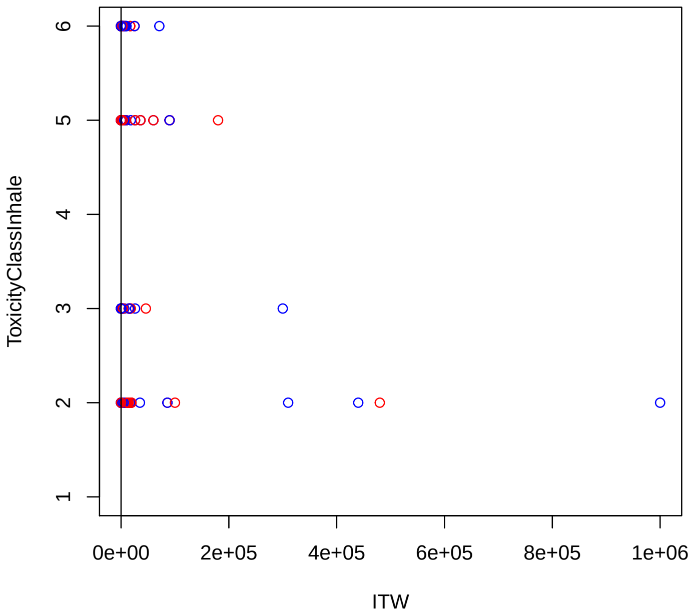
Priyanka Mudiganti

Reading the data and splitting it into training and Validaton data respectively

```
> setwd("C:/Users/bharat/Desktop/JOBS/acedemic projects/Toxic Release Inventory")
> project1 <- read.csv(file="rsei212_chemical.txt", header=T, sep="\t")
> pc=sample(2,nrow(project1),replace=TRUE,prob=c(0.8,0.2))
> train<-project1[pc==1,]
> validate<-project1[pc==2,]
```

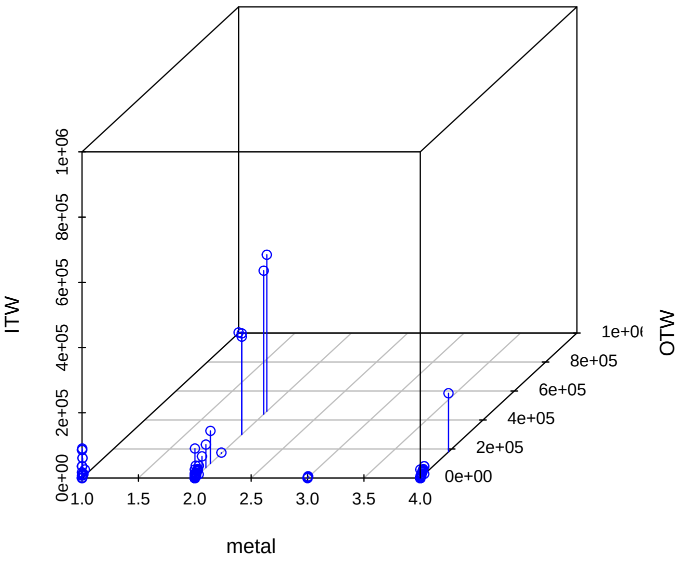
Scatter plot to show the correlation between ITW and ToxicityClassInhale

```
> plot(train$ITW,train$ToxicityClassInhale,xlab="ITW", ylab="ToxicityClassInhale",col=c("rec
> abline(lm(train$ITW ~ train$ToxicityClassInhale))
```



3DScatter plot to show the correlation among Metal,OTW and ITW

```
> library(scatterplot3d)
> scatterplot3d(train$Metal, train$OTW , train$ITW, xlab="metal", ylab="OTW" , zlab="ITW",cc
```

