LAB-9: SUPERVISED LEARNING : CLASSIFICATION

Aim: Implementation and analysis of Classification algorithms like:

A. ID3, C4.5 using iris.csv dataset
- Visualize the output
- predict the test data
- Verify the result

B. Implementation and analysis of Classification algorithms like:Naive Bayesian, K-Nearest Neighbor using iris.csv dataset
- Visualize the output
- Predict the test data
- Verify the result

Objective: To implement, analyze and demonstrate the ID3, C4.5, Naive Bayesian and K-Nearest Neighbor classification algorithms.

Operations:

A. ID3 and C4.5

1. # ID3 decision tree classification:

Output:

```
> install.packages("RWeka")
```

```
* installing *source* package 'RWeka' ...
** package 'RWeka' successfully unpacked and MD5 sums checked
** using staged installation

   ************************************************
   WARNING: this package has a configure script
         It probably needs manual configuration
   ************************************************


** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** installing vignettes
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (RWeka)
```

```
> install.packages("party")
```

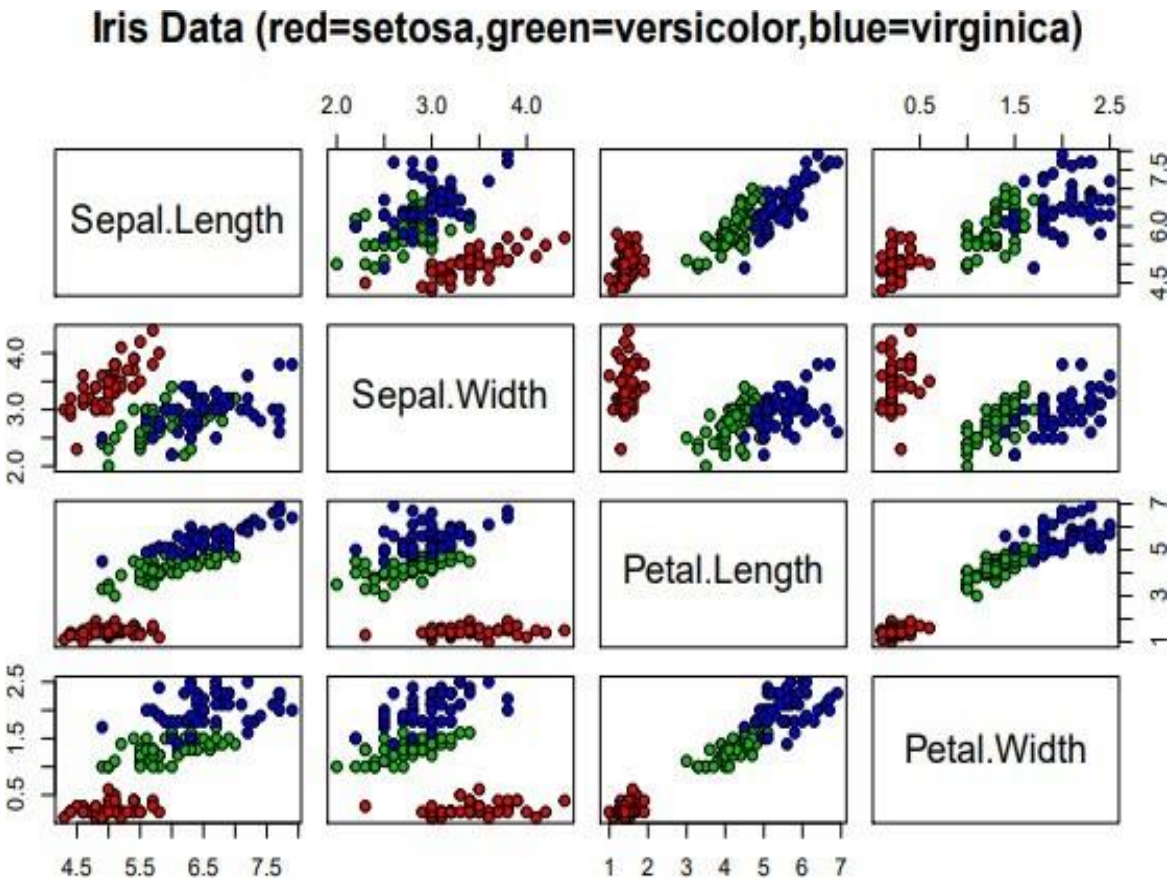
```
package 'TH.data' successfully unpacked and MD5 sums checked
package 'libcoin' successfully unpacked and MD5 sums checked
package 'matrixStats' successfully unpacked and MD5 sums checked
package 'multcomp' successfully unpacked and MD5 sums checked
package 'mvtnorm' successfully unpacked and MD5 sums checked
package 'modeltools' successfully unpacked and MD5 sums checked
package 'strucchange' successfully unpacked and MD5 sums checked
package 'coin' successfully unpacked and MD5 sums checked
package 'party' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Muskan\AppData\Local\Temp\Rtmpiop3N1\downloaded_packages
> install.packages("caret")
Installing package into 'C:/Users/Muskan/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/caret_6.0-93.zip'
Content type 'application/zip' length 3577965 bytes (3.4 MB)
downloaded 3.4 MB

package 'caret' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Muskan\AppData\Local\Temp\RtmpUVirWG\downloaded_packages
> library(RWeka);
> library(party);
Loading required package: grid
Loading required package: mvtnorm
Loading required package: modeltools
Loading required package: stats4
Loading required package: strucchange
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: sandwich
> library(caret);
Loading required package: ggplot2
Use suppressPackageStartupMessages() to eliminate package startup messages
Loading required package: lattice
> my_data <- read.csv("C://Users/Muskan/OneDrive/Documents/iris.csv");
```

```
> my_data
    sepal.length sepal.width petal.length petal.width    variety
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
7            4.6         3.4          1.4         0.3     Setosa
8            5.0         3.4          1.5         0.2     Setosa
9            4.4         2.9          1.4         0.2     Setosa
10           4.9         3.1          1.5         0.1     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
```

