* **Project Code or Command**

1. library(randomForest)
2. library(grid)
3. library(MVT)
4. library(mvtnorm)
5. library(party)
6. library(rpart)
7. library(RColorBrewer)
8. library(basetheme)
9. library(gtools)
10. library(rpart.plot)
11. library(ranger)
12. library(lattice)
13. library(caret)
14. library(RColorBrewer)
15. library(ape)
16. library(treebase)
17. library(ggplot2movies)
18. setwd("F://INTRUSION")
19. dataset<-read.csv("Train\_data.csv",header = T)
20. View(dataset)
21. dim(dataset)
22. str(dataset)
23. shuffle\_index <- sample(1:nrow(dataset))
24. dataset <- dataset[shuffle\_index,]
25. head(dataset)
26. install.packages("rpart.plot")
27. install.packages("rpart")
28. install.packages("multcomp")
29. install.packages("rsq")
30. #### random Forest
31. # split the data into training and testing sets
32. ind<-sample(2,nrow(dataset),replace = T,prob = c(.9,.1))
33. train\_data<-dataset[ind==1,]
34. test\_data<-dataset[ind==2,]
35. dim(train\_data)
36. prop.table(table(train\_data$xAttack))
37. install.packages("rpart")
38. ## random forest for regression ## with all the variable
39. set.seed(1001)
40. random\_dataset<-randomForest(xAttack~.,data=dataset,mtry=3,ntree=250)
41. print(random\_dataset)
42. ## prediction on test data set
43. pred\_dataset<-predict(random\_dataset,test\_data)
44. ##random forest with confirm variable
45. set.seed(1001)
46. install.packages("rpart")
47. fit <-rpart(xAttack~.,data=train\_data,method = "class")
48. plot(xAttack~duration,data=train\_data)
49. plot(train\_data$duration,train\_data$xAttack)
50. box(train\_data$duration)
51. par(las=1)
52. plot(c(1,2),train\_data$xAttack,type="1",xlab = "duration",ylab = "xAttack",lty=2,lwd=3,col="red")
53. # We will use the ctree() function to create the decision tree and see its graph.
54. # Load the party package. It will automatically load other
55. # dependent packages.
56. library(rpart)
57. # Create the input data frame.
58. Train\_data.csv <- xAttack[c(1:105),]
59. # S3 method for BinaryTree
60. plot(xAttack, main = NULL, type = c("extended", "simple"),terminal\_panel = NULL, tp\_args = list(),inner\_panel = node\_inner, ip\_args = list(),edge\_panel = edge\_simple, ep\_args = list(),drop\_terminal = (type[1] == "extended"),tnex = (type[1] == "extended") + 1, newpage = TRUE,pop = TRUE)
61. # display the results
62. printcp(fit)
63. # visualize cross-validation results
64. plotcp(fit)
65. # detailed summary of splits
66. summary(fit)
67. # plot tree
68. plot(fit, uniform=TRUE,main="Regression Tree ")
69. text(fit, use.n=TRUE, all=TRUE, cex=.8)
70. # create attractive postscript plot of tree
71. post(fit, file = "F:/Train\_data.csv",title = "Classification Tree")
72. post(fit, file = "train\_data",title = "Classification Tree")
73. # prune the tree
74. pfit<- prune(fit, cp= fit$cptable[which.min(fit$cptable[,"xAttack"]),"CP"])
75. # grow tree
76. fit <- rpart(xAttack~duration + protocol\_type + count + hot,method="anova", data=train\_data)
77. # display the results
78. printcp(fit)
79. # visualize cross-validation results
80. plotcp(fit)
81. # detailed summary of splits
82. summary(fit)
83. # create additional plots
84. # visualize cross-validation results
85. plotcp(fit)
86. # detailed summary of splits
87. summary(fit)
88. # create additional plots
89. # two plots on one page
90. par(mfrow=c(1,2))
91. # visualize cross-validation results
92. rsq.rpart(fit)
93. # plot tree
94. plot(fit, uniform=TRUE,main="Regression Tree ")
95. text(fit, use.n=TRUE, all=TRUE, cex=.8)
96. # create attractive postcript plot of tree
97. post(fit, file = "F:/Train\_data.csv",title = "Regression Tree ")
98. post(fit, file = "train\_data",title = "Regression Tree ")
99. # prune the tree
100. # from cptable
101. pfit<- prune(fit, cp=0.01160389)
102. # plot the pruned tree
103. plot(pfit, uniform=TRUE,main="Pruned Regression Tree")
104. text(pfit, use.n=TRUE, all=TRUE, cex=.8)
105. post(pfit, file = "F:/Train\_data.csv",title = "Pruned Regression Tree")
106. post(pfit, file = "train\_data",title = "Pruned Regression Tree")
107. # Random Forest prediction of data
108. library(randomForest)
109. fit <- randomForest(xAttack ~ protocol\_type + count + duration, data=train\_data)
110. # view results
111. print(fit)