# CLUSTERING

## Project Report

# CLUSTERING:

## Introduction:

Clustering is a technique in machine learning that involves grouping similar data points together. In this analysis, we explore clustering using the KMeans algorithm and hierarchical clustering. The dataset used is from the file **Ads_Data.xlsx.**

# Data Description:

Timestamp: Date and time of the advertisement.

InventoryType: Ad inventory type (Format 1 to 7).

Ad - Length: Length dimension of the ad.

Ad - Width: Width dimension of the ad.

Ad Size: Overall size (Length * Width) of the ad.

Ad Type: Type of the ad.

Platform: Display platform (Web, Video, App).

Device Type: Type of supporting device.

Format: Ad format.

Available Impressions: Frequency of ad display.

Matched Queries: Search queries generating clicks.

Impressions: Count of ad impressions.

Clicks: Count of user clicks.

Spend: Money spent on ads.

Fee: Percentage of advertising fees.

Revenue: Income from the ad.

CTR (Click Through Rate): Clicks per impression percentage.

CPM (Cost Per Mille): Cost per 1000 impressions.

CPC (Cost Per Click): Cost per click.

**STEPS:**

## Step 1: Data Loading and Overview:

Loaded the dataset using pandas and displayed the first and last few rows to understand the data structure.

Ads_data dataset is loaded into the Dataframe.

Data Frame printing rows with Head (Prints top 5 rows) function as below :

```
# Display first few rows
data.head()
```

| | Timestamp | InventoryType | Ad - Length | Ad- Width | Ad Size | Ad Type | Platform | Device Type | Format | Available_Impressions | Matched_Queries | Impressions | Clicks | Sp |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2020-9-2-17 | Format1 | 300 | 250 | 75000 | Inter222 | Video | Desktop | Display | 1806 | 325 | 323 | 1 | |
| 1 | 2020-9-2-10 | Format1 | 300 | 250 | 75000 | Inter227 | App | Mobile | Video | 1780 | 285 | 285 | 1 | |
| 2 | 2020-9-1-22 | Format1 | 300 | 250 | 75000 | Inter222 | Video | Desktop | Display | 2727 | 356 | 355 | 1 | |
| 3 | 2020-9-3-20 | Format1 | 300 | 250 | 75000 | Inter228 | Video | Mobile | Video | 2430 | 497 | 495 | 1 | |
| 4 | 2020-9-4-15 | Format1 | 300 | 250 | 75000 | Inter217 | Web | Desktop | Video | 1218 | 242 | 242 | 1 | |

Data Frame printing rows with Tail (Prints last 5 rows) function as below :

```
# Display last few rows
data.tail()
```

| | Timestamp | InventoryType | Ad - Length | Ad- Width | Ad Size | Ad Type | Platform | Device Type | Format | Available_Impressions | Matched_Queries | Impressions | Click |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 23061 | 2020-9-13-7 | Format5 | 720 | 300 | 216000 | Inter220 | Web | Mobile | Video | 1 | 1 | 1 | |
| 23062 | 2020-11-2-7 | Format5 | 720 | 300 | 216000 | Inter224 | Web | Desktop | Video | 3 | 2 | 2 | |
| 23063 | 2020-9-14-22 | Format5 | 720 | 300 | 216000 | Inter218 | App | Mobile | Video | 2 | 1 | 1 | |
| 23064 | 2020-11-18-2 | Format4 | 120 | 600 | 72000 | inter230 | Video | Mobile | Video | 7 | 1 | 1 | |
| 23065 | 2020-9-14-0 | Format5 | 720 | 300 | 216000 | Inter221 | App | Mobile | Video | 2 | 2 | 2 | |

Data frame shape as below :