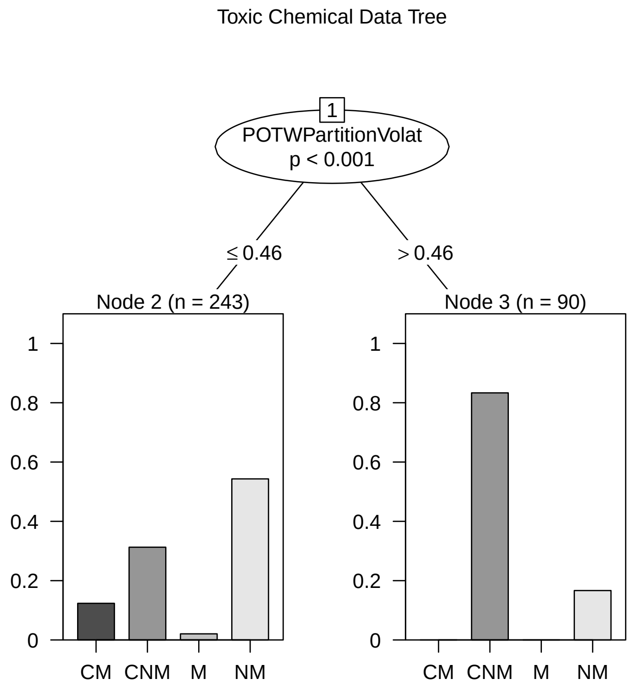


To Plot a Decision Tree by taking Metal as the target variable using Ctree Package in R

```
> train = subset(train, select=c("Metal","ITW","POTWPartitionVolat","BCF"))
> row.names(train) = train$CASNumber
> train = na.omit(train)
> frmla = Metal ~ ITW + POTWPartitionVolat + BCF
> library(party)
> train_output_ctree = ctree(frmla, data = train)
> train_output_ctree
```

```
         Conditional inference tree with 2 terminal nodes

Response:  Metal
Inputs:  ITW, POTWPartitionVolat, BCF
Number of observations:  333

1) POTWPartitionVolat <= 0.46; criterion = 1, statistic = 33.03
  2)*  weights = 243
1) POTWPartitionVolat > 0.46
  3)*  weights = 90
```

```
> plot(train_output_ctree, main="Toxic Chemical Data Tree")
```



prediction for validation data

```
> predict(train_output_ctree,validate,type='prob')

[[1]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[2]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[3]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[4]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[5]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[6]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988
```

```
[[7]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[8]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[9]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[10]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[11]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[12]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[13]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[14]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[15]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[16]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[17]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[18]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[19]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[20]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[21]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[22]]
```

```
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[23]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[24]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[25]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[26]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[27]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[28]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[29]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[30]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[31]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[32]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[33]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[34]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[35]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[36]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[37]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988
```

```
[[38]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[39]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[40]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[41]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[42]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[43]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[44]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[45]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[46]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[47]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[48]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[49]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[50]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[51]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[52]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988
```

```
[[53]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[54]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[55]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[56]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[57]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[58]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[59]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[60]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[61]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[62]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[63]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[64]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[65]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[66]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[67]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[68]]
```

```
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[69]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[70]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[71]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[72]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[73]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[74]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[75]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[76]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[77]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[78]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[79]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[80]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[81]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[82]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[83]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988
```

```
[[84]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[85]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[86]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[87]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[88]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[89]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[90]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[91]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[92]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[93]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[94]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[95]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[96]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[97]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[98]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667
```

```
[[99]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[100]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[101]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[102]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[103]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[104]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[105]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[106]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[107]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[108]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[109]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988

[[110]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[111]]
[1] 0.0000000 0.8333333 0.0000000 0.1666667

[[112]]
[1] 0.12345679 0.31275720 0.02057613 0.54320988
```

miss-classification error for train data

```
> tab<-table(predict(train_output_ctree),train$Metal)
> print(tab)
```

```
       CM CNM   M  NM
  CM    0   0   0   0
  CNM   0  75   0  15
  M     0   0   0   0
  NM   30  76   5 132
```

```
> 1-sum(diag(tab))/sum(tab)
```

```
[1] 0.3783784
```

miss-classification error for validation data

```
> testpred<-predict(train_output_ctree,newdata=validate)
> vtab<-table(testpred,validate$Metal)
> print(vtab)
```

```
testpred CM CNM  M NM
     CM   0   0  0  0
     CNM  0  15  1 17
     M    0   0  0  0
     NM   4  34  0 41
```

```
> 1-sum(diag(vtab))/sum(vtab)
```

```
[1] 0.5
```

