Experiment 2.1

1. Aim:

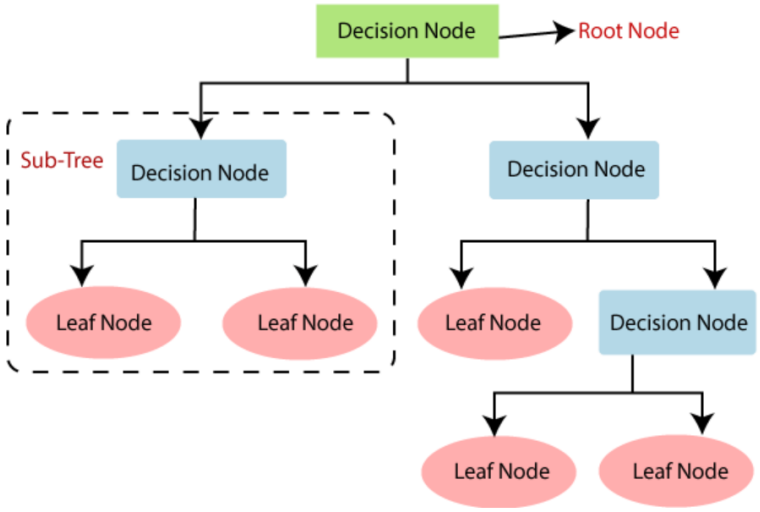
To perform the classification by decision tree induction using WEKA tools.

2. Objective:

Use RWeka Tools to do classification as a part of supervised Machine Learning process through the use of Decision Tree Induction Algorithm.

3. Script and Output:

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal **nodes** represent the **features of a dataset**, **branches** represent the **decision rules** and each **leaf** node represents the **outcome**.



CODE-

1. Installing Packages for the given experiment

```
install.packages("partykit")
install.packages("caTools")
install.packages("rpart")
install.packages("rpart.plot")
library(rpart)
library(rpart.plot)
library(RWeka)
library(partykit)
library(caTools)
```

2. Choosing data set for performing the classification by decision tree induction

3. Iris data get loaded where we do multiple operations such as printing data as string type and showing data summary.

4. Training and Testing of data is done.

```
16
17  data("iris")
18  str(iris)
19  summary(iris)
20
21  #part 1:fit model(recursive partitioning decision tree method)
22  #fit<-rpart(hp~.,data=mtcars)
23
24  spl = sample.split(iris, SplitRatio = 0.7)
25
26  dataTrain = subset(iris, spl==TRUE)
27  dataTest = subset(iris, spl==FALSE)
28  dataTrain
29  dataTest
30
```

Output:

```
> data("iris")
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 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
> spl = sample.split(iris, SplitRatio = 0.7)
>
> dataTrain = subset(iris, spl==TRUE)
> dataTest = subset(iris, spl==FALSE)
> dataTrain
    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
6            5.4         3.9          1.7         0.4     setosa
7            4.6         3.4          1.4         0.3     setosa
8            5.0         3.4          1.5         0.2     setosa
11           5.4         3.7          1.5         0.2     setosa
12           4.8         3.4          1.6         0.2     setosa
13           4.8         3.0          1.4         0.1     setosa
```

```
> dataTest
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
9            4.4         2.9          1.4         0.2     setosa
10           4.9         3.1          1.5         0.1     setosa
14           4.3         3.0          1.1         0.1     setosa
15           5.8         4.0          1.2         0.2     setosa
19           5.7         3.8          1.7         0.3     setosa
20           5.1         3.8          1.5         0.3     setosa
24           5.1         3.3          1.7         0.5     setosa
25           4.8         3.4          1.9         0.2     setosa
29           5.2         3.4          1.4         0.2     setosa
30           4.7         3.2          1.6         0.2     setosa
34           5.5         4.2          1.4         0.2     setosa
```

