Clustering Clean_Ads

Team:

Akhilesh Chauhan
Apoorv Purohit
Preyal Deep Chhabra
Rathi Sadhasivan
Renuka Prasad
Yureka M R

PGP-AIML June'23

18/11/2023

Digital Ads Data:

The ads24x7 is a Digital Marketing company which has now got seed funding of $10 Million. They are expanding their wings in Marketing Analytics. They collected data from their Marketing Intelligence team and now wants you (their newly appointed data analyst) to segment type of ads based on the features provided. Use Clustering procedure to segment ads into homogeneous groups.

The following three features are commonly used in digital marketing:

CPM = (Total Campaign Spend / Number of Impressions) * 1,000

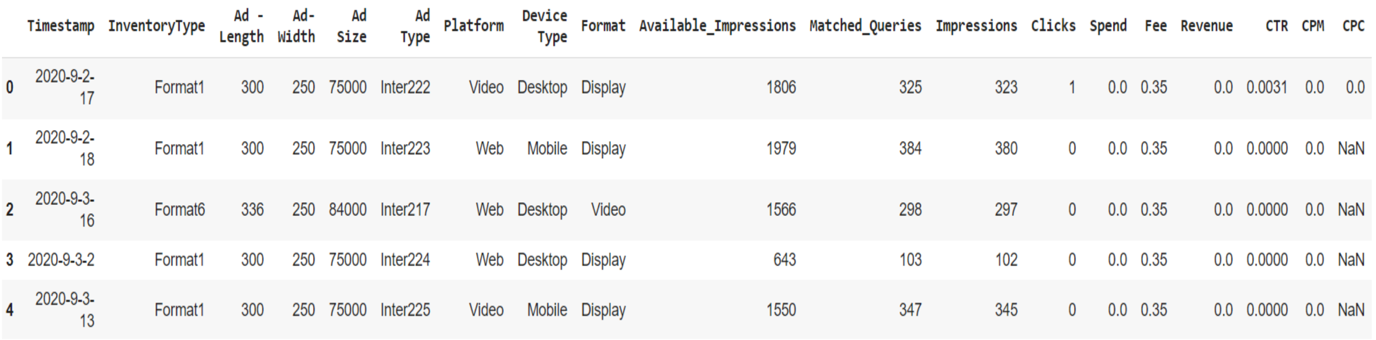
CPC = Total Cost (spend) / Number of Clicks

CTR = Total Measured Clicks / Total Measured Ad Impressions x 100

Data Description

The shared data has 25857 records and 19 columns.

Fig-1 Digital Ads Sample Data



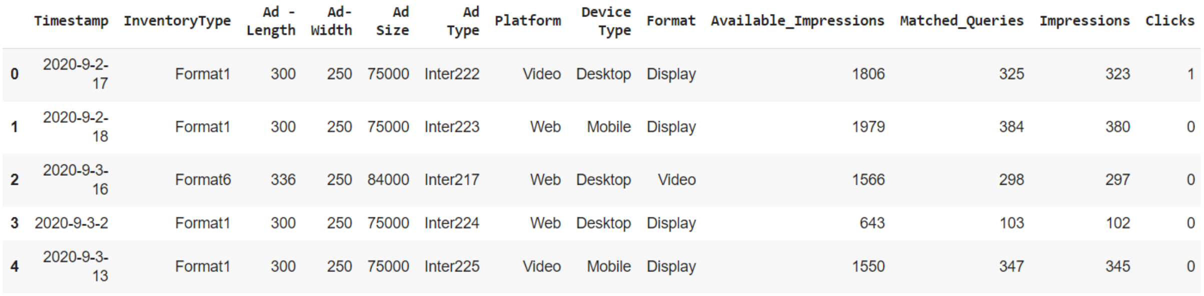
	Timestamp	InventoryType	Ad - Length	Ad- Width	Ad Size	Ad Type	Platform	Device Type	Format	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
0	2020-9-2-17	Format1	300	250	75000	Inter222	Video	Desktop	Display	1806	325	323	1	0.0	0.35	0.0	0.0031	0.0	0.0
1	2020-9-2-18	Format1	300	250	75000	Inter223	Web	Mobile	Display	1979	384	380	0	0.0	0.35	0.0	0.0000	0.0	NaN
2	2020-9-3-16	Format6	336	250	84000	Inter217	Web	Desktop	Video	1566	298	297	0	0.0	0.35	0.0	0.0000	0.0	NaN
3	2020-9-3-2	Format1	300	250	75000	Inter224	Web	Desktop	Display	643	103	102	0	0.0	0.35	0.0	0.0000	0.0	NaN
4	2020-9-3-13	Format1	300	250	75000	Inter225	Video	Mobile	Display	1550	347	345	0	0.0	0.35	0.0	0.0000	0.0	NaN

Question 1: Part 1 - Clustering: Read the data and perform basic analysis such as printing a few rows (head and tail), info, data summary, null values, duplicate values, etc.

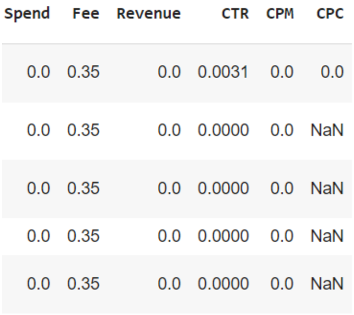
The basic analysis of the provided data has been covered in the supporting document.

We notice that out of the 19 columns, 6 columns are of object data type, 7 columns are of int data type and 6 columns are of float data type.

A snippet of the first 5 rows (head) are as follows:

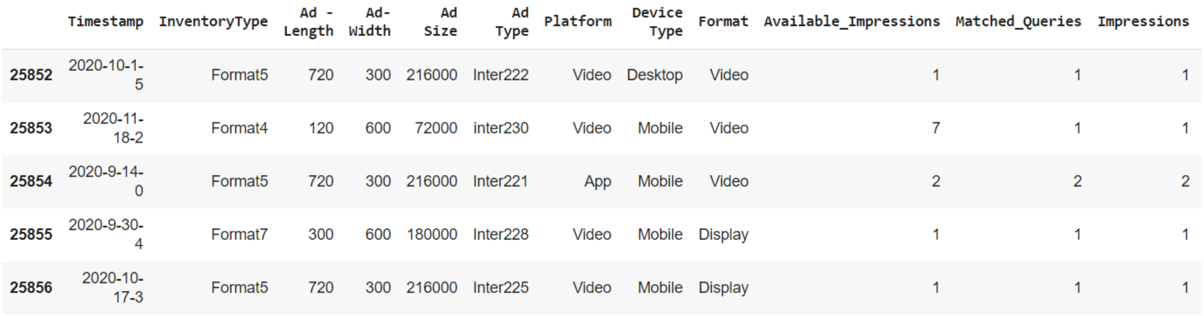


	Timestamp	InventoryType	Ad - Length	Ad- Width	Ad Size	Ad Type	Platform	Device Type	Format	Available_Impressions	Matched_Queries	Impressions	Clicks
0	2020-9-2-17	Format1	300	250	75000	Inter222	Video	Desktop	Display	1806	325	323	1
1	2020-9-2-18	Format1	300	250	75000	Inter223	Web	Mobile	Display	1979	384	380	0
2	2020-9-3-16	Format6	336	250	84000	Inter217	Web	Desktop	Video	1566	298	297	0
3	2020-9-3-2	Format1	300	250	75000	Inter224	Web	Desktop	Display	643	103	102	0
4	2020-9-3-13	Format1	300	250	75000	Inter225	Video	Mobile	Display	1550	347	345	0

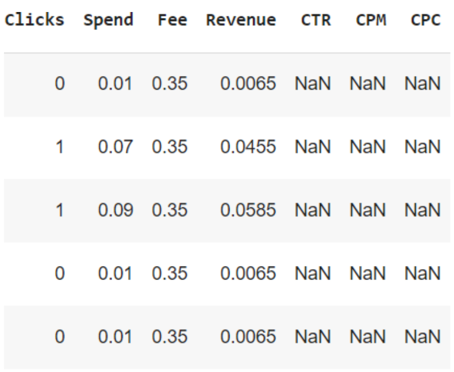


Spend	Fee	Revenue	CTR	CPM	CPC
0.0	0.35	0.0	0.0031	0.0	0.0
0.0	0.35	0.0	0.0000	0.0	NaN
0.0	0.35	0.0	0.0000	0.0	NaN
0.0	0.35	0.0	0.0000	0.0	NaN
0.0	0.35	0.0	0.0000	0.0	NaN

A snippet of the last 5 rows (tail) are as follows:

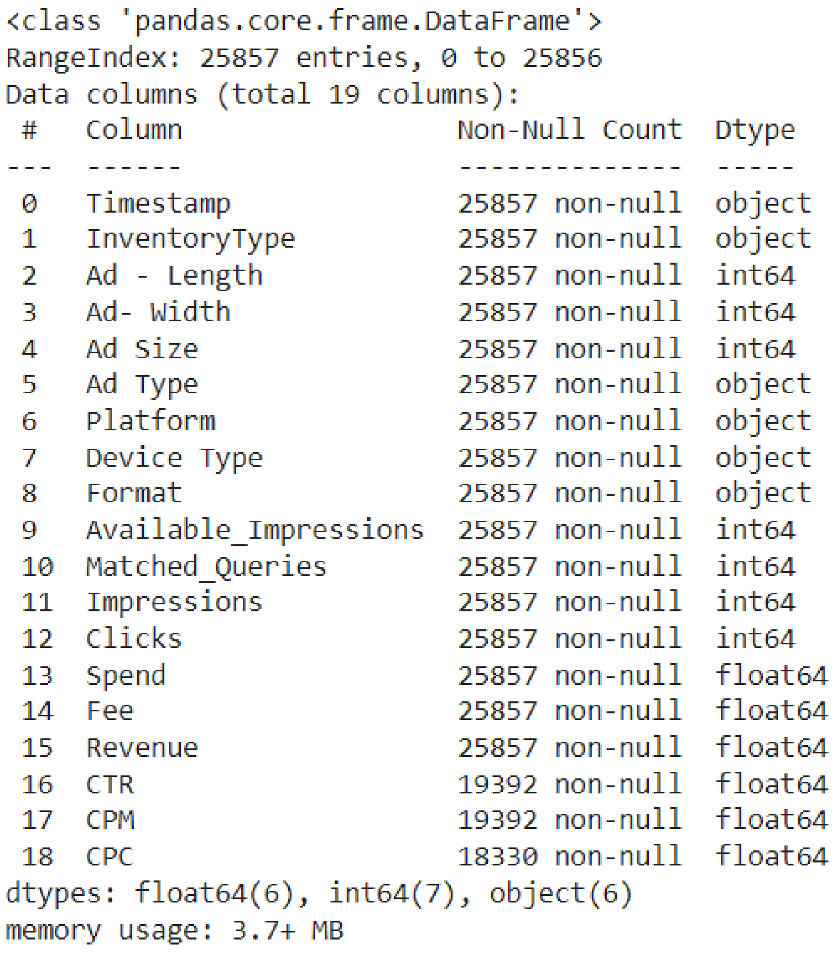


	Timestamp	InventoryType	Ad - Length	Ad- Width	Ad Size	Ad Type	Platform	Device Type	Format	Available_Impressions	Matched_Queries	Impressions
25852	2020-10-1-5	Format5	720	300	216000	Inter222	Video	Desktop	Video	1	1	1
25853	2020-11-18-2	Format4	120	600	72000	inter230	Video	Mobile	Video	7	1	1
25854	2020-9-14-0	Format5	720	300	216000	Inter221	App	Mobile	Video	2	2	2
25855	2020-9-30-4	Format7	300	600	180000	Inter228	Video	Mobile	Display	1	1	1
25856	2020-10-17-3	Format5	720	300	216000	Inter225	Video	Mobile	Display	1	1	1



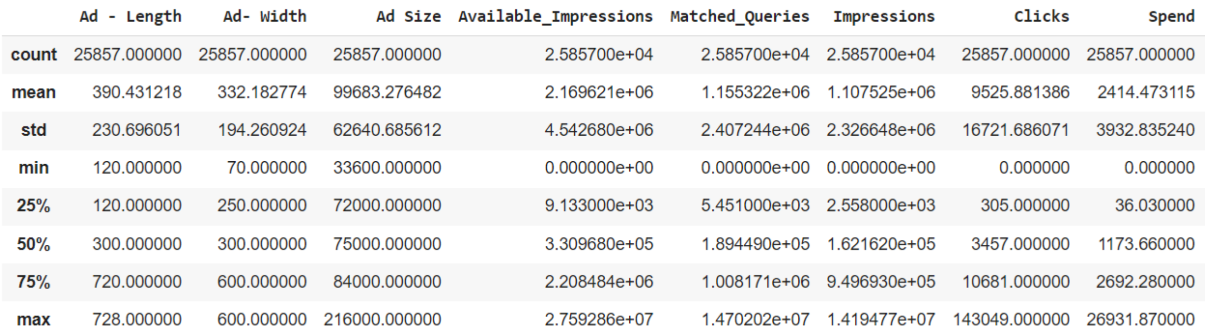
Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
0	0.01	0.35	0.0065	NaN	NaN	NaN
1	0.07	0.35	0.0455	NaN	NaN	NaN
1	0.09	0.35	0.0585	NaN	NaN	NaN
0	0.01	0.35	0.0065	NaN	NaN	NaN
0	0.01	0.35	0.0065	NaN	NaN	NaN

The details of the provided data set are as follows:

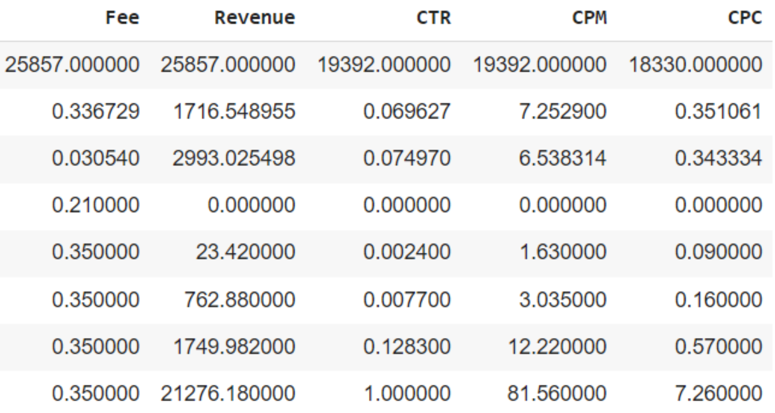


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25857 entries, 0 to 25856
Data columns (total 19 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Timestamp              25857 non-null  object 
 1   InventoryType          25857 non-null  object 
 2   Ad - Length            25857 non-null  int64  
 3   Ad- Width              25857 non-null  int64  
 4   Ad Size                25857 non-null  int64  
 5   Ad Type                25857 non-null  object 
 6   Platform               25857 non-null  object 
 7   Device Type            25857 non-null  object 
 8   Format                 25857 non-null  object 
 9   Available_Impressions  25857 non-null  int64  
 10  Matched_Queries        25857 non-null  int64  
 11  Impressions            25857 non-null  int64  
 12  Clicks                 25857 non-null  int64  
 13  Spend                  25857 non-null  float64
 14  Fee                    25857 non-null  float64
 15  Revenue                25857 non-null  float64
 16  CTR                    19392 non-null  float64
 17  CPM                    19392 non-null  float64
 18  CPC                    18330 non-null  float64
dtypes: float64(6), int64(7), object(6)
memory usage: 3.7+ MB
```

Data summary of the dataset is as follows:

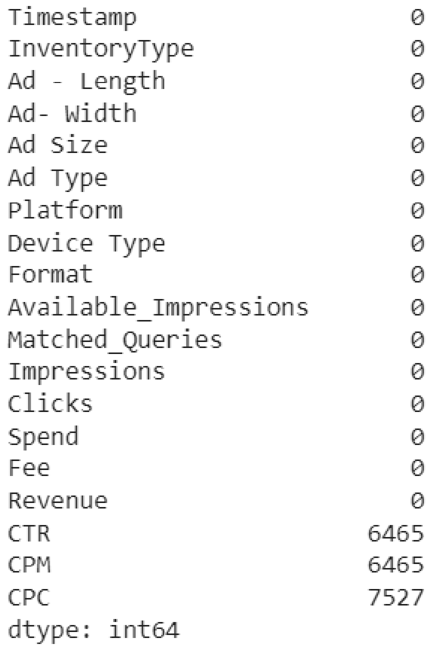


	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend
count	25857.000000	25857.000000	25857.000000	2.585700e+04	2.585700e+04	2.585700e+04	25857.000000	25857.000000
mean	390.431218	332.182774	99683.276482	2.169621e+06	1.155322e+06	1.107525e+06	9525.881386	2414.473115
std	230.696051	194.260924	62640.685612	4.542680e+06	2.407244e+06	2.326648e+06	16721.686071	3932.835240
min	120.000000	70.000000	33600.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000
25%	120.000000	250.000000	72000.000000	9.133000e+03	5.451000e+03	2.558000e+03	305.000000	36.030000
50%	300.000000	300.000000	75000.000000	3.309680e+05	1.894490e+05	1.621620e+05	3457.000000	1173.660000
75%	720.000000	600.000000	84000.000000	2.208484e+06	1.008171e+06	9.496930e+05	10681.000000	2692.280000
max	728.000000	600.000000	216000.000000	2.759286e+07	1.470202e+07	1.419477e+07	143049.000000	26931.870000



Fee	Revenue	CTR	CPM	CPC
25857.000000	25857.000000	19392.000000	19392.000000	18330.000000
0.336729	1716.548955	0.069627	7.252900	0.351061
0.030540	2993.025498	0.074970	6.538314	0.343334
0.210000	0.000000	0.000000	0.000000	0.000000
0.350000	23.420000	0.002400	1.630000	0.090000
0.350000	762.880000	0.007700	3.035000	0.160000
0.350000	1749.982000	0.128300	12.220000	0.570000
0.350000	21276.180000	1.000000	81.560000	7.260000

Also, there are null values for the columns namely - CTR, CPM and CPC.



