

Exp. No : 8

Implement SVM/Decision tree classification techniques

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, ]

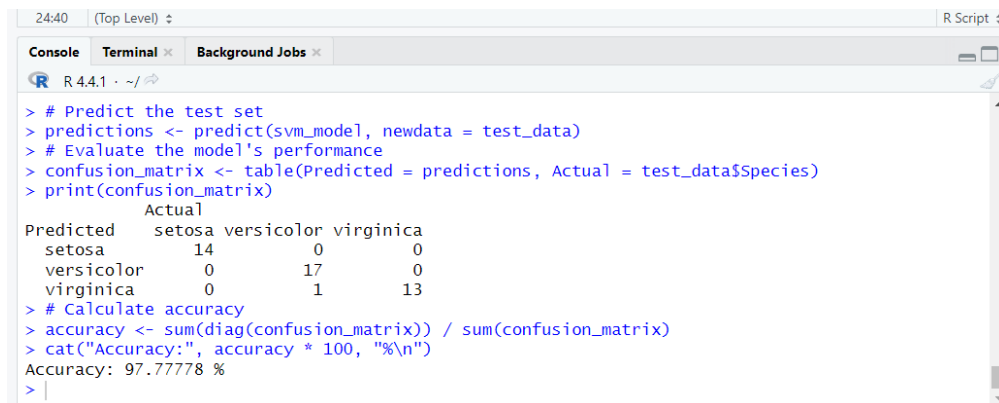
# Fit the SVM model
svm_model <- svm(Species ~ ., data = train_data, kernel = "radial")

# 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 :


```

24:40 (Top Level) R Script
Console Terminal Background Jobs
R 4.4.1 ~ /
> # 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 %
> |

```

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)
```

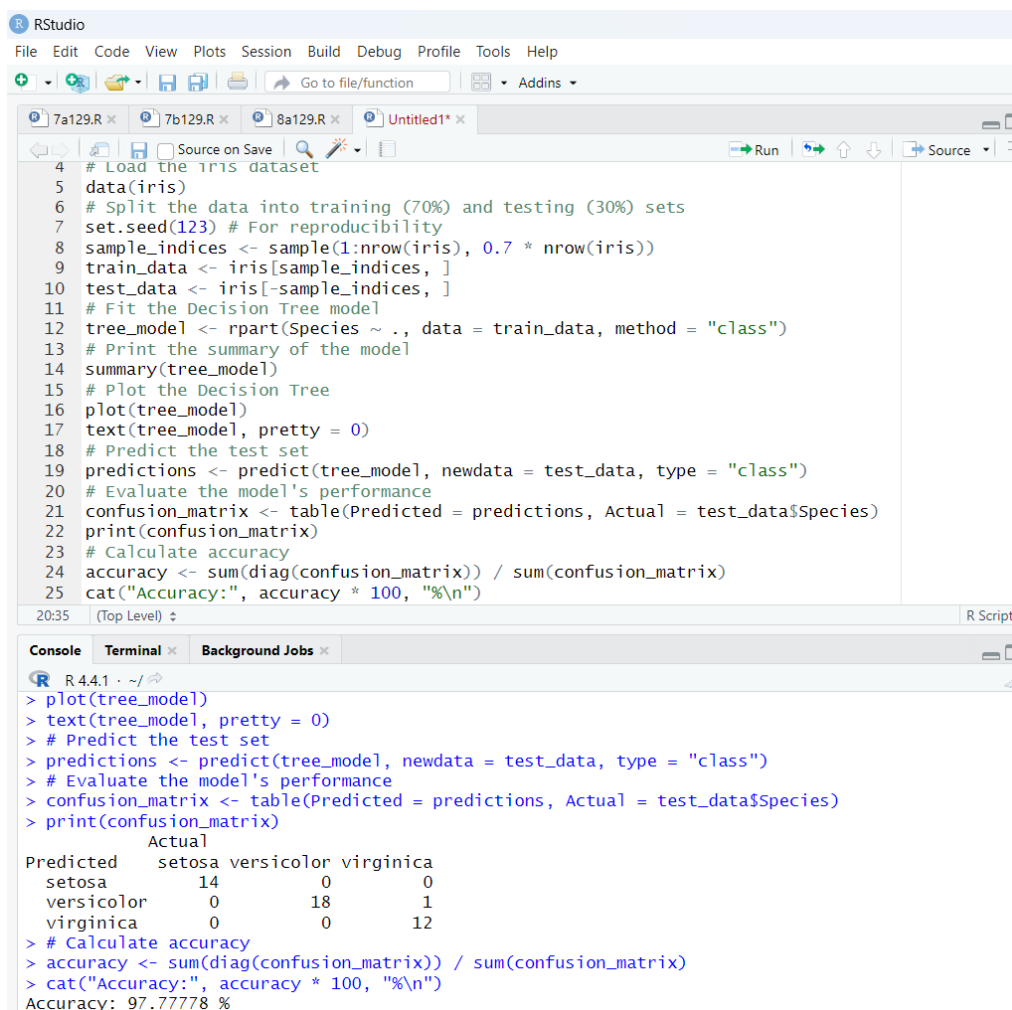
```
print(confusion_matrix)
```

```
# Calculate accuracy
```

```
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
```

```
cat("Accuracy:", accuracy * 100, "%\n")
```

Output :



The screenshot shows the RStudio interface. The script editor contains the following R code:

```

4 # Load the iris dataset
5 data(iris)
6 # Split the data into training (70%) and testing (30%) sets
7 set.seed(123) # For reproducibility
8 sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
9 train_data <- iris[sample_indices, ]
10 test_data <- iris[-sample_indices, ]
11 # Fit the Decision Tree model
12 tree_model <- rpart(Species ~ ., data = train_data, method = "class")
13 # Print the summary of the model
14 summary(tree_model)
15 # Plot the Decision Tree
16 plot(tree_model)
17 text(tree_model, pretty = 0)
18 # Predict the test set
19 predictions <- predict(tree_model, newdata = test_data, type = "class")
20 # Evaluate the model's performance
21 confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
22 print(confusion_matrix)
23 # Calculate accuracy
24 accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
25 cat("Accuracy:", accuracy * 100, "%\n")

```

The console output shows the execution of the code:

```

> 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)
      Actual
Predicted setosa versicolor virginica
setosa     14          0           0
versicolor  0          18          1
virginica   0           0          12
> # Calculate accuracy
> accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
> cat("Accuracy:", accuracy * 100, "%\n")
Accuracy: 97.77778 %

```

