Kira Plastinina

John Maronga

26-03-2-22

1. Problem Definition

1.1 Defining the Question

Kira Plastinina is a Russian brand that is sold through a defunct chain of retail stores in Russia, Ukraine, Kazakhstan, Belarus, China, Philippines, and Armenia. The brand's Sales and Marketing team would like to understand their customer's behavior from data that they have collected over the past year.

1.2 Specifying the Question

Create a model that helps KIra Plastinina learn the characteristics of customer groups

1.3 Defining the Metric of Success

- model that predicts accurately with an accuracy score of 80% whether a user will click on an ad or not
- Give insights to the company concerning their brand from the EDA done
- Challenge and improve the created unsupervied model

1.4 Understanding the Context

Kira Plastininais a Russian fashion designer and entrepreneur. Her brand was sold through a now defunct chain of eponymous retail stores in Russia, Ukraine, Kazakhstan, Belarus, China, Philippinesand Armenia Plastinina was born in Moscow. Her father, Sergei Plastinin, founded Wimm-Bill-Dann Foods OJSC in 1992 and served as its Chief Executive Officer until April 3, 2006. He was impressed with his daughter's passion for design and vision of teenage fashion, and suggested that they launch a fashion brand together.In 2007, the first Kira Plastinina store opened in Moscow, Plastinina introduced her first collection and became one of the youngest fashion designers in the world. Since then, the company has opened over 300 stores in Russia and CIS. In 2008, the Company made an unsuccessful attempt to enter the U.S. market. The U.S. entity eventually filed for bankruptcy. With out analysis segmentation, we can identify why her brand failed in the US maybe help her in her customer segmentation

1.5 Experimental Design taken

1. Data Exploration
2. Data Cleaning and Formatting

3. Univariate Analysis
4. Bivariate Analysis
5. Multivariate Analysis
6. Implementing the solution through unsupervised machine learning,i.e. k-means, hierachical and DBSCAN
7. Conclusion and Next steps

2. Data Sourcing

The data was availed to our data science team by the brand's Sales and Marketing team therefore no data collection and scrapping was needed...We will just load our dataset in RStudio and begin the analysis process

3. Check the Data

```
## Loading packages that we will use during our analysis
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("purrr")
library('tidyverse')

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v ggplot2 3.3.5     v readr   2.1.2
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library('magrittr')

##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:tidyr':
##
##     extract

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

library('corrplot')

## corrplot 0.92 loaded

library('caret')

## Loading required package: lattice

##
## Attaching package: 'caret'

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

library('skimr')
library(readr)

customer_behavior<- read.csv("http://bit.ly/EcommerceCustomersDataset")
## previewing first 6 rows
head(customer_behavior)

##   Administrative Administrative_Duration Informational
Informational_Duration
## 1              0                       0             0
0
## 2              0                       0             0
0
## 3              0                      -1             0
-1
## 4              0                       0             0
0
## 5              0                       0             0
0
## 6              0                       0             0
0
##   ProductRelated ProductRelated_Duration BounceRates ExitRates PageValues
## 1              1                0.000000  0.20000000 0.2000000          0
## 2              2               64.000000  0.00000000 0.1000000          0
## 3              1               -1.000000  0.20000000 0.2000000          0
## 4              2                2.666667  0.05000000 0.1400000          0
## 5             10              627.500000  0.02000000 0.0500000          0
## 6             19              154.216667  0.01578947 0.0245614          0
##   SpecialDay Month OperatingSystems Browser Region TrafficType
```

```
## 1          0   Feb                1       1      1           1
## 2          0   Feb                2       2      1           2
## 3          0   Feb                4       1      9           3
## 4          0   Feb                3       2      2           4
## 5          0   Feb                3       3      1           4
## 6          0   Feb                2       2      1           3
##         VisitorType Weekend Revenue
## 1 Returning_Visitor   FALSE   FALSE
## 2 Returning_Visitor   FALSE   FALSE
## 3 Returning_Visitor   FALSE   FALSE
## 4 Returning_Visitor   FALSE   FALSE
## 5 Returning_Visitor    TRUE   FALSE
## 6 Returning_Visitor   FALSE   FALSE
```

```
##previewing the last 6 rows of the dataset
tail(customer_behavior)
```

```
##       Administrative Administrative_Duration Informational
## 12325              0                      0             1
## 12326              3                    145             0
## 12327              0                      0             0
## 12328              0                      0             0
## 12329              4                     75             0
## 12330              0                      0             0
##       Informational_Duration ProductRelated ProductRelated_Duration
BounceRates
## 12325                      0             16                 503.000
0.000000000
## 12326                      0             53                1783.792
0.007142857
## 12327                      0              5                 465.750
0.000000000
## 12328                      0              6                 184.250
0.083333333
## 12329                      0             15                 346.000
0.000000000
## 12330                      0              3                  21.250
0.000000000
##        ExitRates PageValues SpecialDay Month OperatingSystems Browser
Region
## 12325 0.03764706    0.00000          0   Nov                2       2
1
## 12326 0.02903061   12.24172          0   Dec                4       6
1
## 12327 0.02133333    0.00000          0   Nov                3       2
1
## 12328 0.08666667    0.00000          0   Nov                3       2
1
## 12329 0.02105263    0.00000          0   Nov                2       2
3
```

```
## 12330 0.06666667    0.00000          0   Nov                3       2
1
##       TrafficType       VisitorType Weekend Revenue
## 12325           1 Returning_Visitor   FALSE   FALSE
## 12326           1 Returning_Visitor    TRUE   FALSE
## 12327           8 Returning_Visitor    TRUE   FALSE
## 12328          13 Returning_Visitor    TRUE   FALSE
## 12329          11 Returning_Visitor   FALSE   FALSE
## 12330           2       New_Visitor    TRUE   FALSE
```

```
## Previewing the shape of our dataset
dim(customer_behavior)
```

```
## [1] 12330    18
```

```
### we have 12330     rows and 18 columns!!!BAM!!
#checking the datatypes on the columns
sapply(customer_behavior, class)
```

```
##          Administrative Administrative_Duration           Informational
##               "integer"               "numeric"               "integer"
##  Informational_Duration          ProductRelated ProductRelated_Duration
##               "numeric"               "integer"               "numeric"
##             BounceRates               ExitRates              PageValues
##               "numeric"               "numeric"               "numeric"
##              SpecialDay                   Month        OperatingSystems
##               "numeric"             "character"               "integer"
##                 Browser                  Region             TrafficType
##               "integer"               "integer"               "integer"
##             VisitorType                 Weekend                 Revenue
##             "character"               "logical"               "logical"
```

```
 ## The dataset consists of 10 numerical and 8 categorical attributes.
##checking for structure is using the str()
str(customer_behavior)
```

```
## 'data.frame':    12330 obs. of  18 variables:
##  $ Administrative         : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration: num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ Informational          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ ProductRelated         : int  1 2 1 2 10 19 1 1 2 3 ...
##  $ ProductRelated_Duration: num  0 64 -1 2.67 627.5 ...
##  $ BounceRates            : num  0.2 0 0.2 0.05 0.02 ...
##  $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : num  0 0 0 0 0 0 0.4 0 0.8 0.4 ...
##  $ Month                  : chr  "Feb" "Feb" "Feb" "Feb" ...
##  $ OperatingSystems       : int  1 2 4 3 3 2 2 1 2 2 ...
##  $ Browser                : int  1 2 1 2 3 2 4 2 2 4 ...
##  $ Region                 : int  1 1 9 2 1 1 3 1 2 1 ...
```

```
##  $ TrafficType            : int  1 2 3 4 4 3 3 5 3 2 ...
##  $ VisitorType            : chr  "Returning_Visitor" "Returning_Visitor"
"Returning_Visitor" "Returning_Visitor" ...
##  $ Weekend                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ Revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

