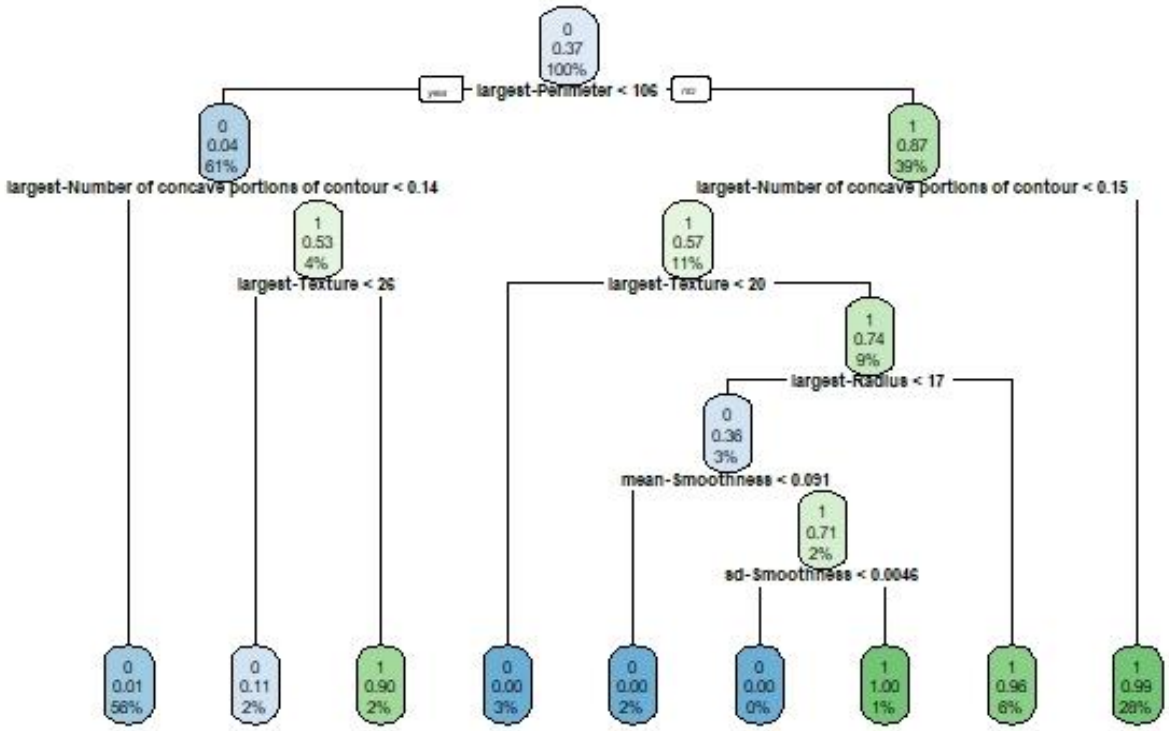
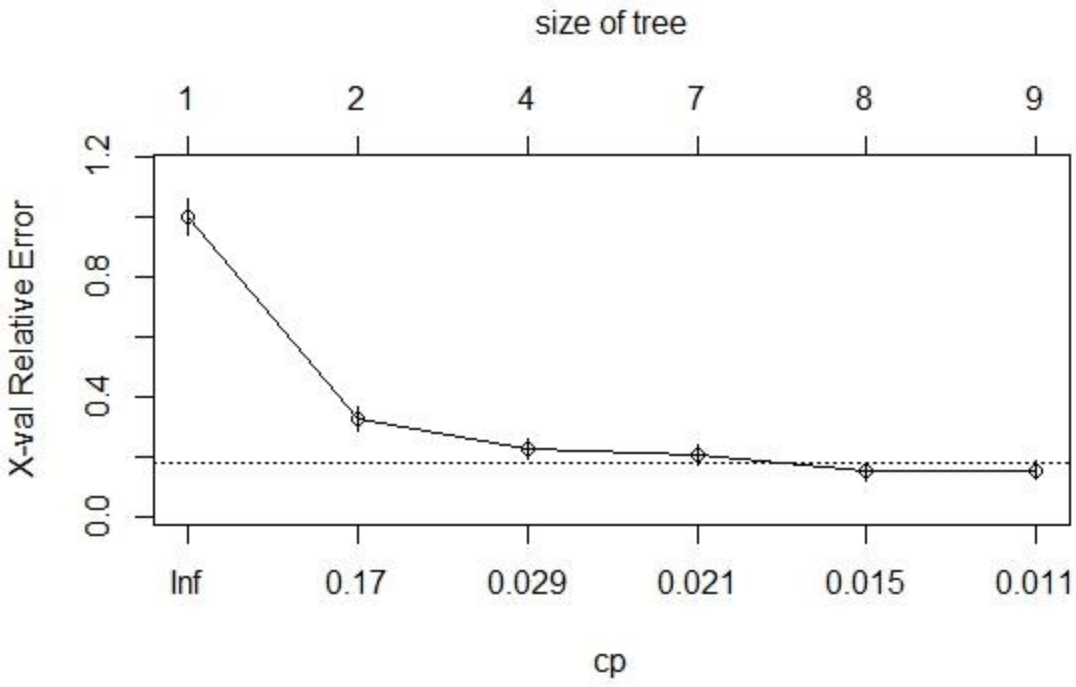
```
##############################################################################################
############################

# CREATED BY: Ankil Mehta

########################

# Load Data File:

########################

trainX <- read.csv(file.choose(), header= TRUE)

trainY <- read.csv(file.choose(), header= TRUE)

testX <- read.csv(file.choose(), header= TRUE)

testY <- read.csv(file.choose(), header= TRUE)



########################

# Data Wrangling and Cleaning

########################

cols<-c("mean-Radius","mean-Texture","mean-Perimeter","mean-Area","mean-Smoothness","mean-
Compactness","mean-Concavity","mean-Number of concave portions of contour","mean-
Symmetry","mean-Fractial dimension","sd-Radius","sd-Texture","sd-Perimeter","sd-Area","sd-
Smoothness","sd-Compactness","sd-Concavity","sd-Number of concave portions of contour","sd-
Symmetry","sd-Fractial dimension","largest-Radius","largest-Texture","largest-Perimeter","largest-
Area","largest-Smoothness","largest-Compactness","largest-Concavity","largest-Number of concave
portions of contour","largest-Symmetry","largest-Fractial dimension")

names(trainX)[1:30]<-cols

names(trainY)<-"Label"

names(testX)[1:30]<-cols

names(testY)<-"Label"


#  Combining trainX and trainY to a single train data

train_data<-cbind(trainX,trainY)


#Label is a target variable and binary
```

```
train_data$Label<-as.factor(train_data$Label)



########################

# Decision Tree model

########################

library(rpart)

library(rpart.plot)


model<-rpart(Label~., data=train_data, parms = list(split = "information"),control=rpart.control(minsplit = 6, cp=0.011))

rpart.plot(model, cex=0.5)
```



```
########################
# Exaluate the model on train data
########################
train_data_pred<-predict(model, train_data, type = "class")
train_data_accuracy<-round(mean(train_data$Label==train_data_pred)*100,2)
train_data_accuracy
#98.68

########################
# Exaluate the model on test data
########################
test_data_pred<-predict(model,testX,type = "class")
test_data_accuracy<-round(mean(testY$Label==test_data_pred)*100,2)
test_data_accuracy
#94.64

########################
# Estimating the cp and minsplit to improve performance of the model
# After pruning, the accuracy of model on train data should reduce but on test data should increase
########################
plotcp(model)
```



