Name: Varun Sudhir

Reg No: 21BDS0040

Exploratory Data Analysis Lab Practice Problem Set-4

Write R programs to implement the KNN algorithm by reading the data and user-specified value.

We will be utilizing the inbuilt Iris dataset and apply KNN on this dataset to predict the Species on the test data

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa

Code:

```
# Varun Sudhir 21BDS0040

library(ggplot2)
library(dplyr)

euclidean_distance <- function(x1, x2) {
  sqrt(sum((x1 - x2) ^ 2))
}

knn_algorithm <- function(train_data, train_labels, test_data, k) {
  predictions <- vector("character", nrow(test_data))

  for (i in 1:nrow(test_data)) {
    distances <- apply(train_data, 1, function(row) euclidean_distance(row,
test_data[i, ]))
    if (i == 1) {
      cat("Distances for first test point:\n")
      print(distances)
    }
    k_nearest_neighbors <- order(distances)[1:k]
    k_nearest_labels <- train_labels[k_nearest_neighbors]
    if (i == 1) {
      cat("Indices of k nearest neighbors for first test point:",
k_nearest_neighbors, "\n")
      cat("Labels of k nearest neighbors for first test point:",
k_nearest_labels, "\n")
    }
    predictions[i] <- names(sort(table(k_nearest_labels), decreasing =
TRUE)[1])
    if (i == 1) {
      cat("Predicted label for first test point:", predictions[i], "\n\n")
    }
  }

  return(predictions)
}

normalize <- function(data) {
  return((data - min(data)) / (max(data) - min(data)))
}


split_data <- function(data, train_ratio = 0.8) {
  set.seed(42)
  sample_index <- sample(1:nrow(data), train_ratio * nrow(data))
  train_data <- data[sample_index, ]
  test_data <- data[-sample_index, ]
```

```
  list(train = train_data, test = test_data)
}

data("iris")
iris_data <- iris %>%
  mutate(across(where(is.numeric), normalize)) %>%
  select(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, Species)


split <- split_data(iris_data)
train_data <- split$train
test_data <- split$test

k <- as.integer(readline(prompt = "Enter the value of k: "))
train_features <- train_data %>% select(-Species)
train_labels <- train_data$Species
test_features <- test_data %>% select(-Species)
test_labels <- test_data$Species


predictions <- knn_algorithm(as.matrix(train_features), train_labels,
as.matrix(test_features), k)
accuracy <- sum(predictions == test_labels) / length(test_labels)
cat("Accuracy:", accuracy * 100, "%\n")
cat("Varun Sudhir 21BDS0040")

test_data$Predicted <- predictions
ggplot(test_data, aes(x = Petal.Length, y = Petal.Width)) +
  geom_point(aes(color = Species), shape = 16, size = 3) +
  geom_point(aes(color = Predicted), shape = 1, size = 3) +
  labs(title = paste("KNN Classification with k =", k," (Varun Sudhir
21BDS0040)"),
       x = "Petal Length (normalized)",
       y = "Petal Width (normalized)",
       color = "Label") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_manual(values = c("setosa" = "blue", "versicolor" = "green",
"virginica" = "red"))

plot_decision_boundary <- function(train_features, train_labels, k) {
  x_min <- min(train_features$Petal.Length) - 0.05
  x_max <- max(train_features$Petal.Length) + 0.05
  y_min <- min(train_features$Petal.Width) - 0.05
  y_max <- max(train_features$Petal.Width) + 0.05
  x_seq <- seq(x_min, x_max, by = 0.01)
  y_seq <- seq(y_min, y_max, by = 0.01)
  grid <- expand.grid(Petal.Length = x_seq, Petal.Width = y_seq)
```

