# R Programming: K-Means Clustering

## Example   
# ---  
# Question: Perform clustering analysis on the following dataset using the K-Means clustering algorithm.  
# ---  
#   
  
# Load and view the dataset   
# ---  
# Importing the dataset  
# ---  
#   
require("datasets")  
  
# Loading the Iris dataset  
# ---  
#   
data("iris")  
  
# Viewing the structure of the dataset  
# ---  
#  
str(iris)

## 'data.frame': 150 obs. of 5 variables:  
## $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...  
## $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...  
## $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...  
## $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...  
## $ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...

# Viewing the statistical summary of the dataset  
# ---  
#   
summary(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width   
## Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100   
## 1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300   
## Median :5.800 Median :3.000 Median :4.350 Median :1.300   
## Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199   
## 3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800   
## Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500   
## Species   
## setosa :50   
## versicolor:50   
## virginica :50   
##   
##   
##

# Previewing the dataset  
# ---  
#   
head(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 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 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa

# Preprocessing the dataset  
# ---  
# Since clustering is a type of Unsupervised Learning,   
# we would not require Class Label(output) during execution of our algorithm.   
# We will, therefore, remove Class Attribute âSpeciesâ and store it in another variable.   
# We would then normalize the attributes between 0 and 1 using our own function.  
# ---  
#  
iris.new<- iris[, c(1, 2, 3, 4)]  
iris.class<- iris[, "Species"]  
head(iris.new)

## Sepal.Length Sepal.Width Petal.Length Petal.Width  
## 1 5.1 3.5 1.4 0.2  
## 2 4.9 3.0 1.4 0.2  
## 3 4.7 3.2 1.3 0.2  
## 4 4.6 3.1 1.5 0.2  
## 5 5.0 3.6 1.4 0.2  
## 6 5.4 3.9 1.7 0.4

# Previewing the class column  
# ---  
#   
head(iris.class)

## [1] setosa setosa setosa setosa setosa setosa  
## Levels: setosa versicolor virginica

# Normalizing the dataset so that no particular attribute   
# has more impact on clustering algorithm than others.  
# ---  
#   
normalize <- function(x){  
 return ((x-min(x)) / (max(x)-min(x)))  
}  
  
iris.new$Sepal.Length<- normalize(iris.new$Sepal.Length)  
iris.new$Sepal.Width<- normalize(iris.new$Sepal.Width)  
iris.new$Petal.Length<- normalize(iris.new$Petal.Length)  
iris.new$Petal.Width<- normalize(iris.new$Petal.Width)  
head(iris.new)

## Sepal.Length Sepal.Width Petal.Length Petal.Width  
## 1 0.22222222 0.6250000 0.06779661 0.04166667  
## 2 0.16666667 0.4166667 0.06779661 0.04166667  
## 3 0.11111111 0.5000000 0.05084746 0.04166667  
## 4 0.08333333 0.4583333 0.08474576 0.04166667  
## 5 0.19444444 0.6666667 0.06779661 0.04166667  
## 6 0.30555556 0.7916667 0.11864407 0.12500000

# Applying the K-means clustering algorithm with no. of centroids(k)=3  
# ---  
#   
result<- kmeans(iris.new,3)   
  
# Previewing the no. of records in each cluster  
#   
result$size

## [1] 21 96 33

# Getting the value of cluster center datapoint value(3 centers for k=3)  
# ---  
#   
result$centers

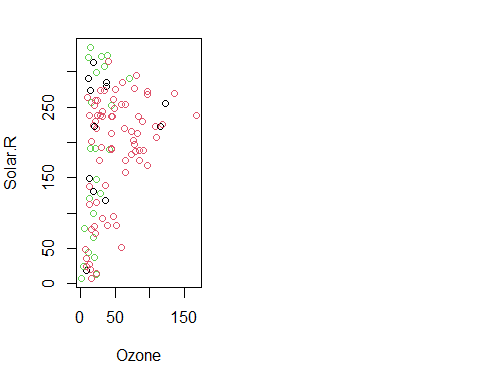
## Sepal.Length Sepal.Width Petal.Length Petal.Width  
## 1 0.1243386 0.3730159 0.12832930 0.1051587  
## 2 0.5596065 0.3732639 0.67355226 0.6679688  
## 3 0.2415825 0.6792929 0.08371854 0.0719697

