

---

# AllLife Bank Customer Segmentation

## Machine Learning - 1

---

## Contents

1.0 Problem Statement and Executive Summary.....	3
1.1 Exploratory Data Analysis.....	4
1.1.1 Univariate Analysis .....	6
1.1.2 CDF Plot of Numerical Variable.....	11
1.1.3 Bivariate Analysis .....	12
1.2 Data Pre Processing.....	15
1.3 Applying K Means Clustering.....	17
1.4 Applying Hierarchical Clustering.....	23
1.5 K-means vs Hierarchical Clustering.....	29
1.6 Actionable Insights & Recommendations.....	30
1.7 PCA Transformation.....	32
1.8 Interpretation of Principal Components.....	33
1.9 Variance Explanation.....	34
1.10 Visualization.....	36
1.11 Dimensionality Reduction Impact.....	40

# Problem Statement

AllLife Bank wants to focus on its credit card customer base in the next financial year. They have been advised by their marketing research team, that the penetration in the market can be improved. Based on this input, the Marketing team proposes to run personalized campaigns to target new customers as well as upsell to existing customers. Another insight from the market research was that the customers perceive the support services of the bank poorly. Based on this, the Operations team wants to upgrade the service delivery model, to ensure that customer queries are resolved faster. Head of Marketing and Head of Delivery both decide to reach out to the Data Science team for help.

## Executive Summary

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. You are given the task to identify the segments based on credit card usage.

## Introduction

The purpose is to explore the data set and find the spending areas of the customers as accordance to their credit profile, so promotional offers can be provided based on their transaction history.

## Data Description

The data provided is of various customers of a bank and their financial attributes like credit limit, the total number of credit cards the customer has, and different channels through which customers have contacted the bank for any queries (including visiting the bank, online and through a call center).

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

## Data Dictionary


SI\_No: Primary key of the records  
Customer Key: Customer identification number  
Average Credit Limit: Average credit limit of each customer for all credit cards  
Total credit cards: Total number of credit cards possessed by the customer  
Total visits bank: Total number of visits that customer made (yearly) personally to the bank  
Total visits online: Total number of visits or online logins made by the customer (yearly)  
Total calls made: Total number of calls made by the customer to the bank or its customer service department (yearly)

## 1.0 EDA- Exploratory Data Analysis



- Checking the shape of the dataset

There are 660 rows and 7 columns.

- Displaying few rows of the dataset



	Sl_No	Customer Key	Avg_Credit_Limit	Total_Credit_Cards	Total_visits_bank	Total_visits_online	Total_calls_made
0	1	87073	100000	2	1	1	0
1	2	38414	50000	3	0	10	9
2	3	17341	50000	7	1	3	4
3	4	40496	30000	5	1	1	4
4	5	47437	100000	6	0	12	3



