HW2 - Splitting Data and K Means

2023-01-21

Question 3.1

I am creating Training, Validation and Testing data from the "credit_card_data-headers.txt" data set. I decided to split the data into 60% Training, 20% Validation, and 20% Testing.

```
# Question 3.1
setwd("C:\\Users\\anoop\\OneDrive\\Documents\\GA Analytics\\ISYE6501\\Homework 2")
data <- read.delim("data 3.1\\credit_card_data-headers.txt",
    header = TRUE)

# Splitting data into 60% training and 40% non training
dt <- sort(sample(nrow(data), nrow(data) * 0.6))
train <- data[dt, ]
nontrain <- data[-dt, ]

# Splitting nontraining data into validation and test data
# (can use nontrain as test data is validation isn't
# needed)
dt2 <- sort(sample(nrow(nontrain), nrow(nontrain) * 0.5))
val <- nontrain[dt2, ]
test <- nontrain[-dt2, ]
```

```
library(kknn)
acc = rep(0, 29)
# Training KNN Model with different K values
for (k in 1:30) {
    knn_model <- kknn(R1 ~ ., train, val, k = k, scale = TRUE)
    pred <- as.integer(fitted(knn_model) + 0.5)
    acc[k] = sum(pred == val$R1)/nrow(val)

}
acc
```

```
##  [1] 0.7557252 0.7557252 0.7557252 0.7557252 0.8167939 0.8244275 0.8167939
##  [8] 0.8167939 0.8167939 0.8396947 0.8396947 0.8396947 0.8396947 0.8396947
## [15] 0.8396947 0.8396947 0.8396947 0.8396947 0.8396947 0.8396947 0.8396947
## [22] 0.8396947 0.8320611 0.8320611 0.8320611 0.8320611 0.8320611 0.8320611
## [29] 0.8320611 0.8320611
```

```
cat("The K Value with the Highest Accuracy: ", which.max(acc),
    " with an accuracy of (%): ", max(acc))
```

```
## The K Value with the Highest Accuracy:  10  with an accuracy of (%):  0.8396947
```

```r
# Testing accuracy using test data from model chosen from
# above
knn_model <- kknn(R1 ~ ., train, test, k = which.max(acc), scale = TRUE)
pred <- as.integer(fitted(knn_model) + 0.5)
cat("The accuracy using the test data is: ", sum(pred == test$R1)/nrow(test))
```

```
## The accuracy using the test data is:  0.8778626
```

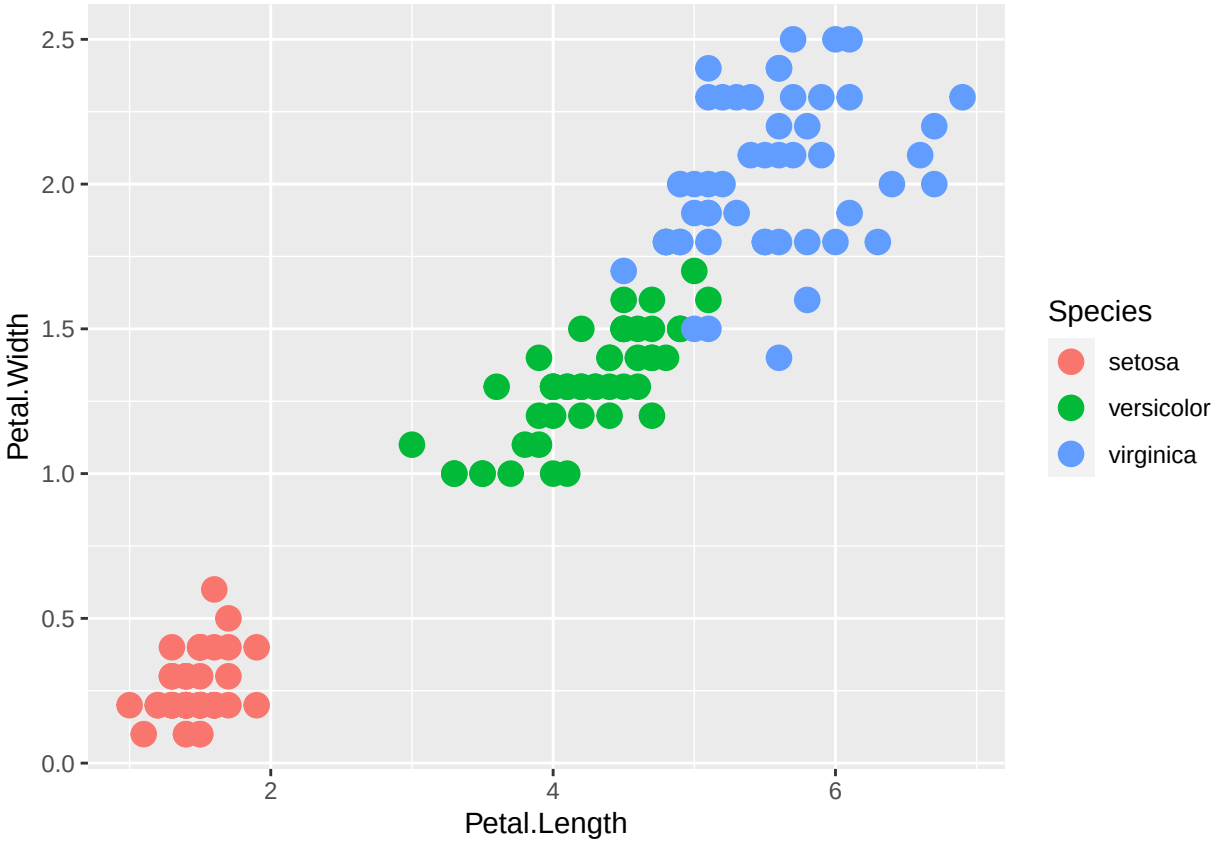
Question 4.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a clustering model would be appropriate. List some (up to 5) predictors that you might use.