```
Timestamp                  0
InventoryType              0
Ad - Length                0
Ad- Width                  0
Ad Size                    0
Ad Type                    0
Platform                   0
Device Type                0
Format                     0
Available_Impressions      0
Matched_Queries            0
Impressions                0
Clicks                     0
Spend                      0
Fee                        0
Revenue                    0
CTR                     6465
CPM                     6465
CPC                     7527
dtype: int64
```

Duplicate values:

There are no duplicate rows found in the dataset.

Question 2: Part 1 - Clustering: Treat missing values in CPC, CTR and CPM using the formula given.

The details of the last 3 columns, which have null values namely - CTR, CPM and CPC are explained as follows:

CPM stands for "cost per 1000 impressions".
CPC stands for "Cost-per-click". Cost-per-click (CPC) bidding means that you pay for each click on your ads.
CTR: stands for "Click through rate". CTR is the number of clicks that your ad receives divided by the number of times your ad is shown.

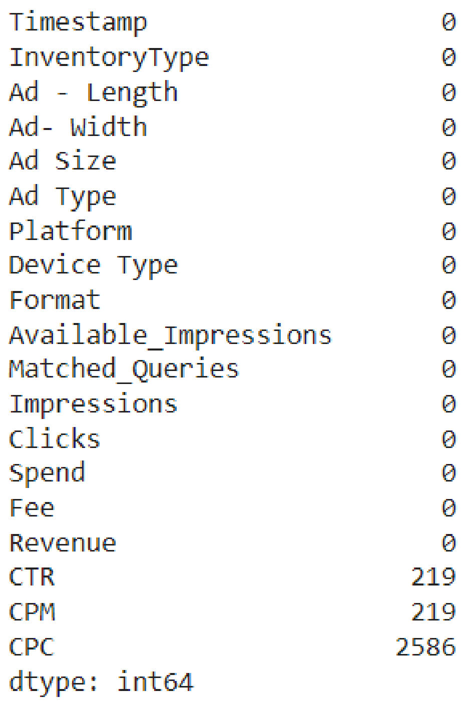
The CTR, CPM and CPC columns are derived using the below formulas:

CPM = (Spend / Impressions) x 1000
CPC = (Spend / Clicks)
CTR = (Clicks / Impressions) x 100

Hence, these null values need to be treated.

First, we define three user defined functions, to calculate the values of CPM, CPC and CTR respectively. Then, we broadcast the output as per the formula using the inbuilt lambda function.

After this treatment is completed, we further check if there are any more null values in these columns. We notice that the last 3 columns still have some null values. The reason for that is there are a few rows where the numerator as well as the denominator has the value 0.

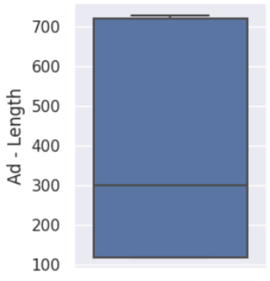


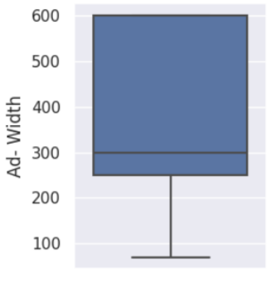
```
Timestamp                  0
InventoryType              0
Ad - Length                0
Ad- Width                  0
Ad Size                    0
Ad Type                    0
Platform                   0
Device Type                0
Format                     0
Available_Impressions      0
Matched_Queries            0
Impressions                0
Clicks                     0
Spend                      0
Fee                        0
Revenue                    0
CTR                      219
CPM                      219
CPC                     2586
dtype: int64
```

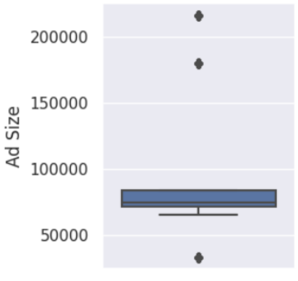