To Plot a Decision Tree by taking Metal as the trget variable using rpart(CART) Package in R

```
> library(rpart)
> train_output_rpart <- rpart(frmla,method="class", data=train)
> printcp(train_output_rpart)
```

```
Classification tree:
rpart(formula = frmla, data = train, method = "class")

Variables actually used in tree construction:
[1] BCF               ITW               POTWPartitionVolat

Root node error: 182/333 = 0.54655

n= 333

        CP nsplit rel error  xerror     xstd
1 0.307692      0   1.00000 1.07692 0.049340
2 0.054945      1   0.69231 0.75275 0.049340
3 0.027473      2   0.63736 0.72527 0.049045
```

```
4 0.024725      3   0.60989 0.71978 0.048980
5 0.013736      5   0.56044 0.71429 0.048913
6 0.010989      7   0.53297 0.71429 0.048913
7 0.010000     11   0.48901 0.70879 0.048845

> summary(train_output_rpart)

Call:
rpart(formula = frmla, data = train, method = "class")
  n= 333

          CP nsplit rel error    xerror       xstd
1 0.30769231      0 1.0000000 1.0769231 0.04933951
2 0.05494505      1 0.6923077 0.7527473 0.04933951
3 0.02747253      2 0.6373626 0.7252747 0.04904463
4 0.02472527      3 0.6098901 0.7197802 0.04897989
5 0.01373626      5 0.5604396 0.7142857 0.04891321
6 0.01098901      7 0.5329670 0.7142857 0.04891321
7 0.01000000     11 0.4890110 0.7087912 0.04884459

Variable importance
POTWPartitionVolat                ITW                BCF
                51                 33                 16

Node number 1: 333 observations,    complexity param=0.3076923
  predicted class=CNM  expected loss=0.5465465  P(node) =1
    class counts:    30   151     5   147
   probabilities: 0.090 0.453 0.015 0.441
  left son=2 (90 obs) right son=3 (243 obs)
  Primary splits:
      POTWPartitionVolat < 0.49   to the right, improve=28.138690, (0 missing)
      BCF                < 740    to the left,  improve=14.546120, (0 missing)
      ITW                < 6550   to the right, improve= 8.027188, (0 missing)
  Surrogate splits:
      ITW < 2.15   to the left,  agree=0.748, adj=0.067, (0 split)

Node number 2: 90 observations
  predicted class=CNM  expected loss=0.1666667  P(node) =0.2702703
    class counts:     0    75     0    15
   probabilities: 0.000 0.833 0.000 0.167

Node number 3: 243 observations,    complexity param=0.05494505
  predicted class=NM   expected loss=0.4567901  P(node) =0.7297297
    class counts:    30    76     5   132
   probabilities: 0.123 0.313 0.021 0.543
  left son=6 (44 obs) right son=7 (199 obs)
  Primary splits:
      ITW                 < 6550   to the right, improve=11.433050, (0 missing)
      BCF                 < 225    to the left,  improve= 7.883446, (0 missing)
      POTWPartitionVolat < 0.0105 to the right, improve= 4.445032, (0 missing)

Node number 6: 44 observations,    complexity param=0.02747253
  predicted class=CM   expected loss=0.5681818  P(node) =0.1321321
    class counts:    19    16     0     9
   probabilities: 0.432 0.364 0.000 0.205
  left son=12 (35 obs) right son=13 (9 obs)
  Primary splits:
      POTWPartitionVolat < 0.005  to the left,  improve=3.577633, (0 missing)
      BCF                 < 620    to the left,  improve=2.742424, (0 missing)
      ITW                 < 30500  to the right, improve=1.901881, (0 missing)
  Surrogate splits:
      ITW < 95000  to the left,  agree=0.841, adj=0.222, (0 split)

Node number 7: 199 observations,    complexity param=0.02472527
  predicted class=NM   expected loss=0.3819095  P(node) =0.5975976
    class counts:    11    60     5   123
   probabilities: 0.055 0.302 0.025 0.618
  left son=14 (167 obs) right son=15 (32 obs)
  Primary splits:
      BCF                 < 1400   to the left,  improve=6.793344, (0 missing)
      POTWPartitionVolat < 0.0105 to the right, improve=4.647202, (0 missing)
      ITW                 < 4.75   to the left,  improve=2.509818, (0 missing)
  Surrogate splits:
      ITW < 5500   to the left,  agree=0.844, adj=0.031, (0 split)

Node number 12: 35 observations,    complexity param=0.01098901
  predicted class=CM   expected loss=0.4571429  P(node) =0.1051051
    class counts:    19    12     0     4
   probabilities: 0.543 0.343 0.000 0.114
  left son=24 (12 obs) right son=25 (23 obs)
  Primary splits:
      ITW < 30500  to the right, improve=2.346170, (0 missing)
      BCF < 298    to the left,  improve=1.951893, (0 missing)
  Surrogate splits:
      BCF < 3765   to the right, agree=0.714, adj=0.167, (0 split)

Node number 13: 9 observations
  predicted class=NM   expected loss=0.4444444  P(node) =0.02702703
    class counts:     0     4     0     5
   probabilities: 0.000 0.444 0.000 0.556

Node number 14: 167 observations,    complexity param=0.02472527
```

```
  predicted class=NM   expected loss=0.4431138  P(node) =0.5015015
    class counts:    11    59     4    93
   probabilities: 0.066 0.353 0.024 0.557
  left son=28 (40 obs) right son=29 (127 obs)
  Primary splits:
      POTWPartitionVolat < 0.0105 to the right, improve=5.169713, (0 missing)
      BCF                < 11.5   to the left,  improve=2.204193, (0 missing)
      ITW                < 4.75   to the left,  improve=2.009489, (0 missing)
  Surrogate splits:
      ITW < 3750   to the right, agree=0.772, adj=0.05, (0 split)

Node number 15: 32 observations
  predicted class=NM   expected loss=0.0625  P(node) =0.0960961
    class counts:     0     1     1    30
   probabilities: 0.000 0.031 0.031 0.938

Node number 24: 12 observations
  predicted class=CM   expected loss=0.1666667  P(node) =0.03603604
    class counts:    10     2     0     0
   probabilities: 0.833 0.167 0.000 0.000

Node number 25: 23 observations,    complexity param=0.01098901
  predicted class=CNM  expected loss=0.5652174  P(node) =0.06906907
    class counts:     9    10     0     4
   probabilities: 0.391 0.435 0.000 0.174
  left son=50 (9 obs) right son=51 (14 obs)
  Primary splits:
      ITW < 9150   to the left,  improve=1.720497, (0 missing)
      BCF < 49     to the left,  improve=1.418116, (0 missing)
  Surrogate splits:
      BCF < 14.5   to the left,  agree=0.696, adj=0.222, (0 split)

Node number 28: 40 observations,    complexity param=0.01373626
  predicted class=CNM  expected loss=0.4  P(node) =0.1201201
    class counts:     0    24     1    15
   probabilities: 0.000 0.600 0.025 0.375
  left son=56 (7 obs) right son=57 (33 obs)
  Primary splits:
      ITW                < 4.4    to the left,  improve=2.556061, (0 missing)
      BCF                < 63     to the left,  improve=2.147802, (0 missing)
      POTWPartitionVolat < 0.12   to the left,  improve=1.283333, (0 missing)
  Surrogate splits:
      POTWPartitionVolat < 0.012  to the left,  agree=0.9, adj=0.429, (0 split)

Node number 29: 127 observations,    complexity param=0.01098901
  predicted class=NM   expected loss=0.3858268  P(node) =0.3813814
```

```
    class counts:    11    35     3    78
   probabilities: 0.087 0.276 0.024 0.614 
  left son=58 (55 obs) right son=59 (72 obs)
  Primary splits:
      BCF < 10.5   to the left,  improve=2.1236820, (0 missing)
      ITW < 39     to the right, improve=0.8527293, (0 missing)
  Surrogate splits:
      ITW < 67     to the right, agree=0.606, adj=0.091, (0 split)

Node number 50: 9 observations
  predicted class=CM   expected loss=0.3333333  P(node) =0.02702703
    class counts:     6     3     0     0
   probabilities: 0.667 0.333 0.000 0.000 

Node number 51: 14 observations
  predicted class=CNM  expected loss=0.5  P(node) =0.04204204
    class counts:     3     7     0     4
   probabilities: 0.214 0.500 0.000 0.286 

Node number 56: 7 observations
  predicted class=CNM  expected loss=0  P(node) =0.02102102
    class counts:     0     7     0     0
   probabilities: 0.000 1.000 0.000 0.000 

Node number 57: 33 observations,    complexity param=0.01373626
  predicted class=CNM  expected loss=0.4848485  P(node) =0.0990991
    class counts:     0    17     1    15
   probabilities: 0.000 0.515 0.030 0.455 
  left son=114 (10 obs) right son=115 (23 obs)
  Primary splits:
      ITW                < 950    to the right, improve=3.037418, (0 missing)
      BCF                < 63     to the left,  improve=1.524709, (0 missing)
      POTWPartitionVolat < 0.0195 to the right, improve=1.075258, (0 missing)
  Surrogate splits:
      BCF < 590    to the right, agree=0.727, adj=0.1, (0 split)

Node number 58: 55 observations,    complexity param=0.01098901
  predicted class=NM   expected loss=0.5090909  P(node) =0.1651652
    class counts:     7    19     2    27
   probabilities: 0.127 0.345 0.036 0.491 
  left son=116 (7 obs) right son=117 (48 obs)
  Primary splits:
      BCF < 7.65   to the right, improve=2.075325, (0 missing)
      ITW < 105    to the left,  improve=1.218182, (0 missing)

Node number 59: 72 observations
```

