PROJECT REPORT

SOMDEB PRAMANIK

CONTENTS

LIST OF TABLES

LIST OF FIGURES

PART 1: CLUSTERING

Clustering is an unsupervised learning technique where the records in a dataset are segmented into groups based on similarities and dissimilarities between the observations, and the groups are formed in such a way that there is significant similarity / homogeneity between the observations in the same group and dissimilarities / heterogeneity exists between observations belonging to different groups.

The dataset on which the clustering was implemented deals with the records from an advertisement company. The analysis was performed in python and the step by step process followed for the clustering exercise and the insights extracted from the clusters are described in details in this report.

The analysis was started by going through the dataset which was provided in the form of an excel file.

The data dictionary was used for understanding the underlying meaning of each and every feature in the data.

Some of the relevant libraries were imported for doing the necessary analysis. Some of the remaining libraries and functions were imported later as and when the requirement was felt.

The dataset was loaded into the jupyter notebook and the top and bottom 5 records were viewed to ensure that the data was properly loaded into a dataframe which was used in the further analysis. It was found that the dataset consisted of 23066 rows and 19 columns. Each row / record represents one particular advertisement published by the company and each column represents one attribute of that advertisement.

While checking the nature of the variables and other information related to the dataset it was found that out of the total 19 features 13 features are of numeric datatype (to be considered for clustering analysis) and the remaining 6 features were of object datatype (categorical).

```
<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
```

TABLE 1.1 : Nature of Variables and other informations related to Ads 24 X 7 Dataset

From the above output we can also conclude that there are some missing values / null values in the last three features (CTR, CPM, and CPC) since the number of non-null values is lesser than the total number of records for each of these three features. The columnwise count of null values was also checked by the traditional method and only these three columns were found to consist of null values.

For imputing these null values in the dataset the following formulae were used for the corresponding features.

$$CTR\ (Click\ Through\ Rate) = \frac{Total\ measured\ clicks}{Total\ measured\ Ad\ impressions} \times 100$$

$$CPM\ (Cost\ Per\ 1000\ Impressions) = \frac{Total\ campaign\ spend}{Number\ of\ impressions} \times 1000$$

$$CPC\ (Cost\ Per\ Click) = \frac{Total\ cost\ (spend)}{Number\ of\ clicks}$$

After imputing the null values the columnwise count of null values was also checked and no column was found to contain any further null values.

No two rows in the dataset were found to be duplicate. The statistical summary was checked separately for the numerical and the categorical columns. The exercise was repeated for the numerical columns after the null value treatment. The different features were found to be on different scales and therefore scaling is a prerequisite for clustering for this dataset because the clustering is a distance based process.

DETECTION OF OUTLIERS AND NECESSITY FOR OUTLIER TREATMENT

The boxplot was obtained for each numeric feature in the dataset (please refer to code file) and they revealed the presence of outliers (and many of them in most cases) for most of the features.

Now the question arises that whether we need to treat the outliers or not. Now there are reasons why we should have treated outliers and then there are reasons why we should avoid outlier treatment too. If we don't treat the outliers, then the efficiency of the k-means clustering will be affected as k-means algorithm is dependant on means and means are sensitive to outliers. Again since most of the features contained too many outliers it is expected that after outlier

treatment the descriptive statistics (i.e the nature) of the data may change (this was indeed the case as was revealed by comparing the descriptive statistical summary of the data before and after outlier treatment) and we may miss out on clusters which might have been composed only of extreme observations and their characteristics.

Therefore in order to keep both these set of contradictory set of reasoning under consideration throughout our analysis, we proceeded with the clustering exercise once with the original data (with outliers) and then once more the exercise was repeated with the modified dataset (obtained by treating the outliers by the IQR method where each outlier was either capped at the upper bound or floored at the lower bound). The results obtained in each case may be subjected to comparison.

CLUSTERING WITH NO OUTLIER TREATMENT

SEPARATING OUT THE NUMERICAL FEATURES FROM THE DATASET

A dataframe was created as a subset of the original dataframe and this new one consisted only of the numerical features from the original dataset. Clustering being a distance based process can be applied only on the numeric features and due to this reason the categorical variables were eliminated. We could have proceeded with the encoded versions of the categorical variables but in that case it would have been extremely difficult to do the cluster profiling (interpretation of the clusters) in terms of the values of the categorical variables.

SCALING THE DATA

Since the numerical features were found to be in different scales, the z score normalisation or scaling was applied on them using the sklearn library of python. After scaling the statistical summary of the scaled data was checked and all the features post scaling were found to have a mean and standard deviation very close to 0 and 1 respectively.

	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
0	-0.364496	-0.432797	-0.352218	-0.512407	-0.515248	-0.510918	-0.615311	-0.665372	0.465447	-0.619693	-0.332572	-0.92711	-0.98660
1	-0.364496	-0.432797	-0.352218	-0.512413	-0.515264	-0.510933	-0.615311	-0.665372	0.465447	-0.619693	-0.332521	-0.92711	-0.98660
2	-0.364496	-0.432797	-0.352218	-0.512213	-0.515235	-0.510905	-0.615311	-0.665372	0.465447	-0.619693	-0.332610	-0.92711	-0.98660
3	-0.364496	-0.432797	-0.352218	-0.512276	-0.515179	-0.510847	-0.615311	-0.665372	0.465447	-0.619693	-0.332712	-0.92711	-0.98660
4	-0.364496	-0.432797	-0.352218	-0.512531	-0.515281	-0.510951	-0.615311	-0.665372	0.465447	-0.619693	-0.332444	-0.92711	-0.98660

TABLE 1.2 : Top 5 rows of the numerical columns after z score scaling (without outlier treatment)

	count	mean	std	min	25%	50%	75%	max
Ad - Length	23066.0	-0.0	1.0	-1.13	-1.13	-0.36	1.43	1.47
Ad- Width	23066.0	0.0	1.0	-1.32	-0.43	-0.19	1.29	1.29
Ad Size	23066.0	-0.0	1.0	-1.02	-0.40	-0.40	-0.21	1.94
Available_Impressions	23066.0	0.0	1.0	-0.51	-0.51	-0.41	0.02	5.31
Matched_Queries	23066.0	0.0	1.0	-0.52	-0.51	-0.41	-0.05	5.34
Impressions	23066.0	0.0	1.0	-0.51	-0.51	-0.42	-0.05	5.33
Clicks	23066.0	0.0	1.0	-0.62	-0.57	-0.36	0.12	7.63
Spend	23066.0	0.0	1.0	-0.67	-0.64	-0.32	0.10	5.96
Fee	23066.0	-0.0	1.0	-3.91	-0.16	0.47	0.47	0.47
Revenue	23066.0	-0.0	1.0	-0.62	-0.60	-0.32	0.05	6.23
CTR	23066.0	0.0	1.0	-0.33	-0.33	-0.32	-0.31	25.13
CPM	23066.0	0.0	1.0	-0.93	-0.73	-0.00	0.51	78.02
CPC	23066.0	-0.0	1.0	-0.99	-0.72	-0.58	0.63	20.29

TABLE 1.3 : Statistical summary of the scaled data (without outlier treatment)

CLUSTERING

K-MEANS CLUSTERING

IDENTIFICATION OF OPTIMUM NUMBER OF CLUSTERS

First we have to decide on the optimum number of clusters to be formed. Since the hierarchical clustering does not provide any tool for scientifically arriving at the optimum number of clusters and also since the dataset was a large one with 23066 records in it, we took up k-means clustering first.

WSS PLOT / ELBOW PLOT APPROACH

The WSS (Within cluster Sum of Squares) values were computed for applications of kmeans clustering for different number of clusters and were plotted as follows.

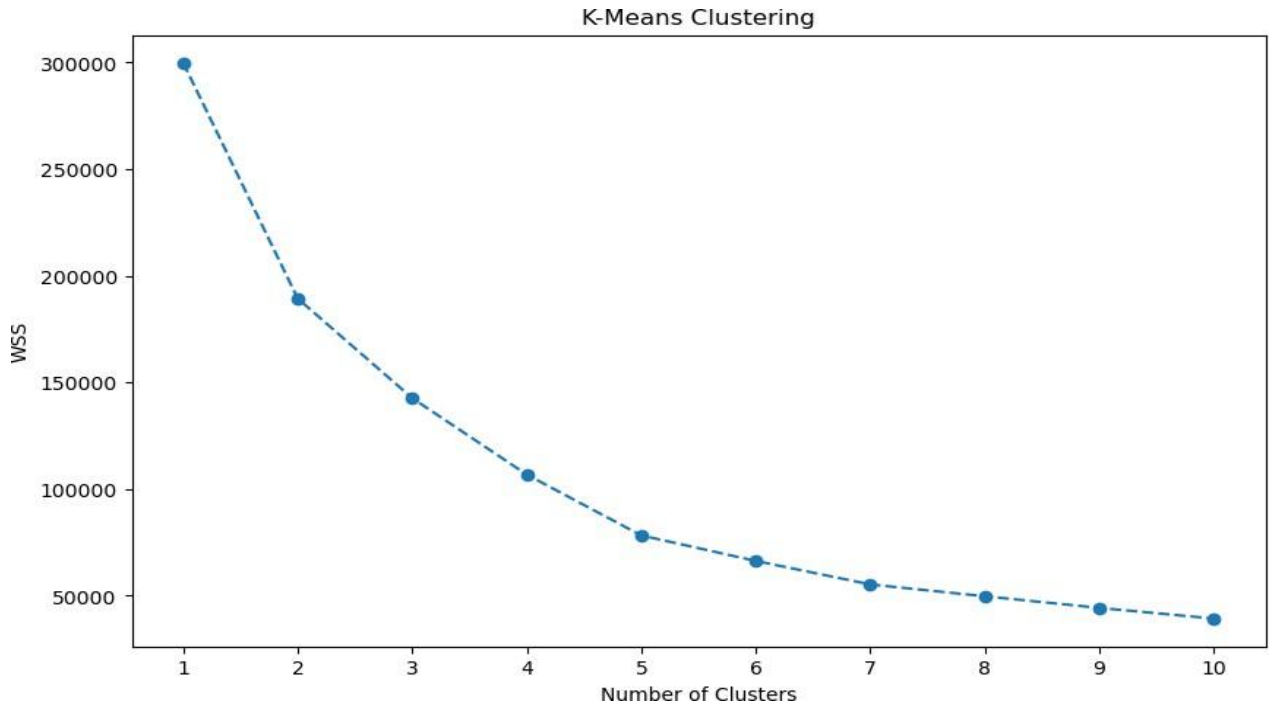

FIGURE 1.1: WSS Plot (without outlier treatment)

Here we find that there are few elbows (points from where the slope decreases or the rate of decrease of WSS is somewhat arrested) but none of them is very significant. So we took the help of the Silhouette Score method to for identifying the optimum number of clusters.