For such cases, we update the specific value to 0.0, instead of NaN. Post this, there are no null values in the dataset.

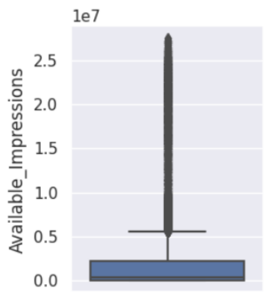
Question 3: Part 1 - Clustering: Check if there are any outliers. Do you think treating outliers is necessary for K-Means clustering? Based on your judgement decide whether to treat outliers and if yes, which method to employ. (As an analyst your judgement may be different from another analyst).

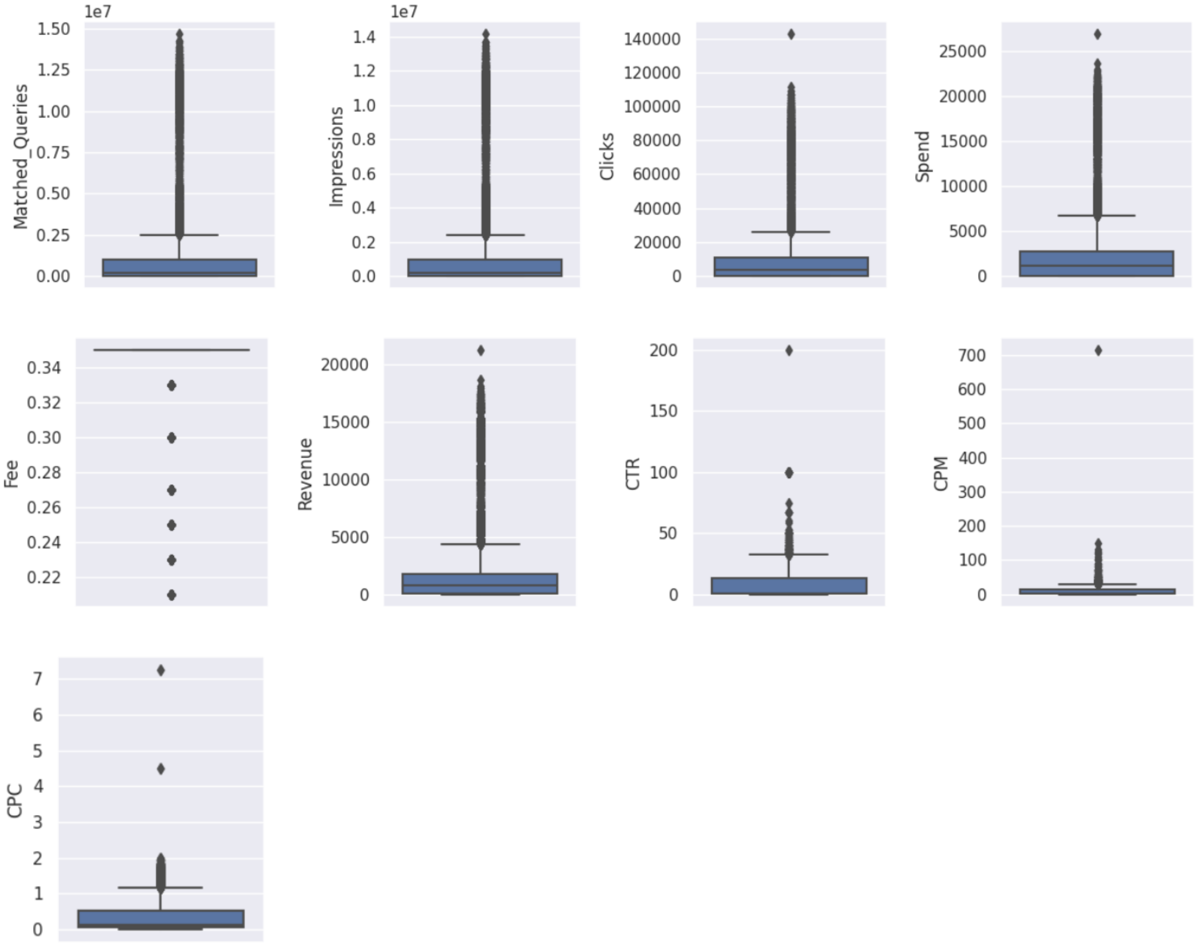
We are using box plots to check for outliers in the data. For that, we are considering all the numeric columns.











From the above box plots, we have the following observations:

- Except Ad-Length and Ad-Width, all the numerical columns have outliers.
- Outliers affect the clustering algorithm heavily.
- We can treat the outliers by either deleting them or imputing them.

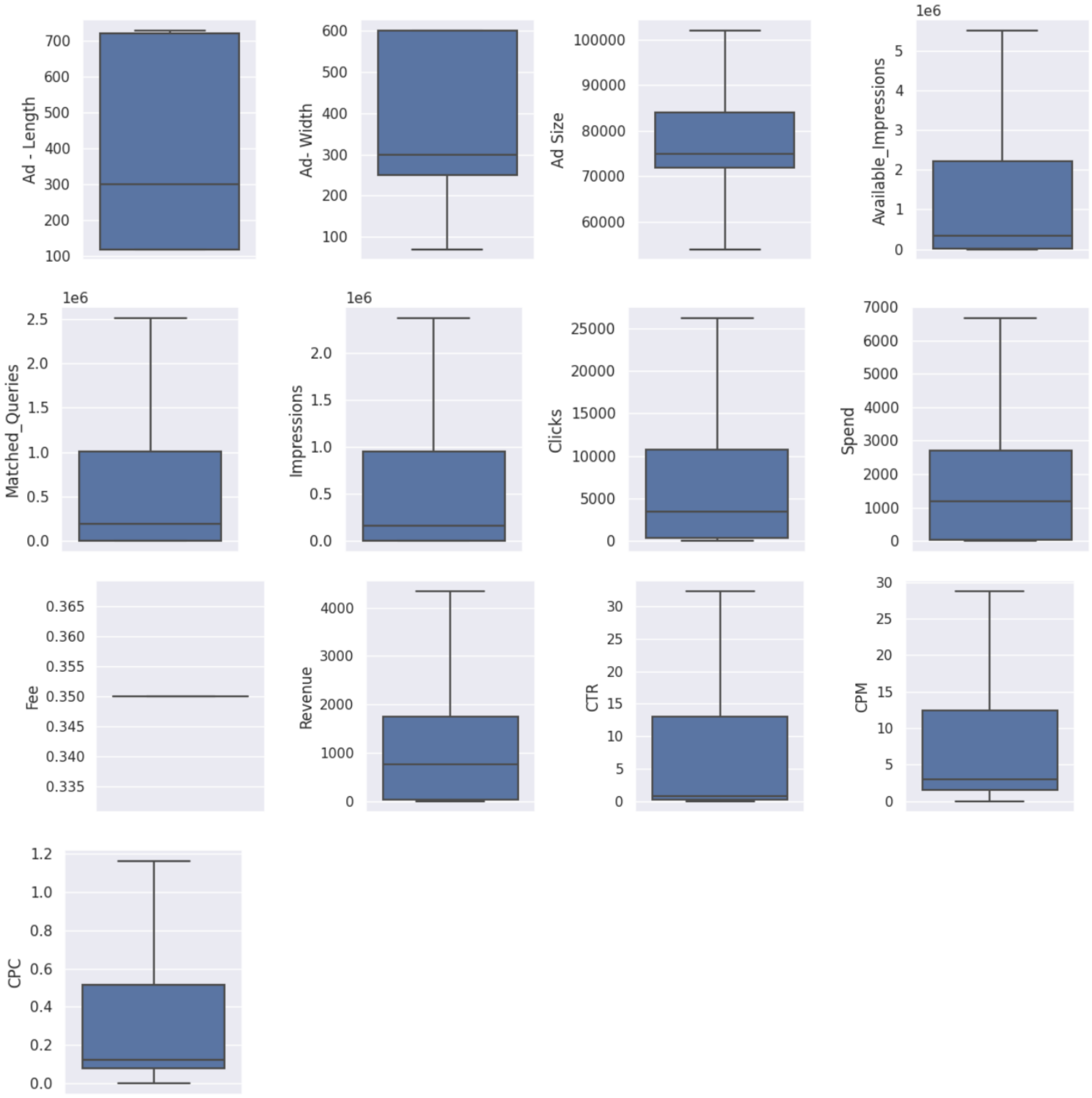
K-Means clustering algorithm is most sensitive to outliers, as it uses the mean of cluster data points to find the cluster center. Hence, the outliers need to be treated here.

Further, we choose to treat the outliers using the boxplot method.
We shall calculate the lower range and upper range using the IQR and then shall bring the data which is outside the ranges, closer to the median.

The high level algorithm used for treating outliers here is as follows:

a. Identify the first quartile (Q1) and the third quartile (Q3).
b. Calculate IQR = Q3 - Q1
c. Calculate the lower range using the formula : Q3 = Q1 - (1.5*IQR)
d. Calculate the upper range using the formula : Q3 = Q1 + (1.5*IQR)
e. Replace the values below LR and above UR to LR and UR respectively.

Once this treatment is completed, we again re-check if there are any outliers and then we notice that there are no outliers.

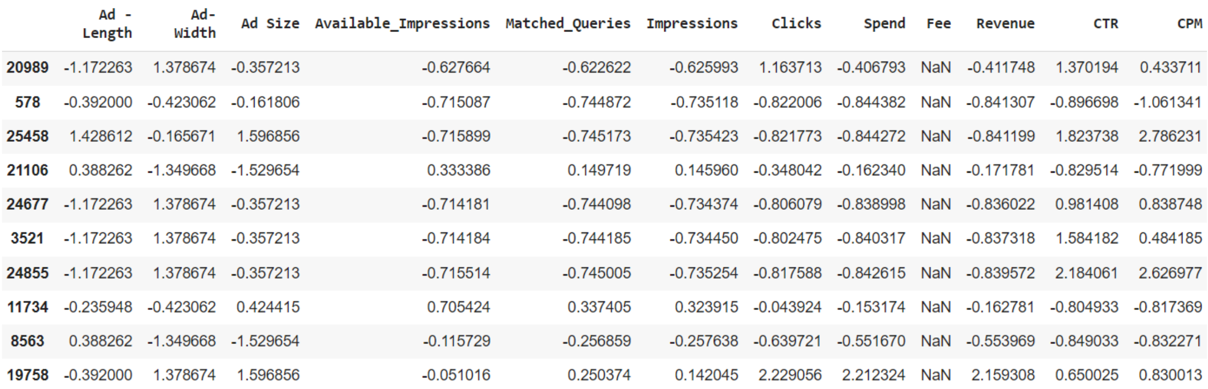


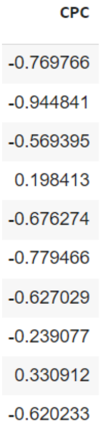
Question 4 : Part 1 - Clustering: Perform z-score scaling and discuss how it affects the speed of the algorithm.

Z-score is a variation of scaling that represents the number of standard deviations away from the mean.
We can use the apply function to calculate the z-score of individual values by column.

Data with different magnitudes, especially higher magnitude affects distance based algorithms. We need to bring the data on the same scale. Therefore, we shall be using the z score scaling.

After performing the z-score scaling, the snippet of the sample dataset is as follows:





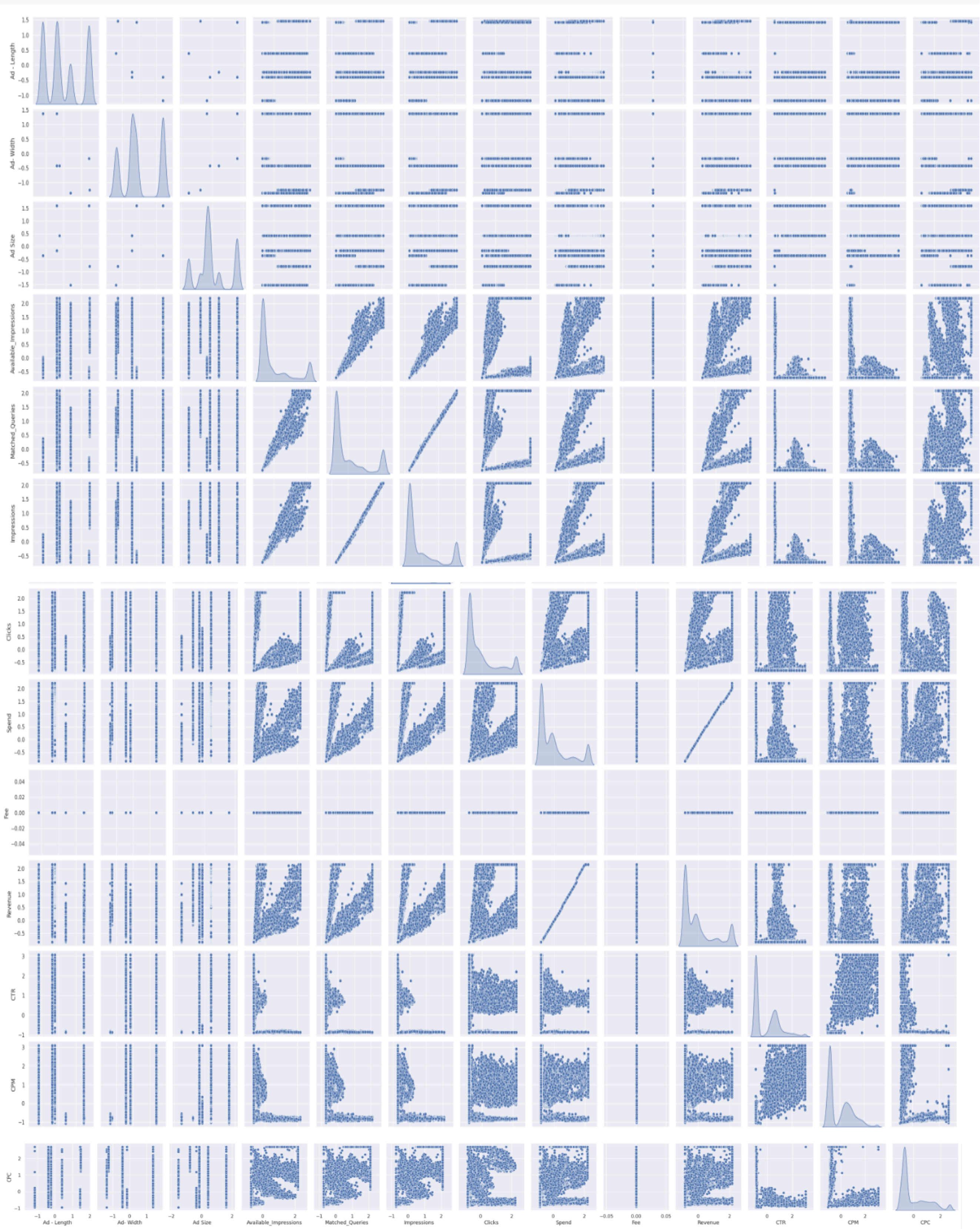
	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
