Assignment 8.2

Scott Breitbach

7/19/2020

## Assignment 8.2: Introduction to Machine Learning

BCdata <- read\_csv("completed/Week8/data/binary-classifier-data.csv")  
BCdata$label <- factor(BCdata$label)  
summary(BCdata)

## label x y   
## 0:767 Min. : -5.20 Min. : -4.019   
## 1:731 1st Qu.: 19.77 1st Qu.: 21.207   
## Median : 41.76 Median : 44.632   
## Mean : 45.07 Mean : 45.011   
## 3rd Qu.: 66.39 3rd Qu.: 68.698   
## Max. :104.58 Max. :106.896

str(BCdata)

## tibble [1,498 x 3] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ label: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ x : num [1:1498] 70.9 75 73.8 66.4 69.1 ...  
## $ y : num [1:1498] 83.2 87.9 92.2 81.1 84.5 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. label = col\_double(),  
## .. x = col\_double(),  
## .. y = col\_double()  
## .. )

TCdata <- read\_csv("completed/Week8/data/trinary-classifier-data.csv")  
TCdata$label <- factor(TCdata$label)  
summary(TCdata)

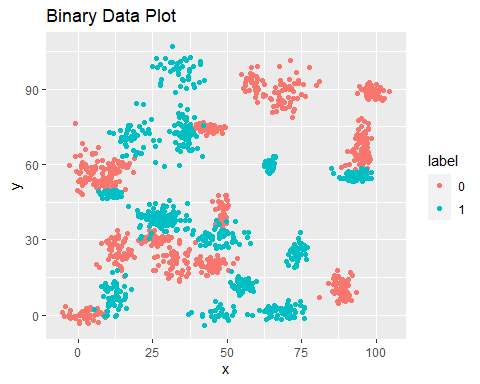
## label x y   
## 0:394 Min. :-10.26 Min. : -1.541   
## 1:722 1st Qu.: 31.15 1st Qu.: 35.906   
## 2:452 Median : 45.59 Median : 55.073   
## Mean : 48.86 Mean : 55.282   
## 3rd Qu.: 66.27 3rd Qu.: 77.403   
## Max. :108.56 Max. :104.293

str(TCdata)

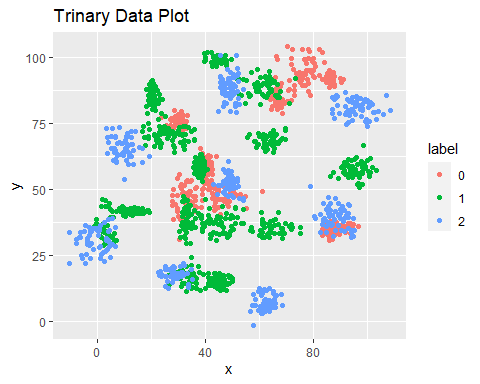
## tibble [1,568 x 3] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ label: Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...  
## $ x : num [1:1568] 30.1 31.3 34.1 32.6 34.7 ...  
## $ y : num [1:1568] 39.6 51.8 49.3 41.2 45.5 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. label = col\_double(),  
## .. x = col\_double(),  
## .. y = col\_double()  
## .. )

### a. Plot the data from each dataset using a scatter plot.

ggplot(data = BCdata, mapping = aes(x, y, color = label)) + geom\_point() + ggtitle("Binary Data Plot")



ggplot(data = TCdata, mapping = aes(x, y, color = label)) + geom\_point() + ggtitle("Trinary Data Plot")



### b. Determine which points are nearest by calculating the Euclidean distance between two points.

#### Fit a k nearest neighbors model for each dataset for k=3, k=5, k=10, k=15, k=20, and k=25. Compute the accuracy of the resulting models for each value of k.

## Set up testing and training groups from data sets  
set.seed(42)  
  
indBC <- sample(2, nrow(BCdata), replace=TRUE, prob=c(0.67, 0.33))  
BC.train <- BCdata[indBC==1, 2:3]  
BC.test <- BCdata[indBC==2, 2:3]  
BC.trainLabels <- BCdata[indBC==1,1]  
BC.trainLabels <- BC.trainLabels$label  
BC.testLabels <- BCdata[indBC==2,1]  
BC.testLabels <- BC.testLabels$label  
  
indTC <- sample(2, nrow(TCdata), replace=TRUE, prob=c(0.67, 0.33))  
TC.train <- TCdata[indTC==1, 2:3]  
TC.test <- TCdata[indTC==2, 2:3]  
TC.trainLabels <- TCdata[indTC==1,1]  
TC.trainLabels <- TC.trainLabels$label  
TC.testLabels <- TCdata[indTC==2,1]  
TC.testLabels <- TC.testLabels$label

BCpred.3 <- knn(train = BC.train, test = BC.test, cl = BC.trainLabels, k=3)  
AccBC.3 <- sum(BC.testLabels==BCpred.3)/length(BCpred.3)  
"Accuracy: Binary, k=3"

## [1] "Accuracy: Binary, k=3"

round(AccBC.3 \* 100, digits = 1)

## [1] 96.1

TCpred.3 <- knn(train = TC.train, test = TC.test, cl = TC.trainLabels, k=3)  
AccTC.3 <- sum(TC.testLabels==TCpred.3)/length(TCpred.3)  
"Accuracy: Trinary, k=3"

## [1] "Accuracy: Trinary, k=3"

round(AccTC.3 \* 100, digits = 1)

## [1] 86.1

BCpred.5 <- knn(train = BC.train, test = BC.test, cl = BC.trainLabels, k=5)  
AccBC.5 <- sum(BC.testLabels==BCpred.5)/length(BCpred.5)  
"Accuracy: Binary, k=5"