(23066, 19) There are 23066 rows, and 19 columns into the Data.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23066 entries, 0 to 23065
Data columns (total 19 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Timestamp              23066 non-null  object
 1   InventoryType          23066 non-null  object
 2   Ad - Length            23066 non-null  int64
 3   Ad- Width              23066 non-null  int64
 4   Ad Size                23066 non-null  int64
 5   Ad Type                23066 non-null  object
 6   Platform               23066 non-null  object
 7   Device Type            23066 non-null  object
 8   Format                 23066 non-null  object
 9   Available_Impressions  23066 non-null  int64
 10  Matched_Queries        23066 non-null  int64
 11  Impressions            23066 non-null  int64
 12  Clicks                 23066 non-null  int64
 13  Spend                  23066 non-null  float64
 14  Fee                    23066 non-null  float64
 15  Revenue                23066 non-null  float64
 16  CTR                    18330 non-null  float64
 17  CPM                    18330 non-null  float64
 18  CPC                    18330 non-null  float64
dtypes: float64(6), int64(7), object(6)
memory usage: 3.3+ MB
```

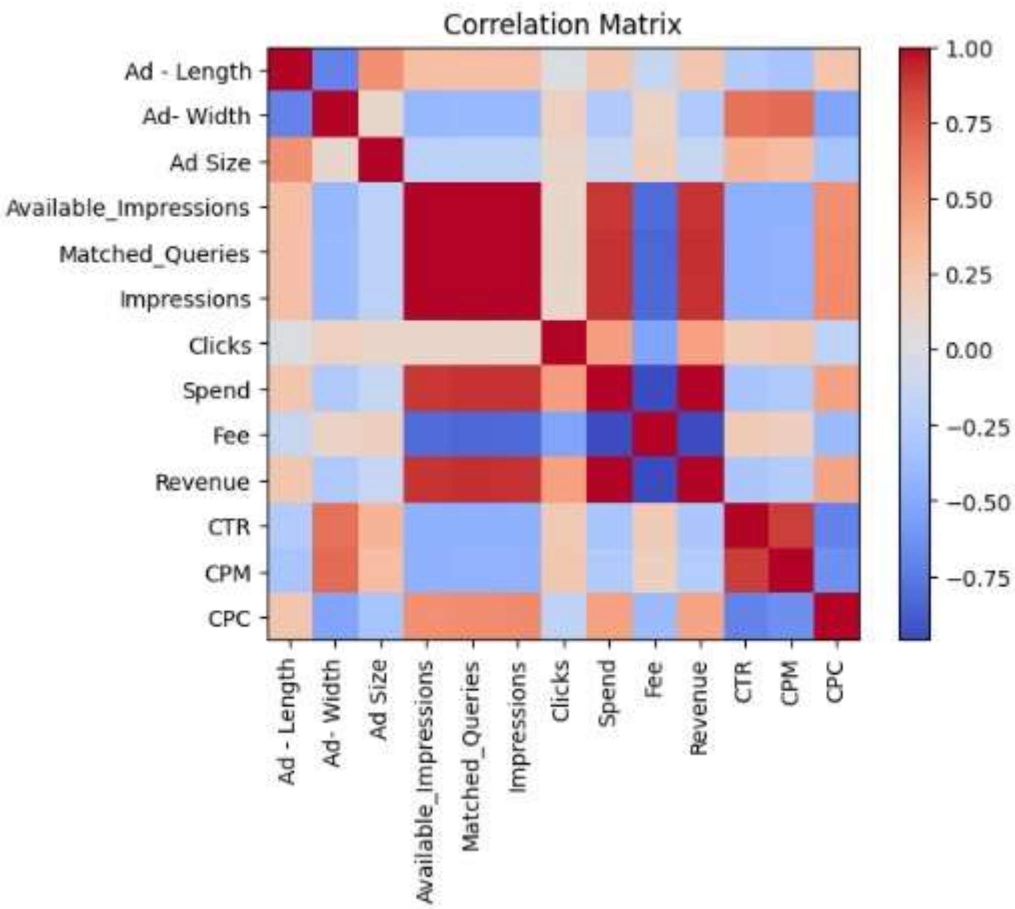
There are no duplicate values in dataframe.

There are 4636 Null values in CTR, CPM, and CPC Columns.