SILHOUETTE SCORE APPROACH

The plot of Silhouette scores for different applications of kmeans clustering with different values for number of clusters is as follows.

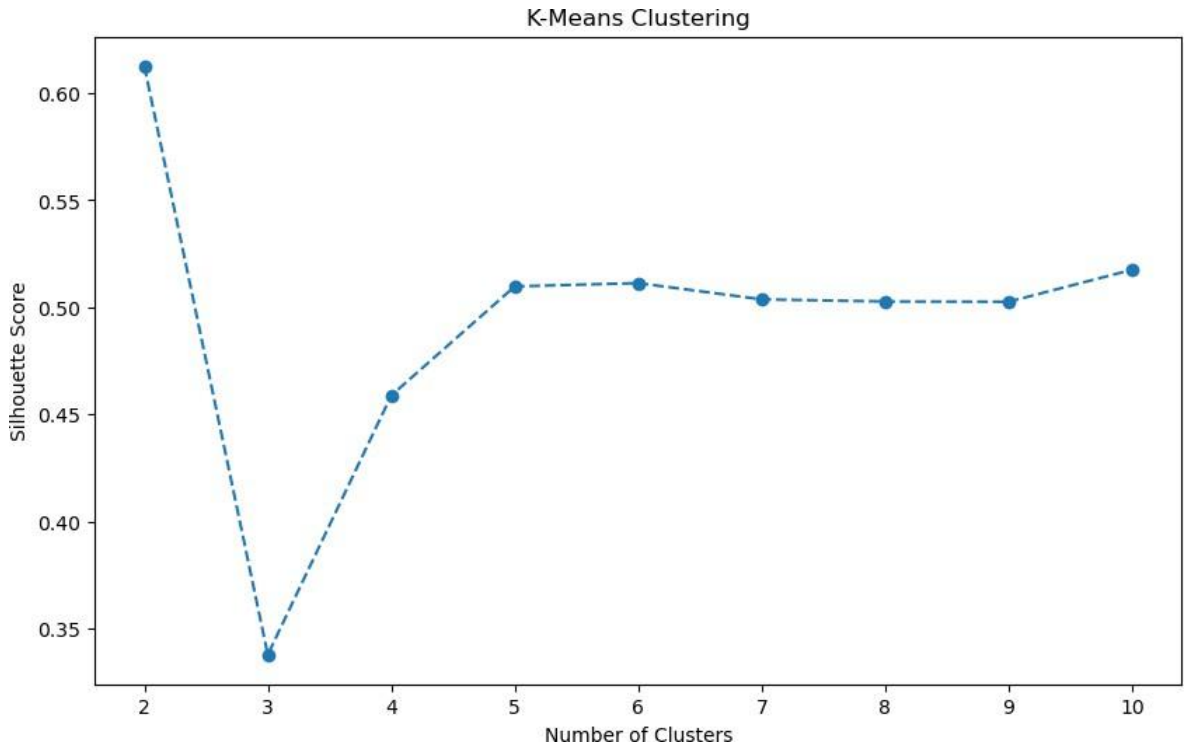

FIGURE 1.2: Variation of Silhouette scores with number of clusters (without outlier treatment)

The maximum Silhouette score was registered for number of clusters = 2.

Now if we go back and look at the WSS plot we can see a bend / elbow at number of clusters = 2 and therefore nothing prevents us from using 2 as the optimum number of clusters.

KMEANS CLUSTERING WITH NUMBER OF CLUSTERS = 2

The kmeans algorithm was applied on the data with number of clusters as 2 and the cluster labels were extracted. Some of the records belonged to cluster 0 and the rest belonged to cluster 1. The cluster label for each record in the dataset was inserted as an additional column in the original dataset.

CLUSTER PROFILING

Let us try to find out the characteristics of each of the two clusters.

cluster-kmeans	Ad - Length	Ad-Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC	count_in_cluster
0	364.61	353.14	98516.05	1350544.69	717946.30	685416.14	10195.19	1820.06	0.34	1235.25	2.79	8.86	0.30	21584
1	684.52	115.88	69853.52	18183105.72	9700812.42	9340666.22	17717.72	15618.60	0.24	11958.99	0.03	1.70	0.91	1482

TABLE 1.4 : K-means cluster profiling (without outlier treatment)

We can observe the following about the two clusters.

- Cluster 0 is more populated than cluster 1.
- Cluster 0 contains the advertisements with greater size (but of smaller length), lesser costing and generate lesser revenue for the company.
- The advertisements in cluster 1 have a lower value of CTR and CPM but a higher value of CPC compared to the advertisements in cluster 0.

Let us also try to extract some insights about the clusters through appropriate visualisations.

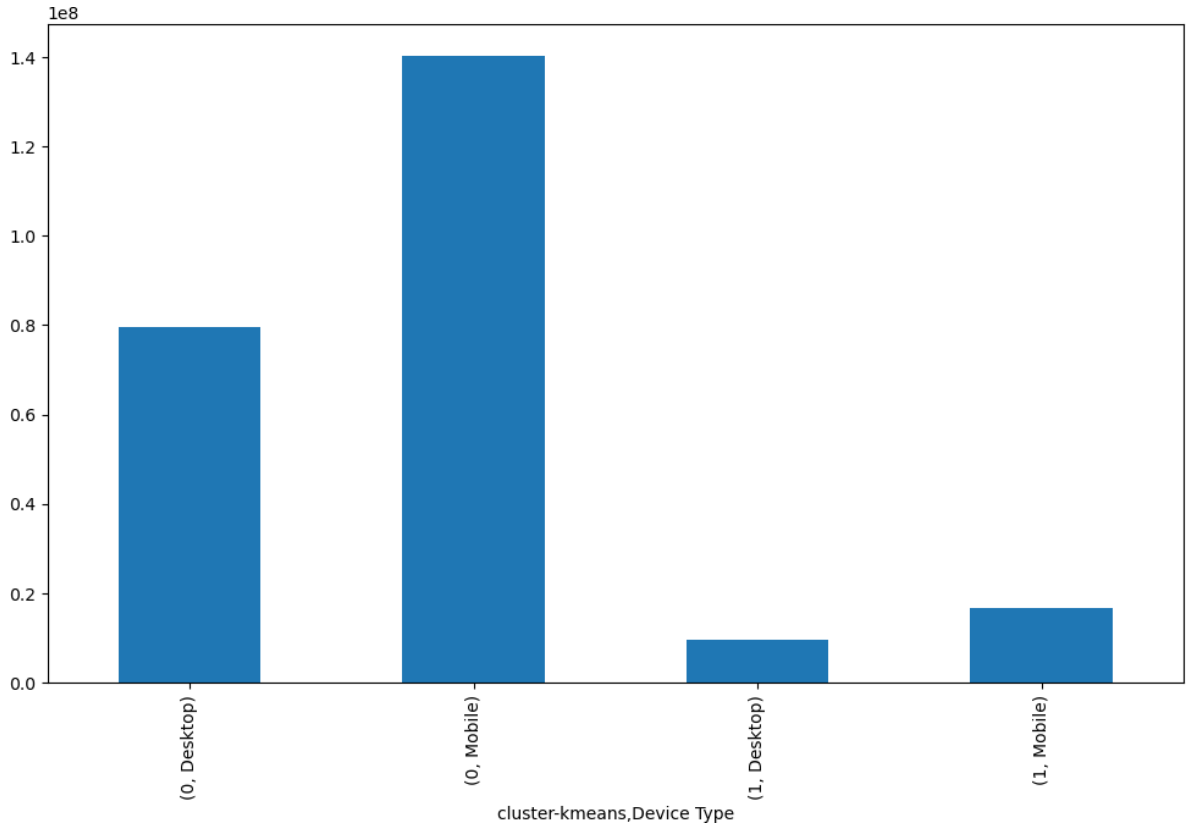

FIGURE 1.3 : Variation of Device types among k-means clusters (without outlier treatment)

From the above plot we can see that for both clusters the advertisements received greater attention (clicks) when they appeared on mobile compared to desktop. Also as already mentioned the majority of the advertisements belonged to cluster 0.

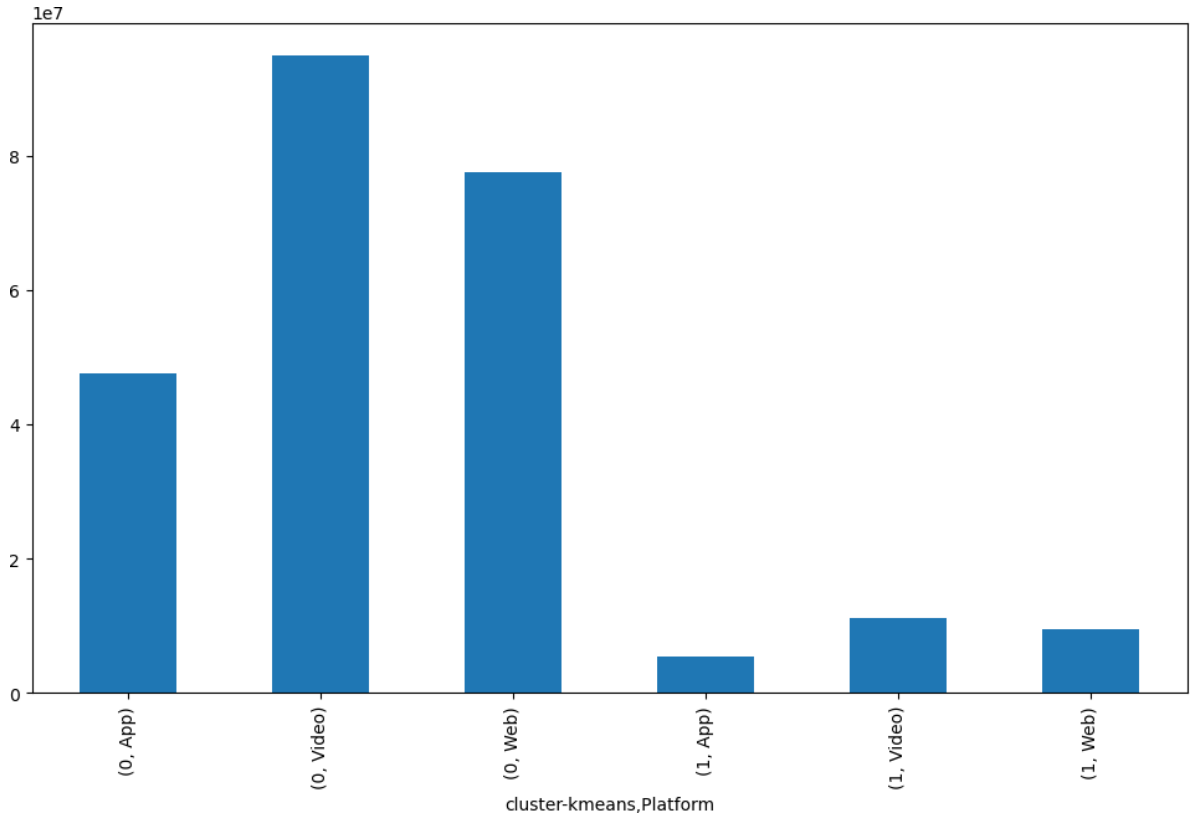

FIGURE 1.4 : Variation of Platforms among k-means clusters (without outlier treatment)

For both clusters the video platform received the maximum attention followed by the web platform and the app platform was the least popular.

