

Exp8:

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")
```

```

1 # Install and load the e1071 package (if not already installed)
2 install.packages("e1071")
3 library(e1071)
4 # Load the iris dataset
5 data(iris)
6 # Inspect the first few rows of the dataset
7 head(iris)
8 # Split the data into training (70%) and testing (30%) sets
9 set.seed(123) # For reproducibility
10 sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
11 train_data <- iris[sample_indices, ]
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")

```

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	
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 ...
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

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 %
>

```

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")

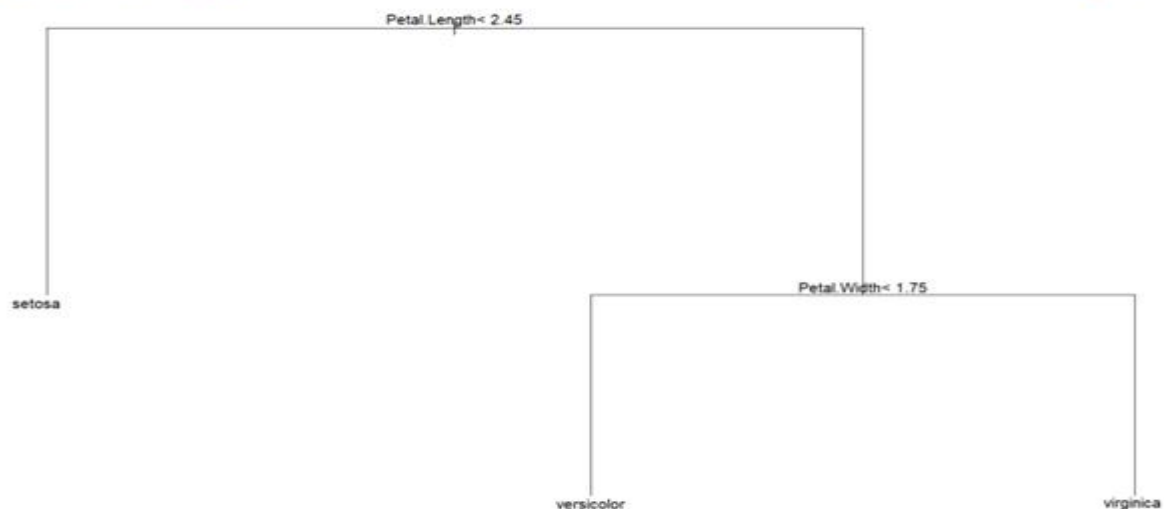
# Evaluate the model's performance
confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
print(confusion_matrix)
```

```
# Calculate accuracy
```

```
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
```

```
cat("Accuracy:", accuracy * 100, "%\n")
```

```
6 data(iris)
7
8 # Split the data into training (70%) and testing (30%) sets
9 set.seed(123) # For reproducibility
10 sample_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
11 train_data <- iris[sample_indices, ]
12 test_data <- iris[-sample_indices, ]
13
14 # Fit the Decision Tree model
15 tree_model <- rpart(Species ~ ., data = train_data, method = "class")
16
17 # Print the summary of the model
18 summary(tree_model)
19
20 # Plot the Decision Tree with smaller margins
21 par(mar = c(1, 1, 1, 1)) # Adjust the margins to make them smaller
22 plot(tree_model)
23 text(tree_model, pretty = 0)
24
25 # Predict the test set
26 predictions <- predict(tree_model, newdata = test_data, type = "class")
27
28 # Evaluate the model's performance
29 confusion_matrix <- table(Predicted = predictions, Actual = test_data$Species)
30 print(confusion_matrix)
31
32 # Calculate accuracy
33 accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
34 cat("Accuracy:", accuracy * 100, "%\n")
35
```



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	
tree_model	List of 14	
Values		
accuracy	0.977777777777778	
confusion_matrix	'table' int [1:3, 1:3] 14 0 0 0 18 0 0 1 12	
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...	
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	