Practical 10- Decision Tree

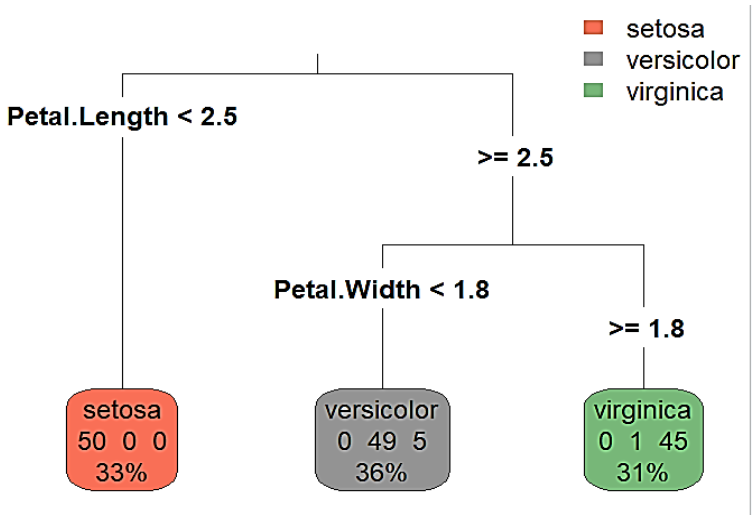
Aim: Decision Tree

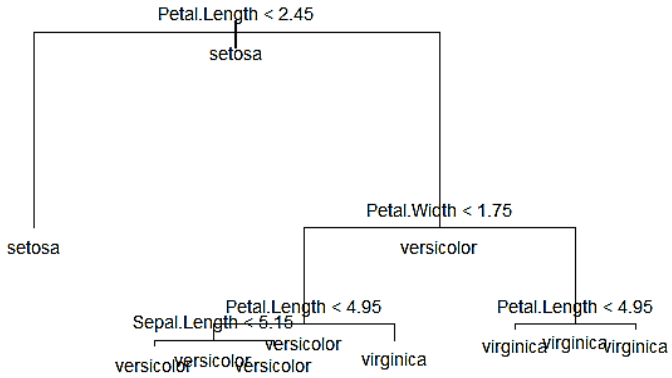
1. Decision tree classification on the iris dataset using different tree-building algorithms.

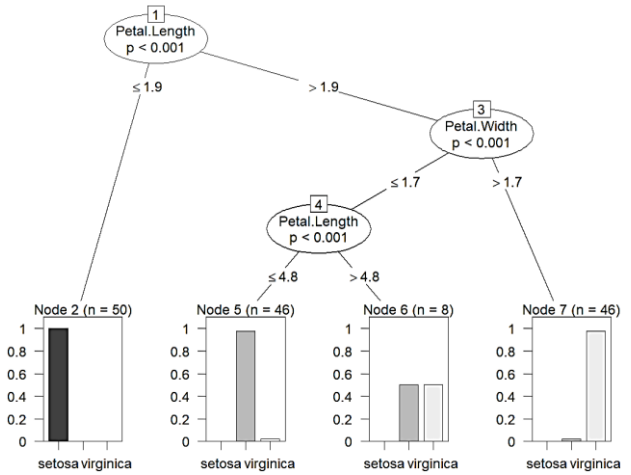
```
library(rpart)
library(rpart.plot)
library(tree)
library(party)
library(caret)
library(e1071)
mydata <- data.frame(iris)
attach(mydata)
model <- rpart(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
          data = mydata, method = "class")
rpart.plot(model, type = 3, extra = 101, fallen.leaves = TRUE)
model1 <- tree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
          data = mydata)
plot(model1)
text(model1, all = TRUE, cex = 0.6)
model2 <- ctree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
           data = mydata)
plot(model2)
model1 <- tree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, data =
mydata, control = tree.control(nobs = nrow(mydata), mincut = 10))
plot(model1)
text(model1, all = TRUE, cex = 0.6)
predict(model1, iris)
model2 <- ctree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
          data = mydata, controls = ctree_control(maxdepth = 2))
plot(model2)
set.seed(123)
train_index <- createDataPartition(mydata$Species, p = 0.7, list = FALSE)
train_data <- mydata[train_index, ]
test_data <- mydata[-train_index, ]
model_rpart <- rpart(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
               data = train_data, method = "class")
pred_rpart <- predict(model_rpart, test_data, type = "class")
conf_matrix_rpart <- confusionMatrix(pred_rpart, test_data$Species)
print(conf_matrix_rpart)
model_tree <- tree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
              data = train_data)
pred_tree <- predict(model_tree, test_data, type = "class")
conf_matrix_tree <- confusionMatrix(pred_tree, test_data$Species)
print(conf_matrix_tree)
model_ctree <- ctree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
               data = train_data)
```

