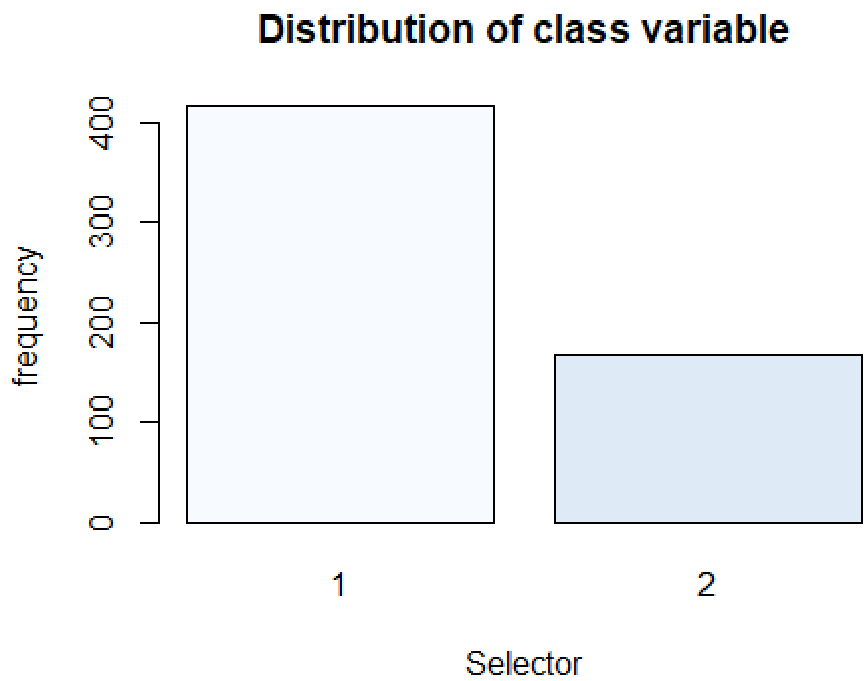
```
str(df)
```

```
## 'data.frame':    583 obs. of  11 variables:
##  $ Age      : int  65 62 62 58 72 46 26 29 17 55 ...
##  $ Gender   : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 2 1 1 2 2 ...
##  $ TB       : num  0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0.7 ...
##  $ DB       : num  0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.2 ...
##  $ Alkphos  : int  187 699 490 182 195 208 154 202 202 290 ...
##  $ Sgpt     : int  16 64 60 14 27 19 16 14 22 53 ...
##  $ Sgot     : int  18 100 68 20 59 14 12 11 19 58 ...
##  $ TP       : num  6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8 ...
##  $ ALB      : num  3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1 3.4 ...
##  $ A.G.Ratio: num  0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1 ...
##  $ Selector : int  1 1 1 1 1 1 1 1 2 1 ...
```

```
# Distribution of the target variable
table(df$Selector)
```

```
## 
##   1   2 
## 416 167
```

```
X <- table (df$Selector)
barplot(X, main = "Distribution of class variable", xlab = "Selector", ylab = 
"frequency", col = blues9)
```



Observations : a) There are 11 attributes and 583 data points in the data set b) Dependent variable “Selector” has 2 classes and is highly imbalanced c) There is only one categorical independent variables, “Gender” and the rest are numerical variables

3) Cleaning the data

```
# a) Checking for missing values
sum(is.na.data.frame(df))

## [1] 4

# Removing the missing values since there are only 4 of them
df <- na.omit(df)

# Checking for missing values again
sum(is.na.data.frame(df))

## [1] 0
```

Observations : a) Data set had just 4 missing values which has been removed

```
# b) Removing duplicate entries
dim(df)

## [1] 579  11

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.2.3
```

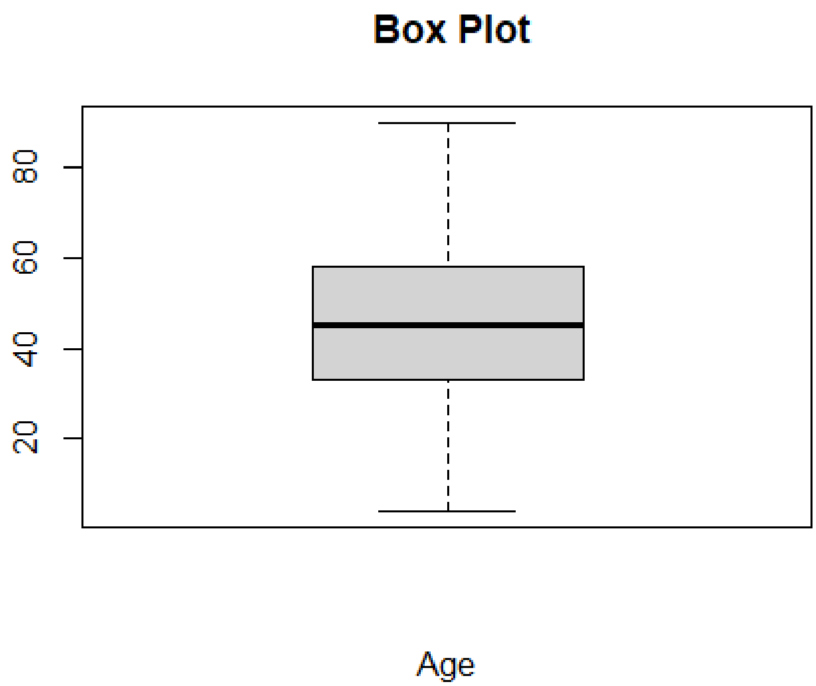
```
##
## Attaching package: 'dplyr'

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

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

df <- distinct(df)
dim(df)

## [1] 566  11
```

