# Problem 1 : Bank Customer Segmentation Case Study

## ****Problem Statement:****

A leading bank wants to develop a customer segmentation to give promotional offers to its customers. They collected a sample that summarizes the activities of users during the past few months. We have been given the task to identify the segments based on credit card usage.

## Data Description:

The data consists of 210 users, their activities during the past few months based on credit card usage. The data of 210 users can be found in the following csv file: bank\_marketing\_part1\_Data.csv

## Domain:

Credit Card (Sales & Promotion)

## Context:

A leading bank wants to develop a customer segmentation to give promotional offers to its customers. They collected a sample that summarizes the activities of users during the past few months. We have been given the task to identify the segments based on credit card usage.

## Attribute Information:

* spending: Amount spent by the customer per month (in 1000s)
* advance\_payments: Amount paid by the customer in advance by cash (in 100s)
* probability\_of\_full\_payment: Probability of payment done in full by the customer to the bank
* current\_balance: Balance amount left in the account to make purchases (in 1000s)
* credit\_limit: Limit of the amount in credit card (10000s)
* min\_payment\_amt : minimum paid by the customer while making payments for purchases made monthly (in 100s)
* max\_spent\_in\_single\_shopping: Maximum amount spent in one purchase (in 1000s)

## Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis).

### Basic EDA summary:-

* Data contains 210 observations and 7 attributes.
* All columns are numeric variables with decimal data type.
* There is no null data and duplicate data in any of these columns
* From box plot and data summary (shown below in Univariate analysis), it can be seen that there are few outliers in **min\_payment\_amt** and **probability\_of\_full\_payment**. Rest columns doesnt have outliers. does contains many outliers.

### Univariate Analysis

### Summary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **spending** | **advance\_payments** | **probability\_of\_full\_payment** | **current\_balance** | **credit\_limit** | **min\_payment\_amt** | **max\_spent\_in\_single\_shopping** |
| **count** | 210 | 210 | 210 | 210 | 210 | 210 | 210 |
| **mean** | 14.8475 | 14.5593 | 0.871 | 5.6285 | 3.2586 | 3.7002 | 5.4081 |
| **std** | 2.9097 | 1.306 | 0.0236 | 0.4431 | 0.3777 | 1.5036 | 0.4915 |
| **min** | 10.59 | 12.41 | 0.8081 | 4.899 | 2.63 | 0.7651 | 4.519 |
| **25%** | 12.27 | 13.45 | 0.8569 | 5.2622 | 2.944 | 2.5615 | 5.045 |
| **50%** | 14.355 | 14.32 | 0.8734 | 5.5235 | 3.237 | 3.599 | 5.223 |
| **75%** | 17.305 | 15.715 | 0.8878 | 5.9798 | 3.5618 | 4.7688 | 5.877 |
| **max** | 21.18 | 17.25 | 0.9183 | 6.675 | 4.033 | 8.456 | 6.55 |

From the summary, we can see that :-

* Bank is able to get 10% of the Average spending (in 1000s) of 210 customers as advance payment.
* Mean probability of getting full payment is 87% i.e. 13% customers tend to default or pay late
* Mean current balance for further shopping left for 210 customers is 5628.5 which is around 17% of the mean credit limit (32586). It means most of the credit limit is in use by the customers.
* Max spent in single shopping is very high with mean value as 5408 as compared to the mean of min payment amount (370) given in cash by customers to bank.

### Checking IQR, Coeffiecient of Variation, IQR, lower range and upper range of numerical cols with summary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | spending | advance\_  payments | probability\_  of\_full\_payment | current\_  balance | credit\_  limit | min\_  payment  \_amt | max\_spent  \_in\_single  \_shopping |
| count | 210 | 210 | 210 | 210 | 210 | 210 | 210 |
| mean | 14.8475 | 14.5593 | 0.871 | 5.6285 | 3.2586 | 3.7002 | 5.4081 |
| std | 2.9097 | 1.306 | 0.0236 | 0.4431 | 0.3777 | 1.5036 | 0.4915 |
| min | 10.59 | 12.41 | 0.8081 | 4.899 | 2.63 | 0.7651 | 4.519 |
| 25% | 12.27 | 13.45 | 0.8569 | 5.2622 | 2.944 | 2.5615 | 5.045 |
| 50% | 14.355 | 14.32 | 0.8734 | 5.5235 | 3.237 | 3.599 | 5.223 |
| 75% | 17.305 | 15.715 | 0.8878 | 5.9798 | 3.5618 | 4.7688 | 5.877 |
| max | 21.18 | 17.25 | 0.9183 | 6.675 | 4.033 | 8.456 | 6.55 |
| CV | 0.2 | 0.09 | 0.03 | 0.08 | 0.12 | 0.41 | 0.09 |
| Skew | 0.4 | 0.39 | -0.54 | 0.53 | 0.13 | 0.4 | 0.56 |
| IQR | 5.04 | 2.27 | 0.03 | 0.72 | 0.62 | 2.21 | 0.83 |
| UR | 24.86 | 19.11 | 0.93 | 7.06 | 4.49 | 8.08 | 7.12 |
| LR | 4.72 | 10.05 | 0.81 | 4.19 | 2.02 | -0.75 | 3.8 |