# Getting the cluster vector that shows the cluster where each record falls  
# ---  
#   
result$cluster

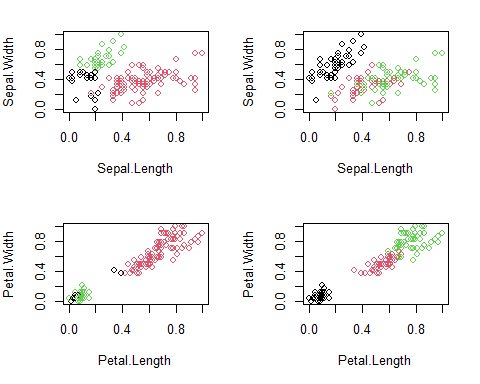
## [1] 3 1 1 1 3 3 3 3 1 1 3 3 1 1 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 1 1 3 3 3 1 1 3  
## [38] 3 1 3 3 1 1 3 3 1 3 1 3 3 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [75] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2  
## [112] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [149] 2 2

# The graph shows that we have got 3 clearly distinguishable clusters for Ozone and Solar.R data points.  
# Letâs see how clustering has performed on Wind and Temp attributes.

# Visualizing the clustering results  
# ---  
#   
par(mfrow = c(1,2), mar = c(5,4,2,2))  
  
# Plotting to see how Ozone and Solar.R data points have been distributed in clusters  
# ---  
#  
plot(airquality[,1:2], col = result$cluster)



# Verifying the results of clustering  
# ---  
#   
par(mfrow = c(2,2), mar = c(5,4,2,2))  
  
# Plotting to see how Sepal.Length and Sepal.Width data points have been distributed in clusters  
plot(iris.new[c(1,2)], col = result$cluster)  
  
# Plotting to see how Sepal.Length and Sepal.Width data points have been distributed   
# originally as per "class" attribute in dataset  
# ---  
#  
plot(iris.new[c(1,2)], col = iris.class)  
  
# Plotting to see how Petal.Length and Petal.Width data points have been distributed in clusters  
# ---  
#   
plot(iris.new[c(3,4)], col = result$cluster)  
plot(iris.new[c(3,4)], col = iris.class)



# Result of table shows that Cluster 1 corresponds to Virginica,   
# Cluster 2 corresponds to Versicolor and Cluster 3 to Setosa.  
# ---  
#   
table(result$cluster, iris.class)

## iris.class  
## setosa versicolor virginica  
## 1 17 4 0  
## 2 0 46 50  
## 3 33 0 0

In order to improve this accuracy further, we may try different values of âkâ. In some cases, it is also beneficial to change the algorithm in case k-means is unable to yield good results.

## Challenge 1  
# ---  
# Question: Apply unsupervised learning to the given airquality dataset below.  
# ---  
#   
#   
  
# Load and view the dataset   
# ---  
# Importing the dataset  
# ---  
#   
data("airquality")  
str(airquality)

## 'data.frame': 153 obs. of 6 variables:  
## $ Ozone : int 41 36 12 18 NA 28 23 19 8 NA ...  
## $ Solar.R: int 190 118 149 313 NA NA 299 99 19 194 ...  
## $ Wind : num 7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...  
## $ Temp : int 67 72 74 62 56 66 65 59 61 69 ...  
## $ Month : int 5 5 5 5 5 5 5 5 5 5 ...  
## $ Day : int 1 2 3 4 5 6 7 8 9 10 ...

head(airquality)

## Ozone Solar.R Wind Temp Month Day  
## 1 41 190 7.4 67 5 1  
## 2 36 118 8.0 72 5 2  
## 3 12 149 12.6 74 5 3  
## 4 18 313 11.5 62 5 4  
## 5 NA NA 14.3 56 5 5  
## 6 28 NA 14.9 66 5 6

# The output shows that only Ozone and Solar.R attributes have NA i.e. some missing value.  
# Impute monthly mean in Ozone by running the code shown below  
for (i in 1:nrow(airquality)){  
 if(is.na(airquality[i,"Ozone"])){  
 airquality[i,"Ozone"]<- mean(airquality[which(airquality[,"Month"]==airquality[i,"Month"]),"Ozone"],na.rm = TRUE)  
 }}

# Impute monthly mean in Solar.R  
# ---  
for (i in 1:nrow(airquality)){  
 if(is.na(airquality[i,"Solar.R"])){  
 airquality[i,"Solar.R"]<-mean(airquality[which(airquality[,"Month"]==airquality[i,"Month"]),"Solar.R"],na.rm = TRUE)  
 }}

airquality.new<- airquality[, c(1, 2, 3, 4,6)]  
airquality.class<- airquality[, "Month"]  
head(airquality.new)

## Ozone Solar.R Wind Temp Day  
## 1 41.00000 190.0000 7.4 67 1  
## 2 36.00000 118.0000 8.0 72 2  
## 3 12.00000 149.0000 12.6 74 3  
## 4 18.00000 313.0000 11.5 62 4  
## 5 23.61538 181.2963 14.3 56 5  
## 6 28.00000 181.2963 14.9 66 6