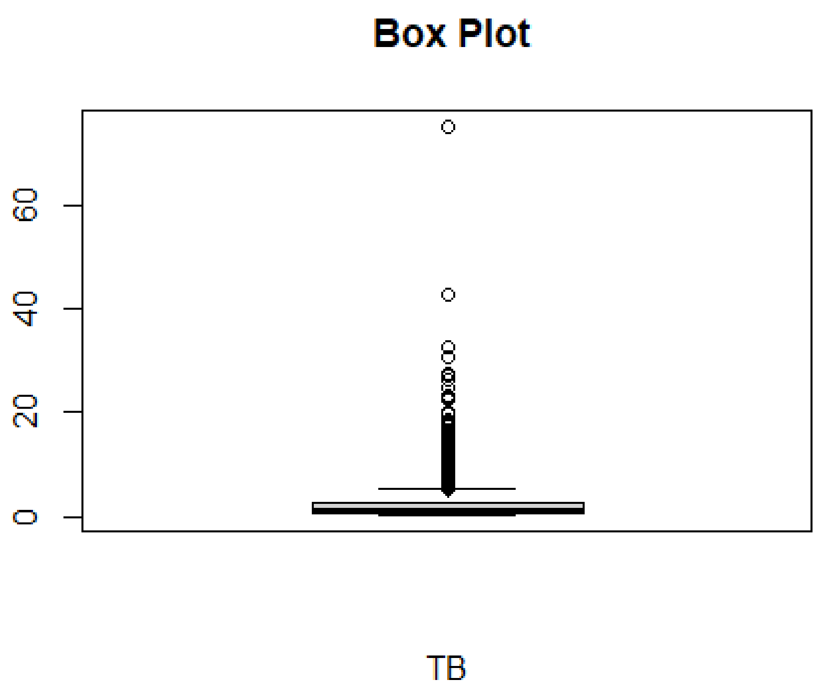
Observations : a) The dimensions of the data has changed after removing the duplicates, from 579 to 566 which means there were 13 duplicate entries in the data.

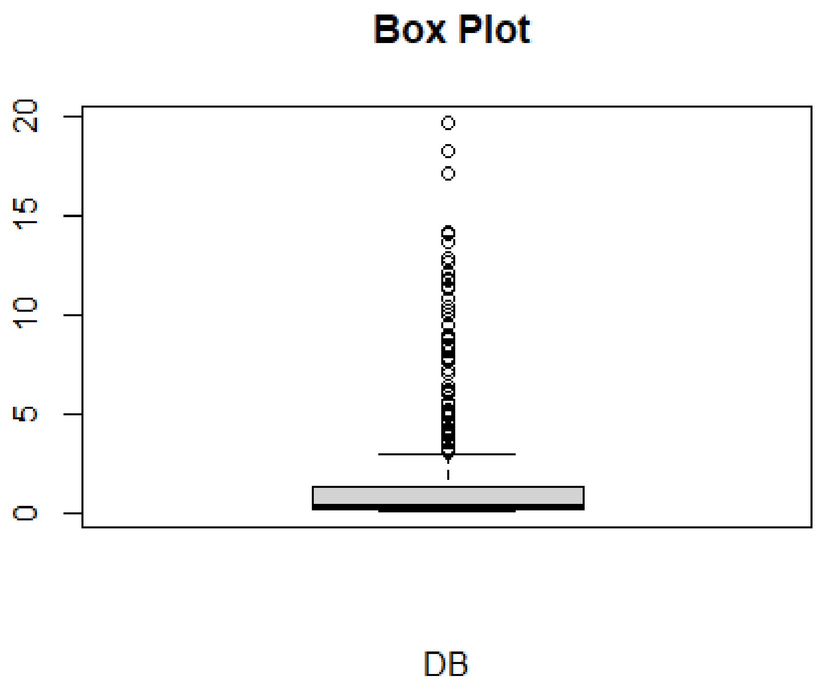
```
# d) Factorizing category variables
str(df)

## 'data.frame':    566 obs. of  11 variables:
##  $ Age      : int  65 62 62 58 72 46 26 29 17 55 ...
##  $ Gender   : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 2 1 1 2 2 ...
##  $ TB       : num  0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0.7 ...
##  $ DB       : num  0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.2 ...
##  $ Alkphos  : int  187 699 490 182 195 208 154 202 202 290 ...
##  $ Sgpt     : int  16 64 60 14 27 19 16 14 22 53 ...
##  $ Sgot     : int  18 100 68 20 59 14 12 11 19 58 ...
##  $ TP       : num  6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8 ...
##  $ ALB      : num  3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1 3.4 ...
##  $ A.G.Ratio: num  0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1 ...
##  $ Selector : int  1 1 1 1 1 1 1 1 2 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:4] 210 242 254 313
##   ..- attr(*, "names")= chr [1:4] "210" "242" "254" "313"

df$Gender <- as.factor(df$Gender)
df$Selector <- as.factor(df$Selector)
str(df)

## 'data.frame':    566 obs. of  11 variables:
##  $ Age      : int  65 62 62 58 72 46 26 29 17 55 ...
##  $ Gender   : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 2 1 1 2 2 ...
##  $ TB       : num  0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0.7 ...
##  $ DB       : num  0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.2 ...
##  $ Alkphos  : int  187 699 490 182 195 208 154 202 202 290 ...
##  $ Sgpt     : int  16 64 60 14 27 19 16 14 22 53 ...
##  $ Sgot     : int  18 100 68 20 59 14 12 11 19 58 ...
##  $ TP       : num  6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8 ...
##  $ ALB      : num  3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1 3.4 ...
##  $ A.G.Ratio: num  0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1 ...
```

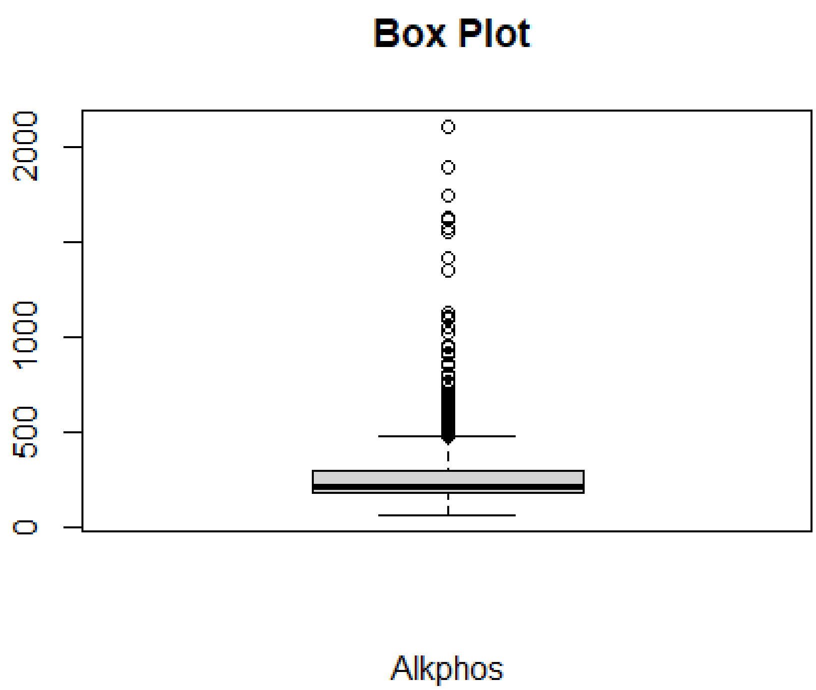
```
##  $ Selector : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 2 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:4] 210 242 254 313
##   ..- attr(*, "names")= chr [1:4] "210" "242" "254" "313"
```

