Implement SVM/Decision tree classification techniques

<u>a)</u> SVM IN R

CODE:

```
# 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"
```

OUTPUT:

File Edit Code View Plots Session Build Debug Profile Tools Help

ex7b.R ex8svm.R ex9ahier.R ex9bkmeans.R ex10barchart.R ex10Scatter.R ex10boxpl »

Source on Save | Run | Source

```r
 5  data(iris)
 6  # Inspect the first few rows of the dataset
 7  head(iris)
 8  # Split the data into training (70%) and testing (30%) sets
 9  set.seed(123)  # For reproducibility
10  sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
11  train_data <- iris[sample_indices, ]
12  test_data <- iris[-sample_indices, ]
13  # Fit the SVM model
14  svm_model <- svm(Species ~ ., data = train_data, kernel = "radial")
15  # Print the summary of the model
16  summary(svm_model)
17  # Predict the test set
18  predictions <- predict(svm_model, newdata = test_data)
19  # Evaluate the model's performance
20  confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
21  print(confusion_matrix)
22  # Calculate accuracy
23  accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
24  cat("Accuracy:", accuracy * 100, "%\n")
25  |
26
```

25:1 (Top Level) R Script

Console | Terminal | Background Jobs

R 4.4.1 · ~/

```
(as  Tib  is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/e1071_1.7-14.zip'
Content type 'application/zip' length 671816 bytes (656 KB)
downloaded 656 KB

package 'e1071' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\asus\AppData\Local\Temp\RtmpumGnq5\downloaded_packages
            Actual
Predicted    setosa versicolor virginica
  setosa        14         0         0
  versicolor     0        17         0
  virginica      0         1        13
Accuracy: 97.77778 %
>
```

```
package 'e1071' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\asus\AppData\Local\Temp\RtmpumGnq5\downloaded_packages
            Actual
Predicted    setosa versicolor virginica
  setosa        14         0         0
  versicolor     0        17         0
  virginica      0         1        13
Accuracy: 97.77778 %
```

