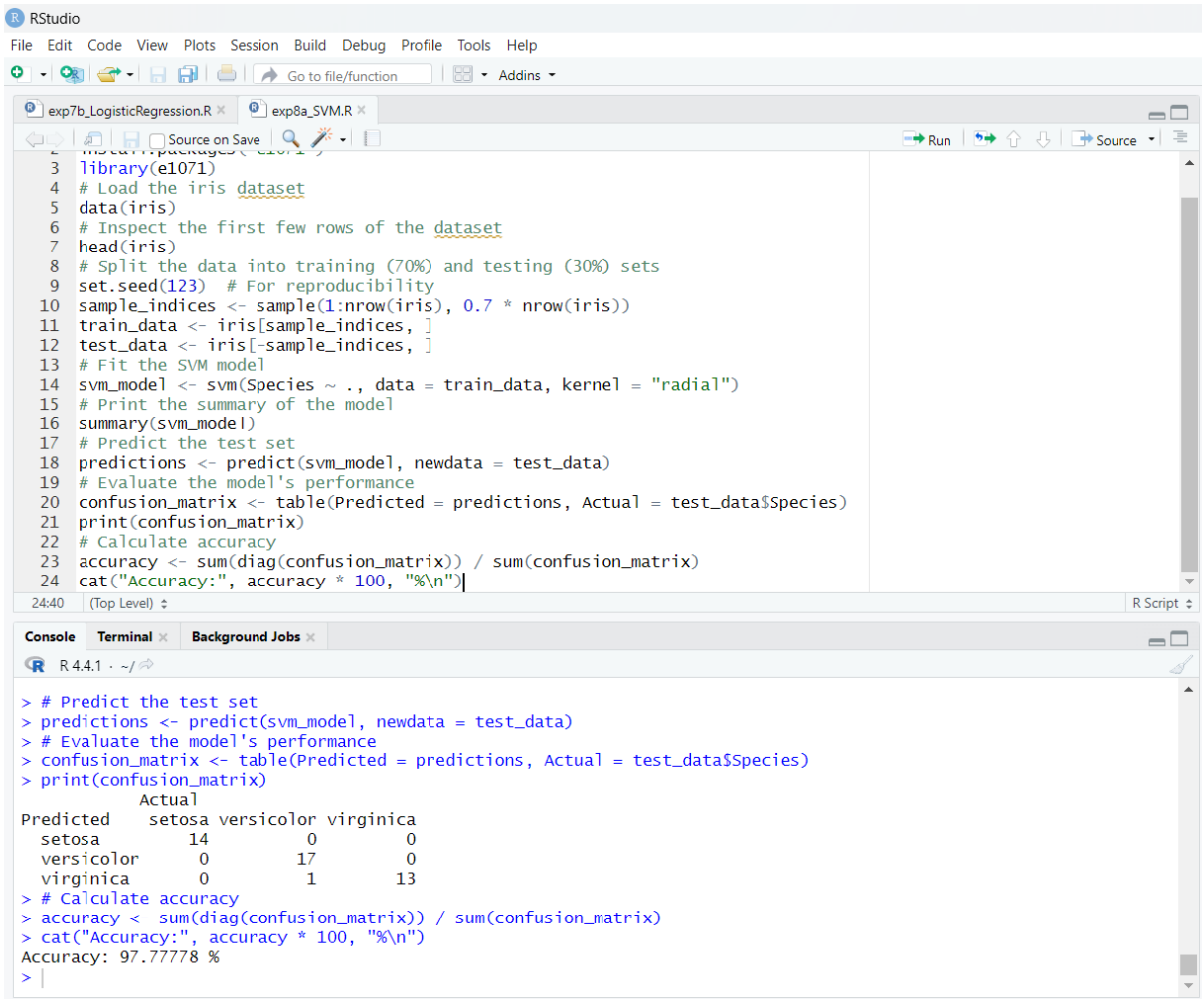
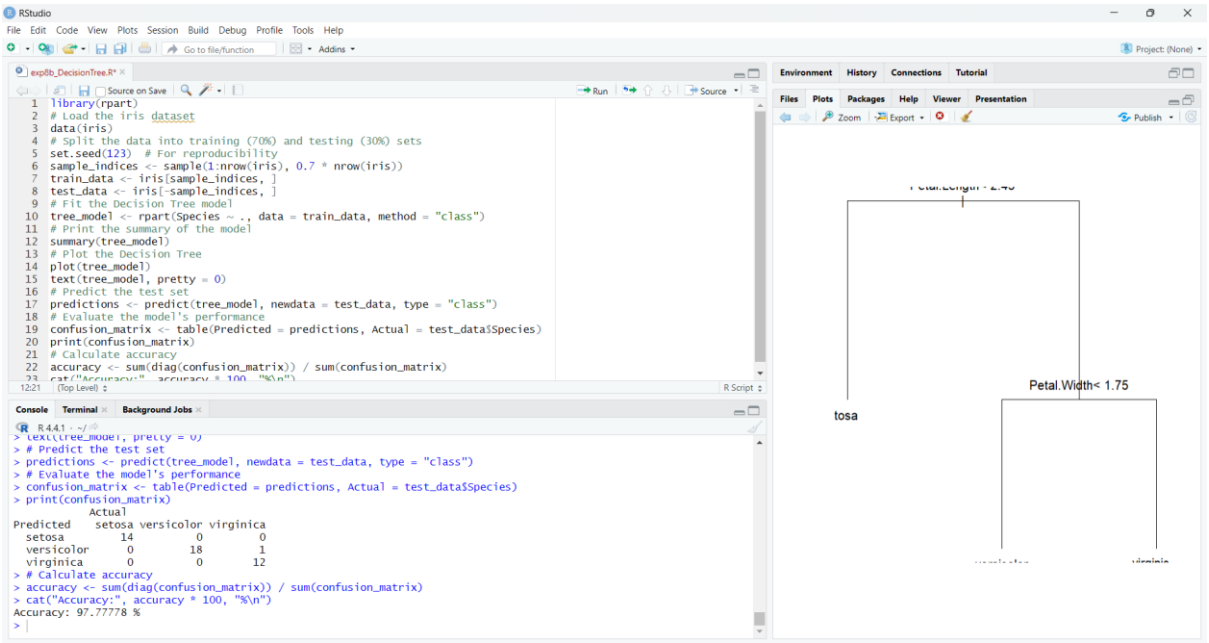
8a

```
# 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")
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
exp7b_LogisticRegression.R
exp8a_SVM.R
Source on Save
Run
Source
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$Species)
print(confusion_matrix)
# Calculate accuracy
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
cat("Accuracy:", accuracy * 100, "%\n")
24:40
(Top Level)
R Script
Console
Terminal
Background Jobs
R 4.4.1 · ~/
> # Predict the test set
> predictions <- predict(svm_model, newdata = test_data)
> # Evaluate the model's performance
> confusion_matrix <- table(Predicted = predictions, Actual = test_data$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 %
>

8b

```
# 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")
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
exp8b_DecisionTree.R*
Source on Save
Run
Source
1 library(rpart)
2 # Load the iris dataset
3 data(iris)
4 # Split the data into training (70%) and testing (30%) sets
5 set.seed(123) # For reproducibility
6 sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
7 train_data <- iris[sample_indices,]
8 test_data <- iris[-sample_indices,]
9 # Fit the Decision Tree model
10 tree_model <- rpart(Species ~ ., data = train_data, method = "class")
11 # Print the summary of the model
12 summary(tree_model)
13 # Plot the Decision Tree
14 plot(tree_model)
15 text(tree_model, pretty = 0)
16 # Predict the test set
17 predictions <- predict(tree_model, newdata = test_data, type = "class")
18 # Evaluate the model's performance
19 confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
20 print(confusion_matrix)
21 # Calculate accuracy
22 accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
23 cat("Accuracy:", accuracy * 100, "%\n")
12:21 (Top Level) R Script
Console Terminal Background Jobs
R 4.4.1 · ~/
> text(tree_model, pretty = 0)
> # Predict the test set
> predictions <- predict(tree_model, newdata = test_data, type = "class")
> # Evaluate the model's performance
> confusion_matrix <- table(Predicted = predictions, Actual = test_data$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 %
Environment History Connections Tutorial
Files Plots Packages Help Viewer Presentation
Zoom Export Publish
Petal.Width< 1.75
tosa