model$cptable

```
CP nsplit           rel error    xerror       xstd
1 0.79166667      0 1.00000000 1.0000000 0.06123510
2 0.03571429      1 0.20833333 0.3273810 0.04138382
3 0.02380952      3 0.13690476 0.2261905 0.03512378
4 0.01785714      6 0.06547619 0.2023810 0.03338312
5 0.01190476      7 0.04761905 0.1488095 0.02893087
6 0.01100000      8 0.03571429 0.1547619 0.02946940
```

#Estimating the best cp

model$cptable[which.min(model$cptable[,"xerror"]),"CP"]

#0.01190476

#Estimating the best minsplit

model$cptable[which.min(model$cptable[,"xerror"]),"nsplit"]

7

#Build a tree using cp=0.011 and nsplit=8

```
#Prune the tree and build a new model

model_pruned<-rpart(Label~., data=train_data, parms = list(split = "information"),
control=rpart.control(minsplit = 7, cp=0.011))


#Evaluate on train data

train_data_pred2<-predict(model_pruned, train_data, type = "class")

train_data_accuracy2<-round(mean(train_data$Label==train_data_pred2)*100,2)

train_data_accuracy2

#98.24


#Evaluate on test data

test_data_pred2<-predict(model_pruned,testX,type = "class")

test_data_accuracy2<-round(mean(testY$Label==test_data_pred2)*100,2)

test_data_accuracy2

#94.64


################################################################################################
########################
```