Capstone project

StepUp Analytics

01/05/2021

What you will learn and Implement

# > Understand kmeans clustering.

# > Understand retail Data for applying kmeans clustering.

# > Understand the objective.

# > Maths of kmeans clustering.

# > understand Optimum number of cluster.

# > Using elbow method to determine the optimal number of cluster.

# > model fitting

# > Evaluation

# > Objective

# The client is a leading Fashion retailer in Australia. Company runs a display advertising campaign for this brand, where it shows ads to users leading them to make a purchase on the brands website.The given dataset is the Sales data for all users who made a purchase online in the first half of October ’17.

# Based on your analysis, arrive at a statistical segmentation of the brands audience based

# on Revenue. The number of segments is up to you. Please provide definitions of each groups

# > desired Library installation

# install.packages("factoextra")  
# install.packages("rpivotTable")  
# install.packages("cluster")  
# install.packages("fpc")  
# install.packages("factoextra")  
# install.packages("rpivotTable")  
# install.packages("ggplot2")  
# install.packages("dplyr")  
# install.packages("magrittr")

library(ggplot2)  
library(dplyr)

##   
## 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

library(magrittr)  
library(cluster)  
library(fpc)  
library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(rpivotTable)

# > Setting working directory

path<-"D:/data science/Industry Application - Kmeans"  
setwd(path)

# > reading data

Revenue<-read.csv("RevenueCluster.csv", header=T, na.strings=c("","NA"))  
head(Revenue, 10)

## timestamp City Country\_Province Country\_Code Revenue.AUD.  
## 1 21-10-2017 20:52 Kaitaia <NA> NZ 212.50  
## 2 21-10-2017 20:55 Shepparton VIC AU 526.36  
## 3 21-10-2017 20:56 Essendon VIC AU 77.23  
## 4 21-10-2017 20:59 Karabar NSW AU 190.91  
## 5 21-10-2017 21:00 Mosman NSW AU 204.09  
## 6 21-10-2017 21:06 Caulfield East VIC AU 122.71  
## 7 21-10-2017 21:08 Lyneham ACT AU 104.46  
## 8 21-10-2017 21:08 Frankston South VIC AU 54.50  
## 9 21-10-2017 21:10 Moonah TAS AU 81.77  
## 10 21-10-2017 21:12 Christchurch <NA> AU 72.68

str(Revenue)

## 'data.frame': 49984 obs. of 5 variables:  
## $ timestamp : chr "21-10-2017 20:52" "21-10-2017 20:55" "21-10-2017 20:56" "21-10-2017 20:59" ...  
## $ City : chr "Kaitaia" "Shepparton" "Essendon" "Karabar" ...  
## $ Country\_Province: chr NA "VIC" "VIC" "NSW" ...  
## $ Country\_Code : chr "NZ" "AU" "AU" "AU" ...  
## $ Revenue.AUD. : num 212.5 526.4 77.2 190.9 204.1 ...

Revenue$timestamp<-as.POSIXct(Revenue$timestamp, format = "%d-%m-%Y %H:%M")

# > subsetting/manipulating data to fit kmeans clustering

# Calculating Missing value

sum(is.na(Revenue))

## [1] 4517

sapply(Revenue, function(x) {  
 sum(is.na(x))  
})

## timestamp City Country\_Province Country\_Code   
## 0 337 4180 0   
## Revenue.AUD.   
## 0

#Treating missing categorical values with their mode.

Mode<-function(v){  
 uniqv <- unique(Revenue$City)  
 uniqv[which.max(tabulate(match(Revenue$City, uniqv)))]  
}  
result<-Mode(Revenue$City)  
print(result)

## [1] "Sydney"

# mode value for city is Sydney.  
# Replacing NA's with Sydney  
Revenue$City[is.na(Revenue$City)]<-"Sydney"

Mode2<-function(v){  
 uniqv2 <- unique(Revenue$Country\_Province)  
 uniqv2[which.max(tabulate(match(Revenue$Country\_Province, uniqv2)))]  
}  
result2<-Mode2(Revenue$Country\_Province)  
print(result2)