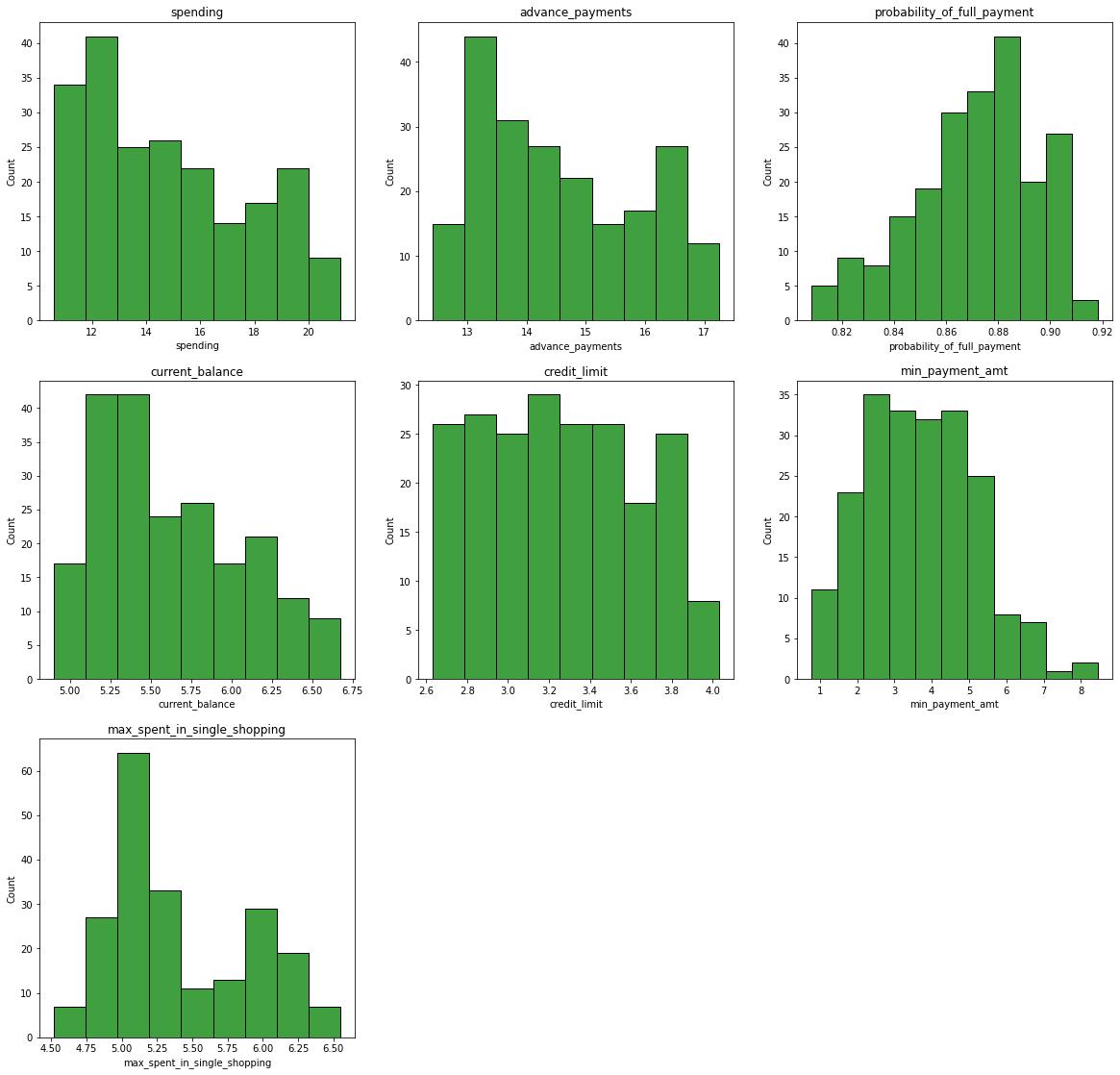
Few things which can be noticed here is :-

* Outlier in probability of full payment and min\_payment\_amt are just slightly beyond the UR/LR range. Hence we are skipping outlier treatment for them
* Only probability of full payment attribute is negative skewed. Rest all are positively skewed

