**Experiment 5**

**Student Name:** Sahil Kaundal **UID:** 21BCS8197

**Branch:** CSE (Lateral Entry)  **Section/Group:** 616/A

**Semester:** 6th **Date of Performance:** 21/04/2023

**Subject Name:** Data Mining Lab **Subject Code:** 20CSP-376

1. **Aim:**

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

1. **Apparatus / Simulation Used:**

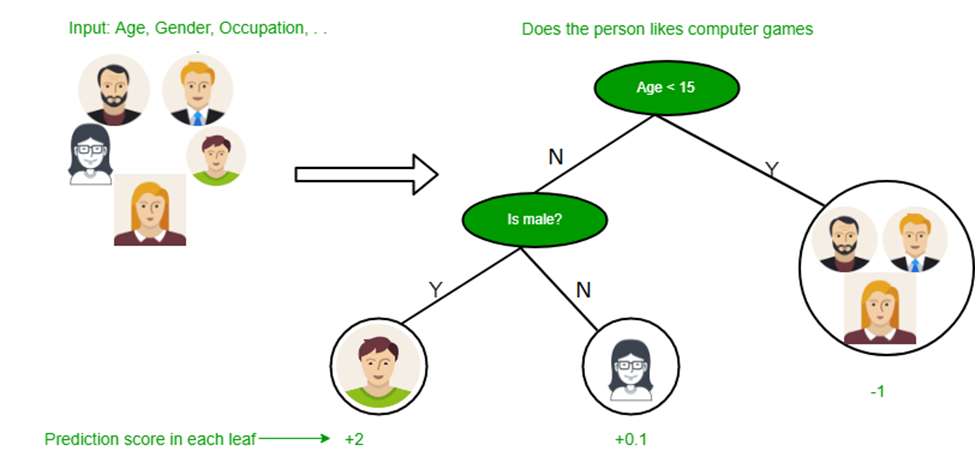
* Windows 7 or above
* R Studio

1. **Objective:**

* Demonstrate the use of WEKA tools.
* Demonstration of Decision Tree induction.

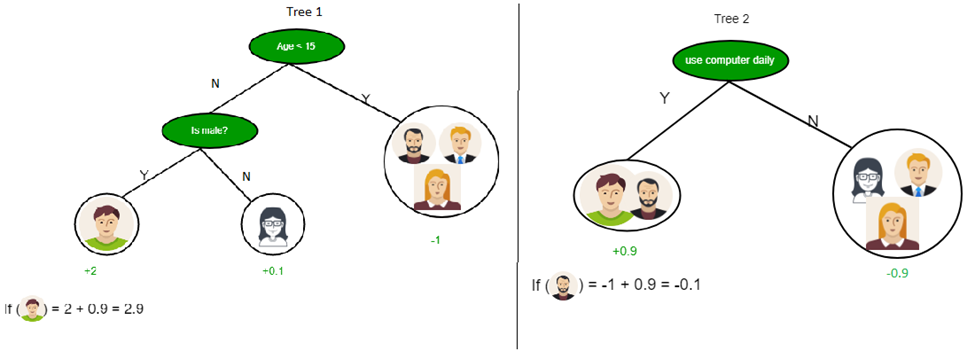
1. **Theory and Output:**

**Decision tree algorithm** falls under the category of [supervised learning](https://www.geeksforgeeks.org/supervised-unsupervised-learning/). They can be used to solve both **regression** and **classification problems**. Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. We can represent any boolean function on discrete attributes using the decision tree.



**Below are some assumptions that we made while using the decision tree:**

* At the beginning, we consider the whole training set as the root.
* Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
* Based on attribute values, records are distributed recursively.
* We use statistical methods for ordering attributes as root or the internal node.



As you can see from the above image the Decision Tree works on the Sum of Product form which is also known as *Disjunctive Normal Form*. In the above image, we are predicting the use of computer in the daily life of people. In the Decision Tree, the major challenge is the identification of the attribute for the root node at each level. This process is known as attribute selection. We have two popular attribute selection measures:

1. Information Gain
2. Gini Index

**1. Information Gain** When we use a node in a decision tree to partition the training instances into smaller subsets the entropy changes. Information gain is a measure of this change in entropy. ***Definition***: Suppose S is a set of instances, A is an attribute, Sv is the subset of S with A = v, and Values (A) is the set of all possible values of A, then **Entropy** Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples. The higher the entropy more the information content. ***Definition***: Suppose S is a set of instances, A is an attribute, Sv is the subset of S with A = v, and Values (A) is the set of all possible values of A, then

Example:

For the set X = {a,a,a,b,b,b,b,b}

Total instances: 8

Instances of b: 5

Instances of a: 3

Entropy H(X)  = -\left [ \left ( \frac{3}{8} \right )log\_{2}\frac{3}{8} + \left ( \frac{5}{8} \right )log\_{2}\frac{5}{8} \right ]

 = -[0.375 \* (-1.415) + 0.625 \* (-0.678)]

= -(-0.53-0.424)

= 0.954

**Building Decision Tree using Information Gain** **The essentials:**

* + Start with all training instances associated with the root node
  + Use info gain to choose which attribute to label each node with
  + *Note:* No root-to-leaf path should contain the same discrete attribute twice
  + Recursively construct each subtree on the subset of training instances that would be classified down that path in the tree.
  + If all positive or all negative training instances remain, the label that node “yes” or “no” accordingly
  + If no attributes remain, label with a majority vote of training instances left at that node
  + If no instances remain, label with a majority vote of the parent’s training instances.

1. **Code:**

library(RWeka)

library(partykit)

library(caTools)

iris\_data = iris

str(iris\_data)

summary(iris\_data)

spl = sample.split(iris\_data, SplitRatio = 0.7)

dataTrain = subset(iris\_data, spl==TRUE)

dataTest = subset(iris\_data, spl==FALSE)

m1 <- J48(Species~., dataTrain)

summary(m1)

dataTestPred <- predict(m1, newdata = dataTest)

table\_matrix <- table(dataTest$Species, dataTestPred)

print(table\_matrix)

accuracy\_Test <- sum(diag(table\_matrix)) / sum(table\_matrix)

cat("Test Accuracy is: ", accuracy\_Test)

# Initate PDF File

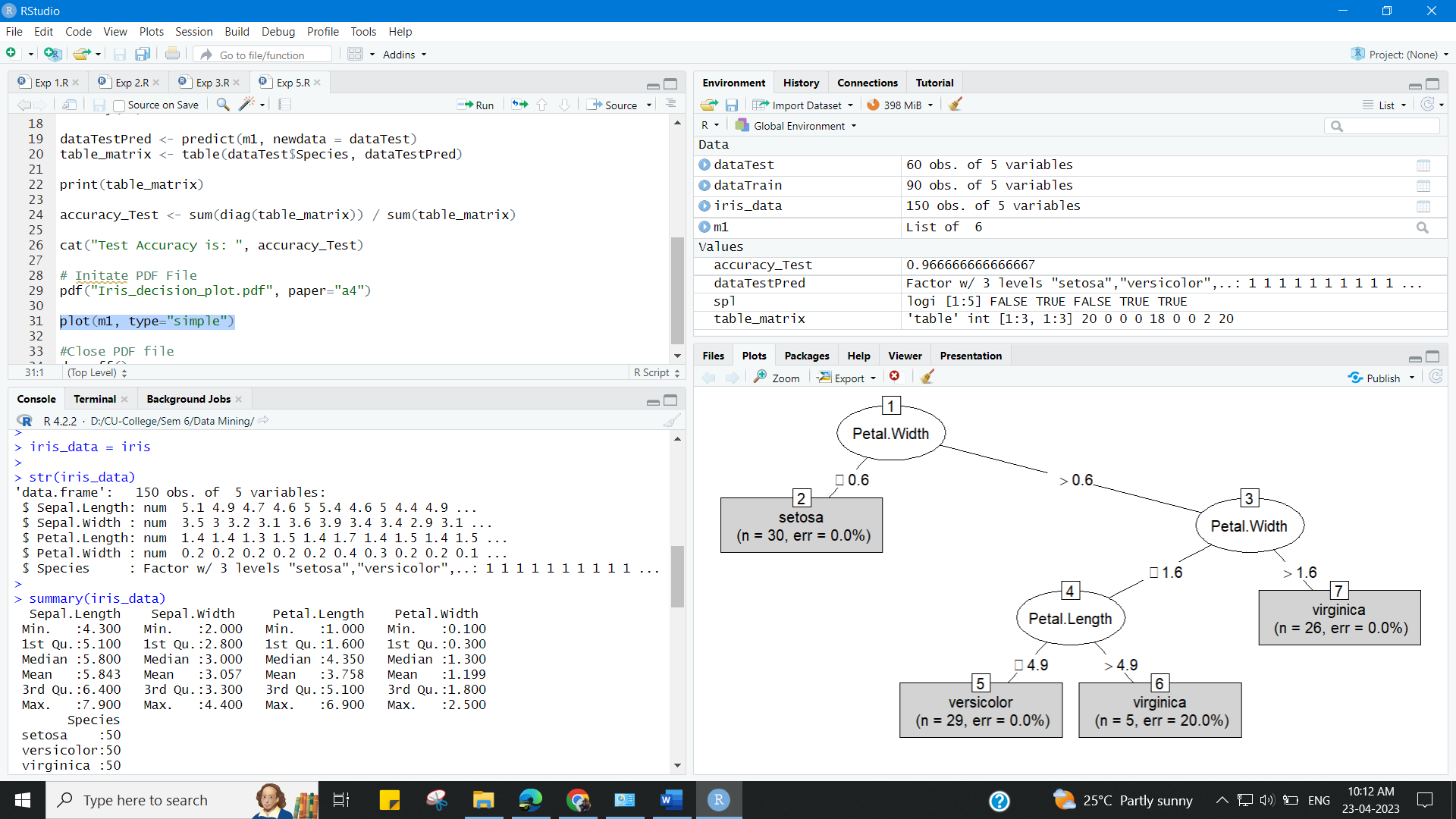
pdf("Iris\_decision\_plot.pdf", paper="a4")

