**NAME**

NOBONITA NONDE

**ID**

20-43819-2

**COURSE**

INTRODUCTION TO DATA SCIENCE

**SECTION**

E

**DEPARTMENT**

COMUTER SCIENCE AND ENGINEERING

**SUBMISSION DATE**

30-04-2023

# Introduction:

K-means may be a clustering calculation broadly utilized in information science for gathering information focuses into clusters based on their likeness. The calculation points to play down the entirety of squared separations between information focuses and their doled-out cluster center. It begins by arbitrarily selecting an indicated number of cluster centers and iteratively allotting information focuses to the closest cluster center, at that point recalculating the cluster center based on the normal of all data points in that cluster. This handle proceeds until the cluster centers now not alter or the calculation comes to an indicated number of cycles. K-means could be a capable and broadly utilized apparatus for client division, picture division, and peculiarity location, among other applications. In any case, it is touchy to starting cluster center arrangement and may battle with non-spherical or covering clusters.

# Data Descriptions

This dataset contains information on Bank Customer Churn Prediction. The dataset includes 12 columns for every customer. There are

1. customer_id: Every customer has a customer_Id that includes this column. Every Id will have different from one another.

2. credit_score: Credit score represents a prediction of customers credit behavior and gives a score that includes this column. A higher credit score typically indicates that the customer is less likely to default on loans or other financial obligations, which may make them less likely to churn.

3. Country: This column specifies the different Countries of the customers. This attribute represents the country where the customer resides, either France, Spain, or Germany.

4. Gender: Gender column specifies the gender of the customers, with values of Male or Female. Gender represents with special values that is 0 and 1. 0 for Male and 1 for Female.

5. Age: Age column specifies the age of the customers. The age of the customer is recorded in a long time. Age might possibly be a critical figure in anticipating churn in case there are contrasts in customer behavior or preferences based on age.

6. Tenure: This quality speaks to the number of a long time the customer has been with the bank. Customers who have been with the bank for a longer period of time may be less likely to churn due to a more grounded relationship with the bank.

7. Balance: This trait speaks to the customer's account adjust. Customers with higher account equalizations may be less likely to churn due to their speculation within the bank.

8. Products_number: This quality speaks to the number of bank items the customer has acquired. Customers who have acquired more items may be less likely to churn due to their expanded speculation within the bank.

9. credit_card: This property speaks to whether or not the customer includes a credit card. Customers with a credit card may be more likely to churn in the event that they have higher credit card equalizations or in case they are troubled with the bank's credit card offerings.

10. Active_member: This property speaks to whether or not the customer is a dynamic part. Dynamic individuals may be less likely to churn due to their engagement with the bank.

11. Estimated_salary: This property speaks to the customer's assessed compensation. Customers with higher pay rates may be less likely to churn due to their higher wage.

12. Churn: churn alludes to the circumstance where a customer stop utilizing the bank's administrations and closes their account.

# Approach:

## Missing Value:

Missing values were checked by this code and there are no missing values.

```
sum(is.na(dataset))
```

```
> sum(is.na(dataset))
[1] 0
```

## Data Type:

Data Type checked by this code. I have found that here are three types of data types.

```
str(dataset)
```

```
> str(dataset) # To seen the data types and Missing values per column
'data.frame':   10000 obs. of  12 variables:
 $ customer_id     : int  15634602 15647311 15619304 15701354 15737888 15574012 15592531 15656148 1
5792365 15592389 ...
 $ credit_score    : int  619 608 502 699 850 645 822 376 501 684 ...
 $ country         : chr  "France" "Spain" "France" "France" ...
 $ gender          : chr  "Female" "Female" "Female" "Female" ...
 $ age             : int  42 41 42 39 43 44 50 29 44 27 ...
 $ tenure          : int  2 1 8 1 2 8 7 4 4 2 ...
 $ balance         : num  0 83808 159661 0 125511 ...
 $ products_number : int  1 1 3 2 1 2 2 4 2 1 ...
 $ credit_card     : int  1 0 1 0 1 1 1 1 0 1 ...
 $ active_member   : int  1 1 0 0 1 0 1 0 1 1 ...
 $ estimated_salary: num  101349 112543 113932 93827 79084 ...
 $ churn           : int  1 0 1 0 0 1 0 1 0 0 ...
>
```