```
## We then a statistical summary of our dataset
summary(customer_behavior)
```

```
##  Administrative   Administrative_Duration Informational
##  Min.   : 0.000   Min.   :  -1.00         Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.:   0.00         1st Qu.: 0.000
##  Median : 1.000   Median :   8.00         Median : 0.000
##  Mean   : 2.318   Mean   :  80.91         Mean   : 0.504
##  3rd Qu.: 4.000   3rd Qu.:  93.50         3rd Qu.: 0.000
##  Max.   :27.000   Max.   :3398.75         Max.   :24.000
##  NA's   :14       NA's   :14              NA's   :14
##  Informational_Duration ProductRelated   ProductRelated_Duration
##  Min.   :  -1.00        Min.   :  0.00   Min.   :   -1.0
##  1st Qu.:   0.00        1st Qu.:  7.00   1st Qu.:  185.0
##  Median :   0.00        Median : 18.00   Median :  599.8
##  Mean   :  34.51        Mean   : 31.76   Mean   : 1196.0
##  3rd Qu.:   0.00        3rd Qu.: 38.00   3rd Qu.: 1466.5
##  Max.   :2549.38        Max.   :705.00   Max.   :63973.5
##  NA's   :14             NA's   :14       NA's   :14
##   BounceRates         ExitRates         PageValues        SpecialDay
##  Min.   :0.000000   Min.   :0.00000   Min.   :  0.000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.01429   1st Qu.:  0.000   1st Qu.:0.00000
##  Median :0.003119   Median :0.02512   Median :  0.000   Median :0.00000
##  Mean   :0.022152   Mean   :0.04300   Mean   :  5.889   Mean   :0.06143
##  3rd Qu.:0.016684   3rd Qu.:0.05000   3rd Qu.:  0.000   3rd Qu.:0.00000
##  Max.   :0.200000   Max.   :0.20000   Max.   :361.764   Max.   :1.00000
##  NA's   :14         NA's   :14
##     Month           OperatingSystems    Browser           Region
##  Length:12330       Min.   :1.000    Min.   : 1.000   Min.   :1.000
##  Class :character   1st Qu.:2.000    1st Qu.: 2.000   1st Qu.:1.000
##  Mode  :character   Median :2.000    Median : 2.000   Median :3.000
##                     Mean   :2.124    Mean   : 2.357   Mean   :3.147
##                     3rd Qu.:3.000    3rd Qu.: 2.000   3rd Qu.:4.000
##                     Max.   :8.000    Max.   :13.000   Max.   :9.000
##
##   TrafficType    VisitorType         Weekend         Revenue
##  Min.   : 1.00   Length:12330       Mode :logical   Mode :logical
##  1st Qu.: 2.00   Class :character   FALSE:9462      FALSE:10422
##  Median : 2.00   Mode  :character   TRUE :2868      TRUE :1908
##  Mean   : 4.07
##  3rd Qu.: 4.00
##  Max.   :20.00
##
```

4. Perform Data Cleaning

To ensure uniformity, I will lowercase all the columns

```
names(customer_behavior)<- tolower(names(customer_behavior))
head(customer_behavior)
```

```
##   administrative administrative_duration informational
informational_duration
## 1              0                       0             0
0
## 2              0                       0             0
0
## 3              0                      -1             0
-1
## 4              0                       0             0
0
## 5              0                       0             0
0
## 6              0                       0             0
0
##   productrelated productrelated_duration bouncerates exitrates pagevalues
## 1              1                 0.000000  0.20000000 0.2000000          0
## 2              2                64.000000  0.00000000 0.1000000          0
## 3              1                -1.000000  0.20000000 0.2000000          0
## 4              2                 2.666667  0.05000000 0.1400000          0
## 5             10               627.500000  0.02000000 0.0500000          0
## 6             19               154.216667  0.01578947 0.0245614          0
##   specialday month operatingsystems browser region traffictype
## 1          0   Feb                1       1      1           1
## 2          0   Feb                2       2      1           2
## 3          0   Feb                4       1      9           3
## 4          0   Feb                3       2      2           4
## 5          0   Feb                3       3      1           4
## 6          0   Feb                2       2      1           3
##         visitortype weekend revenue
## 1 Returning_Visitor   FALSE   FALSE
## 2 Returning_Visitor   FALSE   FALSE
## 3 Returning_Visitor   FALSE   FALSE
## 4 Returning_Visitor   FALSE   FALSE
## 5 Returning_Visitor    TRUE   FALSE
## 6 Returning_Visitor   FALSE   FALSE
```

Change has been effeccted

NEXT I'm going to checking for missing values in our dataset,,Missing values may affect the perfomance of our model, so we will find a way to deal with them

```
##Checking for missing values in each row
colSums(is.na(customer_behavior))
```

```
##          administrative administrative_duration           informational
##                      14                       14                      14
##  informational_duration          productrelated productrelated_duration
##                      14                       14                      14
##             bouncerates               exitrates             pagevalues
##                      14                       14                       0
##              specialday                   month        operatingsystems
##                       0                       0                       0
##                 browser                  region             traffictype
##                       0                       0                       0
##             visitortype                 weekend                 revenue
##                       0                       0                       0
```

We have 14 missing values in the columns administrative, administrative_duration, information, information_duration, productrelated, productrelated_duration, bouncerates ,exitrates
Lets see how we will deal with the missing values

```
#### I will try and omit the missing values and see if it will affect out dataframe,
#### To do this, I will create a temporary dataframe
temp_df<- customer_behavior
temp_df <- na.omit(temp_df)
colSums(is.na(temp_df))
```

```
##          administrative administrative_duration           informational
##                       0                       0                       0
##  informational_duration          productrelated productrelated_duration
##                       0                       0                       0
##             bouncerates               exitrates             pagevalues
##                       0                       0                       0
##              specialday                   month        operatingsystems
##                       0                       0                       0
##                 browser                  region             traffictype
##                       0                       0                       0
##             visitortype                 weekend                 revenue
##                       0                       0                       0
```

```
##### lets check for the shape of our dataset
dim(temp_df)
```

```
## [1] 12316    18
```

```
customer_behavior <- na.omit(customer_behavior)
colSums(is.na(customer_behavior))
```

```
##          administrative administrative_duration           informational
##                       0                       0                       0
##  informational_duration          productrelated productrelated_duration
##                       0                       0                       0
##             bouncerates               exitrates             pagevalues
```

```
##                          0                          0                          0
##                specialday                      month         operatingsystems
##                          0                          0                          0
##                    browser                     region                traffictype
##                          0                          0                          0
##                visitortype                    weekend                    revenue
##                          0                          0                          0
```

```
duplicated_rows<- customer_behavior[duplicated(customer_behavior),]
#duplicated_rows
## This is awe-inspiring, we also dont have duplicated
```

We duplicated rows in our dataset,,,We will remove them