### Histogram

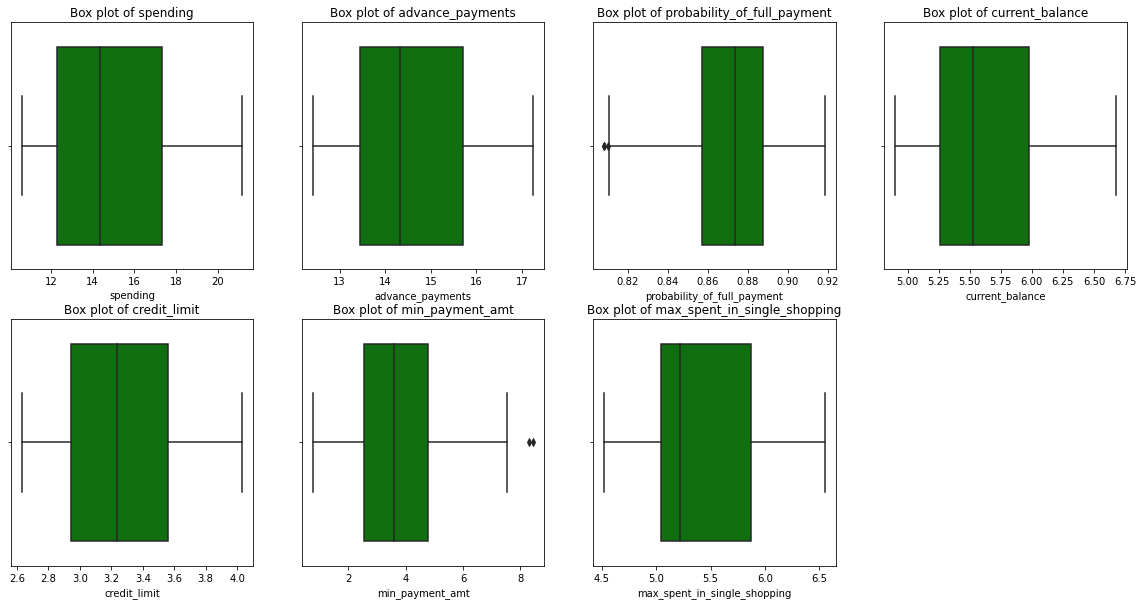
As seen in the histograms (shown below in the figure),

* For majority of the customers , max spend in single shopping lies in the range of 5K to 5.25K.
* In credit limit ranging from 260K to 390K, customer count is almost uniform 20-25. For credit limit exceeding 400K, customer counts are low. It shows that higher credit limit are given to lesser no of customers.



### Box Plots

As evident from the box plots (shown below) , there are few outliers in min\_payment\_amt and probability\_of\_full\_payment. Rest columns don’t have outliers.

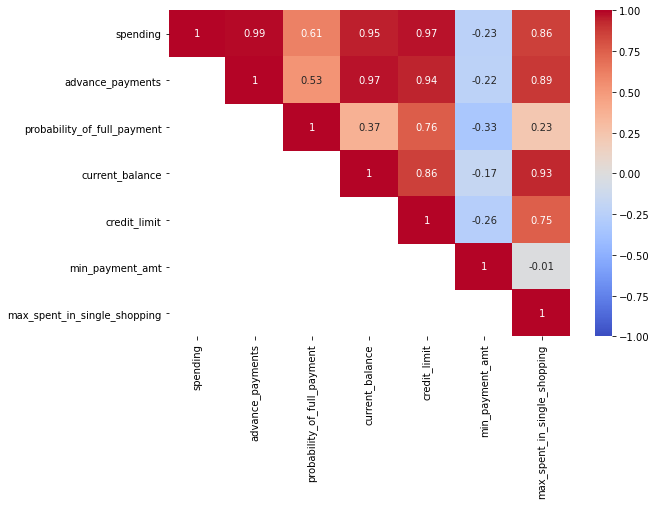
****

### Bivariate and Multivariate analysis

### Correlation matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **spending** | **advance\_payments** | **probability\_of\_full\_payment** | **current\_balance** | **credit\_limit** | **min\_payment\_amt** | **max\_spent\_in\_single\_shopping** |
| **spending** | 1 | 0.99 | 0.61 | 0.95 | 0.97 | -0.23 | 0.86 |
| **advance\_payments** | 0.99 | 1 | 0.53 | 0.97 | 0.94 | -0.22 | 0.89 |
| **probability\_of\_full\_payment** | 0.61 | 0.53 | 1 | 0.37 | 0.76 | -0.33 | 0.23 |
| **current\_balance** | 0.95 | 0.97 | 0.37 | 1 | 0.86 | -0.17 | 0.93 |
| **credit\_limit** | 0.97 | 0.94 | 0.76 | 0.86 | 1 | -0.26 | 0.75 |
| **min\_payment\_amt** | -0.23 | -0.22 | -0.33 | -0.17 | -0.26 | 1 | -0.01 |
| **max\_spent\_in\_single\_shopping** | 0.86 | 0.89 | 0.23 | 0.93 | 0.75 | -0.01 | 1 |