# Dataset Summary:

Dataset summary checked by this code and get mean mode and median. It checks every attribute and give result each attribute has how many missing values.

```
summary(dataset)
```

```
 Median :15690738   Median :652.0   Mode  :character   Mode  :character   Median :37.00
 Mean   :15690941   Mean   :650.5                                         Mean   :38.92
 3rd Qu.:15753234   3rd Qu.:718.0                                         3rd Qu.:44.00
 Max.   :15815690   Max.   :850.0                                         Max.   :92.00
     tenure          balance       products_number  credit_card     active_member
 Min.   : 0.000   Min.   :     0   Min.   :1.00    Min.   :0.0000   Min.   :0.0000
 1st Qu.: 3.000   1st Qu.:     0   1st Qu.:1.00    1st Qu.:0.0000   1st Qu.:0.0000
 Median : 5.000   Median : 97199   Median :1.00    Median :1.0000   Median :1.0000
 Mean   : 5.013   Mean   : 76486   Mean   :1.53    Mean   :0.7055   Mean   :0.5151
 3rd Qu.: 7.000   3rd Qu.:127644   3rd Qu.:2.00    3rd Qu.:1.0000   3rd Qu.:1.0000
 Max.   :10.000   Max.   :250898   Max.   :4.00    Max.   :1.0000   Max.   :1.0000
 estimated_salary        churn
 Min.   :    11.58   Min.   :0.0000
 1st Qu.: 51002.11   1st Qu.:0.0000
 Median :100193.91   Median :0.0000
 Mean   :100090.24   Mean   :0.2037
 3rd Qu.:149388.25   3rd Qu.:0.0000
 Max.   :199992.48   Max.   :1.0000
```

**Categorize the male and female:** I have found that there is attribute called Gender that has two levels which is Male and Female, so I have converted it to 0 and 1.

```
dataset$gender  <- factor(dataset$gender , levels = c("Male", "Female"), labels = c(0, 1))

head(dataset)
```

```
> dataset$gender  <- factor(dataset$gender , levels = c("Male", "Female"), labels = c(0, 1))
> head(dataset)
  customer_id credit_score country gender age tenure   balance products_number credit_card
1    15634602          619  France      1  42      2      0.00               1           1
2    15647311          608   Spain      1  41      1  83807.86               1           0
3    15619304          502  France      1  42      8 159660.80               3           1
4    15701354          699  France      1  39      1      0.00               2           0
5    15737888          850   Spain      1  43      2 125510.82               1           1
6    15574012          645   Spain      0  44      8 113755.78               2           1
  active_member estimated_salary churn
1             1       101348.88     1
2             1       112542.58     0
3             0       113931.57     1
4             0        93826.63     0
5             1        79084.10     0
6             0       149756.71     1
>
```

**Feature Selection:** It is the process where the essential attributes are taken for statistics. It specifies the value and can be worked without non-essential data

```
dataset_data <- select(dataset, -c(customer_id, country))
```

```
> dataset_data
    credit_score gender age tenure   balance products_number credit_card active_member
1            619      2  42      2      0.00               1           1             1
2            608      2  41      1  83807.86               1           0             1
3            502      2  42      8 159660.80               3           1             0
4            699      2  39      1      0.00               2           0             0
5            850      2  43      2 125510.82               1           1             1
6            645      1  44      8 113755.78               2           1             0
7            822      1  50      7      0.00               2           1             1
8            376      2  29      4 115046.74               4           1             0
9            501      1  44      4 142051.07               2           0             1
10           684      1  27      2 134603.88               1           1             1
11           528      1  31      6 102016.72               2           0             0
12           497      1  24      3      0.00               2           1             0
13           476      2  34     10      0.00               2           1             0
14           549      2  25      5      0.00               2           0             0
15           635      2  35      7      0.00               2           1             1
16           616      1  45      3 143129.41               2           0             1
17           653      1  58      1 132602.88               1           1             0
18           549      2  24      9      0.00               2           1             1
```