```
customer_behavior_new<- unique(customer_behavior)
##Lets confirm if change has been effected
# confirming from the data for any duplicated records
anyDuplicated(customer_behavior_new)
```

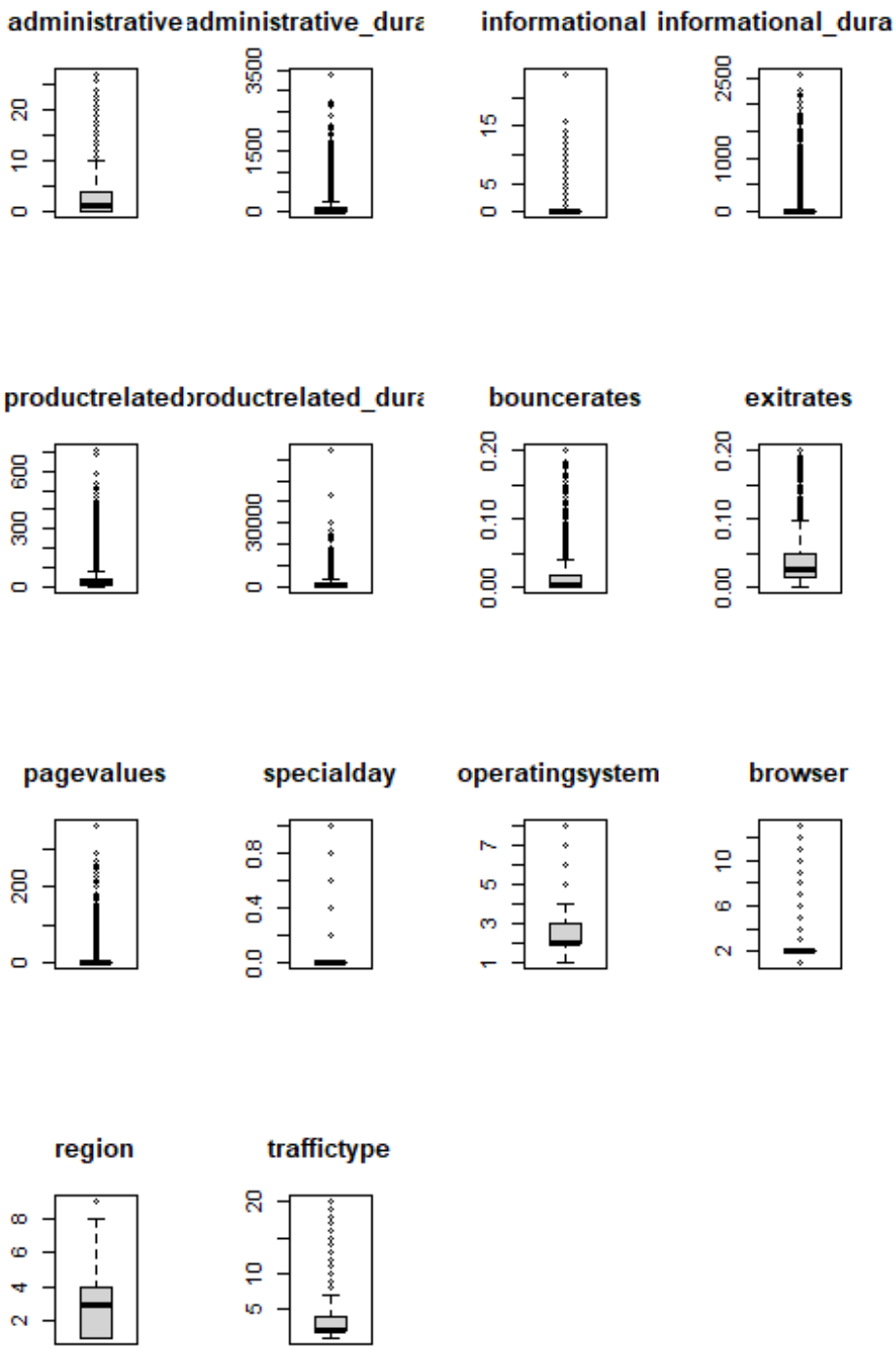
```
## [1] 0
```

```
## obtaining numerical columns
numeric_columns <- unlist(lapply(customer_behavior_new, is.numeric))
numeric_columns
```

```
##          administrative administrative_duration           informational
##                    TRUE                    TRUE                    TRUE
##  informational_duration          productrelated productrelated_duration
##                    TRUE                    TRUE                    TRUE
##             bouncerates               exitrates              pagevalues
##                    TRUE                    TRUE                    TRUE
##              specialday                   month         operatingsystems
##                    TRUE                   FALSE                    TRUE
##                 browser                  region                traffictype
##                    TRUE                    TRUE                    TRUE
##             visitortype                 weekend                 revenue
##                   FALSE                   FALSE                   FALSE
```

```
## I will put the numerical columns in a dataframe
columns_numeric <- customer_behavior_new[ , numeric_columns]
```

```
# using a for lop, I will output boxplots of numerical columns..This will
help me to identify the outliers
par ( mfrow= c (  2, 4 ))
for (i in 1 : length (columns_numeric)) {
boxplot (columns_numeric[,i], main= names (columns_numeric[i]), type= "l" )
}
```



##Next We will check anomalies and inconsistenicies in our dataframe….Since we have categorical column, we will need to convert them to factors

```
lengths(lapply(customer_behavior_new, unique))
```

```
##          administrative administrative_duration           informational
##                      27                    3336                      17
##  informational_duration          productrelated productrelated_duration
##                    1259                     311                    9552
##             bouncerates               exitrates              pagevalues
##                    1872                    4777                    2704
##              specialday                   month        operatingsystems
##                       6                      10                       8
##                 browser                  region             traffictype
##                      13                       9                      20
##             visitortype                 weekend                 revenue
##                       3                       2                       2
```

```
customer_behavior_new$revenue  <- as.factor(customer_behavior_new$revenue)
customer_behavior_new$visitortype<-
as.factor(customer_behavior_new$visitortype)
customer_behavior_new$weekend<- as.factor(customer_behavior_new$weekend)
customer_behavior_new$specialday<-
as.factor(customer_behavior_new$specialday)
customer_behavior_new$month<- as.factor(customer_behavior_new$month)
customer_behavior_new$region<- as.factor(customer_behavior_new$region)
```

```
## checking if change has been effected
str(customer_behavior_new)
```

```
## 'data.frame':    12199 obs. of  18 variables:
##  $ administrative         : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ administrative_duration: num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ informational          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ informational_duration : num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ productrelated         : int  1 2 1 2 10 19 1 1 2 3 ...
##  $ productrelated_duration: num  0 64 -1 2.67 627.5 ...
##  $ bouncerates            : num  0.2 0 0.2 0.05 0.02 ...
##  $ exitrates              : num  0.2 0.1 0.2 0.14 0.05 ...
##  $ pagevalues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ specialday             : Factor w/ 6 levels "0","0.2","0.4",..: 1 1 1 1
1 1 3 1 5 3 ...
##  $ month                  : Factor w/ 10 levels "Aug","Dec","Feb",..: 3 3
3 3 3 3 3 3 3 3 ...
##  $ operatingsystems       : int  1 2 4 3 3 2 2 1 2 2 ...
##  $ browser                : int  1 2 1 2 3 2 4 2 2 4 ...
##  $ region                 : Factor w/ 9 levels "1","2","3","4",..: 1 1 9 2
1 1 3 1 2 1 ...
##  $ traffictype            : int  1 2 3 4 4 3 3 5 3 2 ...
##  $ visitortype            : Factor w/ 3 levels "New_Visitor",..: 3 3 3 3 3
3 3 3 3 3 ...
##  $ weekend                : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 2 1
1 2 1 1 ...
##  $ revenue                : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1
1 1 1 1 ...
```

