Table of Contents

Topic **Page no.**

1. Read the data and do exploratory data analysis. Describe the data briefly

1.a. Understanding the case study:

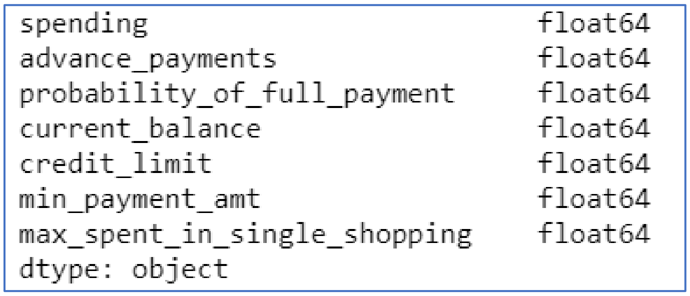
A leading bank wants to develop a customer segmentation to give promotional offers to its customers. They have collected a sample data that summarizes the activities of users during the past few months. We are given the task to identify the segments based on credit card usage.

Following columns are given in the data set and the meaning of each column is interpreted as follows:

- **spending:** This column represents the amount of money spent by the customer per month for the purchase (in 1000s)
- **advance_payments:** This basically tells about the amount of money paid by the customer in advance by cash (in 100s)
- **probability_of_full_payment:** This column refers to the probability of payment done in full by the customer to the bank. The range lies between 0 to 1.
- **current_balance:** This represents the total balance amount left in the account to make purchases for each customer (in 1000s)
- **credit_limit:** The ‘credit_limit’ column shows for each customer what is the limit of the amount in credit card (10000s)
- **min_payment_amt :** This column is about the minimum payment done by the customer while making payments for purchases made monthly (in 100s)
- **max_spent_in_single_shopping:** This tells us the maximum amount spent in one purchase by the customer or the ‘biggest buy’ of the customer (in 1000s)

1.b. Exploratory Data Analysis:

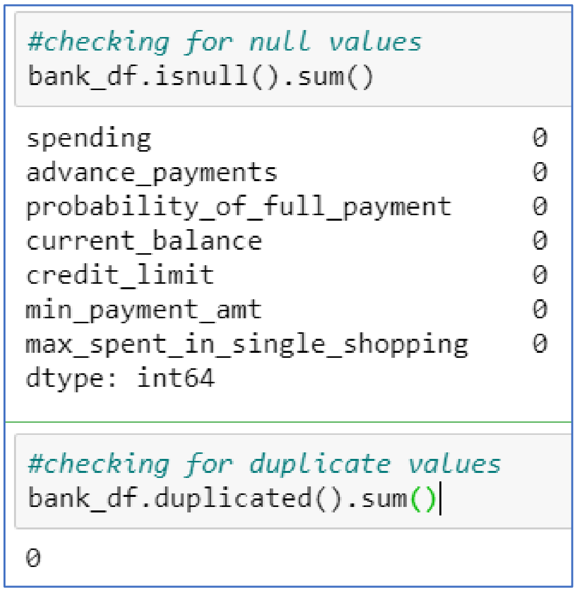
1.b.i. Checking the data type for each column in the data.



```
spending                        float64
advance_payments                float64
probability_of_full_payment     float64
current_balance                 float64
credit_limit                    float64
min_payment_amt                 float64
max_spent_in_single_shopping    float64
dtype: object
```

In Clustering it is important to have all the columns of numerical type. Since it’s a distance-based algorithm it cannot deal with categorical values. Here, we have checked that all variables are of float64 type we can go ahead with this data, and there is no need to drop any column.

