Un-supervised learning project analysis

Problem statement :

AllLife Credit Card Customer Segmentation

Background: 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 personalised 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 back poorly. Based on this, the Operations team wants to upgrade the service delivery model, to ensure that customers queries are resolved faster. Head of Marketing and Head of Delivery both decide to reach out to the Data Science team for help.

Objective: To identify different segments in the existing customer based on their spending patterns as well as past interaction with the bank.

Key Questions:

1. How many different segments of customers are there?
2. How are these segments different from each other?
3. What are your recommendations to the bank on how to better market to and service these customers?

Data Description:

Data is of various customers of a bank with their credit limit, the total number of credit cards the customer has, and different channels through which customer has contacted the bank for any queries, different channels include visiting the bank, online and through a call centre.

Steps to follow:

1. Perform univariate analysis on the data to better understand the variables at your disposal and to get an idea about the no of clusters. Perform EDA, create visualizations to explore data. (**10 marks**)

2. Properly comment on the codes, provide explanations of the steps taken in the notebook and conclude your insights from the graphs. (**5 marks**)

3. Execute K-means clustering use elbow plot and analyse clusters using boxplot (**10 marks**)

4. Execute hierarchical clustering (with different linkages) with the help of dendrogram and cophenetic coeff. Analyse clusters formed using boxplot (**15 marks**)

5. Calculate average silhouette score for both methods. (**5 marks**)

6. Compare K-means clusters with Hierarchical clusters. (**5 marks**)

7. Analysis the clusters formed, tell us how is one cluster different from another and answer all the key questions. (**10 marks**)

Analysis

We have data related to customers of a bank. There are five features (columns) we use for this.

The columns :

Avg_Credit_Limit
Total_Credit_Cards
Total_visits_bank
Total_visits_online
Total_calls_made

Total records : 660. Small set. This shows hierarchical clustering should work better than KMeans.

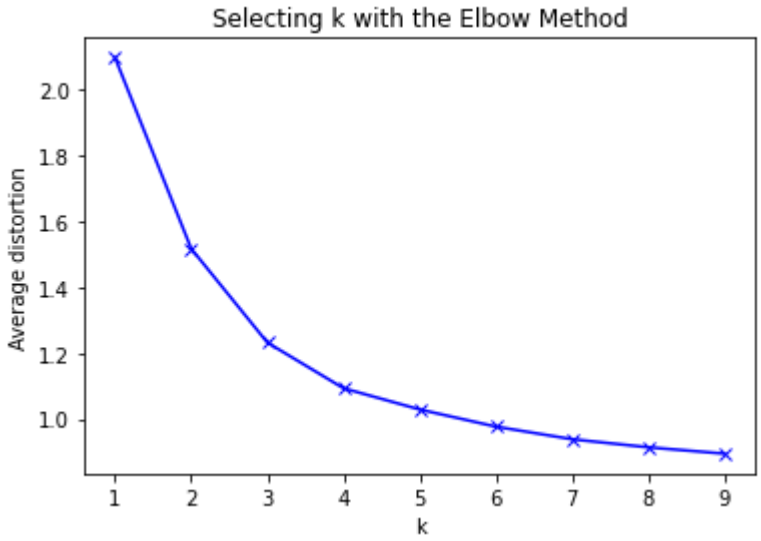
1. Conducted univariate analysis and provided the comments in the note book.
2. There were outliers in Avg_Credit_Limit (39) and Total_visits_online(37)
3. If we look at the distribution plots of individual variables, we can see 4 or 2 clusters. Also, the above two columns are skewed to right.
4. Also, there is no co-relation between any two variables(columns or features).
5. Tried clustering after fixing the outliers and without fixing them.
6. First listed results after fixing outliers.
7. Second listed results without removing outliers and as the scores are better for this, did cluster analysis and recommendations for both hierarchical complete linkage and KMeans clustering.