```
##  - attr(*, "na.action")= 'omit' Named int [1:14] 1066 1133 1134 1135 1136
1137 1474 1475 1476 1477 ...
##   ..- attr(*, "names")= chr [1:14] "1066" "1133" "1134" "1135" ...
```

5. Perform Exploratory Data Analysis (Univariate, Bivariate & Multivariate)

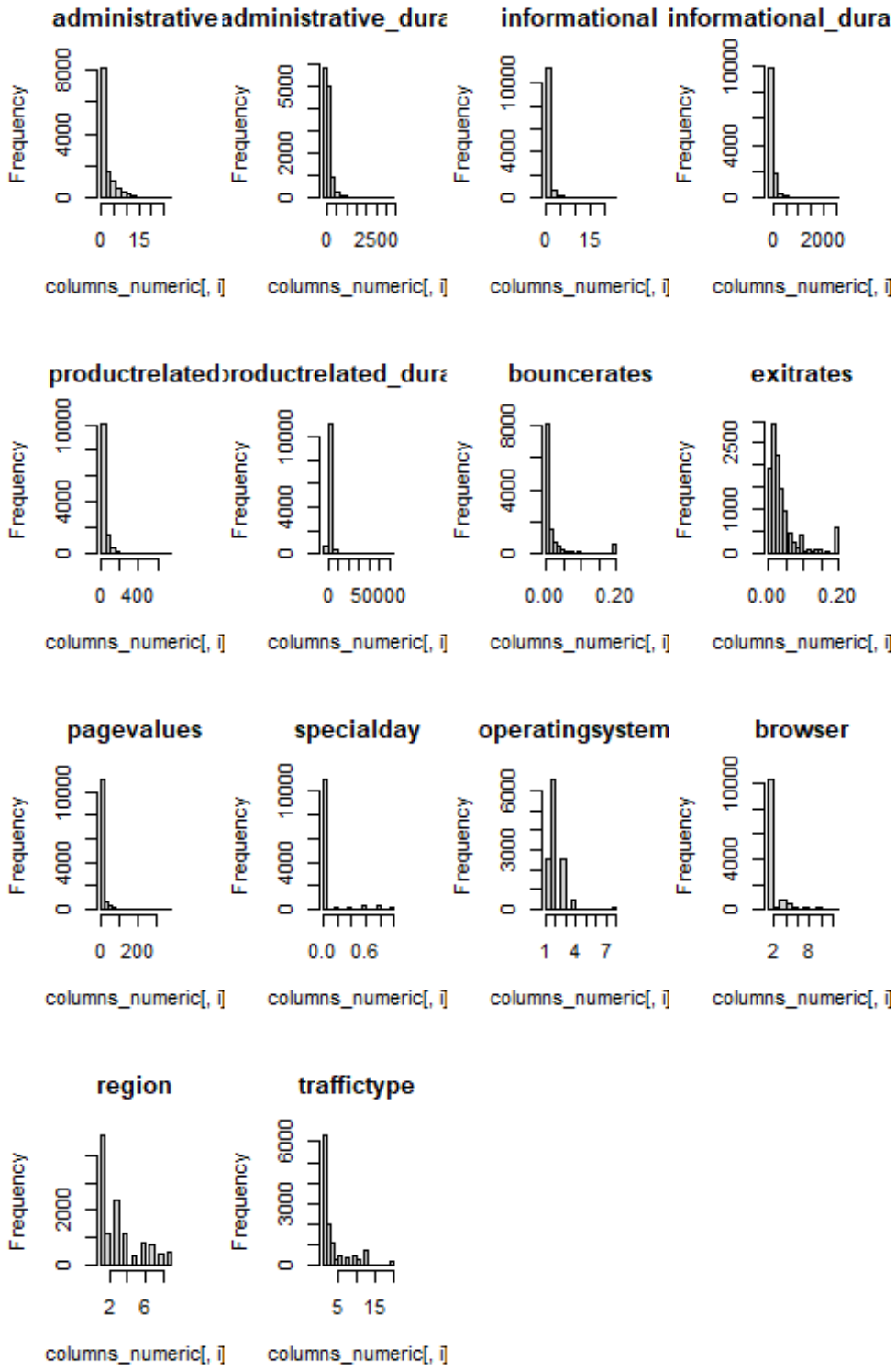
Univariate Analysis

```
summary(customer_behavior_new)

##  administrative  administrative_duration informational
##  Min.   : 0.00   Min.   :  -1.00         Min.   : 0.0000
##  1st Qu.: 0.00   1st Qu.:   0.00         1st Qu.: 0.0000
##  Median : 1.00   Median :   9.00         Median : 0.0000
##  Mean   : 2.34   Mean   :  81.68         Mean   : 0.5088
##  3rd Qu.: 4.00   3rd Qu.:  94.75         3rd Qu.: 0.0000
##  Max.   :27.00   Max.   :3398.75         Max.   :24.0000
##
##  informational_duration productrelated   productrelated_duration
##  Min.   :  -1.00        Min.   :  0.00   Min.   :   -1.0
##  1st Qu.:   0.00        1st Qu.:  8.00   1st Qu.:  193.6
##  Median :   0.00        Median : 18.00   Median :  609.5
##  Mean   :  34.84        Mean   : 32.06   Mean   : 1207.5
##  3rd Qu.:   0.00        3rd Qu.: 38.00   3rd Qu.: 1477.6
##  Max.   :2549.38        Max.   :705.00   Max.   :63973.5
##
##   bouncerates        exitrates         pagevalues      specialday
##  Min.   :0.00000   Min.   :0.00000   Min.   :  0.000   0  :10950
##  1st Qu.:0.00000   1st Qu.:0.01422   1st Qu.:  0.000   0.2:  178
##  Median :0.00293   Median :0.02500   Median :  0.000   0.4:  243
##  Mean   :0.02045   Mean   :0.04150   Mean   :  5.952   0.6:  350
##  3rd Qu.:0.01667   3rd Qu.:0.04848   3rd Qu.:  0.000   0.8:  324
##  Max.   :0.20000   Max.   :0.20000   Max.   :361.764   1  :  154
##
##      month      operatingsystems    browser           region
##  May    :3328   Min.   :1.000    Min.   : 1.000   1      :4711
##  Nov    :2983   1st Qu.:2.000    1st Qu.: 2.000   3      :2382
##  Mar    :1853   Median :2.000    Median : 2.000   4      :1168
##  Dec    :1706   Mean   :2.124    Mean   : 2.358   2      :1127
##  Oct    : 549   3rd Qu.:3.000    3rd Qu.: 2.000   6      : 800
##  Sep    : 448   Max.   :8.000    Max.   :13.000   7      : 758
##  (Other):1332                                     (Other):1253
##   traffictype                  visitortype     weekend      revenue
##  Min.   : 1.000   New_Visitor      : 1693   FALSE:9343   FALSE:10291
##  1st Qu.: 2.000   Other            :   81   TRUE :2856   TRUE : 1908
##  Median : 2.000   Returning_Visitor:10425
##  Mean   : 4.075
##  3rd Qu.: 4.000
```

```
##  Max.   :20.000
##
```

Histograms

```
par(mfrow=c(2, 4))
for (i in 1:length(columns_numeric)) {
        hist(columns_numeric[,i], main=names(columns_numeric[i]))
}
```



Bivariate

Analysis I will check on the relationship between our dependent variable and other columns

##Correlation matrix of all numerical columns

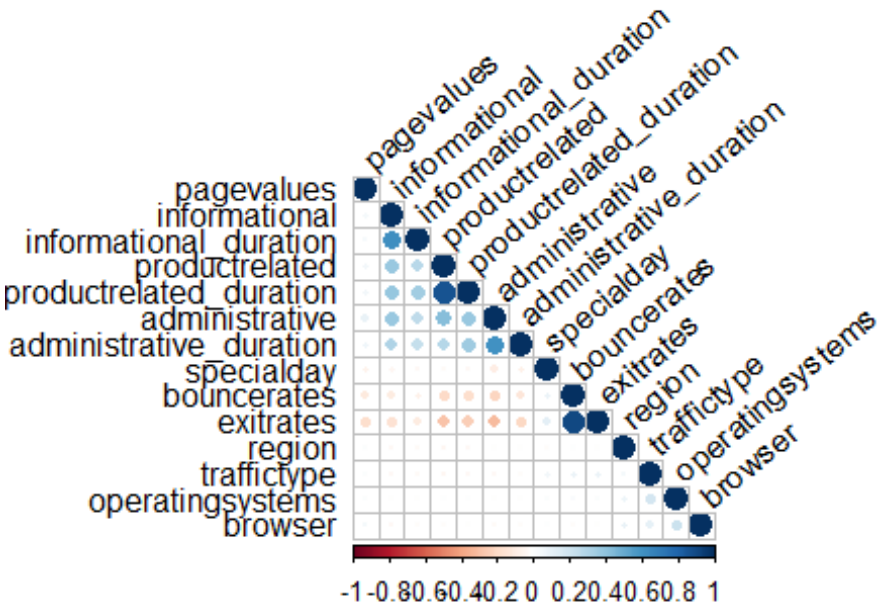
```r
correlations <- cor(columns_numeric, method = "pearson")
round(correlations, 2)
```