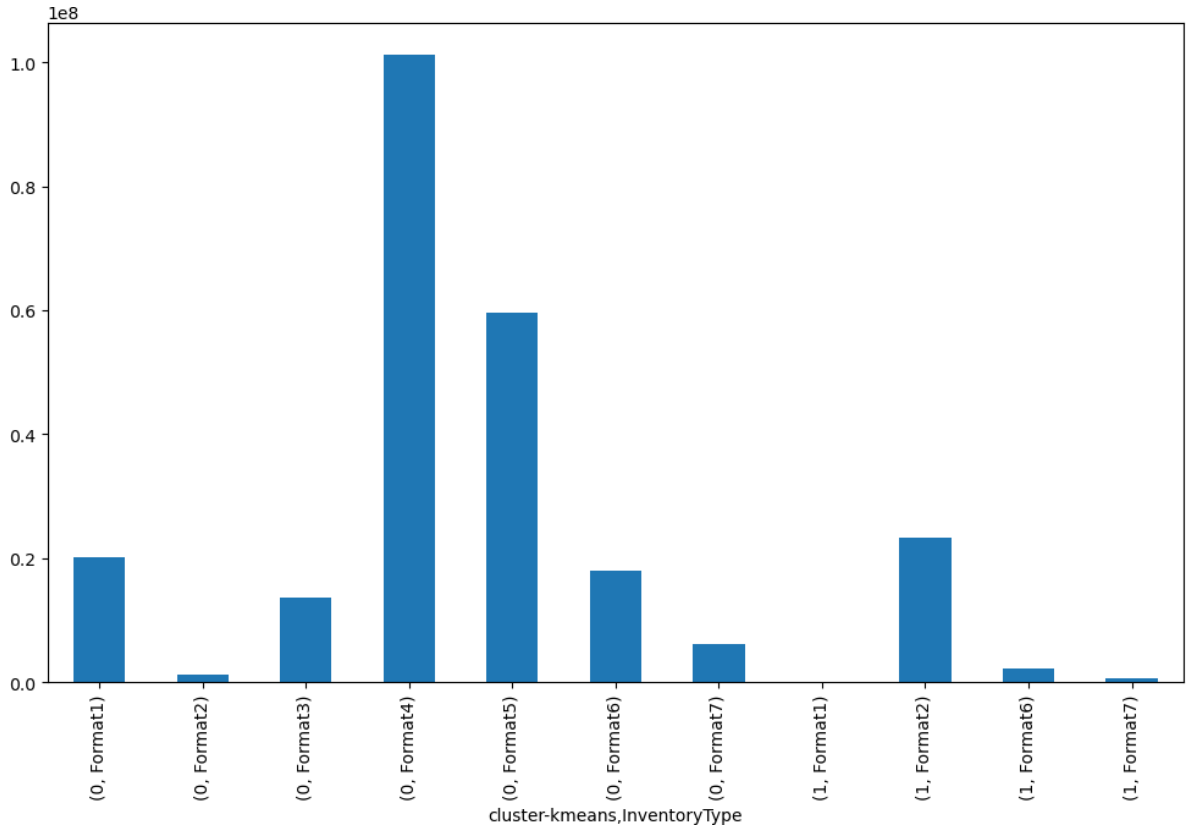

FIGURE 1.5 : Variation of Inventory types among k-means clusters (without outlier treatment)

The advertisements using the format 4 inventory type were the most popular for cluster 0 and the ones using format 2 were the most popular among the members of cluster 1.

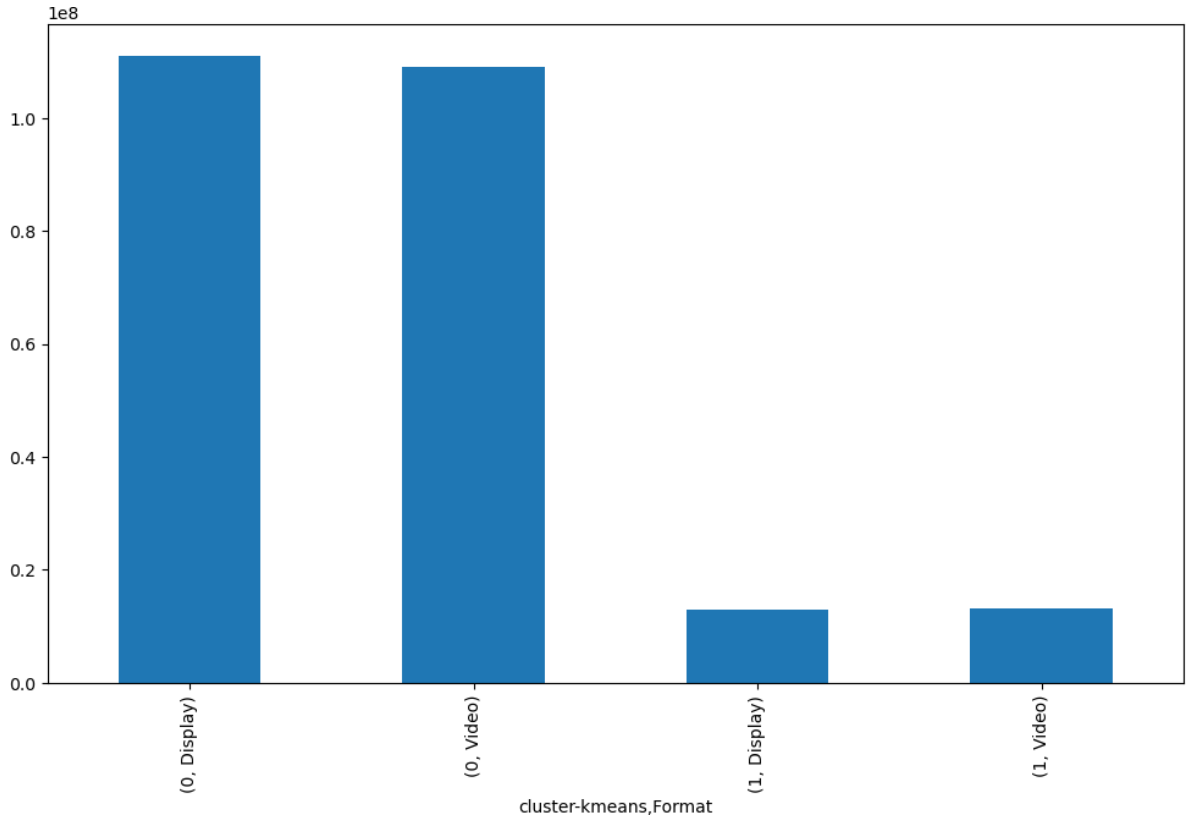

FIGURE 1.6 : Variation of Formats among k-means clusters (without outlier treatment)

One cannot really tell whether the video format or the display format was more effective for any of the two clusters.

While performing the clustering without any outlier treatment we were able to divide the entire dataset into only two clusters (such was the optimum number of clusters yielded by the k-means algorithm). Although the two clusters were reasonably well distinguishable (Silhouette score > 0.6), having only two clusters does not yield a lot of insight and one of the two clusters might have been populated mostly by the extreme values. But here at least we did not change the data a lot and it is very easy for an organisation two look at only two clusters and formulate their strategies accordingly based on their characteristics.

Let us now see the effect of outlier treatment on the number and nature of clusters obtained by k-means clustering.

K-MEANS CLUSTERING WITH THE DATA AFTER OUTLIER TREATMENT

After the data was subjected to outlier treatment (by IQR method) it was subjected to z score scaling (after keeping aside the categorical columns). The scaled data was checked and the statistical summary of the scaled data revealed the mean and the standard deviation of all features converging to 0 and 1 respectively.

	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
0	-0.364496	-0.432797	-0.102518	-0.755333	-0.778949	-0.768478	-0.867488	-0.89317	0.535724	-0.880093	-0.891201	-1.194562	-1.04114
1	-0.364496	-0.432797	-0.102518	-0.755345	-0.778988	-0.768516	-0.867488	-0.89317	0.535724	-0.880093	-0.888615	-1.194562	-1.04114
2	-0.364496	-0.432797	-0.102518	-0.754900	-0.778919	-0.768445	-0.867488	-0.89317	0.535724	-0.880093	-0.893142	-1.194562	-1.04114
3	-0.364496	-0.432797	-0.102518	-0.755040	-0.778781	-0.768302	-0.867488	-0.89317	0.535724	-0.880093	-0.898315	-1.194562	-1.04114
4	-0.364496	-0.432797	-0.102518	-0.755610	-0.779030	-0.768560	-0.867488	-0.89317	0.535724	-0.880093	-0.884734	-1.194562	-1.04114

TABLE 1.5 : Top 5 rows of the numerical columns after z score scaling (after outlier treatment)

	count	mean	std	min	25%	50%	75%	max
Ad - Length	23066.0	-0.0	1.0	-1.13	-1.13	-0.36	1.43	1.47
Ad- Width	23066.0	0.0	1.0	-1.32	-0.43	-0.19	1.29	1.29
Ad Size	23066.0	-0.0	1.0	-1.47	-0.30	-0.30	0.48	1.65
Available_Impressions	23066.0	-0.0	1.0	-0.76	-0.74	-0.53	0.43	2.19
Matched_Queries	23066.0	0.0	1.0	-0.78	-0.76	-0.53	0.37	2.07
Impressions	23066.0	-0.0	1.0	-0.77	-0.76	-0.54	0.37	2.06
Clicks	23066.0	0.0	1.0	-0.87	-0.79	-0.41	0.47	2.36
Spend	23066.0	0.0	1.0	-0.89	-0.86	-0.31	0.39	2.27
Fee	23066.0	-0.0	1.0	-2.22	-0.57	0.54	0.54	0.54
Revenue	23066.0	0.0	1.0	-0.88	-0.85	-0.32	0.39	2.24
CTR	23066.0	-0.0	1.0	-0.91	-0.89	-0.18	0.28	2.03
CPM	23066.0	-0.0	1.0	-1.19	-0.94	0.02	0.70	3.16
CPC	23066.0	0.0	1.0	-1.04	-0.76	-0.60	0.69	2.87

TABLE 1.6 : Statistical summary of the scaled data (after outlier treatment)

CLUSTERING

K-MEANS CLUSTERING

IDENTIFICATION OF OPTIMUM NUMBER OF CLUSTERS

First we have to decide on the optimum number of clusters to be formed.