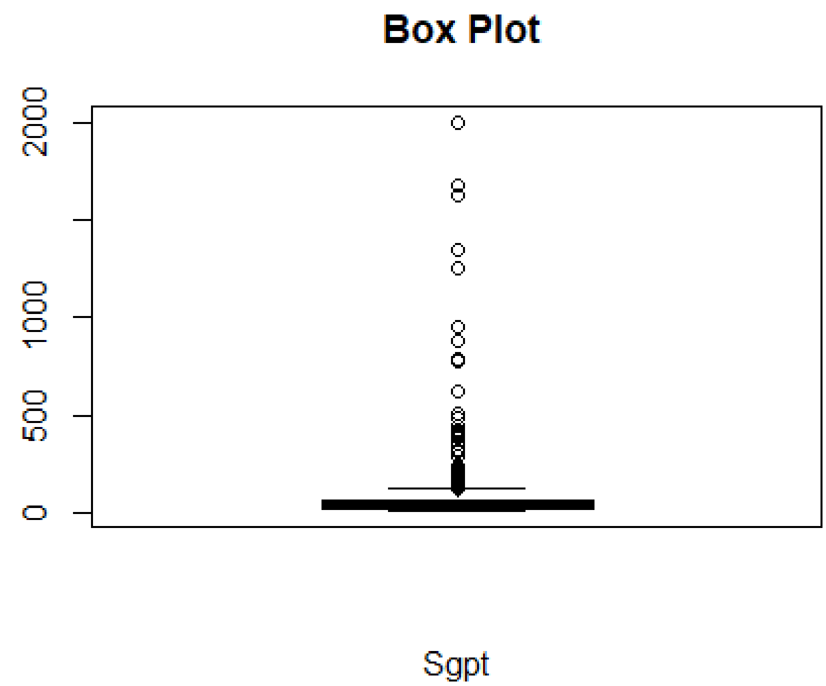
```r
# Checking for outliers using boxplots

N <- c("Age","TB","DB","Alkphos","Sgpt","Sgot","TP","ALB","A.G.Ratio")
for (i in N) {
  boxplot(df[i], main = " Box Plot", xlab = i)
}
```