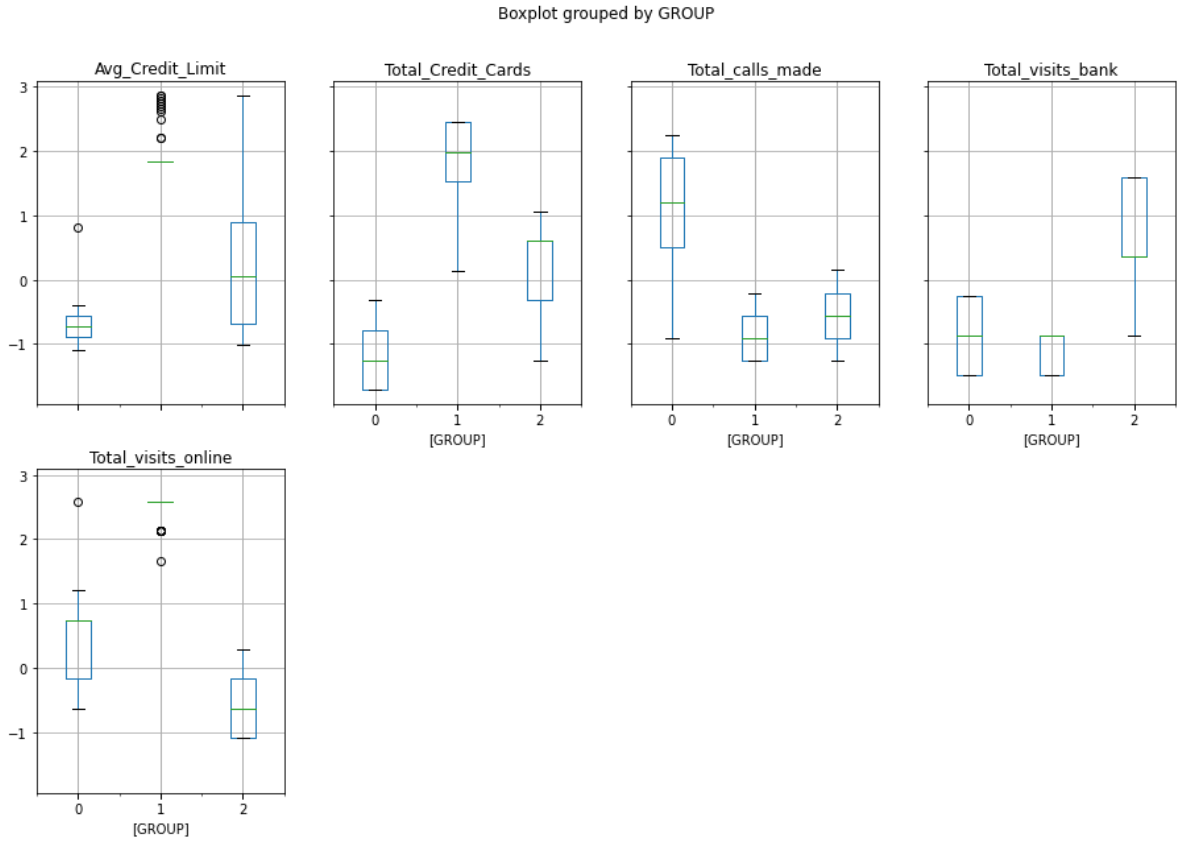
After fixing outliers

KMeans Clustering →

Ran KMeans with 4, 3 clusters as per the elbow method we got 4.
Got a score of 50 with three clusters.
The box plots below. Mostly there is no overlap between clusters.
But we have few elements in group 1 for Avg_Credit_Limit and Total_vi sits_online

Elbow method graph.





In [257]:
```
silhouette_score(df_labeled.drop('labels',axis=1),df_labeled['labels'] )   # We got a score of 50 with three clusters which is
# better than the previous 39.
```

Out[257]: 0.5047881723333618

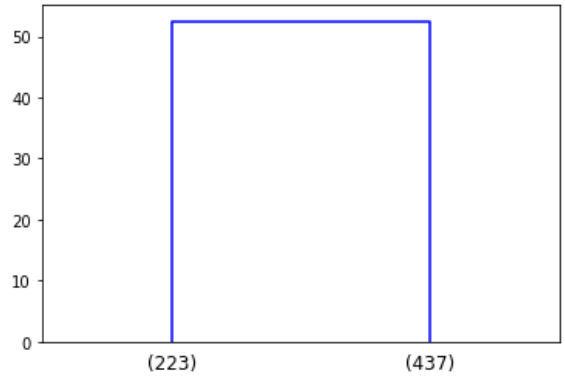
Hierarchical Clustering →

Did hierarchical clustering by using the following linkages. Ward, Complete, Average, Shortest,

Linkage Ward →

Dendogram

In [715]:
```
# Use truncate_mode='lastp' attribute in dendrogram function to arrive at dendrogram
dendrogram(
    Z,
    truncate_mode='lastp',  # show only the last p merged clusters
    p=2,  # show only the last p merged clusters
)
plt.show()
```