- **#Plot graph for Iris Data:**

  Output:

```
> pairs(iris[1:4], main="Iris Data (red=setosa,green=versicolor,blue=virginica)", pch=21, bg=c("red","green3","blue")
[unclass(my_data$variety)]);
```



- **#Use of unclass() function:**

  Output:

```
> cores = c('blue','green','red');

> val = c('setosa','setosa','virginica','versicolor','virginica','setosa');
> val_fac = factor(val);
> unclass(val_fac);
[1] 1 1 3 2 3 1
attr(,"levels")
[1] "setosa"     "versicolor" "virginica"
> cores[unclass(val_fac)];
[1] "blue"  "blue"  "red"   "green" "red"   "blue"
```

```
> bg=c("red","green3","blue")[unclass(iris$Species)];
> bg;
  [1] "red"    "red"    "red"    "red"    "red"    "red"    "red"    "red"
  [9] "red"    "red"    "red"    "red"    "red"    "red"    "red"    "red"
 [17] "red"    "red"    "red"    "red"    "red"    "red"    "red"    "red"
 [25] "red"    "red"    "red"    "red"    "red"    "red"    "red"    "red"
 [33] "red"    "red"    "red"    "red"    "red"    "red"    "red"    "red"
 [41] "red"    "red"    "red"    "red"    "red"    "red"    "red"    "red"
 [49] "red"    "red"    "green3" "green3" "green3" "green3" "green3" "green3"
 [57] "green3" "green3" "green3" "green3" "green3" "green3" "green3" "green3"
 [65] "green3" "green3" "green3" "green3" "green3" "green3" "green3" "green3"
 [73] "green3" "green3" "green3" "green3" "green3" "green3" "green3" "green3"
 [81] "green3" "green3" "green3" "green3" "green3" "green3" "green3" "green3"
 [89] "green3" "green3" "green3" "green3" "green3" "green3" "green3" "green3"
 [97] "green3" "green3" "green3" "green3" "blue"   "blue"   "blue"   "blue"
[105] "blue"   "blue"   "blue"   "blue"   "blue"   "blue"   "blue"   "blue"
[113] "blue"   "blue"   "blue"   "blue"   "blue"   "blue"   "blue"   "blue"
[121] "blue"   "blue"   "blue"   "blue"   "blue"   "blue"   "blue"   "blue"
[129] "blue"   "blue"   "blue"   "blue"   "blue"   "blue"   "blue"   "blue"
[137] "blue"   "blue"   "blue"   "blue"   "blue"   "blue"   "blue"   "blue"
[145] "blue"   "blue"   "blue"   "blue"   "blue"   "blue"
>
```

- **#Converting classification column into factor and randomizing data:**

  Output:

```
> my_data$variety <- as.factor(my_data$ variety);
> my_data <- my_data[sample(nrow(my_data), ), ] ;
> head(my_data);
    sepal.length sepal.width petal.length petal.width    variety
91           5.5         2.6          4.4         1.2 Versicolor
62           5.9         3.0          4.2         1.5 Versicolor
80           5.7         2.6          3.5         1.0 Versicolor
53           6.9         3.1          4.9         1.5 Versicolor
48           4.6         3.2          1.4         0.2     Setosa
124          6.3         2.7          4.9         1.8  Virginica
```

- **#Training testing (70/30) partition:**

  Output:

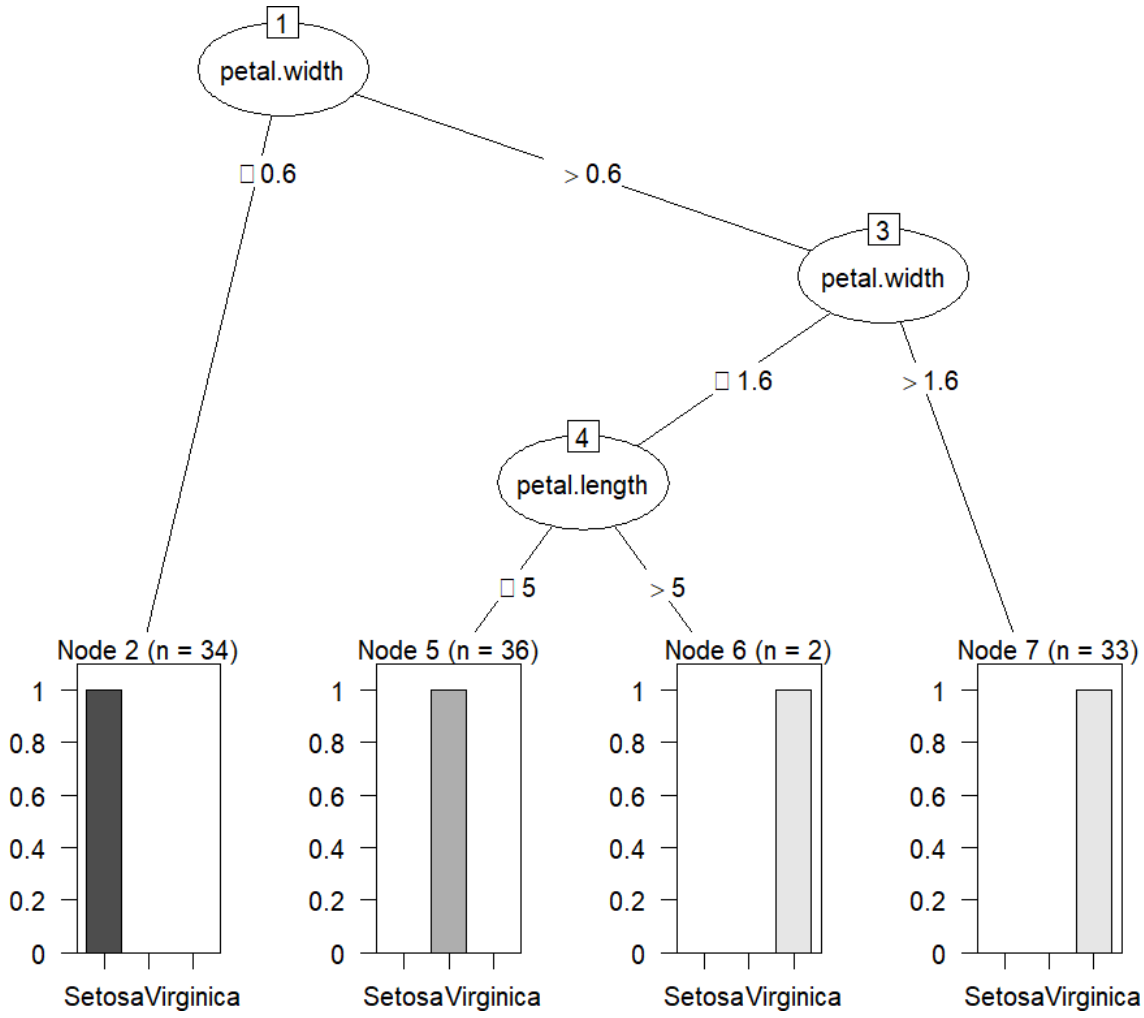
```
> TrainData <- my_data[1:105,]
> TestData <- my_data[106:150,]
```

i. ##Implement ID3 model:

Output:

```
> m1 <- J48(variety~., data = TrainData);
> plot(m1);
```

- **###Plot graph**



```
> summary(m1);

=== Summary ===

Correctly Classified Instances         105               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances              105

=== Confusion Matrix ===

  a  b  c   <-- classified as
 34  0  0 |  a = Setosa
  0 36  0 |  b = Versicolor
  0  0 35 |  c = Virginica
```

- **###Prediction for new data**

  Output:

```
> irisPred <- predict(m1, TestData);
> df<-data.frame(irisPred,TestData$variety);
> df;
     irisPred TestData.variety
1      Setosa           Setosa
2      Setosa           Setosa
3  Versicolor       Versicolor
4   Virginica        Virginica
5  Versicolor       Versicolor
6      Setosa           Setosa
7   Virginica       Versicolor
8   Virginica       Versicolor
9   Virginica        Virginica
10  Virginica       Versicolor
11  Virginica        Virginica
12  Virginica        Virginica
13     Setosa           Setosa
14 Versicolor       Versicolor
15     Setosa           Setosa
16 Versicolor        Virginica
17  Virginica        Virginica
18 Versicolor       Versicolor
19  Virginica        Virginica
20 Versicolor       Versicolor
21     Setosa           Setosa
22     Setosa           Setosa
23 Versicolor       Versicolor
24  Virginica        Virginica
25  Virginica        Virginica
26     Setosa           Setosa
27  Virginica        Virginica
28  Virginica        Virginica
29     Setosa           Setosa
30     Setosa           Setosa
31  Virginica        Virginica
32     Setosa           Setosa
33 Versicolor       Versicolor
34 Versicolor       Versicolor
35 Versicolor       Versicolor
36     Setosa           Setosa
```

2. **# C5.0 decision tree classification:**

- **#get the C5.0 package:**

  Output:

```
> install.packages("C50")
Installing package into 'C:/Users/Muskan/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
also installing the dependency 'Cubist'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/Cubist_0.4.1.zip'
Content type 'application/zip' length 889263 bytes (868 KB)
downloaded 868 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/C50_0.1.7.zip'
Content type 'application/zip' length 343195 bytes (335 KB)
downloaded 335 KB

package 'Cubist' successfully unpacked and MD5 sums checked
package 'C50' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Muskan\AppData\Local\Temp\RtmpCym5Ad\downloaded_packages
> library(C50);
```