### Heat Map

****

We can see in heatmap and correlation matrix that

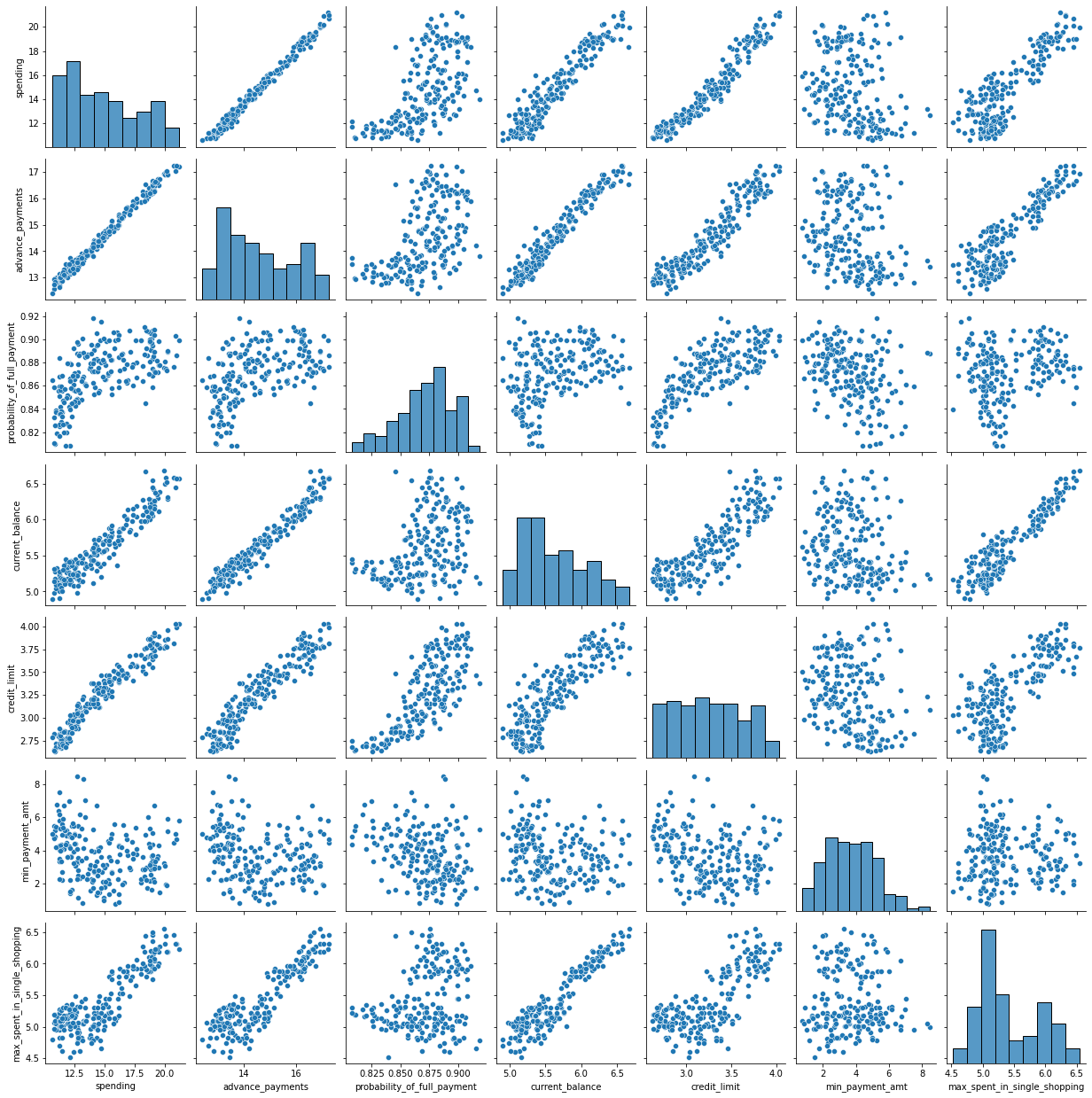
* Spending, advance\_payments, probability\_of\_full\_payment, current\_balance, credit\_limit, max\_spent\_in\_single\_shopping all these attributes has high positive correlation with each other
* min\_payment\_amt has negative correlation with all the other attributes. For max\_spent\_in\_single\_shopping it has no correlation.

