

**Ex.No:8****Implement SVM/Decision tree classification techniques****AIM :**

to implement SVM and Decision tree classification techniques using R language.

**a) SVM IN R**

```
# Install and load the e1071 package (if not already installed)
install.packages("e1071") library(e1071)

# Load the iris dataset data(iris)

# Inspect the first few rows of the dataset head(iris)

# Split the data into training (70%) and testing (30%) sets set.seed(123)
# For reproducibility
sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris)) train_data
<- iris[sample_indices, ]
test_data <- iris[-sample_indices, ]

# Fit the SVM model svm_model <- svm(Species ~ ., data =
train_data, kernel = "radial")

# Print the summary of the model
summary(svm_model)

# Predict the test set predictions <- predict(svm_model,
newdata = test_data)

# Evaluate the model's performance
confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
print(confusion_matrix)

# Calculate accuracy accuracy <- sum(diag(confusion_matrix)) /
sum(confusion_matrix) cat("Accuracy:", accuracy * 100, "%\n")
```

The screenshot shows two windows from an R environment. The left window is the R Console, displaying the output of the code. The right window is the R Editor, showing the source code for training and evaluating an SVM model on the Iris dataset.

**R Console Output:**

```
Number of Classes: 3
Levels:
setosa versicolor virginica

>
> # Predict the test set
> predictions <- predict(svm_model, newdata = test_data)
>
> # Evaluate the model's performance
> confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
> print(confusion_matrix)
      Actual
Predicted setosa versicolor virginica
setosa     14         0         0
versicolor  0         17         0
virginica   0          1        13

>
> # Calculate accuracy
> accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
> cat("Accuracy:", accuracy * 100, "%\n")
Accuracy: 97.77778 %
> |
```

**R Editor Code:**

```
# C:\Users\mahuf\OneDrive\Documents\svm1.r - R Editor
# Install and load the e1071 package (if not already installed)
install.packages("e1071")
library(e1071)

# Load the iris dataset
data(iris)

# Inspect the first few rows of the dataset
head(iris)

# Split the data into training (70%) and testing (30%) sets
set.seed(123) # For reproducibility
sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
train_data <- iris[sample_indices, ]
test_data <- iris[-sample_indices, ]

# Fit the SVM model
svm_model <- svm(Species ~ ., data = train_data, kernel = "radial")

# Print the summary of the model
summary(svm_model)

# Predict the test set
predictions <- predict(svm_model, newdata = test_data)

# Evaluate the model's performance
```

## **b) Decision tree in R**

# Install and load the rpart package (if not already installed)  
install.packages("rpart") library(rpart)

# Load the iris dataset data(iris)

# Split the data into training (70%) and testing (30%) sets set.seed(123)  
# For reproducibility  
sample\_indices <- sample(1:nrow(iris), 0.7 \* nrow(iris)) train\_data  
<- iris[sample\_indices, ]  
test\_data <- iris[-sample\_indices, ]

# Fit the Decision Tree model tree\_model <- rpart(Species ~ ., data  
= train\_data, method = "class")

# Print the summary of the model  
summary(tree\_model)

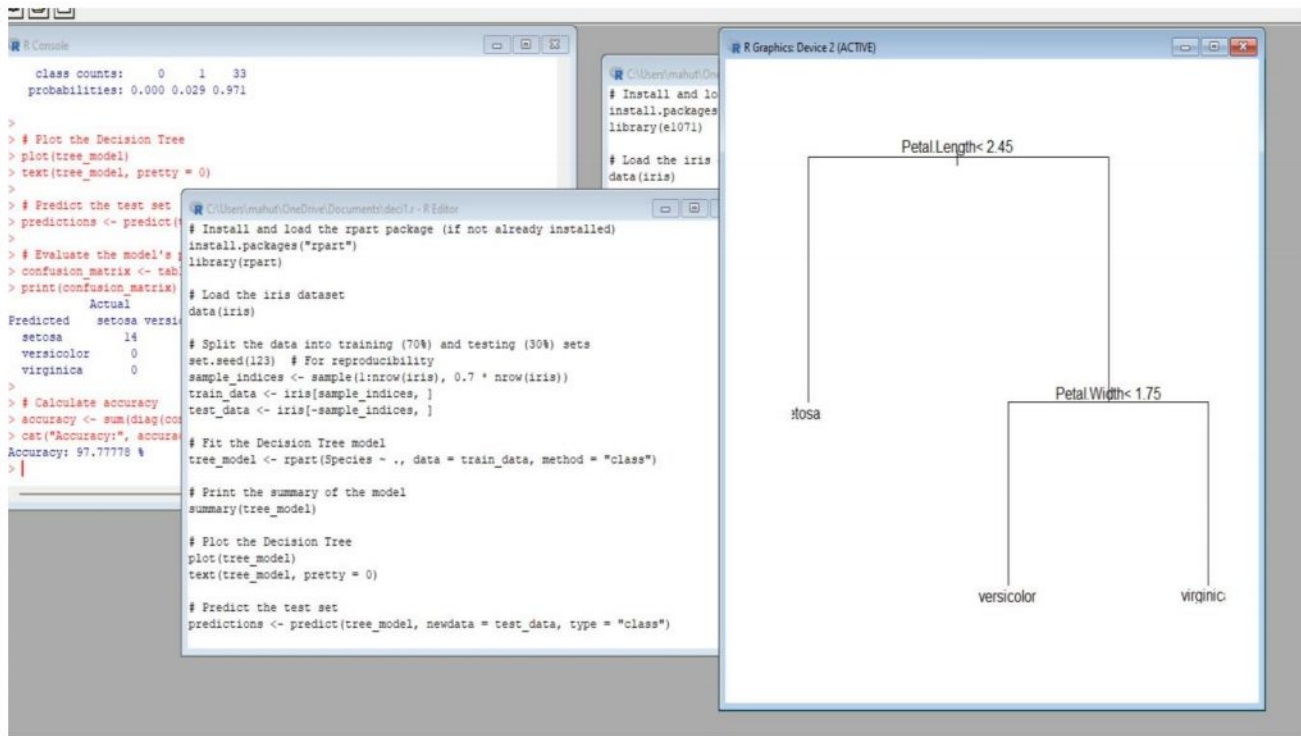
# Plot the Decision Tree  
plot(tree\_model) text(tree\_model,  
pretty = 0)

# Predict the test set predictions <- predict(tree\_model, newdata =  
test\_data, type = "class")

# Evaluate the model's performance  
confusion\_matrix <- table(Predicted = predictions, Actual = test\_data\$Species)  
print(confusion\_matrix)

# Calculate accuracy

```
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix) cat("Accuracy:",  
accuracy * 100, "%\n")
```



## RESULT :

Thus to implement SVM and Decision tree using R language is successfully completed.