Credit Card Segmentation

Abstraction:

In this project we need to find different types of customers based on their card usage, with the help of attributes like customer id, monthly cash advance, balance frequency, online/offline purchases,

Instalment purchases, etc. We need to suggest marketing strategy for the bank/organisation.

Here we used R studio and Jupyter Notebook as platforms to work on the project. At first, we need to clean data using Exploratory Data Analysis like check for Missing Values, Outliners, then Feature Extraction, Feature Selection, Feature Scaling. Achieving the goal was quite challenging. Using Machine learning concepts like Unsupervised learning, we do some clustering using K-means cluster, in order to categorize the customers based on their usage of card, which helps us to suggest a marketing strategy for the bank/organisation/firm.

Project Index

1. INTRODUCTION:

Credit card segmentation is a process of filtering the customers based on their purchases and try to develop some strategies in order to run the bank/organisation.

In this project, our objective is to categorize the customers based on their interest. The data contains 8950 observations and 19 variables. i.e Customer id, Balance, Balance Frequency, Purchases, Online/offline purchases, Instalment Purchases, Cash advance, purchase frequency, online/offline purchase frequency, Purchase instalments frequency, cash advance frequency, cash advance transaction, purchase transaction, credit limit, payments, minimum payments, PRC full payments, Tenure.

Our aim is to categorize the customers based on their interest of their credit card transactions and purchases, we need to suggest the marketing strategy to the bank/organisation.

2. Problem Statement:

This case requires trainees to develop a customer segmentation to define marketing strategy. The sample dataset summarizes the usage behaviour of about 9000 active credit card holders during the last 6 months. The file is at a customer level with 18 behavioural variables.

Number of attributes:

- Customer id
- Balance
- Balance Frequency
- Purchases
- Online/offline purchases
- Instalment Purchases
- Cash advance
- purchase frequency
- online/offline purchase frequency
- Purchase instalments frequency
- cash advance frequency
- cash advance transaction
- purchase transaction
- credit limit
- minimum payments
- PRC full payments
- Tenure

3. Loading Data into R:

After installing important packages.

Here we are using R studio and we are loading data into R environment. Using following command:

```
orginal_data = read.csv("C:/Users/Hp/Desktop/BLESSED 2/credit-card-data.csv",stringsAsFactors = FALSE)
```

```
h = orginal_data
```

All variables are in the correct structure, no need to re-adjust their structure. And eventually, all the variables are numerics.

We will take out the CUST_ID since it was a unique variable and we can’t get further information from it.

```
h = h[,c(-1)]
```

4.Feature Extraction (Deriving New KPI):

I. Monthly average purchase

We are creating Monthly average purchase from Purchases and Tenure, using following command:

```
h$Monthly_Avg_PURCHASES =h
$PURCHASES/(h$PURCHASES_FREQUENCY*h$TENURE)
```

II. Monthly cash advance

We are creating Monthly cash advance from Purchases and Tenure, using following command:

```
h$Monthly_CASH_ADVANCE <-
h$CASH_ADVANCE/(h$CASH_ADVANCE_FREQUENCY*h$TENURE
)
```

III. Limit Usage:

Here we are creating new variable called limit usage from two variables Balance and credit limit. Using following command.

```
h$LIMIT_USAGE <- h$BALANCE/h$CREDIT_LIMIT
```

IV. Minimum payments ratio:

Here we are creating new variable called Minimum payments ratio from payments and minimum payments ratio. Using following command.

```
h$MIN_PAYMENTS_RATIO <-
h$PAYMENTS/h$MINIMUM_PAYMENTS
```