- **# For decision tree classification load Iris dataset:**

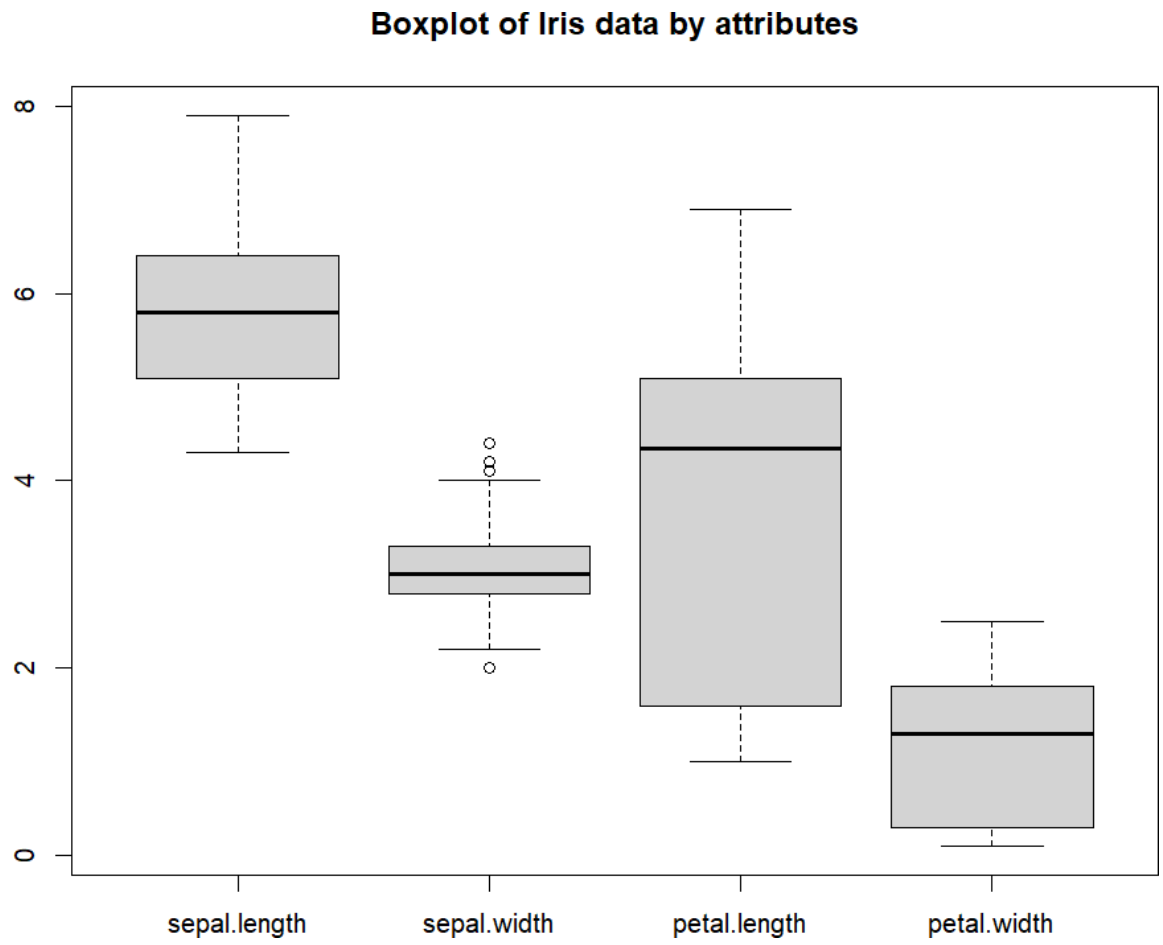
  Output:

```
> ir;
    sepal.length sepal.width petal.length petal.width    variety
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
7            4.6         3.4          1.4         0.3     Setosa
8            5.0         3.4          1.5         0.2     Setosa
9            4.4         2.9          1.4         0.2     Setosa
10           4.9         3.1          1.5         0.1     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
14           4.3         3.0          1.1         0.1     Setosa
15           5.8         4.0          1.2         0.2     Setosa
16           5.7         4.4          1.5         0.4     Setosa
17           5.4         3.9          1.3         0.4     Setosa
18           5.1         3.5          1.4         0.3     Setosa
19           5.7         3.8          1.7         0.3     Setosa
20           5.1         3.8          1.5         0.3     Setosa
21           5.4         3.4          1.7         0.2     Setosa
22           5.1         3.7          1.5         0.4     Setosa
```

- **# summary, boxplot, pairs plot:**

  Output:

```
> summary(ir);
  sepal.length    sepal.width     petal.length    petal.width      variety
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   Length:150
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   Class :character
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   Mode  :character
 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
> boxplot(ir[-5], main = 'Boxplot of Iris data by attributes');
```



- **#Converting classification column into factor and randomizing data:**

  Output:

```
> ir $variety <- as.factor(ir $ variety);
> ir <- ir [sample(nrow(ir), ), ];
> head(ir);
    sepal.length sepal.width petal.length petal.width    variety
129          6.4         2.8          5.6         2.1  Virginica
130          7.2         3.0          5.8         1.6  Virginica
68           5.8         2.7          4.1         1.0 Versicolor
111          6.5         3.2          5.1         2.0  Virginica
124          6.3         2.7          4.9         1.8  Virginica
65           5.6         2.9          3.6         1.3 Versicolor
```