## [1] "NSW"

#mode value for Country Province is NSW  
# Replacing NA's with NSW  
Revenue$Country\_Province[is.na(Revenue$Country\_Province)]<-"NSW"

#Checking the data and data types

str(Revenue)

## 'data.frame': 49984 obs. of 5 variables:  
## $ timestamp : POSIXct, format: "2017-10-21 20:52:00" "2017-10-21 20:55:00" ...  
## $ City : chr "Kaitaia" "Shepparton" "Essendon" "Karabar" ...  
## $ Country\_Province: chr "NSW" "VIC" "VIC" "NSW" ...  
## $ Country\_Code : chr "NZ" "AU" "AU" "AU" ...  
## $ Revenue.AUD. : num 212.5 526.4 77.2 190.9 204.1 ...

summary(Revenue)

## timestamp City Country\_Province   
## Min. :2017-10-01 00:00:00 Length:49984 Length:49984   
## 1st Qu.:2017-10-05 10:09:00 Class :character Class :character   
## Median :2017-10-10 20:12:00 Mode :character Mode :character   
## Mean :2017-10-10 23:29:31   
## 3rd Qu.:2017-10-16 05:01:15   
## Max. :2017-10-21 23:59:00   
## Country\_Code Revenue.AUD.   
## Length:49984 Min. : 0.00   
## Class :character 1st Qu.: 63.59   
## Mode :character Median : 98.18   
## Mean : 134.04   
## 3rd Qu.: 163.54   
## Max. :5387.71

# > fitting basic model taking a random number of cluster say “3”

set.seed(12345)  
model\_km<-kmeans(Revenue$Revenue.AUD., centers = 3) # Elbow methods  
aggregate(Revenue$Revenue.AUD.,by=list(model\_km$cluster),FUN=mean)

## Group.1 x  
## 1 1 85.84684  
## 2 2 782.07275  
## 3 3 268.01854

*Cluster1- Customers with Low Value* *Cluster2- Customers with High Value* *Cluster3- Customers with Moderate Value*

# Customers with High Value are regular in nature and generate high revenue

# Customers with Moderate Value are potential customers

# Customers with low Value are irregular or gone away in nature

# > calculating optimum number cluster

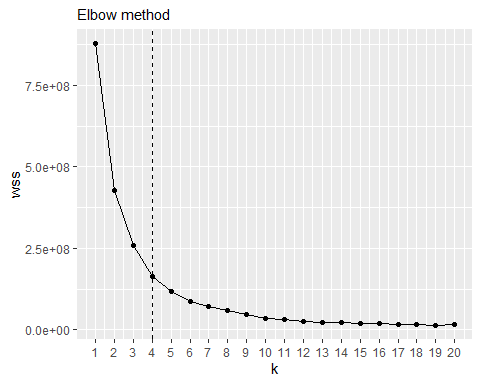
optimcluster <- function(k){  
 cluster <- kmeans(Revenue$Revenue.AUD., k)  
 return(cluster$tot.withinss)  
}  
  
wss <- sapply(1:20, optimcluster)  
  
optimK <- data.frame(k = 1:20, wss)  
head(optimK, 10)

## k wss  
## 1 1 877533743  
## 2 2 426452796  
## 3 3 257740380  
## 4 4 163515162  
## 5 5 117159892  
## 6 6 87911691  
## 7 7 71667954  
## 8 8 60285551  
## 9 9 47341809  
## 10 10 34968891

Interpretation: The method consists of plotting the explained variation as a function of the number of clusters, and picking the elbow of the curve as the number of clusters to use.