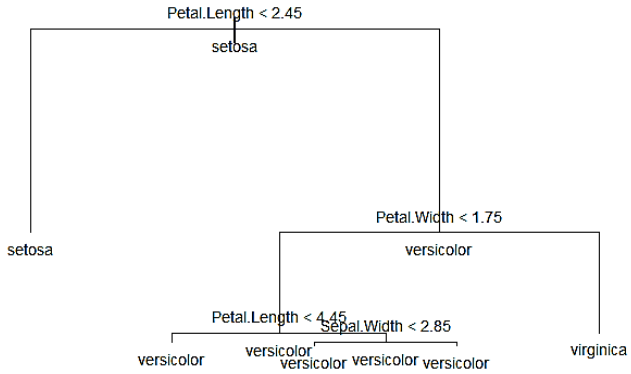
```
pred_ctree <- predict(model_ctree, test_data)
conf_matrix_ctree <- confusionMatrix(pred_ctree, test_data$Species)
print(conf_matrix_ctree)
accuracy_rpart <- conf_matrix_rpart$overall["Accuracy"]
accuracy_tree <- conf_matrix_tree$overall["Accuracy"]
accuracy_ctree <- conf_matrix_ctree$overall["Accuracy"]
accuracy_results <- data.frame(Model = c("rpart", "tree", "ctree"),
                               Accuracy = c(accuracy_rpart, accuracy_tree, accuracy_ctree))
print(accuracy_results)
barplot(accuracy_results$Accuracy, names.arg = accuracy_results$Model,
      col = c("red", "blue", "green"), main = "Decision Tree Model Accuracy",
      ylab = "Accuracy", ylim = c(0, 1))
```