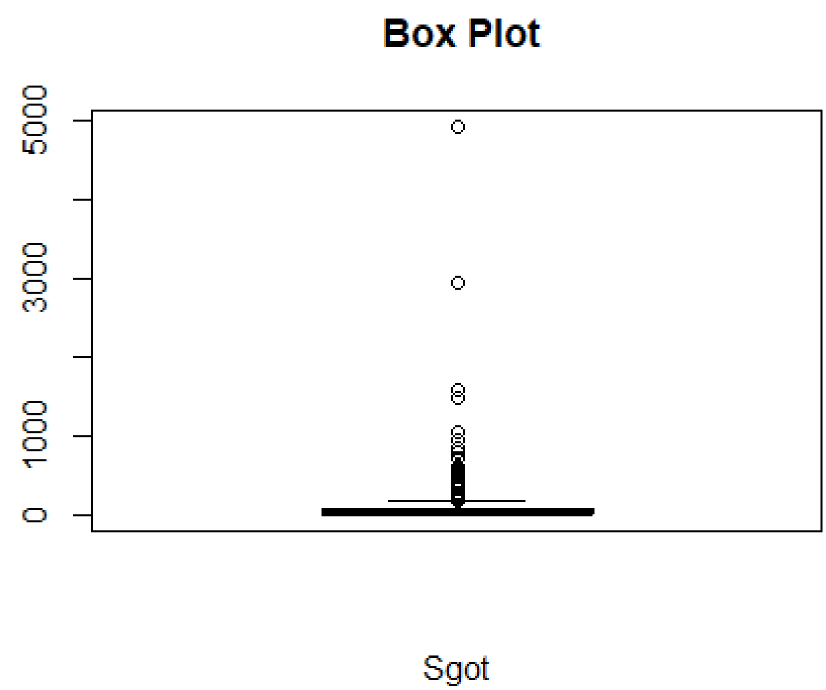

Box Plot
80
60
40
20
Age

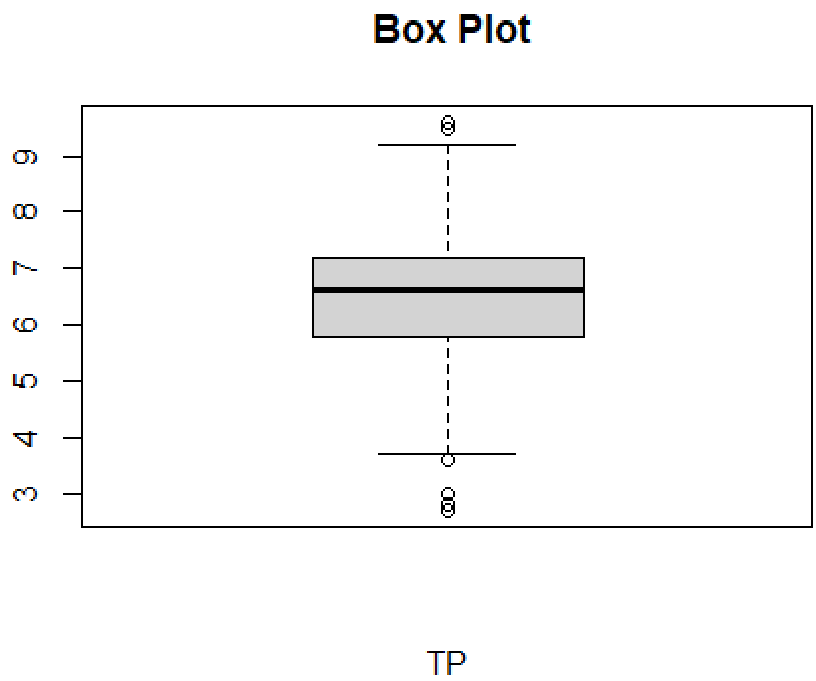

Box Plot
60
40
20
0
TB

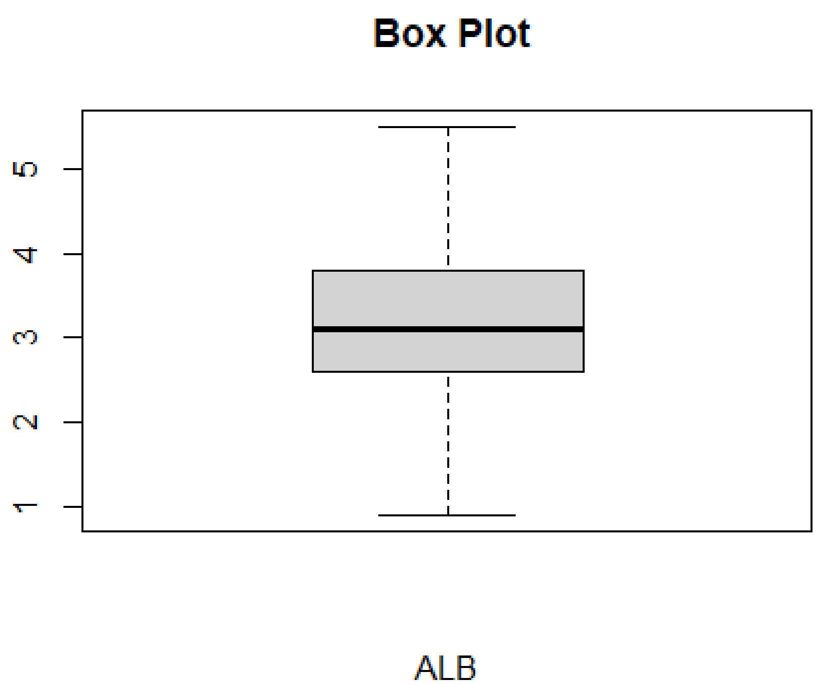

Box Plot
0
5
10
15
20
DB

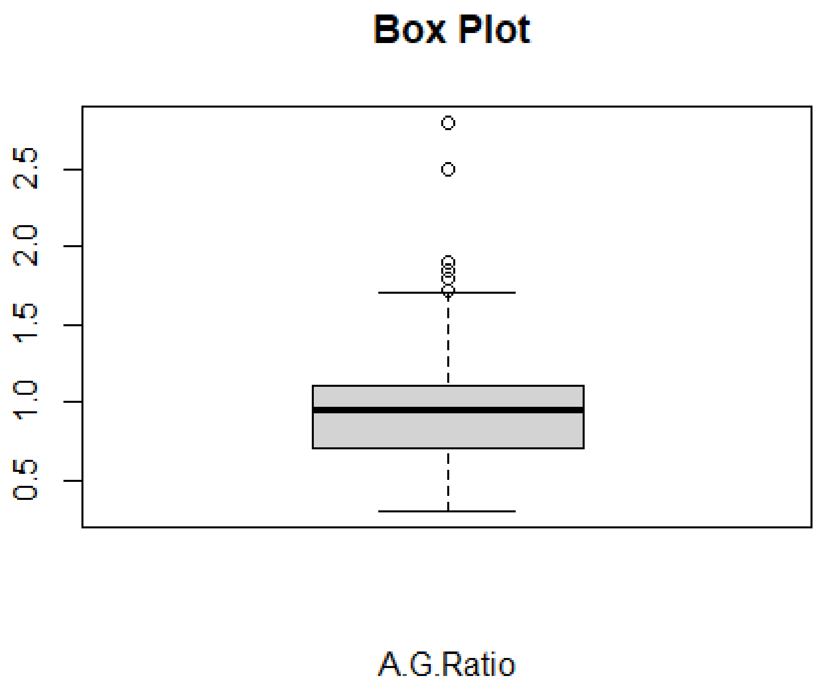

Box Plot
0
500
1000
2000
Alkphos


Box Plot
2000
1000
500
0
Sgpt

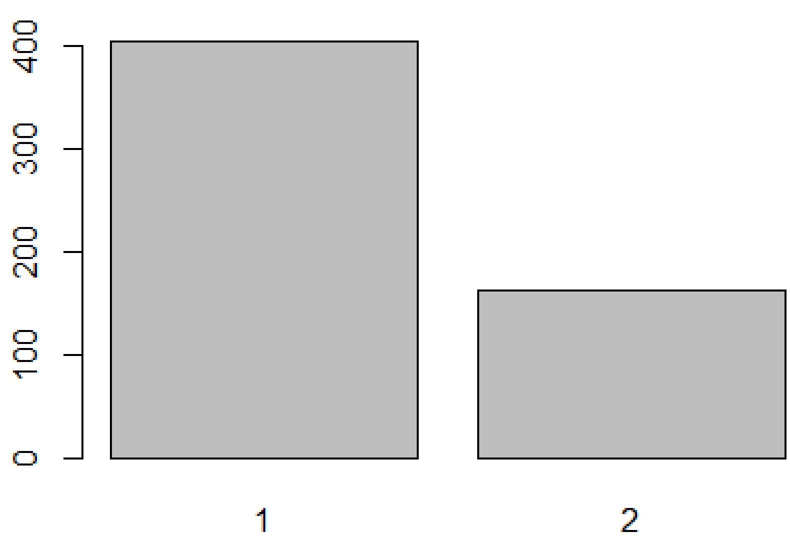

Box Plot
5000
3000
1000
0
Sgot


Box Plot
3
4
5
6
7
8
9
TP


Box Plot
1
2
3
4
5
ALB




```r
# Data balancing with SMOTE_NC

plot(df$Selector)
```



```r
table(df$Selector)
```

```
##
##   1   2
## 404 162
```

```r
library("RSBID")
```

```
## Loading required package: FNN

## Warning: package 'FNN' was built under R version 4.2.3

## Loading required package: clustMixType

## Warning: package 'clustMixType' was built under R version 4.2.3

## Loading required package: klaR

## Warning: package 'klaR' was built under R version 4.2.3

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
set.seed(123)
df <- SMOTE_NC(df,11)
```

```
## Variables are continous and categorical, SMOTE_NC could be used.

##   |
|                                                                      |   0%
|
|                                                                      |   1%
|
|=                                                                     |   1%
|
|=                                                                     |   2%
|
|==                                                                    |   2%
|
|==                                                                    |   3%
|
|===                                                                   |   4%
|
|===                                                                   |   5%
|
|====                                                                  |   6%
|
```