20989	-1.172263	1.378674	-0.357213	-0.627664	-0.622622	-0.625993	1.163713	-0.406793	NaN	-0.411748	1.370194	0.433711	-0.769766
578	-0.392000	-0.423062	-0.161806	-0.715087	-0.744872	-0.735118	-0.822006	-0.844382	NaN	-0.841307	-0.896698	-1.061341	-0.944841
25458	1.428612	-0.165671	1.596856	-0.715899	-0.745173	-0.735423	-0.821773	-0.844272	NaN	-0.841199	1.823738	2.786231	-0.569395
21106	0.388262	-1.349668	-1.529654	0.333386	0.149719	0.145960	-0.348042	-0.162340	NaN	-0.171781	-0.829514	-0.771999	0.198413
24677	-1.172263	1.378674	-0.357213	-0.714181	-0.744098	-0.734374	-0.806079	-0.838998	NaN	-0.836022	0.981408	0.838748	-0.676274
3521	-1.172263	1.378674	-0.357213	-0.714184	-0.744185	-0.734450	-0.802475	-0.840317	NaN	-0.837318	1.584182	0.484185	-0.779466
24855	-1.172263	1.378674	-0.357213	-0.715514	-0.745005	-0.735254	-0.817588	-0.842615	NaN	-0.839572	2.184061	2.626977	-0.627029
11734	-0.235948	-0.423062	0.424415	0.705424	0.337405	0.323915	-0.043924	-0.153174	NaN	-0.162781	-0.804933	-0.817369	-0.239077
8563	0.388262	-1.349668	-1.529654	-0.115729	-0.256859	-0.257638	-0.639721	-0.551670	NaN	-0.553969	-0.849033	-0.832271	0.330912
19758	-0.392000	1.378674	1.596856	-0.051016	0.250374	0.142045	2.229056	2.212324	NaN	2.159308	0.650025	0.830013	-0.620233

We have the following observation, after z-score scaling has been applied.

Scaling has converted the values in the ‘Fee’ column into NaN. This is because, post the outlier treatment (explained earlier), every value in the ‘Fee’ column had converted into value 0.35, as that was the most frequent value then and the remaining values were outliers. The IQR of this column was 0.
We can either choose to remove this column from calculation or convert all the NaNs into 0s.

Here, we choose to convert the NaNs in the ‘Fee’ column to 0s.

Illustration of pairplot on the scaled data is as follows:



A few observations from the above pair plot is as follows:

- The diagonal of the pairplot gives us a brief idea as to how many clusters could be ideal here.
- If we observe from the diagonal element of Spend:
  - We can see around 4 to 5 peaks roughly in the data.
  - If there were categorical variables, we might get different KDE plots superimposed on one another.
- Similarly, Ad-Size is giving an idea of roughly around 5 clusters.

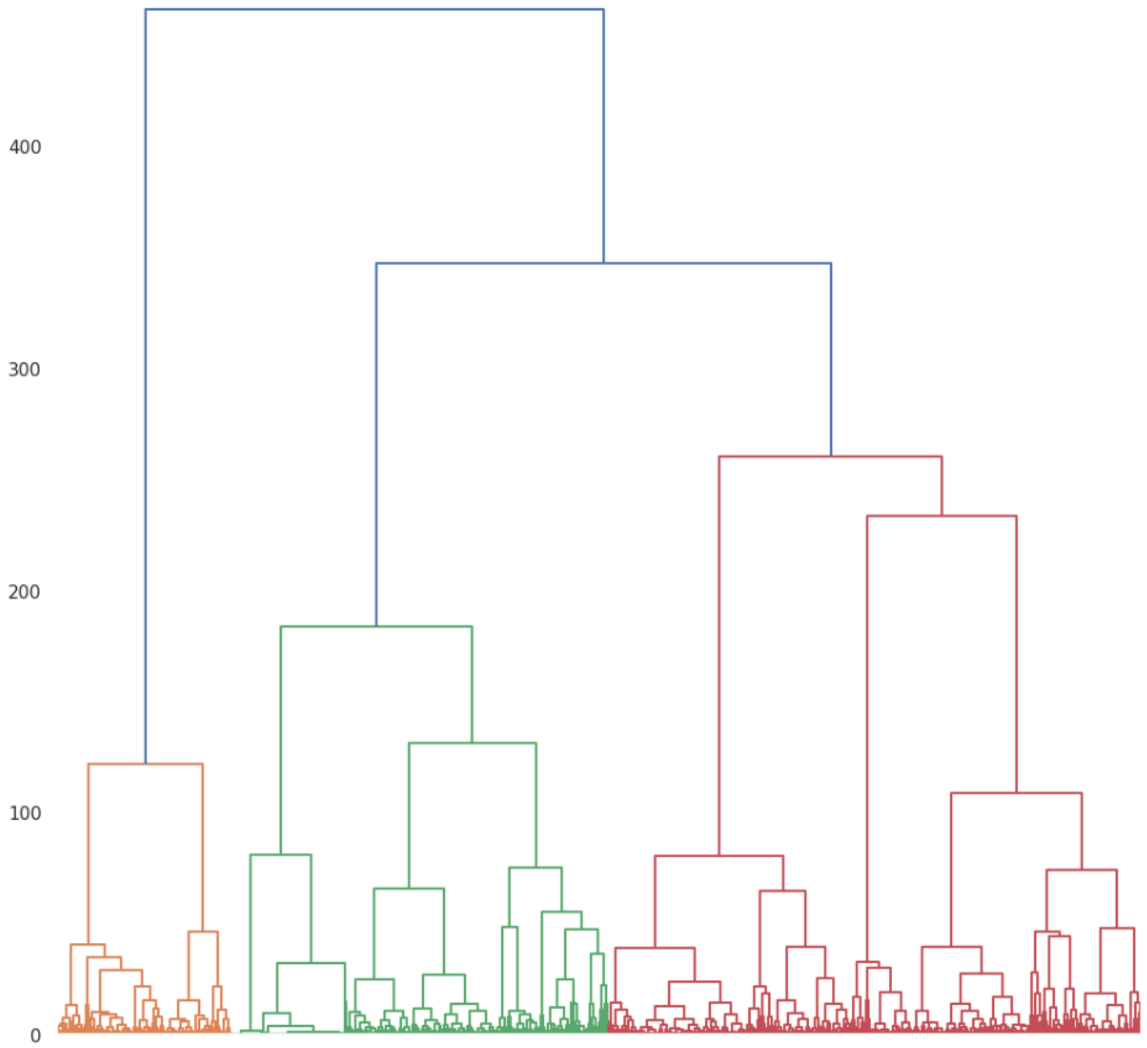
Question 5: Part 1 - Clustering: Perform Hierarchical by constructing a Dendrogram using WARD and Euclidean distance.