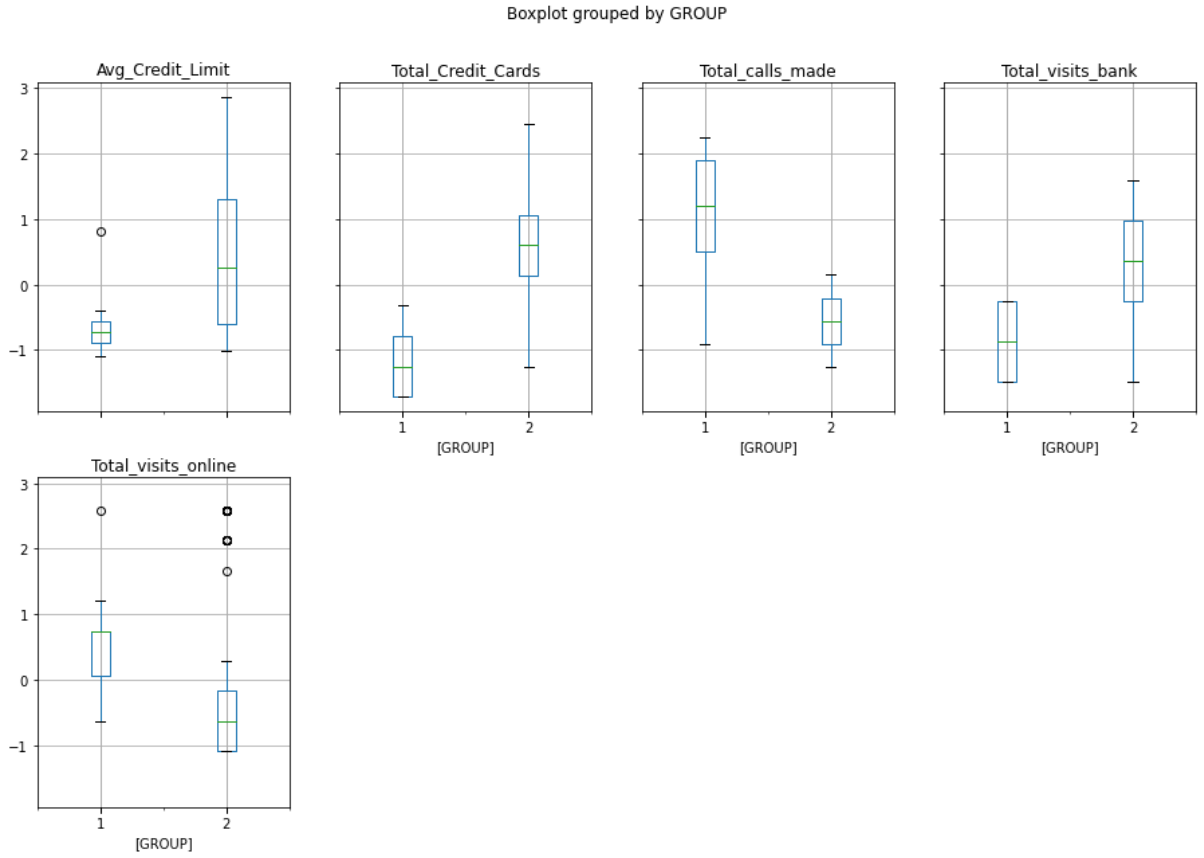


In [472]:
```
# Calculate Silhoutte Score for Ward linkage
silhouette_score(df3,clusters)
```

Out[472]: 0.4176330137256762

In [721]:
```
# cophenet index is a measure of the correlation between the distance of points in feature space and distance on dendrogram
# closer it is to 1, the better is the clustering

Z = linkage(df3, metric='euclidean', method='ward')
c, coph_dists = cophenet(Z , pdist(df3))

c
```

Out[721]: 0.7668112424099262

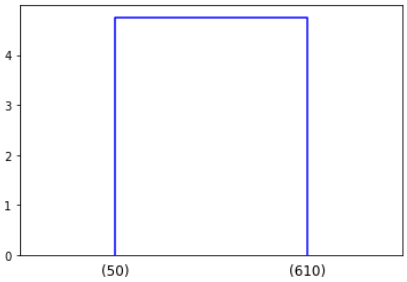


Clusters are not ovelapping and distribution seems to be fine.

Linkage Average →

Dendogram :

```
In [534]: # Use truncate_mode='lastp' attribute in dendrogram function to arrive at dendrogram
dendrogram(
    Z,
    truncate_mode='lastp',  # show only the last p merged clusters
    p=2,  # show only the last p merged clusters
)
plt.show()
```