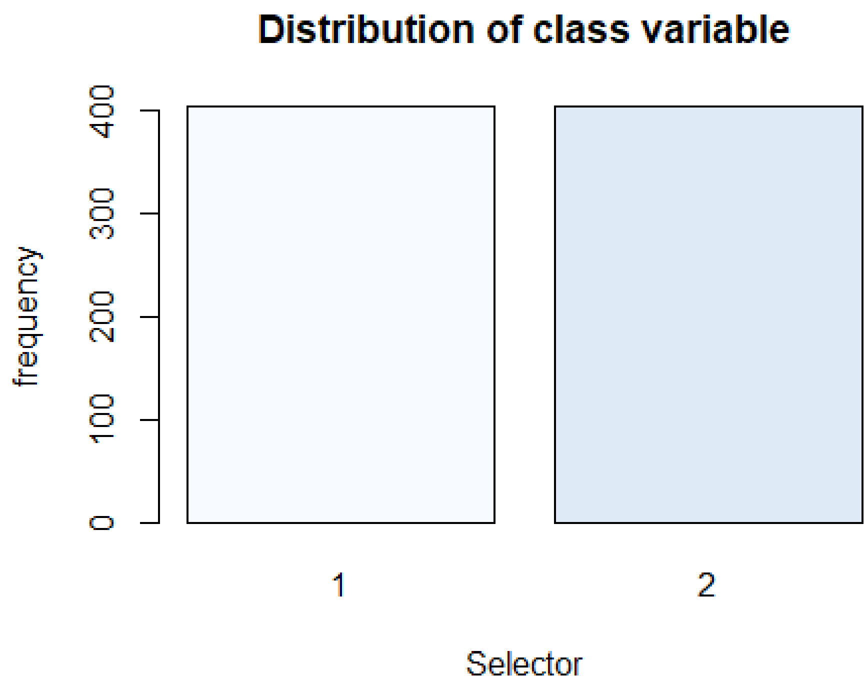
```
|=====                                                                  |   7%
|
|======                                                                 |   8%
|
|======                                                                 |   9%
|
|=======                                                                |   9%
|
|=======                                                                |  10%
|
|=======                                                                |  11%
|
|========                                                               |  11%
|
|========                                                               |  12%
|
|=========                                                              |  12%
|
|=========                                                              |  13%
|
|==========                                                             |  14%
|
|==========                                                             |  15%
|
|===========                                                            |  16%
|
|============                                                           |  17%
|
|=============                                                          |  18%
|
|=============                                                          |  19%
|
|==============                                                         |  20%
|
|===============                                                        |  21%
|
|===============                                                        |  22%
|
|================                                                       |  22%
|
|================                                                       |  23%
|
|=================                                                      |  24%
|
|=================                                                      |  25%
|
|==================                                                     |  25%
|
|==================                                                     |  26%
|
```

```
|===================                                                    |  27%
|
|====================                                                   |  28%
|
|====================                                                   |  29%
|
|=====================                                                  |  30%
|
|======================                                                 |  31%
|
|======================                                                 |  32%
|
|=======================                                                |  32%
|
|=======================                                                |  33%
|
|=======================                                                |  34%
|
|========================                                               |  34%
|
|========================                                               |  35%
|
|=========================                                              |  35%
|
|=========================                                              |  36%
|
|==========================                                             |  37%
|
|===========================                                            |  38%
|
|===========================                                            |  39%
|
|============================                                           |  40%
|
|=============================                                          |  41%
|
|=============================                                          |  42%
|
|==============================                                         |  42%
|
|==============================                                         |  43%
|
|===============================                                        |  44%
|
|===============================                                        |  45%
|
|================================                                       |  45%
|
|================================                                       |  46%
|
```

```
|=================================                                      |  47%
|
|=================================                                      |  48%
|
|==================================                                     |  48%
|
|==================================                                     |  49%
|
|===================================                                    |  50%
|
|====================================                                   |  51%
|
|====================================                                   |  52%
|
|=====================================                                  |  52%
|
|=====================================                                  |  53%
|
|======================================                                 |  54%
|
|======================================                                 |  55%
|
|=======================================                                |  55%
|
|=======================================                                |  56%
|
|========================================                               |  57%
|
|========================================                               |  58%
|
|=========================================                              |  58%
|
|=========================================                              |  59%
|
|==========================================                             |  60%
|
|===========================================                            |  61%
|
|===========================================                            |  62%
|
|============================================                           |  63%
|
|=============================================                          |  64%
|
|=============================================                          |  65%
|
|==============================================                         |  65%
|
|==============================================                         |  66%
|
```

```
|===============================================                       |  66%
|
|===============================================                       |  67%
|
|===============================================                       |  68%
|
|================================================                      |  68%
|
|================================================                      |  69%
|
|=================================================                     |  70%
|
|==================================================                    |  71%
|
|==================================================                    |  72%
|
|===================================================                   |  73%
|
|====================================================                  |  74%
|
|====================================================                  |  75%
|
|=====================================================                 |  75%
|
|=====================================================                 |  76%
|
|======================================================                |  77%
|
|======================================================                |  78%
|
|=======================================================               |  78%
|
|=======================================================               |  79%
|
|========================================================              |  80%
|
|=========================================================             |  81%
|
|=========================================================             |  82%
|
|==========================================================            |  83%
|
|===========================================================           |  84%
|
|============================================================          |  85%
|
|============================================================          |  86%
|
|=============================================================         |  87%
|
```

```
|=============================================================         |  88%
|
|==============================================================        |  88%
|
|==============================================================        |  89%
|
|===============================================================       |  89%
|
|===============================================================       |  90%
|
|===============================================================       |  91%
|
|================================================================      |  91%
|
|================================================================      |  92%
|
|=================================================================     |  93%
|
|==================================================================    |  94%
|
|===================================================================   |  95%
|
|===================================================================   |  96%
|
|====================================================================  |  97%
|
|====================================================================  |  98%
|
|===================================================================== |  98%
|
|===================================================================== |  99%
|
|======================================================================|  99%
|
|======================================================================| 100%

Y <- table (df$Selector)
barplot(Y, main = "Distribution of class variable", xlab = "Selector", ylab =
"frequency", col = blues9)
```




```r
table(df$Selector)
```

```
##
##   1   2
## 404 404
```

```r
# Scaling of the data

summary(df)
```

```
##       Age           Gender          TB               DB
## Min.   : 4.00   Female:167   Min.   : 0.4000   Min.   : 0.1000
## 1st Qu.:32.00   Male  :641   1st Qu.: 0.7696   1st Qu.: 0.2000
## Median :44.70                Median : 0.9000   Median : 0.2807
## Mean   :43.76                Mean   : 2.6562   Mean   : 1.1556
## 3rd Qu.:56.00                3rd Qu.: 1.8000   3rd Qu.: 0.8000
## Max.   :90.00                Max.   :75.0000   Max.   :19.7000
##     Alkphos            Sgpt              Sgot                TP
## Min.   :  63.0   Min.   :  10.00   Min.   :  10.00   Min.   :2.700
## 1st Qu.: 169.9   1st Qu.:  22.00   1st Qu.:  23.26   1st Qu.:5.896
## Median : 198.0   Median :  31.00   Median :  35.59   Median :6.600
## Mean   : 269.2   Mean   :  65.76   Mean   :  88.48   Mean   :6.515
## 3rd Qu.: 282.0   3rd Qu.:  52.00   3rd Qu.:  68.00   3rd Qu.:7.200
## Max.   :2110.0   Max.   :2000.00   Max.   :4929.00   Max.   :9.600
##       ALB          A.G.Ratio       Selector
## Min.   :0.900   Min.   :0.3000   1:404
## 1st Qu.:2.700   1st Qu.:0.8000   2:404
```

```
##  Median :3.200   Median :1.0000
##  Mean   :3.208   Mean   :0.9783
##  3rd Qu.:3.800   3rd Qu.:1.1356
##  Max.   :5.500   Max.   :2.8000
```

```r
names <- c("Age","TB","DB","Alkphos","Sgpt","Sgot","TP","ALB","A.G.Ratio")
for (x in names) {
  df[,x] <- scale(df[,x])

}

summary(df)
```

```
##         Age.V1            Gender           TB.V1               DB.V1
##  Min.   :-2.4514073   Female:167   Min.   :-0.419419   Min.   :-0.431838
##  1st Qu.:-0.7248726   Male  :641   1st Qu.:-0.350707   1st Qu.:-0.390930
##  Median : 0.0583184                Median :-0.326469   Median :-0.357937
##  Mean   : 0.0000000                Mean   : 0.000000   Mean   : 0.000000
##  3rd Qu.: 0.7550143                3rd Qu.:-0.159159   3rd Qu.:-0.145483
##  Max.   : 2.8515207                Max.   :13.448684   Max.   : 7.586100
##      Alkphos.V1            Sgpt.V1             Sgot.V1
##  Min.   :-0.942109   Min.   :-0.361331   Min.   :-0.317862
##  1st Qu.:-0.453906   1st Qu.:-0.283567   1st Qu.:-0.264150
##  Median :-0.325294   Median :-0.225244   Median :-0.214214
##  Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.000000
##  3rd Qu.: 0.058502   3rd Qu.:-0.089157   3rd Qu.:-0.082958
##  Max.   : 8.410631   Max.   :12.534523   Max.   :19.604407
##         TP.V1               ALB.V1             A.G.Ratio.V1     Selector
##  Min.   :-3.710611   Min.   :-3.0507498   Min.   :-2.269295   1:404
##  1st Qu.:-0.602304   1st Qu.:-0.6719122   1st Qu.:-0.596583   2:404
##  Median : 0.082383   Median :-0.0111240   Median : 0.072502
##  Mean   : 0.000000   Mean   : 0.0000000   Mean   : 0.000000
##  3rd Qu.: 0.665920   3rd Qu.: 0.7818218   3rd Qu.: 0.526126
##  Max.   : 3.000070   Max.   : 3.0285017   Max.   : 6.094267
```

```r
# Correlation pot
library(corrplot)
```

