UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

(FSKTM)

SEMESTER II 2024/2025

DATA MINING

BIT 33603

SECTION 03

LAB ASSIGNMENT 06

TITLE

CLASSIFICATION WITH DECISION TREE IN R

LECTURER'S NAME

DR. ROZITA BINTI ABDUL JALIL

NAME	TUAN KHALIDAH SYAZWANA BINTI TUAN MOHD KASMAWI
MATRIC NUMBER	AI220118
DATE SUBMISSION	April 23, 2025

Topic: Classification with Decision Tree in R

Objectives:

1. Understand how to build a classification model using Decision Tree in R.
2. Evaluate classification model performance using accuracy, precision, recall, and F-score.
3. Visualize a Decision Tree model using rpart.plot.

Duration: 2 hours

Assessment Question:

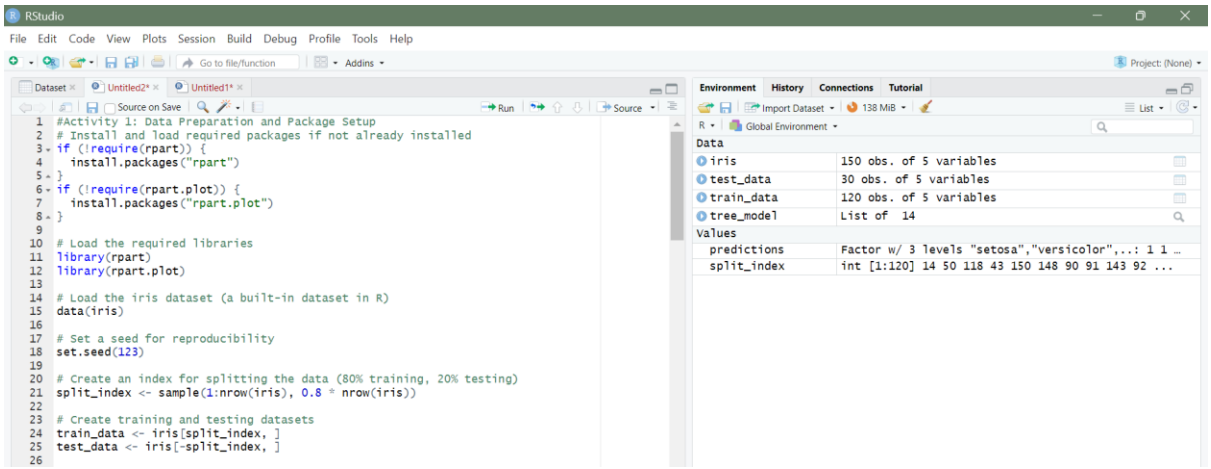
1. Run the provided code in R (Activity 1-4) and understanding the classification using Decision Tree.
2. Submit the visualizations as image/data snapshots for each activity along with a brief explanation of the insights gained.

Activity 1: Data Preparation and Package Setup

Instruction:

In this activity, install and load the necessary libraries (rpart, rpart.plot) and prepare the dataset (iris) by splitting it into training and testing sets.

Source code:



Justification:

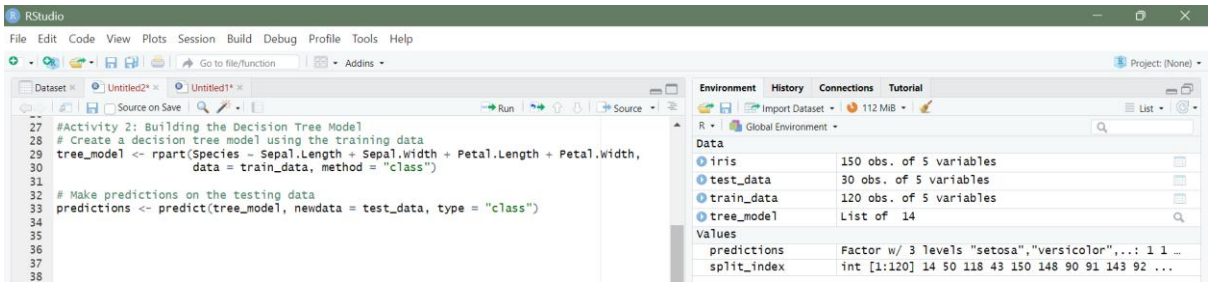
This step gets everything ready. We install the needed libraries (rpart and rpart.plot) and load the Iris dataset, which is already in R. Then, we split the dataset into training (70%) and testing (30%) to train the model on one part and test it on the other. This helps us check how well the model works on new data.

