TSF-GRIP (Feb`23)

Name : **Bhanuprakash**

Task : **Prediction using Decision Tree Algorithm (Task#6)**

To Do : **Create the Decision Tree classifier and visualize it graphically.**

Tool : **R**

08/02/2023

```
#loading iris data
df=iris
head(df)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa

tail(df)

##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 145          6.7         3.3          5.7         2.5 virginica
## 146          6.7         3.0          5.2         2.3 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 148          6.5         3.0          5.2         2.0 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

Summary

```
summary(df)

##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
##                 

#loading required libraries
library(caret)

## Warning: package 'caret' was built under R version 4.1.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.1.3

## Loading required package: lattice

library(rpart)
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.1.3
```

Training the dataset

```
#Create training and testing datasets
set.seed(500)

training_index=createDataPartition(df$Species,p=0.7,list=F)
trainset=df[training_index,]
testset=df[-training_index,]

print("------ Train Completed ------")

## [1] "------ Train Completed ------"
```

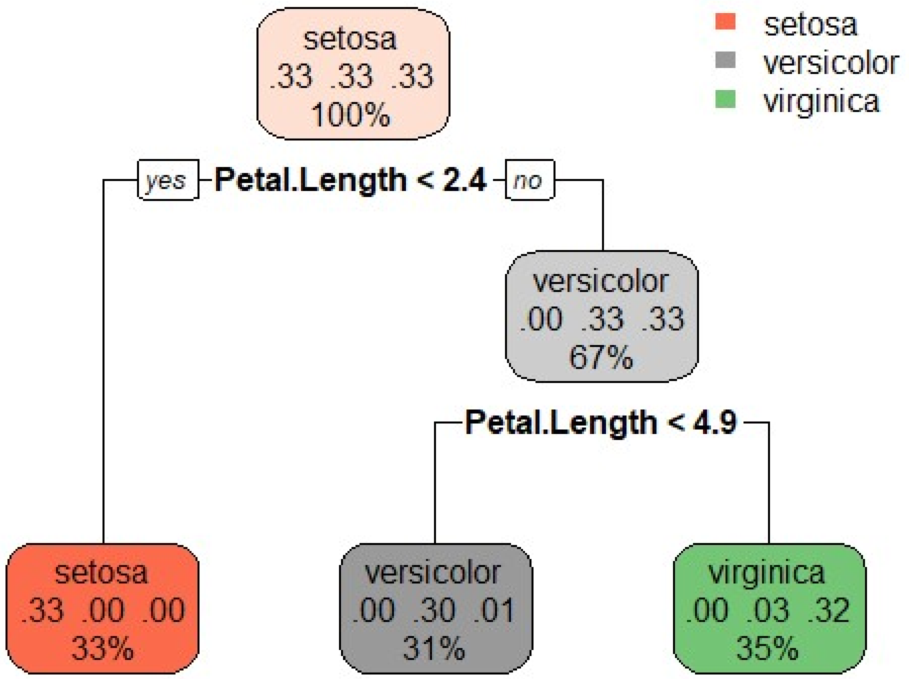
Decision Tree Model computing..

```
#defining decision treee model
deci.tree=rpart(Species~.,data=trainset,parms = list(split="gini"))
deci.tree

## n= 105
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 105 70 setosa (0.33333333 0.33333333 0.33333333)
##   2) Petal.Length< 2.35 35  0 setosa (1.00000000 0.00000000 0.00000000) *
##   3) Petal.Length>=2.35 70 35 versicolor (0.00000000 0.50000000 0.50000000)
##     6) Petal.Length< 4.85 33  1 versicolor (0.00000000 0.96969697 0.03030303) *
##     7) Petal.Length>=4.85 37  3 virginica (0.00000000 0.08108108 0.91891892) *
```

Plotting Model

```
#plotting decision tree model
rpart.plot(deci.tree,extra = 109)
```



Making Predictions

```
# predicton with tree model on train data set.

predict.train <- predict(deci.tree,newdata=trainset,type="class")
addmargins(table(predict.train,trainset$Species))

##
## predict.train setosa versicolor virginica Sum
##    setosa         35          0         0  35
##    versicolor      0         32         1  33
##    virginica       0          3         34  37
##    Sum            35         35         35 105

# predictons with tree model on test data set.

predict.test <- predict(deci.tree,newdata=testset,type="class")
addmargins(table(predict.test,testset$Species))

##
## predict.test setosa versicolor virginica Sum
##   setosa         15          0         0  15
##   versicolor      0         14         2  16
##   virginica       0          1        13  14
##   Sum            15         15        15  45
```

Accuracy of the Model…

```
#confusion matrix for train set
confusionMatrix(predict.train,trainset$Species)

## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         35          0         0
##   versicolor      0         32         1
##   virginica       0          3        34
##
## Overall Statistics
##
##                Accuracy : 0.9619
##                  95% CI : (0.9053, 0.9895)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9429
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            0.9143           0.9714
## Specificity                 1.0000            0.9857           0.9571
## Pos Pred Value              1.0000            0.9697           0.9189
## Neg Pred Value              1.0000            0.9583           0.9853
## Prevalence                  0.3333            0.3333           0.3333
## Detection Rate              0.3333            0.3048           0.3238
## Detection Prevalence        0.3333            0.3143           0.3524
## Balanced Accuracy           1.0000            0.9500           0.9643
```

Interpretation: Accuracy on the trainset data equals 0.9619,which is 96.19% .

```
#confusion matrix for test set
confusionMatrix(predict.test,testset$Species)

## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         15          0         0
##   versicolor      0         14         2
##   virginica       0          1        13
##
## Overall Statistics
##
##                Accuracy : 0.9333
##                  95% CI : (0.8173, 0.986)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
```

```
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            0.9333           0.8667
## Specificity                 1.0000            0.9333           0.9667
## Pos Pred Value              1.0000            0.8750           0.9286
## Neg Pred Value              1.0000            0.9655           0.9355
## Prevalence                  0.3333            0.3333           0.3333
## Detection Rate              0.3333            0.3111           0.2889
## Detection Prevalence        0.3333            0.3556           0.3111
## Balanced Accuracy           1.0000            0.9333           0.9167
```

Interpretation: Accuracy on the testset data equals 0.9333,which is 93.33% .

Finally ,The decision tree model performs well.Even though the accuracy of the testset is lower than the train set accuracy .

——-Thank You——-