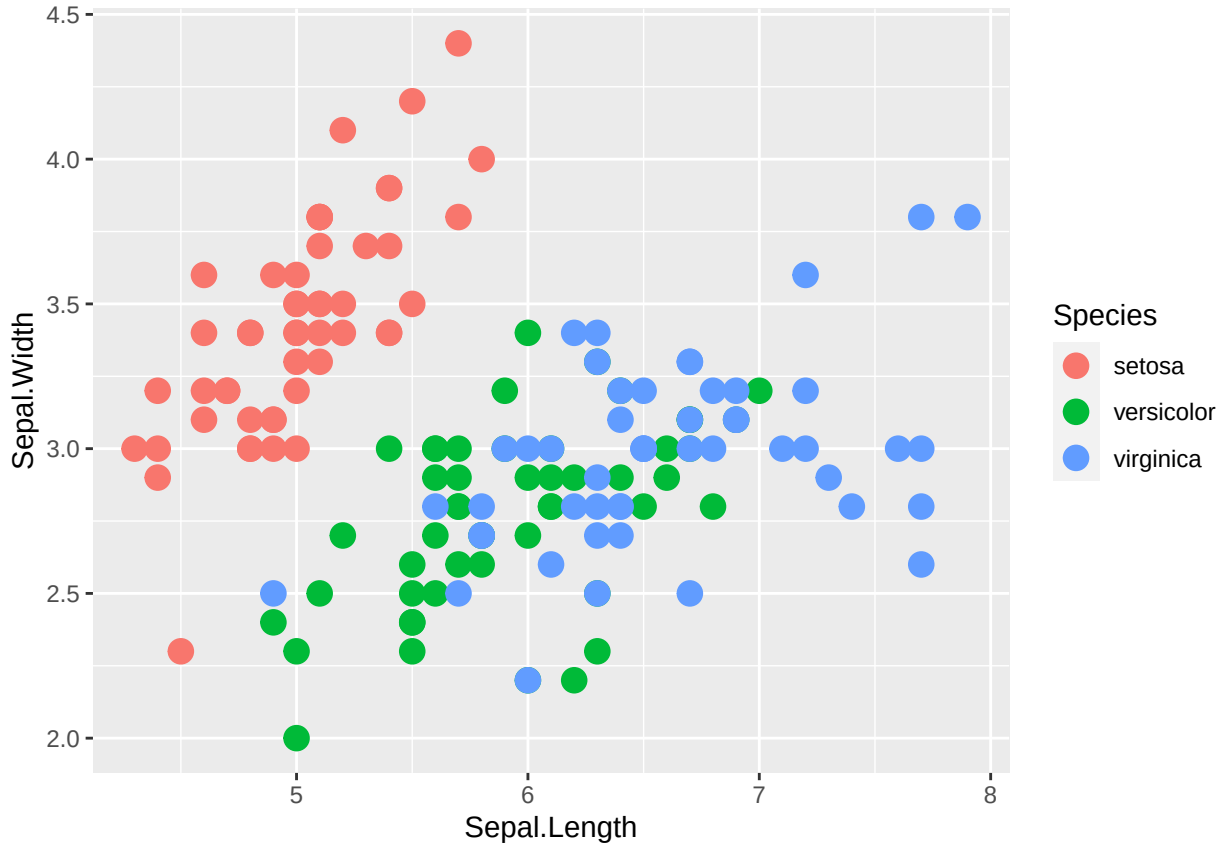
```r
# My team at work needs to determine what user-created data
# needs to be deleted from our databases as we are nearing
# storage capacity. As users are creating data on the
# daily, we are not able to easily identify which data is
# needed to be deleted unless we ask each individual user
# (there are 1000+ employees creating data on the daily). I
# could use a Clustering Model to help group which data
# could be targeted for deletion rather than blindly asking
# users if their data can be removed. Some factors I could
# consider are: Age of Creation for Data, Date of Data last
# accessed, Frequency of Access, and Size of the Data
```