- Checking the data types of the columns for the dataset

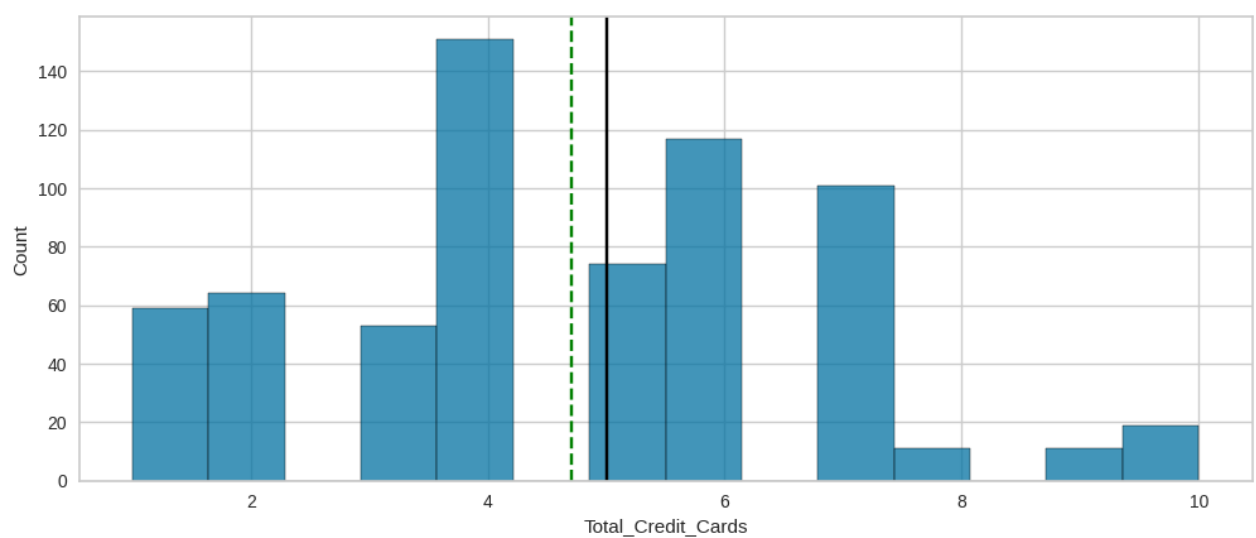
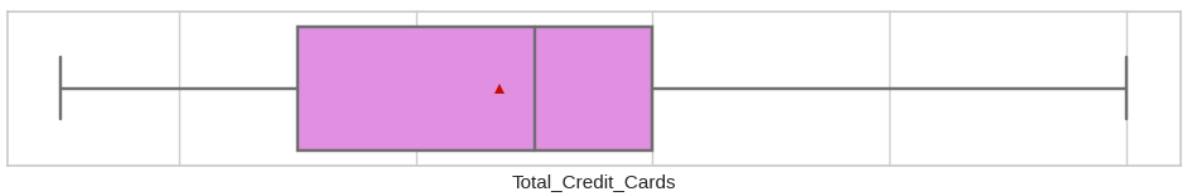
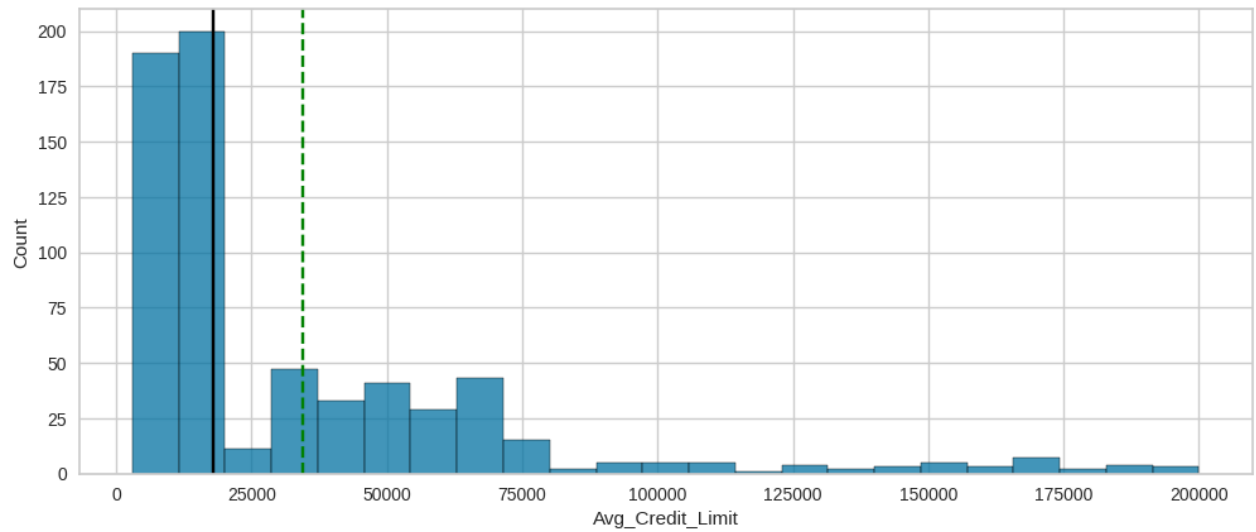
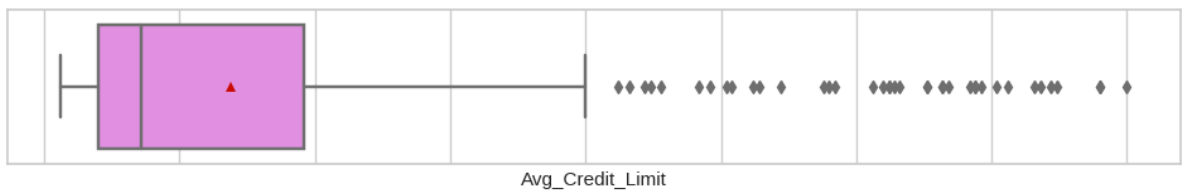
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 660 entries, 0 to 659
Data columns (total 7 columns):
#   Column                      Non-Null Count  Dtype
---  -
0   Sl_No                       660 non-null   int64
1   Customer_Key                660 non-null   int64
2   Avg_Credit_Limit            660 non-null   int64
3   Total_Credit_Cards          660 non-null   int64
4   Total_visits_bank           660 non-null   int64
5   Total_visits_online          660 non-null   int64
6   Total_calls_made             660 non-null   int64
dtypes: int64(7)
memory usage: 36.2 KB
```

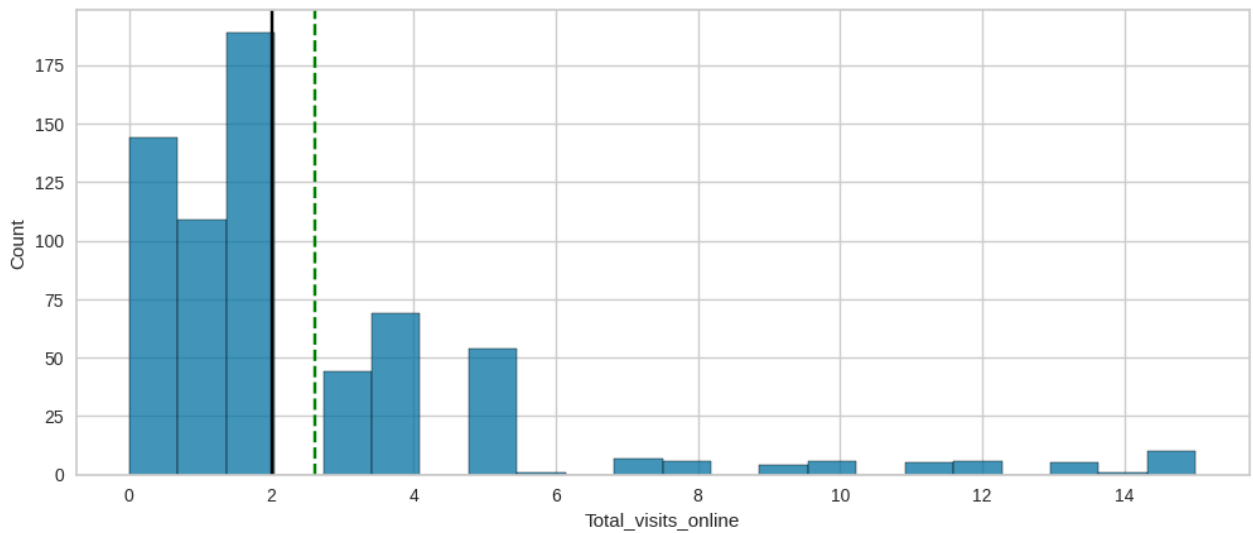
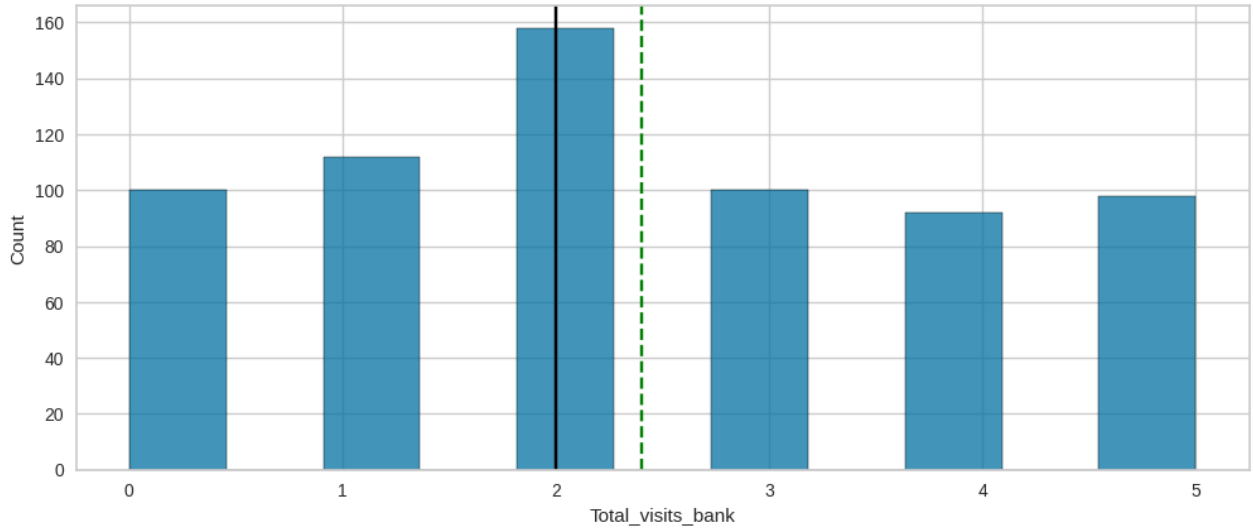
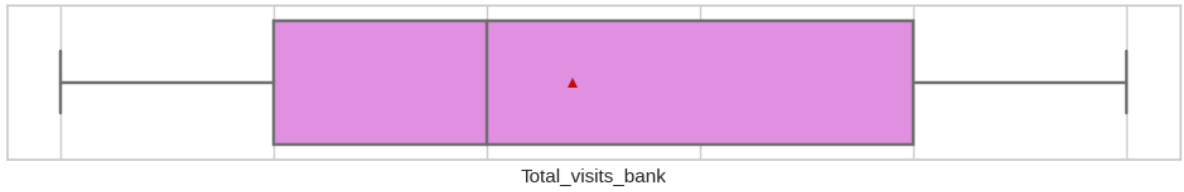
➤ Statistical summary of the dataset

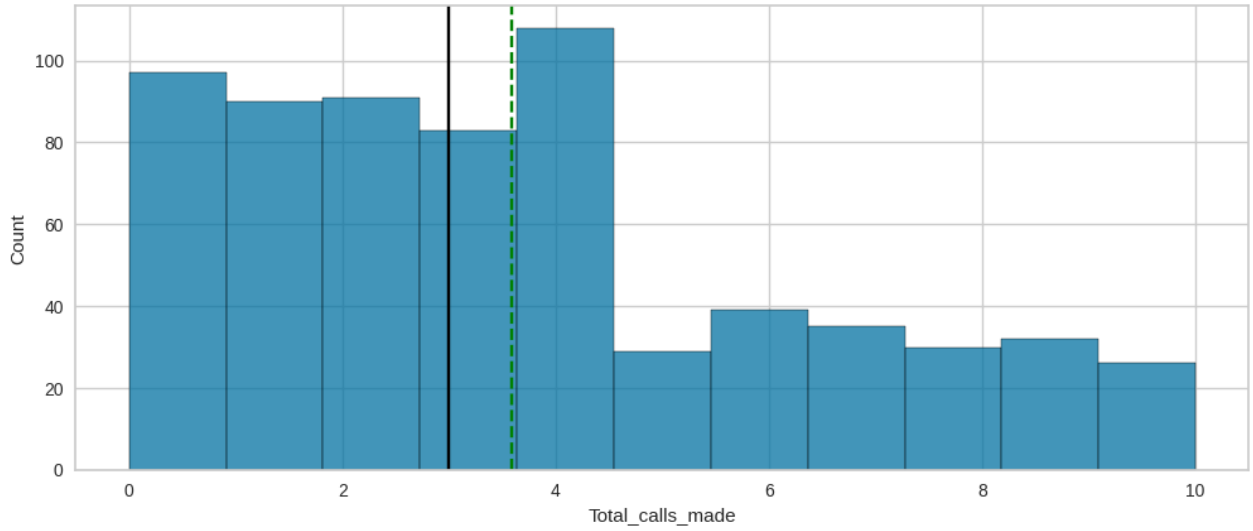
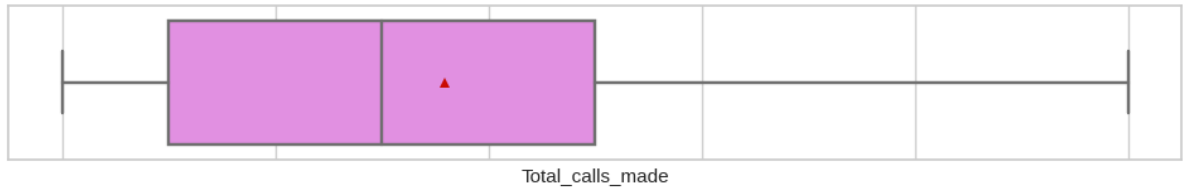