Thus this dataset has multi colinearity problem

### Pairplots

To check the correlation of 2 columns in more detail we can draw **pairplots /Scatter**

**Diagram**



As depicted in the heatmap, we can clearly see the positive correlation between all these fields except min\_payment\_amt for which slight negative correlation is visible

## 1.2 Do you think scaling is necessary for clustering in this case? Justify

|  |  |  |  |
| --- | --- | --- | --- |
|  | advance\_payments | probability\_of\_full\_payment | current\_balance |
| count | 210 | 210 | 210 |
| mean | 14.5593 | 0.871 | 5.6285 |
| std | 1.306 | 0.0236 | 0.4431 |
| min | 12.41 | 0.8081 | 4.899 |
| 25% | 13.45 | 0.8569 | 5.2622 |
| 50% | 14.32 | 0.8734 | 5.5235 |
| 75% | 15.715 | 0.8878 | 5.9798 |
| max | 17.25 | 0.9183 | 6.675 |
| CV | 0.09 | 0.03 | 0.08 |
| Skew | 0.39 | -0.54 | 0.53 |
| IQR | 2.27 | 0.03 | 0.72 |
| UR | 19.11 | 0.93 | 7.06 |
| LR | 10.05 | 0.81 | 4.19 |

We can see above in the summary of only 3 columns that all three columns lies on different measurement scales. Mean values of attribute advance\_payments is high in magnitude as compared to other attributes. probability\_of\_full\_payment contains probability which is always between 0 & 1. This column value is lowest among all the column variables.

So if we don’t perform scaling here then distances between the observations calculated under clustering models will have more weightage of higher magnitude attributes.Importance of lower value attributes will be less. Standardization prevents variables with larger scales from dominating how clusters are defined.

Hence in order to give equal importance to all the attributes for distance calculation, scaling needs to be done on this data.

## 1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them

### Scaling

We have used Sklearn standardscalar to perform scaling (z score) on the data set.

Output of the scaled dataset is as shown below :-

array([[ 1.75435461, 1.81196782, 0.17822987, ..., 1.33857863,

-0.29880602, 2.3289982 ],

[ 0.39358228, 0.25383997, 1.501773 , ..., 0.85823561,

-0.24280501, -0.53858174],

[ 1.41330028, 1.42819249, 0.50487353, ..., 1.317348 ,

-0.22147129, 1.50910692],

...,

[-0.2816364 , -0.30647202, 0.36488339, ..., -0.15287318,

-1.3221578 , -0.83023461],

[ 0.43836719, 0.33827054, 1.23027698, ..., 0.60081421,

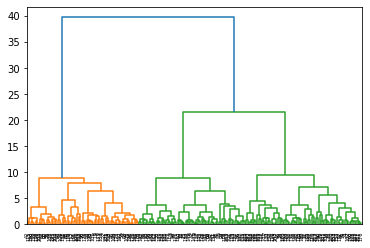
-0.95348449, 0.07123789],

[ 0.24889256, 0.45340314, -0.77624835, ..., -0.07325831,

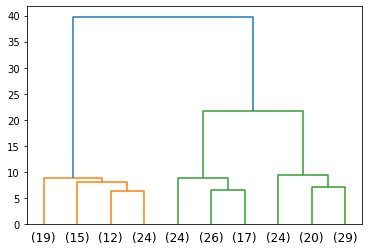
-0.70681338, 0.96047321]])

### Creating the Dendrogram

For creating the dendogram in hierarchical clustering, ward linkage method was chosen. Dendogram created is as shown below in image.



As we can see from the colour coding in dendogram, hierarchical clustering is giving us 2 final optimum clusters from this data. This dendogram doesnt looks clean and we are not able to read the observations count on the x axis. So we have plotted the cleaner and truncated version of this dendogram (shown below) for better understanding of clusters.



We can clearly see that orange cluster has (19+15+12+24) and green cluster has (26+17+25+20+19) observations. It’s clear from the dendogram that Green cluster contains more data. Dendogram is recommending 2 optimum clusters but we can go for 3 clusters if we cut dendogram horizontally at around 10 to 20 distance. 2 Clusters doesn’t seem to be a good choice from business point of view. Hence we are opting for 3 clusters. For cutting 3 clusters we need to cut the dendogram at a distance of 10 to 22.

### Cutting the Dendrogram with suitable clusters

### Label output from hierarchical clustering

In hierarchical clustering, we have 2 ways to obtain the desired clusters. On option is to use the fcluster module from scipy and another is to use sklearn AgglomerativeClustering. Further in fcluser module, labelling can be done via 2 ways using maxclust and using distance. Thus labels were created using scipy fcluster (maxclust & distance) methods and sklearn AgglomerativeClustering. Label outputs from each of these methods were compared and were found to be similar.

Both the methods were tried and output was similar in both these methods. and another option is to cut the dendogram using a horizontal line at a pre-defined distance. Number of interaction of this horizontal line with the dendogram’s vertical lines will give us the desired no of clusters.

