

EX.NO: 8

REGISTER NO: 210701315

DATE:

IMPLEMENT SVM/DECISION TREE CLASSIFICATION TECHNIQUES

AIM:

To implement SVM/Decision tree classification techniques.

PROGRAM CODE:

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")
```

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:

SVM in R:

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

```

1 install.packages("e1071")
2 library(e1071)
3 # Load the iris dataset
4 data(iris)
5 # Inspect the first few rows of the dataset
6 head(iris)
7 # Split the data into training (70%) and testing (30%) sets
8 set.seed(123) # For reproducibility
9 sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
10 train_data <- iris[sample_indices, ]
11 test_data <- iris[-sample_indices, ]
12 # Fit the SVM model
13 svm_model <- svm(Species ~ ., data = train_data, kernel = "radial")
14 # Print the summary of the model
15 summary(svm_model)
16 # Predict the test set
17 predictions <- predict(svm_model, newdata = test_data)
18 # Evaluate the model's performance
19 confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
20 print(confusion_matrix)
21 # Calculate accuracy
22 accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
23 cat("Accuracy:", accuracy * 100, "%\n")

```

The console output shows:

```

R 4.4.1 ~ ./
SVM-kernel: radial
cost: 1

Number of Support Vectors: 45
( 7 18 20 )

Number of Classes: 3
Levels:
setosa versicolor virginica

```

The Environment pane on the right shows the following objects:

- 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
- 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 ...
- sample_indices: int [1:105] 14 50 118 43 150 148 90 91 143 92 ...

The User Library pane shows the following packages:

- cli: 3.6.3
- corplot: 0.94
- cpl11: 0.5.0
- e1071: 1.7-14
- glue: 1.7.0
- igraph: 2.0.3
- lifecycle: 1.0.4
- magrittr: 2.0.3
- mvtnorm: 1.3-1
- pkgconfig: 2.0.3
- proxy: 0.4-27
- RColorBrewer: 1.1-3
- riang: 1.1-4
- vctrs: 0.6.5

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

```

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

```

The console output shows:

```

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 %
> |

```

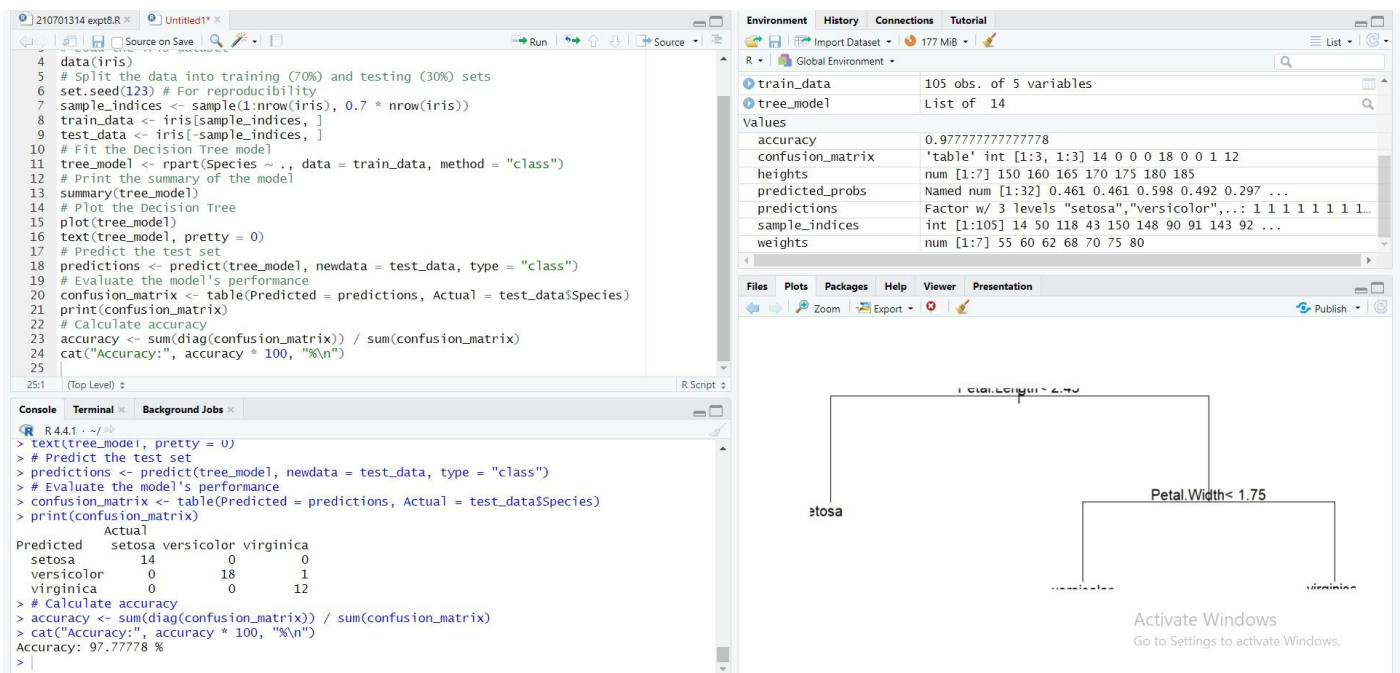
The Environment pane on the right shows the following objects:

- svm_model: List of 31
- test_data: 45 obs. of 5 variables
- train_data: 105 obs. of 5 variables
- 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 1 1 1 1 1 ...
- sample_indices: int [1:105] 14 50 118 43 150 148 90 91 143 92 ...

The User Library pane shows the following packages:

- cli: 3.6.3
- corplot: 0.94
- cpl11: 0.5.0
- e1071: 1.7-14
- glue: 1.7.0
- igraph: 2.0.3
- lifecycle: 1.0.4
- magrittr: 2.0.3
- mvtnorm: 1.3-1
- pkgconfig: 2.0.3
- proxy: 0.4-27
- RColorBrewer: 1.1-3
- riang: 1.1-4
- vctrs: 0.6.5

Decision Tree in R:



RESULT:

Thus the implementation of SVM/Decision tree classification techniques done successfully.