Team_Essay_4

Sahil, Ariel, Ilya, Lekha, Anthony, Trang

4/11/2021

Introduction

In this essay, our team will predict the species of a flower using the K Nearest Neighbors (KNN) algorithm. Given the sepal length, sepal width, petal length, and petal width in centimeters, we will classify the flower as either an Iris-setosa (ISE), Iris-versicolor (IVE), or Iris-virginica (IVI). Our dataset, flowers, contains 50 flowers from each of 3 species of iris.

Formula and Basics

The KNN algorithm is one the most commonly used and simplest in the machine learning field. The way it works is that it takes a new data point and categorizes it based on it's similarity to its neighboring points. The number of neighbors that it is compared to is dependent on the selected K value, a K value of 3 would compare the uncategorized data point to three of it's neighbors. Determining which K number of neighbors are the nearest is based on different distance formulas such Euclidean, Manhattan, and Minkowski. In our case we utilized the euclidean distance formula to do so. This uses the coordinates of the uncategorized data point and one of it's numbers to determine a euclidean distance, the data points with the lowest distances will then be selected for comparison.

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- x_1 is the x coordinate of the first point
- y_1 is the coordinate of the first point
- x_2 is the x coordinate of the second point
- y_2 is the coordinate of the second point

Loading required R packages

```r
library("readxl")
library("class")
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library("gmodels")
library("flexplot")
```

```
##
## Attaching package: 'flexplot'

## The following object is masked from 'package:ggplot2':
##
##     flip_data
```

```
library("caret")
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

Data description

- Sepal_length: represents length of a flower's sepals
- Sepal_width: represents width of a flower's sepals
- Petal_length: represents length of a flower's petals
- Petal_width: represents width of a flower's petals
- Species: indicates the species of the flower (Iris-setosa, Iris-versicolor, Iris-virginica)

Examples of data and problem

```
flower_data <- read_excel("IRIS.xlsx")
flower_data
```

```
## # A tibble: 150 x 5
##    sepal_length sepal_width petal_length petal_width species
##           <dbl>       <dbl>        <dbl>       <dbl> <chr>
##  1          5.1         3.5          1.4         0.2 Iris-setosa
##  2          4.9         3            1.4         0.2 Iris-setosa
##  3          4.7         3.2          1.3         0.2 Iris-setosa
##  4          4.6         3.1          1.5         0.2 Iris-setosa
##  5          5           3.6          1.4         0.2 Iris-setosa
##  6          5.4         3.9          1.7         0.4 Iris-setosa
##  7          4.6         3.4          1.4         0.3 Iris-setosa
##  8          5           3.4          1.5         0.2 Iris-setosa
##  9          4.4         2.9          1.4         0.2 Iris-setosa
## 10          4.9         3.1          1.5         0.1 Iris-setosa
## # ... with 140 more rows
```

Visualization

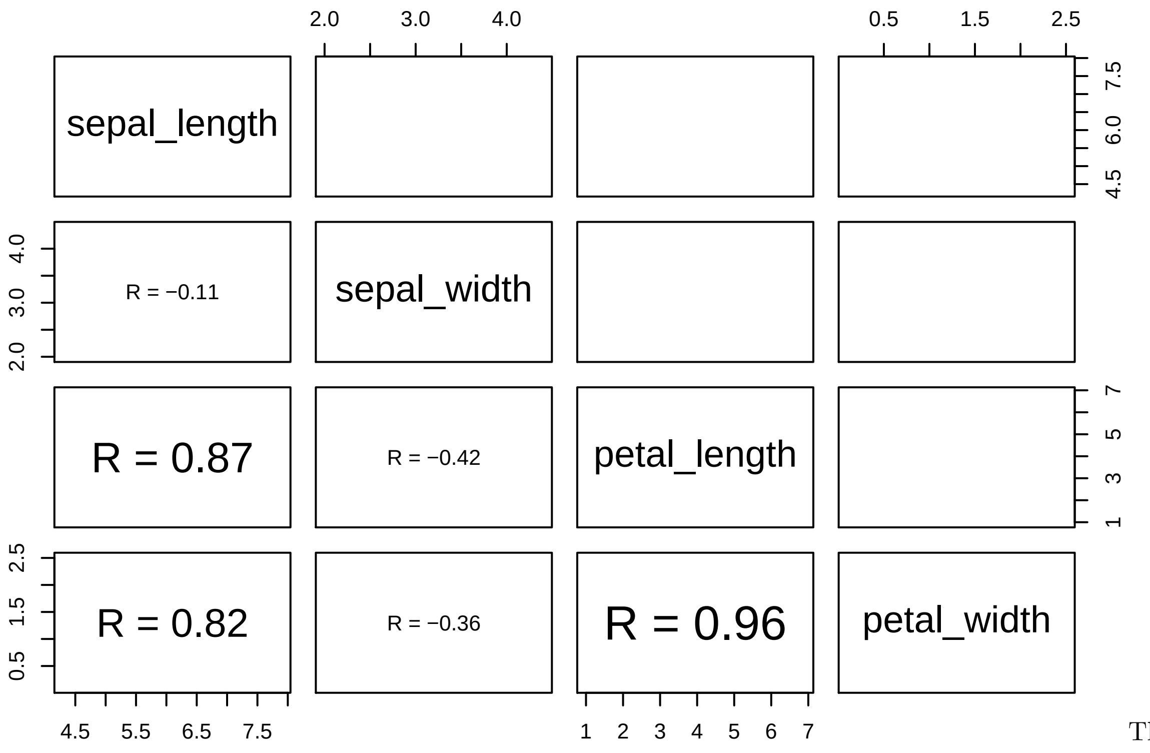
```
flowers <- read_excel("IRIS.xlsx")
stringsAsFactors = FALSE # This command helps to convert every character vector to a factor wherever it
str(flowers) # We use this command to see whether the data is structured or not.
```

