Reading from the data file

> library(readxl)

> DataSetClass <- read\_excel("~/DataSetClass.xlsx")

> View(DataSetClass)

#Checking structure of data set

> str(DataSetClass)

Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 2126 obs. of 22 variables:

$ LB : num 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 : num 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 : num 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 : num 64 130 130 117 117 150 150 68 68 68 ...

$ Min : num 62 68 68 53 53 50 50 62 62 62 ...

$ Max : num 126 198 198 170 170 200 200 130 130 130 ...

$ Nmax : num 2 6 5 11 9 5 6 0 0 1 ...

$ Nzeros : num 0 1 1 0 0 3 3 0 0 0 ...

$ Mode : num 120 141 141 137 137 76 71 122 122 122 ...

$ Mean : num 137 136 135 134 136 107 107 122 122 122 ...

$ Median : num 121 140 138 137 138 107 106 123 123 123 ...

$ Variance: num 73 12 13 13 11 170 215 3 3 1 ...

$ Tendency: num 1 0 0 1 1 0 0 1 1 1 ...

$ NSP : num 2 1 1 1 1 3 3 3 3 3 ...

#Converting NSP from numerical to factor

DataSetClass$NSPF <- factor(DataSetClass$NSP)

> str(DataSetClass)

Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 2126 obs. of 23 variables:

$ LB : num 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 : num 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 : num 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 : num 64 130 130 117 117 150 150 68 68 68 ...

$ Min : num 62 68 68 53 53 50 50 62 62 62 ...

$ Max : num 126 198 198 170 170 200 200 130 130 130 ...

$ Nmax : num 2 6 5 11 9 5 6 0 0 1 ...

$ Nzeros : num 0 1 1 0 0 3 3 0 0 0 ...

$ Mode : num 120 141 141 137 137 76 71 122 122 122 ...

$ Mean : num 137 136 135 134 136 107 107 122 122 122 ...

$ Median : num 121 140 138 137 138 107 106 123 123 123 ...

$ Variance: num 73 12 13 13 11 170 215 3 3 1 ...

$ Tendency: num 1 0 0 1 1 0 0 1 1 1 ...

$ NSP : num 2 1 1 1 1 3 3 3 3 3 ...

$ NSPF : Factor w/ 3 levels "1","2","3": 2 1 1 1 1 3 3 3 3 3 ...

#Sampling the data and dividing into training and validation

> set.seed(1234)

> pd <- sample(2,nrow(DataSetClass),replace = TRUE, prob = c(0.8,0.2))

> trainData <- DataSetClass[pd==1,]

> valData <- DataSetClass[pd==2,]

#We will be using ctree function from party package

> library(party)

> initTree <- ctree(NSPF ~ LB+AC+FM, data = trainData)

> initTree

Conditional inference tree with 10 terminal nodes

Response: NSPF

Inputs: LB, AC, FM

Number of observations: 1718

1) AC <= 0.0008340284; criterion = 1, statistic = 263.403

2) LB <= 136; criterion = 1, statistic = 131.511

3) FM <= 0.111898; criterion = 1, statistic = 35.729

4)\* weights = 405

3) FM > 0.111898

5)\* weights = 11

2) LB > 136

6)\* weights = 314

1) AC > 0.0008340284

7) AC <= 0.002209945; criterion = 1, statistic = 52.155

8) LB <= 136; criterion = 0.999, statistic = 17.292

9) FM <= 0.01213961; criterion = 1, statistic = 42.826

10)\* weights = 103

9) FM > 0.01213961

11)\* weights = 7

8) LB > 136

12)\* weights = 78

7) AC > 0.002209945

13) LB <= 110; criterion = 1, statistic = 18.889

14)\* weights = 18

13) LB > 110

15) LB <= 147; criterion = 0.965, statistic = 8.877

16) FM <= 0.2354892; criterion = 0.986, statistic = 10.725

17)\* weights = 742

16) FM > 0.2354892

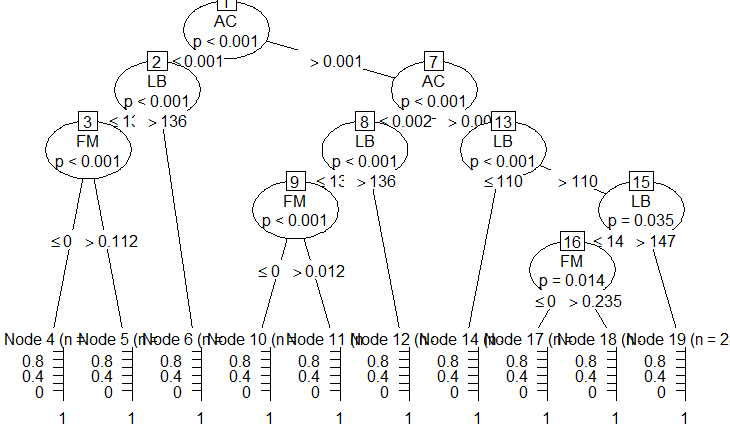
18)\* weights = 12

15) LB > 147

19)\* weights = 28

#Plot the tree

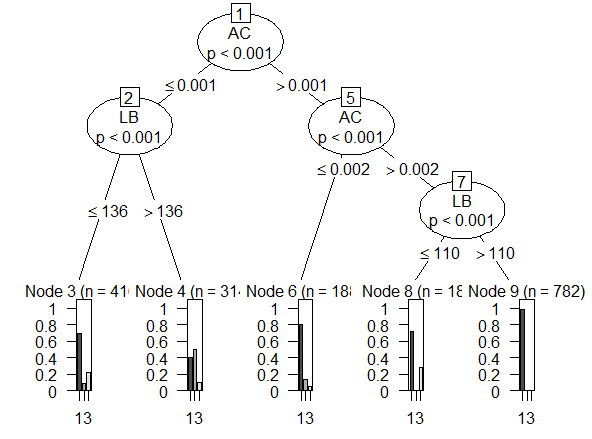
> plot(initTree)



#More pruned model

> initTreeCon <- ctree(NSPF ~ LB+AC+FM, data = trainData, controls = ctree\_control(mincriterion = 0.99, minsplit = 500))

> plot(initTreeCon)



#Validate the dataset

> validate <- predict(initTreeCon, valData)

> mean(validate != valData$NSPF)

[1] 0.2107843

Error in classification is 21% which is average for the model