

Exp.No: 8**IMPLEMENT SVM/DECISION TREE CLASSIFICATION TECHNIQUES****AIM:**

To write an R code to implement SVM/decision tree classification techniques.

PROCEDURE:

1. Install and load the required packages (e1071 for SVM and rpart for Decision Tree) and load the iris dataset.
2. Split the dataset into training (70%) and testing (30%) sets using a reproducible random sampling method.
3. Fit the SVM model with a radial kernel using the training data, print the model summary, and evaluate its performance using a confusion matrix and accuracy calculation.
4. Fit the Decision Tree model using the rpart function with the training data, print the model summary, visualize the tree, and evaluate its performance using a confusion matrix and accuracy calculation.
5. Predict the test set results for both SVM and Decision Tree models and assess their accuracy.

PROGRAM CODE:**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")
```

OUTPUT:

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Contains R code for training an SVM model with a radial kernel on the iris dataset.
- Console:** Displays the output of the SVM training, including parameters (C-classification, radial kernel, cost = 1), number of support vectors (45), and number of classes (3).
- Environment:** Lists objects in the global environment: data (7 obs. of 2 variables), iris (150 obs. of 5 variables), linear_model (List of 12), logistic_model (List of 30), mtcars (32 obs. of 11 variables), svm_model (List of 31), test_data (45 obs. of 5 variables), and train_data (105 obs. of 5 variables).
- Values:** Shows the accuracy (0.977777777777778), confusion_matrix (table), heights (num), predicted_probs (Named num), predictions (Factor w/ 3 levels), sample_indices (int), and weights (num).
- System Library:** Lists installed packages including base, BH, BiocManager, BiocParallel, BiocVersion, boot, class, cli, and cluster.

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)

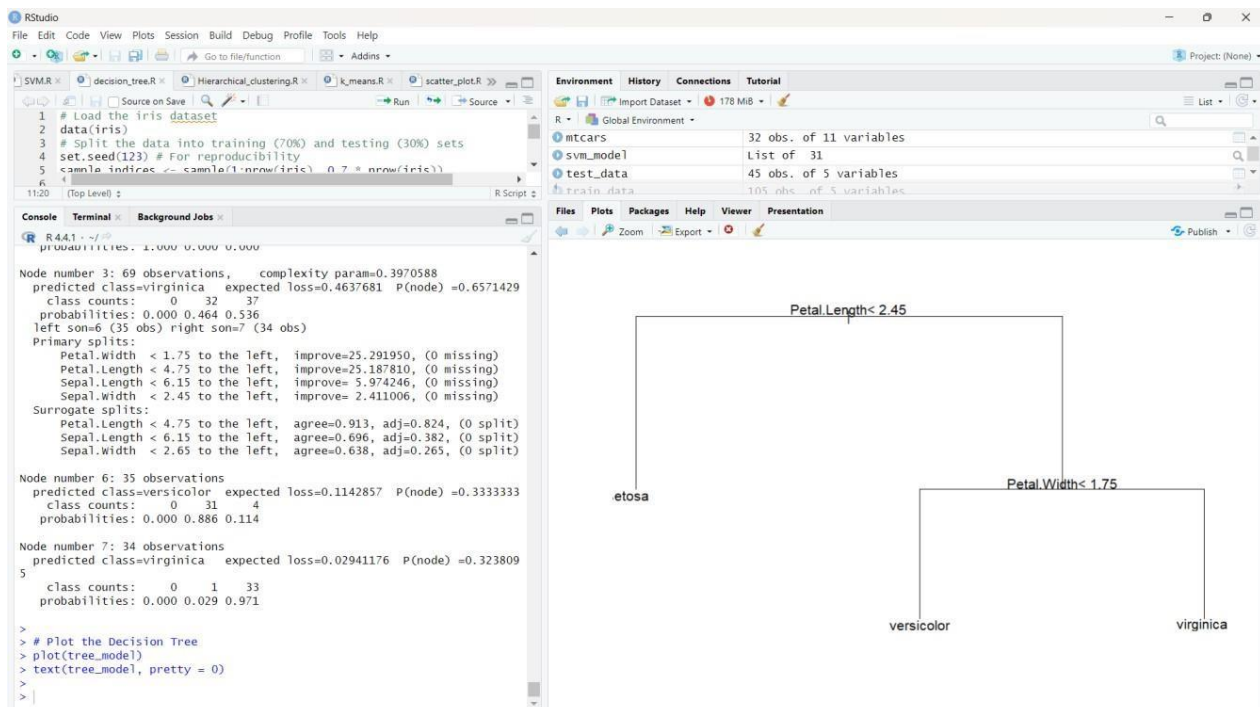
# Predict the test set predictions <- predict(tree_model, newdata =
test_data, type = "class")

# 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")

```

OUTPUT:



RESULT:

Thus the R program to implement SVM/decision tree classification techniques has been executed and verified successfully.