WSS PLOT / ELBOW PLOT APPROACH

The WSS (Within cluster Sum of Squares) values were computed for applications of kmeans clustering for different number of clusters and were plotted as follows.

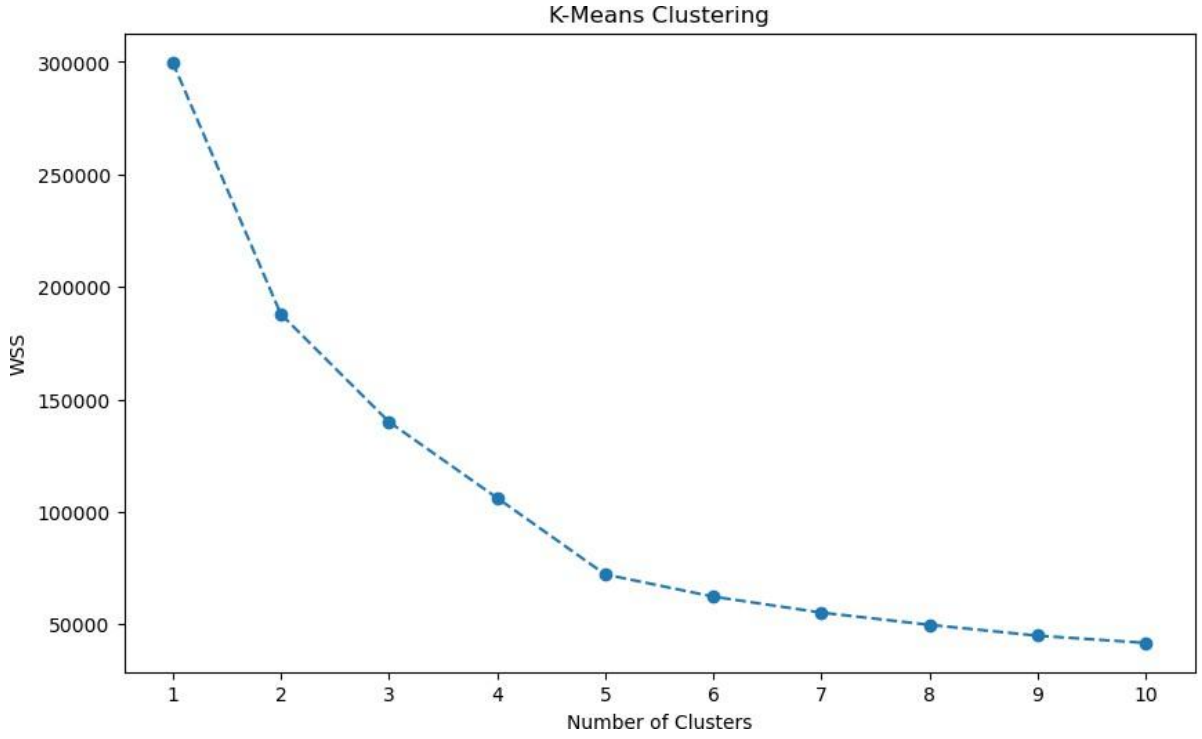

FIGURE 1.7: WSS Plot (after outlier treatment)

Here we find that there are two significant bends / elbows at k = 2 and at k = 5. So we took the help of the Silhouette Score method to for uniquely identifying the optimum number of clusters.

SILHOUETTE SCORE APPROACH

The plot of Silhouette scores for different applications of kmeans clustering with different values for number of clusters is as follows.

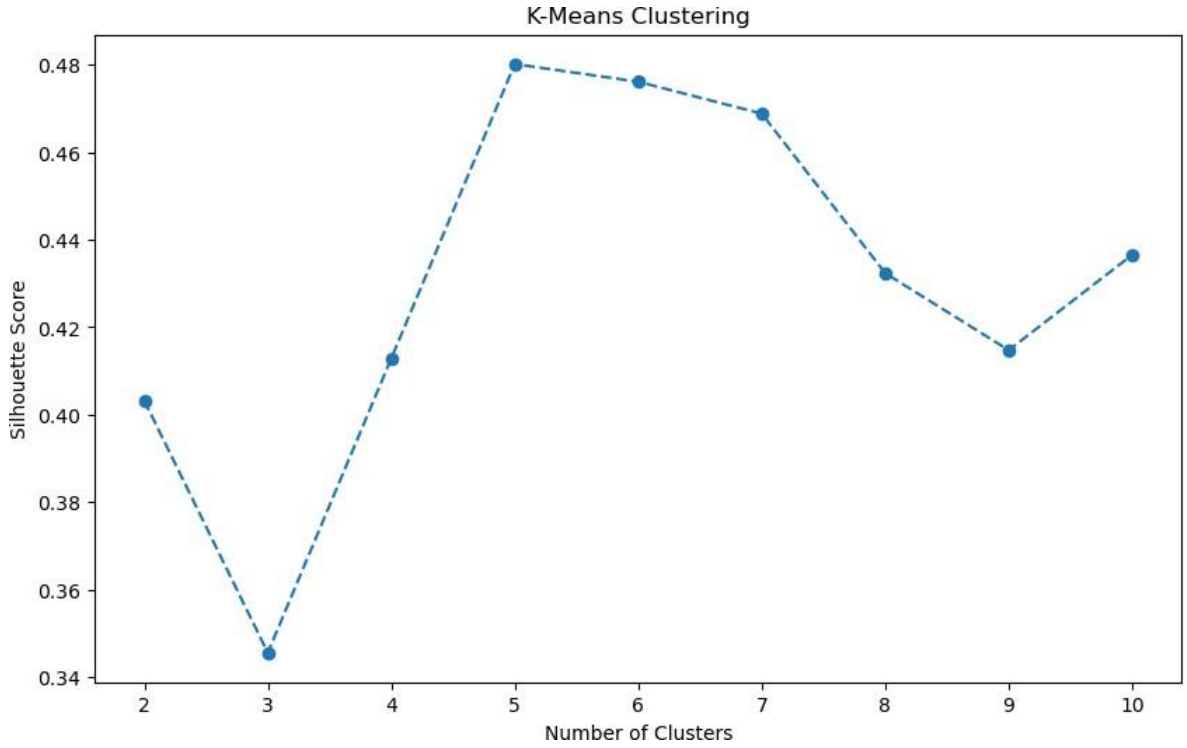

FIGURE 1.8: Variation of Silhouette scores with number of clusters (after outlier treatment)

The maximum Silhouette score was registered for number of clusters k = 5.

Now if we go back and look at the WSS plot and since we can see a bend / elbow at number of clusters k = 5 we can confirm k = 5 as the optimum number of clusters.

KMEANS CLUSTERING WITH NUMBER OF CLUSTERS = 5

The kmeans algorithm was applied on the data with number of clusters as 5 and the cluster labels were extracted. The cluster label for each record in the dataset was inserted as an additional column in the original dataset. The five clusters had labels 0,1,2,3 and 4. The next step would be to do the cluster profiling and come up with insights about the clusters.

cluster-kmeans	Ad - Length	Ad-Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC	count_in_cluster
0	720.00	300.00	102000.00	243989.56	133464.63	113262.32	13738.42	1213.50	0.35	790.32	0.21	11.74	0.09	4247
1	142.23	571.81	73771.81	874997.92	611421.11	515917.36	30917.28	6921.41	0.30	4735.55	0.19	14.94	0.11	1341
2	403.35	172.45	67152.82	2906139.59	1438846.65	1374444.27	4567.44	2496.76	0.34	1668.51	0.05	1.71	0.58	8841
3	156.58	559.26	75300.99	87729.40	48729.12	39823.62	3575.62	413.20	0.35	270.85	0.22	14.06	0.11	7276
4	683.56	117.01	68046.44	6268771.00	2924326.25	2769085.50	18663.62	7675.73	0.30	5145.30	0.03	1.68	0.88	1361

TABLE 1.7 : K-means cluster profiling (after outlier treatment)

We can draw the following conclusions about the five clusters:

- Cluster 0 contains the ads with highest size but these ads are generating very less revenue for the company.
- The cluster 2 contains the highest number of ads but they receive second lowest clicks .
- The cluster 3 contains second lowest number of ads and these ads generate the least revenue.
- The highest revenues are generated by ads belonging to cluster 4.
- It is a common trend that the revenue generation and the spend are proportionate. It will be nice to see for which cluster the difference is maximum and that cluster will be the one with most profitable ads.

Let us now try to extract some insights about the clusters through visualisation also.

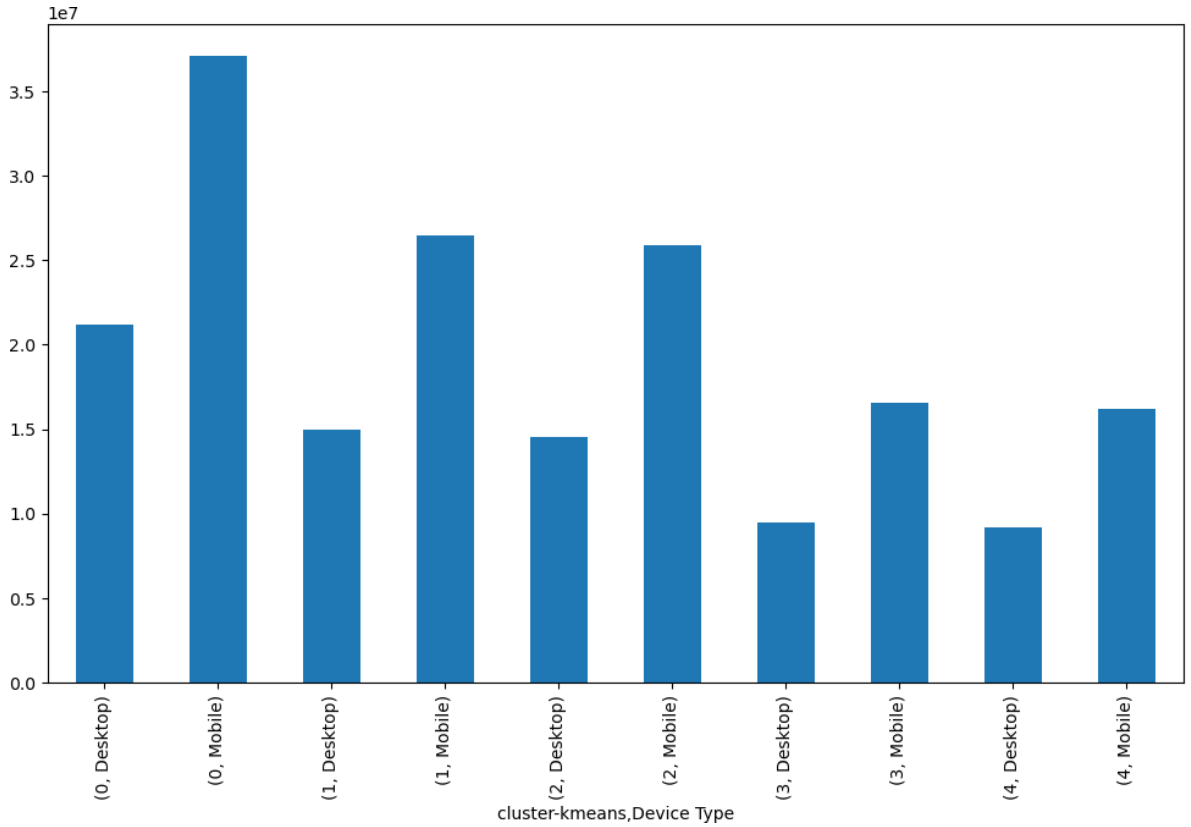

FIGURE 1.9 : Variation of Device types among k-means clusters (after outlier treatment)

For all clusters the mobile ads have received greater attention (clicks) compared to the desktop ads.