Output

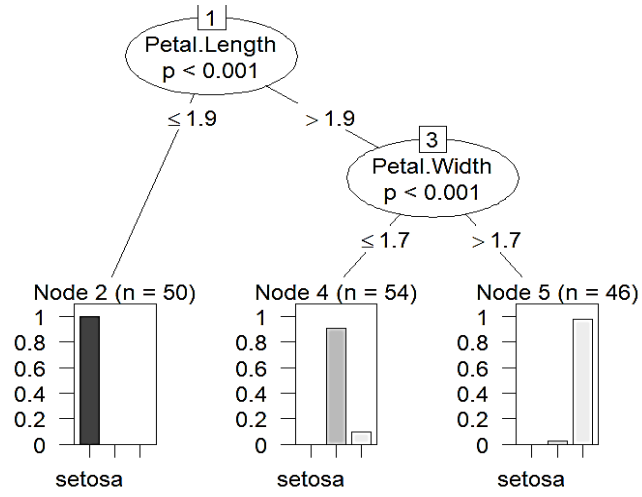






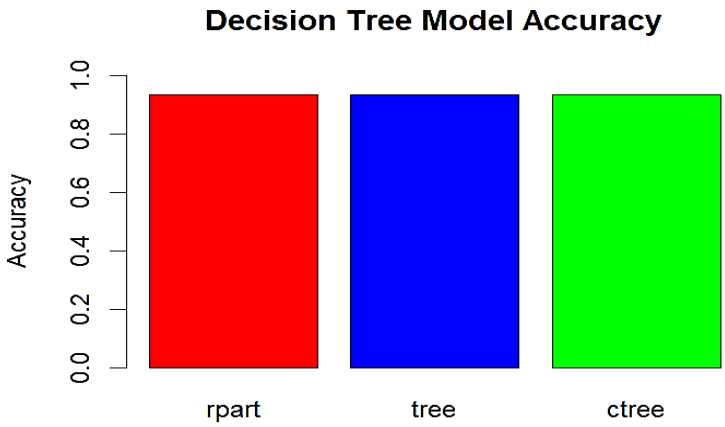



```
> predict(model1, iris)
    setosa versicolor  virginica
1        1 0.00000000 0.00000000
2        1 0.00000000 0.00000000
3        1 0.00000000 0.00000000
4        1 0.00000000 0.00000000
5        1 0.00000000 0.00000000
6        1 0.00000000 0.00000000
7        1 0.00000000 0.00000000
8        1 0.00000000 0.00000000
9        1 0.00000000 0.00000000
10       1 0.00000000 0.00000000
11       1 0.00000000 0.00000000
12       1 0.00000000 0.00000000
13       1 0.00000000 0.00000000
14       1 0.00000000 0.00000000
```



```
> print(conf_matrix_rpart)
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
> print(conf_matrix_tree)
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
> print(conf_matrix_ctree)
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
```

```
> print(accuracy_results)
  Model  Accuracy
1 rpart 0.9333333
2  tree 0.9333333
3 ctree 0.9333333
```