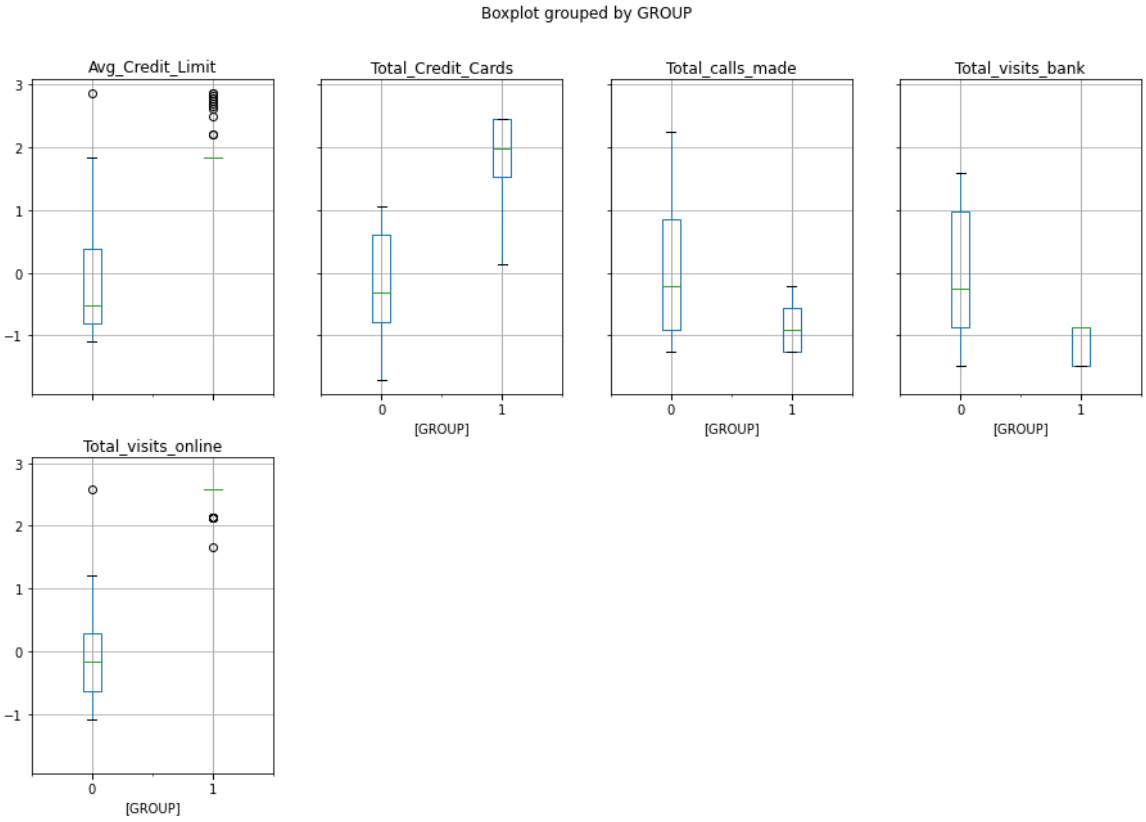



```
In [530]: # Calculate Avg Silhoutte Score
silhouette_score(df3,L)
```

```
Out[530]: 0.47280579123109273
```

In [538]:
```
# cophenet index is a measure of the correlation between the distance of points in feature space and distance on dendrogram
# closer it is to 1, the better is the clustering

Z = linkage(df3, metric='euclidean', method='average')
c, coph_dists = cophenet(Z , pdist(df3))

c
```

Out[538]: 0.8738818374092228



Clusters are not ovelapping and distribution seems to be fine. The cophenet is highest in my entire analysis .87, which is close to 1.

Linkage Complete →

In [658]:
```
# Calculate Avg Silhoutte Score
silhouette_score(df3,L)
```

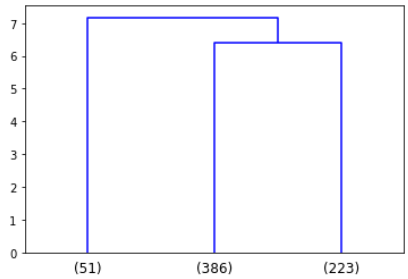
Out[658]: 0.5036210021628501

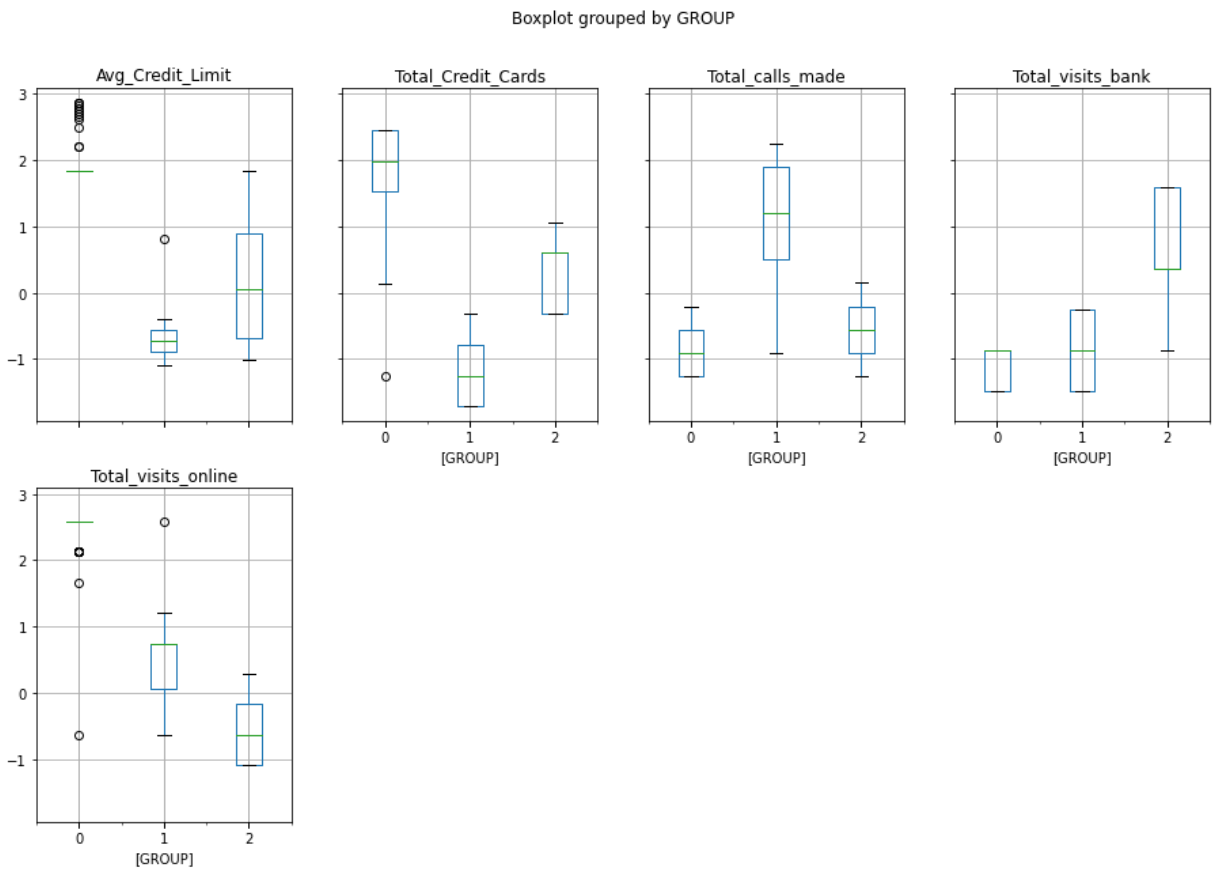
In [659]:
```
# cophenet index is a measure of the correlation between the distance of points in feature space and distance on dendrogram
# closer it is to 1, the better is the clustering

Z = linkage(df3, metric='euclidean', method='complete')
c, coph_dists = cophenet(Z , pdist(df3))

c
```