- **#Training testing (70/30) partition:**

  Output:

```
> TrainData <- ir [1:105,];
> TestData <- ir [106:150,];
```

- **# C5.0 classification model:**

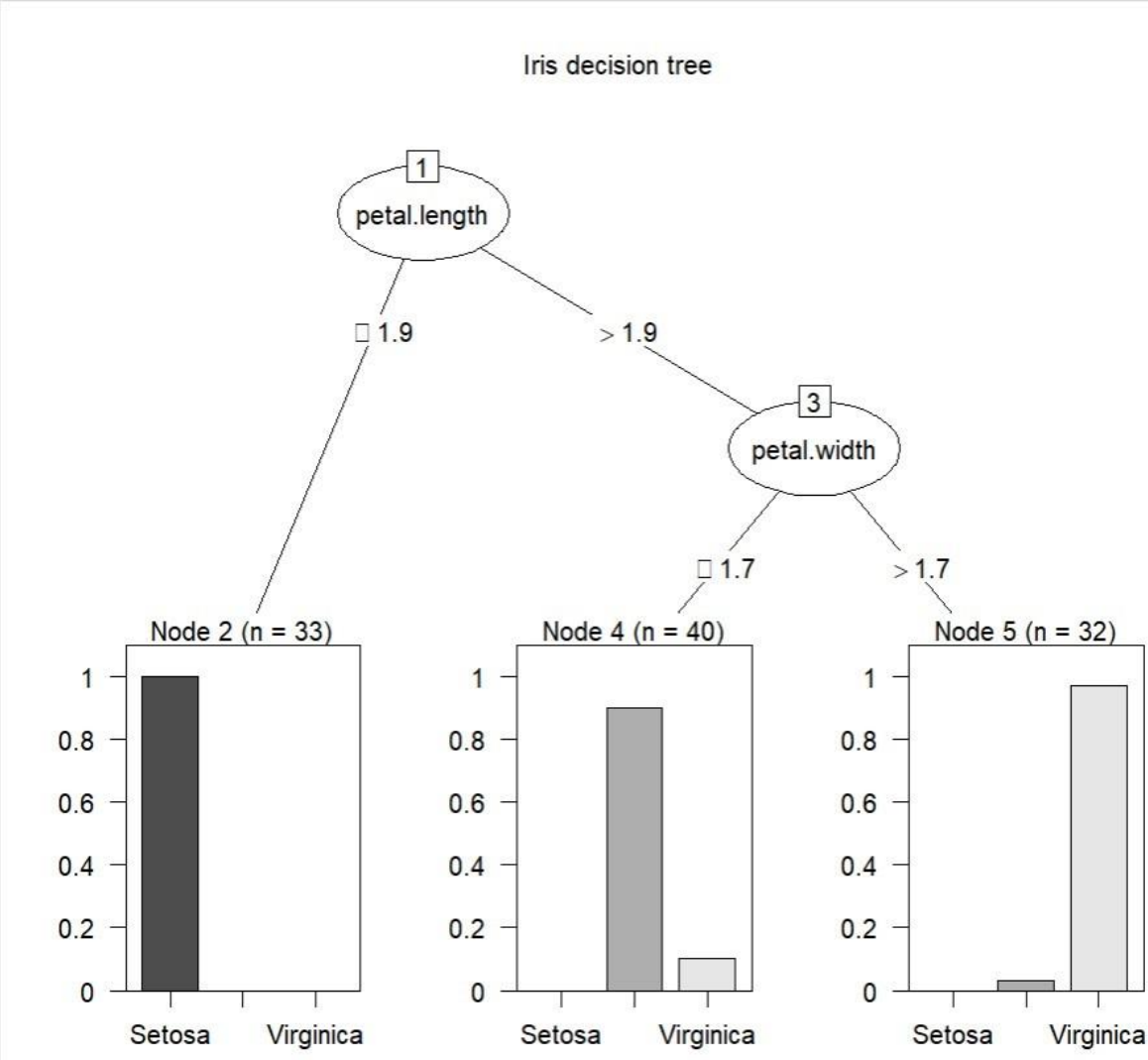
Output:

```
> irTree <- C5.0(TrainData[,-5], TrainData[,5]);
> summary(irTree);

Call:
C5.0.default(x = TrainData[, -5], y = TrainData[, 5])


C5.0 [Release 2.07 GPL Edition]         Sat Feb  4 12:38:10 2023
-------------------------------

Class specified by attribute `outcome'

Read 105 cases (5 attributes) from undefined.data

Decision tree:

petal.length <= 1.9: Setosa (33)
petal.length > 1.9:
:...petal.width <= 1.7: Versicolor (40/4)
    petal.width > 1.7: Virginica (32/1)


Evaluation on training data (105 cases):

           Decision Tree
         ----------------
         Size      Errors

            3    5( 4.8%)   <<


          (a)   (b)   (c)    <-classified as
         ----  ----  ----
           33                (a): class Setosa
                 36     1    (b): class Versicolor
                  4    31    (c): class Virginica


        Attribute usage:

        100.00% petal.length
         68.57% petal.width


Time: 0.0 secs
```

- **###View the model graphically**

  Output:

```
> plot(irTree, main = 'Iris decision tree');
```



- **#Build a rules set:**

  Output:

```
> irRules <- C5.0(TrainData [,-5], TrainData [,5], rules = TRUE);
```