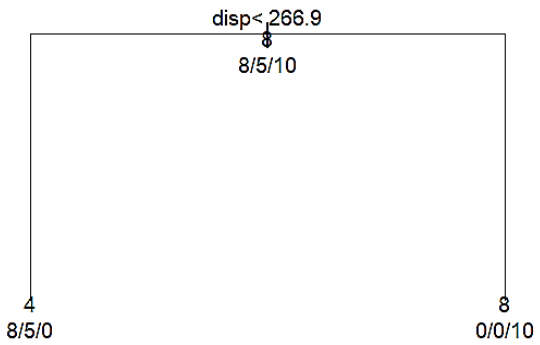


Q 2) Decision Tree on mtcars Dataset

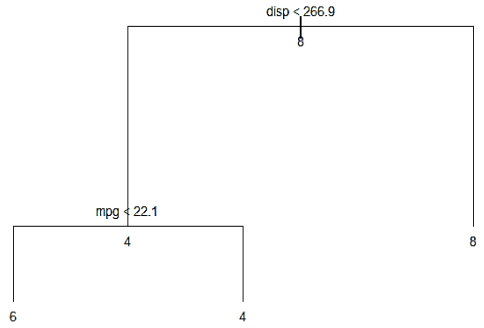
```
library(rpart)
library(tree)
library(party)
library(rpart.plot)
library(caret)
mydata <- data.frame(mtcars)
mydata$cyl <- as.factor(mydata$cyl)
set.seed(123)
train_index <- createDataPartition(mydata$cyl, p = 0.7, list = FALSE)
train_data <- mydata[train_index, ]
test_data <- mydata[-train_index, ]
model_rpart <- rpart(cyl ~ mpg + hp + wt + disp, data = train_data, method = "class",
                     control = rpart.control(cp = 0.01))
plot(model_rpart, uniform = TRUE, margin = 0.1)
text(model_rpart, use.n = TRUE, all = TRUE, cex = 0.8)
pred_rpart <- predict(model_rpart, test_data, type = "class")
conf_matrix_rpart <- confusionMatrix(pred_rpart, test_data$cyl)
print(conf_matrix_rpart)
model_tree <- tree(cyl ~ mpg + hp + wt + disp, data = train_data)
plot(model_tree)
text(model_tree, all = TRUE, cex = 0.6)
pred_tree <- predict(model_tree, test_data, type = "class")
conf_matrix_tree <- confusionMatrix(pred_tree, test_data$cyl)
print(conf_matrix_tree)
model_ctree <- ctree(cyl ~ mpg + hp + wt + disp, data = train_data)
plot(model_ctree)
pred_ctree <- predict(model_ctree, test_data)
conf_matrix_ctree <- confusionMatrix(pred_ctree, test_data$cyl)
print(conf_matrix_ctree)
model_rpart_tuned <- rpart(cyl ~ mpg + hp + wt + disp, data = train_data, method = "class",
                           control = rpart.control(maxdepth = 3))
pred_rpart_tuned <- predict(model_rpart_tuned, test_data, type = "class")
```

```
conf_matrix_rpart_tuned <- confusionMatrix(pred_rpart_tuned, test_data$cyl)
print(conf_matrix_rpart_tuned)

model_tree_tuned <- tree(cyl ~ mpg + hp + wt + disp, data = train_data,
                         control = tree.control(nobs = nrow(train_data), mincut = 5))
pred_tree_tuned <- predict(model_tree_tuned, test_data, type = "class")
conf_matrix_tree_tuned <- confusionMatrix(pred_tree_tuned, test_data$cyl)
print(conf_matrix_tree_tuned)
model_ctree_tuned <- ctree(cyl ~ mpg + hp + wt + disp, data = train_data,
                           controls = ctree_control(maxdepth = 3, minsplit = 5))
pred_ctree_tuned <- predict(model_ctree_tuned, test_data)
conf_matrix_ctree_tuned <- confusionMatrix(pred_ctree_tuned, test_data$cyl)
print(conf_matrix_ctree_tuned)
set.seed(123)
train_control <- trainControl(method = "cv", number = 10)
cv_model_rpart <- train(cyl ~ mpg + hp + wt + disp, data = train_data, method = "rpart",
                        trControl = train_control)
pred_cv_rpart <- predict(cv_model_rpart, test_data)
conf_matrix_cv_rpart <- confusionMatrix(pred_cv_rpart, test_data$cyl)
print(conf_matrix_cv_rpart)
accuracy_results <- data.frame(Model = c("rpart", "tree", "ctree", "rpart (Tuned)", "tree
(Tuned)", "ctree (Tuned)"),
                               Accuracy = c(conf_matrix_rpart$overall["Accuracy"],
                                            conf_matrix_tree$overall["Accuracy"],
                                            conf_matrix_ctree$overall["Accuracy"],
                                            conf_matrix_rpart_tuned$overall["Accuracy"],
                                            conf_matrix_tree_tuned$overall["Accuracy"],
                                            conf_matrix_ctree_tuned$overall["Accuracy"]))
print(accuracy_results)
barplot(accuracy_results$Accuracy, names.arg = accuracy_results$Model,
        col = rainbow(6), main = "Decision Tree Model Accuracy (mtcars)",
        ylab = "Accuracy", ylim = c(0, 1))
```