6. Outlier Analysis

Here we create a function to remove outliers in our dataset, using following command line function:

```
mystats <- function(x) {
 nmiss<-sum(is.na(x))
 a <- x[!is.na(x)]
 m <- mean(a)
 n <- length(a)
 s <- sd(a)
 min <- min(a)
 p1<-quantile(a,0.01)
 p5<-quantile(a,0.05)
 p10<-quantile(a,0.10)
 q1<-quantile(a,0.25)
 q2<-quantile(a,0.5)
 q3<-quantile(a,0.75)
 p90<-quantile(a,0.90)
 p95<-quantile(a,0.95)
 p99<-quantile(a,0.99)
 max <- max(a)
 UC <- m+2*s
 LC <- m-2*s
 outlier_flag<- max>UC | min<LC
 return(c(n=n, nmiss=nmiss, outlier_flag=outlier_flag, mean=m, stdev=s,min = min,
p1=p1,p5=p5,p10=p10,q1=q1,q2=q2,q3=q3,p90=p90,p95=p95,p99=p99,max=max, UC=UC, LC=LC ))
}
Outliers<-t(data.frame(apply(h[Numerical_Variabless], 2, mystats)))
```

7.Missing value Analysis:

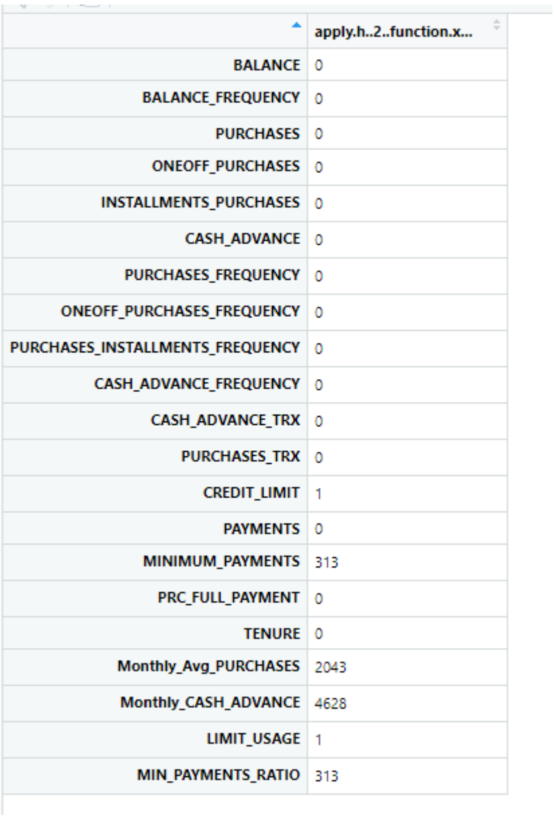
Here we are check for missing values in the dataset like empty rows which was filled with Na. we found some missing values in our dataset. Now missing values can be found by using different techniques like mean, median and Knn imputation.

First, we are selecting 1000 rows randomly and performing mean, median, Knn imputation, so that we can choose any one by comparing actual value and predicted value.

There are 3 types of methods to predict the missing values.

1. MEAN method
2. MEDIAN method
3. KNN

We need to decide which method is suitable for our dataset, for that we are randomly taking actual value of observation of credit limit, minimum payment variables and check the predicted values of above 3 methods with actual value and decide which method is suitable. Here checking for missing values.

