Exp.No: 8

Implement SVM/Decision tree classification techniques

AIM:

To Implement SVM/Decision tree classification techniques using R.

PROCEDURE:

- Load the dataset from sources such as CSV files or databases using suitable libraries.
- Perform data cleaning and preprocessing, including addressing missing values and encoding categorical variables.
- Divide the dataset into training and testing sets to effectively assess model performance.
- Normalize or standardize the features, particularly for SVM, to ensure uniform scaling across inputs.
- Select the appropriate model: use SVM for margin-based classification or Decision Tree for rule-based classification tasks.
- Train the model on the training data using the fit method.
- Generate predictions on the testing data with the predict method.
- Evaluate the model's performance using metrics such as accuracy, confusion matrix, precision, and recall.
- Visualize the outcomes with relevant plots, like decision boundaries for SVM or tree structures for Decision Trees.
- Tune the model by adjusting hyperparameters, such as C for SVM or max_depth for Decision Trees.

PROGRAM:

SVM.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")
```

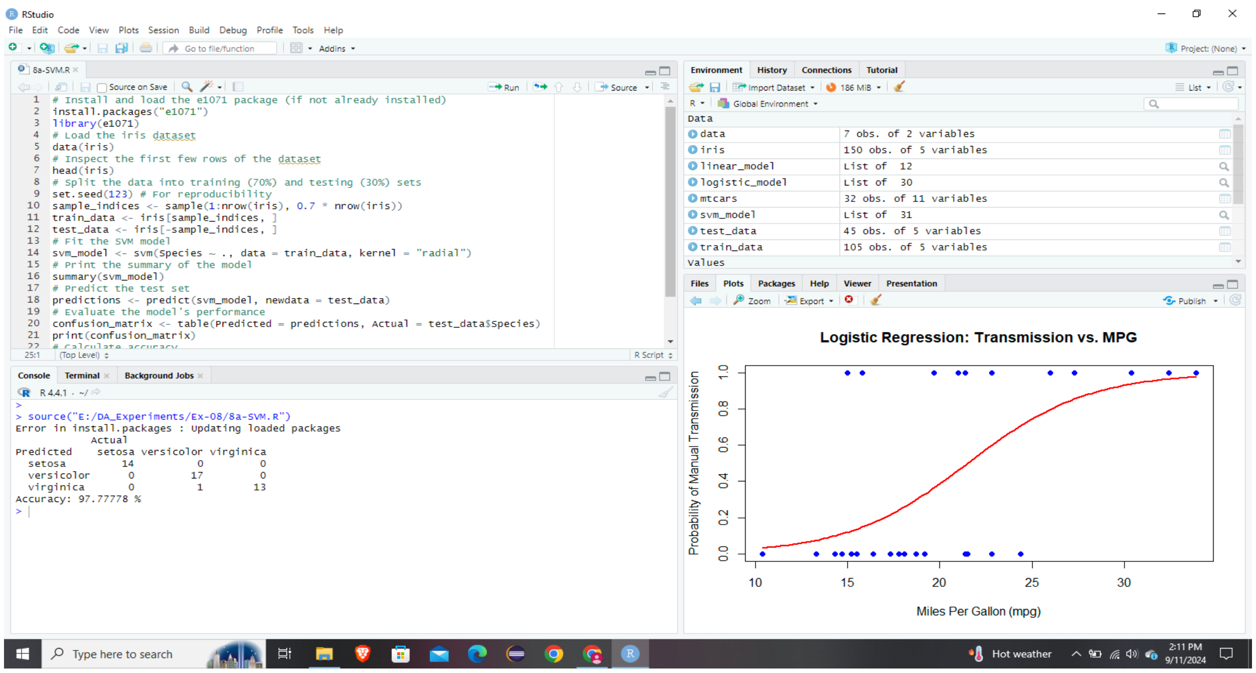
Decision Tree.R:

```
# 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:



