

Exp8: 210701306

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)

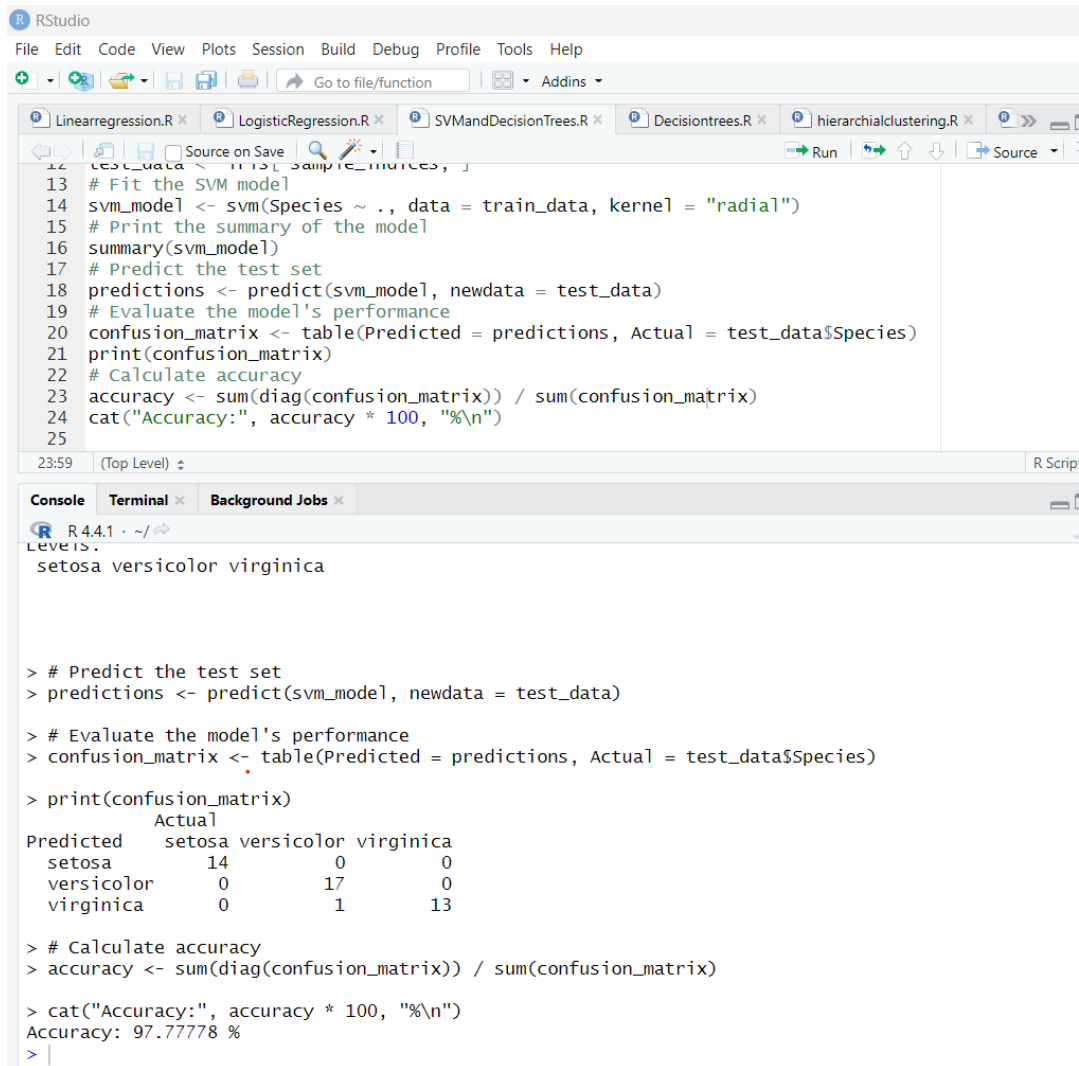
# Predict the test set predictions <- predict(svm_model,
newdata = test_data)

# 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")
```

b) Decision tree in R

```
# Install and load the rpart package (if not already installed)
install.packages("rpart") library(rpart)
```



The screenshot shows the RStudio environment. The script editor contains R code for training and testing an SVM model. The console shows the output of the code, including the confusion matrix and the calculated accuracy.

```
12 test_data <- iris[sample_indices, ]
13 # Fit the SVM model
14 svm_model <- svm(Species ~ ., data = train_data, kernel = "radial")
15 # Print the summary of the model
16 summary(svm_model)
17 # Predict the test set
18 predictions <- predict(svm_model, newdata = test_data)
19 # Evaluate the model's performance
20 confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
21 print(confusion_matrix)
22 # Calculate accuracy
23 accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
24 cat("Accuracy:", accuracy * 100, "%\n")
25
```

Console output:

```
R 4.4.1 ~ /
Levels:
setosa versicolor virginica

> # 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 %
> |
```

```
# 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")
```