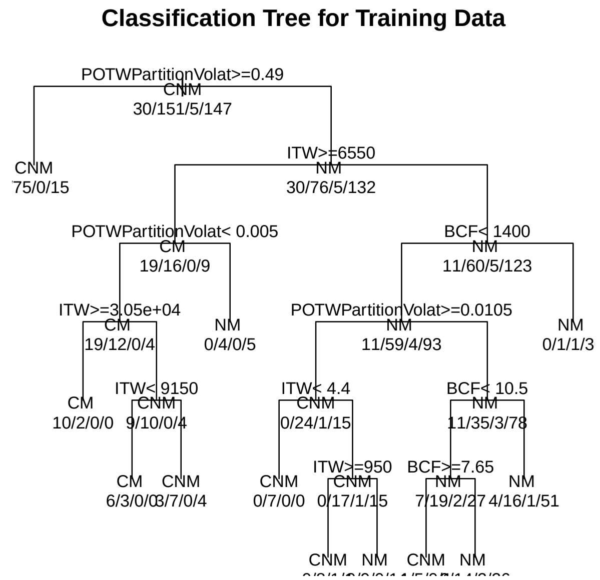
```
  predicted class=NM   expected loss=0.2916667  P(node) =0.2162162
    class counts:     4    16     1    51
   probabilities: 0.056 0.222 0.014 0.708

Node number 114: 10 observations
  predicted class=CNM  expected loss=0.2  P(node) =0.03003003
    class counts:     0     8     1     1
   probabilities: 0.000 0.800 0.100 0.100

Node number 115: 23 observations
  predicted class=NM   expected loss=0.3913043  P(node) =0.06906907
    class counts:     0     9     0    14
   probabilities: 0.000 0.391 0.000 0.609

Node number 116: 7 observations
  predicted class=CNM  expected loss=0.2857143  P(node) =0.02102102
    class counts:     1     5     0     1
   probabilities: 0.143 0.714 0.000 0.143

Node number 117: 48 observations
  predicted class=NM   expected loss=0.4583333  P(node) =0.1441441
    class counts:     6    14     2    26
   probabilities: 0.125 0.292 0.042 0.542
```