Activity 2: Building the Decision Tree Model

Instruction:

Use the training dataset to build a Decision Tree model using the rpart() function and make predictions on the test dataset.

Source code:



Justification:

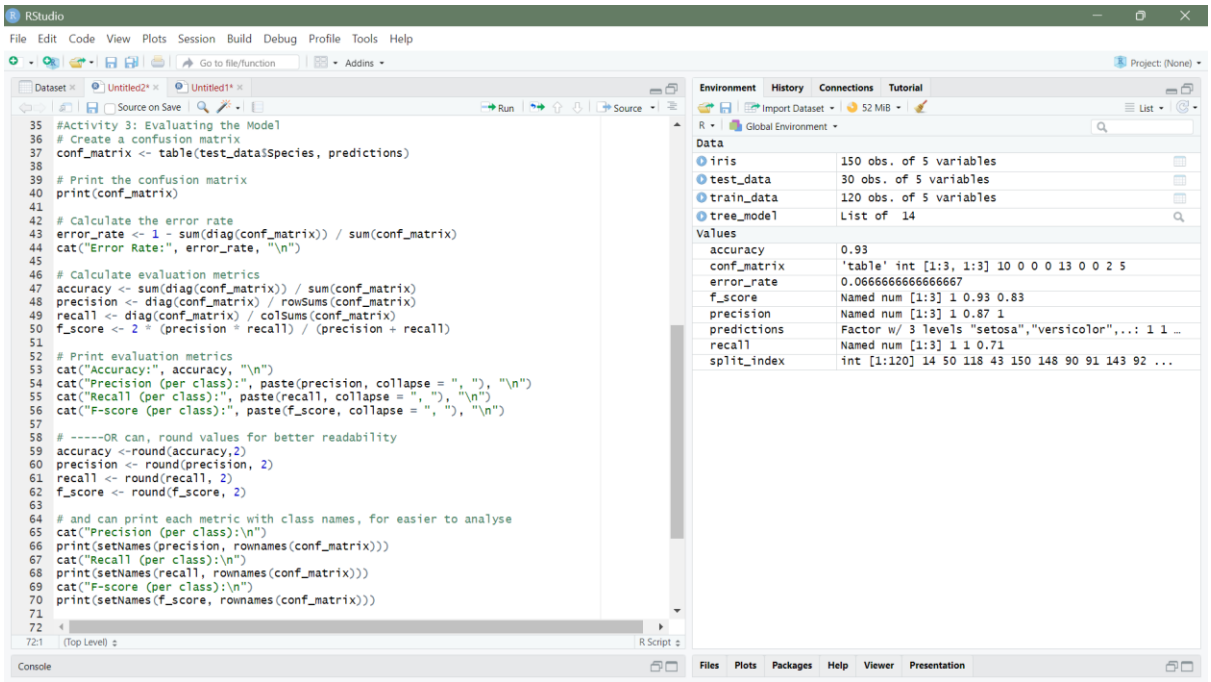
We use the rpart() function to create a Decision Tree model. This model looks at the training data and learns how to decide which species a flower belongs to. After the model is trained, we use it to predict the species of flowers in the testing set.

