assignment_11.2_MunjewarSheetal

Sheetal M

2023-02-25

Install and Load required packages :

```
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(warning = FALSE)
knitr::opts_chunk$set(fig.width = 12, fig.height = 10)
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 70), tidy = TRUE)

# Package names
# packages <- c("ggplot2","dplyr","tidyr","magrittr","tidyverse","purrr","tidy")
packages <- c("broom","dplyr","RWeka","class","ggplot2","caret","formatR")

# Install packages not yet installed
installed_packages <- packages %in% rownames(installed.packages())
if (any(installed_packages == FALSE)) {
  install.packages(packages[!installed_packages])
}

# Packages loading
invisible(lapply(packages, library, character.only = TRUE))
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: lattice
```

```
k_values <- list(3, 5, 10, 15, 20, 25)
knitr::opts_chunk$set(echo = TRUE)
```

KNN - Nearest neighbors algorithm for binary and trinary classifiers

Set the working directory to the root of your DSC 520 directory

setwd("E:\Data_Science_DSC510\DSC520-Statistics\dsc520")

```
# Set the working directory to the root of your DSC 520 directory
setwd("E:\\Data_Science_DSC510\\DSC520-Statistics\\dsc520")

# Load data from data/binary-classifier-data.csv
df1 <- read.csv("data/binary-classifier-data.csv")
str(df1)
```

```
## 'data.frame':    1498 obs. of  3 variables:
##  $ label: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ x    : num  70.9 75 73.8 66.4 69.1 ...
##  $ y    : num  83.2 87.9 92.2 81.1 84.5 ...
```

```
nrow(df1)
```

```
## [1] 1498
```

```
# Load data from data/trinary-classifier-data.csv
df2 <- read.csv("data/trinary-classifier-data.csv")
str(df2)
```

```
## 'data.frame':    1568 obs. of  3 variables:
##  $ label: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ x    : num  30.1 31.3 34.1 32.6 34.7 ...
##  $ y    : num  39.6 51.8 49.3 41.2 45.5 ...
```

```
nrow(df2)
```

```
## [1] 1568
```

```
# distinct values with count in data frame column Label.
table(df1$label)
```

```
##
##   0   1
## 767 731
```

```
table(df2$label)
```

```
##
##   0   1   2
## 394 722 452
```

Convert label column data type into factor

```r
df1$label <- as.factor(df1$label)
str(df1)
```