Out[659]: 0.8482180751274768

In [660]:
```
# Z = linkage(df3, 'complete', metric='euclidean')
```

In [661]:
```
# Use truncate_mode='lastp' attribute in dendrogram function to arrive at dendrogram
dendrogram(
    Z,
    truncate_mode='lastp',  # show only the last p merged clusters
    p=3,  # show only the last p merged clusters
)
plt.show()
```





Mostly there is no overlap between clusters. But we have few elements in group 0 for Avg_Credit_Limit and Total_vi sits_online

Linkage Shortest →

In [667]:
```
# Calculate Avg Silhoutte Score
silhouette_score(df3,L)
```

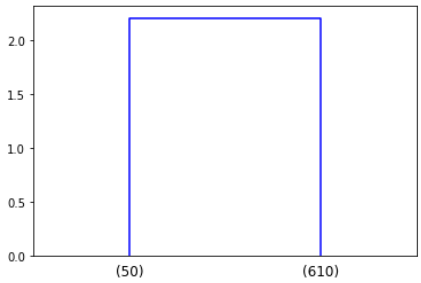
Out[667]: 0.47280579123109273

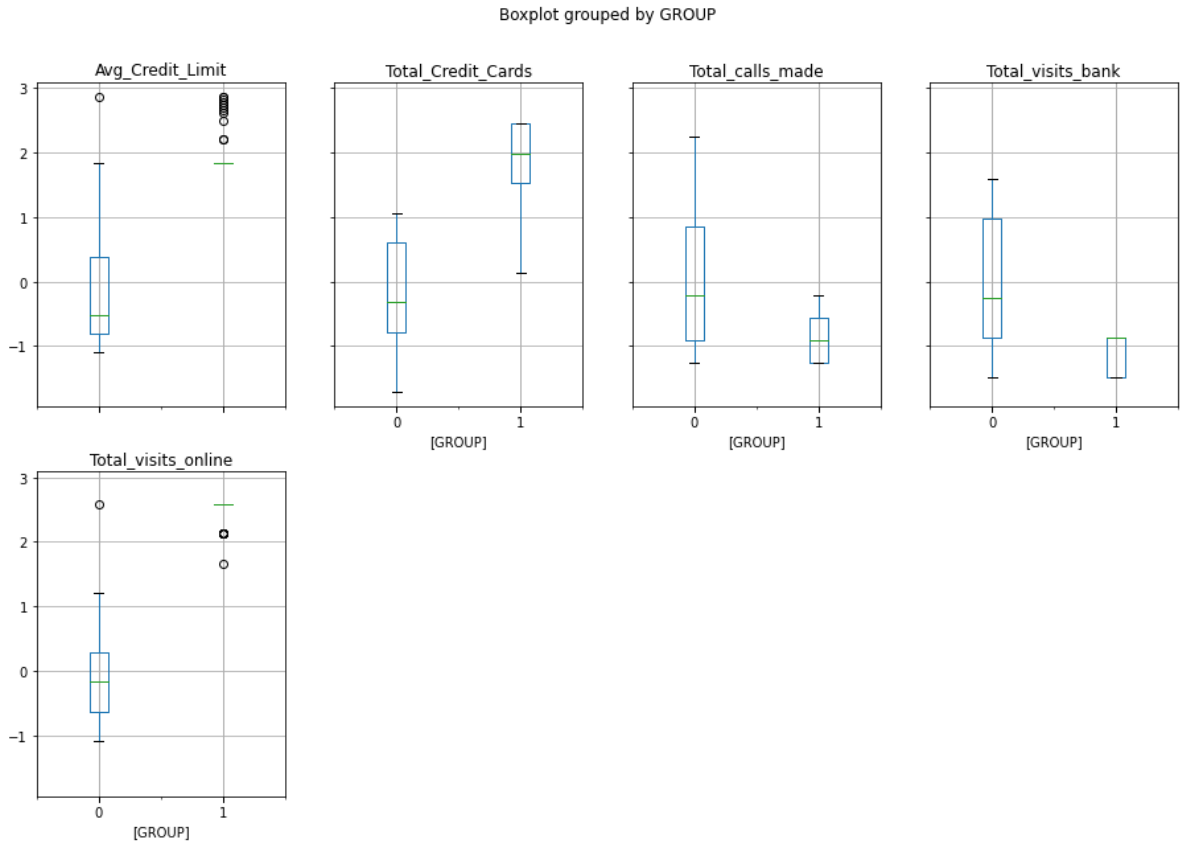
In [668]:
```
# cophenet index is a measure of the correlation between the distance of points in feature space and distance on dendrogram
# closer it is to 1, the better is the clustering

Z = linkage(df3, metric='euclidean', method='single')
c, coph_dists = cophenet(Z , pdist(df3))

c
```

