# Segment customers RFM for marketing efforts

## Segment customers based on their recency,frequency & monetary value (RFM) to better target marketing efforts.

* 1. Analyze customer transaction data to calculate RFM scores.
  2. Segment customers into different groups using clustering algorithms such as k-means or hierarchical clustering.
  3. Perform descriptive analysis on each customer segment to understand their characteristics.
  4. Develop targeted marketing strategies for each segment based on their RFM profiles.

Data preparation

#As a first step, I load all the modules that will be used in this notebook: data<-read.csv("F://Test.csv")

drops <- c("Var\_1","Segmentation") data<-data[ , !(names(data) %in% drops)]

#Using kable for aesthetics, viewing the top few rows of the dataset

knitr::kable(head(data))

#Checking for Missing Values

To find the number of NAs in each variable

lapply(data,function(x) { length(which(is.na(x)))})

To find the number of blanks in each variable

lapply(data,function(x) { length(which(x==""))})

There are blank values in categorical variables Graduated and Profession, replacing them with NAs for better handling data[data==""]<-NA

Replacing NA rows with mode

Creating a User defined function for finding Mode as R does not come with a base mode function getmode <- function(v) {

uniqv <- unique(v) uniqv[which.max(tabulate(match(v, uniqv)))]

}

Replacing categorical variable NAs with their mode and continuous variable NAs with Mean

data$Ever\_Married[is.na(data$Ever\_Married)]<-getmode(data$Ever\_Married) data$Graduated[is.na(data$Graduated)]<-getmode(data$Graduated) data$Profession[is.na(data$Profession)]<-getmode(data$Profession) data$Work\_Experience[is.na(data$Work\_Experience)]<-mean(data$Work\_Experience,na.rm=TRUE) data$Family\_Size[is.na(data$Family\_Size)]<-mean(data$Family\_Size,na.rm=TRUE)

The missing values have successfully been replaced with the mean and mode. To find the underlying trends of each of the variables, univariate distributions are plotted.

Data Characteristics

* The general age of customers in this dataset lies between 18–60.
* Customers have a work experience of 2 -4 years with a few outliers having more than 10 years of work experience
* Family size ranges between 2- 4
* There are more males than females in the customer base
* There are more married people than unmarried
* A major portion of the customers are graduates
* Artist is the most common profession seen among customers
* Most customers belong to the low spending score category

boxplot(data$Age, horizontal = TRUE,col = 'Purple',main="Age") boxplot(data$Work\_Experience, horizontal = TRUE,col = 'Orange',main="Work Ex") boxplot(data$Family\_Size, horizontal = TRUE,col = 'Blue',main="Family Size")

Boxplots of Continuous/ Ordinal Variables ggplot(data) + geom\_bar(aes(x = Gender)) ggplot(data) + geom\_bar(aes(x = Ever\_Married)) ggplot(data) + geom\_bar(aes(x = Graduated))

gplot(data) + geom\_bar(aes(x = Profession)) ggplot(data) + geom\_bar(aes(x = Spending\_Score))

Since the given dataset has a lot of information stored in categorical variables like gender, profession, ever married etc. that cannot be comprehended by our machine, we need to encode them into a numerical format.

There are two options available for doing so:

1. Rank encoding: You basically assign each of the different classes of the variable a unique number. For eg. while encoding Profession, Artist maybe assigned 1, Doctor as 2 and so on.
2. One Hot encoding: In this variation, each class of the categorical variable is spread across as a binary variable and marked 1 if it is present in the given data point and 0 if not. For eg. If spending score were to be split into one hot encoded variables, there would be three new variables called High, Medium and Low and for a particular customer with a high spending score, the variable called High would be encoded as 1.

The algorithm we’ll be using for clustering similar customers is K-Means (This video maybe helpful), which relies on the distance between two data points in an n dimensional space. It is prone to be affected by the magnitude of the underlying data, which means that a customer with an age of 19 would likely be far away from a customer with an age of 65 while plotting them as points in a n dimensional space, which makes sense for numerical variables. However if the variables are categorical like profession, on rank encoding, a rank of 7 is not necessarily greater than a rank of 6. In this case, it is usually recommended to use one hot encoding and not rank encoding, since there is a chance that the algorithm misunderstands a relationship between two numbers when there is none. One Hot encoding categorical variables

cluster\_data<-dummy\_cols(data)

Dropping columns not required for analysis

cat <- c("ID","Gender","Ever\_Married","Graduated","Profession","Spending\_Score") cluster\_data<-cluster\_data[ , !(names(cluster\_data) %in% cat)] knitr::kable(head(cluster\_data))

We are using all variables except ID for this exercise, since an ID variable is merely a unique identifier for each customer. It does not store any concealed information about the customer that should affect their segment and hence is excluded from the analysis.