1.b.ii. Checking the null and duplicated values in the data.



```
#checking for null values
bank_df.isnull().sum()
```

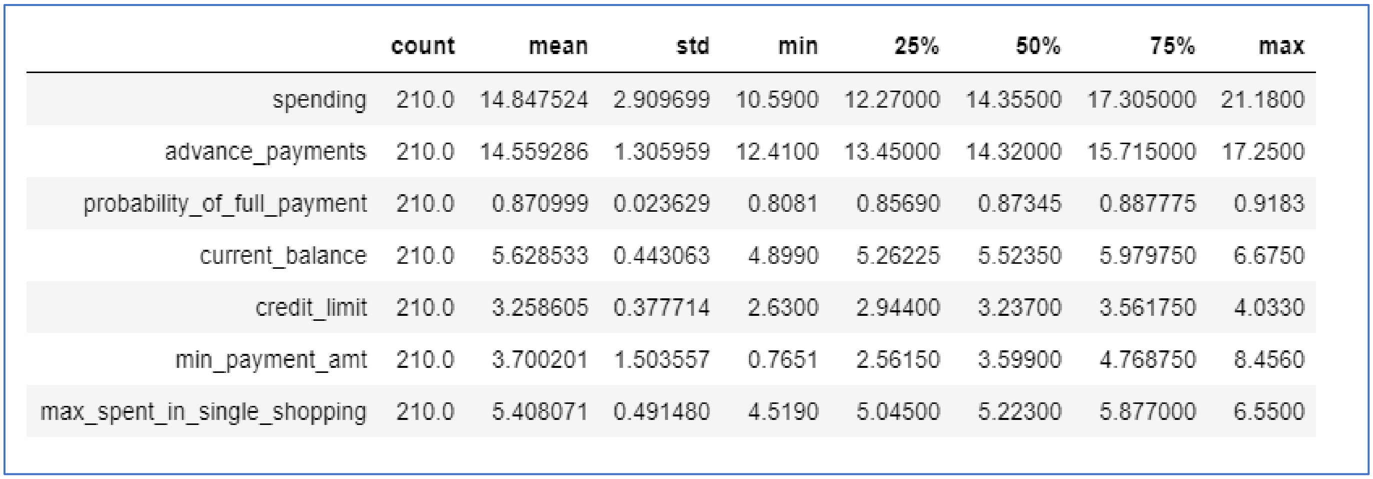
```
spending                        0
advance_payments                0
probability_of_full_payment     0
current_balance                 0
credit_limit                    0
min_payment_amt                 0
max_spent_in_single_shopping    0
dtype: int64
```

```
#checking for duplicate values
bank_df.duplicated().sum()
```

```
0
```

It is checked that there are no null and duplicated values in the data, hence no pre-processing regarding these are needed.

1.b.iii. Checking the descriptive statistics of the data.



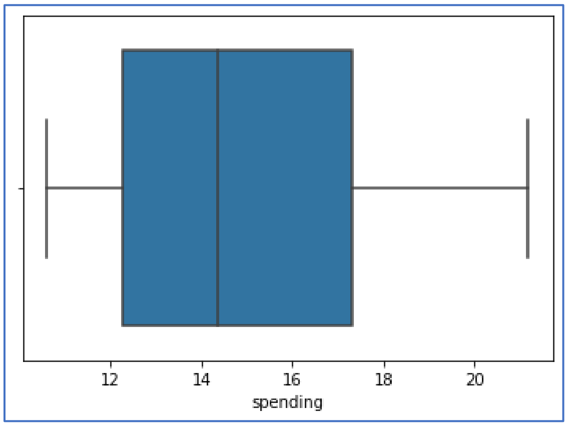
	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.847524	2.909699	10.5900	12.27000	14.35500	17.305000	21.1800
advance_payments	210.0	14.559286	1.305959	12.4100	13.45000	14.32000	15.715000	17.2500
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.85690	0.87345	0.887775	0.9183
current_balance	210.0	5.628533	0.443063	4.8990	5.26225	5.52350	5.979750	6.6750
credit_limit	210.0	3.258605	0.377714	2.6300	2.94400	3.23700	3.561750	4.0330
min_payment_amt	210.0	3.700201	1.503557	0.7651	2.56150	3.59900	4.768750	8.4560
max_spent_in_single_shopping	210.0	5.408071	0.491480	4.5190	5.04500	5.22300	5.877000	6.5500