**Numeric Conversion:** I have converted all the attribute data types to numeric formats.

```
dataset_data$credit_score <- as.numeric(dataset_data$credit_score)
dataset_data$gender <- as.numeric(dataset_data$gender)
dataset_data$age <- as.numeric(dataset_data$age)
dataset_data$tenure <- as.numeric(dataset_data$tenure)
dataset_data$balance <- as.numeric(dataset_data$balance)
dataset_data$products_number <- as.numeric(dataset_data$products_number)
dataset_data$credit_card <- as.numeric(dataset_data$credit_card)
dataset_data$active_member <- as.numeric(dataset_data$active_member)
dataset_data$estimated_salary <- as.numeric(dataset_data$estimated_salary)
dataset_data$churn <- as.numeric(dataset_data$churn)

str(dataset_data)
```

```
> str(dataset_data)
'data.frame':   10000 obs. of  10 variables:
 $ credit_score    : num  619 608 502 699 850 645 822 376 501 684 ...
 $ gender          : num  2 2 2 2 2 1 1 2 1 1 ...
 $ age             : num  42 41 42 39 43 44 50 29 44 27 ...
 $ tenure          : num  2 1 8 1 2 8 7 4 4 2 ...
 $ balance         : num  0 83808 159661 0 125511 ...
 $ products_number : num  1 1 3 2 1 2 2 4 2 1 ...
 $ credit_card     : num  1 0 1 0 1 1 1 1 0 1 ...
 $ active_member   : num  1 1 0 0 1 0 1 0 1 1 ...
 $ estimated_salary: num  101349 112543 113932 93827 79084 ...
 $ churn           : num  1 0 1 0 0 1 0 1 0 0 ...
>
```

