

ROLL NO : 210701268

Exp8:

Implement SVM/Decision tree classification techniques

Aim: To implement SVM/Decision Tree classification techniques in RStudio using R language.

PROCEDURE:

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")
```

```

7 sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
8 train_data <- iris[sample_indices, ]
9 test_data <- iris[-sample_indices, ]
10 # Fit the SVM model
11 svm_model <- svm(Species ~ ., data = train_data, kernel = "radial")
12 # Print the summary of the model
13 summary(svm_model)
14 # Predict the test set
15 predictions <- predict(svm_model, newdata = test_data)
16 # Evaluate the model's performance
17 confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
18 print(confusion_matrix)
19 # Calculate accuracy
20 accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
21 cat("Accuracy:", accuracy * 100, "%\n")
22

```

Console Output:

```

> accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
Error: object 'confusion_matrix' not found
> cat("Accuracy:", accuracy * 100, "%\n")
Error: object 'accuracy' not found
> source("~/active-rstudio-document")

```

Global Environment:

Object	Value
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
train_data	105 obs. of 5 variables

Values:

```

accuracy          0.977777777777778
confusion_matrix  'table' int [1:3, 1:3] 14 0 0 0 17 1 0 0 13
heights          num [1:7] 150 160 165 170 175 180 185
predicted_probs  Named num [1:32] 0.461 0.461 0.598 0.492 0.297...
predictions      Factor w/ 3 levels "setosa","versicolor",...: 1...
sample_indices   int [1:105] 14 50 118 43 150 148 90 91 143 92 ...
weights          num [1:7] 55 60 62 68 70 75 80

```

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)

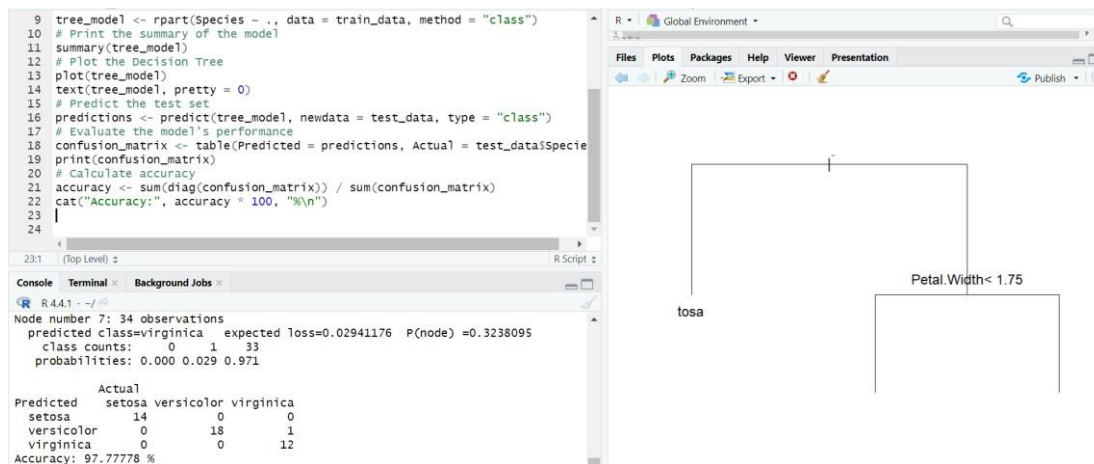
Predict the test set predictions <- predict(tree_model, newdata = test_data,
type = "class")

```
# 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 SVM and Decision Tree techniques are implemented in RStudio using R language.