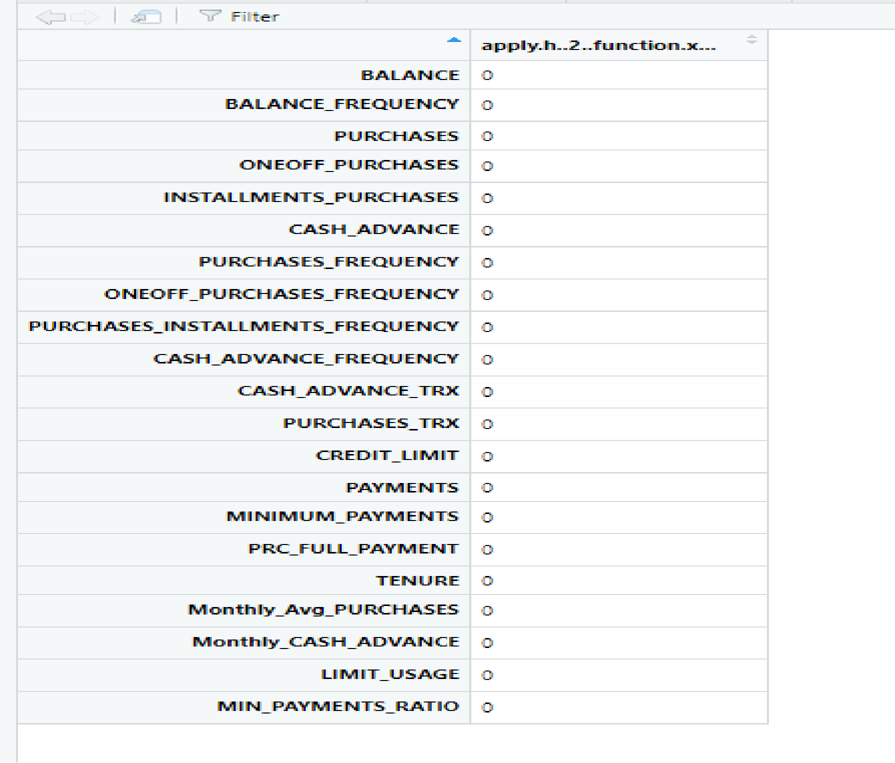


	apply.h..2..function.x...
BALANCE	0
BALANCE_FREQUENCY	0
PURCHASES	0
ONEOFF_PURCHASES	0
INSTALLMENTS_PURCHASES	0
CASH_ADVANCE	0
PURCHASES_FREQUENCY	0
ONEOFF_PURCHASES_FREQUENCY	0
PURCHASES_INSTALLMENTS_FREQUENCY	0
CASH_ADVANCE_FREQUENCY	0
CASH_ADVANCE_TRX	0
PURCHASES_TRX	0
CREDIT_LIMIT	1
PAYMENTS	0
MINIMUM_PAYMENTS	313
PRC_FULL_PAYMENT	0
TENURE	0
Monthly_Avg_PURCHASES	2043
Monthly_CASH_ADVANCE	4628
LIMIT_USAGE	1
MIN_PAYMENTS_RATIO	313

Here after comparing the three missing value techniques, we came to conclusion that, we can use mean method, to impute our missing values in R. Using following commands.

```
h$MINIMUM_PAYMENTS[which(is.na(h$MINIMUM_PAYMENTS))] <- 721.9256368

h$CREDIT_LIMIT[which(is.na(h$CREDIT_LIMIT))] <- 4343.62

h$Monthly_Avg_PURCHASES[which(is.na(h$Monthly_Avg_PURCHASES))] <-184.8991609

h$Monthly_CASH_ADVANCE[which(is.na(h$Monthly_CASH_ADVANCE))] <- 717.7235629

h$LIMIT_USAGE[which(is.na(h$LIMIT_USAGE))] <-0.3889264

h$MIN_PAYMENTS_RATIO[which(is.na(h$MIN_PAYMENTS_RATIO))]  <- 9.3500701
```