	count	mean	std	min	25%	50%	75%	max
<b>Avg_Credit_Limit</b>	660.0	34574.242424	37625.487804	3000.0	10000.0	18000.0	48000.0	200000.0
<b>Total_Credit_Cards</b>	660.0	4.706061	2.167835	1.0	3.0	5.0	6.0	10.0
<b>Total_visits_bank</b>	660.0	2.403030	1.631813	0.0	1.0	2.0	4.0	5.0
<b>Total_visits_online</b>	660.0	2.606061	2.935724	0.0	1.0	2.0	4.0	15.0
<b>Total_calls_made</b>	660.0	3.583333	2.865317	0.0	1.0	3.0	5.0	10.0



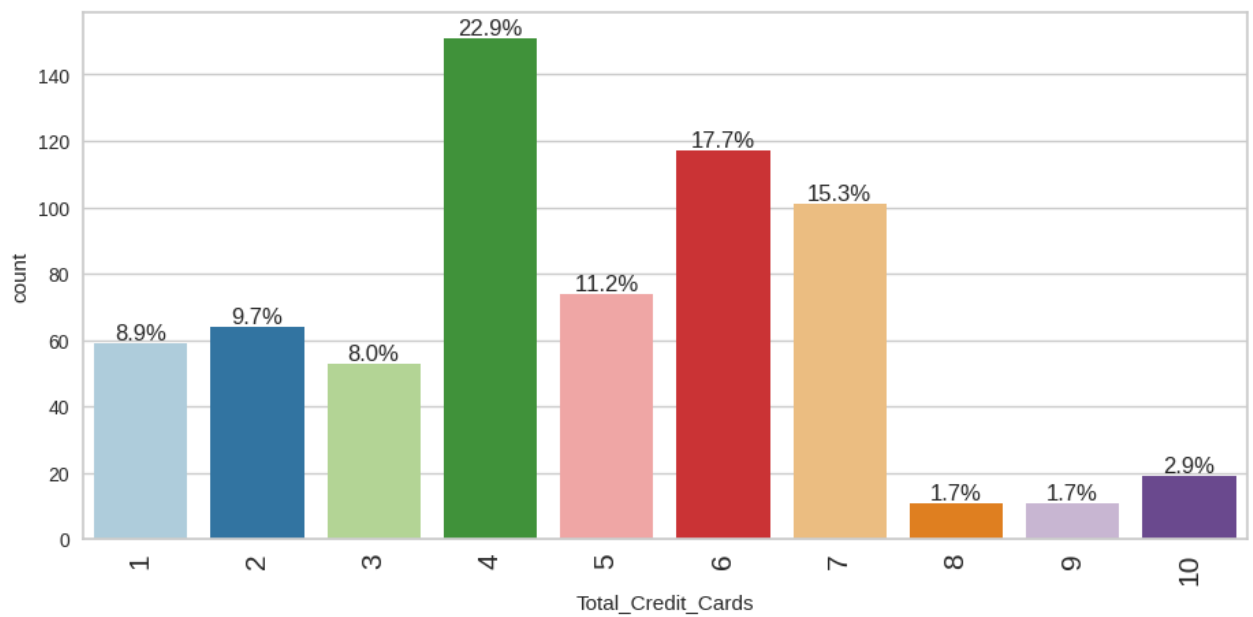
### 1.1.1 Univariate Analysis



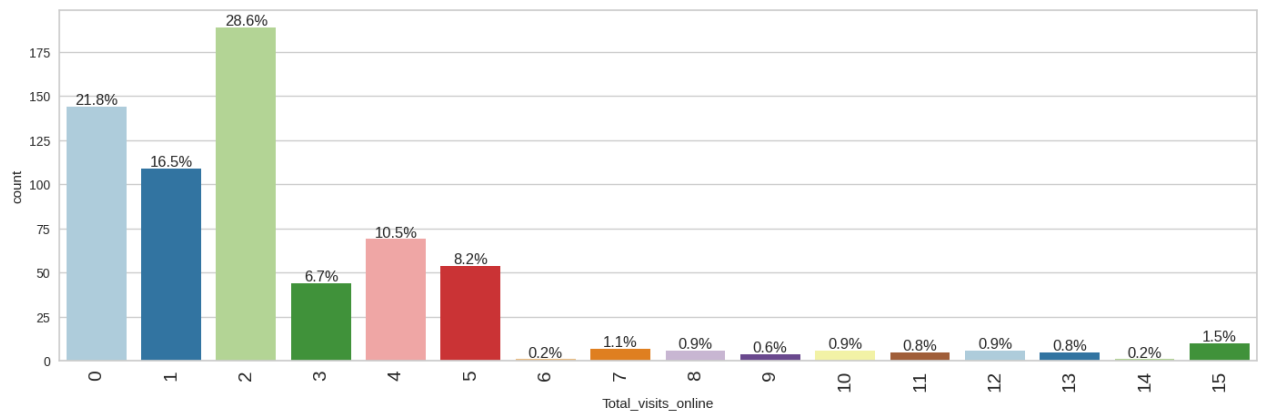
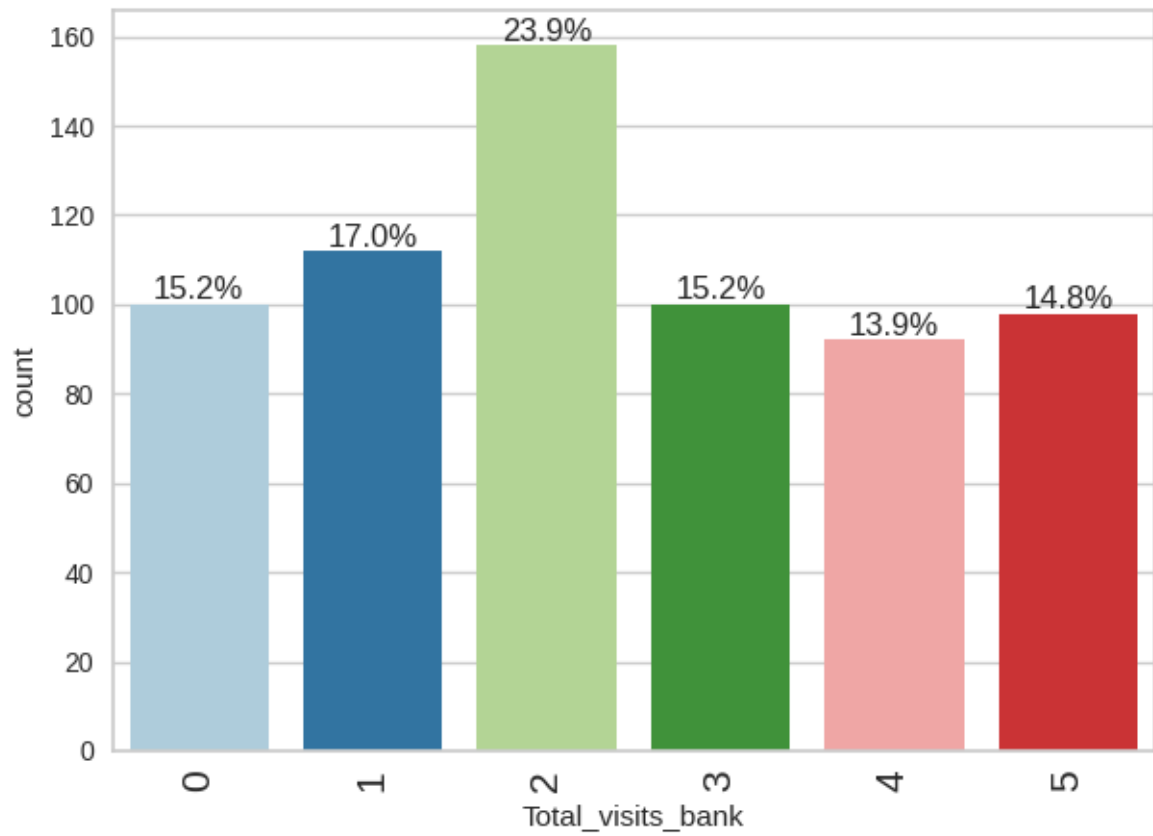


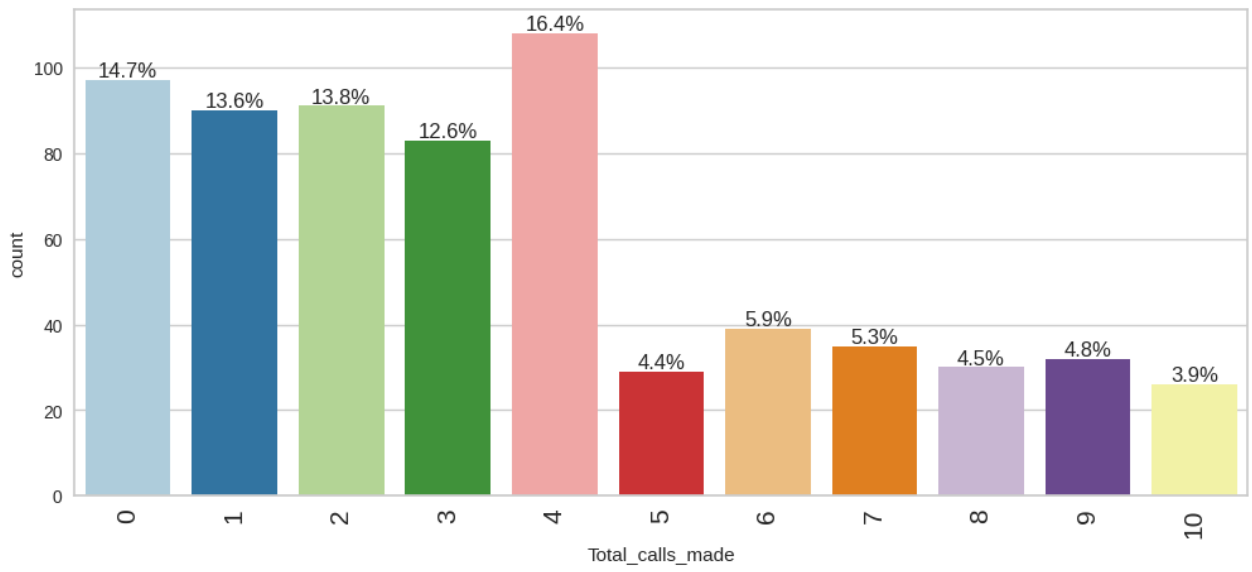


➤ BAR Plots



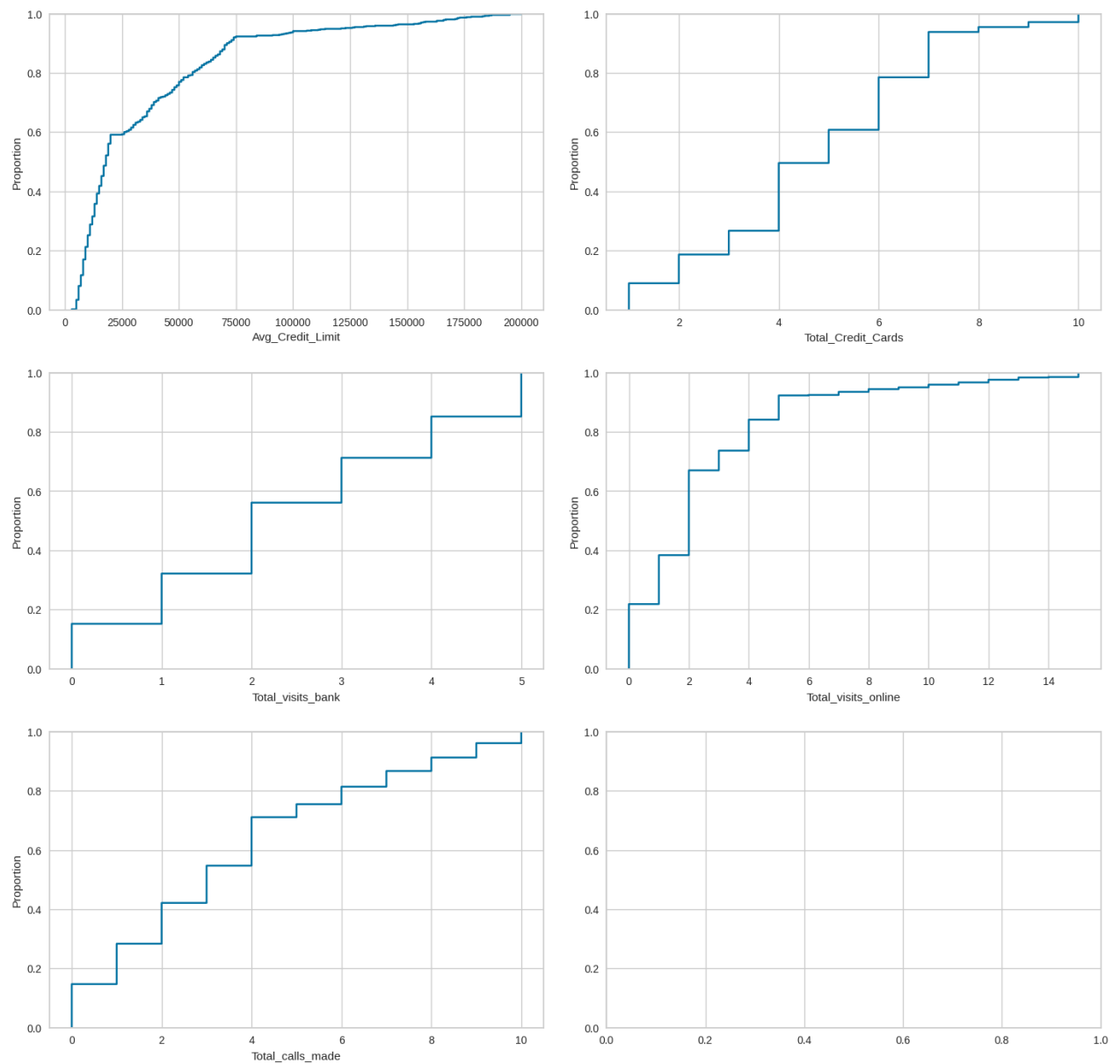




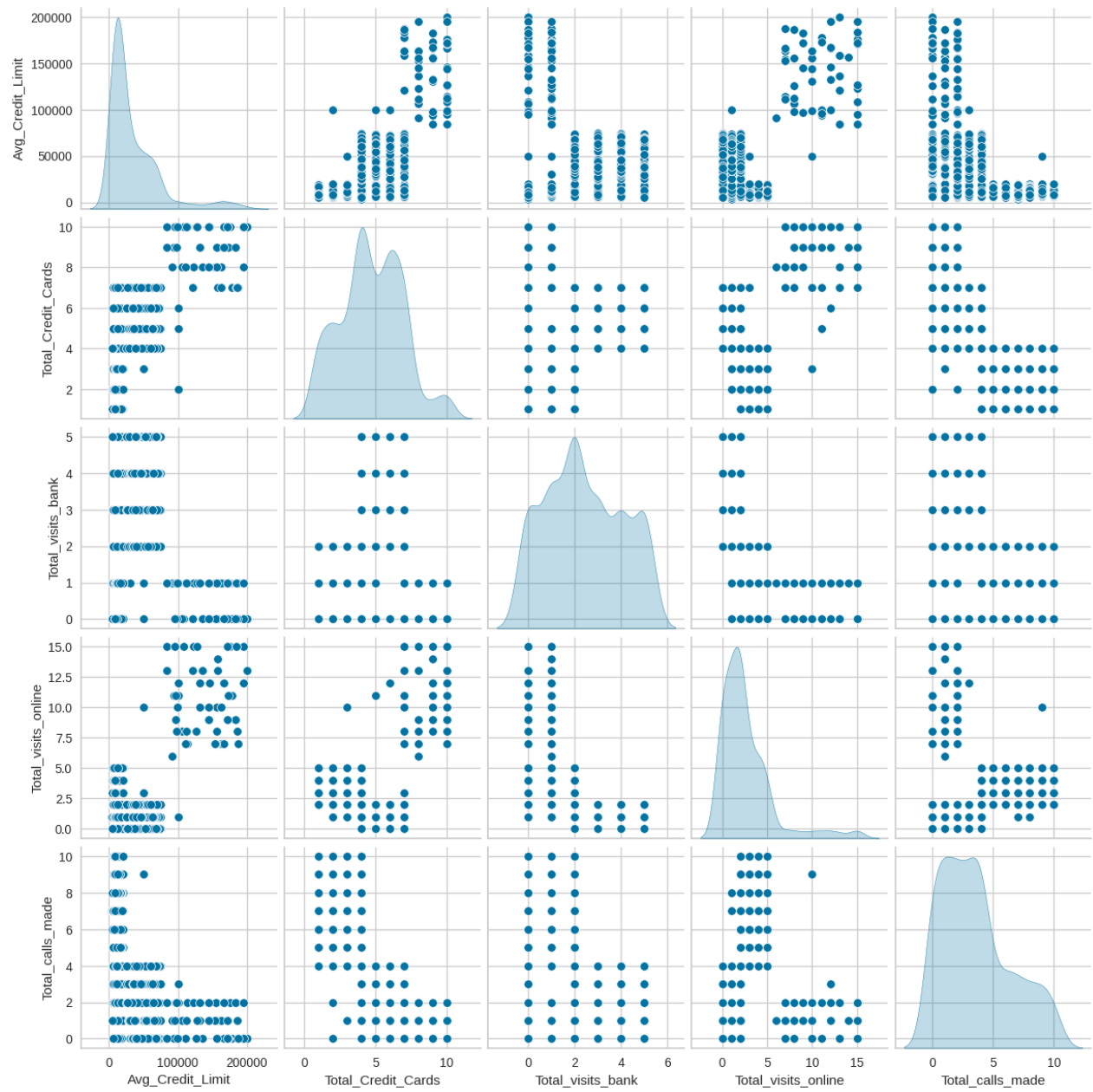


## 1.1.2 CDF Plot of Numerical Variables

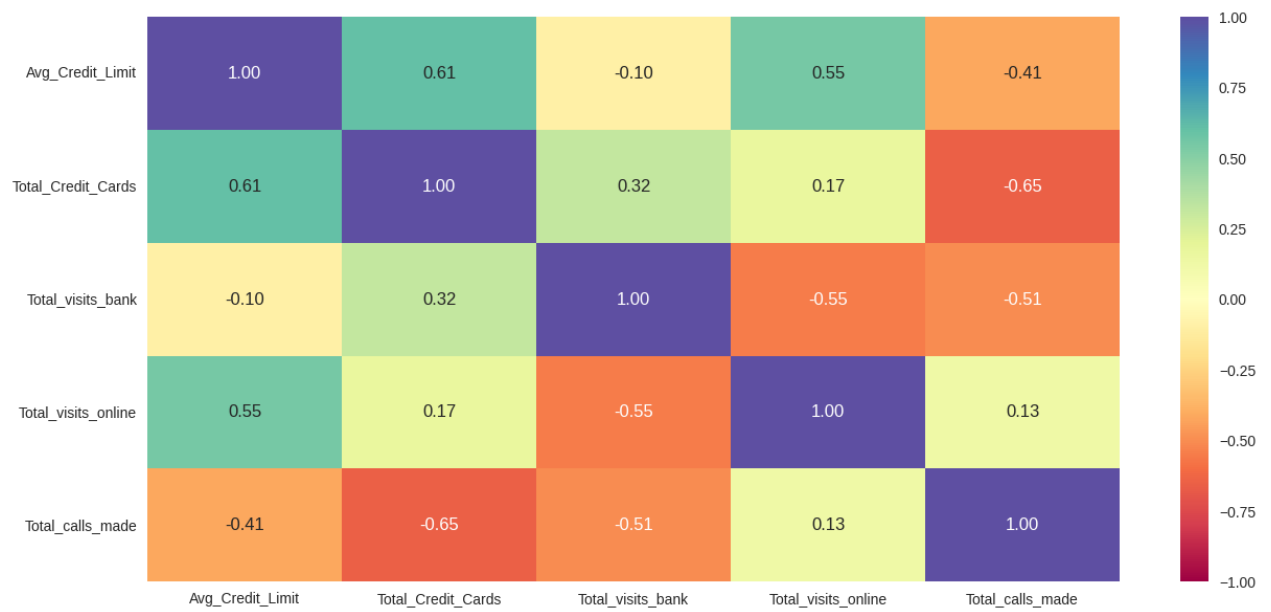
CDF plot of numerical variables



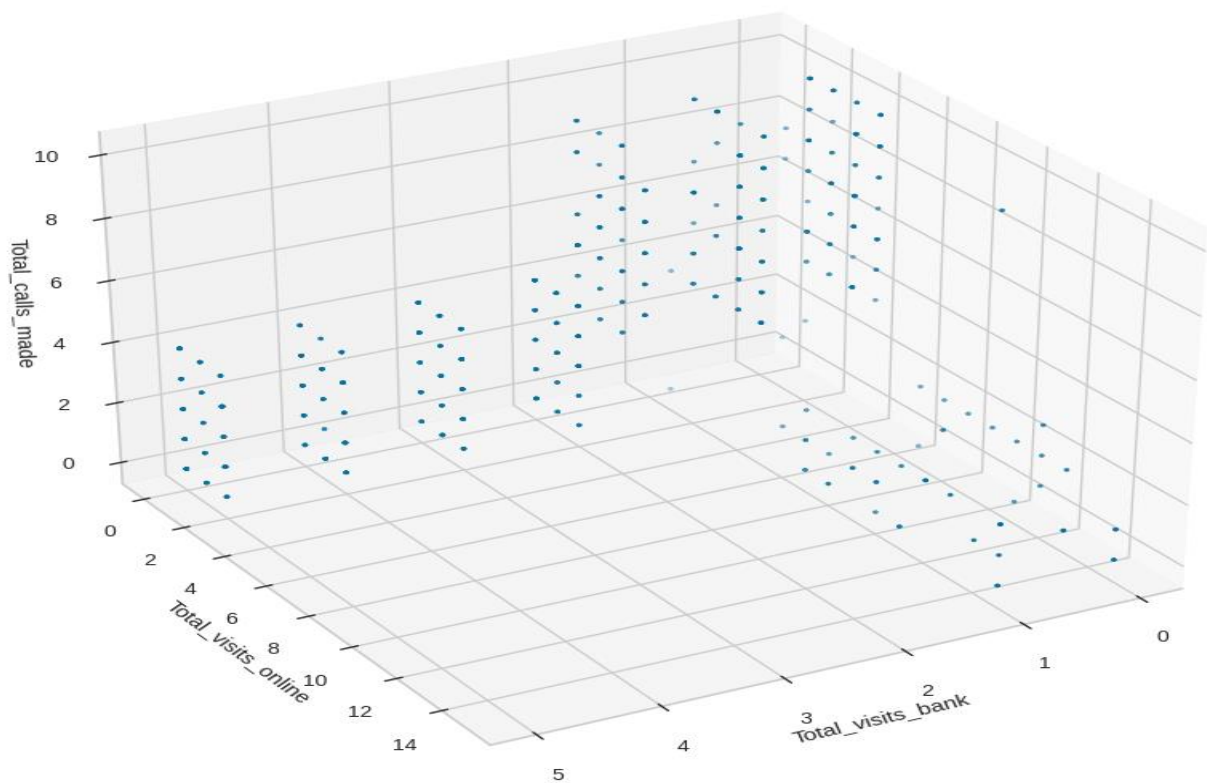
### 1.1.3 Bivariate Analysis



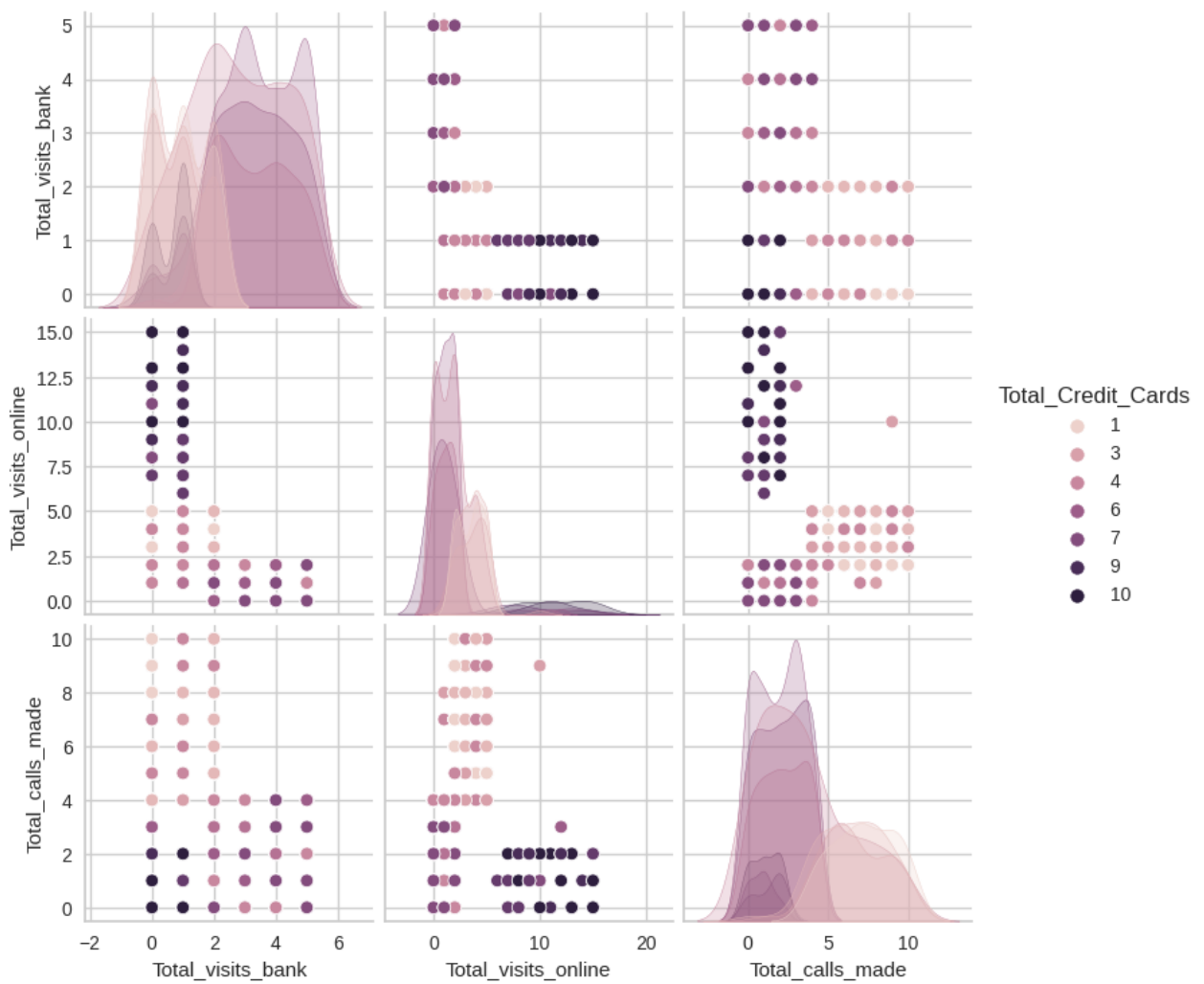
## Correlation Plot



## 3D



## Pair Plot



## 1.2 Data Pre Processing

### ➤ Checking the missing values