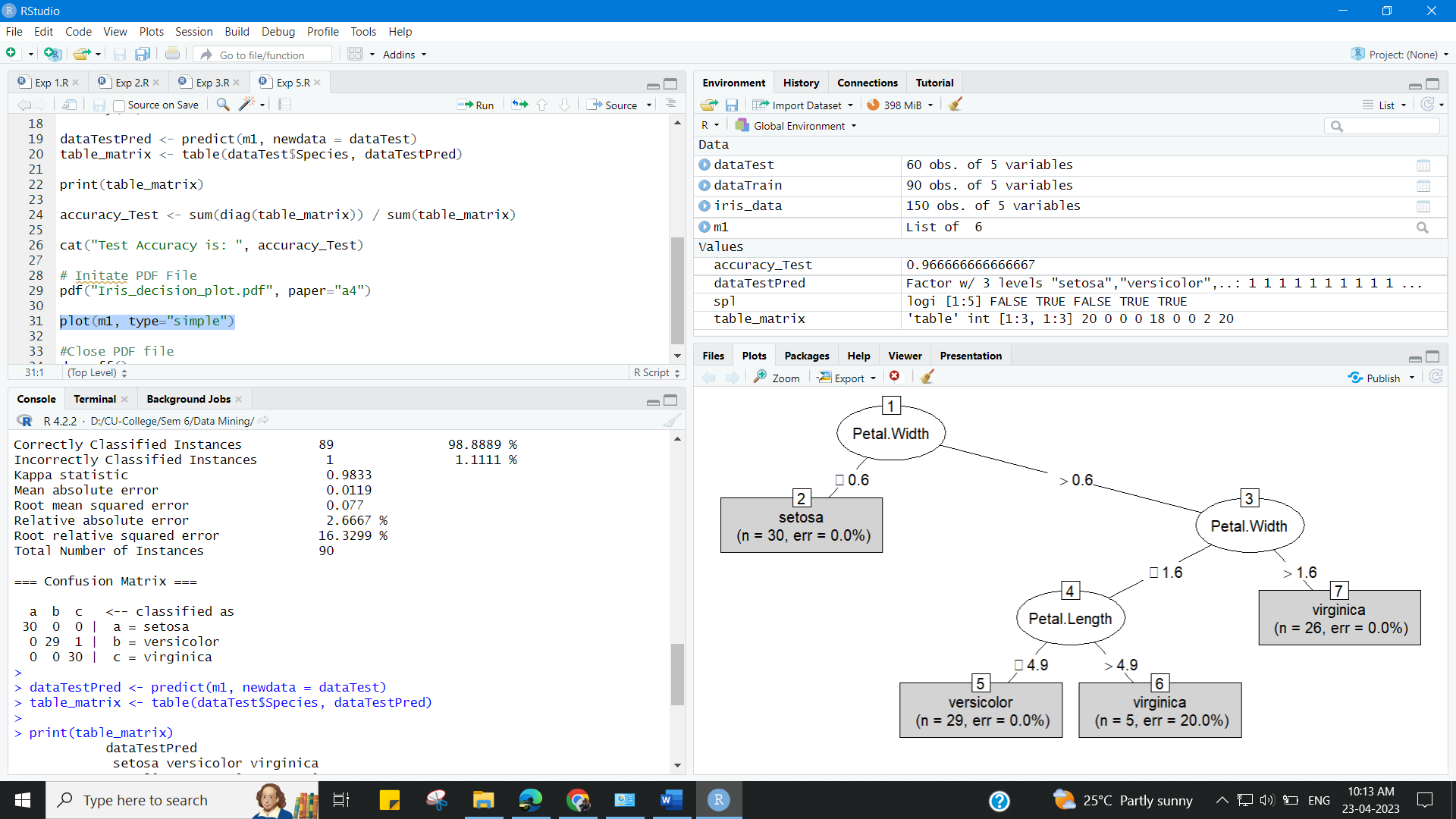
plot(m1, type="simple")