```
Null Values:
Ad - Length                 0
Ad- Width                   0
Ad Size                     0
Available_Impressions       0
Matched_Queries             0
Impressions                 0
Clicks                      0
Spend                       0
Fee                         0
Revenue                     0
CTR                      4736
CPM                      4736
CPC                      4736
dtype: int64
```

Data Types in the Dataset:

```
Timestamp                  object
InventoryType              object
Ad - Length                 int64
Ad- Width                   int64
Ad Size                     int64
Ad Type                    object
Platform                   object
Device Type                object
Format                     object
Available_Impressions       int64
Matched_Queries             int64
Impressions                 int64
Clicks                      int64
Spend                     float64
Fee                       float64
Revenue                   float64
CTR                       float64
CPM                       float64
CPC                       float64
dtype: object
```




**Ad Type:** Inter222 is the most frequently occurring ad type in the dataset.

**Device Type:** Mobile devices appear to be the dominant device type, as indicated by a higher frequency compared to desktop devices.

**Format:** Video ads are more prevalent than display ads or other formats, based on the higher frequency in the dataset

## Step 2: Data Preprocessing:

- **Treat missing values in CPC, CTR and CPM**

We created a function to treat  missing values in CPC, CTR, and CPM columns with the mean values of those columns.

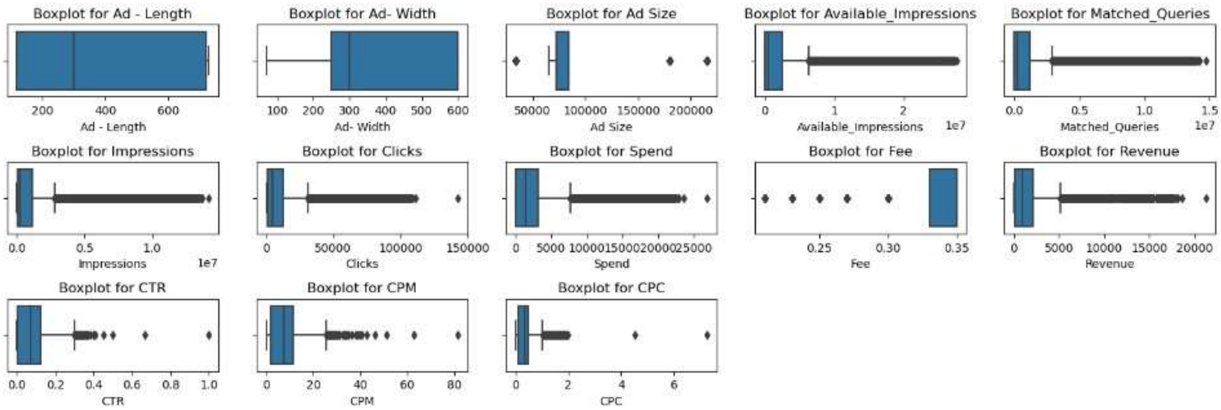
After applying the function to respective columns we got following output.

| Data Set Before Treating Missing Value | Data Set After Treating Missing Value |
|---|---|