```
➡ Sl_No          0
   Customer_Key   0
   Avg_Credit_Limit  0
   Total_Credit_Cards  0
   Total_visits_bank  0
   Total_visits_online  0
   Total_calls_made  0
   dtype: int64
```

- There are no missing values in the data.

### ➤ Checking the number of unique values in each column

```
Sl_No          660
Customer Key    655
Avg_Credit_Limit  110
Total_Credit_Cards  10
Total_visits_bank   6
Total_visits_online  16
Total_calls_made   11
dtype: int64
```

- Checking for duplicates values

	Sl_No	Customer Key	Avg_Credit_Limit	Total_Credit_Cards	Total_visits_bank	Total_visits_online	Total_calls_made
48	49	37252	6000	4	0	2	8
432	433	37252	59000	6	2	1	2
	Sl_No	Customer Key	Avg_Credit_Limit	Total_Credit_Cards	Total_visits_bank	Total_visits_online	Total_calls_made
4	5	47437	100000	6	0	12	3
332	333	47437	17000	7	3	1	0
	Sl_No	Customer Key	Avg_Credit_Limit	Total_Credit_Cards	Total_visits_bank	Total_visits_online	Total_calls_made
411	412	50706	44000	4	5	0	2
541	542	50706	60000	7	5	2	2
	Sl_No	Customer Key	Avg_Credit_Limit	Total_Credit_Cards	Total_visits_bank	Total_visits_online	Total_calls_made
391	392	96929	13000	4	5	0	0
398	399	96929	67000	6	2	2	2
	Sl_No	Customer Key	Avg_Credit_Limit	Total_Credit_Cards	Total_visits_bank	Total_visits_online	Total_calls_made
104	105	97935	17000	2	1	2	10
632	633	97935	187000	7	1	7	0

## ➤ Outlier Detection

The following are the outliers in the data:

```
Avg_Credit_Limit : [153000, 155000, 156000, 156000, 157000, 158000, 163000, 163000, 166000, 166000, 167000, 171000, 172000, 172000, 173000, 176000, 178000, 183000, 184000, 186000, 187000]
```

Total\_Credit\_Cards : []

Total\_visits\_bank : []