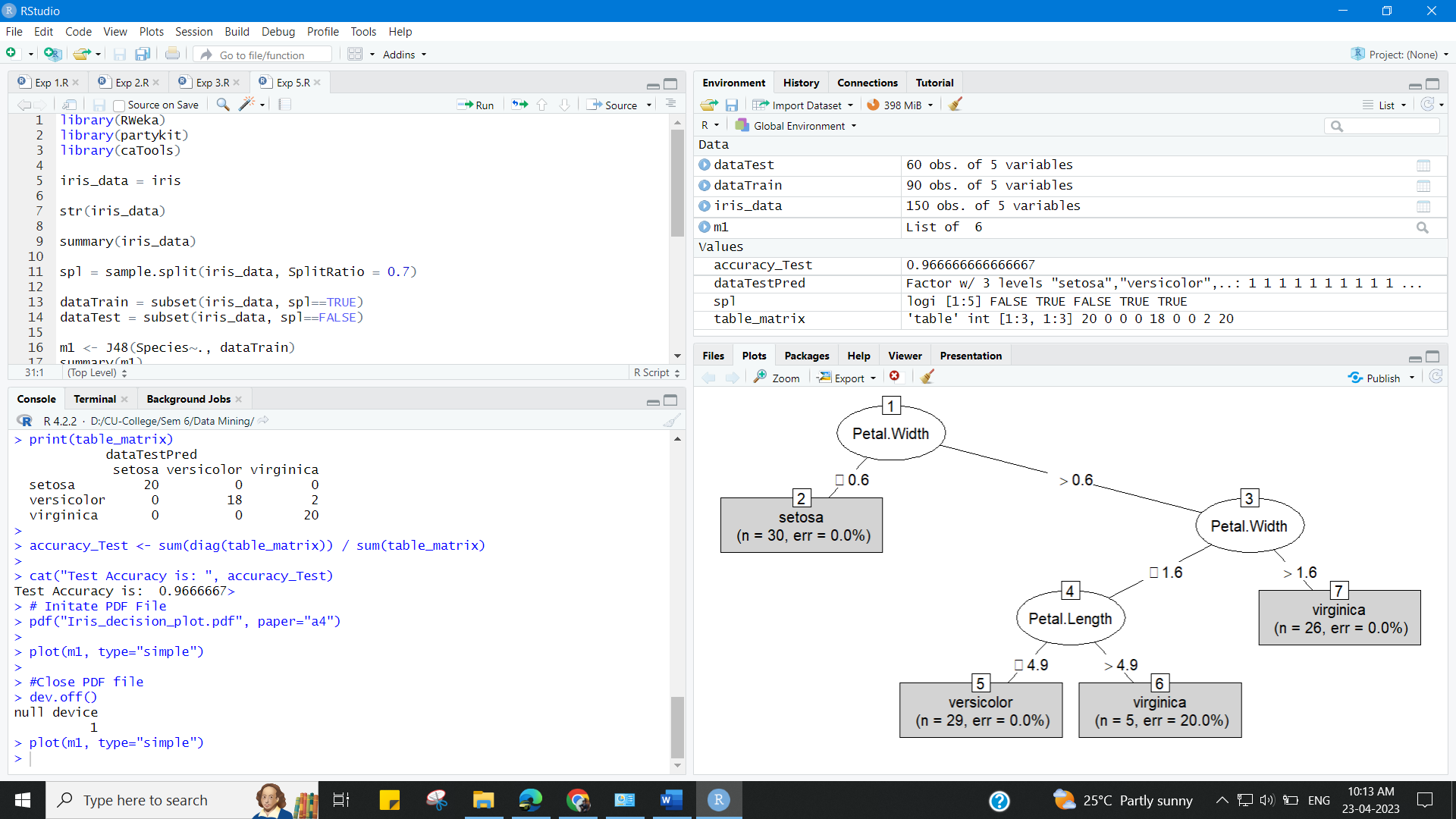
#Close PDF file

dev.off()

1. **Output:**

****

****

****

**Learning outcomes (What I have learnt):**

* Demonstrate the use of WEKA tools.
* Demonstration of Decision Tree induction.