| Null Values:<br>Ad - Length 0<br>Ad- Width 0<br>Ad Size 0<br>Available_Impressions 0<br>Matched_Queries 0<br>Impressions 0<br>Clicks 0<br>Spend 0<br>Fee 0<br>Revenue 0<br>CTR 4736<br>CPM 4736<br>CPC 4736<br>dtype: int64 | Null Values:<br>Ad - Length 0<br>Ad- Width 0<br>Ad Size 0<br>Available_Impressions 0<br>Matched_Queries 0<br>Impressions 0<br>Clicks 0<br>Spend 0<br>Fee 0<br>Revenue 0<br>CTR 0<br>CPM 0<br>CPC 0<br>dtype: int64 |

- **Checking and Treating Outliers:**

I have checked with the data and it seems that there are Outliers. Below is the Boxplot figure of Features before Treating Outliers.

Before Treating Outliers

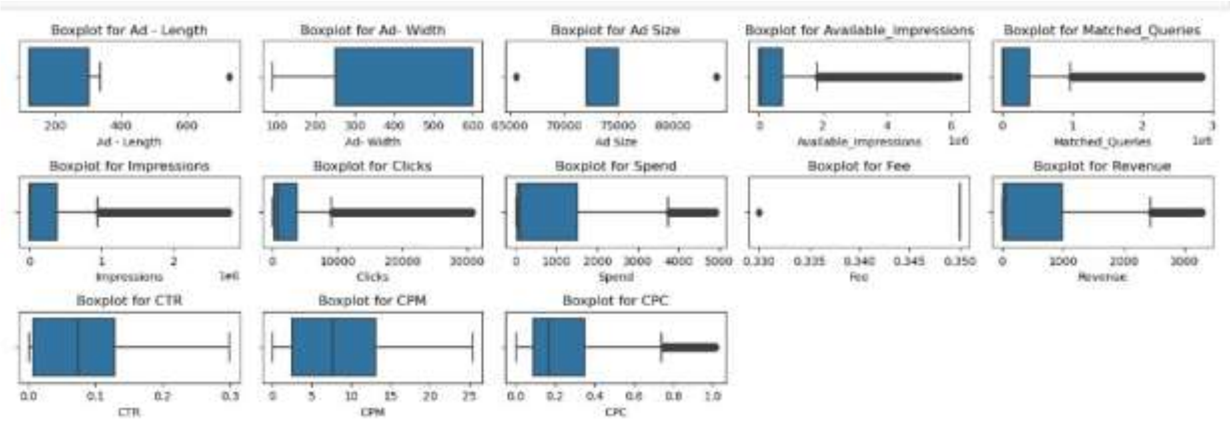



Treating Outliers is necessary for K-Means Clustering. We are going to treat outliers by IQR Method. (IQR : Interquartile Range).

I have created a 'remove_outlier' function using IQR formulas'. We can't perform Outlier Treatment on Categorical features. Hence, I have dropped categorical data and applied 'remove_outlier' function on it. And removed outliers.

Find below Boxplot diagram after treating Outliers.

After Treating Outliers



# Step 3: Standardization:

Standardizing the data using Z-score scaling ensures that all features contribute equally to the clustering process, preventing dominance by variables with larger scales.

Data before z-score scaling is as below

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| Ad - Length | 23066.0 | 3.851631e+02 | 2.336514e+02 | 120.0000 | 120.000000 | 300.000000 | 7.200000e+02 | 728.00 |
| Ad- Width | 23066.0 | 3.378960e+02 | 2.030929e+02 | 70.0000 | 250.000000 | 300.000000 | 6.000000e+02 | 600.00 |
| Ad Size | 23066.0 | 9.667447e+04 | 6.153833e+04 | 33600.0000 | 72000.000000 | 72000.000000 | 8.400000e+04 | 216000.00 |
| Available_Impressions | 23066.0 | 2.432044e+06 | 4.742888e+06 | 1.0000 | 33672.250000 | 483771.000000 | 2.527712e+06 | 27592861.00 |
| Matched_Queries | 23066.0 | 1.295099e+06 | 2.512970e+06 | 1.0000 | 18282.500000 | 258087.500000 | 1.180700e+06 | 14702025.00 |
| Impressions | 23066.0 | 1.241520e+06 | 2.429400e+06 | 1.0000 | 7990.500000 | 225290.000000 | 1.112428e+06 | 14194774.00 |