## [1] "Accuracy: Binary, k=5"

round(AccBC.5 \* 100, digits = 1)

## [1] 96.5

TCpred.5 <- knn(train = TC.train, test = TC.test, cl = TC.trainLabels, k=5)  
AccTC.5 <- sum(TC.testLabels==TCpred.5)/length(TCpred.5)  
"Accuracy: Trinary, k=5"

## [1] "Accuracy: Trinary, k=5"

round(AccTC.5 \* 100, digits = 1)

## [1] 87

BCpred.10 <- knn(train = BC.train, test = BC.test, cl = BC.trainLabels, k=10)  
AccBC.10 <- sum(BC.testLabels==BCpred.10)/length(BCpred.10)  
"Accuracy: Binary, k=10"

## [1] "Accuracy: Binary, k=10"

round(AccBC.10 \* 100, digits = 1)

## [1] 95.9

TCpred.10 <- knn(train = TC.train, test = TC.test, cl = TC.trainLabels, k=10)  
AccTC.10 <- sum(TC.testLabels==TCpred.10)/length(TCpred.10)  
"Accuracy: Trinary, k=10"

## [1] "Accuracy: Trinary, k=10"

round(AccTC.10 \* 100, digits = 1)

## [1] 85.7

BCpred.15 <- knn(train = BC.train, test = BC.test, cl = BC.trainLabels, k=15)  
AccBC.15 <- sum(BC.testLabels==BCpred.15)/length(BCpred.15)  
"Accuracy: Binary, k=15"

## [1] "Accuracy: Binary, k=15"

round(AccBC.15 \* 100, digits = 1)

## [1] 96.1

TCpred.15 <- knn(train = TC.train, test = TC.test, cl = TC.trainLabels, k=15)  
AccTC.15 <- sum(TC.testLabels==TCpred.15)/length(TCpred.15)  
"Accuracy: Trinary, k=15"

## [1] "Accuracy: Trinary, k=15"

round(AccTC.15 \* 100, digits = 1)

## [1] 85.6

BCpred.20 <- knn(train = BC.train, test = BC.test, cl = BC.trainLabels, k=20)  
AccBC.20 <- sum(BC.testLabels==BCpred.20)/length(BCpred.20)  
"Accuracy: Binary, k=20"

## [1] "Accuracy: Binary, k=20"

round(AccBC.20 \* 100, digits = 1)

## [1] 95.9

TCpred.20 <- knn(train = TC.train, test = TC.test, cl = TC.trainLabels, k=20)  
AccTC.20 <- sum(TC.testLabels==TCpred.20)/length(TCpred.20)  
"Accuracy: Trinary, k=20"

## [1] "Accuracy: Trinary, k=20"

round(AccTC.20 \* 100, digits = 1)

## [1] 85

BCpred.25 <- knn(train = BC.train, test = BC.test, cl = BC.trainLabels, k=25)  
AccBC.25 <- sum(BC.testLabels==BCpred.25)/length(BCpred.25)  
"Accuracy: Binary, k=25"

## [1] "Accuracy: Binary, k=25"

round(AccBC.25 \* 100, digits = 1)

## [1] 95.7

TCpred.25 <- knn(train = TC.train, test = TC.test, cl = TC.trainLabels, k=25)  
AccTC.25 <- sum(TC.testLabels==TCpred.25)/length(TCpred.25)  
"Accuracy: Trinary, k=25"

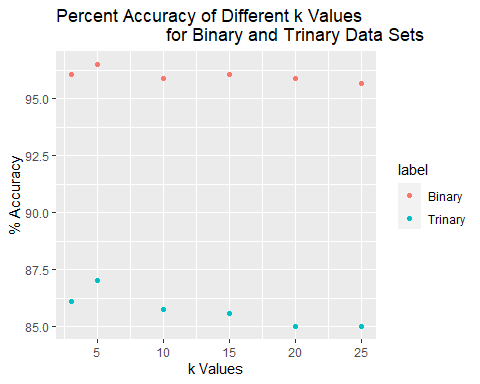
## [1] "Accuracy: Trinary, k=25"

round(AccTC.25 \* 100, digits = 1)

## [1] 85

#### Plot the results in a graph where the x-axis is the different values of k and the y-axis is the accuracy of the model.

## Build the data frame  
kValues <- c(3, 5, 10, 15, 20, 25)  
  
# Binary data  
pctAcc <- c(AccBC.3, AccBC.5, AccBC.10, AccBC.15, AccBC.20, AccBC.25) \* 100  
label <- c("Binary")  
BCAccSet <- data.frame(label, kValues, pctAcc)  
  
# Trinary data  
pctAcc <- c(AccTC.3, AccTC.5, AccTC.10, AccTC.15, AccTC.20, AccTC.25) \* 100  
label <- c("Trinary")  
TCAccSet <- data.frame(label, kValues, pctAcc)  
  
# Combine data sets  
dataSet <- rbind(BCAccSet, TCAccSet)  
  
# Plot the data  
ggplot(data = dataSet, mapping = aes(kValues, pctAcc, color = label)) +   
 geom\_point() + labs(x = "k Values", y = "% Accuracy",   
 title = "Percent Accuracy of Different k Values   
 for Binary and Trinary Data Sets")



### c. Looking back at the plots of the data, do you think a linear classifier would work well on these datasets?

Looking at the plots of the Binary and Trinary data sets, it seems pretty obvious that linear classifiers would not be useful in determining to which groups the individual data points would belong. They are clustered together, which is helpful in the nearest neighbors algorithm, but the clusters are too interspersed with each other to be divided by any sort of reasonable line.