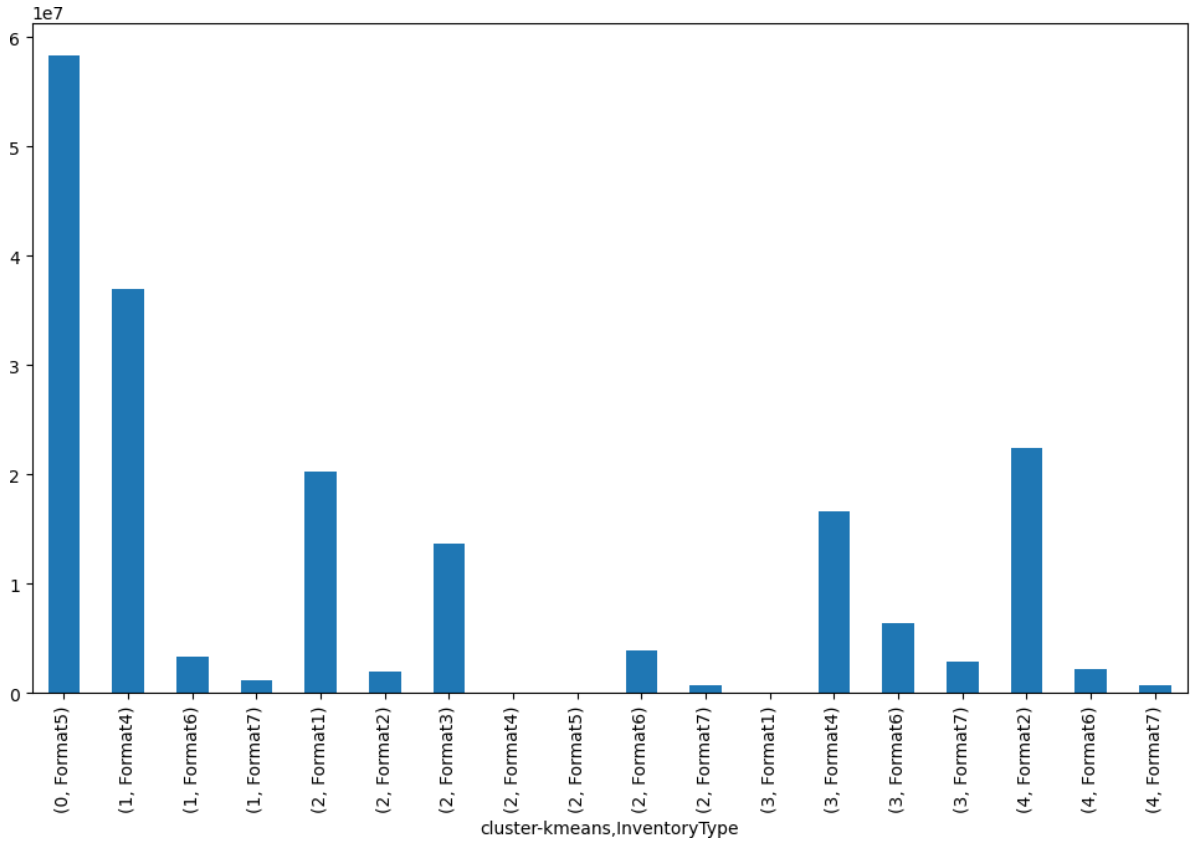

FIGURE 1.10 : Variation of Inventory types among k-means clusters (after outlier treatment)

For clusters 0,1,2,3 and 4 the ads using inventory type 5,4,4,4 and 2 respectively have received the highest attention. So it is recommended to go for format 4 in most cases while publishing ads.

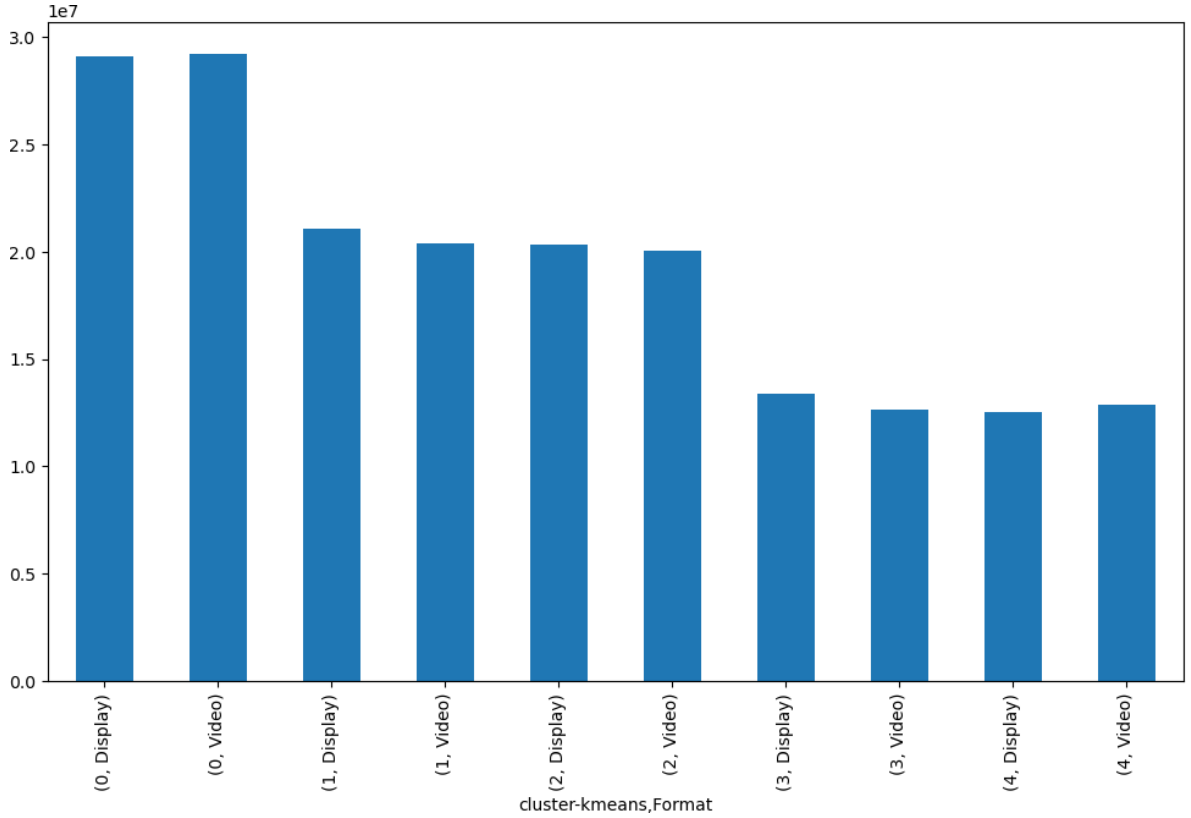

FIGURE 1.11 : Variation of Formats among k-means clusters (after outlier treatment)

Out of the available formats, both display and video have received similar attention for ads belonging to all the 5 clusters.

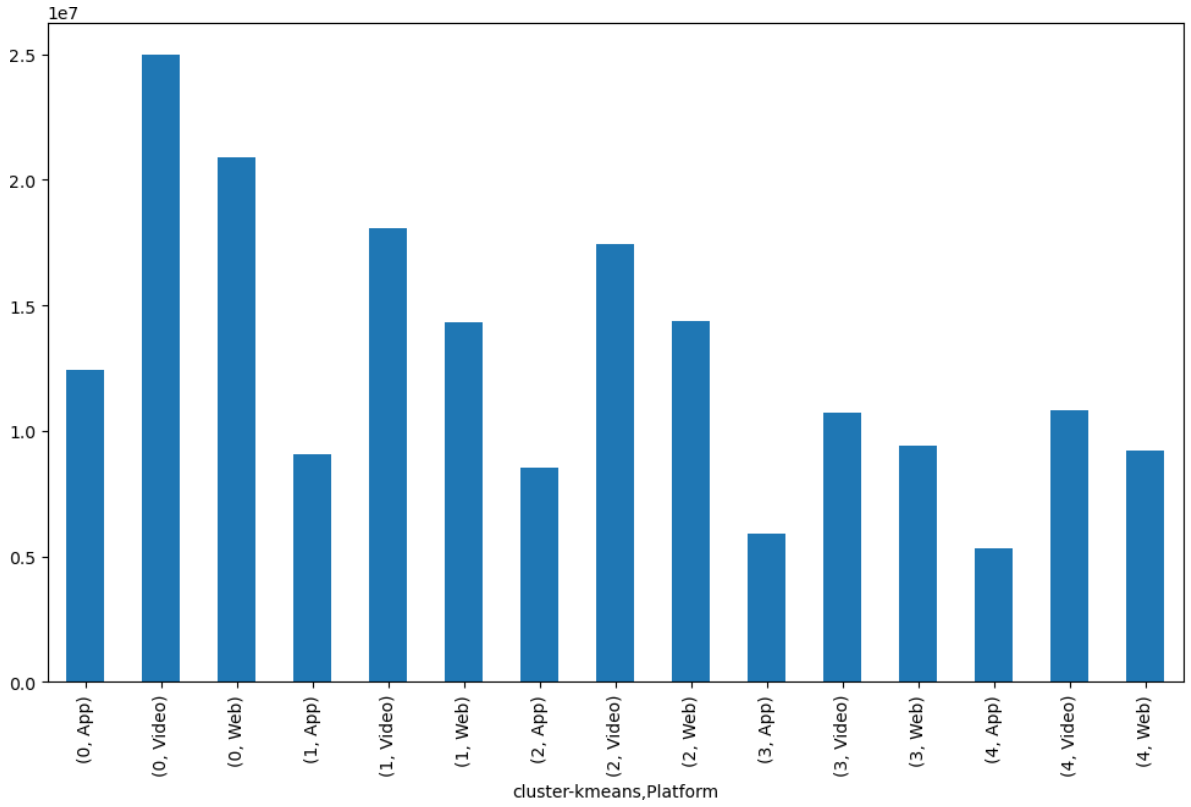

FIGURE 1.12 : Variation of Platforms among k-means clusters (after outlier treatment)

Out of the available platforms the ads using the video platform dominates among ads for all clusters followed by web and app respectively.

