> knitr::opts\_chunk$set(echo = TRUE)

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

> setwd("/Users/binayprasannajena/Documents/GitHub/dsc520/")

The working directory was changed to /Users/binayprasannajena/Documents/GitHub/dsc520 inside a notebook chunk. The working directory will be reset when the chunk is finished running. Use the knitr root.dir option in the setup chunk to change the working directory for notebook chunks.> ## Load the `caTools` library

> library(caTools)

> ## Load the `data/binary-classifier-data.csv` as df

> binary\_classifier\_df <- read.csv("data/binary-classifier-data.csv")

> head(binary\_classifier\_df)

> summary(binary\_classifier\_df)

label x y

Min. :0.000 Min. : -5.20 Min. : -4.019

1st Qu.:0.000 1st Qu.: 19.77 1st Qu.: 21.207

Median :0.000 Median : 41.76 Median : 44.632

Mean :0.488 Mean : 45.07 Mean : 45.011

3rd Qu.:1.000 3rd Qu.: 66.39 3rd Qu.: 68.698

Max. :1.000 Max. :104.58 Max. :106.896

> ## Load the `data/trinary-classifier-data.csv` as df

> trinary\_classifier\_df <- read.csv("data/trinary-classifier-data.csv")

> head(binary\_classifier\_df)

> summary(binary\_classifier\_df)

label x y

Min. :0.000 Min. : -5.20 Min. : -4.019

1st Qu.:0.000 1st Qu.: 19.77 1st Qu.: 21.207

Median :0.000 Median : 41.76 Median : 44.632

Mean :0.488 Mean : 45.07 Mean : 45.011

3rd Qu.:1.000 3rd Qu.: 66.39 3rd Qu.: 68.698

Max. :1.000 Max. :104.58 Max. :106.896

>

> # Since label is number converting to factors, so that it becomes categorical

> binary\_classifier\_df$label <- as.factor(binary\_classifier\_df$label)

> trinary\_classifier\_df$label <- as.factor(trinary\_classifier\_df$label)

> library(ggplot2)

> ggplot(data = binary\_classifier\_df, aes(y = y, x = x, color = label)) +

+ geom\_point() + ggtitle("Actual Data - Binary Classifier")

> ggplot(data = trinary\_classifier\_df, aes(y = y, x = x, color = label)) +

+ geom\_point() + ggtitle("Actual Data - Trinary Classifier")

> # Splitting the dataset for the model into train and test datasets.

> myData\_binary <- sample.split(binary\_classifier\_df$label, SplitRatio=0.8)

> train\_binary <- subset(binary\_classifier\_df, myData\_binary==TRUE)

> test\_binary <- subset(binary\_classifier\_df, myData\_binary==FALSE)

> myData\_trinary <- sample.split(trinary\_classifier\_df$label, SplitRatio=0.8)

> train\_trinary <- subset(trinary\_classifier\_df, myData\_trinary==TRUE)

> test\_trinary <- subset(trinary\_classifier\_df, myData\_trinary==FALSE)

> # 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)

> library(class)

> K\_Binary\_Values <- c()

> Accuracy\_Binary <- c()

> error.rate.b <- c()

> # Running for multiple K Values

> for(i in 1:25){

+ knnmodel.b.i <- knn(train\_binary[2:3],test\_binary[2:3],k=i,cl=train\_binary$label)

+ table.b.i <- table(knnmodel.b.i,test\_binary$label)

+ accur = accuracy(table.b.i)

+ print(paste("Accuracy for Binary Classifier Data Model with K=", i ," is ", accuracy(table.b.i)))

+ K\_Binary\_Values <- c(K\_Binary\_Values, i)

+ Accuracy\_Binary<- c(Accuracy\_Binary, accur)

+ error.rate.b <- c(error.rate.b, mean(test\_binary$label != knnmodel.b.i))

+ }

[1] "Accuracy for Binary Classifier Data Model with K= 1 is 96.3210702341137"

[1] "Accuracy for Binary Classifier Data Model with K= 2 is 96.3210702341137"