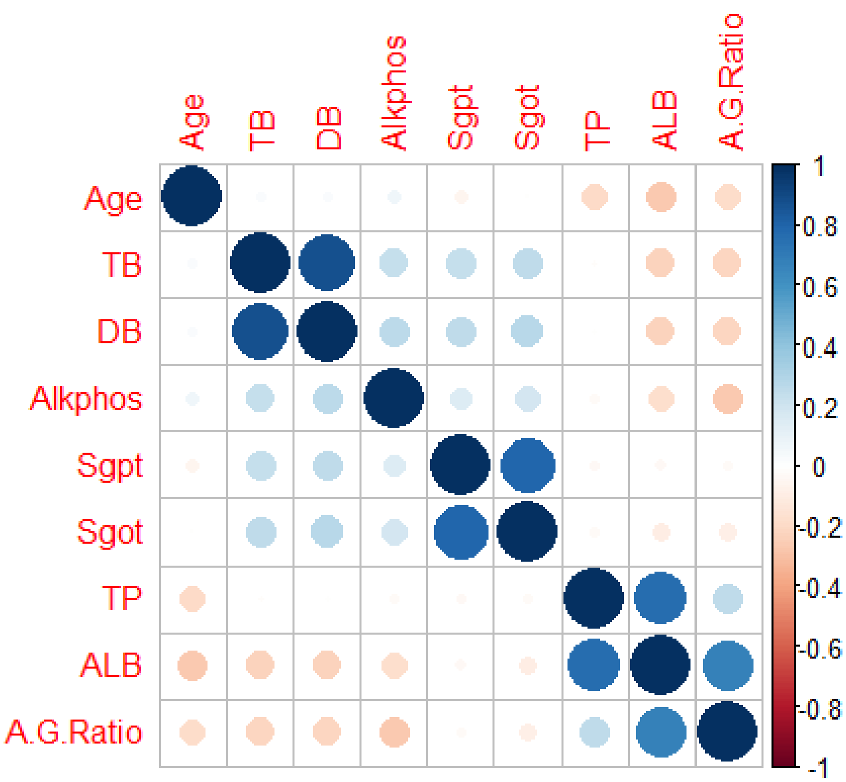
```
## corrplot 0.92 loaded
```

```r
(df.cor <- cor(df[-c(2,11)]))
```

```
##                    Age          TB           DB     Alkphos        Sgpt
## Age        1.000000000  0.02930551  0.027042311  0.06209055 -0.05905788
## TB         0.029305510  1.00000000  0.879148314  0.23384875  0.23831570
## DB         0.027042311  0.87914831  1.000000000  0.26325769  0.25983618
## Alkphos    0.062090550  0.23384875  0.263257690  1.00000000  0.14535007
## Sgpt      -0.059057877  0.23831570  0.259836176  0.14535007  1.00000000
## Sgot      -0.003853151  0.25766611  0.278796031  0.18098235  0.79493806
## TP        -0.192025234 -0.01306675 -0.008628747 -0.02647992 -0.03136306
## ALB       -0.263275249 -0.22210654 -0.228885749 -0.17677225 -0.03847656
```

```
## A.G.Ratio -0.182137561 -0.21417407 -0.213311367 -0.26132969 -0.02549830
##                   Sgot            TP          ALB   A.G.Ratio
## Age        -0.003853151 -0.192025234 -0.26327525 -0.18213756
## TB          0.257666108 -0.013066745 -0.22210654 -0.21417407
## DB          0.278796031 -0.008628747 -0.22888575 -0.21331137
## Alkphos     0.180982352 -0.026479922 -0.17677225 -0.26132969
## Sgpt        0.794938059 -0.031363059 -0.03847656 -0.02549830
## Sgot        1.000000000 -0.023939582 -0.09151649 -0.08220323
## TP         -0.023939582  1.000000000  0.76372122  0.25953102
## ALB        -0.091516487  0.763721220  1.00000000  0.67703006
## A.G.Ratio -0.082203230  0.259531017  0.67703006  1.00000000
```

```r
corrplot(df.cor)
```




```r
# Following pairs seems to have high correlation, so these will be looked at again.
cor(df$TB,df$DB)
```

```
##          [,1]
## [1,] 0.8791483
```

```r
cor(df$Sgpt, df$Sgot)
```

```
##          [,1]
## [1,] 0.7949381
```

```r
cor(df$TP, df$ALB)
```

```
##            [,1]
## [1,] 0.7637212
```