visualize cross-validation results

```
> plotcp(train_output_rpart)
```

Plot the actual decision Tree

```
> plot(train_output_rpart, uniform=TRUE, main="Classification Tree for Training Data")
> text(train_output_rpart, use.n=TRUE, all=TRUE, cex=.8)
```



Classification Tree for Training Data

prediction of rpart for validation data

```
> predict(train_output_rpart,validate,type='prob')
```

```
            CM       CNM          M        NM
14  0.21428571 0.5000000 0.00000000 0.2857143
15  0.00000000 0.4444444 0.00000000 0.5555556
17  0.00000000 0.8333333 0.00000000 0.1666667
19  0.05555556 0.2222222 0.01388889 0.7083333
22  0.05555556 0.2222222 0.01388889 0.7083333
27  0.00000000 0.0312500 0.03125000 0.9375000
36  0.00000000 0.8333333 0.00000000 0.1666667
49  0.00000000 0.3913043 0.00000000 0.6086957
50  0.12500000 0.2916667 0.04166667 0.5416667
55  0.05555556 0.2222222 0.01388889 0.7083333
66  0.12500000 0.2916667 0.04166667 0.5416667
68  0.05555556 0.2222222 0.01388889 0.7083333
72  0.00000000 0.8333333 0.00000000 0.1666667
76  0.05555556 0.2222222 0.01388889 0.7083333
84  0.00000000 0.3913043 0.00000000 0.6086957
85  0.05555556 0.2222222 0.01388889 0.7083333
88  0.12500000 0.2916667 0.04166667 0.5416667
```

```
93  0.12500000 0.2916667 0.04166667 0.5416667
95  0.21428571 0.5000000 0.00000000 0.2857143
97  0.05555556 0.2222222 0.01388889 0.7083333
105 0.00000000 0.8333333 0.00000000 0.1666667
106 0.00000000 0.8333333 0.00000000 0.1666667
107 0.83333333 0.1666667 0.00000000 0.0000000
111 0.00000000 0.8333333 0.00000000 0.1666667
123 0.66666667 0.3333333 0.00000000 0.0000000
128 0.00000000 0.8333333 0.00000000 0.1666667
129 0.00000000 0.8333333 0.00000000 0.1666667
132 0.00000000 0.4444444 0.00000000 0.5555556
135 0.00000000 0.0312500 0.03125000 0.9375000
137 0.00000000 0.8333333 0.00000000 0.1666667
148 0.00000000 0.8333333 0.00000000 0.1666667
150 0.05555556 0.2222222 0.01388889 0.7083333
164 0.00000000 0.8333333 0.00000000 0.1666667
165 0.05555556 0.2222222 0.01388889 0.7083333
169 0.00000000 0.3913043 0.00000000 0.6086957
173 0.00000000 0.0312500 0.03125000 0.9375000
175 0.05555556 0.2222222 0.01388889 0.7083333
179 0.00000000 0.0312500 0.03125000 0.9375000
180 0.05555556 0.2222222 0.01388889 0.7083333
182 0.00000000 0.0312500 0.03125000 0.9375000
203 0.00000000 0.8333333 0.00000000 0.1666667
206 0.00000000 0.8333333 0.00000000 0.1666667
221 0.00000000 0.8333333 0.00000000 0.1666667
223 0.00000000 0.8333333 0.00000000 0.1666667
225 0.00000000 0.8333333 0.00000000 0.1666667
229 0.00000000 0.8333333 0.00000000 0.1666667
232 0.00000000 0.8000000 0.10000000 0.1000000
235 0.00000000 0.8333333 0.00000000 0.1666667
237 0.12500000 0.2916667 0.04166667 0.5416667
238 0.05555556 0.2222222 0.01388889 0.7083333
239 0.00000000 0.8000000 0.10000000 0.1000000
