# Unsupervised Learning With R

Gathoni Njoroge

2022-06-02

## Prediction of with Unsupervised Learning.

### 1) Defining the question.

**a) Specifying the question.**

**What are the characteristics of the customer groups in the given data?**

**b) The metric of success.**

**Identifing and differntiating specific trends/qualities of the customer groups using clusters.**

**c) The context.**

**Determining how customers are alike or how they differ.**

**d) Experimental design.**

**i) Problem Definition**

**ii) Data Sourcing**

**iii) Check the Data**

**iv) Perform Data Cleaning**

**v) Perform Exploratory Data Analysis (Univariate, Bivariate & Multivariate)**

**vi) Implement the Solution**

**vii) Challenge the Solution**

**viii) Follow up Questions**

**e) Appropriateness of the Data Available.**

**The data was collected by Kira Plastinina and is therefore appropriate for this study.**

## 2) Sourcing the data.

```
# loading appropriate libraries
#
library(data.table)
library(tibble)
library(tidyverse)
library(corrplot)
library(ggplot2)
library(GGally)
library(caret)
library(moments)
```

```
# loading the data
#
custom <- fread("http://bit.ly/EcommerceCustomersDataset")
custom
```

## 3) Checking the data.

```
# Checking the top of our data set
#
head(custom)
```

```
# Checking the bottom of our data set
#
tail(custom)
```

```
# Checking the data set
#
glimpse(custom)
```

```
# Convert data into tibble
#
customer <- tibble(custom)
customer
```

```
# checking the elements in the target variable
#
unique(customer$VisitorType)
```

## 4) Data Cleaning.

```
# Check for missing values
#
colSums(is.na(customer))
```

```
# Dropping null values
#
customer1 <- na.omit(customer)
customer1
```

```
# checking for duplicates
#
duplicated <- customer1[duplicated(customer1),]
duplicated
```

```
# removing outliers
#
outliers <- function(x) {

  Q1 <- quantile(x, probs=.25)
  Q3 <- quantile(x, probs=.75)
  iqr = Q3-Q1

 upper_limit = Q3 + (iqr*1.5)
 lower_limit = Q1 - (iqr*1.5)

 x > upper_limit | x < lower_limit
}

remove_outliers <- function(customer1, cols = names(customer1)) {
  for (col in cols) {
    customer1 <- customer1[!outliers(customer1[[col]]),]
  }
  customer1
}
customer2 <- remove_outliers(customer1, c('Administrative', 'Administrative_Duration',
'Informational', 'Informational_Duration', 'ProductRelated', 'ProductRelated_Duration',
'BounceRates', 'ExitRates', 'PageValues', 'OperatingSystems', 'Browser', 'Region'))
customer2
```

```
# Changing the categorical columns into factors
#
customer2$Month <- as.numeric(factor(customer2$Month))
customer2$Weekend <- as.numeric(factor(customer2$Weekend))
customer2$Revenue <- as.numeric(factor(customer2$Revenue))
customer2
```

```
# Dropping the target variable
#
drop <- c("VisitorType")
customers = customer2[,!(names(customer2) %in% drop)]
customers
```

**The repeated values in some variables seem as though they are duplicates but the difference in values of corresponding columns contrasts that. Therefore, the 'duplicates' will not be dropped.**

