Use decision trees to prepare a model on fraud data

treating those who have taxable\_income <= 30000 as "Risky" and others are "Good"

**Ans :**

**R Code :**

## Decision Trees

########## Fraud Check Data Set #########

fraud <- read.csv('D:\\Data Science\\Excelr\\Assignments\\Assignment\\Decision Trees\\Fraud\_check\_r.csv')

#install.packages("caret")

#install.packages("C50")

library(caret)

library(C50)

inTraininglocal <- createDataPartition(fraud$Status,p=.70,list=F)

training <- fraud[inTraininglocal,]

testing <- fraud[-inTraininglocal,]

#Model Building

model <- C5.0(training$Status~.,data=training)

#Generate the model summary

summary(model)

#Predict the test data set

pred <- predict.C5.0(model,testing[,-7])

a <- table(testing$Status,pred)

sum(diag(a))/sum(a)

plot(model)

#### Boosting

inTraininglocal <- createDataPartition(fraud$Status,p=.70,list=F)

training <- fraud[inTraininglocal,]

testing <- fraud[-inTraininglocal,]

#Model Building

model <- C5.0(training$Status~.,data=training,trials=25) #Trials - Boosting parameters

#Generate the model summary

summary(model)

#Predict the test data set

pred <- predict.C5.0(model,testing[,-7])

a <- table(testing$Status,pred)

sum(diag(a))/sum(a)

plot(model)

##### Bagging

acc <- c()

for (i in 1:100)

{

print(i)

##Data Partition

inTraininglocal <- createDataPartition(fraud$Status,p=.70,list=F)

training1 <- fraud[inTraininglocal,]

testing <- fraud[-inTraininglocal,]

## Model Building

fittree <- C5.0(training1$Status~.,data=training1)

#Predicting

pred <- predict.C5.0(fittree,testing[,-7])

a<-table(testing$Status,pred)

#Accuracy

acc<-c(acc,sum(diag(a))/sum(a))

}

summary(acc)

boxplot(acc)

##### Bagging and Boosting

acc <- c()

for (i in 1:100)

{

print(i)

##Data Partition

inTraininglocal <- createDataPartition(fraud$Status,p=.70,list=F)

training1 <- fraud[inTraininglocal,]

testing <- fraud[-inTraininglocal,]

## Model Building

fittree <- C5.0(training1$Status~.,data=training1,trials=20)

#Predicting

pred <- predict.C5.0(fittree,testing[,-7])

a<-table(testing$Status,pred)

#Accuracy

acc<-c(acc,sum(diag(a))/sum(a))

}

summary(acc)

boxplot(acc)

**Results :**

> model <- C5.0(training$Status~.,data=training)

> #Generate the model summary

> summary(model)

Call:

C5.0.formula(formula = training$Status ~ ., data = training)

C5.0 [Release 2.07 GPL Edition] Wed Apr 29 17:20:03 2020

-------------------------------

Class specified by attribute `outcome'

Read 421 cases (7 attributes) from undefined.data

Decision tree:

Taxable.Income <= 29944: Risky (87)

Taxable.Income > 29944: Good (334)

Evaluation on training data (421 cases):

Decision Tree

----------------

Size Errors

2 0( 0.0%) <<

(a) (b) <-classified as

---- ----

334 (a): class Good

87 (b): class Risky

Attribute usage:

100.00% Taxable.Income

Time: 0.0 secs

> #Predict the test data set

> pred <- predict.C5.0(model,testing[,-7])

> a <- table(testing$Status,pred)

> sum(diag(a))/sum(a)

[1] 1

> model <- C5.0(training$Status~.,data=training,trials=25) #Trials - Boosting parameters

> #Generate the model summary

> summary(model)

Call:

C5.0.formula(formula = training$Status ~ ., data = training, trials = 25)

C5.0 [Release 2.07 GPL Edition] Wed Apr 29 17:20:24 2020

-------------------------------

Class specified by attribute `outcome'

Read 421 cases (7 attributes) from undefined.data

----- Trial 0: -----

Decision tree:

Taxable.Income <= 29732: Risky (87)

Taxable.Income > 29732: Good (334)

\*\*\* boosting reduced to 1 trial since last classifier is very accurate

\*\*\* boosting abandoned (too few classifiers)

Evaluation on training data (421 cases):

Decision Tree

----------------

Size Errors

2 0( 0.0%) <<

(a) (b) <-classified as

---- ----

334 (a): class Good

87 (b): class Risky

Attribute usage:

100.00% Taxable.Income

Time: 0.0 secs

> #Predict the test data set

> pred <- predict.C5.0(model,testing[,-7])

> a <- table(testing$Status,pred)

> sum(diag(a))/sum(a)

[1] 0.9888268

Bagging

> summary(acc)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.9777 0.9944 1.0000 0.9974 1.0000 1.0000