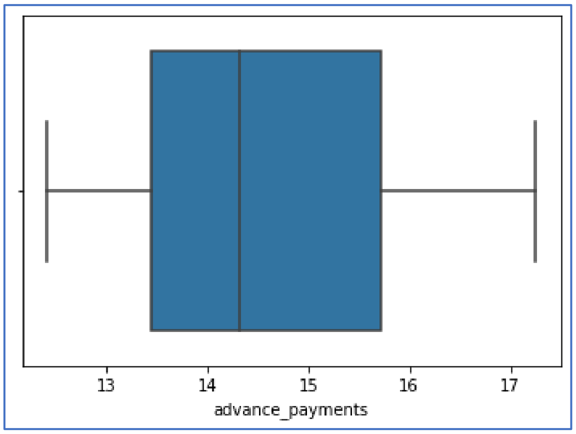
The descriptive stats above shows that there are 210 records with no missing values.

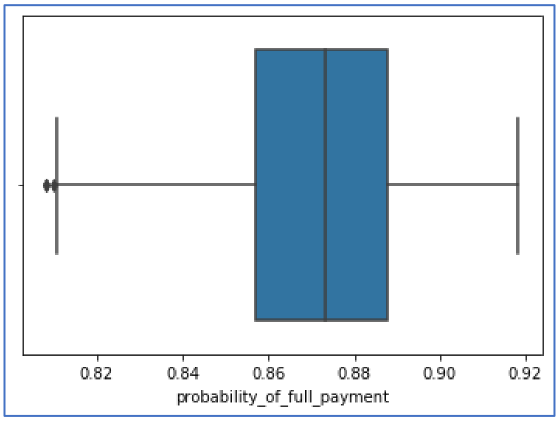
Though in clustering, it is not important to take in account the normalization of data, by comparing the mean and median values for each variable we can say that the data is quite normalized and there is no problem of skewness.

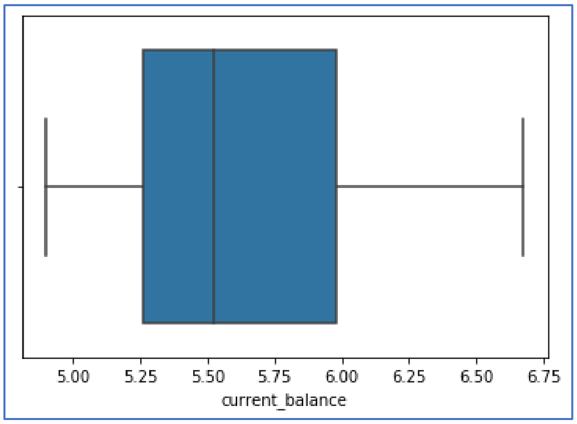
Also, we can see that the range of each variable is different in the data set.

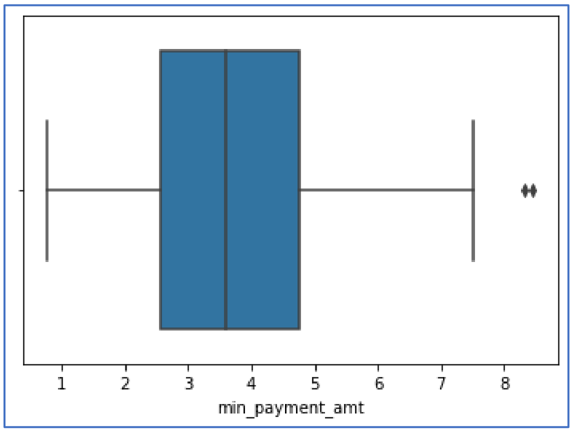
1.b.iv. Checking the outliers in the data.