Question 4.2

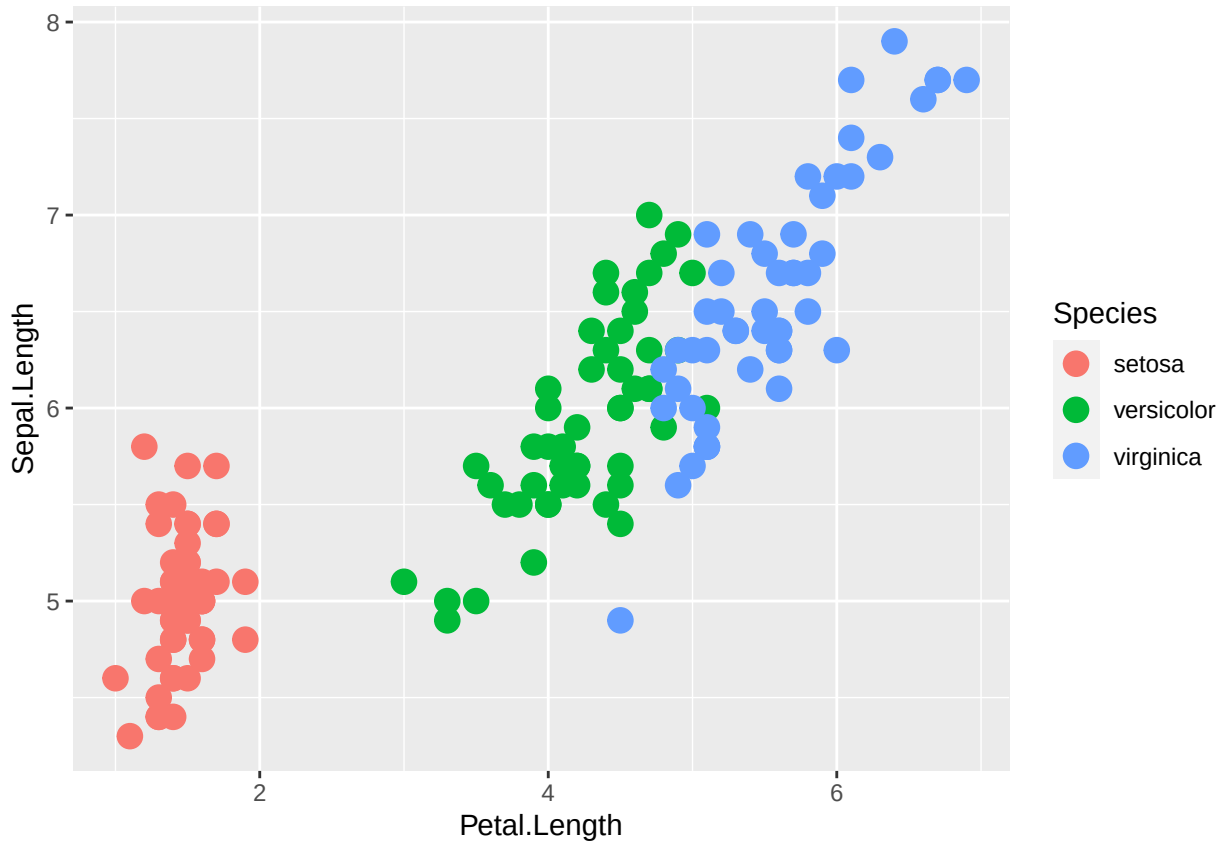
```r
# R has the Iris dataset, so there isn't any need to read
# the data file.
data2 <- iris

library(ggplot2)
# install.packages('ClusterR')
library(ClusterR)

# By comparing different predictors in scatter plot graphs,
# it looks like Petal.Length and Petal.Width are great
# predictors for clustering. It also looks like 3 clusters
# would be perfect for K Means
ggplot(data2, aes(Petal.Length, Petal.Width)) + geom_point(aes(col = Species),
    size = 4)
```




```
ggplot(data2, aes(Sepal.Length, Sepal.Width)) + geom_point(aes(col = Species),
    size = 4)
```




```
ggplot(data2, aes(Petal.Length, Sepal.Length)) + geom_point(aes(col = Species),
    size = 4)
```