Out[668]: 0.6042365834224761

In [669]:
```
#Z = linkage(df3, 'single', metric='euclidean')
```

In [670]:
```
# Use truncate_mode='lastp' attribute in dendrogram function to arrive at dendrogram
dendrogram(
    Z,
    truncate_mode='lastp',  # show only the last p merged clusters
    p=2,  # show only the last p merged clusters
)
plt.show()
```





Mostly there is no overlap between clusters.

But we have few elements in group 1 for Avg_Credit_Limit and Total_vi sits_online

Without removing outliers

Data set used as it is without modifying outliers as we got better scores for both Hierarchical and KMeans.

Hierarchical with linkage as Complete →

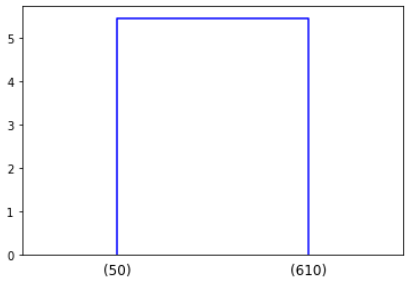
In [684]:
```
# cophenet index is a measure of the correlation between the distance of points in feature space and distance on dendrogram
# closer it is to 1, the better is the clustering
Z = linkage(df3, metric='euclidean', method='complete')
c, coph_dists = cophenet(Z , pdist(df_z_cp))

c
```