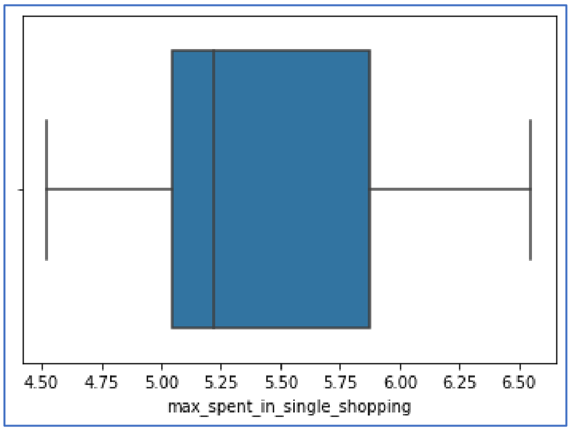












We can see that in 2 variables, ‘probability_of_full_payment’ and ‘min_payment_amt’ has few outliers which we can remove before using the data set for clustering. Since the number of outliers are not very large, we would proceed with the same data.

2. Do you think scaling is necessary for clustering in this case? Justify

Yes, in Clustering scaling is always recommended since it's a distance-based algorithm and both "Hierarchical" and "K-means" clustering follows Euclidean distance as the basis of the distance calculation.

Euclidean distance between data two points p and q is calculated as below:

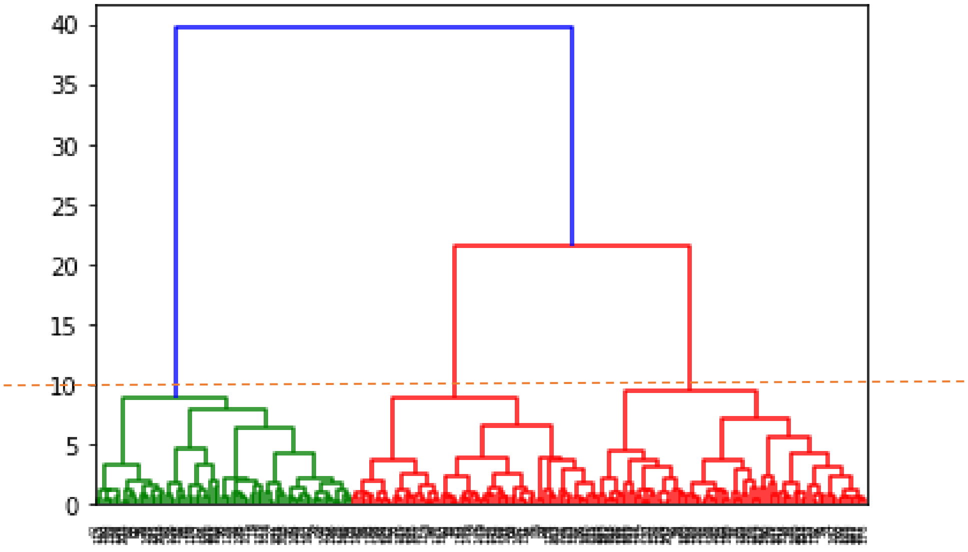
$$\begin{aligned} d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) &= \sqrt{(q_1-p_1)^2+(q_2-p_2)^2+\cdots+(q_n-p_n)^2} \\ &= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}. \end{aligned}$$

So, it becomes important for all the variables to be on the same scale and be given equal importance while contributing to the distance calculation. Scaling can also manage the outliers to some extent.

Moreover, we can see in 1.a that the data is spread over multiple scales for each column, in this case scaling becomes necessary.