```
Total_visits_online : [12, 12, 12, 12, 12, 12, 13, 13, 13, 13, 13, 14, 15, 15, 15, 15, 15, 15, 15, 15, 15]
```

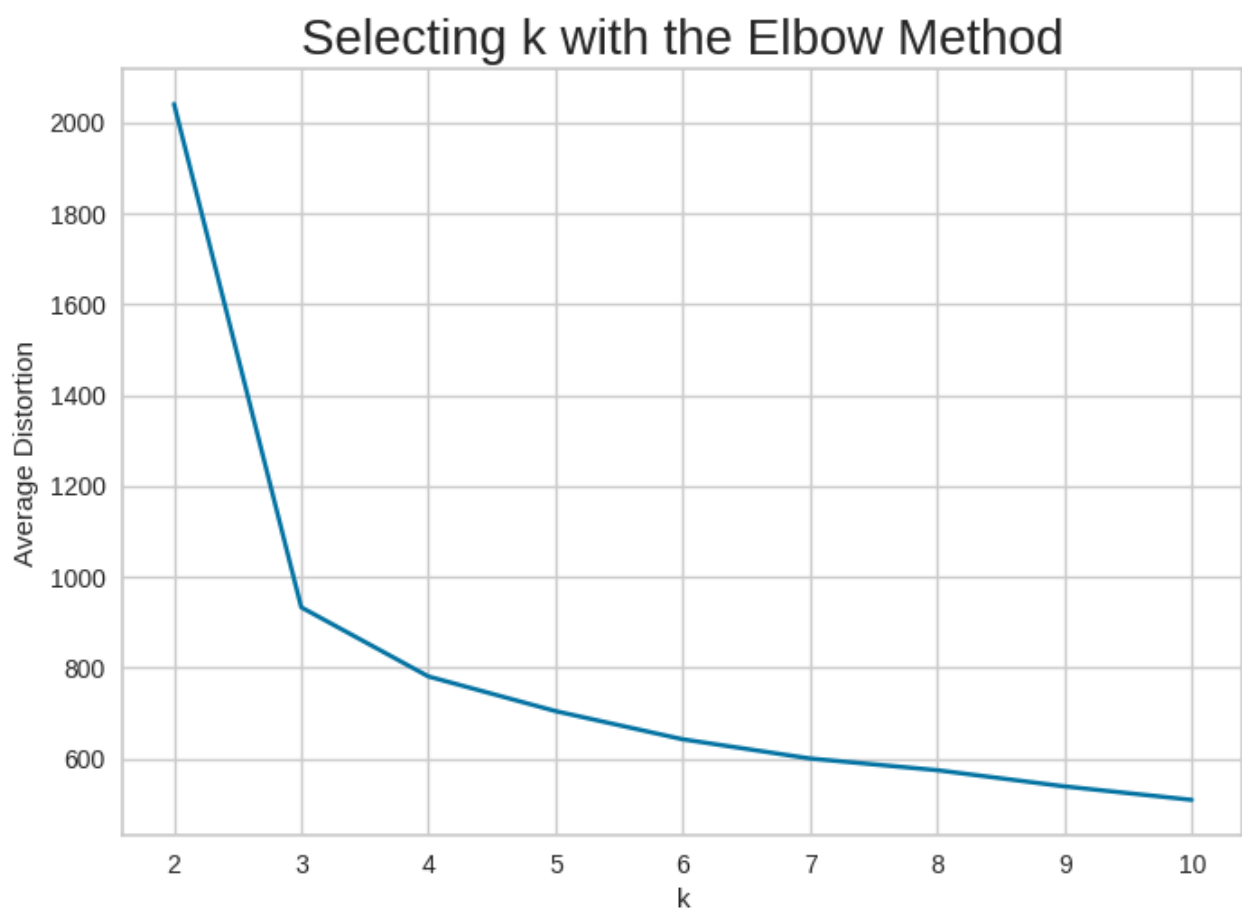
```
Total_calls_made : []
```



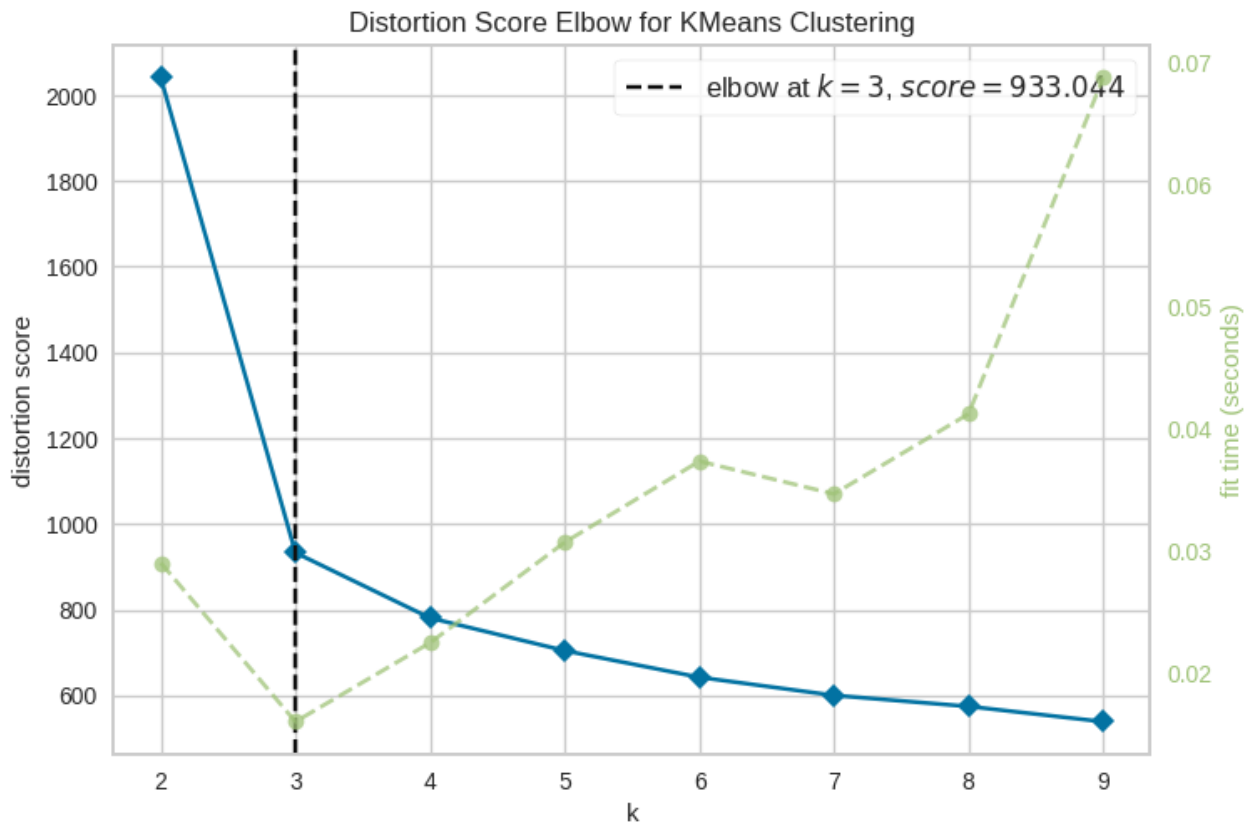
## 1.3 K-means Clustering

K-means is often referred to as Lloyd's algorithm. In basic terms, the algorithm has three steps. The first step chooses the initial centroids, with the most basic method being to choose samples from the dataset. After initialization, K-means consists of looping between the two other steps. The first step assigns each sample to its nearest centroid. The second step creates new centroids by taking the mean value of all of the samples assigned to each previous centroid. The difference between the old and the new centroids are computed and the algorithm repeats these last two steps until this value is less than a threshold. In other words, it repeats until the centroids do not move significantly.

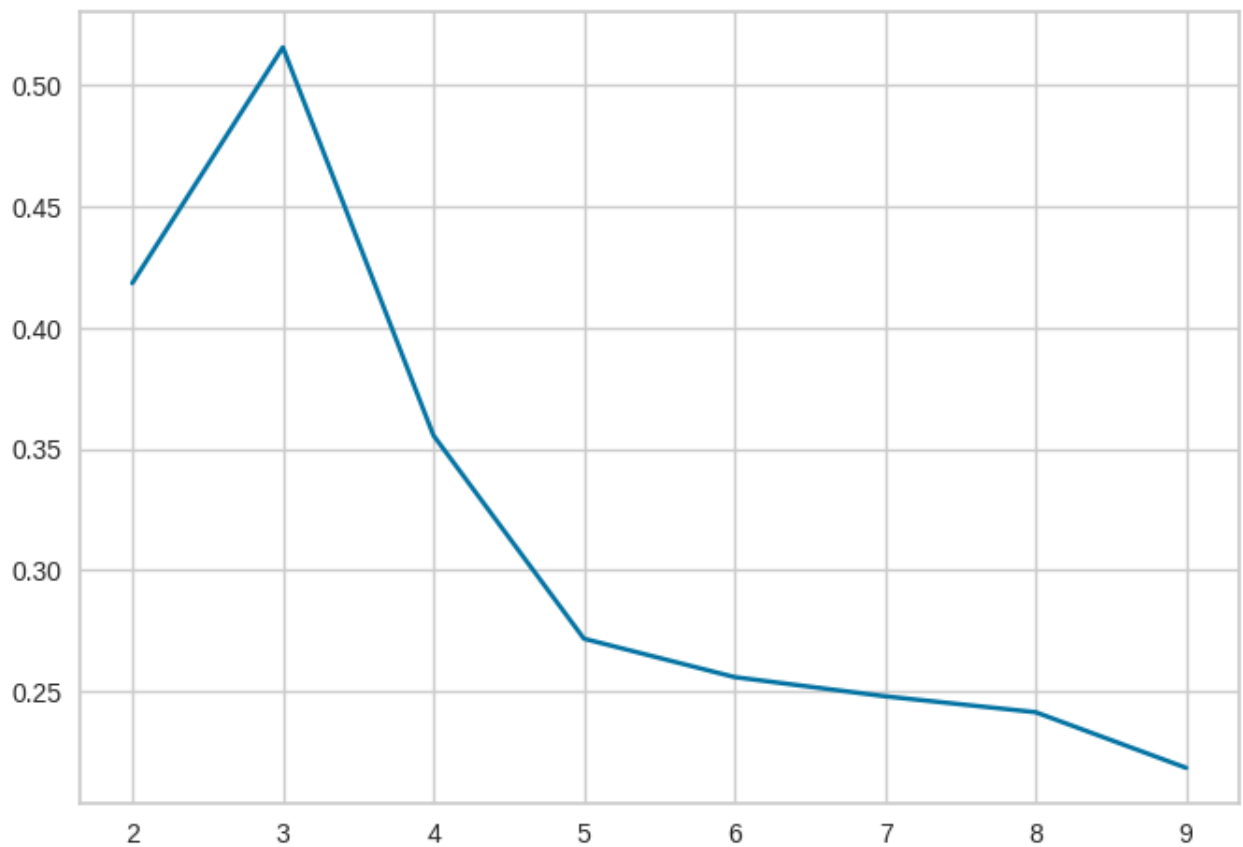
Elbow Curve to get the right number of Clusters



Appropriate value for k seems to be 3

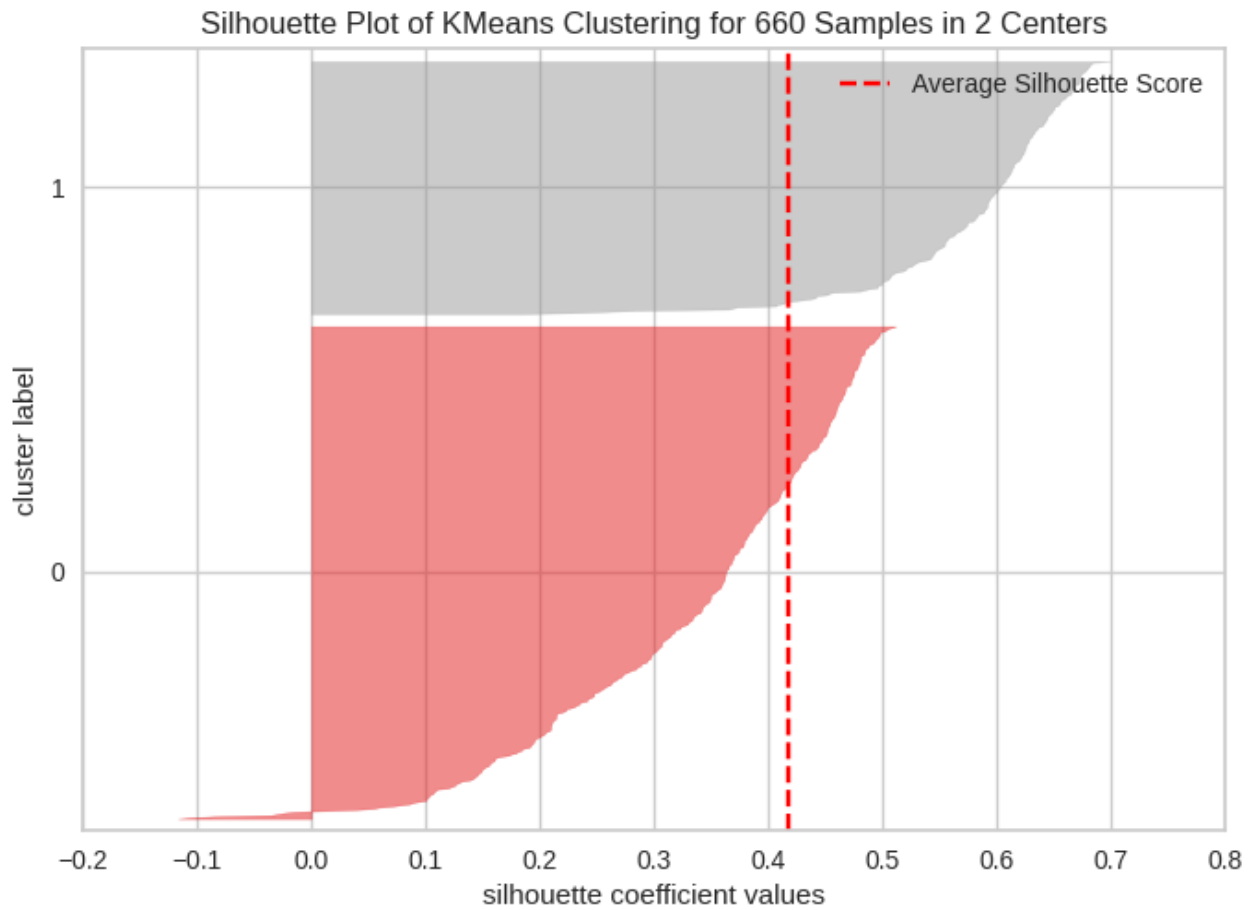


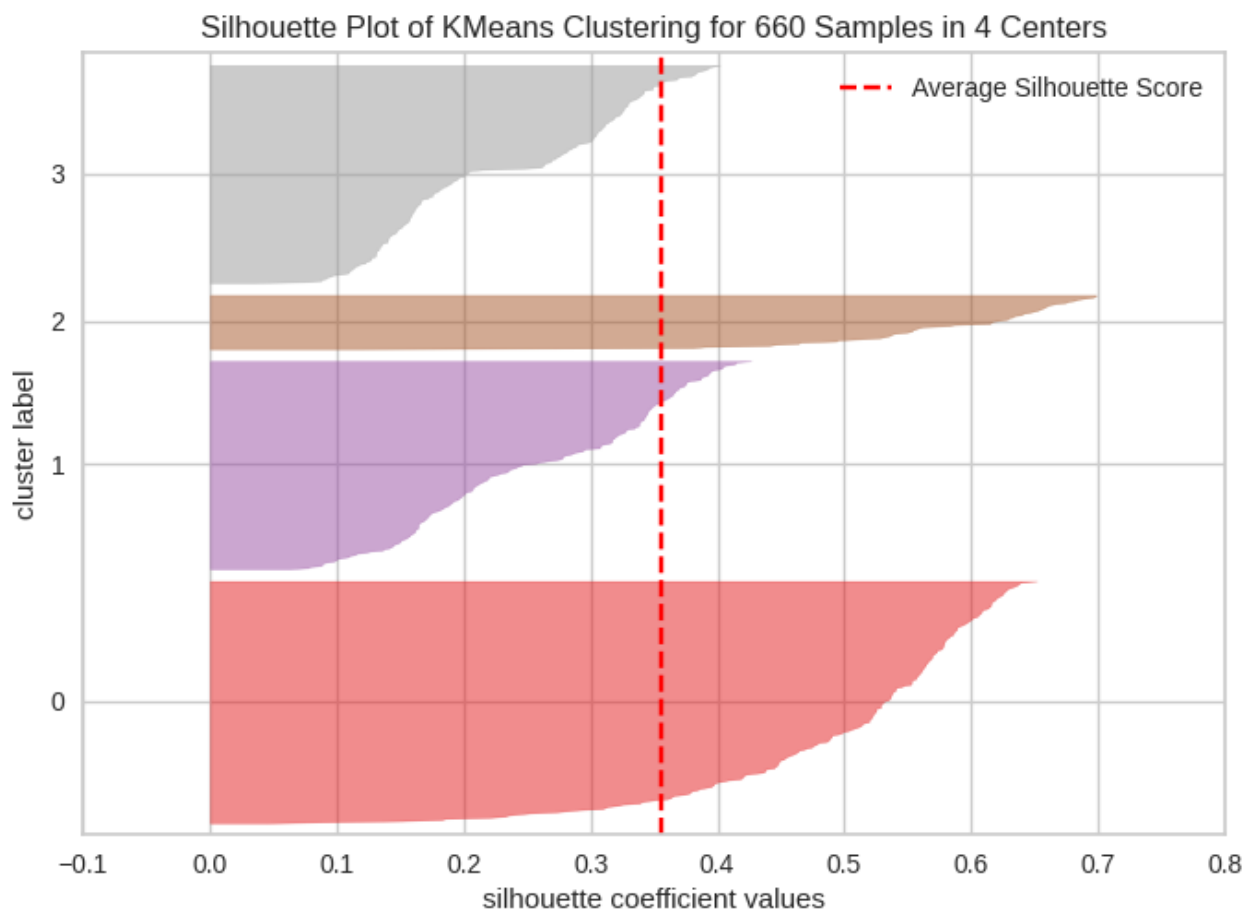
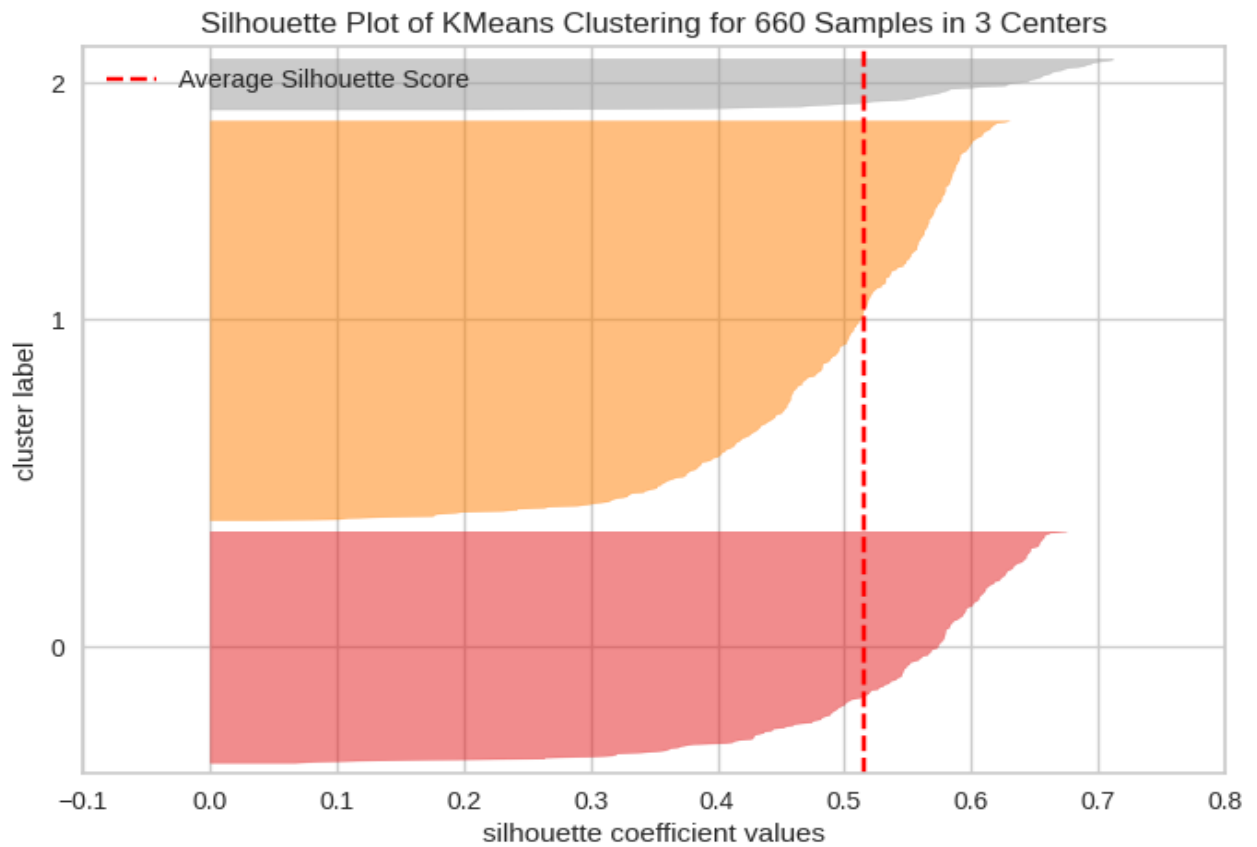
Let's check the silhouette scores

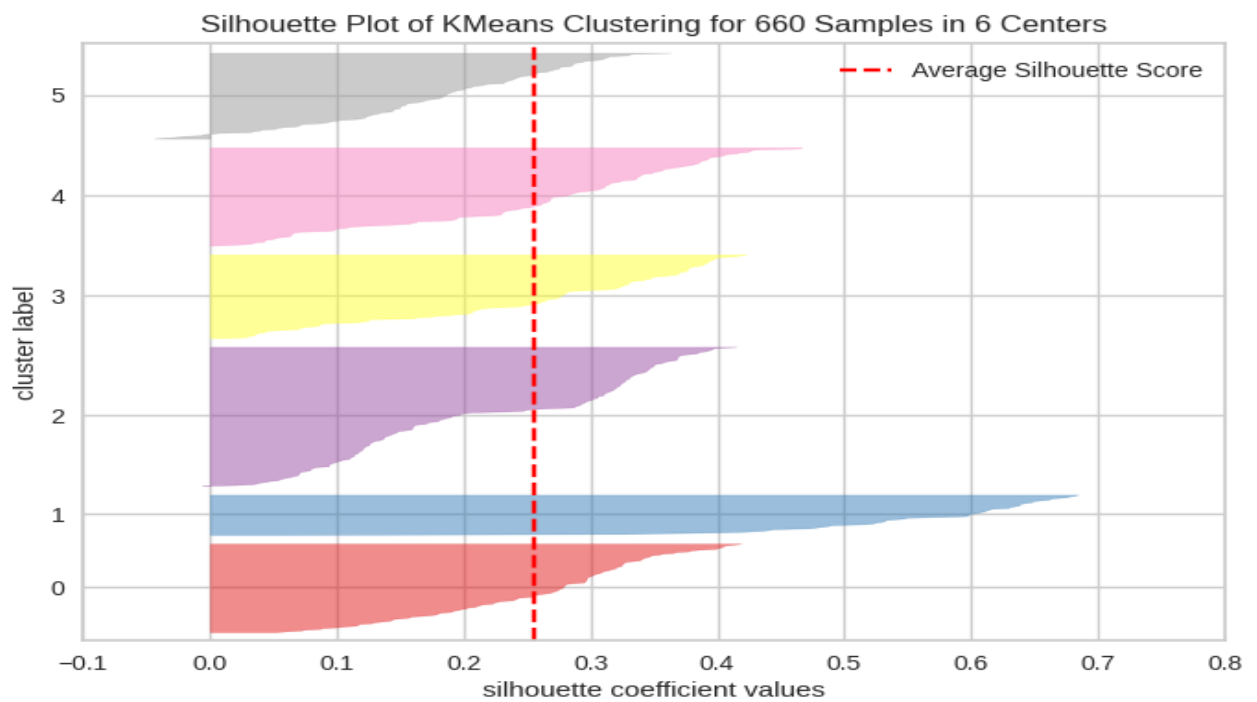
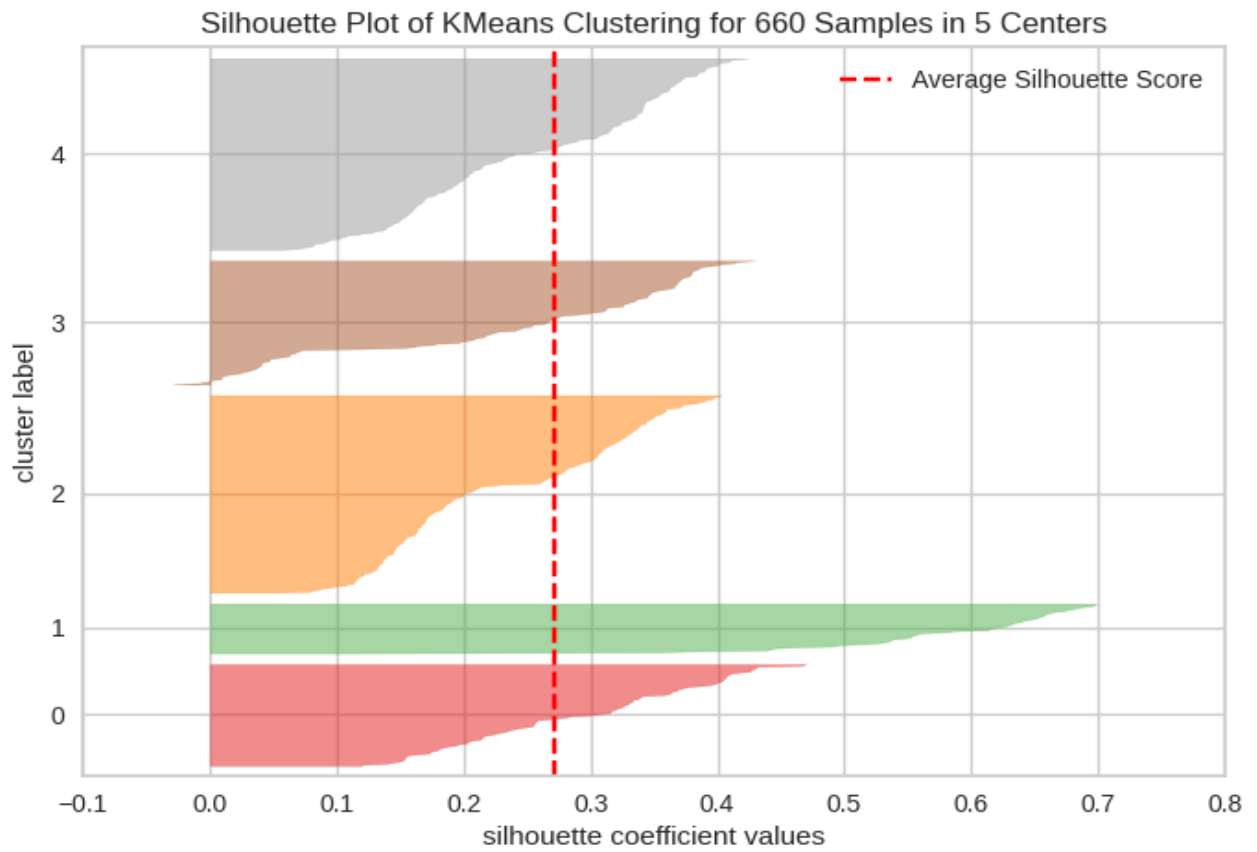


Silhouette score for 3 clusters is highest. So, we will choose 3 as value of k.

Let's also visualize the silhouettes created by each of the clusters for two values of K, 3 and 4







## Creating Final Model



CPU times: user 25 ms, sys: 1.59 ms, total: 26.5 ms  
Wall time: 18.3 ms



KMeans