Hierarchical clustering is a popular method for grouping objects. It creates groups so that objects within a group are similar to each other and different from objects in other groups. Clusters are visually represented in a hierarchical tree called a dendrogram.
A Dendrogram is a tree-like diagram used to visualise the relationship among clusters. More the distance of the vertical lines in the dendrogram, the more the distance between those clusters. The key to interpreting a dendrogram is to concentrate on the height at which any two objects are joined together.

The scipy.cluster package equips us with tools needed for hierarchical clustering and dendrogram plotting. Thus, has to be imported into the environment.
We need to import dendrogram, linkage, fcluster from the package scipy.cluster.hierarchy.

Create a hierarchical binary cluster tree using linkage .Then, plot the dendrogram with a vertical orientation, using the default color threshold. Return handles to the lines so you can change the dendrogram line widths. A linkage function is an essential prerequisite for hierarchical cluster analysis . Its value is a measure of the “distance” between two groups of objects (i.e. between two clusters). Algorithms for hierarchical clustering normally differ by the linkage function used. The fcluster() method forms flat clusters from the hierarchical clustering. This hierarchical clustering is defined by the given linkage matrix, identifying a link between clustered classes.

The dendrogram is as follows:



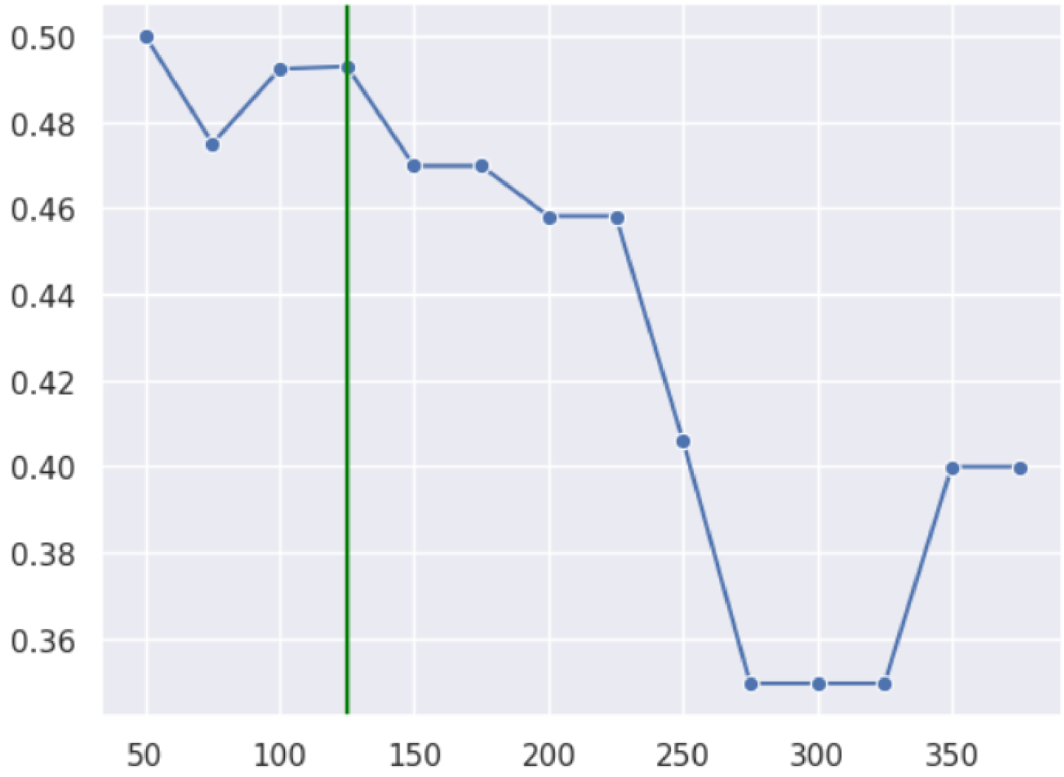
Question 7: Part 1 - Clustering: Print silhouette scores for up to 10 clusters and identify optimum number of clusters.

Silhouette Score is a metric to evaluate the performance of a clustering algorithm. It uses compactness of individual clusters(intra cluster distance) and separation amongst clusters (inter cluster distance) to measure an overall representative score of how well our clustering algorithm has performed.

The silhouette score would always lie between -1 to 1. 1 represents better clustering.

Here, we are finding the silhouette score for distances starting from 50 to 400 with step 25. Do refer to the supporting document for details.

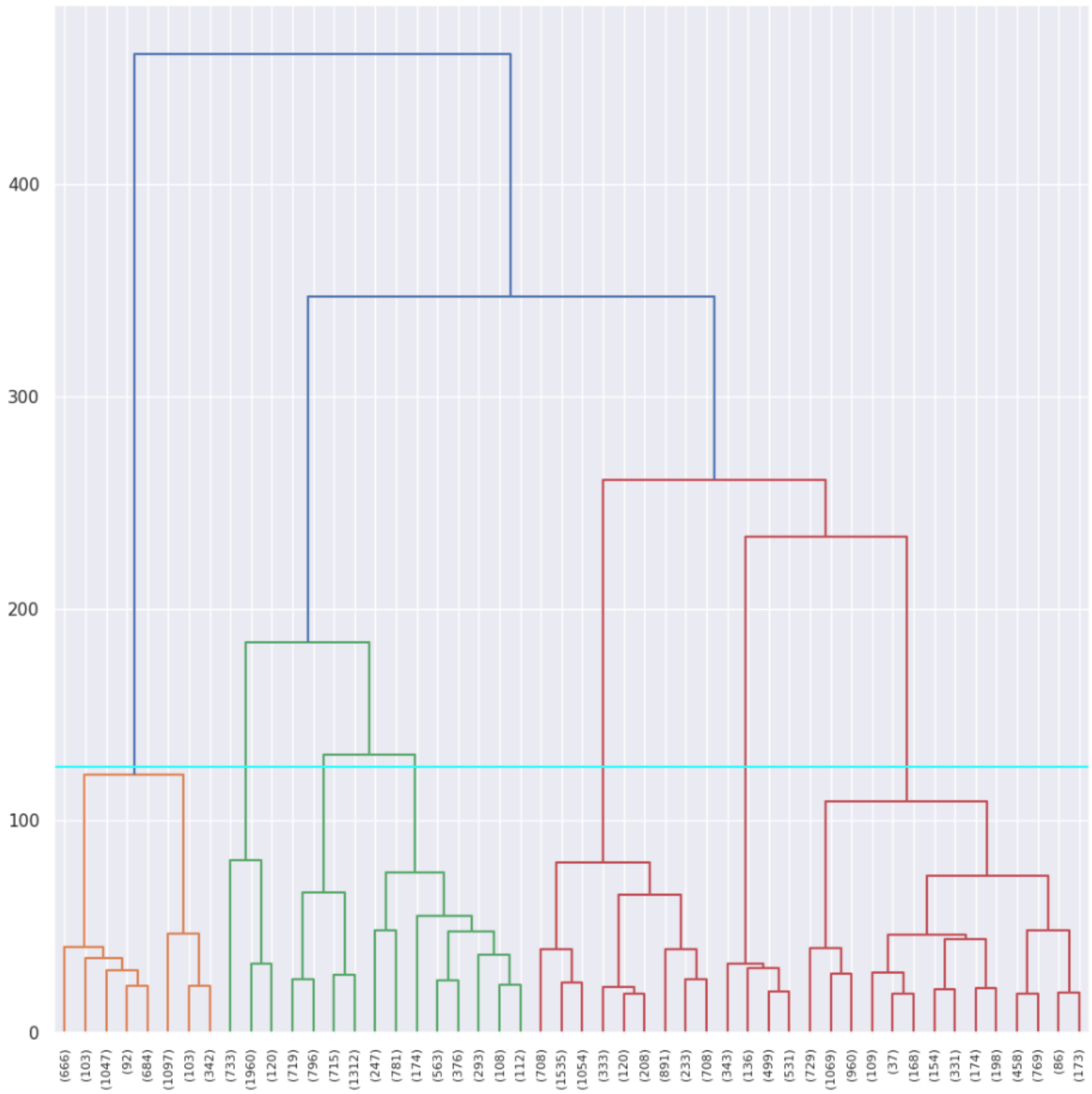
When we plot the graph, we get the following:



From the graph, we notice that the silhouette score for distance line 125 is 0.49. We shall consider the distance 125 and not 50 here, because 50 is a smaller value and it can give a bigger number of clusters. Further, as we decrease the value of distances, we can get more clusters, which can be hard to analyse.