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

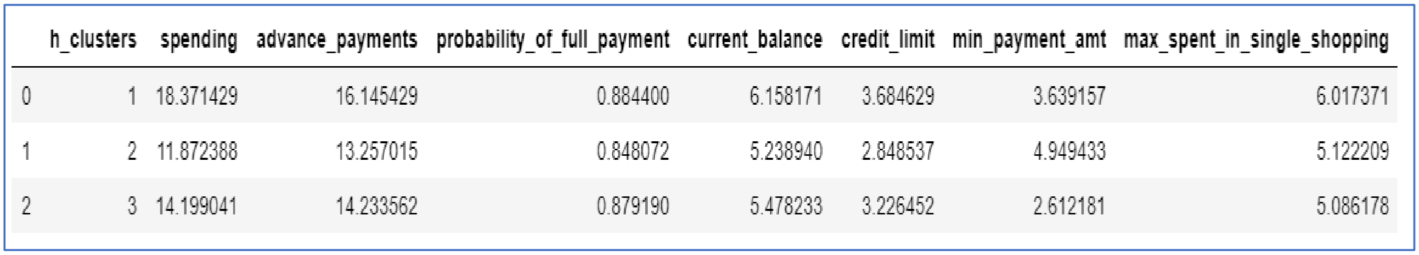
3.a. Dendrogram:



We have created the above Dendrogram using ‘Ward’s method’ as the method of linkage on the scaled data set. The Ward’s method computes the variance with clusters by combining 2 clusters and selects the one with the least variance to link.

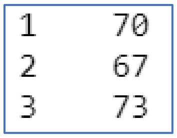
In the Dendrogram, we try to find the optimum number of clusters and we can see, if we cut the line at the height of 10, we get 3 distinct numbers of clusters, one green and other 2 red ones and it also seems to be the optimum number of clusters.

3.b. Cluster Description:



	h_clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371
1	2	11.872388	13.257015	0.848072	5.238940	2.848537	4.949433	5.122209
2	3	14.199041	14.233562	0.879190	5.478233	3.226452	2.612181	5.086178

So, we finally have 3 clusters with their mean values as shown in the above table and their respective frequencies are:



```
1    70
2    67
3    73
```

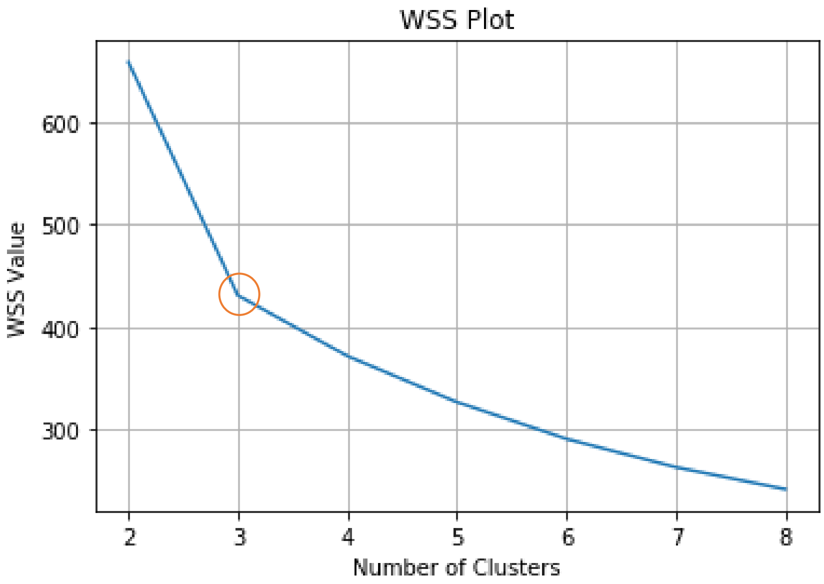
Cluster 1: The first cluster is the group of highest spenders we can say. This group has the highest mean spending, they have also made the most advance payments with largest probability of full payment. They also hold the highest current balance and credit limit of all 3 groups. They are **high-budget spenders**.

Cluster 2: This cluster is the group of customers which spend lowest of all 3 groups, making fewer advance payments, though the probability of full payment is quite good but lowest of 3 groups. They hold least current balance and the credit limit. They are **limited-budget spenders**.