HIERARCHICAL CLUSTERING

Since the optimum number of clusters is 5 (which can lead to a more insightful analysis without the number of clusters being too high or too low) for the data with the outliers treated we are going to do the hierarchical clustering for this part only.

However it is not recommended to use hierarchical clustering for this dataset as it is a large one consisting of 23066 rows and computation time and cost will be high if hierarchical clustering is applied to such datasets.

For hierarchical clustering the Euclidean distance and Wards Linkage methods were used.

After that the clusters labelled as 1,2,3,4 and 5 were formed by using the fcluster function.

Throughout the scipy library was used for hierarchical clustering.

The results are shown in tabular format and through appropriate visualisations as follows

DENDROGRAM FOR HIERARCHICAL CLUSTERING

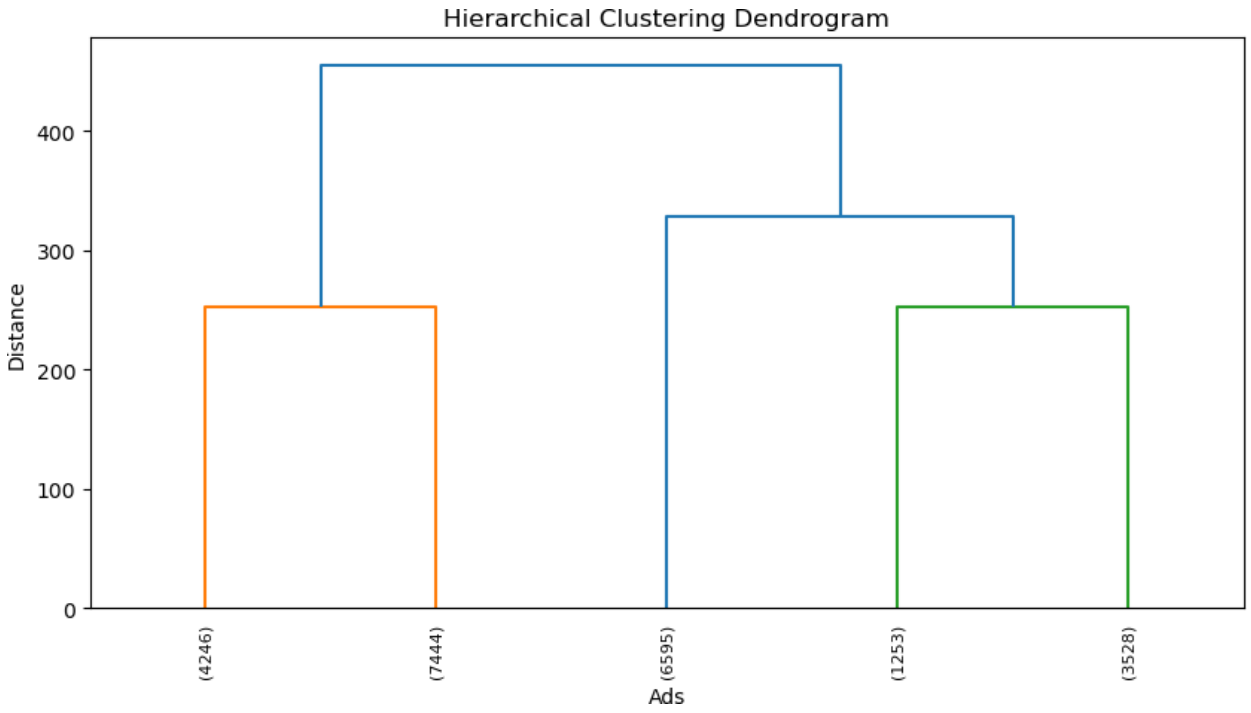

FIGURE 1.13 : Hierarchical (agglomerative) clustering dendrogram

CLUSTER PROFILING

cluster-archical	Ad - Length	Ad-Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC	count_in_cluster
1	720.00	300.00	216000.00	244030.16	133484.40	113278.59	14031.38	1213.78	0.35	790.50	4.94	12.15	0.09	4246
2	158.16	548.99	78108.81	63688.42	39219.73	30359.46	4161.35	432.55	0.35	284.40	4.83	14.16	0.10	7444
3	423.20	158.34	57141.58	2056968.37	1008208.81	967271.82	3509.35	1736.45	0.35	1137.07	0.06	1.84	0.59	6595
4	141.84	572.07	75715.88	878728.68	620739.08	523812.01	71556.26	7645.82	0.28	5532.49	2.28	15.22	0.11	1253
5	476.48	190.60	73581.22	11315339.30	6118794.56	5922454.20	12174.66	9360.91	0.28	6938.79	0.04	1.55	0.75	3528

TABLE 1.8 : Hierarchical (Agglomerative) cluster profiling

It is seen that the clusters obtained by the two approaches (Hierarchical Clustering / Agglomerative Clustering and K-Means Clustering) are not identical. We chose the clusters obtained by K-Means Clustering for further analysis because of the large size of the dataset.

PART 2: PRINCIPAL COMPONENT ANALYSIS

Principal component analysis is a dimension reduction technique and belongs to unsupervised machine learning. This technique was applied on the dataset provided. The dataset consists of several features related to the Census Data for various states and Union Territories of India. The analysis was performed in Python using the relevant available libraries and functions. The step by step process has been described below along with the insights thus extracted. The codes used and the corresponding outputs are attached along with this report for reference.

At the very beginning the data dictionary was studied thoroughly in order to understand the meaning of each and every feature provided in the dataset. The necessary python packages were imported in the jupyter notebook and the dataset was loaded into the same. The top 5 records and the bottom 5 records were looked into and thus it was ensured that the dataset had been loaded properly. The dataset consists of 640 rows and 61 columns. Each row is representing a particular location (a specific Area within a particular State / UT), and each column represents a unique identifier or a variable / metric which was measured for that Area during the Census.

Some of the columns were hidden in the dataframe as the output cell of the jupyter notebook has a limited capacity of displaying upto a certain number of columns. Therefore in the next step the above capacity was increased to 65 so that we can see all the columns of the dataframe without any forced truncation effect.

On checking the ‘info’ of the dataset we were able to extract the following:

Out of the total 61 columns, 59 columns are of numeric (int64) datatype and only the 'State' and 'Area' columns belong to the 'Object' datatype. Moreover it is intuitive that the State Code and the Dist Code fields are acting as the unique identifiers and will not play an important roles in the PCA. Subsequently we should get rid of these 4 fields before proceeding with PCA. Moreover it was found that in each of the 61 features, there are 640 non-null values (which is equal to the number of records in the dataset). Therefore there are no missing values in the dataset. This aspect was examined by alternate methods also and the same conclusion was arrived at.

The dataset was not found to contain any duplicate records.

On checking the columnwise count of null values in the traditional way a truncated output was obtained as follows.

```
State Code       0
Dist.Code        0
State            0
Area Name        0
No_HH            0
                ..
MARG_HH_0_3_F    0
MARG_OT_0_3_M    0
MARG_OT_0_3_F    0
NON_WORK_M       0
NON_WORK_F       0
Length: 61, dtype: int64
```

TABLE 2.1 : Columnwise count of null values in the dataset

Therefore the existence of null values was examined in yet another way. Assuming that there are null values in some of the rows we dropped those rows from the dataframe and then performed a row count to see how many rows have been dropped. It was found that the row count remained the same as the original dataframe. This bears testimony to the fact that the dataset was devoid of missing values.

The statistical summary of the numerical columns was also extracted and it was observed that the scales of the different features were different. So scaling is a must before application of PCA.

EXPLORATORY DATA ANALYSIS (EDA)

For performing EDA on the given dataset the first 29 features were selected as instructed in the question. This included the first 4 features and then the 25 features mentioned. Through the EDA we would like to answer the following questions.

- Which State / UT has the highest Gender Ratio and which has the lowest?
- Which District has the highest Gender Ratio and which has the lowest?
- Which States / UTs have the highest and lowest populations?
- Which States / UTs have the highest and lowest proportion of literate people?
- Which States / UTs have the highest and lowest proportion of employed people?
- Is there a correlation between education (literacy) and employment?

Gender ratio was defined as the ratio of total male and total female population of the concerned region.

Which State / UT has the highest Gender Ratio and which has the lowest?

It was found from both visual and non visual anlysis that among the States / UTs Lakshadweep has the highest Gender ratio of 0.87 and Andhra Pradesh has the lowest gender ratio of 0.53. This can be seen in the following table and figure.

State	
Lakshadweep	0.870000
Haryana	0.778095
NCT of Delhi	0.767778
Uttar Pradesh	0.761690
Punjab	0.748000
Bihar	0.746053
Meghalaya	0.737143
Jammu & Kashmir	0.728636
Daman & Diu	0.705000
Chandigarh	0.700000
Rajasthan	0.688182
Assam	0.686667
Jharkhand	0.678333
Gujarat	0.661538
Sikkim	0.655000
West Bengal	0.648421
Manipur	0.641111
Dadara & Nagar Havelli	0.640000
Karnataka	0.637000
Madhya Pradesh	0.634600
Mizoram	0.633750
Andaman & Nicobar Island	0.633333
Himachal Pradesh	0.628333
Tripura	0.622500
Goa	0.620000
Uttarakhand	0.618462
Puducherry	0.600000
Kerala	0.592857
Nagaland	0.587273
Maharashtra	0.575429
Arunachal Pradesh	0.571250
Odisha	0.555000
Tamil Nadu	0.542813
Chhattisgarh	0.537222
Andhra Pradesh	0.534348