From the below diagram, we see that the distance line 125 is cutting through 7 vertical lines, which indicates that 7 clusters are provided by the Hierarchical Clustering technique, which have a decent silhouette score of 0.49.

Clustering Clean_Ads



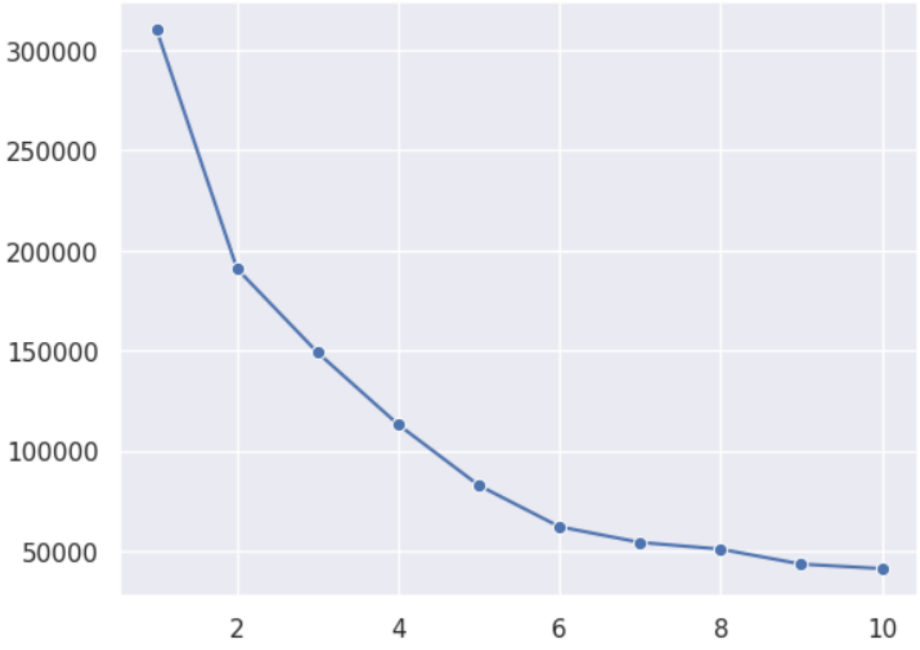
Question 6 : Part 1 - Clustering: Make Elbow plot (up to n=10) and identify optimum number of clusters for k-means algorithm.

As we know in the k-means clustering algorithm, we randomly initialise k clusters and we iteratively adjust these k clusters till these k-centroids reach an equilibrium state. However, the main thing we do before initialising these clusters is to determine how many clusters we have to use.

For determining K(numbers of clusters), we use the Elbow method. The Elbow Method is a technique that we use to determine the number of centroids(k) to use in a k-means clustering algorithm. In this method to determine the k-value, we continuously iterate for k=1 to k=n (Here n is the hyperparameter that we choose as per our requirement). For every value of k, we calculate the within-cluster sum of squares (WCSS) value. WCSS - It is defined as the sum of square distances between the centroids and each point.

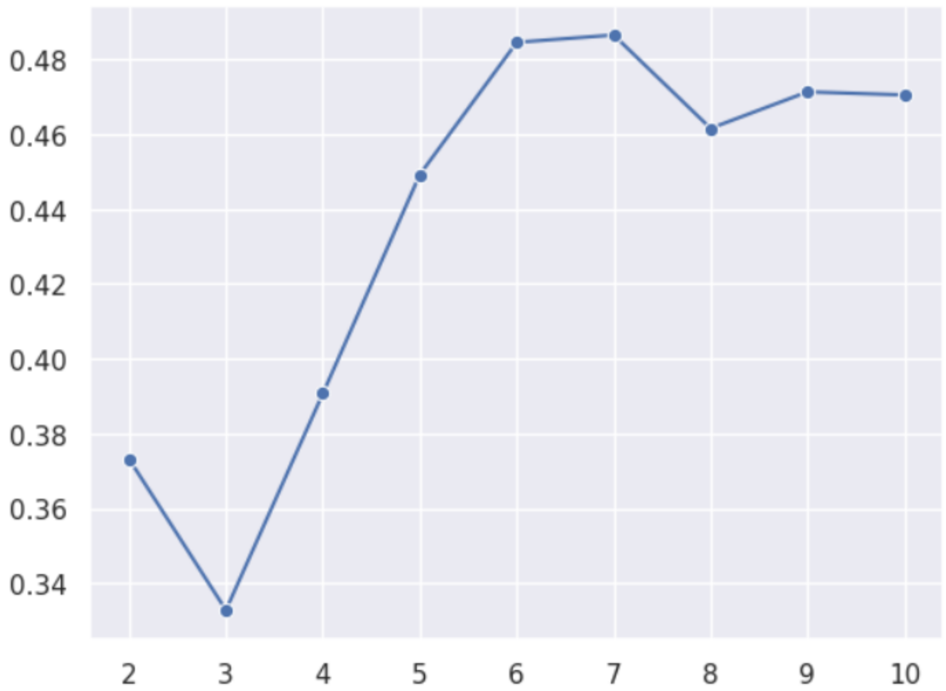
Now, for determining the best number of clusters(k) we plot a graph of k versus their WCSS value. Surprisingly, the graph looks like an elbow. Also, when k=1 the WCSS has the highest value but with increasing k value, WCSS value starts to decrease. We choose that value of k from where the graph starts to look like a straight line

Here, we shall find out the Within Sum of Squares for each k clusters and the silhouette scores of them starting from 2.
Then, we plot a line plot with the number of clusters (k) on the x-axis and WCSS on the y-axis.



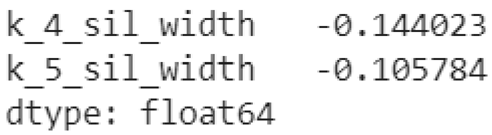
From the above graph, we notice that the number of clusters is coming out to be either 4 or 5.

Let us confirm this using another python function KneeLocator from kneed module. For this function to work, do install the kneed package, using pip install.



According to the KneeLocator, the number of clusters is 5. But, we have the silhouette score of 5 clusters better than 4 clusters.
We do not look at the silhouette scores of 8, 9 or 10 clusters, because when we keep increasing them, the scores shall only improve but more clusters would make it harder to analyse data.

Further, let us find out the silhouette samples of both 4 and 5 clusters and look at their minimum values. Do refer to the supporting document for more details.