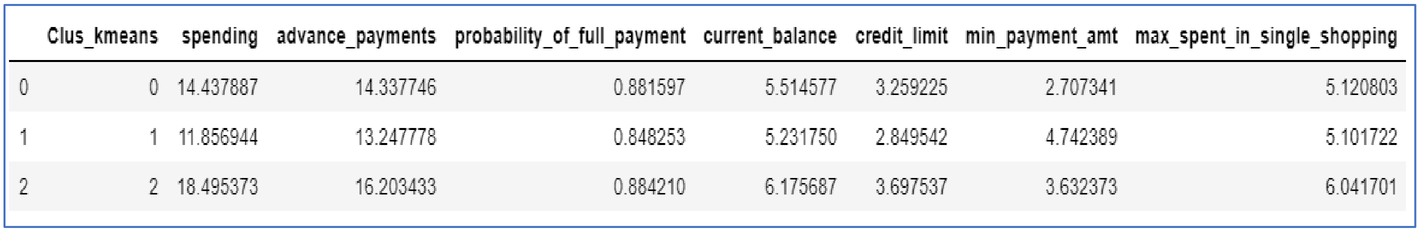
Cluster 3: This group of people lie between the above 2 groups where, customers spend good and make good amount of advance payments too. Their probability of full payments also lies between that of above 2 groups. They hold a decent amount of current balance, but they make lowest of minimum payment amount and also don’t spend much in single shopping. They are **careful spenders**.

4. Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score.

4.a. Elbow curve:

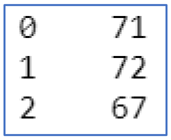


In K- means Clustering, the elbow method is used to determine the optimum number of clusters from the range of clusters in consideration. Here, we must select the value of k at the “elbow” i.e., the point after which the within sum of squares/inertia/distortion start decreasing in a linear pattern. We see that the maximum drop in wss values is between 2 and 3 clusters. Thus, for the given data, we choose that the optimal number of clusters for the data is 3.



	Clus_kmeans	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	0	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803
1	1	11.856944	13.247778	0.848253	5.231750	2.849542	4.742389	5.101722
2	2	18.495373	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701

So, we finally have 3 clusters with their mean values as shown in the above table and their respective frequencies are:



```
0    71
1    72
2    67
```

4.b. Silhouette score

In K- means Clustering, Silhouette score can be used as an indirect model evaluation parameter. In this method, we calculate silhouette width for each data point, using the following formula,

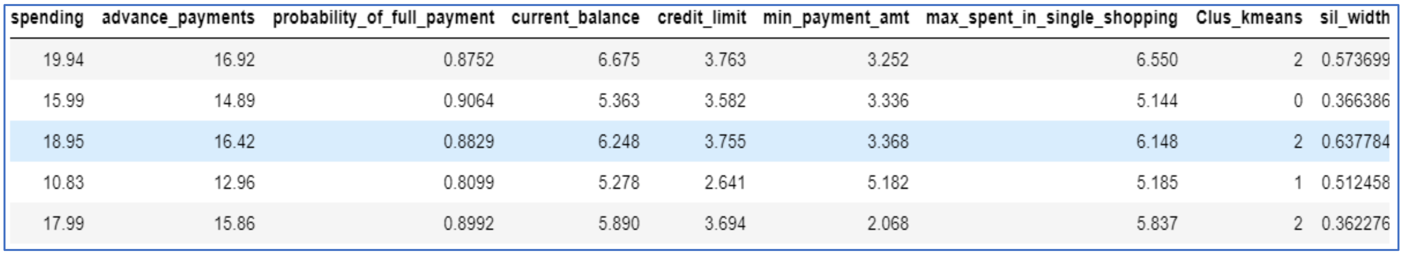
$$\textit{Silhouette width} = (x-y)/\max(x,y)$$