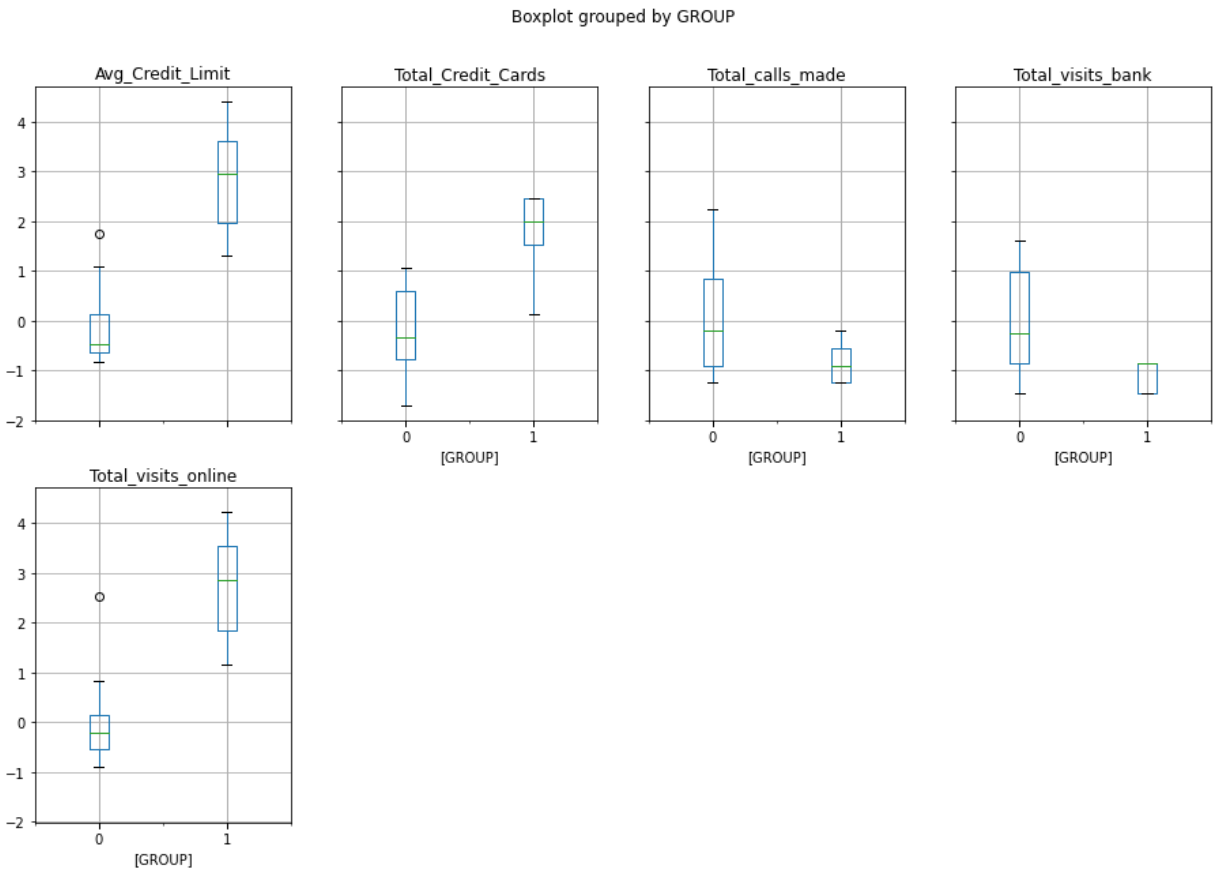
Out[684]: 0.8061661359199105

In [629]:
```
# Calculate Avg Silhoutte Score
silhouette_score(df_z_cp,L)
```

Out[629]: 0.5703183487340514

In [630]:
```
# Use truncate_mode='lastp' attribute in dendrogram function to arrive at dendrogram
dendrogram(
    Z,
    truncate_mode='lastp',  # show only the last p merged clusters
    p=2,  # show only the last p merged clusters
)
plt.show()
```

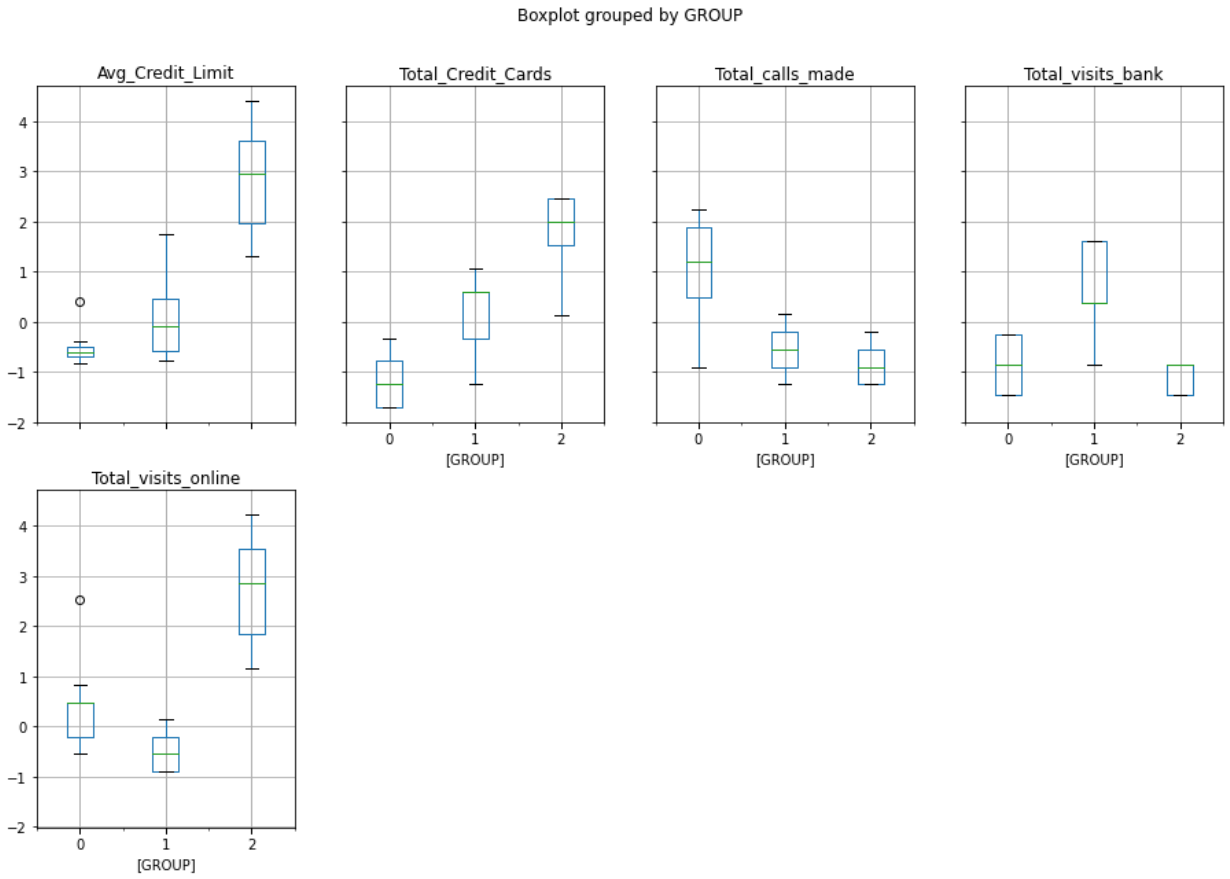



Boxplot grouped by GROUP
Avg_Credit_Limit
Total_Credit_Cards
Total_calls_made
Total_visits_bank
Total_visits_online
[GROUP]
0
1
−2
−1
0
1
2
3
4

