library(tidyverse)

|  |  |
| --- | --- |
| [1] | "lubridate" |
| [2] | "forcats" |
| [3] | "stringr" |
| [4] | "dplyr" |
| [5] | "purrr" |
| [6] | "readr" |
| [7] | "tidyr" |
| [8] | "tibble" |
| [9] | "ggplot2" |
| [10] | "tidyverse" |
| [11] | "stats" |
| [12] | "graphics" |
| [13] | "grDevices" |
| [14] | "utils" |
| [15] | "datasets" |
| [16] | "methods" |
| [17] | "base" |

#load data

df <- data(iris)

# see the structure

head(iris)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
| 1 | 5.1 | 3.5 | 1.4 | 0.2 | "setosa" |
| 2 | 4.9 | 3 | 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 | 3.6 | 1.4 | 0.2 | "setosa" |
| 6 | 5.4 | 3.9 | 1.7 | 0.4 | "setosa" |

#Generate a random number that is 90% of the total number of rows in data set.

ran <- sample(1:nrow(iris), 0.9 \* nrow(iris))

# the normalization function is created

nor <-function(x) { (x -min(x))/(max(x)-min(x)) }

#Run normalization on first 4 columns of data set because they are the predictors

iris\_norm <- as.data.frame(lapply(iris[,c(1,2,3,4)], nor))

summary(iris\_norm)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | [,1] | [,2] | [,3] | [,4] |
| [1,] | "Min. :0.0000 " | "Min. :0.0000 " | "Min. :0.0000 " | "Min. :0.00000 " |
| [2,] | "1st Qu.:0.2222 " | "1st Qu.:0.3333 " | "1st Qu.:0.1017 " | "1st Qu.:0.08333 " |
| [3,] | "Median :0.4167 " | "Median :0.4167 " | "Median :0.5678 " | "Median :0.50000 " |
| [4,] | "Mean :0.4287 " | "Mean :0.4406 " | "Mean :0.4675 " | "Mean :0.45806 " |
| [5,] | "3rd Qu.:0.5833 " | "3rd Qu.:0.5417 " | "3rd Qu.:0.6949 " | "3rd Qu.:0.70833 " |
| [6,] | "Max. :1.0000 " | "Max. :1.0000 " | "Max. :1.0000 " | "Max. :1.00000 " |

#extract training set

iris\_train <- iris\_norm[ran,]

#extract testing set

iris\_test <- iris\_norm[-ran,]

#extract 5th column of train data set because it will be used as 'cl' argument in knn function.

iris\_target\_category <- iris[ran,5]

#extract 5th column if test data set to measure the accuracy

iris\_test\_category <- iris[-ran,5]

#load the package class

library(class)

|  |  |
| --- | --- |
| [1] | "class" |
| [2] | "lubridate" |
| [3] | "forcats" |
| [4] | "stringr" |
| [5] | "dplyr" |
| [6] | "purrr" |
| [7] | "readr" |
| [8] | "tidyr" |
| [9] | "tibble" |
| [10] | "ggplot2" |
| [11] | "tidyverse" |
| [12] | "stats" |
| [13] | "graphics" |
| [14] | "grDevices" |
| [15] | "utils" |
| [16] | "datasets" |
| [17] | "methods" |
| [18] | "base" |

#run knn function

pr <- knn(iris\_train,iris\_test,cl=iris\_target\_category,k=13)

#create confusion matrix

tab <- table(pr,iris\_test\_category)

#this function divides the correct predictions by total number of predictions that tell us how accurate the model is.

accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) \* 100}

accuracy(tab)

[1] 93.3333333333333