| Clicks | 23066.0 | 1.067852e+04 | 1.735341e+04 | 1.0000 | 710.000000 | 4425.000000 | 1.279375e+04 | 143049.00 |
| Spend | 23066.0 | 2.706626e+03 | 4.067927e+03 | 0.0000 | 85.180000 | 1425.125000 | 3.121400e+03 | 26931.87 |
| Fee | 23066.0 | 3.351231e-01 | 3.196322e-02 | 0.2100 | 0.330000 | 0.350000 | 3.500000e-01 | 0.35 |
| Revenue | 23066.0 | 1.924252e+03 | 3.105238e+03 | 0.0000 | 55.365375 | 926.335000 | 2.091338e+03 | 21276.18 |
| CTR | 23066.0 | 7.366054e-02 | 6.700065e-02 | 0.0001 | 0.003400 | 0.073661 | 1.219000e-01 | 1.00 |
| CPM | 23066.0 | 7.672045e+00 | 5.777778e+00 | 0.0000 | 1.850000 | 7.672045 | 1.134000e+01 | 81.56 |
| CPC | 23066.0 | 3.510606e-01 | 3.060619e-01 | 0.0000 | 0.100000 | 0.351061 | 4.700000e-01 | 7.26 |

Here, I have applied z-score method and I got the below output.

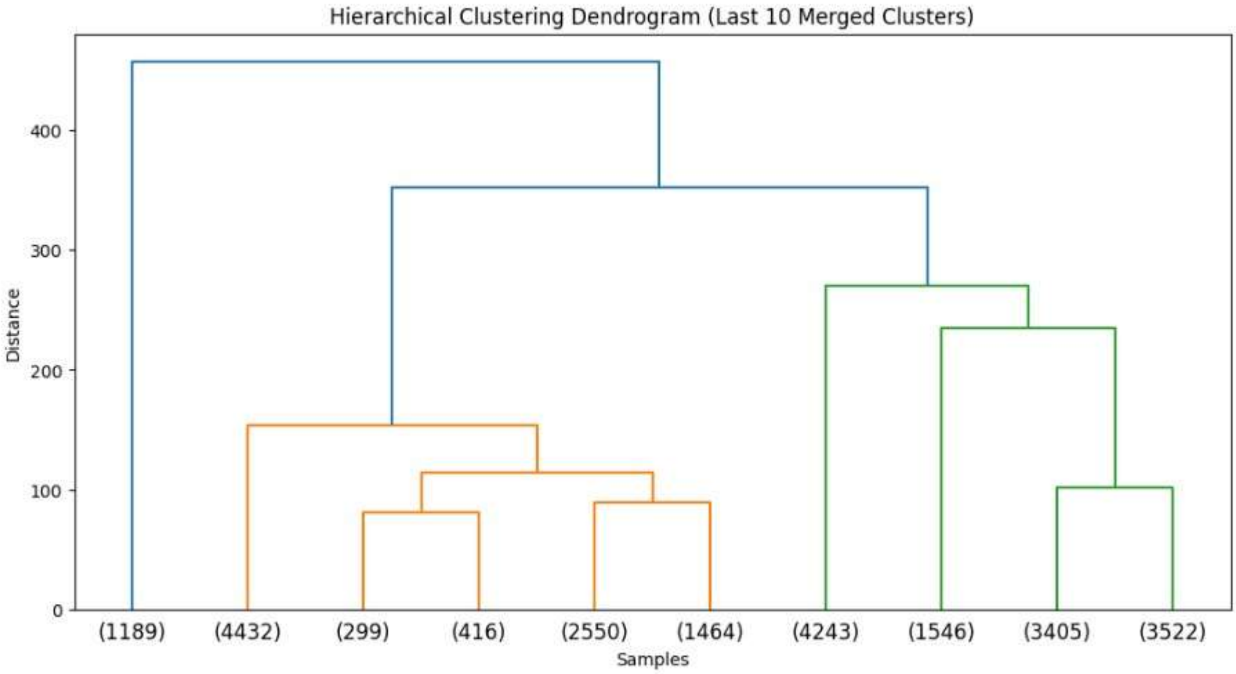
| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| Ad - Length | 23066.0 | 1.281478e-16 | 1.000022 | -1.134891 | -1.134891 | -3.644957e-01 | 1.433093 | 1.467332 |
| Ad- Width | 23066.0 | -1.182903e-16 | 1.000022 | -1.319110 | -0.432797 | -1.865987e-01 | 1.290590 | 1.290590 |
| Ad Size | 23066.0 | 2.464381e-17 | 1.000022 | -1.024985 | -0.400970 | -4.009697e-01 | -0.205965 | 1.939086 |
| Available_Impressions | 23066.0 | -1.971505e-17 | 1.000022 | -0.512788 | -0.505688 | -4.107866e-01 | 0.020171 | 5.305072 |
| Matched_Queries | 23066.0 | -5.914515e-17 | 1.000022 | -0.515377 | -0.508102 | -4.126727e-01 | -0.045524 | 5.335208 |
| Impressions | 23066.0 | -1.971505e-17 | 1.000022 | -0.511050 | -0.507761 | -4.183138e-01 | -0.053138 | 5.331990 |
| Clicks | 23066.0 | -3.943010e-17 | 1.000022 | -0.615311 | -0.574454 | -3.603704e-01 | 0.121894 | 7.628089 |
| Spend | 23066.0 | -3.943010e-17 | 1.000022 | -0.665372 | -0.644432 | -3.150323e-01 | 0.101964 | 5.955310 |
| Fee | 23066.0 | 6.703117e-16 | 1.000022 | -3.914682 | -0.160285 | 4.654474e-01 | 0.465447 | 0.465447 |
| Revenue | 23066.0 | 7.886020e-17 | 1.000022 | -0.619693 | -0.601863 | -3.213727e-01 | 0.053809 | 6.232161 |