```
KMeans(n_clusters=3, random_state=1)
```

## 1.4 Hierarchical Clustering

Hierarchical clustering is a general family of clustering algorithms that build nested clusters by merging or splitting them successively. This hierarchy of clusters is represented as a tree (or dendrogram). The root of the tree is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample.

The Agglomerative Clustering object performs a hierarchical clustering using a bottom up approach: each observation starts in its own cluster, and clusters are successively merged together. The linkage criteria determines the metric used for the merge strategy:

Ward minimizes the sum of squared differences within all clusters. It is a variance-minimizing approach and in this sense is similar to the k-means objective function but tackled with an agglomerative hierarchical approach.

Maximum or complete linkage minimizes the maximum distance between observations of pairs of clusters.

Average linkage minimizes the average of the distances between all observations of pairs of clusters.

Single linkage minimizes the distance between the closest observations of pairs of clusters.

### Cophenetic Correlations

The cophenetic correlation for a cluster tree is defined as the linear correlation coefficient between the cophenetic distances obtained from the tree, and the original distances (or dissimilarities) used to construct the tree. Thus, it is a measure of how faithfully the tree represents the dissimilarities among observations.

The cophenetic distance between two observations is represented in a dendrogram by the height of the link at which those two observations are first joined. That height is the distance between the two subclusters that are merged by that link.

The magnitude of this value should be very close to 1 for a high-quality solution. This measure can be used to compare alternative cluster solutions obtained using different algorithms.

Cophenetic correlation for Euclidean distance and single linkage is 0.7391220243806552.  
Cophenetic correlation for Euclidean distance and complete linkage is 0.8599730607972423.  
Cophenetic correlation for Euclidean distance and average linkage is 0.8977080867389372.  
Cophenetic correlation for Euclidean distance and weighted linkage is 0.8861746814895477.  
Cophenetic correlation for Chebyshev distance and single linkage is 0.7382354769296767.  
Cophenetic correlation for Chebyshev distance and complete linkage is 0.8533474836336782.  
Cophenetic correlation for Chebyshev distance and average linkage is 0.8974159511838106.  
Cophenetic correlation for Chebyshev distance and weighted linkage is 0.8913624010768603.  
Cophenetic correlation for Mahalanobis distance and single linkage is 0.7058064784553605.  
Cophenetic correlation for Mahalanobis distance and complete linkage is 0.6663534463875359.  
Cophenetic correlation for Mahalanobis distance and average linkage is 0.8326994115042136.  
Cophenetic correlation for Mahalanobis distance and weighted linkage is 0.7805990615142518.  
Cophenetic correlation for Cityblock distance and single linkage is 0.7252379350252723.  
Cophenetic correlation for Cityblock distance and complete linkage is 0.8731477899179829.  
Cophenetic correlation for Cityblock distance and average linkage is 0.896329431104133.  
Cophenetic correlation for Cityblock distance and weighted linkage is 0.8825520731498188.

---

➤ Highest cophenetic correlation is 0.8977080867389372, which is obtained with Euclidean distance and average linkage.

---

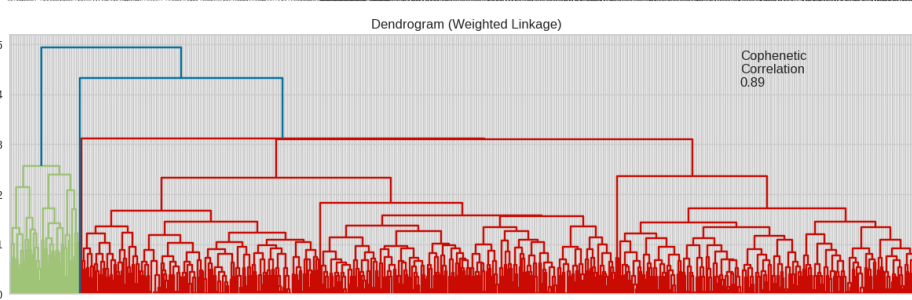
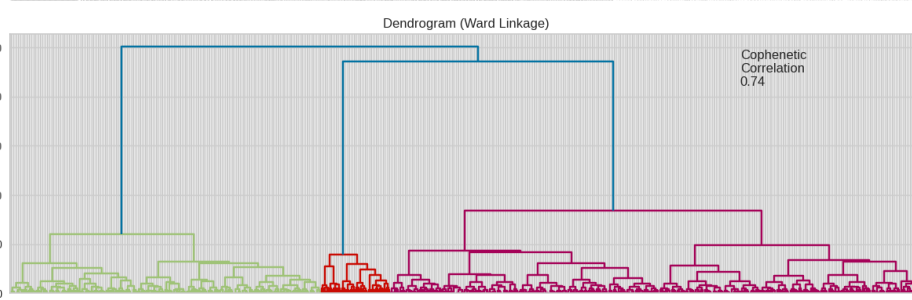
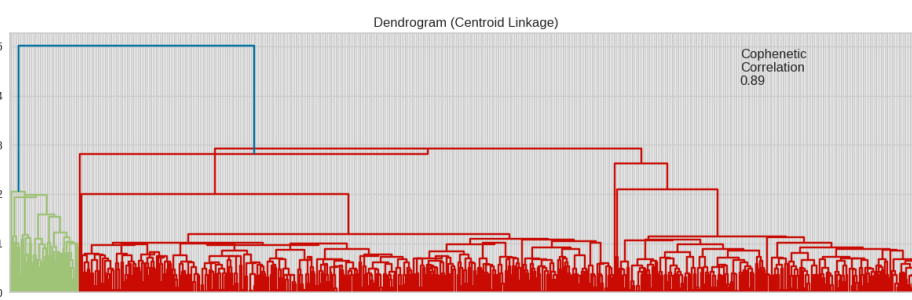
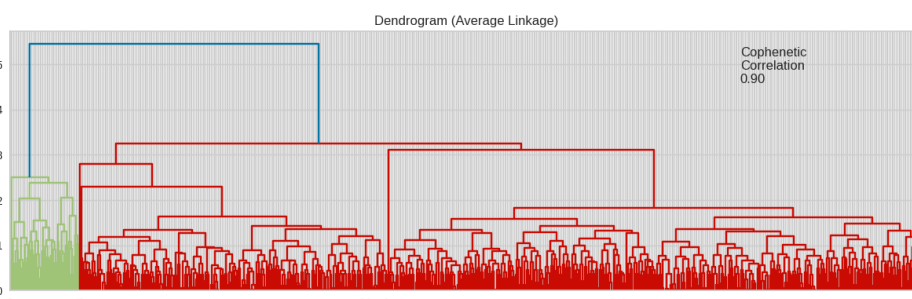
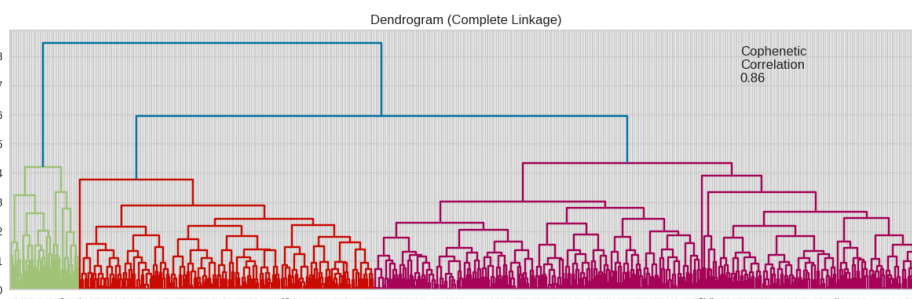
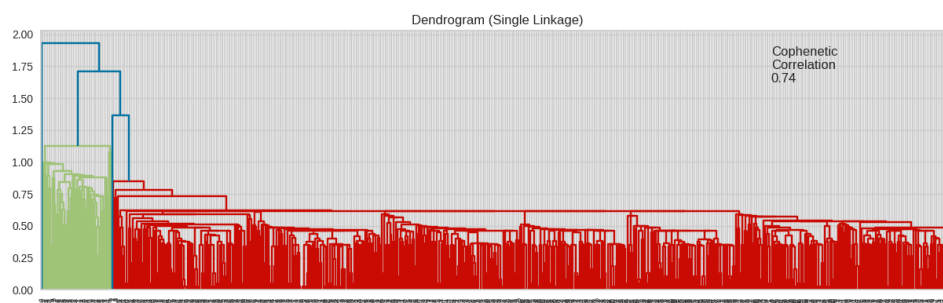
We see that the cophenetic correlation is maximum with Euclidean distance and Average Linkage.

Let's see the dendrograms for the different linkage methods.

#### ➤ Dendograms

A Dendrogram, in general, is a diagram that shows the hierarchical relationship between objects. It is most commonly created as an output from hierarchical clustering. The main use of a dendrogram is to work out the best way to allocate objects to clusters.



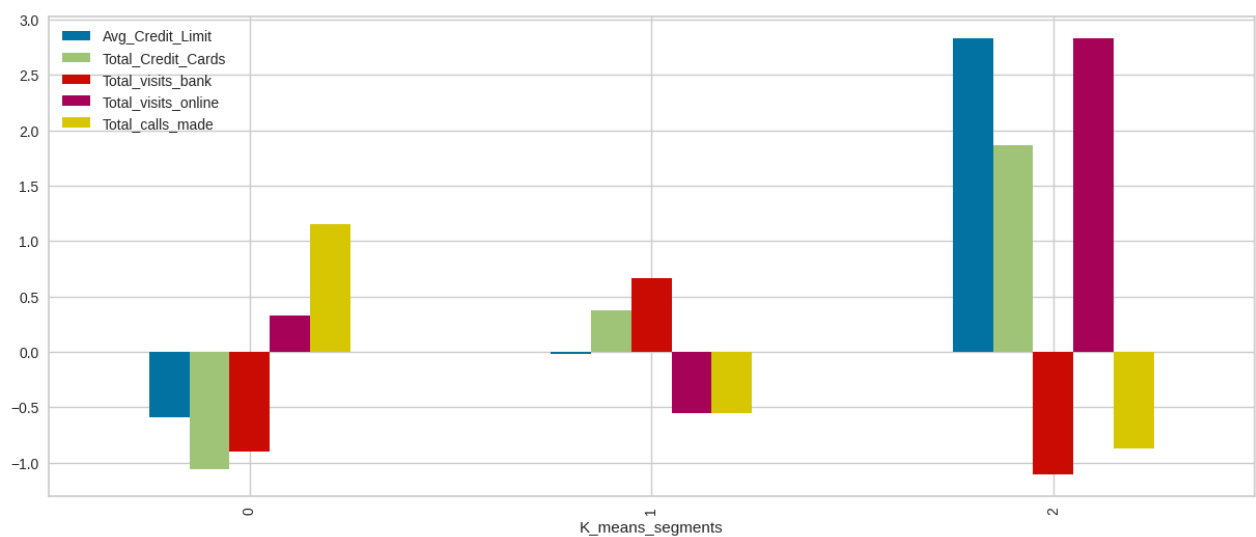


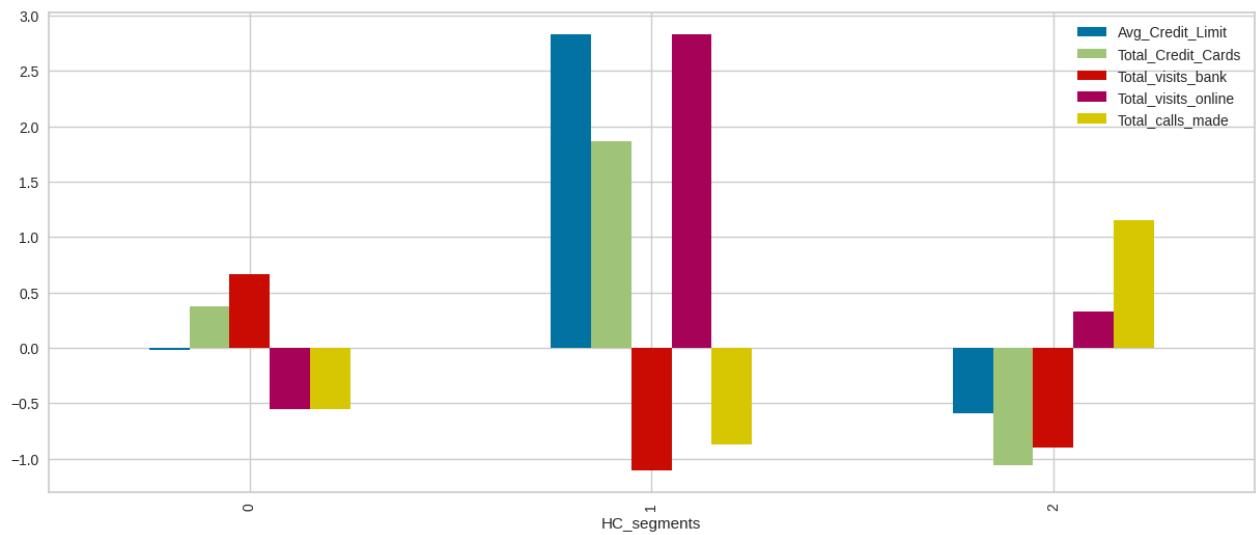
## ➤ Build Agglomerative Clustering model

	Avg_Credit_Limit	Total_Credit_Cards	Total_visits_bank	Total_visits_online	Total_calls_made	count_in_each_segment
HC_segments						