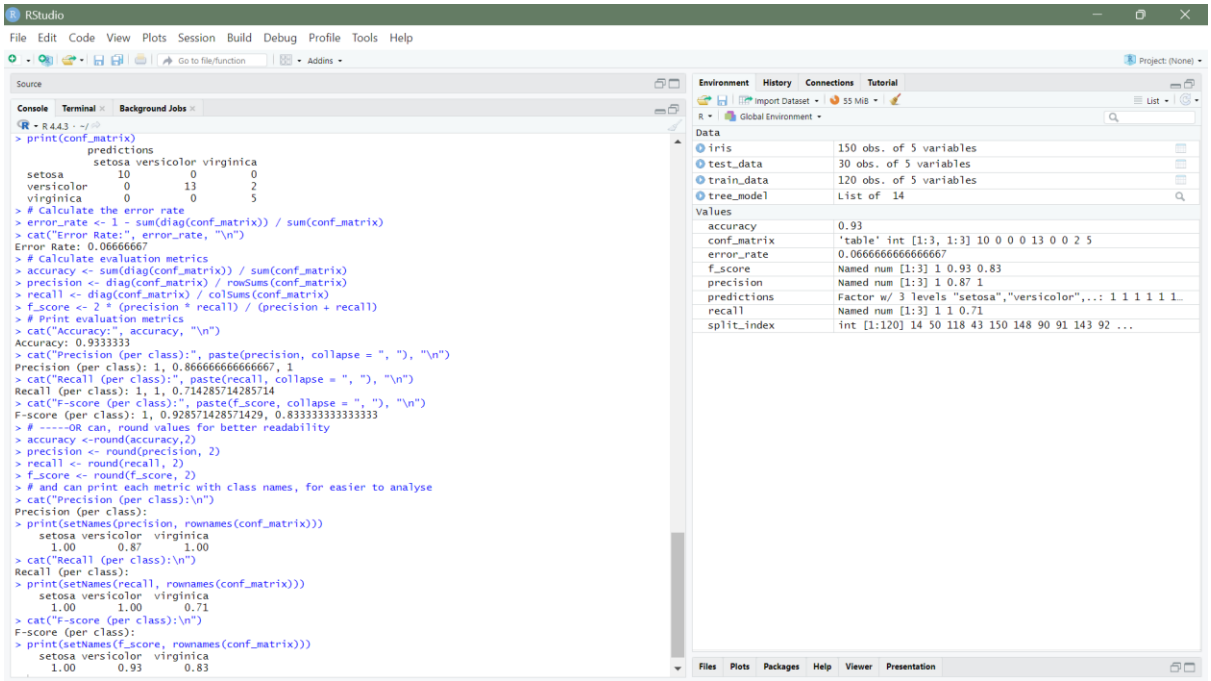
Activity 3: Evaluating the Model

Instruction:

Generate a confusion matrix and compute error rate, accuracy, precision, recall, and F-score to evaluate the performance of the model.

Source code:





Justification:

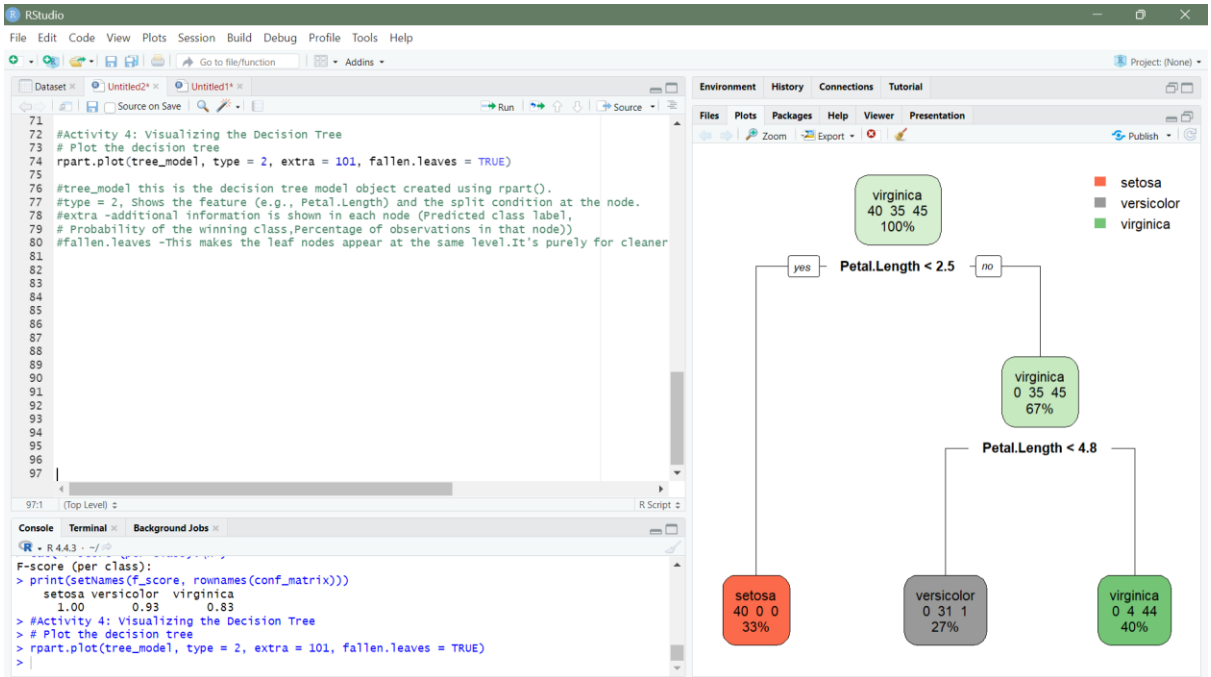
In this step, we check how accurate the model is. We use a confusion matrix to see which predictions were correct or wrong. Then, we calculate accuracy, precision, recall, and F-score to understand the model's strengths and weaknesses for each species.

Activity 4: Visualizing the Decision Tree

Instruction:

Visualize the decision tree model using the rpart.plot() function.