#### Method 1 usi**ng fcluster maxclust:**

In this method, we provide the max clusters as input parameter to the function and in output we get the label output

Using this method our labels comes out as :

array([1, 3, 1, 2, 1, 2, 2, 3, 1, 2, 1, 3, 2, 1, 3, 2, 3, 2, 3, 2, 2, 2,

1, 2, 3, 1, 3, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 1, 1, 3, 1, 1,

2, 2, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 3, 2, 2, 3, 3, 1,

1, 3, 1, 2, 3, 2, 1, 1, 2, 1, 3, 2, 1, 3, 3, 3, 3, 1, 2, 3, 3, 1,

1, 2, 3, 1, 3, 2, 2, 1, 1, 1, 2, 1, 2, 1, 3, 1, 3, 1, 1, 2, 2, 1,

3, 3, 1, 2, 2, 1, 3, 3, 2, 1, 3, 2, 2, 2, 3, 3, 1, 2, 3, 3, 2, 3,

3, 1, 2, 1, 1, 2, 1, 3, 3, 3, 2, 2, 3, 2, 1, 2, 3, 2, 3, 2, 3, 3,

3, 3, 3, 2, 3, 1, 1, 2, 1, 1, 1, 2, 1, 3, 3, 3, 3, 2, 3, 1, 1, 1,

3, 3, 1, 2, 3, 3, 3, 3, 1, 1, 3, 3, 3, 2, 3, 3, 2, 1, 3, 1, 1, 2,

1, 2, 3, 1, 3, 2, 1, 3, 1, 3, 1, 3], dtype=int32)

#### Method 2 using **fcluster d**istance

In this method we pass the distance at which need to cut the dendogram using a horizontal line as a input and in output we get the labels.

Using this method our labels comes out as :

array([1, 3, 1, 2, 1, 2, 2, 3, 1, 2, 1, 3, 2, 1, 3, 2, 3, 2, 3, 2, 2, 2,

1, 2, 3, 1, 3, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 1, 1, 3, 1, 1,

2, 2, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 3, 2, 2, 3, 3, 1,

1, 3, 1, 2, 3, 2, 1, 1, 2, 1, 3, 2, 1, 3, 3, 3, 3, 1, 2, 3, 3, 1,

1, 2, 3, 1, 3, 2, 2, 1, 1, 1, 2, 1, 2, 1, 3, 1, 3, 1, 1, 2, 2, 1,

3, 3, 1, 2, 2, 1, 3, 3, 2, 1, 3, 2, 2, 2, 3, 3, 1, 2, 3, 3, 2, 3,

3, 1, 2, 1, 1, 2, 1, 3, 3, 3, 2, 2, 3, 2, 1, 2, 3, 2, 3, 2, 3, 3,

3, 3, 3, 2, 3, 1, 1, 2, 1, 1, 1, 2, 1, 3, 3, 3, 3, 2, 3, 1, 1, 1,

3, 3, 1, 2, 3, 3, 3, 3, 1, 1, 3, 3, 3, 2, 3, 3, 2, 1, 3, 1, 1, 2,

1, 2, 3, 1, 3, 2, 1, 3, 1, 3, 1, 3], dtype=int32)

#### Method 3 using Agglomerative Clustering

In this method we pass the no of clusters, distance method, linkage method as an input and in output we get the labels.

Using this method our labels comes out as :

[1 0 1 2 1 2 2 0 1 2 1 0 2 1 0 2 0 2 0 2 2 2 1 2 0 1 0 2 2 2 0 2 2 0 2 2 2

2 2 1 1 0 1 1 2 2 0 1 1 1 2 1 1 1 1 1 2 2 2 1 0 2 2 0 0 1 1 0 1 2 0 2 1 1

2 1 0 2 1 0 0 0 0 1 2 0 0 1 1 2 0 1 0 2 2 1 1 1 2 1 2 1 0 1 0 1 1 2 2 1 0

0 1 2 2 1 0 0 2 1 0 2 2 2 0 0 1 2 0 0 2 0 0 1 2 1 1 2 1 0 0 0 2 2 0 2 1 2

0 2 0 2 0 0 0 0 0 2 0 1 1 2 1 1 1 2 1 0 0 0 0 2 0 1 1 1 0 0 1 2 0 0 0 0 1

1 0 0 0 2 0 0 2 1 0 1 1 2 1 2 0 1 0 2 1 0 1 0 1 0]

In Agglomerative clustering methods labels starting from 0 are provided where as in fcluster labels starting from 1 are provided. Except this minor difference, output for all three is exactly the same.