4) Modelling the Data

```
# Removing "TB" from the Data Set
df <- df[-3]

# a) Splitting the data into train(80%) and test(20%)
library(caret)

## Warning: package 'caret' was built under R version 4.2.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.3

## Loading required package: lattice

library(magrittr)

## Warning: package 'magrittr' was built under R version 4.2.3

set.seed(123)
train.index <- df$Selector%>%
  createDataPartition(p = 0.8, list = FALSE)
train <- df[train.index,]
test <- df[-train.index,]
table(train$Selector)

##
##   1   2
## 324 324

table(test$Selector)

##
##  1  2
## 80 80

# b) Classification tree
library(MLmetrics)

## Warning: package 'MLmetrics' was built under R version 4.2.3

##
## Attaching package: 'MLmetrics'

## The following objects are masked from 'package:caret':
##
##     MAE, RMSE
```

```
## The following object is masked from 'package:base':
##
##     Recall
```

```
library(rpart.plot)
```

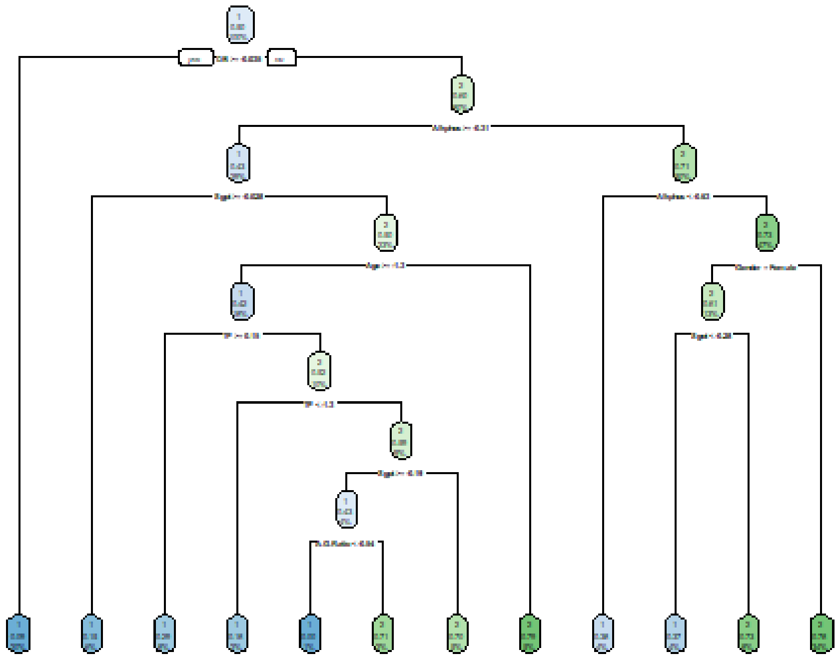
```
## Warning: package 'rpart.plot' was built under R version 4.2.3
```

```
## Loading required package: rpart
```

```
set.seed(123)
CT_Model <- rpart(train$Selector~., data = train)
CT_Model
```

```
## n= 648
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 648 324 1 (0.50000000 0.50000000)
##     2) DB>=-0.03454756 132  12 1 (0.90909091 0.09090909) *
##     3) DB< -0.03454756 516 204 2 (0.39534884 0.60465116)
##       6) Alkphos>=-0.3070586 190  82 1 (0.56842105 0.43157895)
##        12) Sgpt>=-0.02759413 39   6 1 (0.84615385 0.15384615) *
##        13) Sgpt< -0.02759413 151  75 2 (0.49668874 0.50331126)
##          26) Age>=-1.291249 118  50 1 (0.57627119 0.42372881)
##            52) TP>=0.1547283 51  15 1 (0.70588235 0.29411765) *
##            53) TP< 0.1547283 67  32 2 (0.47761194 0.52238806)
##             106) TP< -1.327833 11   2 1 (0.81818182 0.18181818) *
##             107) TP>=-1.327833 56  23 2 (0.41071429 0.58928571)
##               214) Sgpt>=-0.1920452 23  10 1 (0.56521739 0.43478261)
##                 428) A.G.Ratio< -0.5352036 9   0 1 (1.00000000 0.00000000)
*
##                 429) A.G.Ratio>=-0.5352036 14   4 2 (0.28571429
0.71428571) *
##               215) Sgpt< -0.1920452 33  10 2 (0.30303030 0.69696970) *
##          27) Age< -1.291249 33   7 2 (0.21212121 0.78787879) *
##       7) Alkphos< -0.3070586 326  96 2 (0.29447853 0.70552147)
##        14) Alkphos< -0.6321359 24   9 1 (0.62500000 0.37500000) *
##        15) Alkphos>=-0.6321359 302  81 2 (0.26821192 0.73178808)
##          30) Gender=Female 82  32 2 (0.39024390 0.60975610)
##            60) Sgot< -0.2767955 27  10 1 (0.62962963 0.37037037) *
##            61) Sgot>=-0.2767955 55  15 2 (0.27272727 0.72727273) *
##          31) Gender=Male 220  49 2 (0.22272727 0.77727273) *
```

```
rpart.plot(CT_Model)
```



```
CT_pred <- predict(CT_Model, test, type = 'class')
confusionMatrix(CT_pred,test$Selector)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 51 16
##          2 29 64
##
##                Accuracy : 0.7188
##                  95% CI : (0.6423, 0.7869)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 1.489e-08
##
##                   Kappa : 0.4375
##
##  Mcnemar's Test P-Value : 0.07364
##
##             Sensitivity : 0.6375
##             Specificity : 0.8000
##          Pos Pred Value : 0.7612
##          Neg Pred Value : 0.6882
##              Prevalence : 0.5000
##          Detection Rate : 0.3187
##    Detection Prevalence : 0.4188
##       Balanced Accuracy : 0.7188
```

```
##
##        'Positive' Class : 1
##
```

```r
Accuracy(y_pred = CT_pred, y_true = test$Selector)
```

```
## [1] 0.71875
```

```r
Precision(y_pred = CT_pred, y_true = test$Selector, positive = NULL)
```

```
## [1] 0.761194
```

```r
Recall(y_pred = CT_pred, y_true = test$Selector, positive = NULL)
```

```
## [1] 0.6375
```

```r
F1_Score(y_pred = CT_pred, y_true = test$Selector, positive = NULL)
```

```
## [1] 0.6938776
```

```r
# c) SVM Model
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.2.3
```

```r
set.seed(123)
SVM_Model <- svm(train$Selector~., data=train, kernel="linear",
cost=0.10,scale=FALSE)
summary(SVM_Model)
```

```
##
## Call:
## svm(formula = train$Selector ~ ., data = train, kernel = "linear",
##     cost = 0.1, scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.1
##
## Number of Support Vectors:  450
##
##  ( 224 226 )
##
##
## Number of Classes:  2
##
## Levels:
##  1 2
```

```r
SVM_pred <- predict(SVM_Model, test)
confusionMatrix(SVM_pred,test$Selector)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  1  2
##          1 46  6
##          2 34 74
## 
##                Accuracy : 0.75
##                  95% CI : (0.6755, 0.815)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 8.762e-11
## 
##                   Kappa : 0.5
## 
##  Mcnemar's Test P-Value : 1.963e-05
## 
##             Sensitivity : 0.5750
##             Specificity : 0.9250
##          Pos Pred Value : 0.8846
##          Neg Pred Value : 0.6852
##              Prevalence : 0.5000
##          Detection Rate : 0.2875
##    Detection Prevalence : 0.3250
##       Balanced Accuracy : 0.7500
## 
##        'Positive' Class : 1
## 
```

```
Accuracy(y_pred = SVM_pred, y_true = test$Selector)
```

```
## [1] 0.75
```

```
Precision(y_pred = SVM_pred, y_true = test$Selector, positive = NULL)
```

```
## [1] 0.8846154
```

```
Recall(y_pred = SVM_pred, y_true = test$Selector, positive = NULL)
```

```
## [1] 0.575
```

```
F1_Score(y_pred = SVM_pred, y_true = test$Selector, positive = NULL)
```

```
## [1] 0.6969697
```

```
# d) Naive-Bayes Model
library(e1071)
set.seed(123)
NB_Model <- naiveBayes(train$Selector~., train)
summary(NB_Model)
```

```
##           Length Class  Mode
## apriori   2      table  numeric
```

```
## tables    9      -none- list
## levels    2      -none- character
## isnumeric 9      -none- logical
## call      4      -none- call
```

```
NB_pred <- predict(NB_Model, test)
confusionMatrix(NB_pred,test$Selector)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 37  2
##          2 43 78
##
##                Accuracy : 0.7188
##                  95% CI : (0.6423, 0.7869)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 1.489e-08
##
##                   Kappa : 0.4375
##
##  Mcnemar's Test P-Value : 2.479e-09
##
##             Sensitivity : 0.4625
##             Specificity : 0.9750
##          Pos Pred Value : 0.9487
##          Neg Pred Value : 0.6446
##              Prevalence : 0.5000
##          Detection Rate : 0.2313
##    Detection Prevalence : 0.2437
##       Balanced Accuracy : 0.7188
##
##        'Positive' Class : 1
##
```

```
Accuracy(y_pred = NB_pred, y_true = test$Selector)
```

```
## [1] 0.71875
```

```
Precision(y_pred = NB_pred, y_true = test$Selector, positive = NULL)
```

```
## [1] 0.9487179
```

```
Recall(y_pred = NB_pred, y_true = test$Selector, positive = NULL)
```

```
## [1] 0.4625
```

```
F1_Score(y_pred = NB_pred, y_true = test$Selector, positive = NULL)
```

```
## [1] 0.6218487
```