**k-means:** After doing the preprocessing I just trained my model by using 4 clusters.

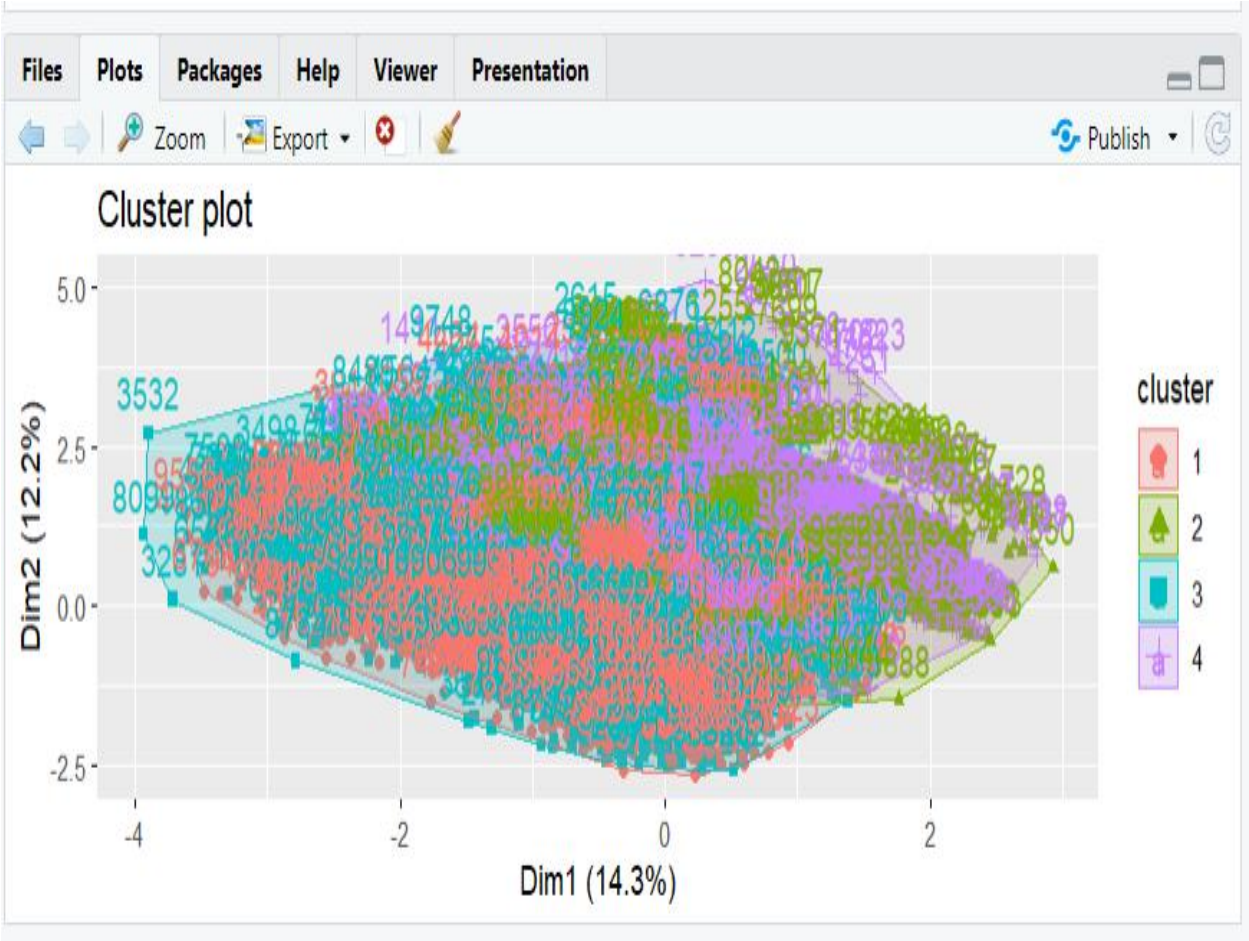
```
data_kmeans <- kmeans(dataset_data, centers = 4, nstart = 25)
```

```
> data_kmeans
K-means clustering with 4 clusters of sizes 3066, 1913, 3132, 1889

Cluster means:
  credit_score   gender      age   tenure    balance products_number credit_card active_member
1     651.6265 1.441292 39.40541 4.961840 121880.265        1.367906   0.6999348     0.5136986
2     650.0178 1.450601 38.54731 5.052797   2291.898        1.771040   0.7276529     0.5248301
3     650.7925 1.462005 38.98372 5.002554 121934.455        1.401980   0.6947637     0.5130907
4     648.8274 1.466384 38.41345 5.071996   2589.142        1.762308   0.7098994     0.5108523
  estimated_salary     churn
1         50472.48 0.2374429
2         49776.23 0.1385259
3        149729.91 0.2429757
4        149273.58 0.1498147

Clustering vector:
   [1] 4 3 3 2 1 3 2 3 1 1 1 2 2 4 2 1 1 2 4 2 4 4 4 2 4 4 3 2 3 2 4 3 1 2 4 1 3 1 2 3 2 3 1 3 3
  [46] 3 3 3 1 3 3 1 4 3 1 3 1 1 2 2 1 3 4 1 1 1 3 3 3 1 1 2 1 1 4 1 4 4 3 1 3 4 2 3 2 4 3 4 2 3
  [91] 4 3 4 4 2 3 1 3 4 2 4 4 1 2 4 1 1 1 1 1 1 1 1 1 3 1 3 2 3 3 1 3 4 3 4 3 2 1 4 3 4 3 1 2 3
 [136] 1 3 1 1 1 3 4 3 4 4 1 3 1 1 2 2 4 4 1 4 2 2 2 3 3 1 2 1 4 1 3 3 3 2 2 3 3 1 3 4 1 4 3 1 1
```

# Cluster Plot:

After training I have 4 clusters here. we have utilized center=4 meaning there will be 4 clusters. Those 4 clusters are highlighted through 4 distinctive colors.

```
fviz_cluster(data_kmeans, data = dataset_data)
```



# Result:

The clusters have sizes of 3066, 1913, 3132, and 1889. The cluster implies are too given, demonstrating the cruel values of the factors in each cluster. The factors included are credit score, sexual orientation, age, residency, adjust, items number, credit card, dynamic part, evaluated compensation, and churn. At long last, a clustering vector is given, which allots each perception to one of the four clusters.