240 0.05555556 0.2222222 0.01388889 0.7083333
246 0.12500000 0.2916667 0.04166667 0.5416667
260 0.00000000 0.3913043 0.00000000 0.6086957
265 0.00000000 0.3913043 0.00000000 0.6086957
267 0.05555556 0.2222222 0.01388889 0.7083333
270 0.00000000 0.0312500 0.03125000 0.9375000
273 0.00000000 0.3913043 0.00000000 0.6086957
285 0.00000000 0.8333333 0.00000000 0.1666667
291 0.00000000 0.8333333 0.00000000 0.1666667
297 0.05555556 0.2222222 0.01388889 0.7083333
300 0.00000000 0.0312500 0.03125000 0.9375000
305 0.05555556 0.2222222 0.01388889 0.7083333
```

```
313 0.12500000 0.2916667 0.04166667 0.5416667
321 0.00000000 0.8333333 0.00000000 0.1666667
328 0.83333333 0.1666667 0.00000000 0.0000000
330 0.00000000 0.8333333 0.00000000 0.1666667
331 0.12500000 0.2916667 0.04166667 0.5416667
341 0.05555556 0.2222222 0.01388889 0.7083333
347 0.05555556 0.2222222 0.01388889 0.7083333
361 0.00000000 0.8333333 0.00000000 0.1666667
370 0.00000000 0.8333333 0.00000000 0.1666667
376 0.00000000 0.3913043 0.00000000 0.6086957
377 0.00000000 0.8333333 0.00000000 0.1666667
378 0.05555556 0.2222222 0.01388889 0.7083333
383 0.14285714 0.7142857 0.00000000 0.1428571
386 0.12500000 0.2916667 0.04166667 0.5416667
390 0.00000000 0.3913043 0.00000000 0.6086957
403 0.83333333 0.1666667 0.00000000 0.0000000
405 0.12500000 0.2916667 0.04166667 0.5416667
412 0.00000000 0.8000000 0.10000000 0.1000000
416 0.14285714 0.7142857 0.00000000 0.1428571
420 0.83333333 0.1666667 0.00000000 0.0000000
423 0.00000000 0.4444444 0.00000000 0.5555556
439 0.12500000 0.2916667 0.04166667 0.5416667
447 0.05555556 0.2222222 0.01388889 0.7083333
448 0.00000000 0.8000000 0.10000000 0.1000000
455 0.12500000 0.2916667 0.04166667 0.5416667
467 0.00000000 0.0312500 0.03125000 0.9375000
484 0.05555556 0.2222222 0.01388889 0.7083333
486 0.00000000 0.8333333 0.00000000 0.1666667
499 0.12500000 0.2916667 0.04166667 0.5416667
506 0.05555556 0.2222222 0.01388889 0.7083333
508 0.05555556 0.2222222 0.01388889 0.7083333
514 0.12500000 0.2916667 0.04166667 0.5416667
524 0.00000000 0.8333333 0.00000000 0.1666667
525 0.00000000 0.8333333 0.00000000 0.1666667
526 0.00000000 0.8333333 0.00000000 0.1666667
533 0.66666667 0.3333333 0.00000000 0.0000000
535 0.00000000 0.4444444 0.00000000 0.5555556
541 0.12500000 0.2916667 0.04166667 0.5416667
547 0.00000000 0.4444444 0.00000000 0.5555556
551 0.00000000 0.3913043 0.00000000 0.6086957
555 0.12500000 0.2916667 0.04166667 0.5416667
558 0.00000000 0.8333333 0.00000000 0.1666667
559 0.12500000 0.2916667 0.04166667 0.5416667
560 0.05555556 0.2222222 0.01388889 0.7083333
566 0.00000000 0.0312500 0.03125000 0.9375000
580 0.05555556 0.2222222 0.01388889 0.7083333
```

```
582 0.00000000 0.8333333 0.00000000 0.1666667
588 0.00000000 0.8333333 0.00000000 0.1666667
597 0.00000000 0.3913043 0.00000000 0.6086957
```