After imputing, we are checking for missing values

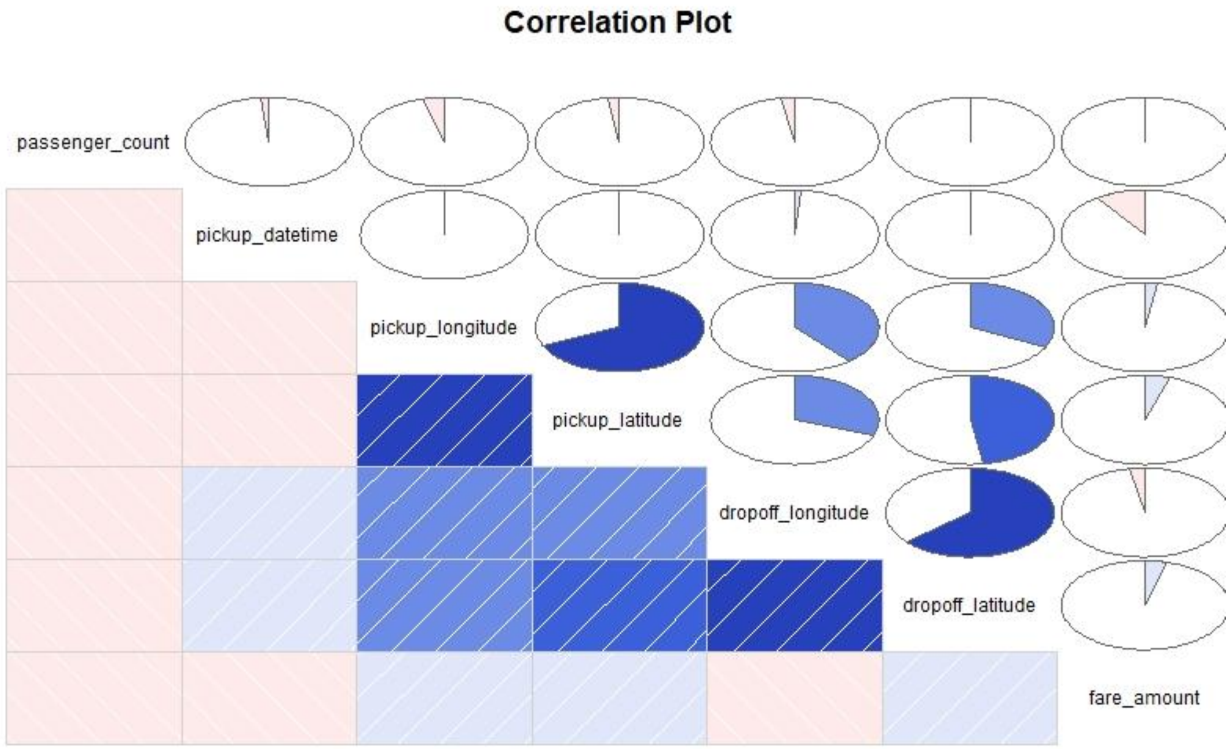


	apply.h..2..function.x...
BALANCE	0
BALANCE_FREQUENCY	0
PURCHASES	0
ONEOFF_PURCHASES	0
INSTALLMENTS_PURCHASES	0
CASH_ADVANCE	0
PURCHASES_FREQUENCY	0
ONEOFF_PURCHASES_FREQUENCY	0
PURCHASES_INSTALLMENTS_FREQUENCY	0
CASH_ADVANCE_FREQUENCY	0
CASH_ADVANCE_TRX	0
PURCHASES_TRX	0
CREDIT_LIMIT	0
PAYMENTS	0
MINIMUM_PAYMENTS	0
PRC_FULL_PAYMENT	0
TENURE	0
Monthly_Avg_PURCHASES	0
Monthly_CASH_ADVANCE	0
LIMIT_USAGE	0
MIN_PAYMENTS_RATIO	0

7. Feature selection

Here as we are dealing with numerical variables, we are finding corelation between variables using a heat map technique.Following command is used to find the corelation between variables.

```
ggcorr(h, label = T, label_size = 3,label_round = 2,hjust = 1,
size=3, color = "royalblue",layout.exp = 5,low =
"dodgerblue",mid = "gray95", high = "red2",
    name = "Correlation")
```