b) DECISION TREE IN R

CODE:

```
# 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")
```
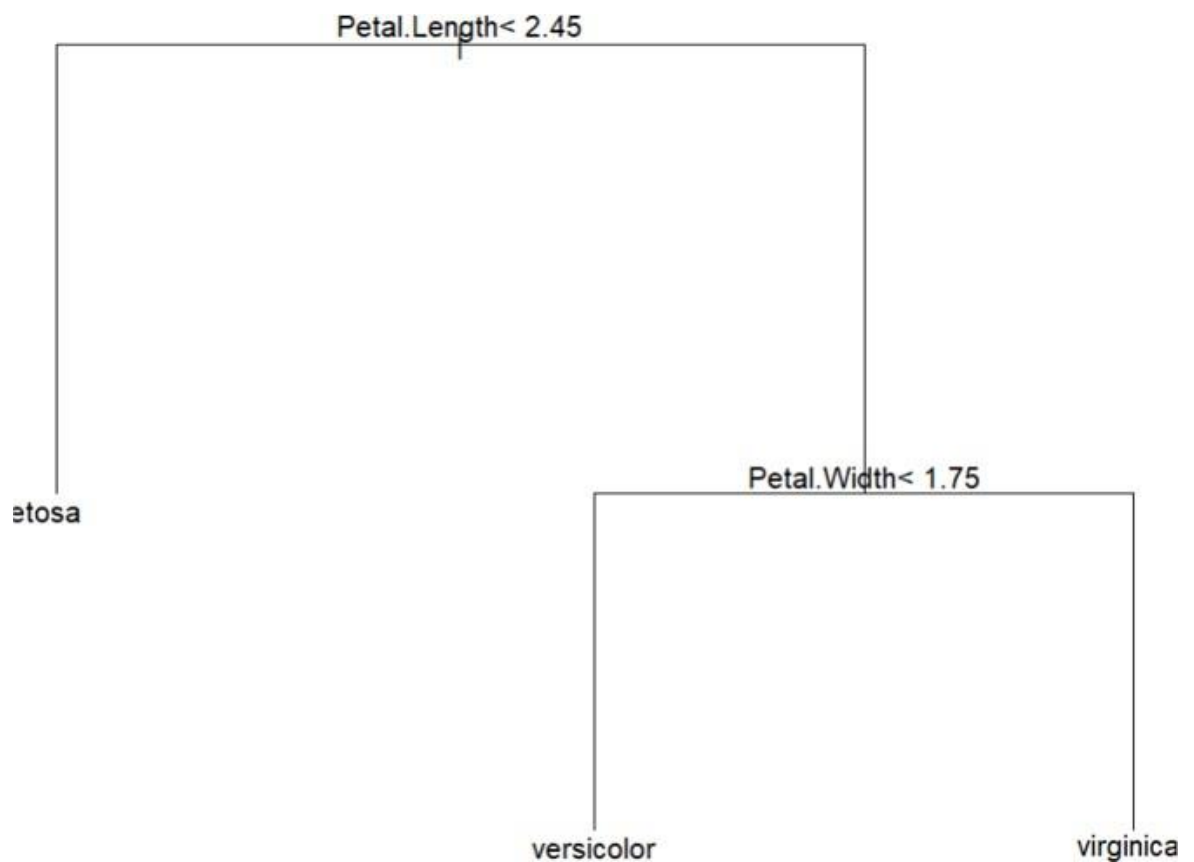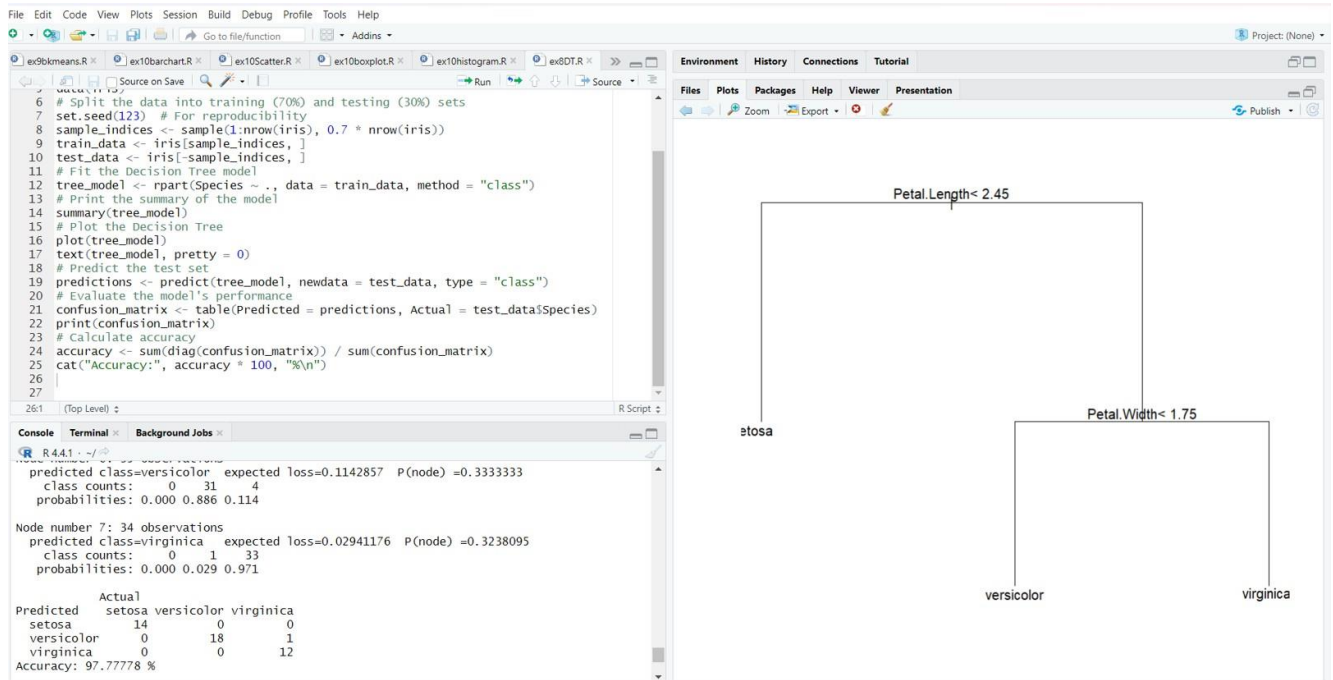
OUTPUT:

RESULT:

SVM and Decision tree classification techniques are implemented Successfully.