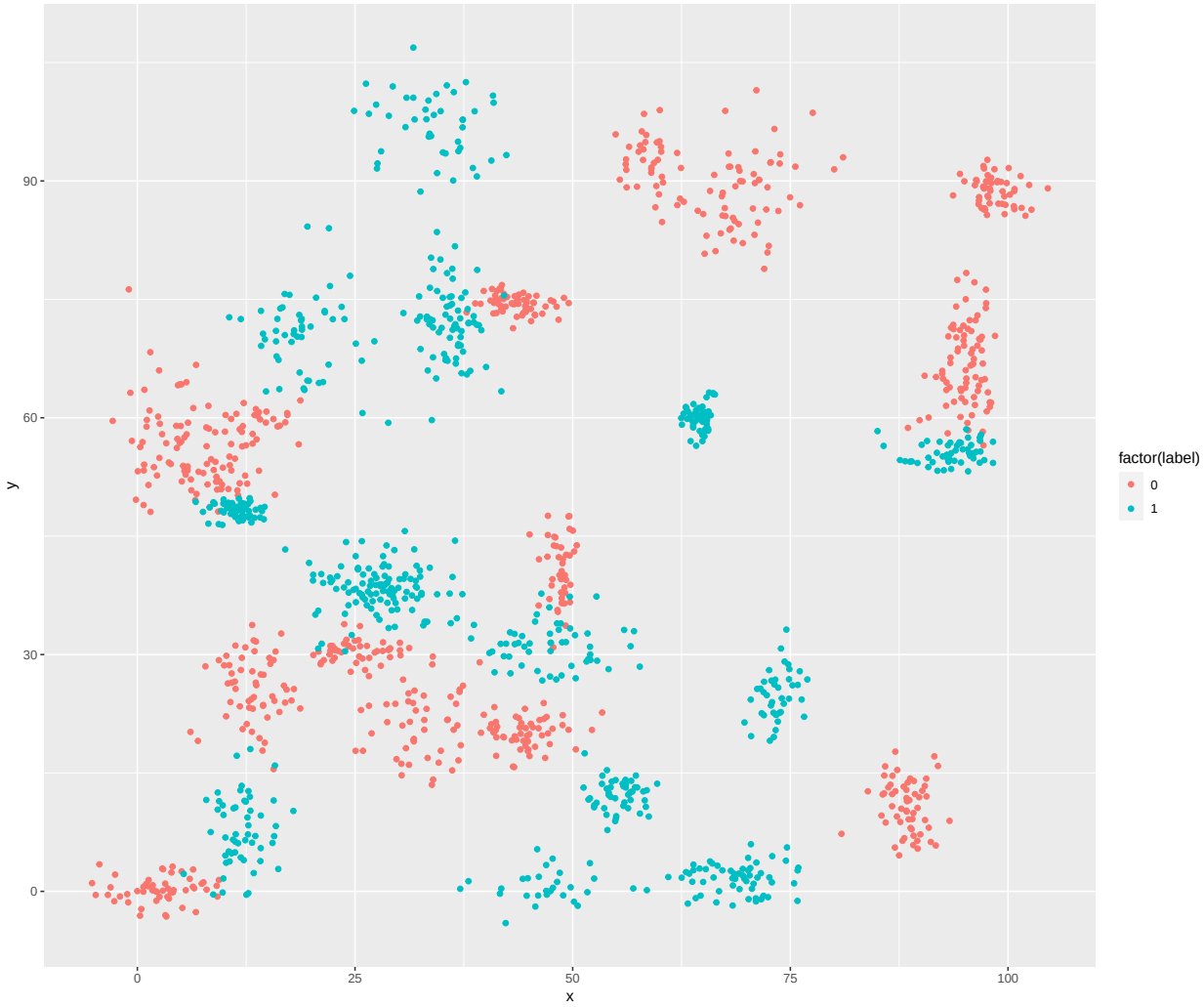
```
## 'data.frame':    1498 obs. of  3 variables:
##  $ label: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ x    : num  70.9 75 73.8 66.4 69.1 ...
##  $ y    : num  83.2 87.9 92.2 81.1 84.5 ...
```

```r
df2$label <- as.factor(df2$label)
str(df2)
```