Name: gender_ratio, dtype: float64

TABLE 2.2 : Statewise gender ratio values

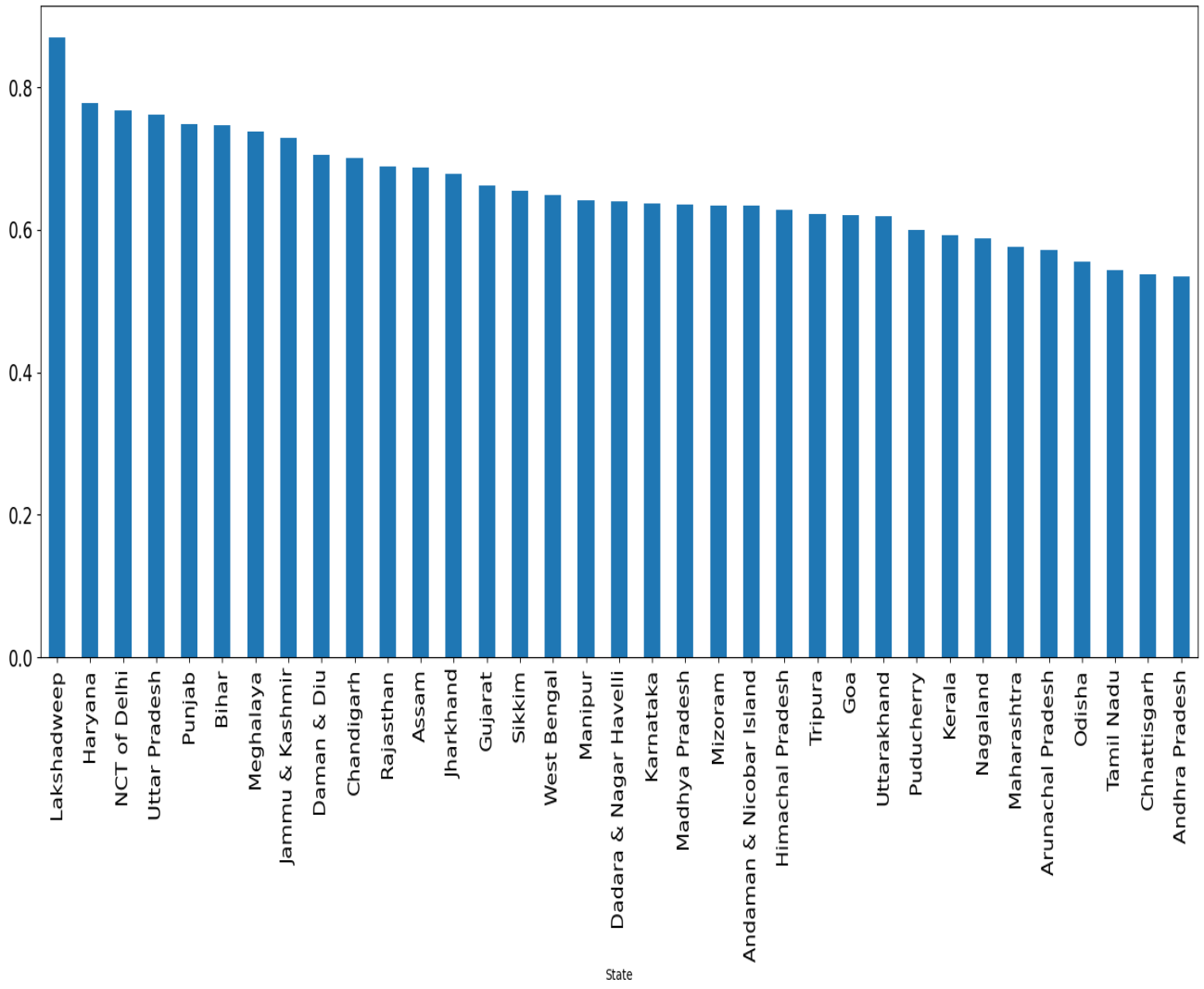

FIGURE 2.1: Statewise variation of gender ratio

Which District has the highest Gender Ratio and which has the lowest?

Among the districts (subdivisions within the states /UTs) the highest gender ratio was recorded for Lakshadweep (0.87) and the lowest gender ratio was recorder for Koraput district of Odisha (0.44) and Krishna district of Andhra Pradesh(0.44). This can be seen from the following table.

State	Area Name	
Lakshadweep	Lakshadweep	0.87
Uttar Pradesh	Mahamaya Nagar	0.85
Rajasthan	Dhaulpur	0.85
Jammu & Kashmir	Badgam	0.85
Uttar Pradesh	Baghpat	0.84
		...
Tamil Nadu	Erode	0.45
Odisha	Baudh	0.45
Tamil Nadu	Virudhunagar	0.45
Andhra Pradesh	Krishna	0.44
Odisha	Koraput	0.44

Name: gender_ratio, Length: 640, dtype: float64

TABLE 2.3 : Districtwise gender ratio values

Which States have the highest and lowest populations?

The total population for a State / UT is determined by adding the male and female populations and by using ‘sum’ as the aggregate function while using the group by function of pandas to generate the total population for each State / UT. The highest population The highest population was recorded in Uttar Pradesh and the lowest population was recorded in Dadara & Nagar Havelli.

State	
Uttar Pradesh	21067854
Maharashtra	11334687
West Bengal	9928671
Bihar	9431081
Andhra Pradesh	9371598
Karnataka	8755157
Tamil Nadu	8684319
Kerala	7776182
Madhya Pradesh	5525353
Rajasthan	5029059
Gujarat	4923157
Odisha	3997011
Punjab	3700830
Assam	3530700
Jharkhand	2966507
Haryana	2666689
Chhattisgarh	2364996
NCT of Delhi	1908680
Uttarakhand	1587071
Himachal Pradesh	1235443
Jammu & Kashmir	994172
Meghalaya	624391
Tripura	416827
Manipur	372487
Goa	310372
Nagaland	199441
Puducherry	189460
Mizoram	154997
Arunachal Pradesh	138648
Chandigarh	101397
Sikkim	68182
Andaman & Nicobar Island	47417
Daman & Diu	31859
Lakshadweep	27595
Dadara & Nagar Havelli	17813

Name: T_POP, dtype: int64

TABLE 2.4 : Statewise total population values

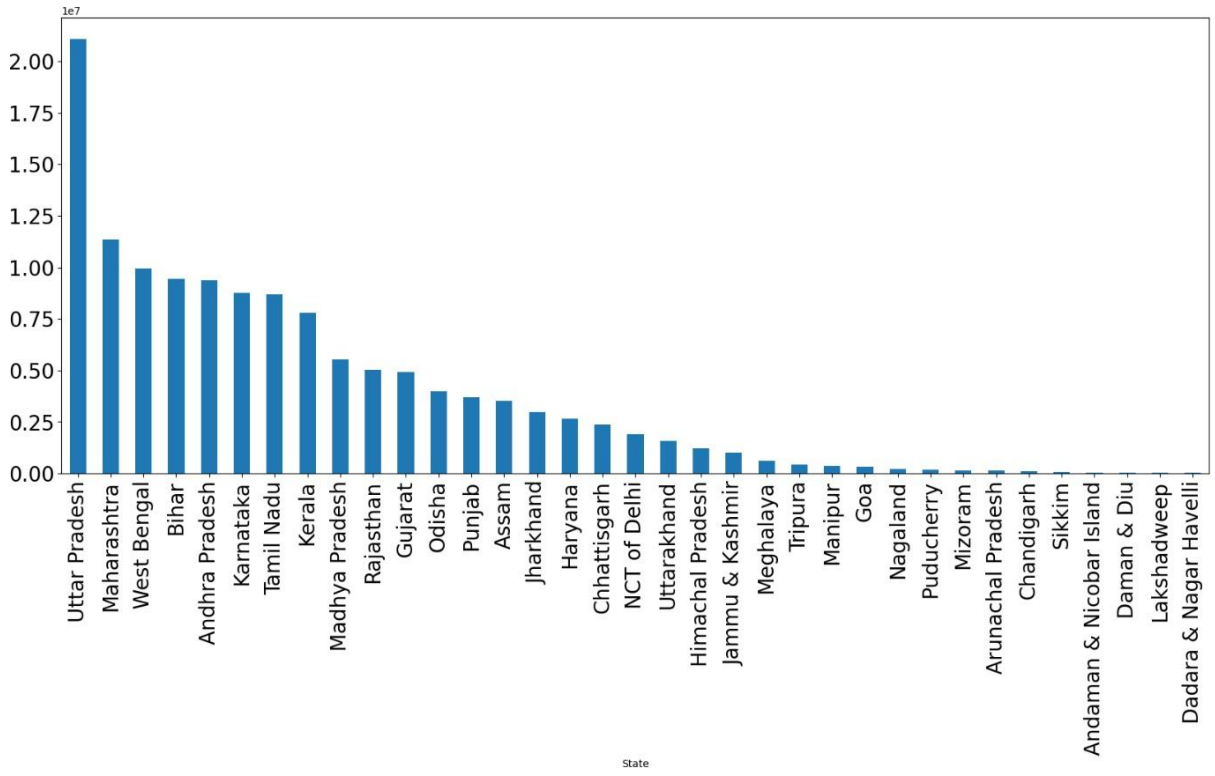

FIGURE 2.2: Statewise variation of total population

Which States / UTs have the highest and lowest proportion of literate people?

The proportion of literate people for a particular state / UT was calculated by taking the ratio of the literate population (sum of no of literate males and literate females) and the total population. The states of Kerala and Bihar recorded the highest (0.806) and lowest (0.48) proportions of literate population respectively.

State	
Kerala	0.806429
Lakshadweep	0.790000
Mizoram	0.788750
Goa	0.770000
Chandigarh	0.760000
Tripura	0.745000
Puducherry	0.740000
NCT of Delhi	0.738889
Daman & Diu	0.735000
Andaman & Nicobar Island	0.723333
Himachal Pradesh	0.693333
Sikkim	0.692500
Maharashtra	0.674571
Nagaland	0.667273
Uttarakhand	0.657692
Punjab	0.651500
Tamil Nadu	0.645312
Manipur	0.644444
Haryana	0.633333
Gujarat	0.631538
West Bengal	0.624737
Meghalaya	0.612857
Assam	0.611852
Karnataka	0.606333
Dadara & Nagar Havelli	0.590000
Odisha	0.570667
Arunachal Pradesh	0.552500
Madhya Pradesh	0.552400
Uttar Pradesh	0.542113
Andhra Pradesh	0.534348
Jammu & Kashmir	0.530909
Rajasthan	0.528182
Chhattisgarh	0.522778
Jharkhand	0.515417
Bihar	0.480263