[1] "Accuracy for Binary Classifier Data Model with K= 3 is 96.989966555184"

[1] "Accuracy for Binary Classifier Data Model with K= 4 is 96.6555183946488"

[1] "Accuracy for Binary Classifier Data Model with K= 5 is 96.989966555184"

[1] "Accuracy for Binary Classifier Data Model with K= 6 is 96.989966555184"

[1] "Accuracy for Binary Classifier Data Model with K= 7 is 96.6555183946488"

[1] "Accuracy for Binary Classifier Data Model with K= 8 is 97.3244147157191"

[1] "Accuracy for Binary Classifier Data Model with K= 9 is 96.989966555184"

[1] "Accuracy for Binary Classifier Data Model with K= 10 is 96.989966555184"

[1] "Accuracy for Binary Classifier Data Model with K= 11 is 96.989966555184"

[1] "Accuracy for Binary Classifier Data Model with K= 12 is 97.3244147157191"

[1] "Accuracy for Binary Classifier Data Model with K= 13 is 96.6555183946488"

[1] "Accuracy for Binary Classifier Data Model with K= 14 is 96.6555183946488"

[1] "Accuracy for Binary Classifier Data Model with K= 15 is 96.3210702341137"

[1] "Accuracy for Binary Classifier Data Model with K= 16 is 95.6521739130435"

[1] "Accuracy for Binary Classifier Data Model with K= 17 is 95.9866220735786"

[1] "Accuracy for Binary Classifier Data Model with K= 18 is 95.9866220735786"

[1] "Accuracy for Binary Classifier Data Model with K= 19 is 96.3210702341137"

[1] "Accuracy for Binary Classifier Data Model with K= 20 is 95.9866220735786"

[1] "Accuracy for Binary Classifier Data Model with K= 21 is 95.6521739130435"

[1] "Accuracy for Binary Classifier Data Model with K= 22 is 95.6521739130435"

[1] "Accuracy for Binary Classifier Data Model with K= 23 is 95.6521739130435"

[1] "Accuracy for Binary Classifier Data Model with K= 24 is 95.3177257525084"

[1] "Accuracy for Binary Classifier Data Model with K= 25 is 95.9866220735786"

> print(K\_Binary\_Values)

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

> print(Accuracy\_Binary)

[1] 96.32107 96.32107 96.98997 96.65552 96.98997 96.98997 96.65552 97.32441 96.98997 96.98997 96.98997

[12] 97.32441 96.65552 96.65552 96.32107 95.65217 95.98662 95.98662 96.32107 95.98662 95.65217 95.65217

[23] 95.65217 95.31773 95.98662

> print(error.rate.b\*100)

[1] 3.678930 3.678930 3.010033 3.344482 3.010033 3.010033 3.344482 2.675585 3.010033 3.010033 3.010033

[12] 2.675585 3.344482 3.344482 3.678930 4.347826 4.013378 4.013378 3.678930 4.013378 4.347826 4.347826

[23] 4.347826 4.682274 4.013378

> print(Accuracy\_Binary+error.rate.b\*100)

[1] 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100

> accuracy\_binary\_df <- data.frame(K\_Binary\_Values, Accuracy\_Binary, error.rate.b\*100)

> accuracy\_binary\_df

> K\_Trinary\_Values <- c()

> Accuracy\_Trinary <- c()

> error.rate.t <- c()

> # Running for multiple K Values

> for(i in 1:25){

+ knnmodel.t.i <- knn(train\_trinary[2:3],test\_trinary[2:3],k=i,cl=train\_trinary$label)

+ table.t.i <- table(knnmodel.t.i,test\_trinary$label)

+ accur = accuracy(table.t.i)

+ print(paste("Accuracy for Trinary Classifier Data Model with K=", i ," is ", accur))

+ K\_Trinary\_Values <- c(K\_Trinary\_Values, i)

+ Accuracy\_Trinary<- c(Accuracy\_Trinary, accur)

+ error.rate.t <- c(error.rate.t, mean(test\_trinary$label != knnmodel.t.i))