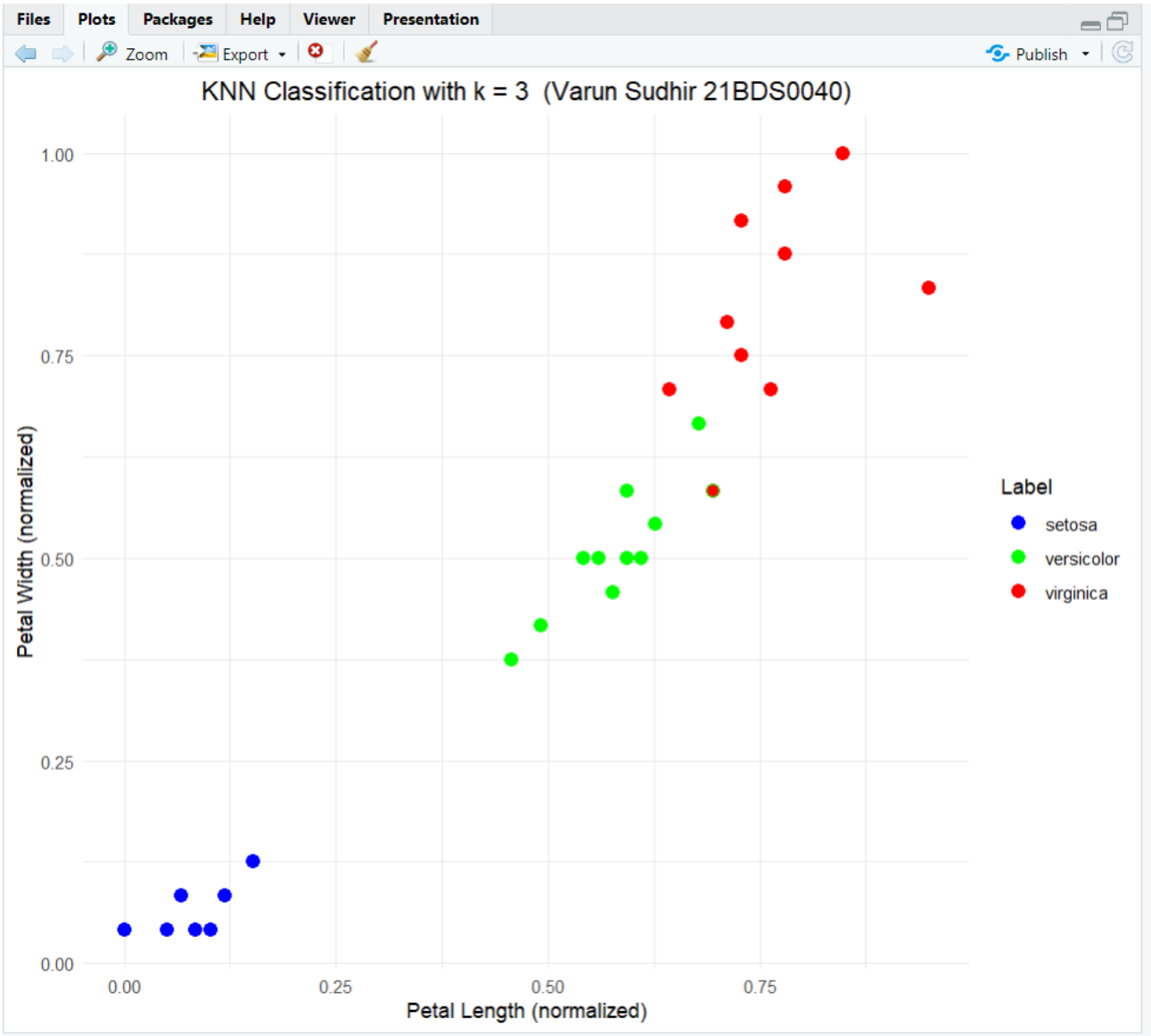
```
  grid$Predicted <- knn_algorithm(as.matrix(train_features), train_labels,
as.matrix(grid), k)

  ggplot() +
    geom_tile(data = grid, aes(x = Petal.Length, y = Petal.Width, fill =
Predicted), alpha = 0.3) +
    geom_point(data = train_data, aes(x = Petal.Length, y = Petal.Width, color
= Species), size = 2) +
    labs(title = paste("Decision Boundary with k =", k," (Varun Sudhir
21BDS0040)"),
         x = "Petal Length (normalized)",
         y = "Petal Width (normalized)") +
    scale_fill_manual(values = c("setosa" = "blue", "versicolor" = "green",
"virginica" = "red")) +
    theme_minimal()
}

plot_decision_boundary(train_features, train_labels, k)
```