- **# Test the prediction [classification] capability:**

Output:

```
> pred <- predict(irRules, newdata = TestData[, -5]);
> df <- data.frame(TestData$ variety, pred);
> df;
   TestData.variety       pred
1         Virginica  Virginica
2        Versicolor Versicolor
3            Setosa     Setosa
4         Virginica  Virginica
5            Setosa     Setosa
6            Setosa     Setosa
7            Setosa     Setosa
8            Setosa     Setosa
9         Virginica  Virginica
10           Setosa     Setosa
11       Versicolor Versicolor
12       Versicolor Versicolor
13           Setosa     Setosa
14        Virginica  Virginica
15           Setosa     Setosa
16        Virginica Versicolor
17       Versicolor Versicolor
18        Virginica  Virginica
19        Virginica  Virginica
20        Virginica  Virginica
```

- **# See the confusion matrix:**

Output:

```
> confusion <- table(TestData$ variety, pred);
> confusion;
            pred
             Setosa Versicolor Virginica
  Setosa         17          0         0
  Versicolor      0         13         0
  Virginica       0          1        14
```

```
> confusionMatrix(confusion);
Confusion Matrix and Statistics

            pred
             Setosa Versicolor Virginica
  Setosa         17          0         0
  Versicolor      0         13         0
  Virginica       0          1        14

Overall Statistics

               Accuracy : 0.9778
                 95% CI : (0.8823, 0.9994)
    No Information Rate : 0.3778
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9665

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Setosa Class: Versicolor Class: Virginica
Sensitivity                 1.0000            0.9286           1.0000
Specificity                 1.0000            1.0000           0.9677
Pos Pred Value              1.0000            1.0000           0.9333
Neg Pred Value              1.0000            0.9688           1.0000
Prevalence                  0.3778            0.3111           0.3111
Detection Rate              0.3778            0.2889           0.3111
Detection Prevalence        0.3778            0.2889           0.3333
Balanced Accuracy           1.0000            0.9643           0.9839
```

B. Supervised Learning: Naive Bayesian and KNN:

1. **Performing Naive Bayes Classification on iris dataset:**

- **#Installing packages:**

Output:

```
> install.packages("e1071");
Installing package into 'C:/Users/Muskan/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)

  There is a binary version available but the source version
  is later:
      binary source needs_compilation
e1071 1.7-12 1.7-13              TRUE

installing the source package 'e1071'
```

```
> install.packages("caTools");
Installing package into 'C:/Users/Muskan/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
also installing the dependency 'bitops'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/bitops_1.0-7.zip'
Content type 'application/zip' length 31679 bytes (30 KB)
downloaded 30 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/caTools_1.18.2.zip'
Content type 'application/zip' length 246225 bytes (240 KB)
downloaded 240 KB

package 'bitops' successfully unpacked and MD5 sums checked
package 'caTools' successfully unpacked and MD5 sums checked
```

```
> install.packages("caret");
Installing package into 'C:/Users/Muskan/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/caret_6.0-93.zip'
Content type 'application/zip' length 3577965 bytes (3.4 MB)
downloaded 3.4 MB

package 'caret' successfully unpacked and MD5 sums checked
Warning in install.packages :
  cannot remove prior installation of package 'caret'
Warning in install.packages :
  problem copying C:\Users\Muskan\AppData\Local\R\win-library\4.2\00LOCK\caret\libs\x64\caret.dll to C:\Users\Muskan\A
denied
Warning in install.packages :
  restored 'caret'

The downloaded binary packages are in
        C:\Users\Muskan\AppData\Local\Temp\RtmpiuyVkM\downloaded_packages
```

- **#Loading packages:**

Output:

```
> library(e1071);
> library(caTools);
> library(caret);
Loading required package: ggplot2
Loading required package: lattice
>
```

- **#Load Data:**

Output:

```
> my_data <- read.csv("C://Users/Muskan/OneDrive/Documents/Iris.csv");
> my_data;
    sepal.length sepal.width petal.length petal.width    variety
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
7            4.6         3.4          1.4         0.3     Setosa
8            5.0         3.4          1.5         0.2     Setosa
9            4.4         2.9          1.4         0.2     Setosa
10           4.9         3.1          1.5         0.1     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
14           4.3         3.0          1.1         0.1     Setosa
15           5.8         4.0          1.2         0.2     Setosa
16           5.7         4.4          1.5         0.4     Setosa
17           5.4         3.9          1.3         0.4     Setosa
18           5.1         3.5          1.4         0.3     Setosa
19           5.7         3.8          1.7         0.3     Setosa
```

- **#Randomize data:**

Output:

```
> my_data <- my_data[sample(nrow(my_data), ), ];
> head(my_data);
    sepal.length sepal.width petal.length petal.width   variety
119          7.7         2.6          6.9         2.3 Virginica
123          7.7         2.8          6.7         2.0 Virginica
139          6.0         3.0          4.8         1.8 Virginica
125          6.7         3.3          5.7         2.1 Virginica
38           4.9         3.6          1.4         0.1    Setosa
8            5.0         3.4          1.5         0.2    Setosa
```

- **#Splitting data into train and test** data using SplitRatio parameter of sample.split function in R which automatically divided data in given ratio:

Output:

```
> split <- sample.split(my_data, SplitRatio = 0.7);
> TrainData <- subset(my_data, split == "TRUE");
> TestData <- subset(my_data, split == "FALSE");
```