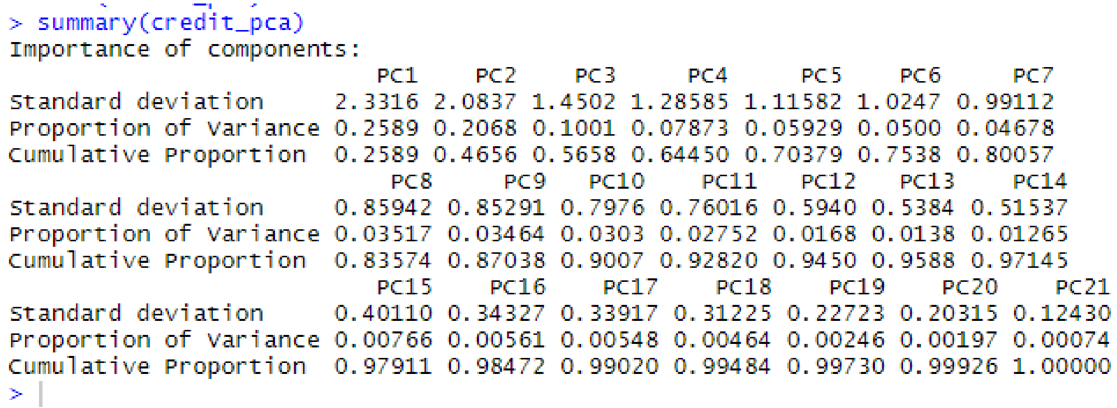
Applying PCA: (Feature selection)

Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation which converts a set of correlated variables to a set of uncorrelated variables. PCA is a most widely used tool in exploratory data analysis and in machine learning for predictive models. Moreover, PCA is an unsupervised statistical technique used to examine the interrelations among a set of variables. It is also known as a general factor analysis where regression determines a line of best fit.

Using following command to apply PCA technique:

```
scaled.credit = scale(h)

credit_pca = prcomp(scaled.credit)

credit_pca
```

looking for variance:

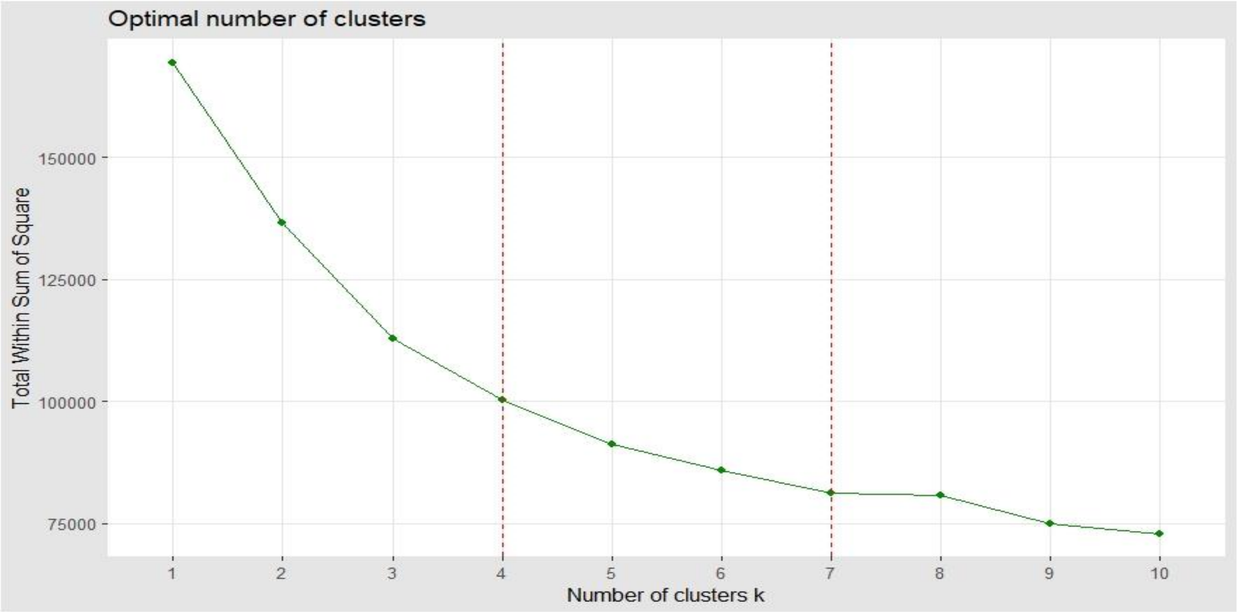


```
> summary(credit_pca)
Importance of components:
                          PC1    PC2    PC3     PC4     PC5    PC6     PC7
Standard deviation     2.3316 2.0837 1.4502 1.28585 1.11582 1.0247 0.99112
Proportion of Variance 0.2589 0.2068 0.1001 0.07873 0.05929 0.0500 0.04678
Cumulative Proportion  0.2589 0.4656 0.5658 0.64450 0.70379 0.7538 0.80057
                           PC8     PC9   PC10    PC11   PC12   PC13    PC14
Standard deviation     0.85942 0.85291 0.7976 0.76016 0.5940 0.5384 0.51537
Proportion of Variance 0.03517 0.03464 0.0303 0.02752 0.0168 0.0138 0.01265
Cumulative Proportion  0.83574 0.87038 0.9007 0.92820 0.9450 0.9588 0.97145
                          PC15    PC16    PC17    PC18    PC19    PC20    PC21
Standard deviation     0.40110 0.34327 0.33917 0.31225 0.22723 0.20315 0.12430
Proportion of Variance 0.00766 0.00561 0.00548 0.00464 0.00246 0.00197 0.00074
Cumulative Proportion  0.97911 0.98472 0.99020 0.99484 0.99730 0.99926 1.00000
> 
```