# Previewing the class column  
# ---  
#   
head(airquality.class)

## [1] 5 5 5 5 5 5

# Normalizing the dataset so that no particular attribute   
# has more impact on clustering algorithm than others.  
# ---  
#   
normalize <- function(x){  
 return ((x-min(x)) / (max(x)-min(x)))  
}  
  
airquality.new$Ozone<- normalize(airquality.new$Ozone)  
airquality.new$Solar.R<- normalize(airquality.new$Solar.R)  
airquality.new$Wind<- normalize(airquality.new$Wind)  
airquality.new$Temp<- normalize(airquality.new$Temp)  
airquality.new$Day<- normalize(airquality.new$Day)  
head(airquality.new)

## Ozone Solar.R Wind Temp Day  
## 1 0.23952096 0.5596330 0.3000000 0.2682927 0.00000000  
## 2 0.20958084 0.3394495 0.3315789 0.3902439 0.03333333  
## 3 0.06586826 0.4342508 0.5736842 0.4390244 0.06666667  
## 4 0.10179641 0.9357798 0.5157895 0.1463415 0.10000000  
## 5 0.13542146 0.5330162 0.6631579 0.0000000 0.13333333  
## 6 0.16167665 0.5330162 0.6947368 0.2439024 0.16666667

# Applying the K-means clustering algorithm with no. of centroids(k)=3  
# ---  
#   
result1<- kmeans(airquality.new,3)   
  
# Previewing the no. of records in each cluster  
#   
result1$size

## [1] 57 52 44

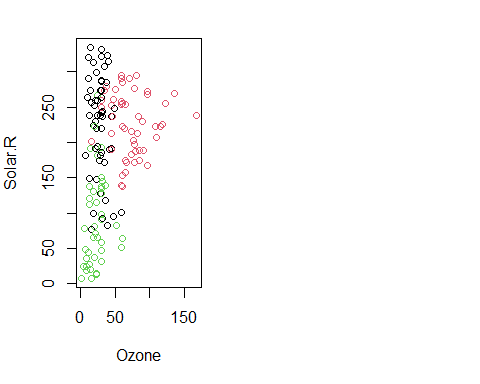
# Getting the value of cluster center datapoint value(3 centers for k=3)  
# ---  
#   
result1$centers

## Ozone Solar.R Wind Temp Day  
## 1 0.1547795 0.6619318 0.5050785 0.4741121 0.2853801  
## 2 0.4246477 0.6725308 0.3218623 0.7335835 0.5378205  
## 3 0.1274164 0.2461980 0.4765550 0.3747228 0.7106061

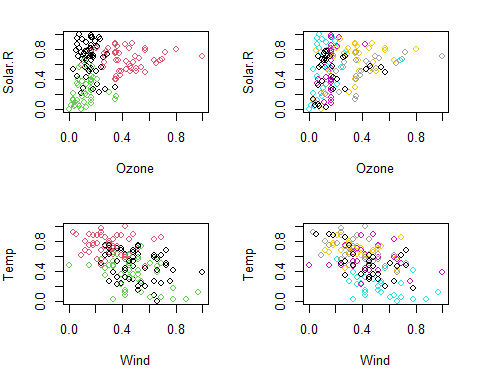
# Getting the cluster vector that shows the cluster where each record falls  
# ---  
#   
result1$cluster

## [1] 1 1 1 1 1 1 1 1 3 1 1 1 1 1 3 1 1 3 1 3 3 1 3 3 3 3 3 3 2 2 2 1 1 1 1 1 1  
## [38] 1 1 2 1 2 2 1 1 1 1 1 3 3 3 3 3 3 2 3 3 3 3 3 3 2 1 1 1 2 1 2 2 2 2 2 1 1  
## [75] 2 3 2 2 2 2 2 3 2 2 2 2 3 3 2 2 2 2 1 1 1 2 1 2 2 2 2 2 2 1 1 2 3 3 3 3 1  
## [112] 2 1 3 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 3 3 2 1 3 1 2 1 3 3 3 3  
## [149] 3 3 3 3 3

# Visualizing the clustering results  
# ---  
#   
par(mfrow = c(1,2), mar = c(5,4,2,2))  
  
# Plotting to see how Ozone and Solar.R data points have been distributed in clusters  
# ---  
#  
plot(airquality[,1:2], col = result1$cluster)



# Verifying the results of clustering  
# ---  
#   
par(mfrow = c(2,2), mar = c(5,4,2,2))  
  
# Plotting to see how Sepal.Length and Sepal.Width data points have been distributed in clusters  
plot(airquality.new[c(1,2)], col = result1$cluster)  
  
# Plotting to see how Sepal.Length and Sepal.Width data points have been distributed   
# originally as per "class" attribute in dataset  
# ---  
#  
plot(airquality.new[c(1,2)], col = airquality.class)  
  
