Online Shopper Intention

Patrick Atak

18/07/2020

Research 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. More specifically, they would like to learn the characteristics of customer groups.

Metric of Success

In order to work on the above problem, I need to do the following:

1.Perform clustering stating insights drawn from your analysis and visualizations.

2.Upon implementation, provide comparisons between the approaches learned this week i.e. K-Means clustering vs Hierarchical clustering highlighting the strengths and limitations of each approach in the context of your analysis

Reading in the CSV

```
online <- read.csv("online_shoppers_intention.csv")
head(online)
```

Performing EDA

```
# Identifying missing data
is.na(online)

# Find total missing values in each column
colSums(is.na(online))

# Dealing with missing values
online <- na.omit(online)

# Previewing the structure
str(online)

# Preview dimension of the data
```

```r
dim(online)

# ======================

# Handling Outliers
boxplot.stats(online$Administrative)$Out

# Handling Duplicate Data
anyDuplicated(online)

# Finding duplicated rows
duplicated_row <- online[duplicated(online),]

# Removing duplicated rows
unique_items <- unique(online)
```

```r
# Univariate EDA
# Measures of central tendancy

x = unique_items$Administrative
# Mean
mean(x)

# Median
median(x)

# Max
max(x)

# Min
min(x)

# Range
range(x)

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

getmode(x)
```

```r
y = unique_items$ProductRelated_Duration

# Standard Deviation
sd(y)

# Quantile
quantile(y)

# Statistical Summary
summary(online)
```

Univariate & Bivariate Graphical EDA

Performing Unsupervised Learning

```
# K-Means Clustering


# 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 "Species" and store it in another variable.
# We would then normalize the attributes between 0 and 1 using our own function.
# ---
#

online.new<- online[, c(1, 2, 3, 4,5,6,7,8,9,10)]
head(online.new,3)
```

```
# 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)))
}
```

```
# Applying the K-means clustering algorithm with no. of centroids(k)=3
# ---
#
result<- kmeans(online.new,3)

# Previewing the no. of records in each cluster
#
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
# ---
#
result$cluster

# The graph shows that we have got 3 clearly distinguishable clusters
```

```
# Hierarchical 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
```

```r
# ---
#
online_hclust <- scale(unique_items[1:10])
head(online_hclust)
```

```r
# We now use the R function hclust() for hierarchical clustering
# ---
#

# 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(online_hclust, method = "euclidean")

# We then hierarchical clustering using the Ward's method
# ---
#
res <- hclust(d, method = "ward.D2" )
```

```r
# Lastly, we plot the obtained dendrogram
# ---
#
plot(res, cex = 0.6, hang = -1)
```

```r
# DBSCAN
# Loading the required library
# ---
#
library("dbscan")

# Selecting the necessary columns
# ---
#
m1<-unique_items[,c(1,2,3,4,5,6,7,8,9,10)]
head(m1)
```

```r
# Applying our DBSCAN algorithm
# ---
# We want minimum 4 points with in a distance of eps(0.4)
#
db<-dbscan(m1,eps=0.4,minPts = 4)
```

```r
# Printing out the clustering results
# ---
#
print(db)
```

```r
# We also plot our clusters as shown
# ---
# The dataset and cluster method of dbscan is used to plot the clusters.
```

```
#
hullplot(m1,db$cluster)
```