+ }

[1] "Accuracy for Trinary Classifier Data Model with K= 1 is 87.5399361022364"

[1] "Accuracy for Trinary Classifier Data Model with K= 2 is 88.8178913738019"

[1] "Accuracy for Trinary Classifier Data Model with K= 3 is 90.0958466453674"

[1] "Accuracy for Trinary Classifier Data Model with K= 4 is 88.8178913738019"

[1] "Accuracy for Trinary Classifier Data Model with K= 5 is 89.4568690095847"

[1] "Accuracy for Trinary Classifier Data Model with K= 6 is 87.8594249201278"

[1] "Accuracy for Trinary Classifier Data Model with K= 7 is 88.1789137380192"

[1] "Accuracy for Trinary Classifier Data Model with K= 8 is 88.1789137380192"

[1] "Accuracy for Trinary Classifier Data Model with K= 9 is 87.5399361022364"

[1] "Accuracy for Trinary Classifier Data Model with K= 10 is 88.8178913738019"

[1] "Accuracy for Trinary Classifier Data Model with K= 11 is 86.9009584664537"

[1] "Accuracy for Trinary Classifier Data Model with K= 12 is 87.5399361022364"

[1] "Accuracy for Trinary Classifier Data Model with K= 13 is 86.9009584664537"

[1] "Accuracy for Trinary Classifier Data Model with K= 14 is 86.5814696485623"

[1] "Accuracy for Trinary Classifier Data Model with K= 15 is 87.8594249201278"

[1] "Accuracy for Trinary Classifier Data Model with K= 16 is 88.1789137380192"

[1] "Accuracy for Trinary Classifier Data Model with K= 17 is 88.1789137380192"

[1] "Accuracy for Trinary Classifier Data Model with K= 18 is 87.5399361022364"

[1] "Accuracy for Trinary Classifier Data Model with K= 19 is 88.1789137380192"

[1] "Accuracy for Trinary Classifier Data Model with K= 20 is 87.8594249201278"

[1] "Accuracy for Trinary Classifier Data Model with K= 21 is 88.1789137380192"

[1] "Accuracy for Trinary Classifier Data Model with K= 22 is 88.4984025559105"

[1] "Accuracy for Trinary Classifier Data Model with K= 23 is 88.1789137380192"

[1] "Accuracy for Trinary Classifier Data Model with K= 24 is 88.4984025559105"

[1] "Accuracy for Trinary Classifier Data Model with K= 25 is 88.8178913738019"

> print(K\_Trinary\_Values)

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

> print(Accuracy\_Trinary)

[1] 87.53994 88.81789 90.09585 88.81789 89.45687 87.85942 88.17891 88.17891 87.53994 88.81789 86.90096

[12] 87.53994 86.90096 86.58147 87.85942 88.17891 88.17891 87.53994 88.17891 87.85942 88.17891 88.49840

[23] 88.17891 88.49840 88.81789

> print(error.rate.t\*100)

[1] 12.460064 11.182109 9.904153 11.182109 10.543131 12.140575 11.821086 11.821086 12.460064 11.182109

[11] 13.099042 12.460064 13.099042 13.418530 12.140575 11.821086 11.821086 12.460064 11.821086 12.140575

[21] 11.821086 11.501597 11.821086 11.501597 11.182109

> print(Accuracy\_Trinary+error.rate.t\*100)

[1] 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100

> accuracy\_trinary\_df <- data.frame(K\_Trinary\_Values, Accuracy\_Trinary, error.rate.t\*100)

> accuracy\_trinary\_df

> library(ggplot2)

> ggplot(data = accuracy\_binary\_df, aes(y = Accuracy\_Binary, x = K\_Binary\_Values)) +

+ geom\_line() + ggtitle("Accuracy vs K Value - Binary Classifier") +

+ ylab("Accuracy %") + xlab("K")

> # Inverted Plot of Above

> ggplot(data = accuracy\_binary\_df, aes(y = error.rate.b, x = K\_Binary\_Values)) +