miss-classification error for train data using rpart

```
> tab_rpart<-table(predict(train_output_rpart,type="class"),train$Metal)
> print(tab_rpart)
```

```
      CM CNM   M  NM
  CM   16   5   0   0
  CNM   4 102   1  21
  M     0   0   0   0
  NM   10  44   4 126
```

```
> 1-sum(diag(tab_rpart))/sum(tab_rpart)
```

```
[1] 0.2672673
```

miss-classification error for validation data using rpart

```
> testpred_tab_rpart<-predict(train_output_rpart,type="class",newdata=validate)
> vtab_rpart<-table(testpred_tab_rpart,validate$Metal)
> print(vtab_rpart)
```

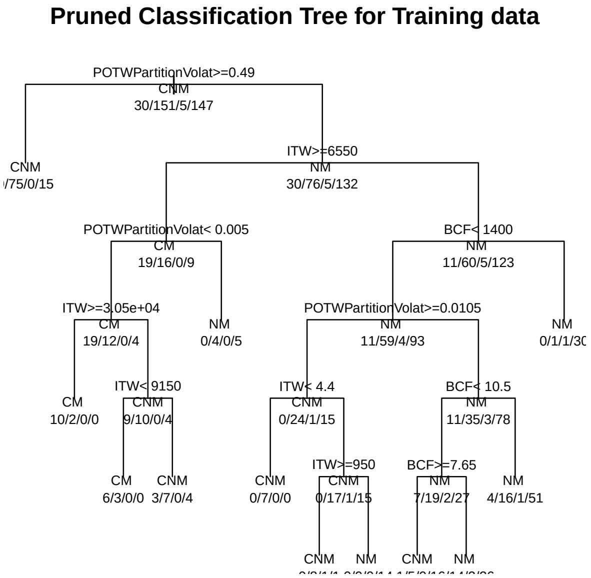
```
testpred_tab_rpart CM CNM  M NM
               CM   3   3  0  0
               CNM  0  21  1 18
               M    0   0  0  0
               NM   1  25  0 40
```

```
> 1-sum(diag(vtab_rpart))/sum(vtab_rpart)
```

```
[1] 0.4285714
```

Prune the tree to avoid overfitting

```
> p_train_output_rpart<- prune(train_output_rpart, cp=train_output_rpart$cptable[which.min(t
> plot(p_train_output_rpart, uniform=TRUE, main="Pruned Classification Tree for Training dat
> text(p_train_output_rpart, use.n=TRUE, all=TRUE, cex=.7)
```


Pruned Classification Tree for Training data

miss-classification error for train data using rpart after pruning

```
> ptab_rpart<-table(predict(p_train_output_rpart,type="class"),train$Metal)
> print(ptab_rpart)
```

```
      CM CNM   M  NM
  CM   16   5   0   0
  CNM   4 102   1  21
  M     0   0   0   0
  NM   10  44   4 126
```

```
> 1-sum(diag(ptab_rpart))/sum(ptab_rpart)
```

```
[1] 0.2672673
```

miss-classification error for validation data using rpart after pruning

```
> testpred_tab_rpart<-predict(p_train_output_rpart,type="class",newdata=validate)
> pv_tab_rpart<-table(testpred_tab_rpart,validate$Metal)
> print(pv_tab_rpart)
```

```
testpred_tab_rpart CM CNM  M NM
               CM   3   3  0  0
               CNM  0  21  1 18
               M    0   0  0  0
               NM   1  25  0 40
```

```
> 1-sum(diag(pv_tab_rpart))/sum(pv_tab_rpart)
```

```
[1] 0.4285714
```