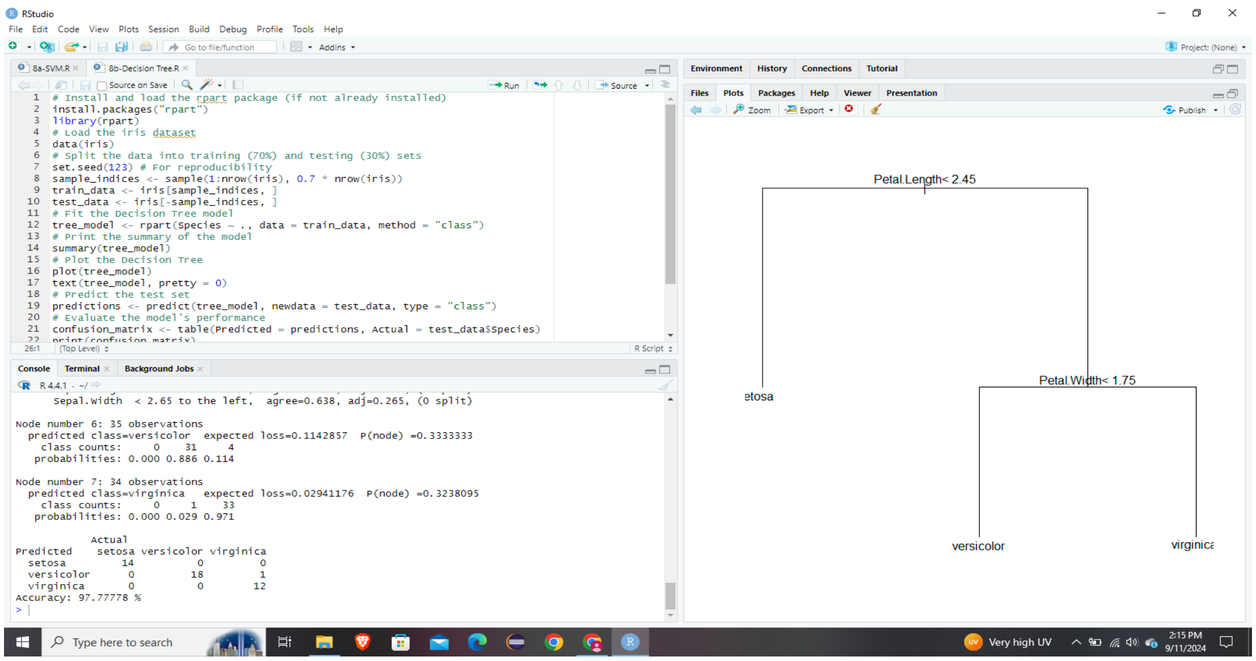
```
# Install and load the e1071 package (if not already installed)
install.packages("e1071")
library(e1071)
# Load the iris dataset
data(iris)
# Inspect the first few rows of the dataset
head(iris)
# Split the data into training (70%) and testing (30%) sets
set.seed(123) # For reproducibility
sample_indices <- 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
```

```
>
> source("E:/DA_Experiments/Ex-08/8a-SVM.R")
Error in install.packages : Updating loaded packages
           Actual
Predicted    setosa versicolor virginica
  setosa         14          0         0
  versicolor      0         17         0
  virginica       0          1        13
Accuracy: 97.77778 %
>
```

```
Data
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
```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
8a-SVM.R
Environment History Connections Tutorial
Files Plots Packages Help Viewer Presentation
Logistic Regression: Transmission vs. MPG
Probability of Manual Transmission
Miles Per Gallon (mpg)
0.0
0.2
0.4
0.6
0.8
1.0
10
15
20
25
30
Type here to search
Hot weather
2:11 PM
9/11/2024

Decision tree:



```
# 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)
```

```
      Sepal.Width  < 2.65 to the left,  agree=0.638, adj=0.265, (0 split)

Node number 6: 35 observations
  predicted class=versicolor  expected loss=0.1142857  P(node) =0.3333333
    class counts:     0    31     4
   probabilities: 0.000 0.886 0.114

Node number 7: 34 observations
  predicted class=virginica   expected loss=0.02941176  P(node) =0.3238095
    class counts:     0     1    33
   probabilities: 0.000 0.029 0.971

           Actual
Predicted    setosa versicolor virginica
  setosa         14          0         0
  versicolor      0         18         1
  virginica       0          0        12
Accuracy: 97.77778 %
>
```

RESULT:

Thus SVM and Decision tree classification techniques has been successfully executed.