+ geom\_line() + ggtitle("Error Rate vs K Value - Binary Classifier") +

+ ylab("Error Rate %") + xlab("K")

> ggplot(data = accuracy\_trinary\_df, aes(y = Accuracy\_Trinary, x = K\_Trinary\_Values)) +

+ geom\_line() + ggtitle("Accuracy vs K Value - Trinary Classifier") +

+ ylab("Accuracy %") + xlab("K")

> # Inverted Plot of Above

> ggplot(data = accuracy\_trinary\_df, aes(y = error.rate.t, x = K\_Trinary\_Values)) +

+ geom\_line() + ggtitle("Error Rate vs K Value - Trinary Classifier") +

+ ylab("Error Rate %") + xlab("K")

Restarting R session...

> knitr::opts\_chunk$set(echo = TRUE)

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

> setwd("C:/git-bellevue/dsc520-fork")

Error in setwd("C:/git-bellevue/dsc520-fork") :

cannot change working directory

Restarting R session...

> knitr::opts\_chunk$set(echo = TRUE)

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

> setwd("/Users/binayprasannajena/Documents/GitHub/dsc520/")

The working directory was changed to /Users/binayprasannajena/Documents/GitHub/dsc520 inside a notebook chunk. The working directory will be reset when the chunk is finished running. Use the knitr root.dir option in the setup chunk to change the working directory for notebook chunks.> ## Load the `caTools` library

> library(caTools)

> ## Load the `data/clustering-data.csv` to

> clustering\_df <- read.csv("data/clustering-data.csv")

> head(clustering\_df)

> summary(clustering\_df)

x y

Min. : 0.0 Min. :134.0

1st Qu.: 56.0 1st Qu.:141.0

Median : 82.0 Median :154.0

Mean :109.6 Mean :175.7

3rd Qu.:180.0 3rd Qu.:218.0

Max. :249.0 Max. :236.0

>

> library(ggplot2)

> ggplot(data = clustering\_df, aes(y = y, x = x)) +

+ geom\_point() + ggtitle("Actual Data - Clustering Data")

> set.seed(100)

> k\_values <- c()

> tot.withinss\_values <- c()

> errors <- c()

> for(i in 2:12){

+ # Read Once and Mapping the same object Multiple Times

+ df <- clustering\_df

+ df.cluster <- kmeans(df, i)

+ df$cluster <- as.factor(df.cluster$cluster)

+ p <- ggplot(data = df,

+ aes(x = x,

+ y = y,

+ color = cluster)) +

+ geom\_point(size = 0.5) +

+ geom\_point(data = as.data.frame(df.cluster$centers),

+ color = "black",

+ shape = 10,

+ size = 2) +

+ ggtitle(paste("K-Means Cluster Plot for K = ", i, sep ="")) +

+ theme\_bw()

+ print(p)

+ # For Later Analysis & Plot

+ k\_values<- c(k\_values, i)

+ tot.withinss\_values <- c(tot.withinss\_values, df.cluster$tot.withinss)

+ x.dist <- df.cluster$centers[df$cluster] - df$x

+ y.dist <- df.cluster$centers[as.numeric(df$cluster) + i] - df$y

+ tot.dist <- sqrt((x.dist \*\* 2) + (y.dist \*\* 2))

+ errors <- c(errors, mean(tot.dist))

+ }

> elbow\_df <- data.frame(k\_values, tot.withinss\_values, errors)

> elbow\_df

> ggplot(data = elbow\_df, aes(x = k\_values, y = errors)) +

+ geom\_line(color = "red") +

+ ggtitle("Average Distance from the Center for each K-Mean") +

+ xlab("K-value") + ylab("Error") + xlim(2,12)

> # Also Plotting tot.withinss, as mentioned in the referenced article

> ggplot(data = elbow\_df, aes(x = k\_values, y = tot.withinss\_values)) +

+ geom\_line(color = "red") +

+ ggtitle("tot.withinss for each K-Mean") +

+ xlab("K-value") + ylab("tot.withinss") + xlim(2,12)