```
##                          administrative administrative_duration
informational
## administrative                     1.00                    0.60
0.38
## administrative_duration            0.60                    1.00
0.30
## informational                      0.38                    0.30
1.00
## informational_duration             0.25                    0.24
0.62
## productrelated                     0.43                    0.29
0.37
## productrelated_duration            0.37                    0.35
0.39
## bouncerates                       -0.21                   -0.14         -
0.11
## exitrates                         -0.31                   -0.20         -
0.16
## pagevalues                         0.10                    0.07
0.05
## specialday                        -0.10                   -0.07         -
0.05
## operatingsystems                  -0.01                   -0.01         -
0.01
## browser                           -0.03                   -0.02         -
0.04
## region                            -0.01                   -0.01         -
0.03
## traffictype                       -0.03                   -0.02         -
0.04
##                          informational_duration productrelated
## administrative                             0.25           0.43
## administrative_duration                    0.24           0.29
## informational                              0.62           0.37
## informational_duration                     1.00           0.28
## productrelated                             0.28           1.00
## productrelated_duration                    0.35           0.86
## bouncerates                               -0.07          -0.19
## exitrates                                 -0.10          -0.29
## pagevalues                                 0.03           0.05
## specialday                                -0.03          -0.03
## operatingsystems                          -0.01           0.00
## browser                                   -0.02          -0.01
## region                                    -0.03          -0.04
## traffictype                               -0.03          -0.04
##                          productrelated_duration bouncerates exitrates
## administrative                              0.37       -0.21     -0.31
```

```
## administrative_duration                    0.35       -0.14     -0.20
## informational                              0.39       -0.11     -0.16
## informational_duration                     0.35       -0.07     -0.10
## productrelated                             0.86       -0.19     -0.29
## productrelated_duration                    1.00       -0.17     -0.25
## bouncerates                               -0.17        1.00      0.90
## exitrates                                 -0.25        0.90      1.00
## pagevalues                                 0.05       -0.12     -0.17
## specialday                                -0.04        0.09      0.12
## operatingsystems                           0.00        0.03      0.02
## browser                                   -0.01       -0.02      0.00
## region                                    -0.03        0.00      0.00
## traffictype                               -0.04        0.09      0.09
##                          pagevalues specialday operatingsystems browser
region
## administrative                 0.10      -0.10            -0.01   -0.03  -
0.01
## administrative_duration        0.07      -0.07            -0.01   -0.02  -
0.01
## informational                  0.05      -0.05            -0.01   -0.04  -
0.03
## informational_duration         0.03      -0.03            -0.01   -0.02  -
0.03
## productrelated                 0.05      -0.03             0.00   -0.01  -
0.04
## productrelated_duration        0.05      -0.04             0.00   -0.01  -
0.03
## bouncerates                   -0.12       0.09             0.03   -0.02
0.00
## exitrates                     -0.17       0.12             0.02    0.00
0.00
## pagevalues                     1.00      -0.06             0.02    0.05
0.01
## specialday                    -0.06       1.00             0.01    0.00  -
0.02
## operatingsystems               0.02       0.01             1.00    0.21
0.07
## browser                        0.05       0.00             0.21    1.00
0.09
## region                         0.01      -0.02             0.07    0.09
1.00
## traffictype                    0.01       0.05             0.18    0.10
0.04
##                          traffictype
## administrative                 -0.03
## administrative_duration        -0.02
## informational                  -0.04
## informational_duration         -0.03
## productrelated                 -0.04
## productrelated_duration        -0.04
```

```
## bouncerates                    0.09
## exitrates                      0.09
## pagevalues                     0.01
## specialday                     0.05
## operatingsystems               0.18
## browser                        0.10
## region                         0.04
## traffictype                    1.00
```

```
library('corrplot')
corrplot(correlations, type = "lower", order = "hclust",tl.col = "black",
tl.srt = 40)
```



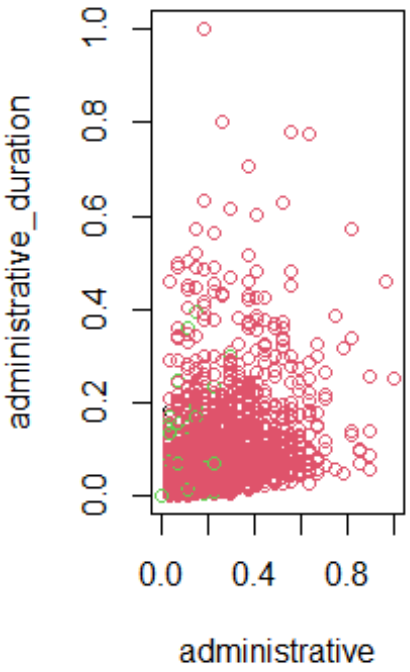

```
##K-Means Clustering
# Since clustering is a type of Unsupervised Learning, we would not require
Class Labelduring execution of our algorithm.
# We will, therefore, remove Class Attribute “Revenue” and store it in
another variable.
# We would then normalize the attributes between 0 and 1 using our own
function.

customer<- customer_behavior_new[,c(1,2,3,4,5,6,7,8,9)]
head(customer)

##   administrative administrative_duration informational
informational_duration
## 1              0                       0             0
```

```
0
## 2              0                        0             0
0
## 3              0                       -1             0
-1
## 4              0                        0             0
0
## 5              0                        0             0
0
## 6              0                        0             0
0
##   productrelated productrelated_duration bouncerates exitrates pagevalues
## 1              1                 0.000000  0.20000000 0.2000000          0
## 2              2                64.000000  0.00000000 0.1000000          0
## 3              1                -1.000000  0.20000000 0.2000000          0
## 4              2                 2.666667  0.05000000 0.1400000          0
## 5             10               627.500000  0.02000000 0.0500000          0
## 6             19               154.216667  0.01578947 0.0245614          0
```