0	33713.178295	5.511628	3.485788	0.984496	2.005168	387
1	141040.000000	8.740000	0.600000	10.900000	1.080000	50
2	12197.309417	2.403587	0.928251	3.560538	6.883408	223

	Avg_Credit_Limit	Total_Credit_Cards	Total_visits_bank	Total_visits_online	Total_calls_made	count_in_each_segment
HC_Clusters						
0	33713.178295	5.511628	3.485788	0.984496	2.005168	387
1	141040.000000	8.740000	0.600000	10.900000	1.080000	50
2	12197.309417	2.403587	0.928251	3.560538	6.883408	223

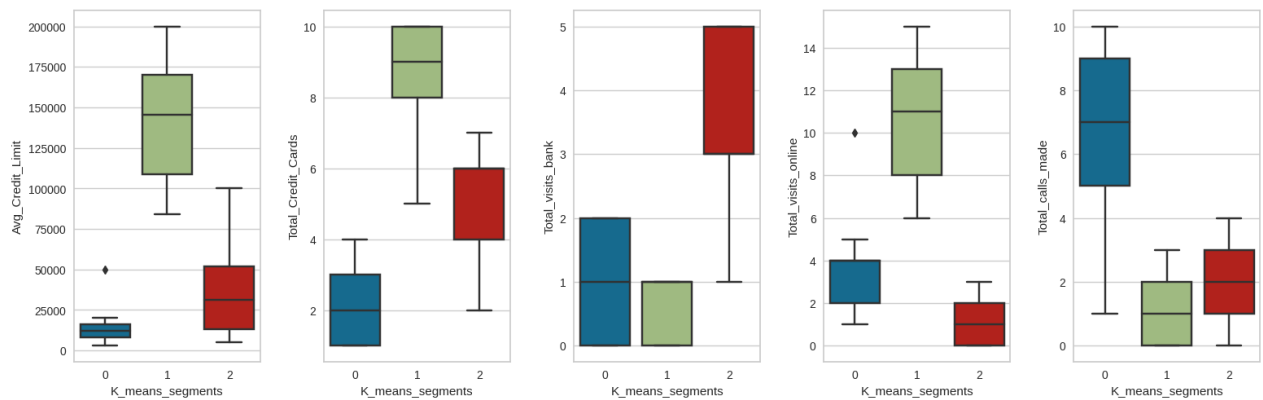
## Analyzing the segments using Box Plot



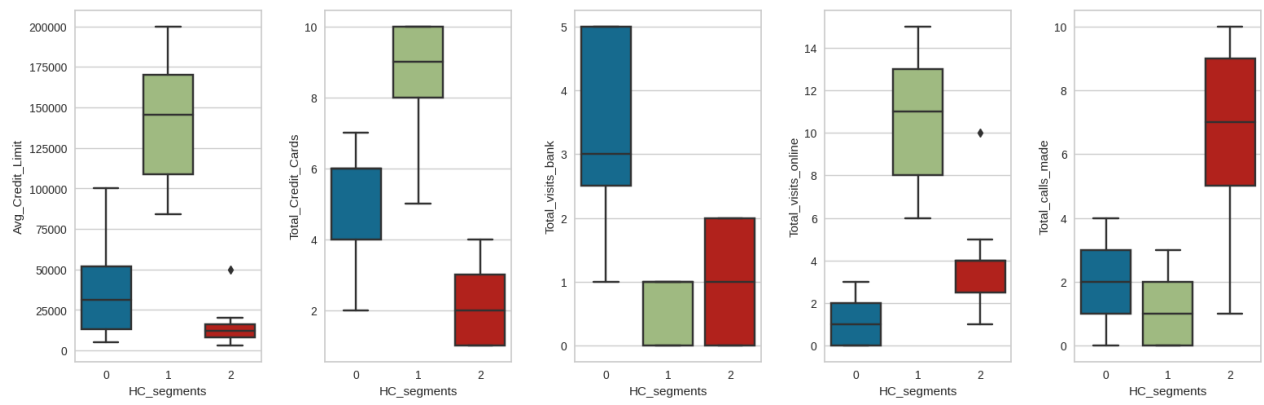


Let's create some plots on the original data to understand the customer distribution among the clusters.

Boxplot of numerical variables for each cluster obtained using K-means Clustering



Boxplot of numerical variables for each cluster obtained using Hierarchical Clustering



## 1.5 K-Means Vs Hierarchical Clustering

### ➤ Conclusions K-means

- Cluster 0 : Seems to be type of clients with the lowest credit limit, more willing to visit the bank.
- Cluster 1 : Mid range type of client a mix between cluster 2 and cluster 0.
- Cluster 2 : Seems to be the type of client with the highest credit limit, more willing to use online banking system.

### ➤ Conclusions Hierarchical clusters

- Cluster 0 : Seems to be type of clients with the lowest credit limit.
- Cluster 1 : Seems to be type of clients with the highest credit limit. A client that demands online and mobile contact.
- Cluster 2 : Seems to be the type of client with the mid credit limit range, a type of client that do not visit the bank neither use the online banking

***Accessing customers with an upper credit limit seems to be the best strategy based on the analysis of the clusters.***

## 1.6 Actionable Insights & Recommendations

3 different clusters with two different methods kmeans and hierarchical clustering.

### Insights K-means

- **Cluster 0 :**
  - Avg\_Credit\_Limit: The mid end type of client.
  - Total\_Credit\_Cards: The mid end type of client.
  - Total\_visits\_bank: Visit the most the bank.
  - Total\_visits\_online: Doesn't access much the online bank.
  - Total\_calls\_made: Don't call as much as expected.
- **Cluster 1 :**
  - Avg\_Credit\_Limit: The lowest end type of client.
  - Total\_Credit\_Cards: The lowest end type of client.
  - Total\_visits\_bank: Doesn't visit much the bank.
  - Total\_visits\_online: Average end in terms of online banking usage.
  - Total\_calls\_made: The highest end type of client.
- **Cluster 2 :**
  - Avg\_Credit\_Limit: The highest end type of client.
  - Total\_Credit\_Cards: The highest end type of client.
  - Total\_visits\_bank: The lowest end type of client.
  - Total\_visits\_online: The highest end type of client.
  - Total\_calls\_made: The lowest end type of client.

### Insights hierarchical clustering

- **Cluster 0 :**
  - Avg\_Credit\_Limit: The lowest end type of client.
  - Total\_Credit\_Cards: The lowest end type of client.
  - Total\_visits\_bank: The lowest end type of client.
  - Total\_visits\_online: The mid end type of client.
  - Total\_calls\_made: The mid end type of client.
- **Cluster 1 :**
  - Avg\_Credit\_Limit: The highest end type of client.
  - Total\_Credit\_Cards: The mid end type of client.
  - Total\_visits\_bank: The lowest end type of client.
  - Total\_visits\_online: The highest end type of client.
  - Total\_calls\_made: The highest end type of client.
- **Cluster 2 :**
  - Avg\_Credit\_Limit: The mid end type of client.
  - Total\_Credit\_Cards: The highest end type of client.
  - Total\_visits\_bank: The mid end type of client.
  - Total\_visits\_online: The lowest end type of client.
  - Total\_calls\_made: The lowest end type of client.

## Recommendations

***Accessing customers with an upper credit limit seems to be the best strategy based on the analysis of the clusters.***

- Kmeans -> Cluster 2 : Explore online marketing campaigns to this type of client. High financial potential in comparison with others clusters and desirous to access the bank online.
- Hierarchical -> Cluster 1 : Explore online marketing campaigns to this type of client, and also develop a better approach in the call center. This type of client is willing to access the bank online however needs a better call center service.

## 1.7 PCA Transformation

Although there are only 5 dimensions, it'll be really cool to be able to visualize the clusters at 3 dimensional space without losing much of the information. Let's use PCA to reduce the dimensions so that 80% of the variance in the data is explained.