## 5) Exploratory Data Analysis.

**Univariate Analysis.**

```
# Measures of central Tendency
# mean
#
print('mean')
mean(customers$Administrative)
mean(customers$Administrative_Duration)
mean(customers$Informational)
mean(customers$Informational_Duration)
mean(customers$ProductRelated)
mean(customers$ProductRelated_Duration)
mean(customers$BounceRates)
mean(customers$ExitRates)
mean(customers$PageValues)
mean(customers$SpecialDay)
mean(customers$Month)
mean(customers$OperatingSystems)
mean(customers$Browser)
mean(customers$Region)
mean(customers$TrafficType)

# median
#
print('median')
median(customers$Administrative)
median(customers$Administrative_Duration)
median(customers$Informational)
median(customers$Informational_Duration)
median(customers$ProductRelated)
median(customers$ProductRelated_Duration)
median(customers$BounceRates)
median(customers$ExitRates)
median(customers$PageValues)
median(customers$SpecialDay)
median(customers$Month)
median(customers$OperatingSystems)
median(customers$Browser)
```

```
median(customers$Region)
median(customers$TrafficType)
median(customers$Weekend)
median(customers$Revenue)

#mode
#
print('mode')
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

getmode(customers$Administrative)
getmode(customers$Administrative_Duration)
getmode(customers$Informational)
getmode(customers$Informational_Duration)
getmode(customers$ProductRelated)
getmode(customers$ProductRelated_Duration)
getmode(customers$BounceRates)
getmode(customers$ExitRates)
getmode(customers$PageValues)
getmode(customers$SpecialDay)
getmode(customers$Month)
getmode(customers$OperatingSystems)
getmode(customers$Browser)
getmode(customers$Region)
getmode(customers$TrafficType)
getmode(customers$Weekend)
getmode(customers$Revenue)
```

```
# measures of dispersion
# stardard deviation
#
print('Standard Deviation')

sd(customers$Administrative)
sd(customers$Administrative_Duration)
sd(customers$Informational)
sd(customers$Informational_Duration)
sd(customers$ProductRelated)
sd(customers$ProductRelated_Duration)
sd(customers$BounceRates)
sd(customers$ExitRates)
sd(customers$PageValues)
sd(customers$SpecialDay)
sd(customers$Month)
sd(customers$OperatingSystems)
sd(customers$Browser)
sd(customers$Region)
sd(customers$TrafficType)
sd(customers$Weekend)
sd(customers$Revenue)
```

```
# variance
#
print('Variance')

var(customers$Administrative)
var(customers$Administrative_Duration)
var(customers$Informational)
var(customers$Informational_Duration)
var(customers$ProductRelated)
var(customers$ProductRelated_Duration)
var(customers$BounceRates)
var(customers$ExitRates)
var(customers$PageValues)
var(customers$SpecialDay)
var(customers$Month)
var(customers$OperatingSystems)
var(customers$Browser)
var(customers$Region)
var(customers$TrafficType)
var(customers$Weekend)
var(customers$Revenue)

# skewness
#
print('skewness')

skewness(customers$Administrative)
skewness(customers$Administrative_Duration)
skewness(customers$Informational)
skewness(customers$Informational_Duration)
skewness(customers$ProductRelated)
skewness(customers$ProductRelated_Duration)
skewness(customers$BounceRates)
skewness(customers$ExitRates)
skewness(customers$PageValues)
skewness(customers$SpecialDay)
skewness(customers$Month)
skewness(customers$OperatingSystems)
skewness(customers$Browser)
skewness(customers$Region)
skewness(customers$TrafficType)
skewness(customers$Weekend)
skewness(customers$Revenue)

# Kurtosis
#
print('kurtosis')

kurtosis(customers$Administrative)
kurtosis(customers$Administrative_Duration)
kurtosis(customers$Informational)
kurtosis(customers$Informational_Duration)
kurtosis(customers$ProductRelated)
```

```
kurtosis(customers$ProductRelated_Duration)
kurtosis(customers$BounceRates)
kurtosis(customers$ExitRates)
kurtosis(customers$PageValues)
kurtosis(customers$SpecialDay)
kurtosis(customers$Month)
kurtosis(customers$OperatingSystems)
kurtosis(customers$Browser)
kurtosis(customers$Region)
kurtosis(customers$TrafficType)
kurtosis(customers$Weekend)
kurtosis(customers$Revenue)

# quantiles
#
print('quantile')

quantile(customers$Administrative)
quantile(customers$Administrative_Duration)
quantile(customers$Informational)
quantile(customers$Informational_Duration)
quantile(customers$ProductRelated)
quantile(customers$ProductRelated_Duration)
quantile(customers$BounceRates)
quantile(customers$ExitRates)
quantile(customers$PageValues)
quantile(customers$SpecialDay)
quantile(customers$Month)
quantile(customers$OperatingSystems)
quantile(customers$Browser)
quantile(customers$Region)
quantile(customers$TrafficType)
quantile(customers$Weekend)
quantile(customers$Revenue)
```