```
# 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)))
}
##normalizing our columns
customer$administrative<- normalize(customer$administrative)
customer$administrative_duration<-
normalize(customer$administrative_duration)
customer$informational<- normalize(customer$informational)
customer$informational_duration<- normalize(customer$informational_duration)
customer$productrelated<- normalize(customer$productrelated)
customer$productrelated_duration<-
normalize(customer$productrelated_duration)
customer$bouncerates<- normalize(customer$bouncerates)
customer$exitrates<- normalize(customer$exitrates)
customer$pagevalues<- normalize(customer$pagevalues)
summary(customer)
```

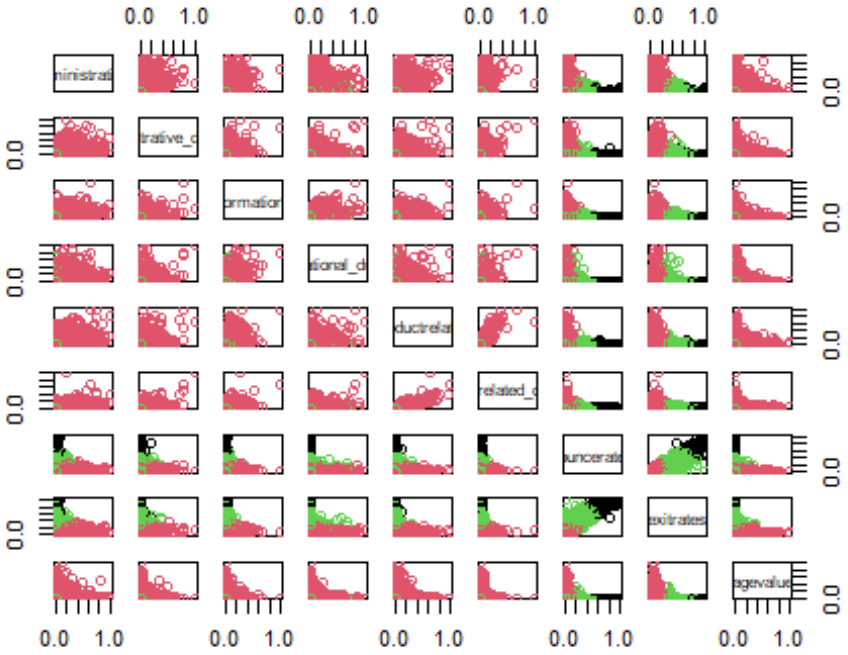
```
##  administrative    administrative_duration informational
##  Min.   :0.00000   Min.   :0.0000000       Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0002941       1st Qu.:0.0000
##  Median :0.03704   Median :0.0029414       Median :0.0000
##  Mean   :0.08667   Mean   :0.0243201       Mean   :0.0212
##  3rd Qu.:0.14815   3rd Qu.:0.0281638       3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.0000000       Max.   :1.0000
##  informational_duration productrelated    productrelated_duration
##  Min.   :0.0000000      Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.0003921      1st Qu.:0.01135   1st Qu.:0.003042
##  Median :0.0003921      Median :0.02553   Median :0.009543
##  Mean   :0.0140518      Mean   :0.04547   Mean   :0.018891
```

```
##  3rd Qu.:0.0003921      3rd Qu.:0.05390   3rd Qu.:0.023112
##  Max.   :1.0000000      Max.   :1.00000   Max.   :1.000000
##   bouncerates        exitrates         pagevalues
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.07111   1st Qu.:0.00000
##  Median :0.01465   Median :0.12500   Median :0.00000
##  Mean   :0.10223   Mean   :0.20748   Mean   :0.01645
##  3rd Qu.:0.08333   3rd Qu.:0.24242   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000

##WE can see that all our minimums are 0 and maximum is 1.....Our columns are
ready for clustering

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

## [1]  743 8834 2622

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

##   administrative administrative_duration informational
informational_duration
## 1    0.001046807            0.0006421843  0.0003925527
0.0003762675
## 2    0.114238758            0.0318647524  0.0274271753
0.0182117839
## 3    0.018038252            0.0056102655  0.0061181032
0.0039112633
##   productrelated productrelated_duration bouncerates exitrates  pagevalues
## 1    0.003497418            0.0006149543  0.91762855 0.9520778 0.000000000
## 2    0.057015528            0.0236245749  0.02227566 0.1000085 0.021898781
## 3    0.018478667            0.0081191161  0.14056728 0.3585917 0.002772636

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

# Visualizing the  clustering results
# ---
#
par(mfrow=c(1,2), mar=c(5,4,2,2))
plot(customer[,1:2], col=output$cluster) ##plot to see how administrative and
administrativeduration have been clustered
```



- This graph shows that Wind and Temp data points have not been clustered properly. Let us find out which attributes have been taken into consideration more by k-means algorithm. For this, we will plot all possible combinations of attributes!

```
plot(customer[,], col=output$cluster) # Plot to see all attribute
combinations
```



From the above plot, it can be seen that k-means algorithm has successfully clustered the columns but it is not that good,,,lets examine the hierachical clustering...

```
# As we don't want the hierarchical clustering result to depend to an
arbitrary variable unit,
# we start by scaling the data using the R function scale() as follows
customer_h<- customer_behavior_new[,c(1,2,3,4,5,6,7,8,9)]
head(customer_h)
```

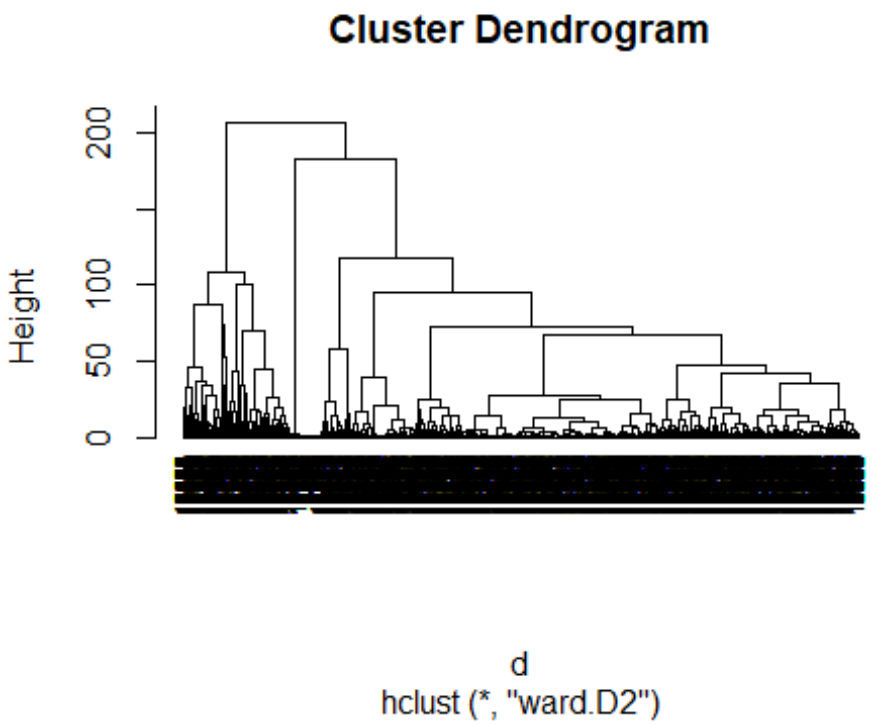
```
##   administrative administrative_duration informational
informational_duration
## 1              0                       0             0
0
## 2              0                       0             0
0
## 3              0                      -1             0
-1
## 4              0                       0             0
0
## 5              0                       0             0
0
## 6              0                       0             0
0
##   productrelated productrelated_duration bouncerates exitrates pagevalues
## 1              1                0.000000  0.20000000 0.2000000          0
## 2              2               64.000000  0.00000000 0.1000000          0
## 3              1               -1.000000  0.20000000 0.2000000          0
```

```
## 4              2                2.666667  0.05000000 0.1400000          0
## 5             10              627.500000  0.02000000 0.0500000          0
## 6             19              154.216667  0.01578947 0.0245614          0
```