5.Prediction of data is done along with accuracy testing

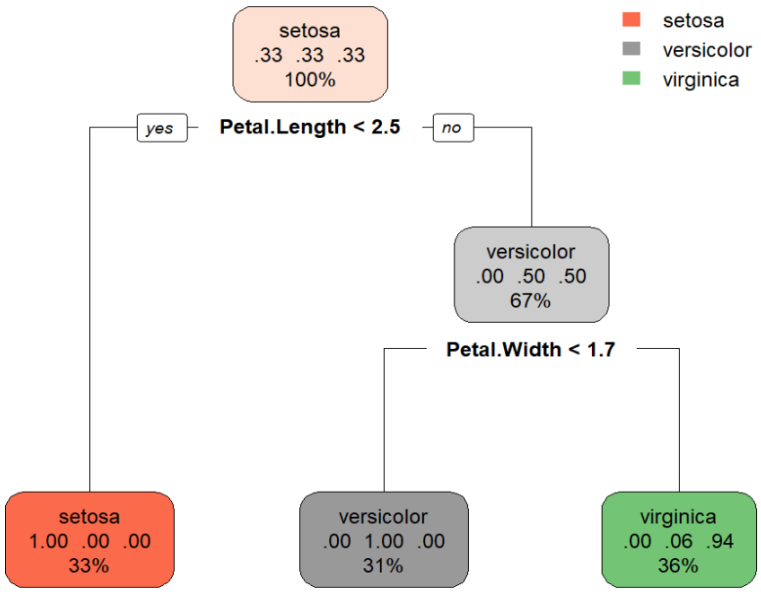
```
m1<- rpart(Species~., dataTrain)

#summarize the fit
summary(m1)
rpart.plot(m1)

# make predictions
#predictions <- predict(fit, mtcars[,1:4], type="class")
predictions1 <- predict(m1, newdata = dataTest, type="class")


# summarize accuracy
table_matrix <- table(dataTest$Species, predictions1)
print(table_matrix)

#table(predictions, iris$Species)
accuracy_Test <- sum(diag(table_matrix)) / sum(table_matrix)
cat("Test Accuracy is: ", accuracy_Test)

```

Output:

```
> m1<- rpart(Species~., dataTrain)
> summary(m1)
Call:
rpart(formula = Species ~ ., data = dataTrain)
  n= 90

         CP nsplit  rel error    xerror       xstd
1 0.5000000      0 1.00000000 1.2333333 0.06045098
2 0.4666667      1 0.50000000 0.9833333 0.07513363
3 0.0100000      2 0.03333333 0.0500000 0.02838231

Variable importance
 Petal.Width Petal.Length Sepal.Length  Sepal.Width
          34           32           21           14

Node number 1: 90 observations,    complexity param=0.5
  predicted class=setosa      expected loss=0.6666667  P(node) =1
    class counts:    30    30    30
   probabilities: 0.333 0.333 0.333
  left son=2 (30 obs) right son=3 (60 obs)
  Primary splits:
      Petal.Length < 2.5  to the left,  improve=30.00000, (0 missing)
      Petal.Width  < 0.7  to the left,  improve=30.00000, (0 missing)
      Sepal.Length < 5.45 to the left,  improve=23.06178, (0 missing)
      Sepal.Width  < 3.35 to the right, improve=12.38636, (0 missing)
  Surrogate splits:
      Petal.Width  < 0.7  to the left,  agree=1.000, adj=1.000, (0 split)
      Sepal.Length < 5.45 to the left,  agree=0.944, adj=0.833, (0 split)
      Sepal.Width  < 3.35 to the right, agree=0.844, adj=0.533, (0 split)

Node number 2: 30 observations
```

```
> rpart.plot(m1)
> # make predictions
> #predictions <- predict(fit, mtcars[,1:4], type="class")
> predictions1 <- predict(m1, newdata = dataTest, type="class")
> # summarize accuracy
> table_matrix <- table(dataTest$Species, predictions1)
> print(table_matrix)
            predictions1
             setosa versicolor virginica
  setosa         20          0         0
  versicolor      0         20         0
  virginica       0          4        16
> #table(predictions, iris$Species)
> accuracy_Test <- sum(diag(table_matrix)) / sum(table_matrix)
> cat("Test Accuracy is: ", accuracy_Test)
Test Accuracy is:  0.9333333
> #table(predictions, iris$Species)
> accuracy_Test <- sum(diag(table_matrix)) / sum(table_matrix)
> cat("Test Accuracy is: ", accuracy_Test)
Test Accuracy is:  0.9333333
```