### Cluster Profiles

After getting the label outputs, Mean of each column cluster wise was obtained (shown below) to profile them and get meaningful insights.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | spending | advance\_  payments | Probability  \_of  \_full  \_payment | Current  \_balance | Credit  \_limit | min\_  payment  \_amt | max\_  spent  \_in\_  Single  \_shopping | Clus\_kmeans | Freq |
| Agglo\_  CLusters |  |  |  |  |  |  |  |  |  |
| 0 | 14.199041 | 14.233562 | 0.87919 | 5.478233 | 3.226452 | 2.612181 | 5.086178 | 0.109589 | 73 |
| 1 | 18.371429 | 16.145429 | 0.8844 | 6.158171 | 3.684629 | 3.639157 | 6.017371 | 1.885714 | 70 |
| 2 | 11.872388 | 13.257015 | 0.848072 | 5.23894 | 2.848537 | 4.949433 | 5.122209 | 0.985075 | 67 |

We can have following profile :

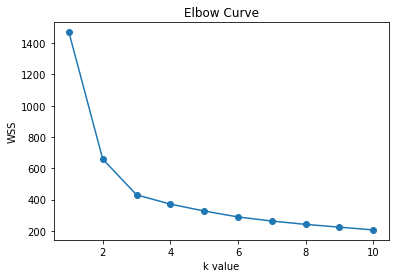
* Cluster 0 (Medium Spender with low advance payment) : These customers are the medium spending group and give low advance payments to the bank and have low current balance left for further credit card usage.
* Cluster 1 (High Spender with high advance payment) : These customers are spending highly on monthly basis and have high current balance left, higher credit limit and these customers give more advance payment to the banks. These are safe customers.
* Cluster 2 (Low Spender with low advance payment) : These customers are spending least on monthly basis and have high low balance and these customers give least advance payment to the banks. These customers have lowest credit limit.

## 1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Explain the results properly. Interpret and write inferences on the finalized clusters.

K-Means clustering is a non-hierarchical clustering approach where we specify the number of clusters needed as output lets say, k before running the model. We need to provide the no of clusters that we require as an input parameter to the function.

So we have taken the no of clusters (k) as an input and calculated the WSS (With cluster sum of square variance) for all k value and plotted the elbow curve (shown below)

### Calculating WSS for values of K from 1 to 10 - Elbow Method



We can see that WSS (with in cluster sum of square) decreases if we increase the value of k(clusters). At k=1, there is only 1 cluster so it will have maximum with in cluster sum of square of variance. For 2 clusters i.e. (k=2), WSS for cluster 1 and cluster 2 is computed and then added. This WSS value is going to significantly lesser than the WSS that we obtained from the previous cluster that is K=1. On graph we can notice that from 1 when we move to 2 the total WSS has significantly dropped (around 800) and on changing K from 2 to 3 WSS drop is nearly 230.

After 3, drop is less than 100 and it reduces further with the increasing value of K.

As drop is not that significant after k=3 and an elbow is forming at K value 2 to 3, we can opt for optimum value of K as 2 or 3. But since only 2 clusters will not provide much business sense, we are opting K value as 3 as the optimised no of clustering.

### Silhouette Score

Further silhouette score for K value (2 to 4) was calculated. If the silhouette score is close to +1 then we can say the clusters are well separated from each other on an average. If the silhouette score is close to 0, then we can say the clusters are not separated from each other.

If the silhouette score is close to -1 then we can say the model has done a blunder in terms of clustering the data

Scores are as shown below :-

#### for 2 clusters (k=2)

0.46577247686580914

#### for 3 clusters(k=3)

0.40072705527512986

#### for 4 clusters(k=4)

0.3276547677266192

we can see that Silhouette score is best for K value as 2 and for k value as 3, score is around 0.4 which is also good.

### The silhouette width

Sil-Width = (b - a) / max (a,b)

b = distance between observation and the neighbour cluster centroid (c2)

a = distance between observation and its own cluster centroid(c1)

if a > b, then it will give negative term. Sil-Width has min value of -1 (worst value) and max of + 1 (best value).

**So we have calculated the silhouette width for each observation and fount that for all observations silhouette width is positive i.e. distance between observation and its nearest cluster centroid is more than the distance between observation and its centroid. Hence clustering appears to be good enough on this data set.**

# 1.5 Describe cluster profiles for the clusters defined. Recommend different promotional strategies for different clusters.

### Cluster Profiles

After getting the label outputs, we have grouped the data based on clusters and Mean of each column along with its frequency was obtained (shown below) to profile them and get meaningful insights.