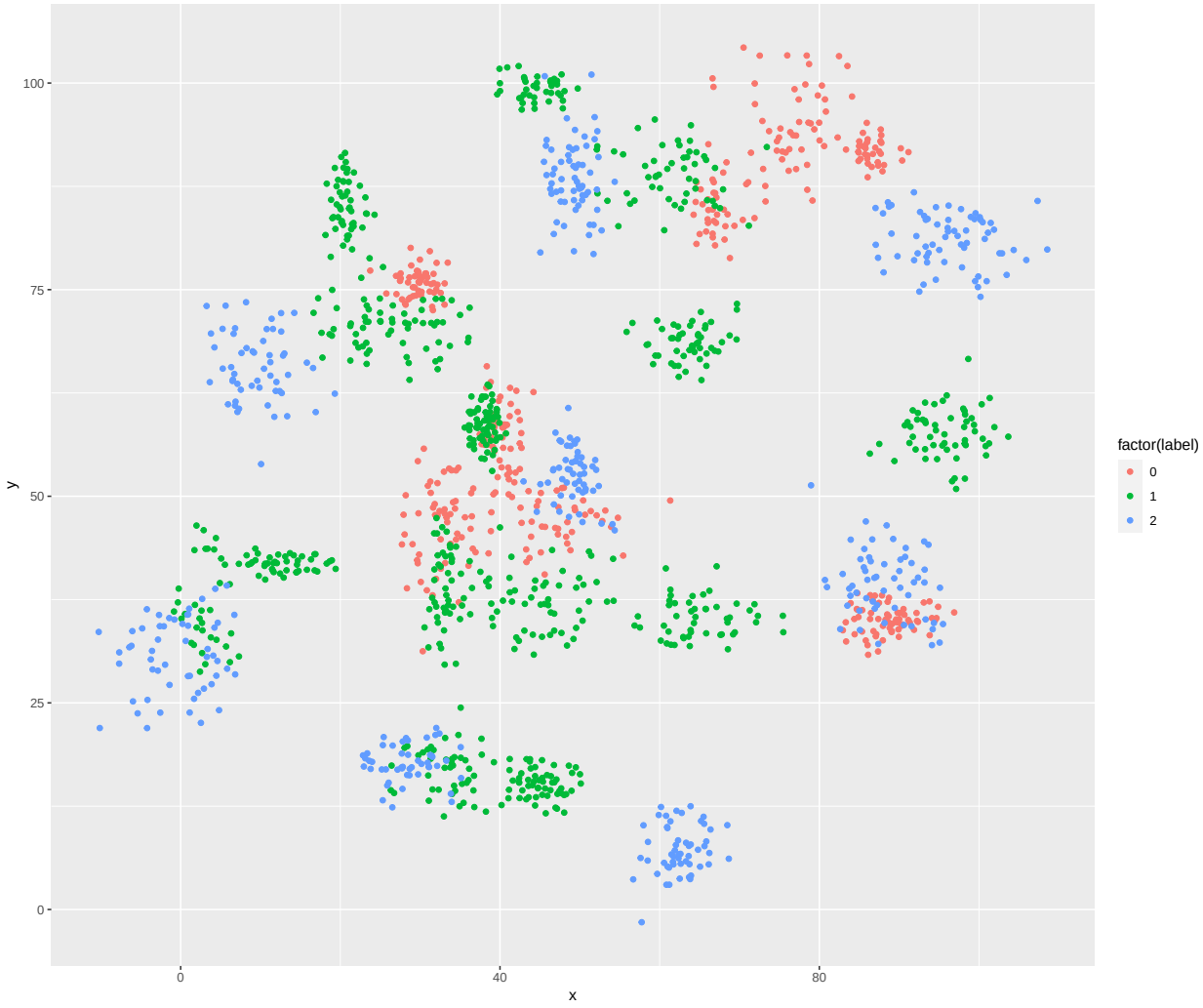
```
## 'data.frame':    1568 obs. of  3 variables:
##  $ label: Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ x    : num  30.1 31.3 34.1 32.6 34.7 ...
##  $ y    : num  39.6 51.8 49.3 41.2 45.5 ...
```

Visualize data

```r
# df1 plot
ggplot(data = df1, aes(x, y, color = factor(label))) + geom_point()
```




```
# df2 plot
ggplot(data = df2, aes(x, y, color = factor(label))) + geom_point()
```