KMeans →

```
In [704]: silhouette_score(df_labeled.drop('labels',axis=1),df_labeled['labels'] )   # We got a score of 50 with three clusters which is
          # better than the previous 39.
```

```
Out[704]: 0.5157182558881063
```



Cluster comparison between KMeans and Hierachical

Let's compare Hierarchical Linkage Complete, and KMeans clustering.
After analyze their clusters we can conclude as follows.

1. Even though the number of clusters in them one with 2 sets other with three sets, the results are very similar.
2. There are broadly two types of customers. Higher average credit limit and lower average credit limit.
3. These guys have a little different usage.
4. Higher credit limit guys are using online banking more.
5. They have more credit cards.
6. They visit bank less often and calls made were also fewer.
7. The lower average credit limit guys are coming to the bank more, less credit cards. Calls made are more and online usage less.

The above is true in both clustering algorithms, only difference is KMeans split the data into three clusters Vs hierarchical two clusters.

Hierarchical Complete Linkage cluster comparison →

Group	Feature	Minimum	Maximum
0	Avg_Credit_Limit	3000	100000
1		84000	200000
0	Total_Credit_Cards	1	7
1		5	10
0	Total_visits_bank	0	5
1		0	1
0	Total_visits_online	0	10
1		6	15
0	Total_calls_made	0	10
1		0	3

If you look at the clusters, it looks as follows.

1. For Avg_Credit_Limit, Total_Credit_Cards, and Total_visits_online the values used by the clusters are literally like lower range for lower cluster and upper range for upper cluster. They are almost not over lapping.
2. Total_visits_bank, Total_calls_made just reversed. What I mean by that is for Avg_Credit_Limit lower cluster these are high and for higher cluster these are low.
3. This can also be explained as follows.
4. People with higher credit limit, more number of credit cards are accessing online banking more and coming to the bank less and number of calls made are also less.

5. People with lower credit limit, less number of credit cards are accessing online banking less and coming to the bank more and number of calls made are also more.
6. The interaction with the call center is more for the second group. So may be they are more unhappy.

Recommendations to the bank :

1. Make the online banking more available by making it easier and intuitive. Also available on all devic es. Provide some training to the customers. Add more features to it.
2. This way interaction will be less, then customers may be more happy.
3. Also do a random survey on a selected few lower and upper cluster (credit limit) guys and find out i f coming to the bank, interacting with help desk (calls) or online banking is the issue and fix the issue. I doubt the last one is an issue but it may help us to get more insight so that what new features they want in online banking.
4. We should look at the lower cluster customers and market more credit cards and based on their Credit worthiness update their credit limit.
5. We should look at the upper credit limit guys to see if we can offer some higher rate CDs, low interest loans etc.

KMeans clustering comparison →

Group	Feature	Minimum	Maximum
0	Avg_Credit_Limit	3000	50000
1		5000	100000
2		84000	200000
0	Total_Credit_Cards	1	4
1		2	7
2		5	10
0	Total_visits_bank	0	2
1		1	5
2		0	1
0	Total_visits_online	1	10
1		0	3
2		6	15
0	Total_calls_made	1	10
1		0	4
2		0	3

Even though it got three clusters the results are very similar to hierarchical clustering results.

We can apply the same results …

If you look at the clusters, it looks as follows.

1 For Avg_Credit_Limit, Total_Credit_Cards, and Total_visits_online the values used by the clusters are literally like lower range for lower clusters and upper range for upper cluster. Here they are overlapping little bit.
2 Total_visits_bank, Total_calls_made just reversed to some extent but not fully as hierarchical.
3 This can also be explained as follows.
4 People with higher credit limit, more number of credit cards are accessing online banking more and coming to the bank less and number of calls made are also less.
5 People with lower credit limit, less number of credit cards are accessing online banking less and

coming to the bank more and number of calls made are also more.

6 The interaction with the call center is more for the second group. So may be they are more unhappy.

Recommendations to the bank :

1. Make the online banking more available by making it easier and intuitive. Also available on all devic es. Provide some training to the customers. Add more features to it.
2. This way interaction will be less, then customers may be more happy.
3. Also do a random survey on a selected few lower and upper cluster (credit limit) guys and find out i f coming to the bank, interacting with help desk (calls) or online banking is the issue and fix the issue. I doubt the last one is an issue but it may help us to get more insight so that we can add new features they want in online banking.
4. We should look at the lower cluster customers and market more credit cards and based on their credit worthiness update their credit limit.
5. We should look at the upper credit limit guys to see if we can offer some higher rate CDs, low interest loans etc.