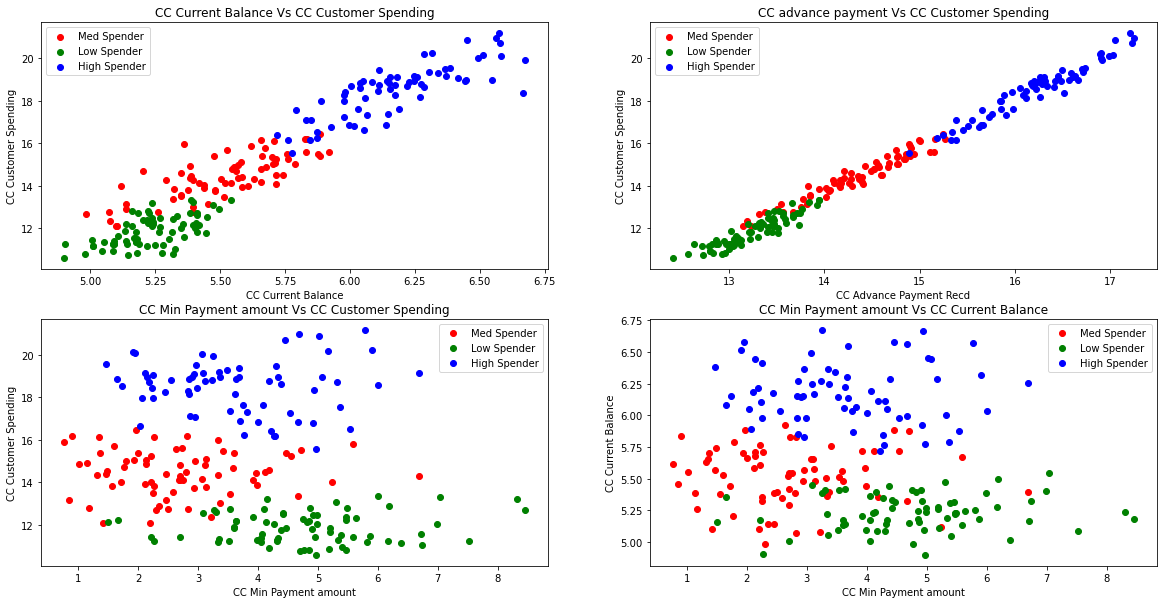
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **spending** | **advance\_payments** | **probability\_of\_full\_payment** | **current\_balance** | **credit\_limit** | **min\_payment\_amt** | **max\_spent\_in\_single\_shopping** | **freq** |
| **Clus\_kmeans** |  |  |  |  |  |  |  |  |
| **0** | 14.437887 | 14.337746 | 0.881597 | 5.514577 | 3.259225 | 2.707341 | 5.120803 | 71 |
| **1** | 11.856944 | 13.247778 | 0.848253 | 5.23175 | 2.849542 | 4.742389 | 5.101722 | 72 |
| **2** | 18.495373 | 16.203433 | 0.88421 | 6.175687 | 3.697537 | 3.632373 | 6.041701 | 67 |

Cluster 0 (Medium Spender with low advance payment) : These customers are the medium spending group and give low advance payments to the bank and have low current balance left for further credit card usage.

Cluster 1 (Low Spender with low advance payment) : These customers are spending least on monthly basis and have high low balance and these customers give least advance payment to the banks. These customers have lowest credit limit.

Cluster 2 (High Spender with high advance payment) : These customers are spending highly on monthly basis and have high current balance left, higher credit limit and these customers give more advance payment to the banks. These are safe customers.

### Visualising the groups



We can see that all three clusters are separated well. One thing to notice in min payment amount vs spending graph is that min. payment amount for some customers in low spending group is high. It is even exceeding the highest spending group. So it looks that they are spending less and are clearing that lesser amount in one go only.

### Promotional strategy Recommendations for different clusters

1. High Spenders group are spending the most among the given sample of customers and they also have high credit card balance. So these are the safe customers who are spending most with stable financial income.

* Bank can offer credit card upgrades offering most expensive credit cards loaded with tons on benefits to these customers so that these customers can get more benefits like earning/loyality points while spending money on credit card. It will provide more value to these customers and will help them in retaining them.

1. Medium Spender group are the customers which are spending slightly less than the high spender group and these customers have less credit card balance left for further spending. These may not be the most stable financially but these customers have a potential to spend more.

* These customers could be interested in getting bank loans and bank can offer Loans to these customers depending on their profile.
* These customers could be targeted with marketing offers in which if they are spending some more amount per month on their cards they could get some extra loyality points/gifts with Bank logo. It would help Bank in branding and increasing the credit card sales.

1. Low spender group are the customers who are spending less on credit cards.

* Banks could run survey on them to check if they are genuinely low spenders/low income customers or they are not getting the desired services from the bank which is forcing them to spend less on the current bank credit card.
* Bank could also check that if these customers belong to low income groups, then the credit limit of these card should not be very high and bank should take adequate precautions if these customers opt for the cards with higher credit limits.