```
## tibble [150 x 5] (S3: tbl_df/tbl/data.frame)
##  $ sepal_length: num [1:150] 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ sepal_width : num [1:150] 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ petal_length: num [1:150] 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ petal_width : num [1:150] 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ species     : chr [1:150] "Iris-setosa" "Iris-setosa" "Iris-setosa" "Iris-setosa" ...
```

```
table(flowers$species) # It gives the number of indiviudal species
```

```
##
##     Iris-setosa Iris-versicolor  Iris-virginica
##              50              50              50
```

```
my_cols <- c("#00AFBB", "#E7B800", "#FC4E07")
# Correlation panel
panel.cor <- function(x, y){
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- round(cor(x, y), digits=2)
    txt <- paste0("R = ", r)
    cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}
# Customize upper panel
upper.panel<-function(x, y){
  points(x,y, pch = 19, col = my_cols[flowers$species])
}
# Create the plots
pairs(flowers[,1:4],
      lower.panel = panel.cor,
      upper.panel = upper.panel)
```



The scatter plot matrix above graphs all the pairs of predictor dimensions with the blue points representing the ISE species, the yellow points representing the IVE species, and the red points representing the IVI species. We observe that petal width and petal length are highly correlated with a value of R = 0.96 as well as sepal length and petal length with a value of R = 0.87.

Analysis

Computation and Model Evaluation

```
# Random splitting of iris data as 70% train and 30% test datasets
ind <- sample(2, nrow(flowers), replace=TRUE, prob=c(0.7, 0.3))
trainData <- flowers[ind==1,]
testData <- flowers[ind==2,]

# Removing factor variable from training and test datasets
trainData1 <- trainData[-5]
testData1 <- testData[-5]

flowers_train_labels <- trainData$species # This shows how we classify our observation

flowers_test_labels <- testData$species # This shows how we classify our observation
```

Model Assessment

```
# k is taken by sqrt(number of data points)
flowers_test_pred1 <- knn(train = trainData1, test = testData1, cl= flowers_train_labels, k = 11, prob=
flowers_test_pred2 <- knn(train = trainData1, test = testData1, cl= flowers_train_labels, k = 12, prob=

# Model accuracy based on what k is
ACC.11 <- 100 * sum(flowers_test_labels == flowers_test_pred1)/NROW(flowers_test_labels)
ACC.12 <- 100 * sum(flowers_test_labels == flowers_test_pred2)/NROW(flowers_test_labels)