```
customer_h <- scale(customer_h)
head(customer_h)
```

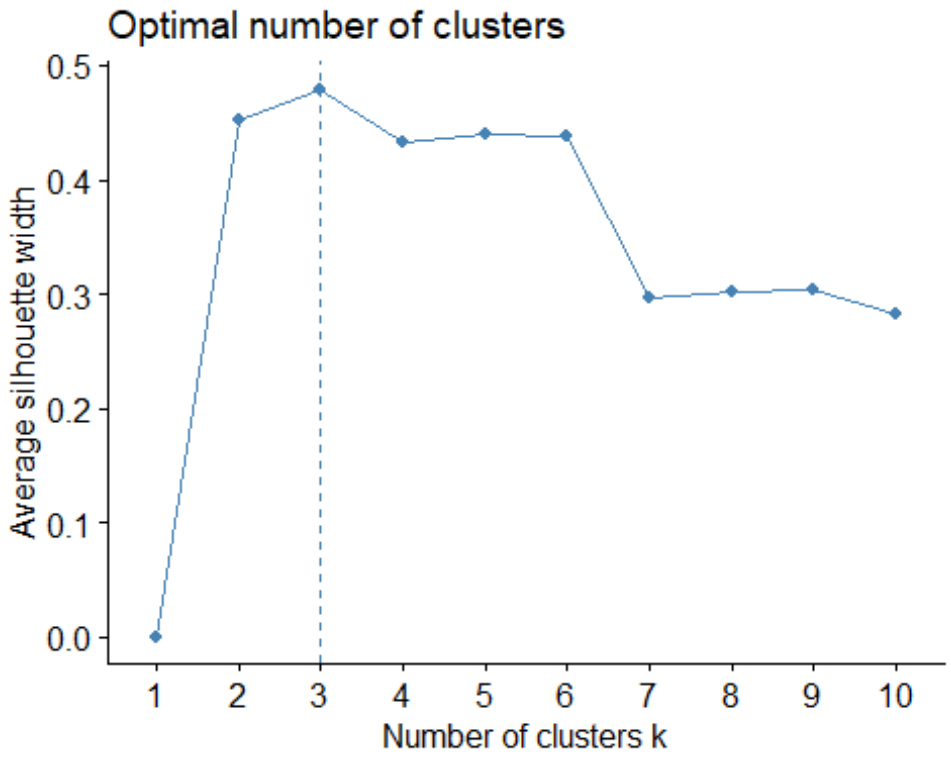
```
##   administrative administrative_duration informational
informational_duration
## 1     -0.7025315              -0.4601081    -0.3988128             -
0.2462725
## 2     -0.7025315              -0.4601081    -0.3988128             -
0.2462725
## 3     -0.7025315              -0.4657410    -0.3988128             -
0.2533417
## 4     -0.7025315              -0.4601081    -0.3988128             -
0.2462725
## 5     -0.7025315              -0.4601081    -0.3988128             -
0.2462725
## 6     -0.7025315              -0.4601081    -0.3988128             -
0.2462725
##   productrelated productrelated_duration  bouncerates  exitrates
pagevalues
## 1     -0.6963635              -0.6289343  3.954699721  3.4273070 -
0.3190356
## 2     -0.6739424              -0.5955997 -0.450343788  1.2650121 -
0.3190356
## 3     -0.6963635              -0.6294551  3.954699721  3.4273070 -
0.3190356
## 4     -0.6739424              -0.6275453  0.650917089  2.1299300 -
0.3190356
## 5     -0.4945739              -0.3020990 -0.009839437  0.1838646 -
0.3190356
## 6     -0.2927843              -0.5486101 -0.102577188 -0.3661929 -
0.3190356
```

```
# First we use the dist() function to compute the Euclidean distance between
observations,
# d will be the first argument in the hclust() function dissimilarity matrix
# ---
d <- dist(customer_h, method = "euclidean")
# We then hierarchical clustering using the Ward's method
# ---
#
res.hc <- hclust(d, method = "ward.D2" )
# Lastly, we plot the obtained dendrogram
# ---
#
plot(res.hc, cex = 0.6, hang = -1)
```



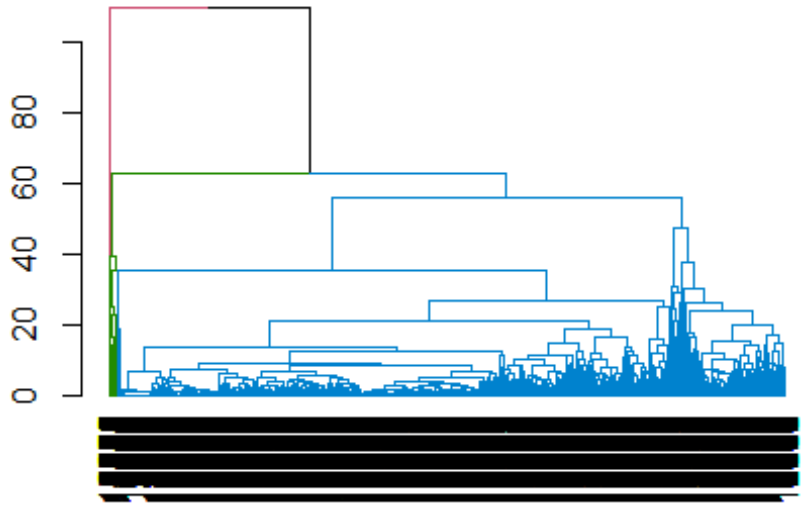

```
## Advanced Hierachical Clustering
library(ggplot2)
library(dplyr)
library(dendextend)

##
## ---------------------
## Welcome to dendextend version 1.15.2
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at:
https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:
suppressPackageStartupMessages(library(dendextend))
## ---------------------

##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree

library(factoextra)

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

library(cluster)
#Determine optimal number of clusters
#We will use fviz_nbclust() method to check optimum number of clusters using
silhouette, wss and gap_stat.
# Method 1 - Silhouette
#Clustering
head(customer_h)

##   administrative administrative_duration informational
informational_duration
## 1     -0.7025315             -0.4601081    -0.3988128             -
0.2462725
## 2     -0.7025315             -0.4601081    -0.3988128             -
0.2462725
## 3     -0.7025315             -0.4657410    -0.3988128             -
0.2533417
## 4     -0.7025315             -0.4601081    -0.3988128             -
0.2462725
## 5     -0.7025315             -0.4601081    -0.3988128             -
0.2462725
## 6     -0.7025315             -0.4601081    -0.3988128             -
0.2462725
##   productrelated productrelated_duration  bouncerates  exitrates
pagevalues
## 1     -0.6963635             -0.6289343  3.954699721  3.4273070 -
0.3190356
## 2     -0.6739424             -0.5955997 -0.450343788  1.2650121 -
0.3190356
## 3     -0.6963635             -0.6294551  3.954699721  3.4273070 -
0.3190356
## 4     -0.6739424             -0.6275453  0.650917089  2.1299300 -
0.3190356
## 5     -0.4945739             -0.3020990 -0.009839437  0.1838646 -
0.3190356
## 6     -0.2927843             -0.5486101 -0.102577188 -0.3661929 -
0.3190356