### ploting the k value from 1 to 20 against within sum of square (wss)

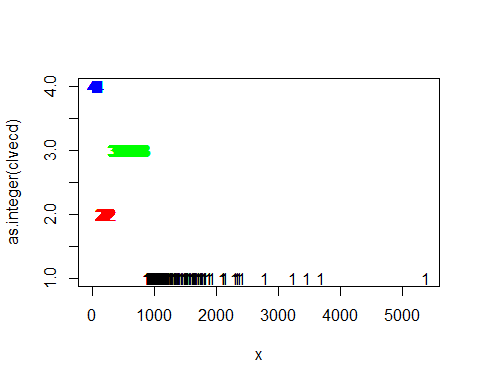
ggplot(data = optimK) + aes(x=k, y= wss) + geom\_point() + geom\_line()+  
 scale\_x\_continuous(breaks = seq(1,20,by = 1)) +geom\_vline(xintercept = 4, linetype = 2)+  
 labs(subtitle = "Elbow method")



OptimumClusters<- "Optimum number of cluster is 4"

# > Model fitting using optimum k

set.seed(500000)  
model\_km\_optm<-kmeans(Revenue$Revenue.AUD., centers = 4) # Elbow methods  
plotcluster(Revenue$Revenue.AUD., model\_km\_optm$cluster)



model\_km\_optm$centers #Display cluster centers

## [,1]  
## 1 1328.24702  
## 2 196.90208  
## 3 455.10944  
## 4 73.24984

Cluster\_size<-table(model\_km\_optm$cluster) #Give a count of data points in each cluster  
print(Cluster\_size)

##   
## 1 2 3 4   
## 171 14061 2842 32910

names(Cluster\_size)<-c("High Value","Deal seekers","Loyal","Infrequent")  
Cluster\_size

## High Value Deal seekers Loyal Infrequent   
## 171 14061 2842 32910

Aggregate<-aggregate(Revenue$Revenue.AUD.,by=list(model\_km\_optm$cluster),FUN=mean)  
Aggregate$Group.1<-plyr::mapvalues( Aggregate$Group.1,  
 from = c(1,2,3,4),  
 to = c("High Value", "Deal seekers","Loyal","Infrequent"))  
  
Revenue$Clusters <- model\_km\_optm$cluster  
Revenue$Groups <- plyr::mapvalues( Revenue$Clusters,  
 from = c(1,2,3,4),  
 to = c("High Value", "Deal seekers","Loyal","Infrequent"))  
head(Revenue, 20)

## timestamp City Country\_Province Country\_Code  
## 1 2017-10-21 20:52:00 Kaitaia NSW NZ  
## 2 2017-10-21 20:55:00 Shepparton VIC AU  
## 3 2017-10-21 20:56:00 Essendon VIC AU  
## 4 2017-10-21 20:59:00 Karabar NSW AU  
## 5 2017-10-21 21:00:00 Mosman NSW AU  
## 6 2017-10-21 21:06:00 Caulfield East VIC AU  
## 7 2017-10-21 21:08:00 Lyneham ACT AU  
## 8 2017-10-21 21:08:00 Frankston South VIC AU  
## 9 2017-10-21 21:10:00 Moonah TAS AU  
## 10 2017-10-21 21:12:00 Christchurch NSW AU  
## 11 2017-10-21 21:13:00 Christchurch NSW NZ  
## 12 2017-10-21 21:16:00 Mooroopna VIC AU  
## 13 2017-10-21 21:24:00 Warana QLD AU  
## 14 2017-10-21 21:24:00 Campbellfield VIC AU  
## 15 2017-10-21 21:26:00 Kelso NSW AU  
## 16 2017-10-21 21:26:00 Hawthorn VIC AU  
## 17 2017-10-21 21:27:00 Aldinga Beach SA AU  
## 18 2017-10-21 21:30:00 Melbourne VIC AU  
## 19 2017-10-21 21:31:00 GOODWOOD SA AU  
## 20 2017-10-21 21:34:00 Palmerston North NSW AU  
## Revenue.AUD. Clusters Groups  
## 1 212.50 2 Deal seekers  
## 2 526.36 3 Loyal  
## 3 77.23 4 Infrequent  
## 4 190.91 2 Deal seekers  
## 5 204.09 2 Deal seekers  
## 6 122.71 4 Infrequent  
## 7 104.46 4 Infrequent  
## 8 54.50 4 Infrequent  
## 9 81.77 4 Infrequent  
## 10 72.68 4 Infrequent  
## 11 127.23 4 Infrequent  
## 12 63.59 4 Infrequent  
## 13 154.50 2 Deal seekers  
## 14 100.88 4 Infrequent  
## 15 35.60 4 Infrequent  
## 16 79.45 4 Infrequent  
## 17 67.19 4 Infrequent  
## 18 172.68 2 Deal seekers  
## 19 49.36 4 Infrequent  
## 20 345.45 3 Loyal