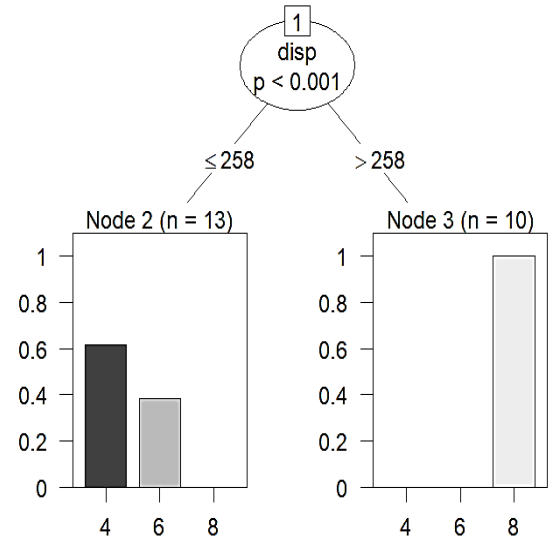
Output:



```
> print(conf_matrix_rpart)
Confusion Matrix and Statistics

          Reference
Prediction 4 6 8
         4 3 2 0
         6 0 0 0
         8 0 0 4

Overall Statistics

               Accuracy : 0.7778
                 95% CI : (0.3999, 0.9719)
    No Information Rate : 0.4444
    P-Value [Acc > NIR] : 0.04635

                  Kappa : 0.64

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 4 Class: 6 Class: 8
Sensitivity            1.0000   0.0000   1.0000
Specificity            0.6667   1.0000   1.0000
Pos Pred Value         0.6000      NaN   1.0000
```




```
> print(conf_matrix_tree)
Confusion Matrix and Statistics

          Reference
Prediction 4 6 8
         4 2 0 0
         6 1 2 0
         8 0 0 4

Overall Statistics

               Accuracy : 0.8889
                 95% CI : (0.5175, 0.9972)
    No Information Rate : 0.4444
    P-Value [Acc > NIR] : 0.008289

                  Kappa : 0.8302

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 4 Class: 6 Class: 8
Sensitivity            0.6667   1.0000   1.0000
Specificity            1.0000   0.8571   1.0000
Pos Pred Value         1.0000   0.6667   1.0000
```




```
> print(conf_matrix_ctree)
Confusion Matrix and Statistics

          Reference
Prediction 4 6 8
         4 3 2 0
         6 0 0 0
         8 0 0 4

Overall Statistics

               Accuracy : 0.7778
                 95% CI : (0.3999, 0.9719)
    No Information Rate : 0.4444
    P-Value [Acc > NIR] : 0.04635

                  Kappa : 0.64

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 4 Class: 6 Class: 8
Sensitivity            1.0000   0.0000   1.0000
Specificity            0.6667   1.0000   1.0000
```

```
> print(conf_matrix_rpart_tuned)
Confusion Matrix and Statistics

          Reference
Prediction 4 6 8
         4 3 2 0
         6 0 0 0
         8 0 0 4

Overall Statistics

               Accuracy : 0.7778
                 95% CI : (0.3999, 0.9719)
    No Information Rate : 0.4444
    P-Value [Acc > NIR] : 0.04635

                  Kappa : 0.64

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 4 Class: 6 Class: 8
Sensitivity            1.0000   0.0000   1.0000
Specificity            0.6667   1.0000   1.0000
```

```
> print(conf_matrix_tree_tuned)
Confusion Matrix and Statistics

          Reference
Prediction 4 6 8
         4 2 0 0
         6 1 2 0
         8 0 0 4

Overall Statistics

               Accuracy : 0.8889
                 95% CI : (0.5175, 0.9972)
    No Information Rate : 0.4444
    P-Value [Acc > NIR] : 0.008289

                  Kappa : 0.8302

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 4 Class: 6 Class: 8
Sensitivity            0.6667   1.0000   1.0000
Specificity            1.0000   0.8571   1.0000


> print(conf_matrix_ctree_tuned)
Confusion Matrix and Statistics

          Reference
Prediction 4 6 8
         4 3 2 0
         6 0 0 0
         8 0 0 4

Overall Statistics

               Accuracy : 0.7778
                 95% CI : (0.3999, 0.9719)
    No Information Rate : 0.4444
    P-Value [Acc > NIR] : 0.04635

                  Kappa : 0.64

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 4 Class: 6 Class: 8
Sensitivity            1.0000   0.0000   1.0000
Specificity            0.6667   1.0000   1.0000


> print(conf_matrix_cv_rpart)
Confusion Matrix and Statistics

          Reference
Prediction 4 6 8
         4 3 2 0
         6 0 0 0
         8 0 0 4

Overall Statistics

               Accuracy : 0.7778
                 95% CI : (0.3999, 0.9719)
    No Information Rate : 0.4444
    P-Value [Acc > NIR] : 0.04635

                  Kappa : 0.64

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 4 Class: 6 Class: 8
Sensitivity            1.0000   0.0000   1.0000
Specificity            0.6667   1.0000   1.0000
```

```
> print(accuracy_results)
          Model  Accuracy
1         rpart 0.7777778
2          tree 0.8888889
3         ctree 0.7777778
4 rpart (Tuned) 0.7777778
5  tree (Tuned) 0.8888889
6 ctree (Tuned) 0.7777778
```