where, x= distance of observed data point from its cluster centroid

y= distance of observed data point from centroid of nearest cluster

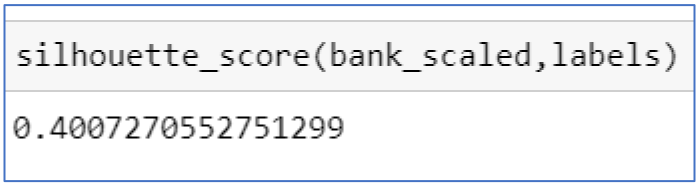
The coefficient varies between -1 and 1. A value close to 1 implies that the instance is close to its cluster is a part of the right cluster. Whereas, a value close to -1 means that the value is assigned to the wrong cluster.

In table shown below, we have calculated silhouette width for each individual data point as 'sil_width'.



spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Clus_kmeans	sil_width
19.94	16.92	0.8752	6.675	3.763	3.252	6.550	2	0.573699
15.99	14.89	0.9064	5.363	3.582	3.336	5.144	0	0.366386
18.95	16.42	0.8829	6.248	3.755	3.368	6.148	2	0.637784
10.83	12.96	0.8099	5.278	2.641	5.182	5.185	1	0.512458
17.99	15.86	0.8992	5.890	3.694	2.068	5.837	2	0.362276

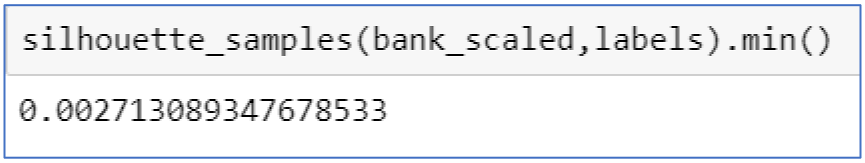
Silhouette score for this model is 0.400 which is calculated as the mean value of silhouette width of all data points as shown below:



```
silhouette_score(bank_scaled,labels)
```

```
0.4007270552751299
```

After taking the minimum of all 'sil_width', we can observe that no silhouette width is calculated as negative, i.e., no data point is clustered incorrectly.

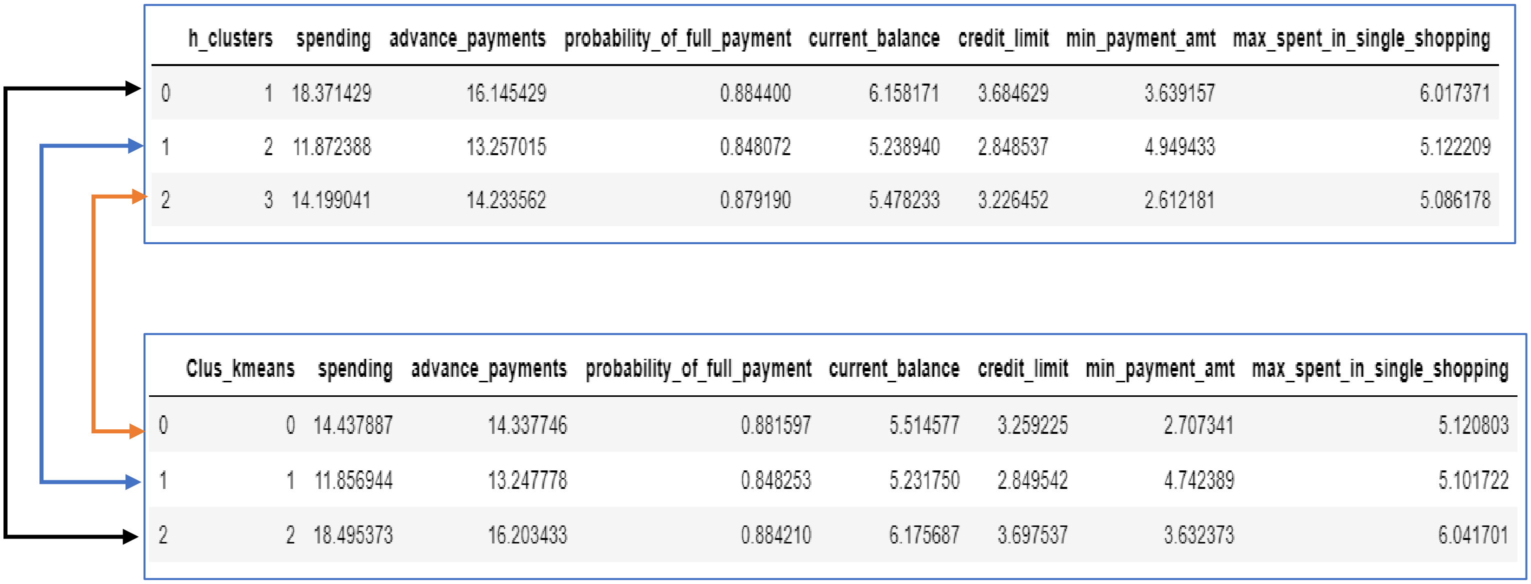


```
silhouette_samples(bank_scaled,labels).min()
```

```
0.002713089347678533
```