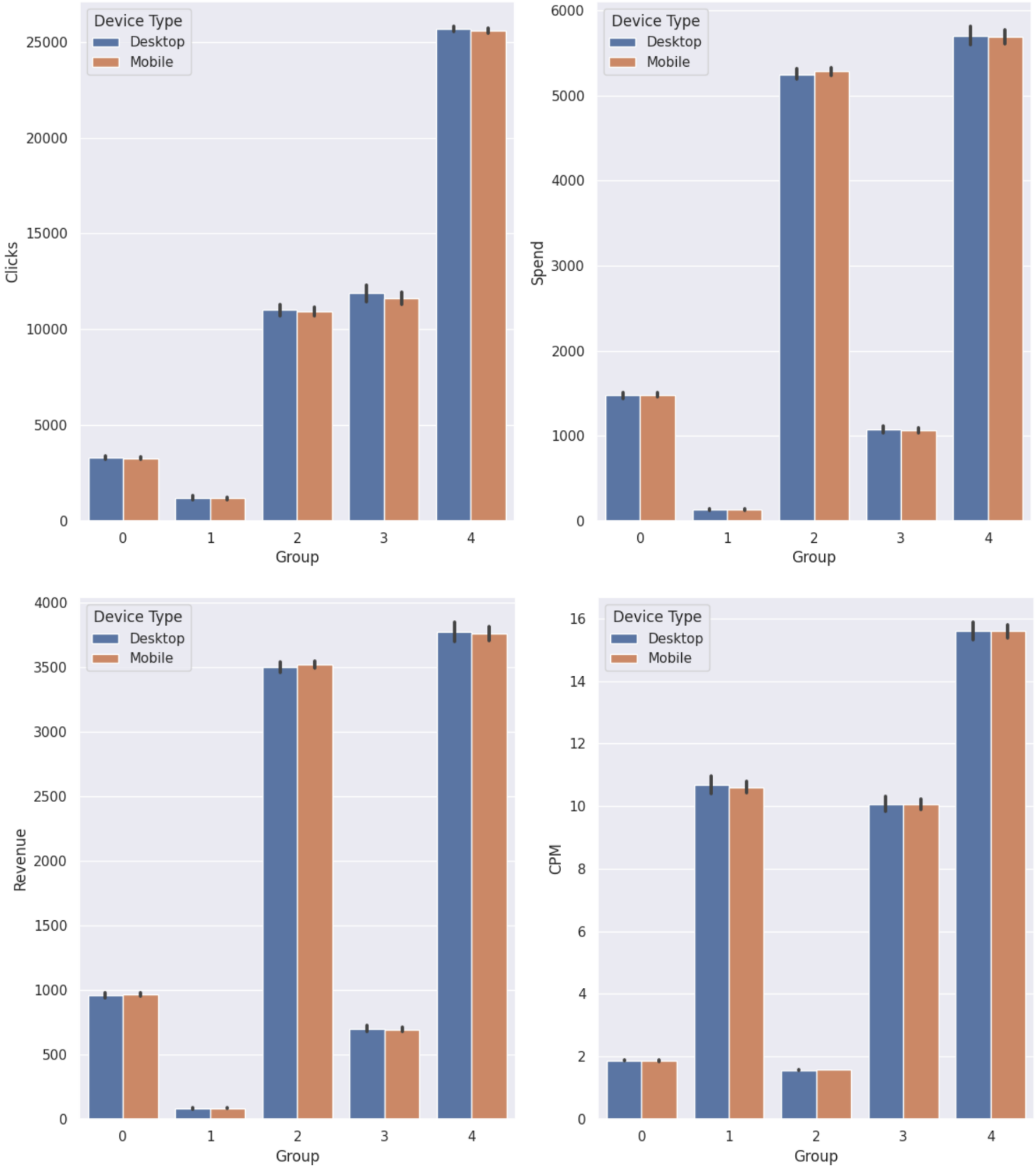


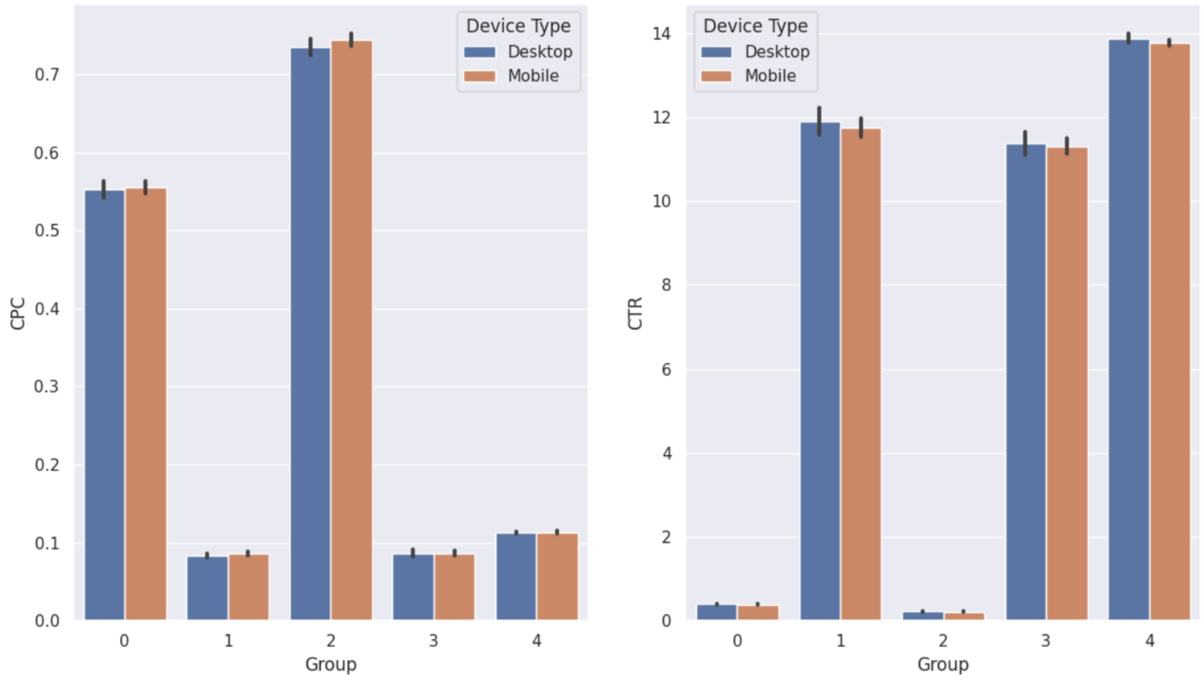
```
k_4_sil_width   -0.144023
k_5_sil_width   -0.105784
dtype: float64
```

The minimum silhouette width for k=5 clusters is smaller than k=4. Although both are negative which means there are some values for which the mapping is not ideal with their clusters. But, k=5 is slightly better than k=4.

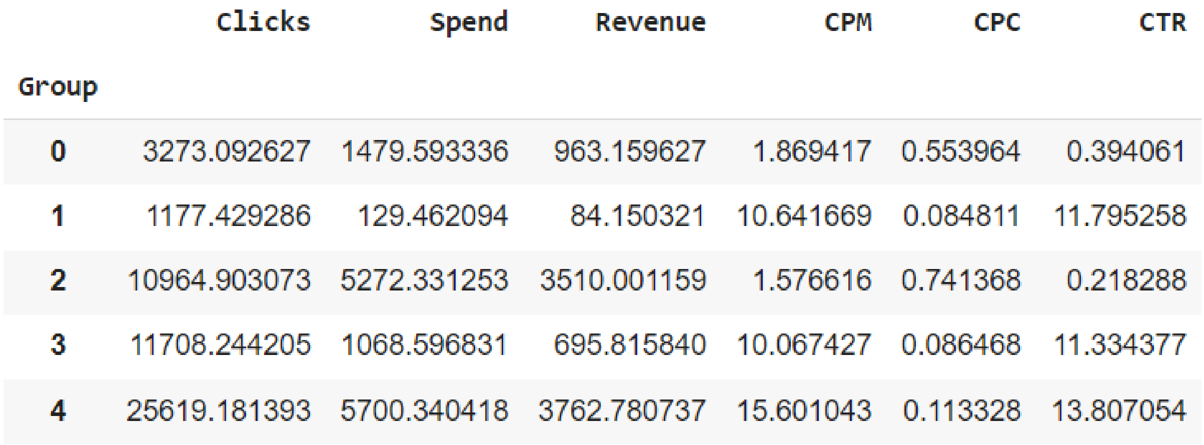
Question 8 - Part 1 - Clustering: Profile the ads based on optimum number of clusters using silhouette score and your domain understanding [Hint: Group the data by clusters and take sum or mean to identify trends in Clicks, spend, revenue, CPM, CTR, & CPC based on Device Type. Make bar plots].

The following columns are considered - 'Clicks', 'Spend', 'Revenue', 'CPM', 'CPC', 'CTR' and a bar plot is plotted by grouping the data by clusters and taking the sum or mean to identify the trends in the specific columns based on Device Type.





The mean values of each of the specific columns for each group is shown below:



Group	Clicks	Spend	Revenue	CPM	CPC	CTR
0	3273.092627	1479.593336	963.159627	1.869417	0.553964	0.394061
1	1177.429286	129.462094	84.150321	10.641669	0.084811	11.795258
2	10964.903073	5272.331253	3510.001159	1.576616	0.741368	0.218288
3	11708.244205	1068.596831	695.815840	10.067427	0.086468	11.334377
4	25619.181393	5700.340418	3762.780737	15.601043	0.113328	13.807054

Question 9 - Part 1 - Clustering: Conclude the project by providing a summary of your learnings.

The following are the conclusions:

a. Group 4 has the maximum clicks both on desktop as well as mobile.
   Group 1 has the least clicks both on desktop as well as mobile.

b. Group 4 has spent the maximum money both on desktop as well as mobile.
   Group 1 has spent the least money both on desktop as well as mobile.

c. Group 4 has the maximum revenue both on desktop as well as mobile. Group 1 has the minimum revenue both on desktop as well as mobile.

d. Group 4 has the maximum value for CPM on desktop as well as mobile. Group 2 has the minimum value for CPM on desktop as well as mobile.

e. Group 2 has the maximum value for CPC on desktop as well as mobile. Group 1 has the minimum value for CPC on desktop as well as mobile.

f. Group 4 has the maximum value for CTR on desktop as well as mobile. Group 2 has the minimum value for CTR on desktop as well as mobile.

g. There is no difference in the various trends across the specific columns between the 2 device types namely - Desktop and Mobile.