| CTR | 23066.0 | 9.857525e-18 | 1.000022 | -1.097932 | -1.048677 | -2.071337e-16 | 0.720001 | 13.826128 |
| CPM | 23066.0 | -9.611087e-17 | 1.000022 | -1.327883 | -1.007683 | -1.537265e-16 | 0.634852 | 12.788576 |
| CPC | 23066.0 | -9.857525e-17 | 1.000022 | -1.147050 | -0.820311 | 0.000000e+00 | 0.388621 | 22.574160 |

Scaling can increase the computational complexity of algorithms, as it involves additional computations to transform the data.

# Step 4: Hierarchical Clustering - Dendrogram:

Dendrogram performed for Hierarchical using WARD and Euclidean Distance on the Scaled Data such as “data1_scaled”.

In this Dendrogram, value of P = 10, which means that only the last 10 merged clusters are shown.



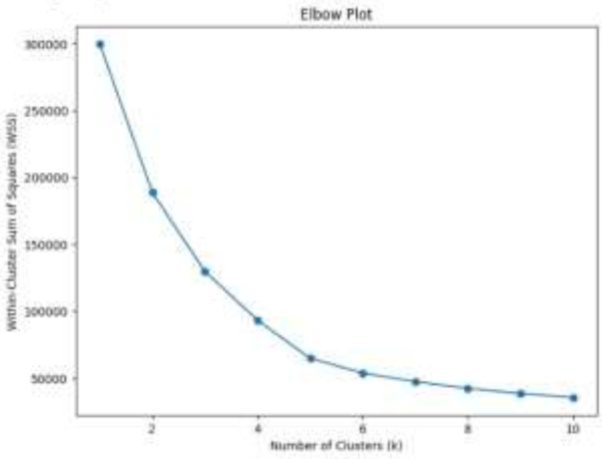

Hierarchical clustering with a dendrogram provides a visual representation of potential clusters, aiding in the selection of an optimal number of clusters for subsequent KMeans analysis.

# Step 5: KMeans Clustering:

- **Elbow Plot**

**We created an Elbow plot (n=10) and identified optimum number of clusters for k-means algorithm.**

Elbow Plot (up to n=10)



Elbow Plot
Within-Cluster Sum of Squares (WSS)
300000
250000
200000
150000
100000
50000
2
4
6
8
10
Number of Clusters (k)

For checking the Optimal number of clusters we use WSS (Within Sum Of Square)

As per the check
When we move from K=1 to K=2, We see that there is a significant drop in the value. Also when we move from k=2 to k-3, k=3 to k=4, k=4 to k=5 there is a significant drop aswell. k=5 to k=6, the drop in values reduces significantly.

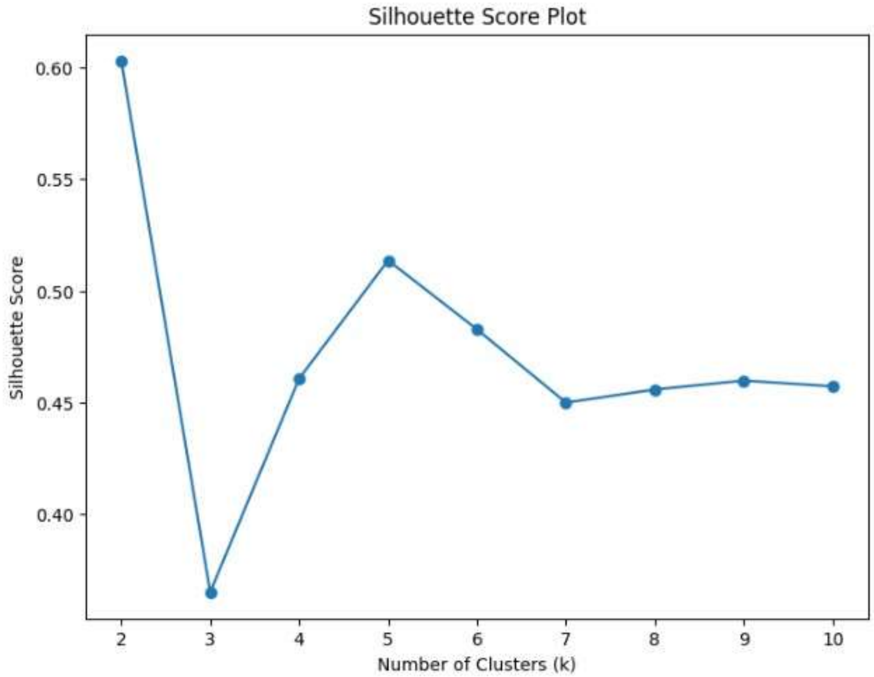
Hence In this case, the WSS is not significantly dropping beyond 5, so 5 is optimal number of clusters.

The Elbow Plot helps determine the optimal number of clusters by identifying the point where adding more clusters does not significantly reduce the within-cluster sum of squares (WSS).

- **Silhouette score:**

**Then we Printed  silhouette scores for up to 10 clusters and identified optimum number of clusters.**

Silhouette scores for up to 10 clusters:




```
Number of Clusters (k) = 2: Silhouette Score = 0.602856419557812
Number of Clusters (k) = 3: Silhouette Score = 0.3652575679239419
Number of Clusters (k) = 4: Silhouette Score = 0.46072044314349486
Number of Clusters (k) = 5: Silhouette Score = 0.5135883146481809
Number of Clusters (k) = 6: Silhouette Score = 0.48271573962694464
Number of Clusters (k) = 7: Silhouette Score = 0.44997366925914933
Number of Clusters (k) = 8: Silhouette Score = 0.45584674165165107
Number of Clusters (k) = 9: Silhouette Score = 0.45983041055564045
Number of Clusters (k) = 10: Silhouette Score = 0.45726048689932824
Optimal number of clusters: 5
```

I have calculated Silhouette Score for scaled data using the silhouette_score() function.
The Silhouette Score is a measure of how similar an object is to its own cluster compared to other clusters, and it ranges from -1 to 1, with higher values indicating better clustering.

As per Elbow plot/scree-plot, we concluded that the optimal number of clusters should be 5. Because 2 would be very less number of clusters.

# Step 6: Conclusion:

- There are 23066 rows, and 19 columns into the Dataset.

- There are no duplicate values in dataframe.
- There are 4636 Null values in CTR, CPM, and CPC Columns.
- I have treated missing values in CPC, CTR, and CPM columns using the given formula
- It seems that there are Outliers into the Dataset
- We treated outliers using IQR method
- I have applied z-score method on the dataframe for scaling.
- I have plotted Dendrogram for value of P = 10
- Plotted elbow plot and got optimum value is 5
- As per Elbow plot/scree-plot, we concluded that the optimal number of clusters should be 5.
- I have created 5 clusters for the Dataset.

Conclusion after Clustering :

- When Click on Ads gets increases then Revenue also increases.
- When amount of money spent on specific ad variations within a specific campaign or ad set is increases then Revenue also increases.
- When impression count of the particular Advertisement increases then Revenue also increases

# References:

# Dataset: data.gov

## Towards Data Science:

https://towardsdatascience.com/unsupervised-machine-learning-clustering-analysis-d40f2b34ae7e

## Analytics Vidhya:

https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/

## Medium:

https://james-thorn.medium.com/how-to-do-a-clustering-project-step-by-step-4b41e94fad1/