# Plotting to see how Petal.Length and Petal.Width data points have been distributed in clusters  
# ---  
#   
plot(airquality.new[c(3,4)], col = result1$cluster)  
plot(airquality.new[c(3,4)], col = airquality.class)



#   
table(result1$cluster, airquality.class)

## airquality.class  
## 5 6 7 8 9  
## 1 17 14 6 8 12  
## 2 3 4 21 18 6  
## 3 11 12 4 5 12

## Challenge 2  
# ---  
# Question: Create a model that clusters the following dataset.  
# ---  
# Dataset = http://bit.ly/SalaryDatasetClustering  
# ---  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.1.0 v dplyr 1.0.5  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(data.table)

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

## The following object is masked from 'package:purrr':  
##   
## transpose

salary <- fread("http://bit.ly/SalaryDatasetClustering")  
head(salary)

## Id EmployeeName JobTitle  
## 1: 1 NATHANIEL FORD GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY  
## 2: 2 GARY JIMENEZ CAPTAIN III (POLICE DEPARTMENT)  
## 3: 3 ALBERT PARDINI CAPTAIN III (POLICE DEPARTMENT)  
## 4: 4 CHRISTOPHER CHONG WIRE ROPE CABLE MAINTENANCE MECHANIC  
## 5: 5 PATRICK GARDNER DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)  
## 6: 6 DAVID SULLIVAN ASSISTANT DEPUTY CHIEF II  
## BasePay OvertimePay OtherPay Benefits TotalPay TotalPayBenefits Year  
## 1: 167411.18 0.0 400184.25 567595.4 567595.4 2011  
## 2: 155966.02 245131.88 137811.38 538909.3 538909.3 2011  
## 3: 212739.13 106088.18 16452.6 335279.9 335279.9 2011  
## 4: 77916.0 56120.71 198306.9 332343.6 332343.6 2011  
## 5: 134401.6 9737.0 182234.59 326373.2 326373.2 2011  
## 6: 118602.0 8601.0 189082.74 316285.7 316285.7 2011  
## Notes Agency Status  
## 1: NA San Francisco   
## 2: NA San Francisco   
## 3: NA San Francisco   
## 4: NA San Francisco   
## 5: NA San Francisco   
## 6: NA San Francisco

colSums(is.na(salary))

## Id EmployeeName JobTitle BasePay   
## 0 0 0 0   
## OvertimePay OtherPay Benefits TotalPay   
## 0 0 0 0   
## TotalPayBenefits Year Notes Agency   
## 0 0 148654 0   
## Status   
## 0

salary.class<- salary[, "Status"]  
  
head(salary.class)

## Status  
## 1:   
## 2:   
## 3:   
## 4:   
## 5:   
## 6:

salary.new<- salary[, c(1, 2, 3, 4,6)]  
airquality.class<- airquality[, "Month"]  
head(airquality.new)

## Ozone Solar.R Wind Temp Day  
## 1 0.23952096 0.5596330 0.3000000 0.2682927 0.00000000  
## 2 0.20958084 0.3394495 0.3315789 0.3902439 0.03333333  
## 3 0.06586826 0.4342508 0.5736842 0.4390244 0.06666667  
## 4 0.10179641 0.9357798 0.5157895 0.1463415 0.10000000  
## 5 0.13542146 0.5330162 0.6631579 0.0000000 0.13333333  
## 6 0.16167665 0.5330162 0.6947368 0.2439024 0.16666667

## Challenge 3  
# ---  
# Question: Cluster customers from the given wholesale customer database.  
# ---  
# Dataset source = https://archive.ics.uci.edu/ml/datasets/Wholesale+customers  
# ---  
  
customer <- fread("https://archive.ics.uci.edu/ml/datasets/Wholesale+customers")

## Warning in fread("https://archive.ics.uci.edu/ml/datasets/Wholesale+customers"):  
## Detected 1 column names but the data has 2 columns (i.e. invalid file). Added 1  
## extra default column name for the first column which is guessed to be row names  
## or an index. Use setnames() afterwards if this guess is not correct, or fix the  
## file write command that created the file to create a valid file.

## Warning in fread("https://archive.ics.uci.edu/ml/datasets/Wholesale+customers"):  
## Stopped early on line 27. Expected 2 fields but found 1. Consider fill=TRUE and  
## comment.char=. First discarded non-empty line: <<}>>

head(customer)

## V1 {  
## 1: alert( "Please enter search terms." ) NA  
## 2: form.q.focus() NA  
## 3: return false NA

customer

customer

## V1 {  
## 1: alert( "Please enter search terms." ) NA  
## 2: form.q.focus() NA  
## 3: return false NA