- **# Fitting Naive Bayes Model to training dataset**

Output:

```
> classifier_cl <- naiveBayes(variety ~ ., data = TrainData);
> classifier_cl;

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
    Setosa Versicolor  Virginica
 0.3333333  0.3222222  0.3444444

Conditional probabilities:
            sepal.length
Y                [,1]      [,2]
  Setosa     4.980000 0.3800181
  Versicolor 5.968966 0.4583382
  Virginica  6.648387 0.6179919

            sepal.width
Y                [,1]      [,2]
  Setosa     3.453333 0.3775907
  Versicolor 2.758621 0.3179367
  Virginica  3.029032 0.3407771

            petal.length
Y                [,1]      [,2]
  Setosa     1.486667 0.1814374
  Versicolor 4.272414 0.3844259
  Virginica  5.551613 0.5726377

            petal.width
Y                [,1]      [,2]
  Setosa     0.240000 0.1162637
  Versicolor 1.313793 0.1726353
  Virginica  2.022581 0.2789381
```

- **# Predicting on test data:**

Output:

```
> y_pred <- predict(classifier_cl, newdata =TestData);
> y_pred;
 [1] Virginica  Virginica  Setosa     Versicolor Virginica  Versicolor Setosa     Setosa     Versicolor
Setosa     Setosa     Virginica  Versicolor Setosa     Virginica  Setosa
[17] Setosa     Setosa     Setosa     Virginica  Setosa     Versicolor Virginica  Virginica  Virginica
Virginica  Virginica  Setosa     Setosa     Versicolor Virginica  Virginica
[33] Setosa     Virginica  Versicolor Setosa     Versicolor Versicolor Versicolor Virginica  Virginica
Virginica  Virginica  Setosa     Setosa     Versicolor Virginica  Versicolor
[49] Versicolor Versicolor Virginica  Versicolor Setosa     Virginica  Versicolor Versicolor Setosa
Versicolor Versicolor Setosa
Levels: Setosa Versicolor Virginica
```

- **#Confusion matrix:**

Output:

```
> cm <- table(TestData$variety, y_pred);
> cm;
            y_pred
             Setosa Versicolor Virginica
  Setosa         20          0         0
  Versicolor      0         18         3
  Virginica       0          1        18
```

- **#Model Evaluation:**

  Output:

```
> confusionMatrix(cm);
Confusion Matrix and Statistics

            y_pred
             Setosa Versicolor Virginica
  Setosa         20          0         0
  Versicolor      0         18         3
  Virginica       0          1        18

Overall Statistics

               Accuracy : 0.9333
                 95% CI : (0.838, 0.9815)
    No Information Rate : 0.35
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9001

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Setosa Class: Versicolor Class: Virginica
Sensitivity                 1.0000            0.9474           0.8571
Specificity                 1.0000            0.9268           0.9744
Pos Pred Value              1.0000            0.8571           0.9474
Neg Pred Value              1.0000            0.9744           0.9268
Prevalence                  0.3333            0.3167           0.3500
Detection Rate              0.3333            0.3000           0.3000
Detection Prevalence        0.3333            0.3500           0.3167
Balanced Accuracy           1.0000            0.9371           0.9158
```

2. **Performing KNN (K- Nearest Neighbour) Classification on iris dataset:**

- **#Installing packages:**

  Output:

```
> install.packages("e1071");
Installing package into 'C:/Users/Muskan/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)

  There is a binary version available but the source version
  is later:
      binary source needs_compilation
e1071 1.7-12 1.7-13              TRUE

installing the source package 'e1071'

installing to C:/Users/Muskan/AppData/Local/R/win-library/4.2/00LOCK-e1071/00new/e1071/libs/x64
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** installing vignettes
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (e1071)

The downloaded source packages are in
        'C:\Users\Muskan\AppData\Local\Temp\RtmpiuyVkM\downloaded_packages'

> install.packages("caTools");
Installing package into 'C:/Users/Muskan/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
also installing the dependency 'bitops'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/bitops_1.0-7.zip'
Content type 'application/zip' length 31679 bytes (30 KB)
downloaded 30 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/caTools_1.18.2.zip'
Content type 'application/zip' length 246225 bytes (240 KB)
downloaded 240 KB

package 'bitops' successfully unpacked and MD5 sums checked
package 'caTools' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Muskan\AppData\Local\Temp\RtmpiuyVkM\downloaded_packages
```

```
> install.packages("caret");
Installing package into 'C:/Users/Muskan/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/caret_6.0-93.zip'
Content type 'application/zip' length 3577965 bytes (3.4 MB)
downloaded 3.4 MB

package 'caret' successfully unpacked and MD5 sums checked
Warning in install.packages :
  cannot remove prior installation of package 'caret'
Warning in install.packages :
  problem copying C:\Users\Muskan\AppData\Local\R\win-library\4.2\00LOCK\caret\libs\x64\caret.dll to C:\Users\Muskan\A
denied
Warning in install.packages :
  restored 'caret'

The downloaded binary packages are in
        C:\Users\Muskan\AppData\Local\Temp\RtmpiuyVkM\downloaded_packages
```

- **#Loading packages:**

Output:

```
> library(e1071);
> library(caTools);
> library(caret);
Loading required package: ggplot2
Loading required package: lattice
>
```