As seen, the dataset is now reduced to a collection of neat numerical variables that can be easily understood by the machine.

To give a very rudimentary overview of what K Means does, it starts out with assigning k (a number that you specify) random cluster centers. The next step is to calculate the distance between these centers and all data points, like your traditional 2 dimensional coordinate geometry, only that the point is in a n dimensional space; something that our brain is not very adept at visualizing. The points are assigned to the cluster center closest to them. Once an iteration of such clustering data points is complete, the cluster centroid is recalculated and the process reiterated.

There is a well established method of identifying the optimal number of clusters (k) to choose using a plot called an Elbow or a Scree plot, which I am not delving into here. We are choosing to keep 4 clusters here, which is what the dataset originally came with.

install.packages("ClusterR")

install.packages("cluster")

library(cluster)

library(ClusterR)

set.seed(240) # Setting seed

kmeans\_1 <- kmeans(cluster\_data, centers = 4, nstart = 20) #kmeans\_1

#kmeans\_1$cluster

# Assigning the segments back to the dataset data$segment<-kmeans\_1$cluster knitr::kable(head(data))

Each data point, which represents a customer has now been assigned a cluster number ranging from 1–4. In order to better understand what each of these clusters i.e. customer segments represent, they are visualized on two dimensional scatter plots. Since it is not possible to visualize the cluster in a multi dimensional space using all the variables that were used, we have plotted three separate plots of clusters using the continuous variables of Age, Work Experience and Family Size.

y\_kmeans <- kmeans\_1$cluster

# Visualizing segments in terms of Work experience and Age clusplot(data[, c("Age", "Work\_Experience")],

y\_kmeans, lines = 0, shade = TRUE, color = TRUE,

labels = 0, # To remove data labels from the plot plotchar = TRUE,

span = TRUE,

main = paste("Customer Segments"),

xlab = 'Age',

ylab = 'Work\_Experience')

clusplot(data[, c("Family\_Size", "Work\_Experience")], y\_kmeans,

lines = 0, shade = TRUE, color = TRUE, labels = 0,

plotchar = FALSE, span = TRUE,

main = paste("Customer Segments"), xlab = 'Family\_Size',

ylab = 'Work\_Experience')

clusplot(data[, c("Family\_Size", "Age")], y\_kmeans,

lines = 0, shade = TRUE, color = TRUE, labels = 0,

plotchar = FALSE, span = TRUE,

main = paste("Customer Segments"), xlab = 'Family\_Size',

ylab = 'Age')

Cluster Profiling

Summarizing customer attributes in each of the segments; data %>% group\_by(segment) %>%

summarise(Min\_Age=min(Age), Work\_Experience=mean(Work\_Experience),Family\_Size=mean(Family\_Size), Graduated=getmode(Graduated),Gender=getmode(Gender),

Married=getmode(Ever\_Married),Profession=getmode(Profession), Spend=getmode(Spending\_Score))

knitr::kable(segment\_summary)

### Output:

**Interpretation and Implications:**

On understanding the average attributes we observe that the age group varies greatly across each of the segments while family size and Work experience are comparable across segments 1 and 4.

Most of the customer segments are dominated by males, which is the overall trend in the underlying dataset as well. Segments 1,2 and 4 are dominated by Married persons while segment 3 has more unmarried people, which is in line with the observation that it is a younger segment of customers with a minimum age of 18.

It is also interesting to note that the segments 2 & 3 with younger customers i.e. 18–45 year old have low spend scores whereas segments 1 and 4 with middle aged and elderly customers have average to high spend scores.

Based on this analysis, the automobile dealer’s marketing team can identify what kind of communication to send to each of these segments. Segments 1 and 4 can be targeted for luxury Sedans since they have good spending scores and not typically large family sizes. Segment 3 can be targeted for entry level cars and Segment 2 can be targeted for mid sized SUVs.