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

Before profiling the clusters, we can try to map the clusters formed in Hierarchical and K-means clustering.



	h_clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371
1	2	11.872388	13.257015	0.848072	5.238940	2.848537	4.949433	5.122209
2	3	14.199041	14.233562	0.879190	5.478233	3.226452	2.612181	5.086178

	Clus_kmeans	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	0	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803
1	1	11.856944	13.247778	0.848253	5.231750	2.849542	4.742389	5.101722
2	2	18.495373	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701

From the above two tables we observe that:

- **cluster 1** of Hierarchical clustering is similar to **cluster 2** of K-means clustering, let's call it **Cluster A**.
- **cluster 2** of Hierarchical clustering is similar to **cluster 1** of K-means clustering, let's call it **Cluster B**.
- **cluster 3** of Hierarchical clustering is similar to **cluster 0** of K-means clustering, let's call it **Cluster C**.

These Clusters A, B and C formed from two different approaches differ a bit in frequencies. For example. Cluster A has 70 records in Hierarchical Clustering, while it has 67 records in K-means clustering. This happens because some records at the boundaries of clusters are not clustered into the same group in both methods of clustering.

5.a. Cluster Profiling:

Cluster A: As described in section 3.b, this group can be profiled as **high-budget spenders**. The first cluster is the group of highest spenders we can say. This group has the highest mean spending, they have also made the most advance payments with largest probability of full payment. They also hold the highest current balance and credit limit of all 3 groups.

Cluster B: As described in section 3.b, this group can be profiled as **limited-budget spenders**. This cluster is the group of customers which spend lowest of all 3 groups, making fewer advance payments, though the probability of full payment is quite good but lowest of 3 groups. They hold least current balance and the credit limit.

Cluster C: As described in section 3.b, this group can be profiled as **careful spenders**. This group of people lie between the above 2 groups where, customers spend good and make good amount of advance payments too. Their probability of full payments also lies between that of above 2 groups. They hold a decent amount of current balance, but they make lowest of minimum payment amount and don’t spend much in single shopping.

5.b. Promotional Strategies:

For **Cluster A** i.e., for **high-budget spenders**, they could be the people with good income and high maintenance so we can offer them discount on luxury items like Travel, Dining and Hotel packages.

For **Cluster B** i.e., for **limited-budget spenders**, they could be the people with low income or students with pocket money so we can offer them discount on Online Shopping, Movies and Cashback offers can be also recommended for them.

For **Cluster C** i.e., for **careful spenders**, they could be the people with decent income but maybe they are selective shoppers so we can offer them discount on essentials like Fuel or Groceries or Online food so they are encouraged to use it more.