**Visualization.**

```
# Histograms to visualize frequenct, skewness and kurtosis
#
hist(customers$Administrative)
hist(customers$Administrative_Duration)
hist(customers$Informational)
hist(customers$Informational_Duration)
hist(customers$ProductRelated)
hist(customers$ProductRelated_Duration)
hist(customers$BounceRates)
hist(customers$ExitRates)
hist(customers$PageValues)
hist(customers$OperatingSystems)
hist(customers$Browser)
hist(customers$Region)
```

```
hist(customers$TrafficType)
```

```
# box plots
#
boxplot(customers$Administrative)
boxplot(customers$Administrative_Duration)
boxplot(customers$Informational)
boxplot(customers$Informational_Duration)
boxplot(customers$ProductRelated)
boxplot(customers$ProductRelated_Duration)
boxplot(customers$BounceRates)
boxplot(customers$ExitRates)
boxplot(customers$PageValues)
boxplot(customers$SpecialDay)
boxplot(customers$Month)
boxplot(customers$OperatingSystems)
boxplot(customers$Browser)
boxplot(customers$Region)
boxplot(customers$TrafficType)
boxplot(customers$Weekend)
boxplot(customers$Revenue)
```

**Bivariate Analysis.**

```
# Pearson's Corelation matrix
#
#cor(customers)
# drop columns with sd=0
#
drop <- c("Informational", "Informational_Duration", "PageValues", "Browser")
cust = customers[,!(names(customers) %in% drop)]
cust
cor(cust)
```

```
# covariance matrix
#
cov(cust)
```

**Visualization.**

```
# correlation matrix
#
library(corrplot)
library(RColorBrewer)
x <- cor(cust)
corrplot(x, method = 'color', addCoef.col = 'brown', col = COL2('BrBG'),
         number.cex = 0.8, tl.cex = 0.8, tl.col = 'black')
#png(file = "corrplot.png", width = 40, height = 40)
```

```
#dev.off()
```

```
# scatter plot matrix
#
plot(cust)
```

**Multivariate Analysis.**

## 6) Implementing the Solution.

**With K-means Clustering.**

```
#drop columns with high correlation
#
drop <- c("Informational", "Informational_Duration", "PageValues", "Browser")
customer3 = customer2[,!(names(customer2) %in% drop)]
customer3
customer4 <- customer3 %>% relocate(Weekend, Revenue)
customer4
```

```
customer5<- customer4[, c(1, 2, 3, 4,5,6,7,8,9,10,11,12,13)]
class<- customer4$VisitorType
head(customer5)
(class)
```

```
# Normalizing the data
#
normalize <- function(x){
  return ((x-min(x)) / (max(x)-min(x)))
}
customer_ <- normalize(customer5)
customer_
```

```
# Setting cluster size
#
result <- kmeans(customer_,3)
result$size
```

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

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

```
# Visualizing the  clustering results
# ---
#
par(mfrow = c(1,2), mar = c(6,5,3,3))

plot(customer_[,5:6], col = kclusters)
```

```
# Table to show how customers are clustered
#
table(class, kclusters)
```

**Hierarchial Clustering.**

```
# Scaling the data
#
customer_ <- scale(customer_)
head(customer_)
```

```
# Compute the Euclidean distance between observations.
#
distance <- dist(customer4[1:13], method = "euclidean")

customer.hc <- hclust(d, method = "ward.D2" )
```

{r, fig.height=15,fig.width=15} # Lastly, we plot the obtained dendrogram # --- #  plot(customer.hc, cex = 0.6, hang = -1)

```
# increase the number of clusters
#
clusters = cutree(customer.hc, k = 3)
table(clusters, class)
```

```
# computing accuracy
#
cm = as.matrix(table(Actual = class, Predicted = clusters))
accuracy = sum(diag(cm)) / sum(cm)
print(accuracy)
```

## Comparison of models:

**Kmeans Clustering; It is simple to implement, scales well to large data sets and is flexible even when new data is added.However, it is sensitive outliers and this may lead to valuable information being lost when outliers are dropped. Scaling data with alot of dimentions could also prove difficult.**

**Heirarchial Clustering; It is also simple and easy to implement which saves on time. However, it does not do well with bulky multi-dimensional data like the one we have. This has brought poor visualization and a low accuracy in clustering.**

**For this analysis Kmeans clustering should be chosen over heirarchial clustering.**

## 7) Challenging the Solution.

**Supervised Learning techniques should be implemented for comparison.**

## 8) Conclusion.

**With the above analysis and implementation of both models, a common trend among returning visitors is that they are more active and have higher results. New visitors are fairly active and the others may not qualify to by target customers.**

## 9) Follow up Questions.

**a) Did we have the right data?**

**Yes.**

**b) Do we need other data to answer our question?**

**Yes this data is very vague about the type of customers and specific characteristics**

**c) Did we have the right question?**

**Yes.**