Source code:



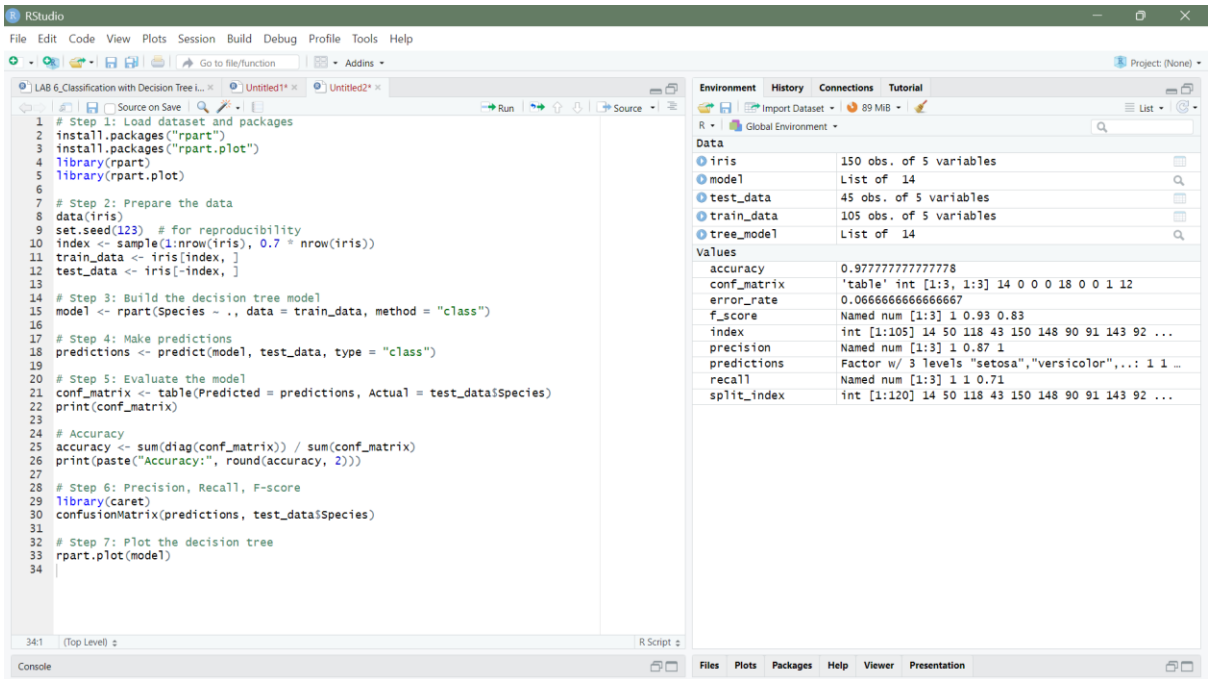
Justification:

We use rpart.plot() to draw the decision tree. This shows us how the model makes decisions. It's a visual way to understand the rules the tree uses to predict flower species based on features like petal length and width.

3. What does the confusion matrix tell you about model performance? Identify any misclassifications.

A confusion matrix shows how many predictions were correct and how many were wrong. It compares the actual class with the predicted class. From the matrix, we can see which species the model predicted correctly and where it made mistakes (misclassifications). If numbers on the diagonal are high and off-diagonal are low, the model is doing well.

4. Using the iris dataset in R, perform a classification task using the Decision Tree algorithm with the following steps:
    1. Split the data into 70% training and 30% testing. Make sure to use set.seed() so your result is reproducible.
    2. Build a Decision Tree classifier using the training set to predict the flower species.
    3. Evaluate your model using the testing set and report:
        1. Confusion matrix
        2. Accuracy
        3. Precision, Recall, and F-score for each class
        4. Plot the decision tree and observe its structure.



```
[1] "Accuracy: 0.98"
> # Step 6: Precision, Recall, F-score
> library(caret)
Loading required package: ggplot2
Loading required package: lattice
> confusionMatrix(predictions, test_data$Species)
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         14          0         0
  versicolor      0         18         1
  virginica       0          0        12

Overall Statistics

               Accuracy : 0.9778
                 95% CI : (0.8823, 0.9994)
    No Information Rate : 0.4
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9662

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            1.0000           0.9231
Specificity                 1.0000            0.9630           1.0000
Pos Pred Value              1.0000            0.9474           1.0000
Neg Pred Value              1.0000            1.0000           0.9697
Prevalence                  0.3111            0.4000           0.2889
Detection Rate              0.3111            0.4000           0.2667
Detection Prevalence        0.3111            0.4222           0.2667
Balanced Accuracy           1.0000            0.9815           0.9615
> # Step 7: Plot the decision tree
> rpart.plot(model)
>
```