Fit the model for dataset-df1 and its accuracy

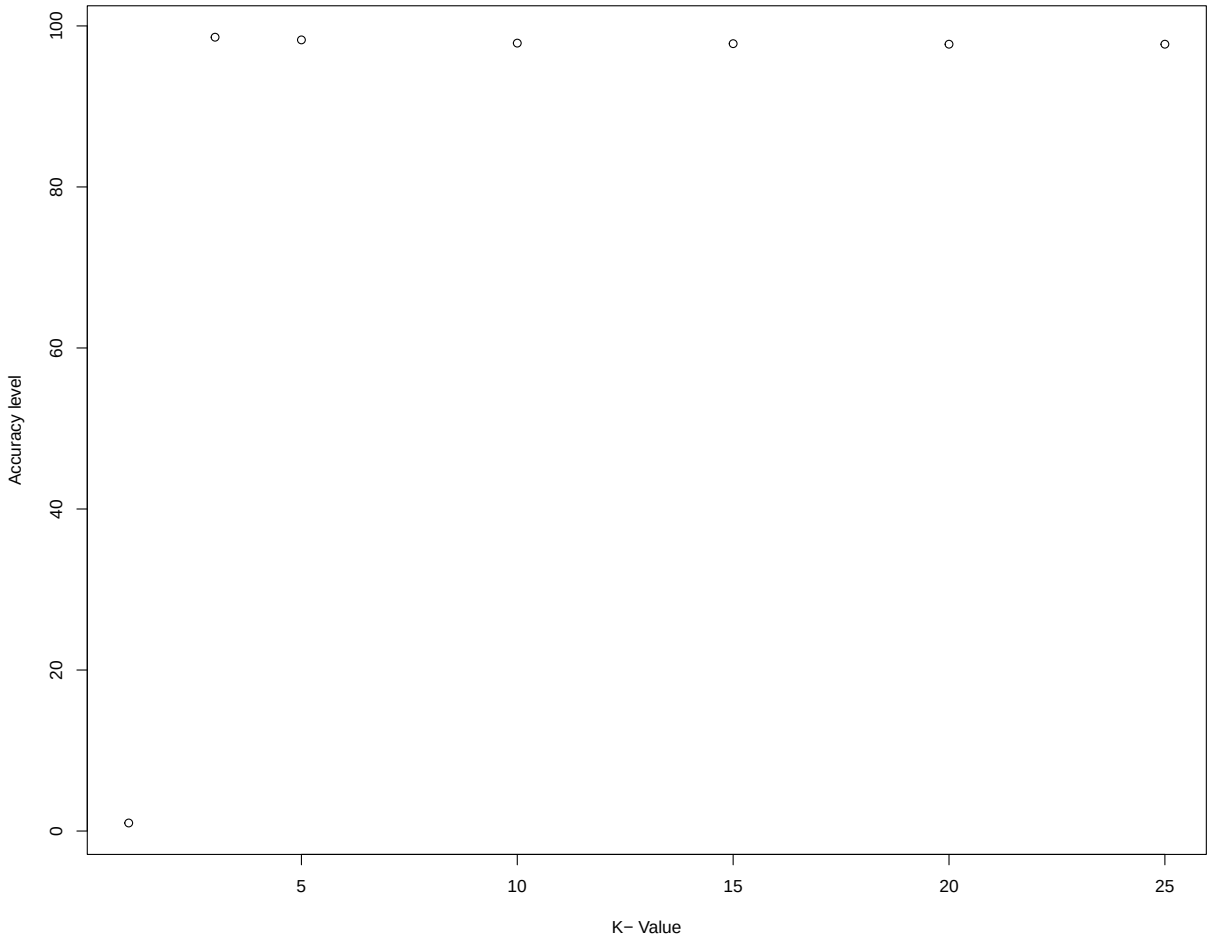
```
i = 1
k.optm = 1
for (i in k_values) {
    knn.mod <- knn(train = df1, test = df1, cl = df1$label, k = i)
    # print(paste('Printing : ',knn.mod ))
    summary(knn.mod)
    k.optm[i] <- 100 * sum(df1$label == knn.mod)/NROW(df1)
    k = i
    cat(k, "=", k.optm[i], "")
}
```

```
## 3 = 98.59813 5 = 98.26435 10 = 97.86382 15 = 97.79706 20 = 97.73031 25 = 97.73031
```

```
# k.optm
```

Accuracy plot for dataset-df1

```
plot(k.optm, type = "b", xlab = "K- Value", ylab = "Accuracy level")
```



The above graph shows that for 'K' value of 3(98.59813) we get the maximum accuracy.