Plot:



6. Table Matrix is prepared using j48 algorithm

```
#Part 2: fit model using j48 package of R
#The C4.5 algorithm is an extension of the ID3(Iterative Dichotomiser 3) algorith
fit2 <- J48(Species~., dataTrain)

# summarize the fit
summary(fit2)

# make predictions
predictionsj48 <- predict(fit2, newdata = dataTest, type="class")

# summarize accuracy
table_matrixj48 <- table(dataTest$Species, predictionsj48)
print(table_matrixj48)

#table(predictions, iris$Species)
accuracy_Testj48 <- sum(diag(table_matrixj48)) / sum(table_matrixj48)
cat("Test Accuracy is: ", accuracy_Testj48)

```

Output:

```
=== Summary ===

Correctly Classified Instances          88               97.7778 %
Incorrectly Classified Instances         2                2.2222 %
Kappa statistic                          0.9667
Mean absolute error                      0.0278
Root mean squared error                  0.1179
Relative absolute error                  6.25   %
Root relative squared error             25      %
Total Number of Instances               90

=== Confusion Matrix ===

  a  b  c   <-- classified as
 30  0  0 |  a = setosa
  0 28  2 |  b = versicolor
  0  0 30 |  c = virginica
> # make predictions
> predictionsj48 <- predict(fit2, newdata = dataTest, type="class")
> # summarize accuracy
> table_matrixj48 <- table(dataTest$Species, predictionsj48)
> print(table_matrixj48)
            predictionsj48
             setosa versicolor virginica
  setosa         18          2         0
  versicolor      0         20         0
  virginica       0          4        16
> #table(predictions, iris$Species)
> accuracy_Testj48 <- sum(diag(table_matrixj48)) / sum(table_matrixj48)
> cat("Test Accuracy is: ", accuracy_Testj48)
Test Accuracy is:  0.9
```

7. Table Matrix and accuracy testing performed for fit3.

```
68  #Part 3: fit model usig PART weka
69  #PART is rule system that creates pruned c4.5 decision tree for data sets
70  fit3 <- PART(Species~., dataTrain)
71
72  # summarize the fit
73  summary(fit3)
74
75  # make predictions
76  predictionsPART <- predict(fit3, newdata = dataTest, type="class")
77
78  # summarize accuracy
79  table_matrixPART <- table(dataTest$Species, predictionsPART)
80  print(table_matrixPART)
81
82  #table(predictions, iris$Species)
83  accuracy_TestPART <- sum(diag(table_matrixPART)) / sum(table_matrixPART)
84  cat("Test Accuracy is: ", accuracy_TestPART)
85
```

Output:

```
=== Summary ===

Correctly Classified Instances          88               97.7778 %
Incorrectly Classified Instances         2                2.2222 %
Kappa statistic                          0.9667
Mean absolute error                      0.0278
Root mean squared error                  0.1179
Relative absolute error                  6.25   %
Root relative squared error             25      %
Total Number of Instances               90

=== Confusion Matrix ===

  a  b  c   <-- classified as
 30  0  0 |  a = setosa
  0 28  2 |  b = versicolor
  0  0 30 |  c = virginica
> # make predictions
> predictionsPART <- predict(fit3, newdata = dataTest, type="class")
> # summarize accuracy
> table_matrixPART <- table(dataTest$Species, predictionsPART)
> print(table_matrixPART)
            predictionsPART
             setosa versicolor virginica
  setosa         18          2         0
  versicolor      0         20         0
  virginica       0          4        16
> #table(predictions, iris$Species)
> accuracy_TestPART <- sum(diag(table_matrixPART)) / sum(table_matrixPART)
> cat("Test Accuracy is: ", accuracy_TestPART)
Test Accuracy is:  0.9
>
```