ACC.11
```

```
## [1] 98.11321
```

```
ACC.12
```

```
## [1] 98.11321
```

```
CrossTable(x = flowers_test_labels, y = flowers_test_pred2, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  53
##
##
##                      | flowers_test_pred2
## flowers_test_labels |     Iris-setosa | Iris-versicolor |  Iris-virginica |       Row Total |
## --------------------|-----------------|-----------------|-----------------|-----------------|
##         Iris-setosa |              14 |               0 |               0 |              14 |
##                     |           1.000 |           0.000 |           0.000 |           0.264 |
##                      |        1.000 |        0.000 |        0.000 |              |
##                      |        0.264 |        0.000 |        0.000 |              |
## ---------------------|--------------|--------------|--------------|--------------|
##      Iris-versicolor |            0 |           20 |            1 |           21 |
##                      |        0.000 |        0.952 |        0.048 |        0.396 |
##                      |        0.000 |        1.000 |        0.053 |              |
##                      |        0.000 |        0.377 |        0.019 |              |
## ---------------------|--------------|--------------|--------------|--------------|
##       Iris-virginica |            0 |            0 |           18 |           18 |
##                      |        0.000 |        0.000 |        1.000 |        0.340 |
##                      |        0.000 |        0.000 |        0.947 |              |
##                      |        0.000 |        0.000 |        0.340 |              |
## ---------------------|--------------|--------------|--------------|--------------|
##         Column Total |           14 |           20 |           19 |           53 |
##                      |        0.264 |        0.377 |        0.358 |              |
## ---------------------|--------------|--------------|--------------|--------------|
##
##
```

Interpretation and Model Summary

- We trained our model on 70% of our data set and tested it on the remaining 30%. Then, we tested the model's ability to predict the species of a lilly given sepal length, sepal width, petal length, and petal width. We did this by running a knn algorithm on our testing sets and created a cross table to see how accurate our model's predictive capabilities are. We found that with k = 12, our model is 95% accurate.

Prediction and Model Accuracy

- The model accuracy when k = 11 is 93% and when k = 12 is 95%. So we see that k = 12 works better for our model.
- The model predicts the setosa, versicolor and virginica correctly almost every time without giving any false predictions.
- False predictions occur in case of predicting versicolor, where at times it is predicted as virginica or vice-versa.

Conclusion

As a whole, we costructed our prediction model on the basis of the KNN algorithm and tested its validity, using a dataset put together from a total of 150 collected flowers (each flower being one data point) and what each flower's observed physical attributes were, including whether they're one of three species: setosa, versicolor, or virginica. The physical attributes considered were sepal length, sepal width, petal length, and petal width, all of which were quantitative predictor variables. The purpose of our model was to predict the species of the Iris flower given these associated values.

Using R code to carry it out, we first install the relevant R packages required for us to analyze and modify our data accordingly. For applicability, we then converted every character vector found in our data to a factor wherever found reasonable, checked whether the data is well-structured or not, and carried out a count of the species distribution of the data points, to find that they are equally distributed amongst the three species (50-50-50).

Since we chose to use the Euclidean distance formula for determining which K number of neighbors, we had to analyze the correlation between each of the predictor dimensions based on their values, and via the built scatterplot matrix, we found the pairs that were best correlated to each other. The firstmost pair was petal width and petal length as they had a coefficient of correlation of 0.96, followed by sepal length and petal length with that of 0.87.

Following this, we wanted to test the accuracy of the model, and to do this we split our data randomly, with 70% of data points used for training dataset to train the model and remaining 30% for testing dataset to test it. We considered two values of k for our KNN algorithm based on the square root of the numbe of data points: 11 and 12. After running the algorithm and creating a cross table, we found that with k = 12 our model's precision was relatively the most ideal at 95% accuracy. Then, we tested the model's ability to predict the species of a lilly given sepal length, sepal width, petal length, and petal width. We did this by running a knn algorithm on our testing sets and created a cross table to see how accurate our model's predictive capabilities are. We found that with k = 12, our model is 95% accurate. Though there were specific errors, as in when false predictions occured in the case of predicting versicolor when the model should have predicted virginia to be the flower species and vice versa, overall we can conclude that our model was able to predict the correct species using KNN-algorithm, and it can be used as a reliable prediction model to test with Iris flowers.

Reference

https://rpubs.com/Drmadhu/IRISclassification