Fit the model for dataset-df2 and its accuracy

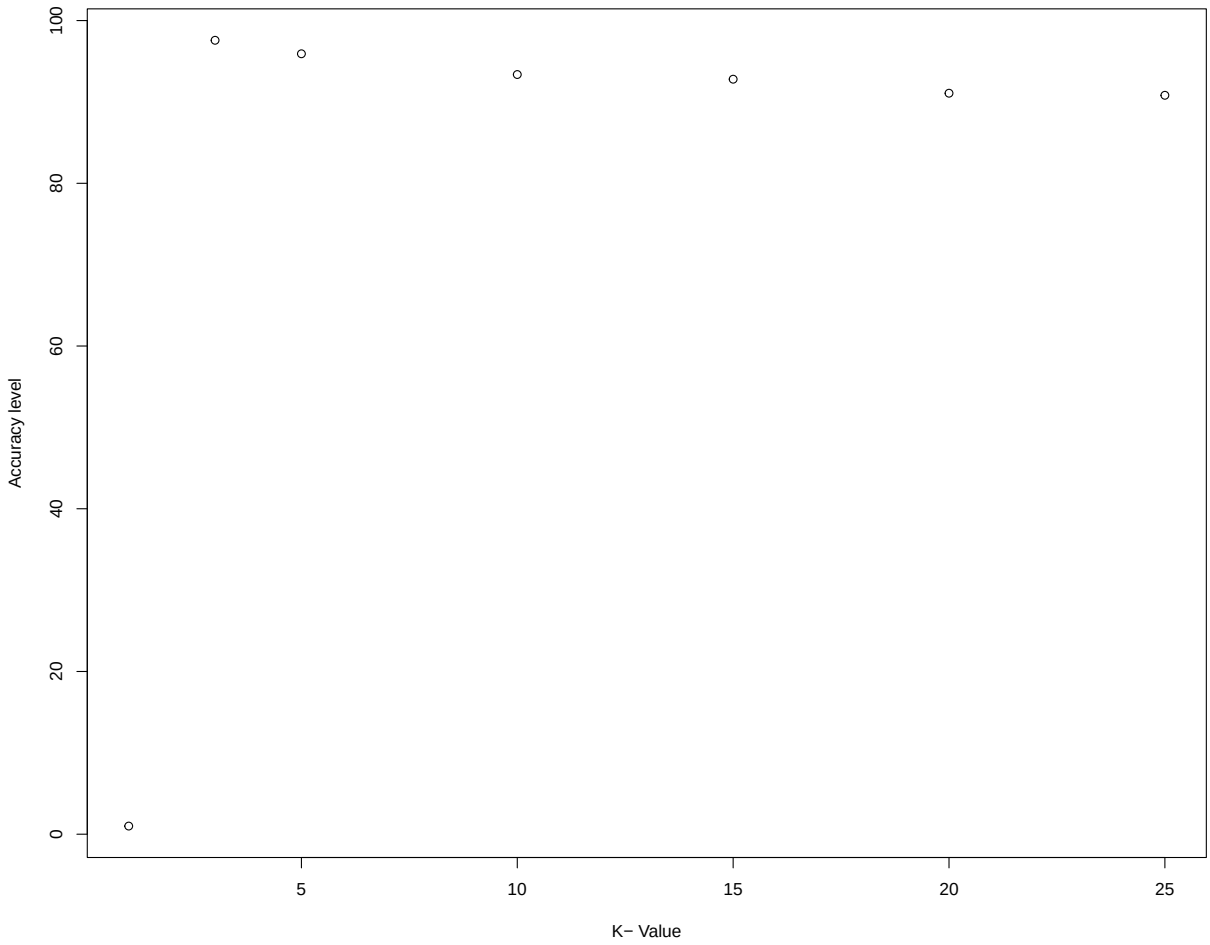
```
i = 1
k.optm = 1
for (i in k_values) {
    knn.mod <- knn(train = df2, test = df2, cl = df2$label, k = i)
    # print(paste('Printing : ',knn.mod ))
    summary(knn.mod)
    k.optm[i] <- 100 * sum(df2$label == knn.mod)/NROW(df2)
    k = i
    cat(k, "=", k.optm[i], "")
}
```

```
## 3 = 97.57653 5 = 95.91837 10 = 93.36735 15 = 92.79337 20 = 91.07143 25 = 90.81633
```

```
# k.optm
```

Accuracy plot for dataset-df2

```
plot(k.optm, type = "b", xlab = "K- Value", ylab = "Accuracy level")
```



The above graph shows that for 'K' value of 3 (97.57653) we get the maximum accuracy.

Linear classifiers.

Visuals for both datasets df1 and df2 are scattered in small clusters (reference - scatter plot drawn for df1 and df2), cannot be classified in linear model to show relationship among variables by plotting straight line, hence linear model may not be right fit for this data sets.