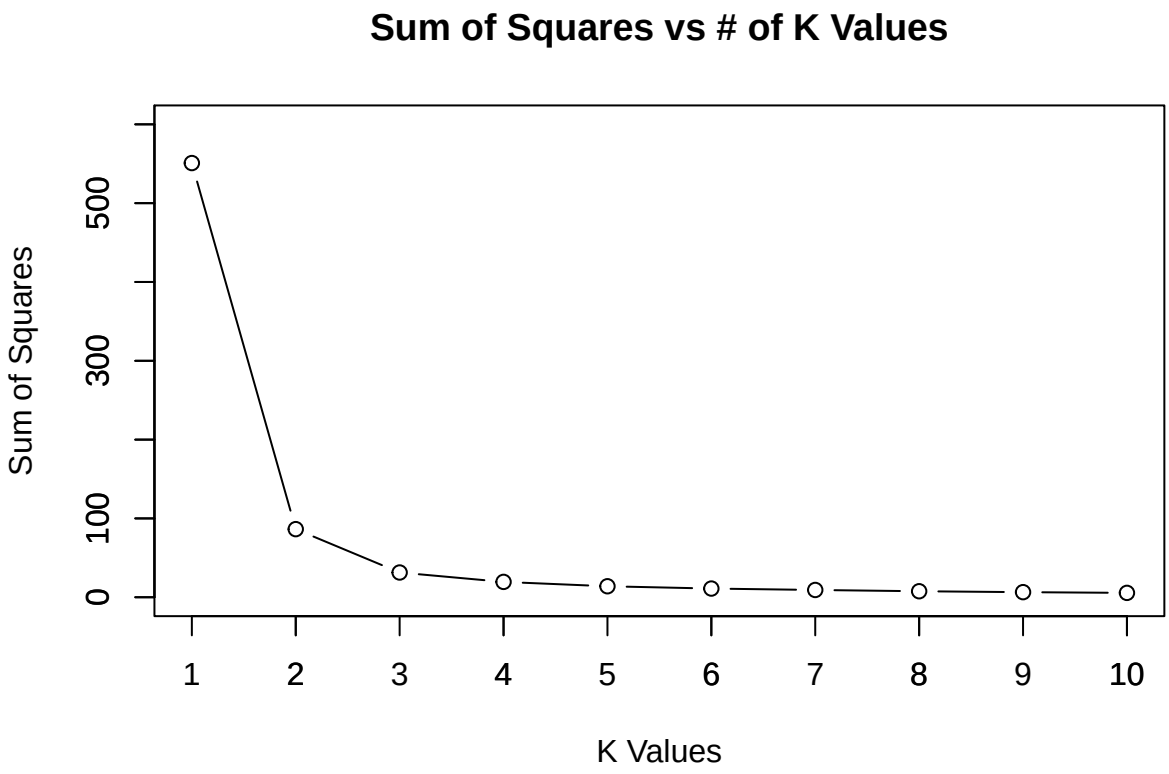



```
ggplot(data2, aes(Petal.Width, Sepal.Width)) + geom_point(aes(col = Species),
    size = 4)
```


```
tot_withinss = rep(0, 9)
for (i in 1:10) {
    # using Petal Length and Width for Predictors
    model_km <- kmeans(data2[, 3:4], centers = i, nstart = 25)

    # tot.withinss -> Total within-cluster sum of squares
    tot_withinss[i] = model_km$tot.withinss
}
tot_withinss
```

```
##  [1] 550.895333  86.390220  31.371359  19.465989  13.916909  11.025145
##  [7]   9.236596   7.615402   6.456495   5.528149
```

```
# Plotting the K Values against the Sum of Squares
# (tot.withinss)
plot(seq(1, 10, 1), tot_withinss, type = "b", main = "Sum of Squares vs # of K Values",
    xlab = "K Values", ylab = "Sum of Squares", ylim = c(0, 600))
axis(side = 1, at = seq(1, 10, 1))
axis(side = 2, at = seq(0, 900, 100))
```




```
# While the plots above indicate that the ideal cluster
# value should be 3, the Sum of Square vs # of K Values
# plot also support for the model to have 3 clusters.

model_km2 <- kmeans(data2[, 3:4], centers = 3, nstart = 25)
acc_table <- table(model_km2$cluster, data2$Species)
acc_table
```

```
##
##     setosa versicolor virginica
##   1      0          2        46
##   2      0         48         4
##   3     50          0         0
```

```
model_km2
```

```
## K-means clustering with 3 clusters of sizes 48, 52, 50
##
## Cluster means:
##   Petal.Length Petal.Width
## 1     5.595833    2.037500
## 2     4.269231    1.342308
## 3     1.462000    0.246000
##
## Clustering vector:
```

```
##   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [38] 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [75] 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
## [149] 1 1
##
## Within cluster sum of squares by cluster:
## [1] 16.29167 13.05769  2.02200
##  (between_SS / total_SS =  94.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
# Using 3 Centers and Petal Length and Width as predictors,
# the model is able to accurately (Accuracy of 94.3%)
# categorize the flowers.
```