NAME : VISHAL KUMAR MAHATHA

Reg. No. : 20BRS1168

IoT Domain Analyst

Lab Assignment 11

IoT Data analysis using Decision Tree in RStudio

Practice 1 :

Decision Tree using rpart on Iris dataset

Code :

```
library(rpart)
data(iris)
set.seed(1234)
train <- sample(nrow(iris), 0.7 * nrow(iris))
iris_train <- iris[train, ]
iris_test <- iris[-train, ]
model <- rpart(Species ~ ., data = iris_train, method = "class")
#install.packages("rpart.plot")
library(rpart.plot)
rpart.plot(model)
predictions <- predict(model, iris_test, type = "class")
```

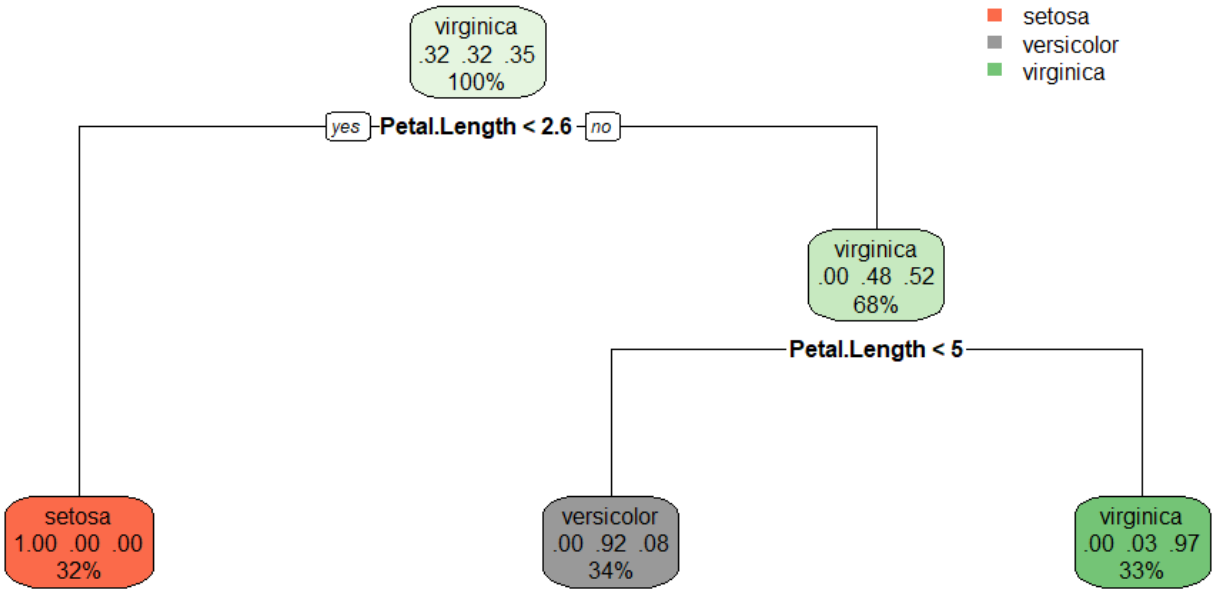
```
m_at <- table(iris_test$Species, predictions)

m_at
```

Output :

"Accuracy for test is found to be 0.911111111111111"

```
            predictions
             setosa versicolor virginica
setosa           16          0         0
versicolor        0         15         1
virginica         0          3        10
```



Practice 2 :

Decision Tree using ctree on readingSkills dataset

Code :