**Analysis**

# High value customers are those customers who generates high values of revenue for the retailer in range of 892 to 5387 AUD

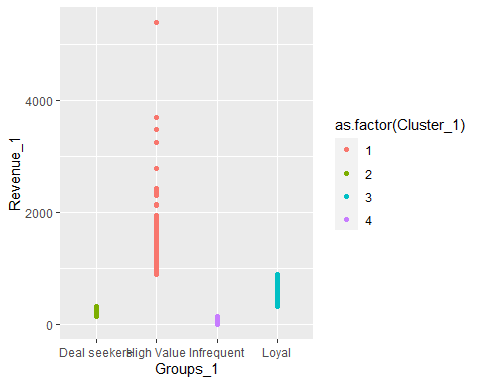
# Deal Seekers are customers who needs a little push and incentives in terms of sales and deals. They generate revenue in range of 135 to 325 AUD

# Loyal customers are the potential customers who are regular in purchasing and generate revenue in range of 326 to 890 AUD

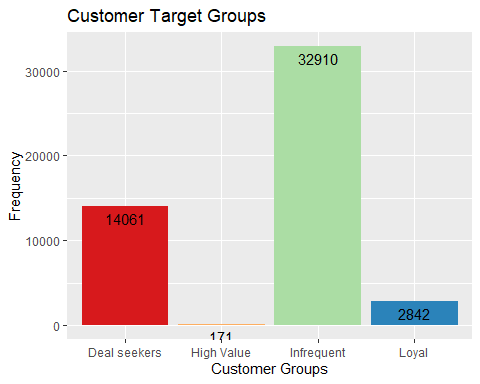
# Infrequent customers are those customers who have either not generated any income or have brought revenue to the retailer less than 138 AUD

o=order(model\_km\_optm$cluster)  
Revenue\_1<-Revenue$Revenue.AUD.[o]  
Cluster\_1<-model\_km\_optm$cluster[o]  
Groups\_1<- Revenue$Groups[o]  
Customer\_segmentation<-data.frame(Revenue\_1, Cluster\_1, Groups\_1)

ggplot(Customer\_segmentation, aes(x=Groups\_1, y= Revenue\_1, col = as.factor(Cluster\_1))) +  
 geom\_point()



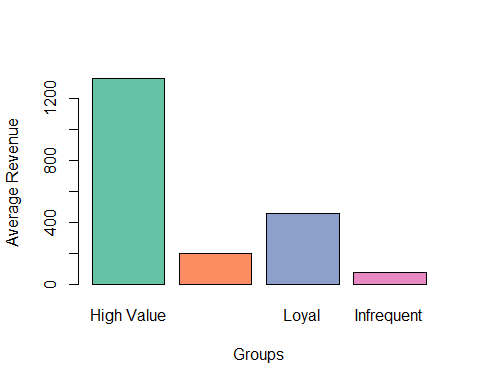
library(RColorBrewer)  
coul2 <- brewer.pal(4, "Spectral")  
ggplot(Revenue, aes(x=Groups))+geom\_bar(fill= coul2 )+xlab("Customer Groups")+ylab("Frequency")+  
ggtitle("Customer Target Groups")+ geom\_text(aes(label = ..count..), stat = "count", vjust = 1.5, colour = "black")



Aggregate$x<-as.numeric(Aggregate$x)  
str(Aggregate$x)

## num [1:4] 1328.2 196.9 455.1 73.2

coul1 <- brewer.pal(4, "Set2")  
barp<-barplot(Aggregate$x, names.arg =Aggregate$Group.1, col=coul1, xlab = "Groups", ylab= "Average Revenue")



write.csv(Customer\_segmentation, 'Customer\_segmentation.csv')