- **#Load Data:**

Output:

```
> my_data <- read.csv("C://Users/Muskan/OneDrive/Documents/Iris.csv");
> my_data;
    sepal.length sepal.width petal.length petal.width    variety
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
7            4.6         3.4          1.4         0.3     Setosa
8            5.0         3.4          1.5         0.2     Setosa
9            4.4         2.9          1.4         0.2     Setosa
10           4.9         3.1          1.5         0.1     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
14           4.3         3.0          1.1         0.1     Setosa
15           5.8         4.0          1.2         0.2     Setosa
16           5.7         4.4          1.5         0.4     Setosa
17           5.4         3.9          1.3         0.4     Setosa
18           5.1         3.5          1.4         0.3     Setosa
19           5.7         3.8          1.7         0.3     Setosa
```

- **#Randomize data:**

Output:

```
> my_data <- my_data[sample(nrow(my_data), ), ];
> head(my_data);
    sepal.length sepal.width petal.length petal.width    variety
27           5.0         3.4          1.6         0.4     Setosa
60           5.2         2.7          3.9         1.4 Versicolor
150          5.9         3.0          5.1         1.8  Virginica
68           5.8         2.7          4.1         1.0 Versicolor
144          6.8         3.2          5.9         2.3  Virginica
30           4.7         3.2          1.6         0.2     Setosa
```

- **#Splitting data into train and test** data using SplitRatio parameter of sample.split function in R which automatically divided data in given ratio:

Output:

```
> split <- sample.split(my_data, SplitRatio = 0.7);
> TrainData <- subset(my_data, split == "TRUE");
> TestData <- subset(my_data, split == "FALSE");
```

- **# Feature Scaling:**

  Output:

```
> train_scale <- scale(TrainData [, 1:4]);
> test_scale <- scale(TestData [, 1:4]);
```

- **#Fitting KNN Model to training dataset:**

  Output:

```
> library(class);
> classifier_knn <- knn(train = train_scale, test = test_scale, cl = TrainData$variety, k = 1);

> classifier_knn;
 [1] Virginica  Virginica  Setosa     Setosa     Setosa
 [6] Versicolor Virginica  Setosa     Setosa     Virginica
[11] Setosa     Virginica  Virginica  Virginica  Virginica
[16] Virginica  Versicolor Virginica  Virginica  Versicolor
[21] Virginica  Versicolor Setosa     Setosa     Virginica
[26] Versicolor Virginica  Versicolor Setosa     Virginica
[31] Virginica  Versicolor Setosa     Setosa     Versicolor
[36] Versicolor Virginica  Setosa     Setosa     Versicolor
[41] Versicolor Setosa     Versicolor Setosa     Versicolor
[46] Versicolor Setosa     Versicolor Setosa     Setosa
[51] Virginica  Versicolor Virginica  Versicolor Setosa
[56] Versicolor Virginica  Virginica  Versicolor Virginica
Levels: Setosa Versicolor Virginica
```

- **#Confusion Matrix:**

  Output:

```
> cm <- table(TestData$variety, classifier_knn)
> cm;
            classifier_knn
             Setosa Versicolor Virginica
  Setosa         19          0         0
  Versicolor      0         19         2
  Virginica       0          0        20
```

- **#Model Evaluation:**

  i. ## Calculate out of Sample error:

  Output:

```
> misClassError <- mean(classifier_knn != TestData$variety);
> print(paste('Accuracy =', 1-misClassError));
[1] "Accuracy = 0.966666666666667"
```

```
> classifier_knn <- knn(train = train_scale, test = test_scale, cl = TrainData$variety, k = 3);
> misClassError <- mean(classifier_knn != TestData$variety);
> print(paste('Accuracy =', 1-misClassError));
[1] "Accuracy = 0.966666666666667"

> classifier_knn <- knn(train = train_scale, test = test_scale, cl = TrainData$variety, k = 5);
> misClassError <- mean(classifier_knn != TestData$variety);
> print(paste('Accuracy =', 1-misClassError));
[1] "Accuracy = 0.95"

> #K = 7
> classifier_knn <- knn(train = train_scale, test = test_scale, cl = TrainData$variety, k = 7);
> misClassError <- mean(classifier_knn != TestData$variety);
> print(paste('Accuracy =', 1-misClassError));
[1] "Accuracy = 0.933333333333333"

> classifier_knn <- knn(train = train_scale, test = test_scale, cl = TrainData$variety, k = 9);
> misClassError <- mean(classifier_knn != TestData$variety);
> print(paste('Accuracy =', 1-misClassError));
[1] "Accuracy = 0.983333333333333"

> classifier_knn <- knn(train = train_scale, test = test_scale, cl = TrainData$variety, k = 15);
> misClassError <- mean(classifier_knn != TestData$variety);
> print(paste('Accuracy =', 1-misClassError));
[1] "Accuracy = 0.933333333333333"

> classifier_knn <- knn(train = train_scale, test = test_scale, cl = TrainData$variety, k = 19);
> misClassError <- mean(classifier_knn != TestData$variety);
> print(paste('Accuracy =', 1-misClassError));
[1] "Accuracy = 0.933333333333333"
```

The model achieved the highest accuracy of 98.33% with k = 9;

Conclusion:

Hence, we successfully implemented and analyzed the ID3, C4.5, Naive Bayes and KNN classification algorithms using the R programming language.