set.seed(123)
#Determining the number of optimal clusters
#Determining optimal number of Clusters (Cluster Evaluation Method 1)
fviz_nbclust(customer_h, FUN = hcut, method = "silhouette")
```



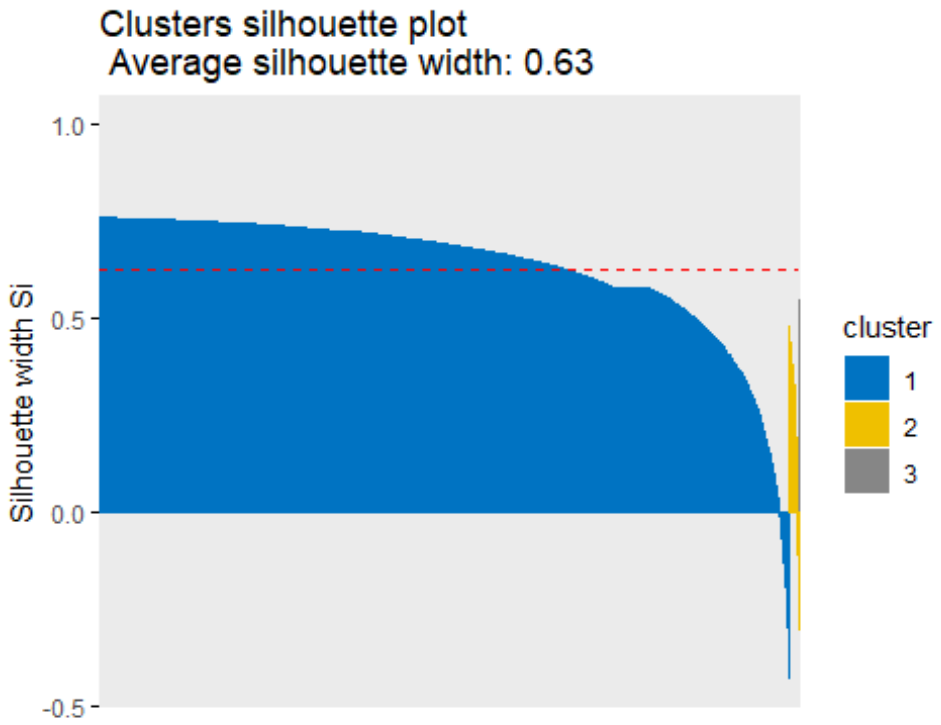
#The optimum number of clusters is 3

```
##Create distance matrix
#We now use manhattan distance formula to create a distance matrix. Using Manhattan distance, the silhouette plots obtained were better with higher co-efficient. Hence, we have used Manhattan distance.
#calculate manhattan distance
data2di <- dist(customer_h, method = "man")
##Now that we have created our distance matrix we can create our hierarchical cluster with optimal number of clusters as 3.
#Method 1 - Complete linkage
#complete
data2hc <- hclust(data2di, method = "complete")
data2as <- cutree(data2hc, k = 3)
dend_data <- as.dendrogram(data2hc)
cc <- color_branches(dend_data, k=3)
plot(cc)
```




```
sil <- silhouette(data2as, data2di)
fviz_silhouette(sil,palette= "jco",ggtheme = theme_minimal())
```

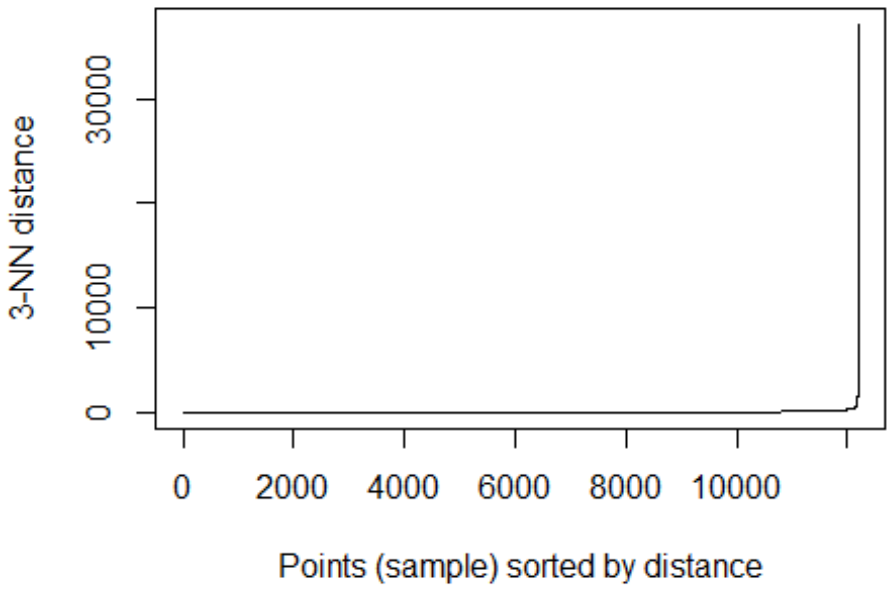
```
##   cluster  size ave.sil.width
## 1       1 12028          0.63
## 2       2   168          0.25
## 3       3     3          0.46
```



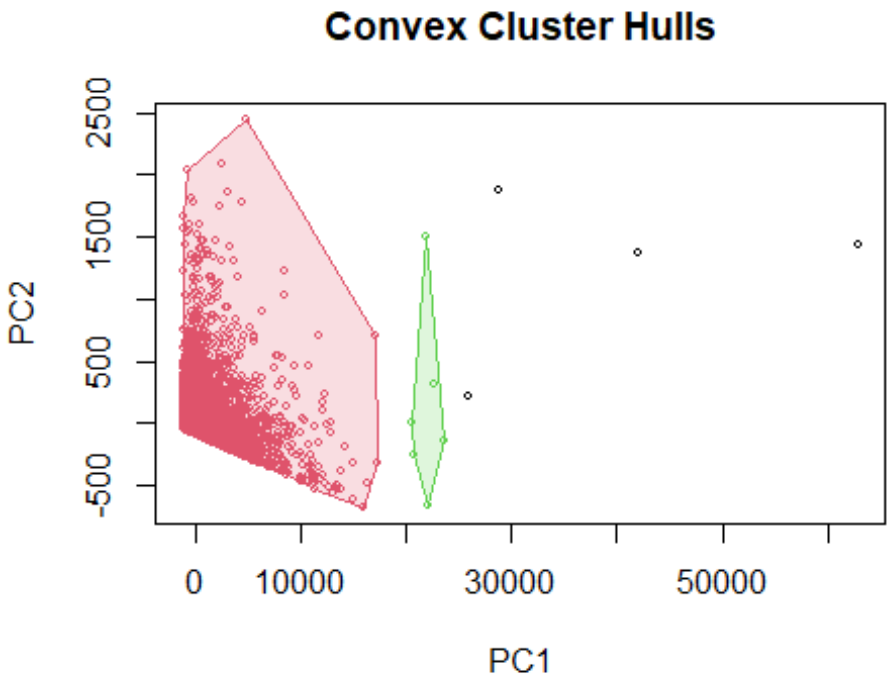
As seen in the plot above the average silhouette score is higher and only the grey cluster shows a minimal negative score. The negative score denotes a few observations are not in the right cluster.

7. Challenge the Solution

```
# We challenge the solution using DBSCAN algorithm to see if it performs
better clustering
# Loading necessary libraries
pacman :: p_load(dbscan)
# obtaining optimal nearest neighbours
kNNdistplot(customer_behavior_new[,1:9],k=3)
```




```
# shows optimal distance at approx 2000 for k value which we already know as
2 based on revenue class

# We want minimum 2 Cluster points with in a distance of eps(2000)
#
output_df <- dbscan(customer_behavior_new[,1:9],eps=2000,MinPts = 2,
borderPoints = TRUE)

## Warning in dbscan(customer_behavior_new[, 1:9], eps = 2000, MinPts = 2, :
## converting argument MinPts (fpc) to minPts (dbscan)!

output_df

## DBSCAN clustering for 12199 objects.
## Parameters: eps = 2000, minPts = 2
## The clustering contains 2 cluster(s) and 4 noise points.
##
##     0     1     2
##     4 12189     6
##
## Available fields: cluster, eps, minPts

# plot clusters using hullplot()
hullplot(customer_behavior_new[,1:9],output_df$cluster)
```



Conclusions 1. The K_Means gave us 3 clear clusters while the Hierarchical clustering model, clusters were crowded 2. Because our dataset was large, K-Means may be computationally faster than hierarchical clustering (if K is small). 4. Hierarchical clustering outputs a hierarchy, ie a structure that is more informative than the unstructured set of flat clusters returned by k-means.