Based on interpretations from above, we will decide to take only 10 dimensions and put it to new dataset.

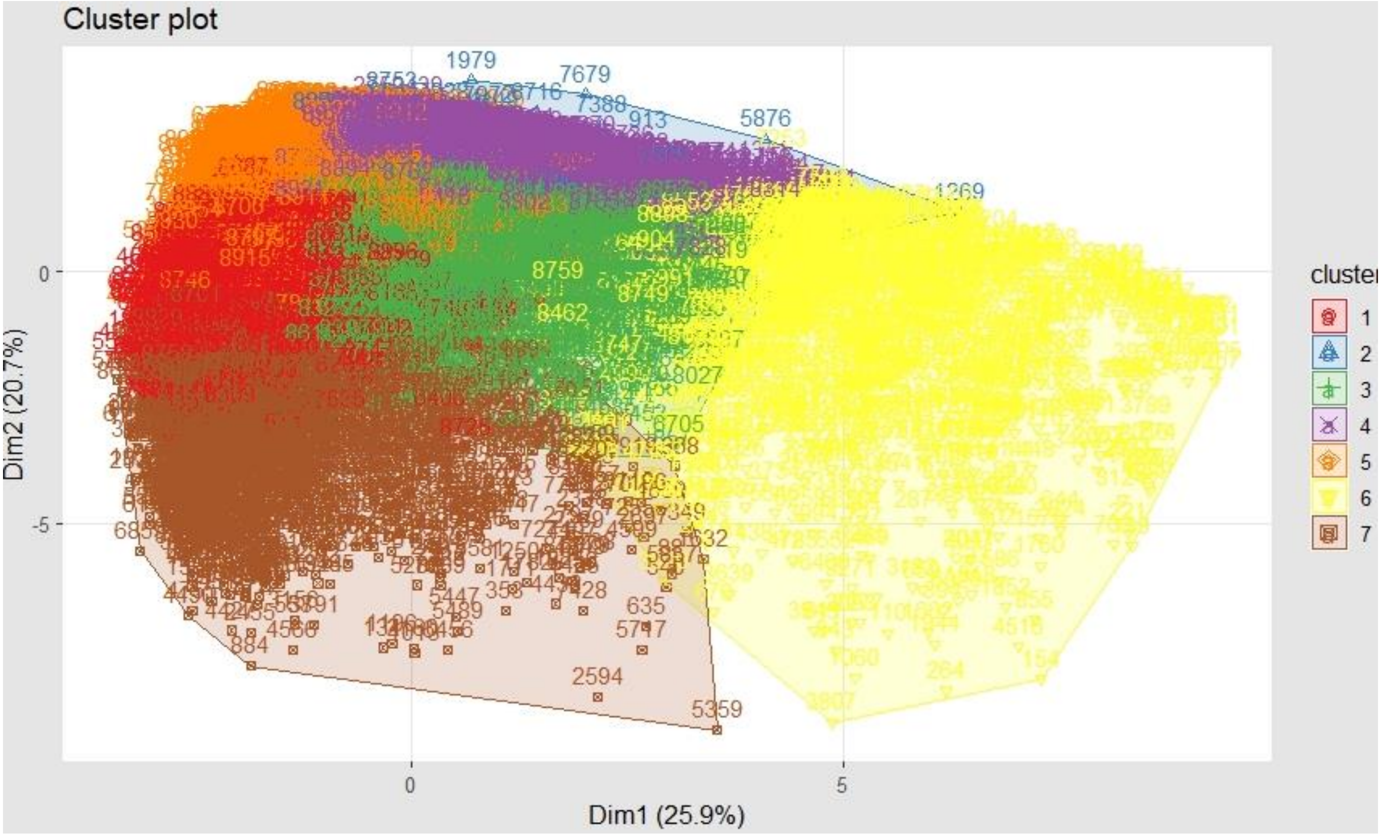
8. K means clustering:

After defining which dimensions that are going to used in Clustering, now we will use K-Means to determine how many Clusters do we need to divide Customers, which may represents their profile and hopefully we can determine what kind of tretment should be given to them.



Deciding whether, how many number of clusters are useful for the categorizing the customers,using following command:

```
for(i in 4:7){
 set.seed(289)
 model <- kmeans(credit_new, i)
 print(paste("WSS of K",i, "=",model$tot.withinss))
 print(paste("BSS Proportion of K",i, "=",
model$betweenss/model$totss))
 print(paste("Cluster Size of K",i, "="))
 print(paste(model$size))
 print(fviz_cluster(model, h, palette = "Set1") +
        theme_igray())
}
```



when we see the Cluster Plot, there is no significance different between K = 7 and the K = 6, therefore we will use K = 6 as the number of Clusters, since we don’t want too many Clusters and focusing our treatment to the Customers.

Number of clusters	For 4 clusters	For 5 clusters	For 6 clusters	For 7 clusters
WSS value	107125.996780914	95302.5447078897	85934.4485720659	80895.0217931879
Rsquare value	0.367106301040727	0.436958517513148	0.492304644445652	0.52207726314288

WSS value should be smaller , R square value should be greater.

9. Conclusion :

Category 1 (Red) with 2765 Customers: Customer on this Cluster tend to keep their Credit Card and avoid to use it for transactions (Low Purchases).

Cluster 2 (Blue) with 616 Customers: This Clusters filled with Customers that tend to have low Credit Limit and Payments.

Cluster 3 (Green) with 3472 Customers: Have a similar behaviour with Cluster 1 and 2, but they also have low Cash Advance. This might indicate they are non-potential customers.

Cluster 4 (Purple) with 734 Customers: Customers in this Cluster are potential, since they spend Purchases in medium amount

Cluster 5 (Orange) with Customers: This is our “Big Spenders”. They use their Credit Cards effectively with big Purchases.

Cluster 6 (Yellow) with 1019 Customers: This Clusters actually a quite potential Customers. They give a lot cash in advance, but unfortunately their spending on Purchases is still minimum.

10. Marketing Strategy:

- Category 1, we could give them some gimmicks as the first user, so they will be attracted to use their Credit Card more often.
- Category2, probably most of them are still on approval/review process.
- Category 3, A cash back or discount program may attract them to use their credit card in bigger Purchase.
- Category 4, Point reward/Loyalty Program may keep them on using the Credit Card.
- Category 5, We should attract them to apply instalment even with the 0% interest at the beginning.