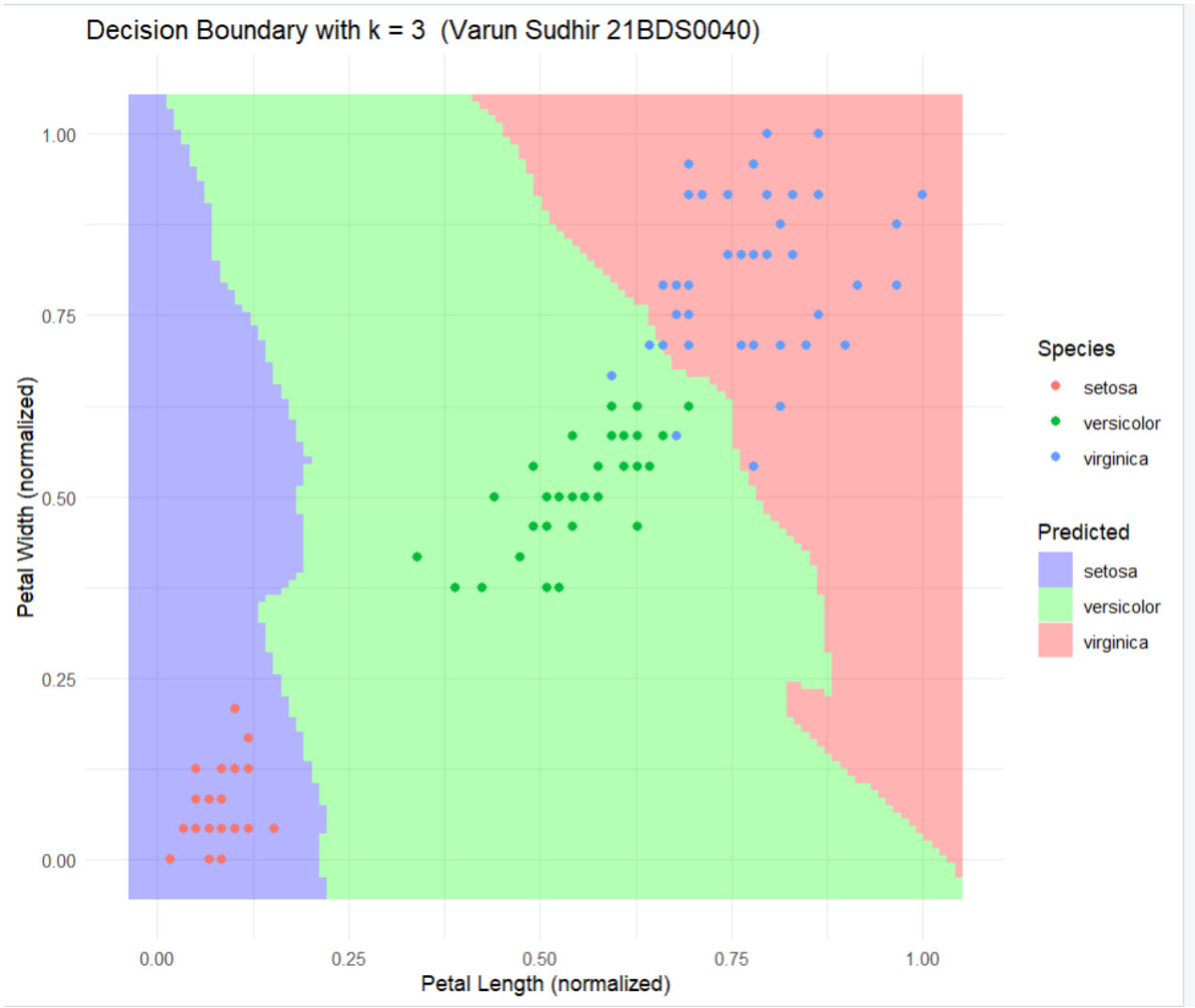
Output:

For k = 3

```
Distances for first test point:
        49         65         74        146        122        150        128
0.23549316 0.65819062 0.83040788 1.21545190 0.99664787 0.97060899 0.97155765
        47         24         71        100         89        110         20
0.22350192 0.17480315 0.92739375 0.73409665 0.69854990 1.41542040 0.21761244
       114        111        131         41        139         27        109
1.05263105 1.08651173 1.32154095 0.11987103 0.94959245 0.12341347 1.19485974
         5         84         34         92        104          3         58
0.14500426 0.96072087 0.41874482 0.84001588 1.07878290 0.09868857 0.60772595
        97         42        142         30         43         15         22
0.73184747 0.45948702 1.22944144 0.10296240 0.10979184 0.42011465 0.19219393
       123          8         36         68         86         18        130
1.45351315 0.11987103 0.14891383 0.70048025 0.84894490 0.14500426 1.18277112
       126         69          4         98         50         99         88
1.23573690 0.98671220 0.13284722 0.81002592 0.12576924 0.58699942 0.93017244
        87        145         26          6        105          2        124
0.95851130 1.30899231 0.20738530 0.31162007 1.22033819 0.19094065 1.02498714
        21         96        115         10         40        129         33
0.23174186 0.69784641 1.15434566 0.17263013 0.14599147 1.17547902 0.34652408
       140         73         29         76          9         35         16
1.20247330 0.98261345 0.17179607 0.89723776 0.21960262 0.15682101 0.51865084
       107         93        120        138         80         55         90
0.87402364 0.74635400 1.01170222 1.06731308 0.64516082 0.94086010 0.75560120
        94         57        121         77         13         53         54
0.64124130 0.91157767 1.28091261 0.98899682 0.19444444 1.01277558 0.80023598
        32         60         85         17         44         83         72
0.22672914 0.70884609 0.77667921 0.30791050 0.17565843 0.71857614 0.77712244
       135        118        149         48        136         64         38
1.00077100 1.48424333 1.16258907 0.09316950 1.44855871 0.86018863 0.14433757
         1        144         14        132         61         81        103
0.15087195 1.28708005 0.21036190 1.44498308 0.75125271 0.71602508 1.28618049
       108        141         62        102         67         25         66
1.29856540 1.27604025 0.79586535 1.01681011 0.79436035 0.10956448 0.90803007
        63        125        147        143         31        119         39
0.82492489 1.19819994 1.08644838 1.01681011 0.14695791 1.55437160 0.18134934
       113
1.20276422
Indices of k nearest neighbors for first test point: 95 27 32
Labels of k nearest neighbors for first test point: 1 1 1
Predicted label for first test point: setosa
```

```
> accuracy <- sum(predictions == test_labels) / length(test_labels)
> cat("Accuracy:", accuracy * 100, "%\n")
Accuracy: 96.66667 %
> cat("Varun Sudhir 21BDS0040")
Varun Sudhir 21BDS0040
```




Decision Boundary with k = 3 (Varun Sudhir 21BDS0040)
Petal Width (normalized)
Petal Length (normalized)
0.00
0.25
0.50
0.75
1.00
Species
setosa
versicolor
virginica
Predicted
setosa
versicolor
virginica