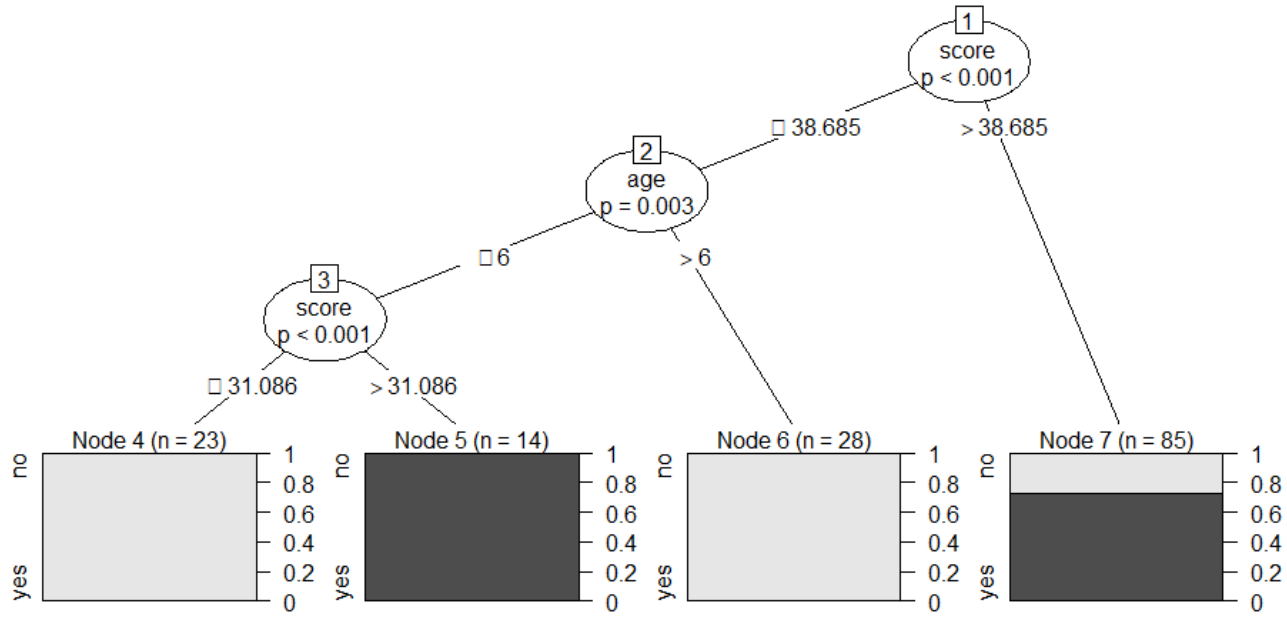
```
library(datasets)
library(caTools)
library(party)
library(dplyr)
library(magrittr)
data("readingSkills")
head(readingSkills)

sample_data = sample.split(readingSkills, SplitRatio = 0.8)
train_data <- subset(readingSkills, sample_data == TRUE)
test_data <- subset(readingSkills, sample_data == FALSE)
model<- ctree(nativeSpeaker ~ ., train_data)
plot(model)
# testing the people who are native speakers
# and those who are not
predict_model<-predict(model, test_data)
# creates a table to count how many are classified
# as native speakers and how many are not
m_at <- table(test_data$nativeSpeaker, predict_model)
m_at
ac_Test <- (sum(diag(m_at)) / sum(m_at))
print(paste('Accuracy for test is found to be', ac_Test))
```

Output :

"Accuracy for test is found to be 0.74"

```
     predict_model
      no yes
no    13  13
yes    0  24
```



Exercise 1 :

Decision tree using rpart on Advertisement dataset

Code :

dataset=Advertisement

head(dataset, 10)

```
# Encoding the target feature as factor
dataset$Gender = factor(dataset$Gender,levels = c(0, 1))

dataset=select(dataset,-Gender)

# Splitting the dataset into the Training set and Test set
library(caTools)
set.seed(123)

split = sample.split(dataset$Purchased,SplitRatio = 0.75)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)

# Fitting Decision Tree Classification to the Training set
library(rpart)
classifier = rpart(Purchased ~ .,training_set)

# Predicting the Test set results
y_pred = predict(classifier,test_set)

# Making the Confusion Matrix
```

```
cm = table(test_set$Age, y_pred)

# Plotting the tree
plot(classifier)
text(classifier)

ac_Test <- (sum(diag(cm)) / sum(cm))
print(paste('Accuracy for test is found to be', ac_Test))
```

Output :

"Accuracy for test is found to be 0.44"

```
   y_pred
    0.032967032967033 0.25 0.454545454545455 0.857142857142857 0.941176470588235 0.947368421052632 0.975  1
  0                55    1                 1                 1                 0                 0     4  2
  1                 2    4                 2                 5                 7                 4     9  3
> |
```

Exercise 2 :

Decision tree using ctree on Advertisement

Code :

```
library(datasets)
library(caTools)
library(party)
library(dplyr)
library(magrittr)
dataset=Advertisement
head(data)
# Encoding the target feature as factor
dataset$Gender = factor(dataset$Gender,levels = c(0, 1))
split = sample.split(dataset$Purchased,SplitRatio = 0.75)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)
# Feature Scaling
#training_set$Age = scale(training_set$Age)
#test_set$Age = scale(test_set$Age)
head(training_set)
model<- ctree(Purchased ~ ., training_set)
plot(model)
```

```
predict_model<-predict(model, test_set)
m_at <- table(test_set$Purchased, predict_model)
m_at
ac_Test <- (sum(diag(m_at)) / sum(m_at))
print(paste('Accuracy for test is found to be', ac_Test))
```

Output :

"Accuracy for test is found to be 0.58”

```
  predict_model
   0.0179640718562874 0.25 0.805555555555556 0.847058823529412
0                  58    2                  1                  3
1                   3    0                 11                 22
```