	Avg_Credit_Limit	Total_Credit_Cards	Total_visits_bank	Total_visits_online	Total_calls_made	HC_Clusters
0	2.3989	-1.2492	-0.8605	-0.6198	-1.2515	2
1	0.6436	-0.7876	-1.4737	2.7058	1.8919	0
2	0.6436	1.0590	-0.8605	0.2671	0.1455	0
3	-0.0585	0.1357	-0.8605	-0.6198	0.1455	0
4	2.3989	0.5973	-1.4737	2.7058	-0.2037	1
...	...	...	...	...	...	...
655	2.3638	2.4439	-0.8605	2.7058	-1.2515	1
656	1.8372	2.4439	-0.8605	2.7058	-0.5530	1
657	2.5745	1.5206	-0.8605	2.7058	-0.9023	1
658	2.5745	2.4439	-0.8605	2.7058	-1.2515	1
659	2.5745	1.9823	-1.4737	2.7058	-0.5530	1

660 rows × 6 columns

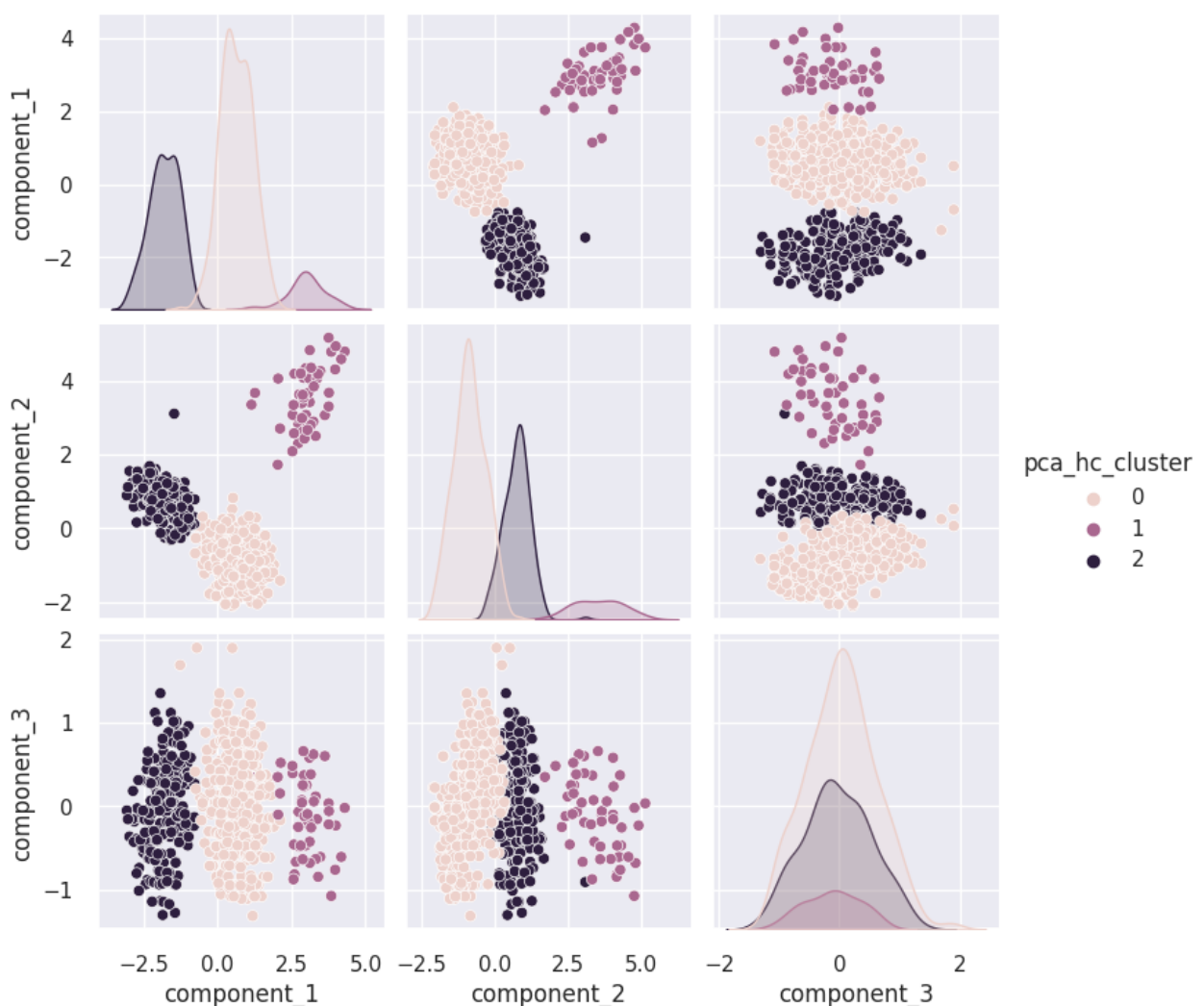


## 1.8 Interpretation of Principal Components

To interpret each component, we must compute the correlations between the original data and each principal component.

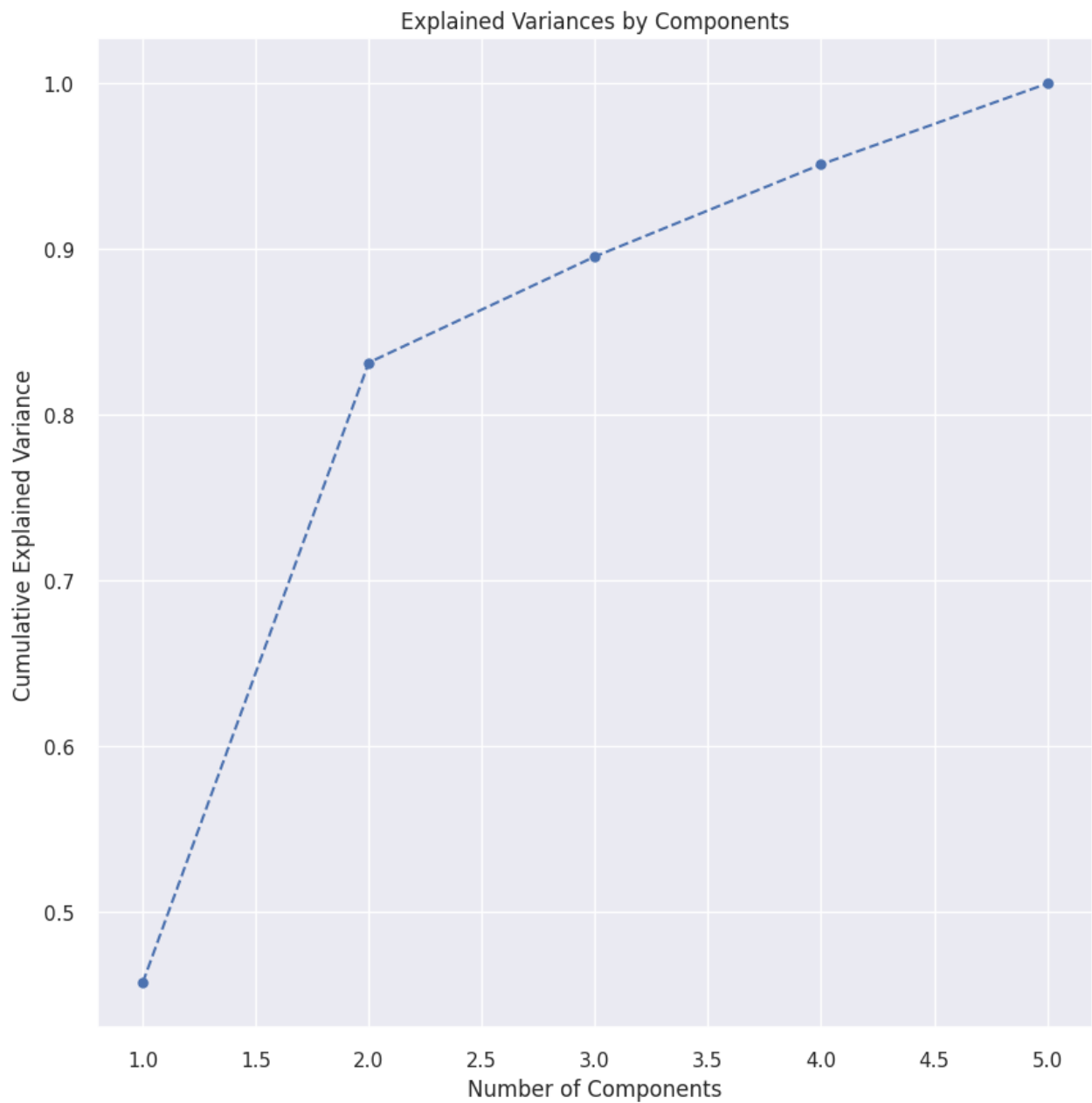
These correlations are obtained using the correlation procedure. In the variable statement, we include the first three principal components, component\_1, component\_2, and pcomponent\_3", in addition to all nine of the original variables. We use the correlations between the principal components and the original variables to interpret these principal components.

Because of standardization, all principal components will have a mean of 0. The standard deviation is also given for each of the components and these are the square root of the eigenvalue.



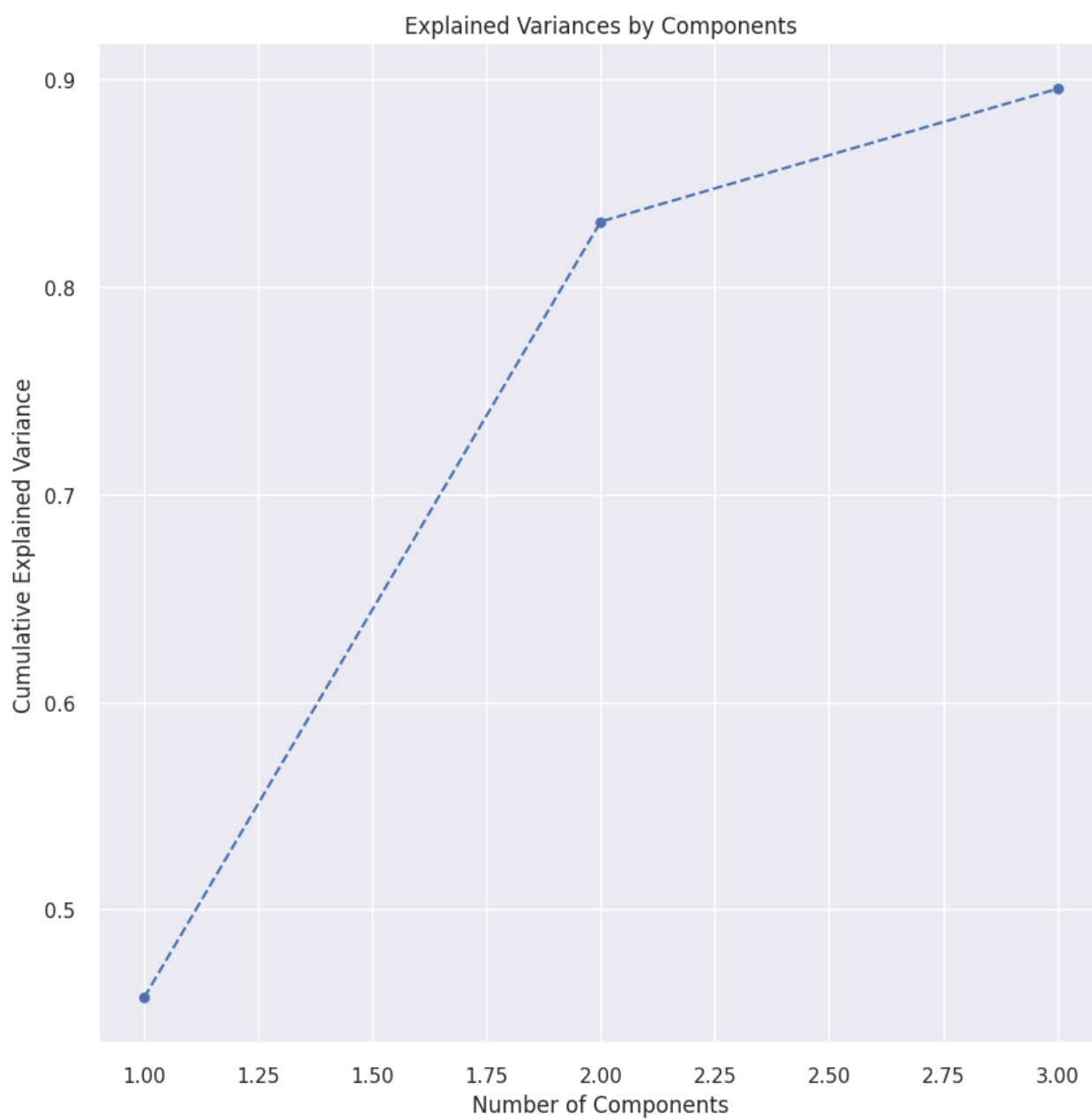
## 1.9 Variance Explanation

Let's check the variance explained by individual components.

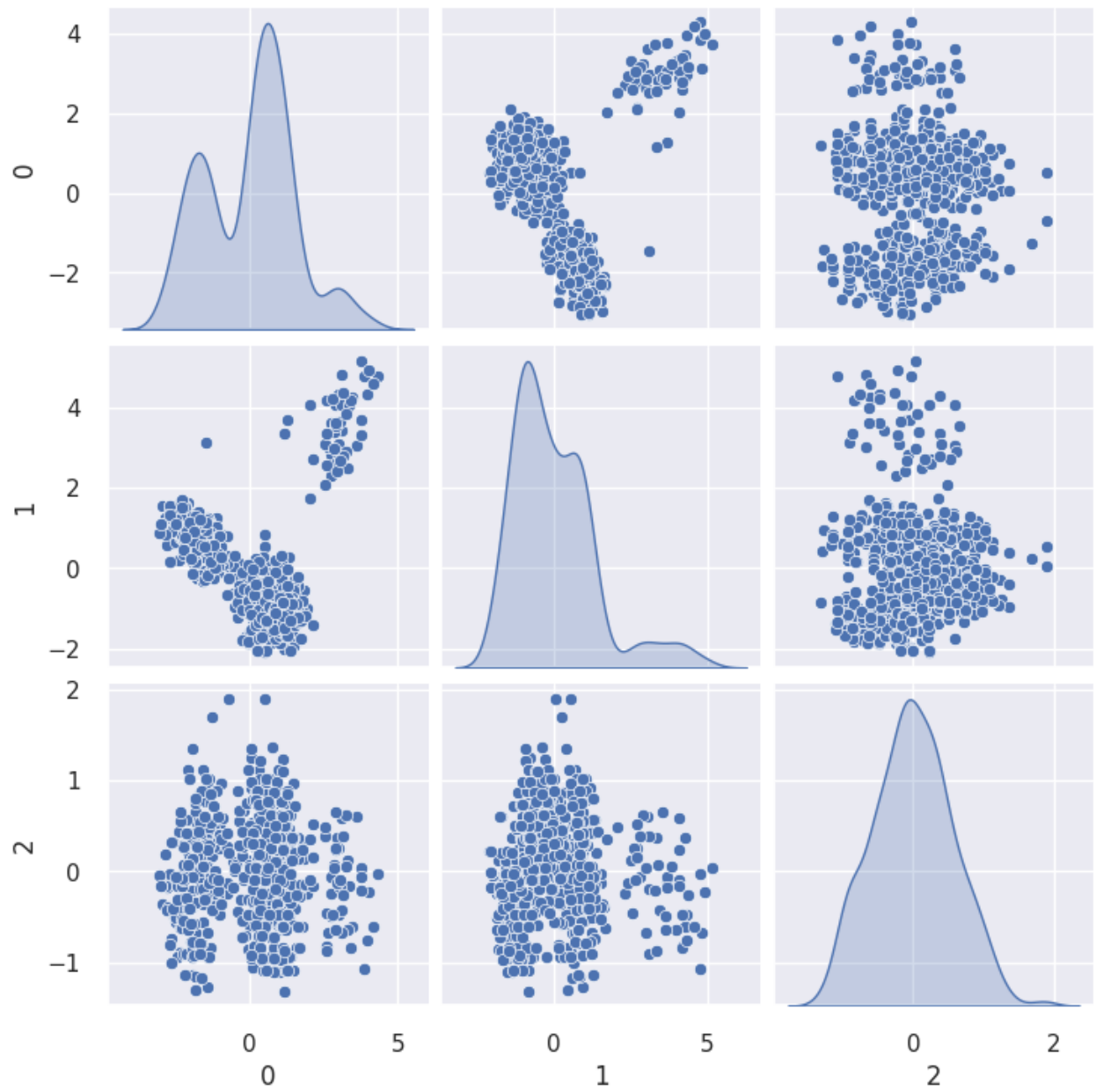


For 90% variance explained, the number of components looks to be 3.

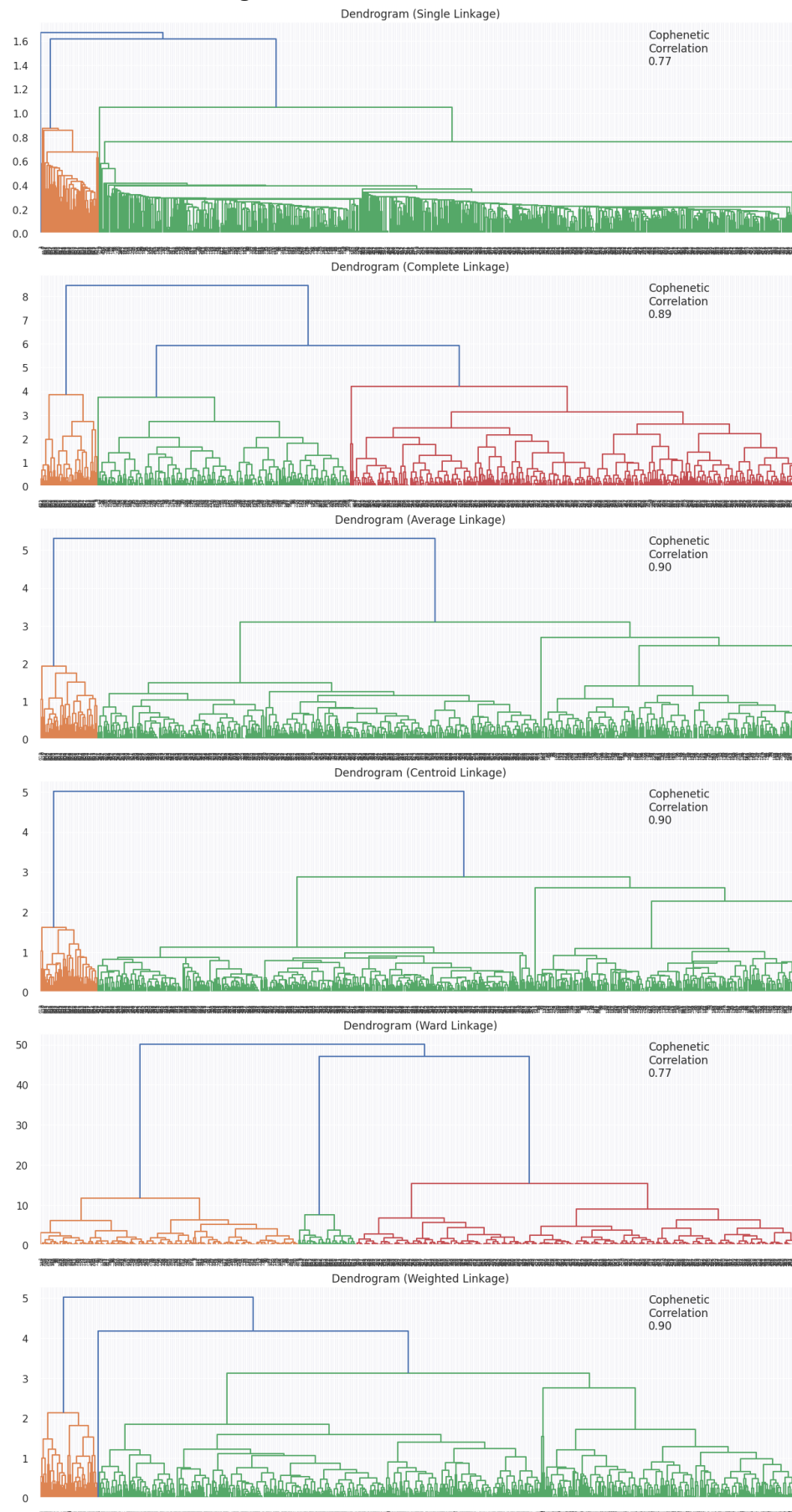
```
PCA  
PCA(n_components=3, svd_solver='full')
```



## 1.10 Visualization



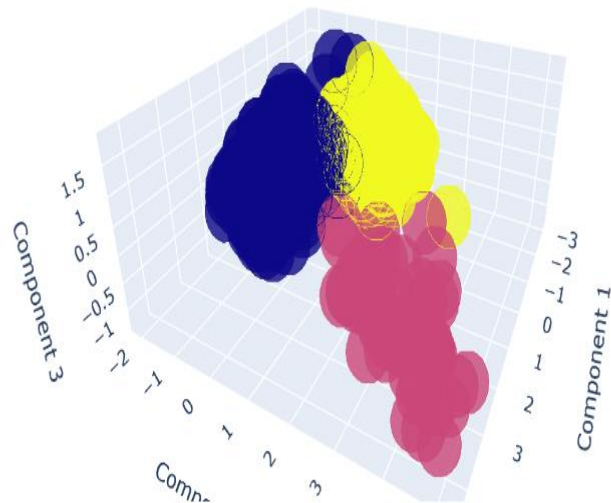
## Hierarchical Clustering on lower-dimensional data



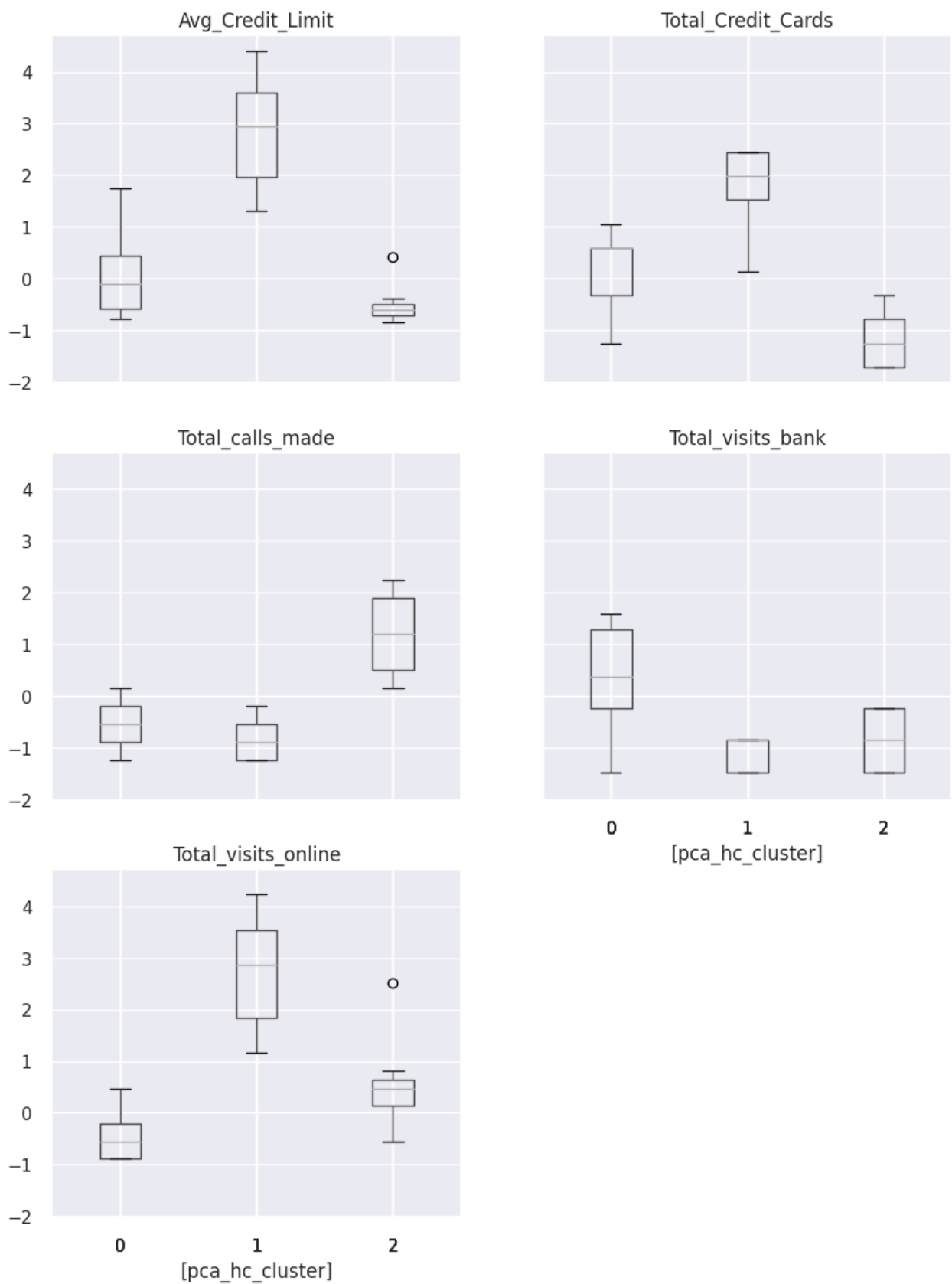
## Observations

- The cophenetic correlation is highest for average and centroid linkage method, but I will check with complete linkage as it has more distinct and separated clusters, and a cophenetic correlation of 0.89 (highest being 0.9).
- 3 appears to be the appropriate number of clusters from the dendrogram for complete linkage as well.

## Clusters



Boxplot grouped by pca\_hc\_cluster



## 1.11 Dimensionality Reduction Impact

### Impact

- **Cluster 0**
  - Second lowest Avg\_Credit\_Limit with a higher variance.
  - Second highest number in Total\_Credit\_Cards.
  - Total\_visits\_bank biggest one.
  - Total\_visits\_online smallest one.
  - Total\_calls\_made avg of 2.
  - Clients visit in person.
- **This type of customer has a good Avg\_Credit\_Limit and likes to visit the bank in person. It is important to identify visiting patterns and improve your experience.**
- **Cluster 1**
  - The lowest Avg\_Credit\_Limit with a smaller variance.
  - The lowest number in Total\_Credit\_Cards.
  - Total\_visits\_bank second smallest.
  - Total\_visits\_online second biggest.
  - Total\_calls\_made The highest number of clients whom make phone calls.
  - Clients would rather call.
- **This type of customer has a bad Avg\_Credit\_Limit and likes to call the bank. It is important to identify whether they are the type of customer the bank wants to invest in. Mainly because developing a better call center experience can be expensive and customers in this cluster enjoy the phone call experience.**
- **Cluster 2**
  - The highest Avg\_Credit\_Limit with a smallest variance.
  - The highest number in Total\_Credit\_Cards.
  - Total\_visits\_bank the smallest.
  - Total\_visits\_online the biggest.
  - Total\_calls\_made The smallest.
  - Clients would visit online.
- **This type of customer has a good Avg\_Credit\_Limit and likes to visit the online bank. It is important to identify patterns of online visits and improve your experience by tracking your internet flow showing new products and services.**
- **Cluster 3**
  - The second highest Avg\_Credit\_Limit with a bigger variance.
  - The second biggest number in Total\_Credit\_Cards.
  - Total\_visits\_bank second smallest.
  - Total\_visits\_online the smallest.
  - Total\_calls\_made The second smallest.
  - Clients visit in person.
- **This type of customer has a good Avg\_Credit\_Limit and likes to visit the bank in person. It is important to identify visiting patterns and improve their experience.**