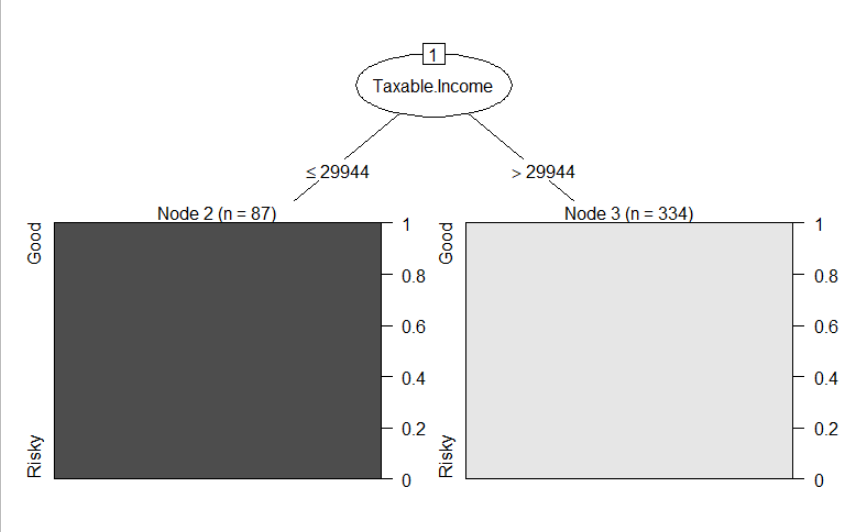
Bagging & Boosting

> summary(acc)

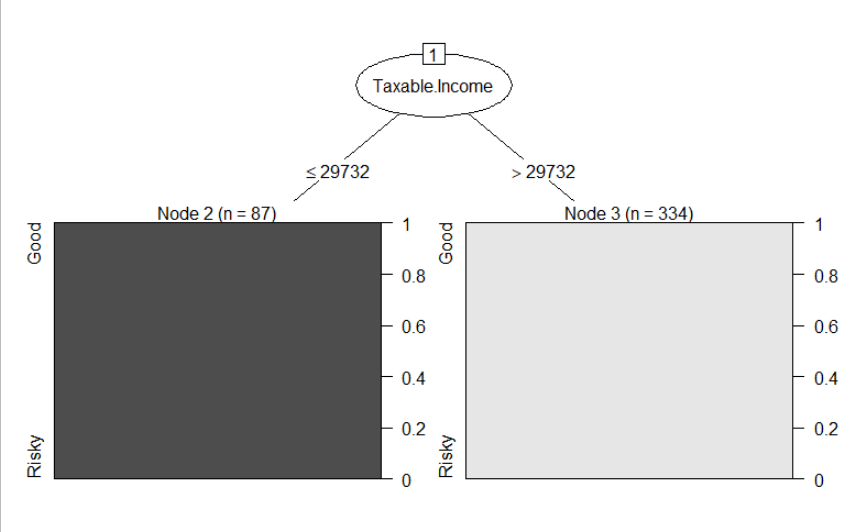
Min. 1st Qu. Median Mean 3rd Qu. Max.

0.9721 0.9944 1.0000 0.9977 1.0000 1.0000

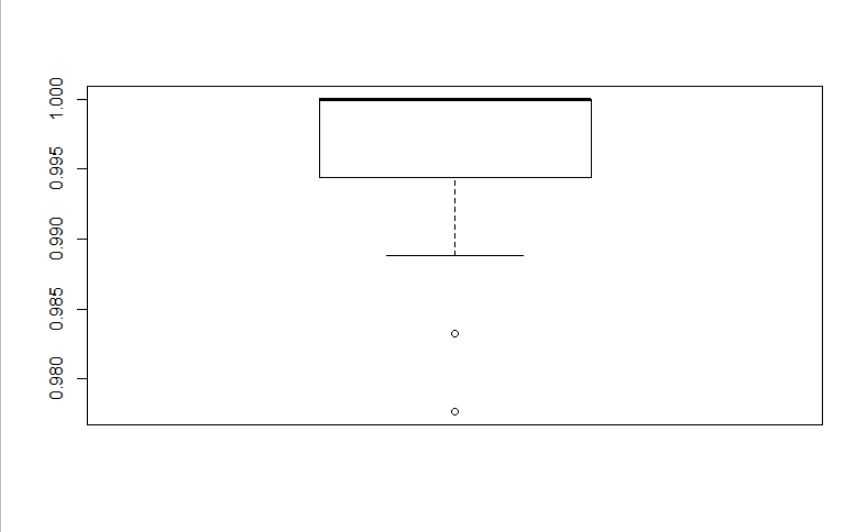
**Plots :**



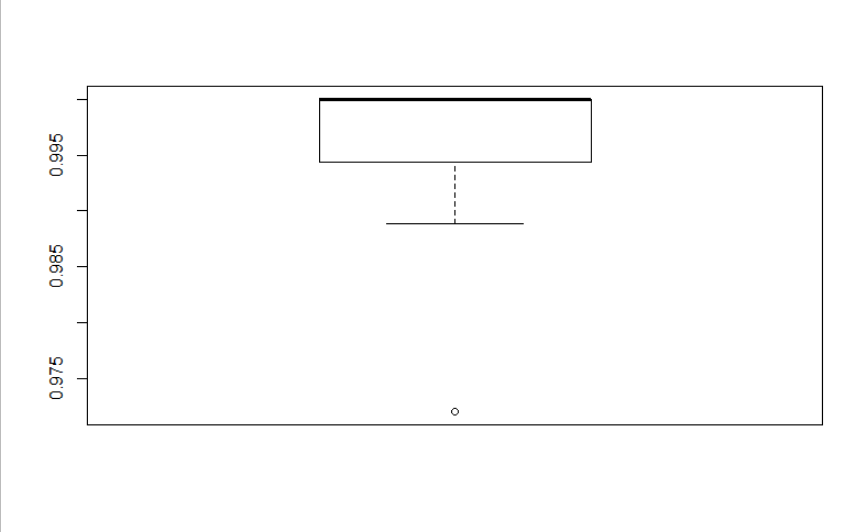
Boosting



Bagging



Bagging & Boosting



**Inference :**

Taxable Income was found as the important feature and was considered as the root node.