Name: LIT_RATE, dtype: float64

TABLE 2.5 : Statewise proportions of literate population

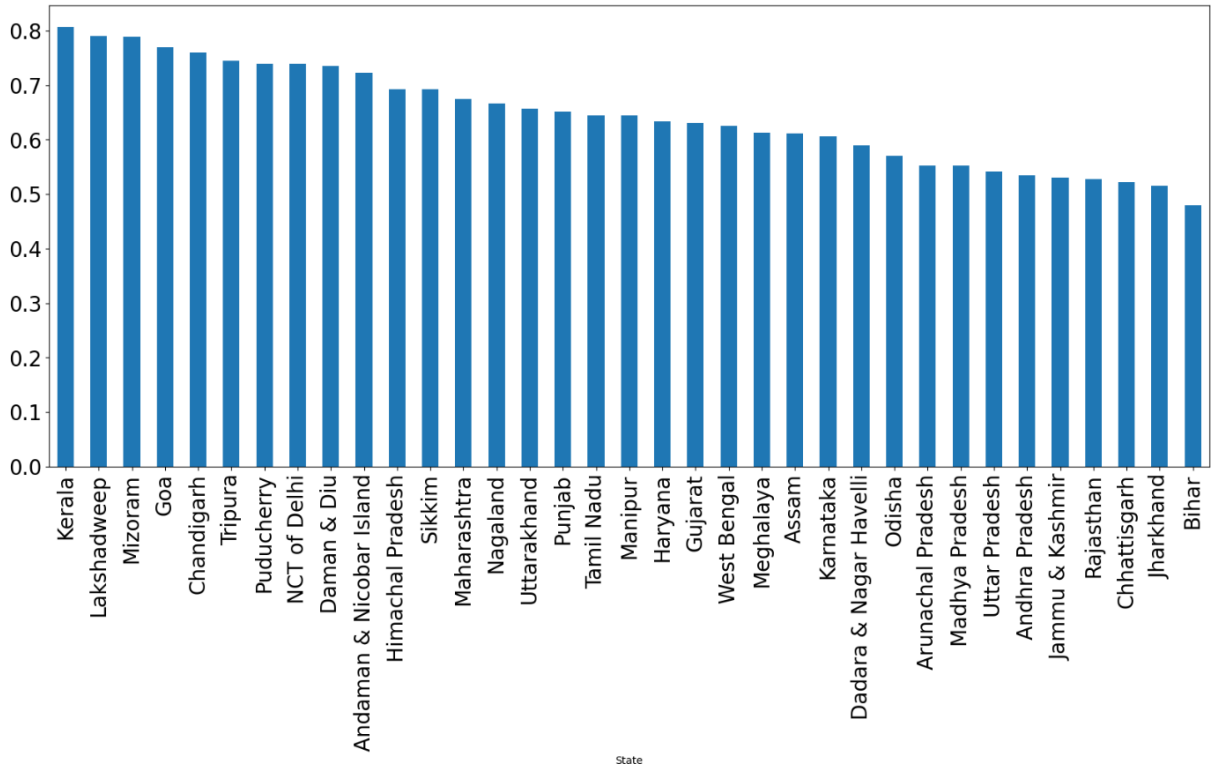

FIGURE 2.3 : Statewise variation of proportions of literate population

Which States / UTs have the highest and lowest proportion of employed people?

The proportion of employed people for a particular state / UT was calculated by taking the ratio of the working population (sum of no of working males and working females) and the total population. Nagaland and Lakshadweep recorded the highest (0.54) and lowest (0.25) proportions of working population respectively.

State	
Nagaland	0.538182
Himachal Pradesh	0.511667
Sikkim	0.507500
Chhattisgarh	0.498333
Tamil Nadu	0.485938
Andhra Pradesh	0.484348
Karnataka	0.477333
Manipur	0.475556
Mizoram	0.460000
Maharashtra	0.455143
Madhya Pradesh	0.442200
Odisha	0.437000
Arunachal Pradesh	0.426250
Meghalaya	0.425714
Dadara & Nagar Havelli	0.420000
Tripura	0.420000
Jharkhand	0.419167
Rajasthan	0.417576
Gujarat	0.415000
Assam	0.414815
West Bengal	0.401579
Andaman & Nicobar Island	0.393333
Uttarakhand	0.379231
Goa	0.365000
Chandigarh	0.360000
NCT of Delhi	0.344444
Daman & Diu	0.340000
Puducherry	0.330000
Bihar	0.328421
Haryana	0.328095
Uttar Pradesh	0.326338
Punjab	0.320000
Kerala	0.316429
Jammu & Kashmir	0.315000
Lakshadweep	0.250000

Name: EMP_RATE, dtype: float64

TABLE 2.6 : Statewise proportions of employed population

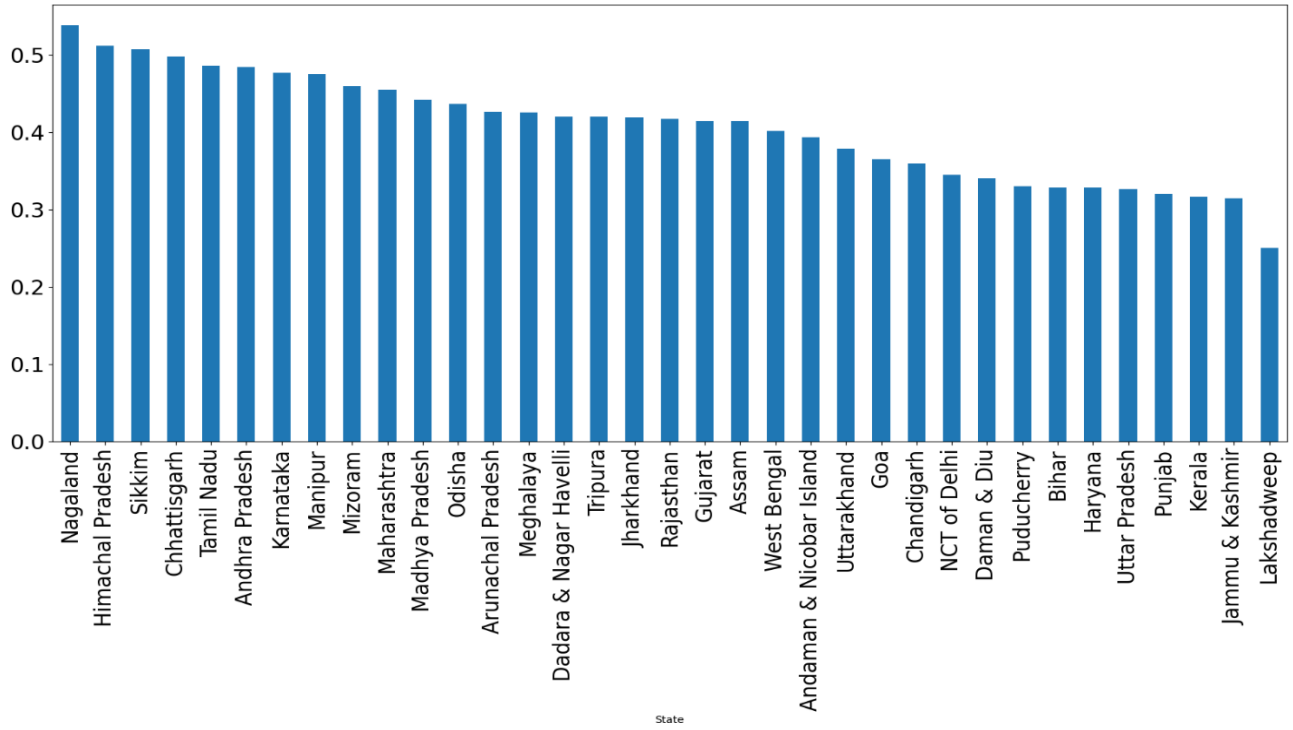

FIGURE 2.4 : Statewise variation in proportion of employed population

Is there a correlation between education (literacy) and employment?

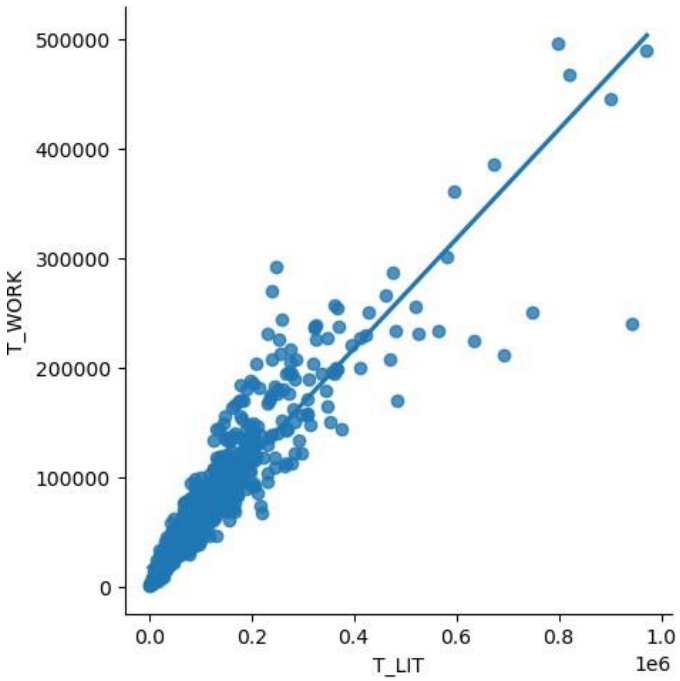

FIGURE 2.5: LMPlot of Literate population and Employed Population

From the above scatterplot (lmplot) it is quite evident that there is a strong positive correlation between the number of literate / educated people and the number of employed people.

OUTLIER TREATMENT AND SCALING

For performing PCA, we need only the independent numerical columns. Therefore a dataframe was created from the main dataframe with only the numerical columns from the main dataset. This dataframe consisted of 57 columns because in addition to the 2 columns with object datatype the State Code and District Code columns were also removed as they will not play any significant role in PCA. The number of rows in the new dataframe was still 640.

The boxplot of each numerical feature was plotted and almost all features were found to contain outliers (please refer to the attached code file for boxplots).

Now the data was subjected to z-score scaling to minimise chances of the results being impacted unduly by features with higher scales in the unscaled dataset. The z-score scaling was performed using zscore function of scipy library and also the StandardScaler function under sklearn. In both cases the scaled data that was obtained were identical. From the statistical summary of the scaled data it was found that each feature had a mean of almost 0 and standard deviation almost equal to 1.

	count	mean	std	min	25%	50%	75%	max
No_HH	640.0	0.0	1.0	-1.06	-0.66	-0.32	0.37	5.39
TOT_M	640.0	-0.0	1.0	-1.08	-0.68	-0.29	0.38	5.53
TOT_F	640.0	-0.0	1.0	-1.07	-0.67	-0.31	0.37	5.53
M_06	640.0	-0.0	1.0	-1.07	-0.66	-0.27	0.37	7.30
F_06	640.0	0.0	1.0	-1.05	-0.64	-0.29	0.35	7.35
M_SC	640.0	-0.0	1.0	-0.96	-0.72	-0.29	0.39	6.21
F_SC	640.0	0.0	1.0	-0.96	-0.70	-0.33	0.39	6.25
M_ST	640.0	-0.0	1.0	-0.63	-0.60	-0.39	0.15	9.15
F_ST	640.0	-0.0	1.0	-0.64	-0.61	-0.40	0.15	7.56
M_LIT	640.0	0.0	1.0	-1.03	-0.66	-0.27	0.36	6.18
F_LIT	640.0	-0.0	1.0	-0.88	-0.61	-0.30	0.25	6.73

TABLE 2.7 : Statistical Summary after z score scaling

Now after scaling again the boxplot was plotted for each feature and it was established that scaling did not have any influence on the outliers (please refer to code file for boxplots).

Now according to the instructions the outliers were not treated but according to my personal judgement outliers must always be treated before applying PCA. PCA is a technique that relies on correlations and correlations in turn depend on means. Now since means are sensitive to outliers, anything that depends on mean will be sensitive to outliers.

STEP BY STEP PCA ON THE SCALED DATA

The correlations among the different pairs of variables in the scaled dataset was visualised using a heatmap.

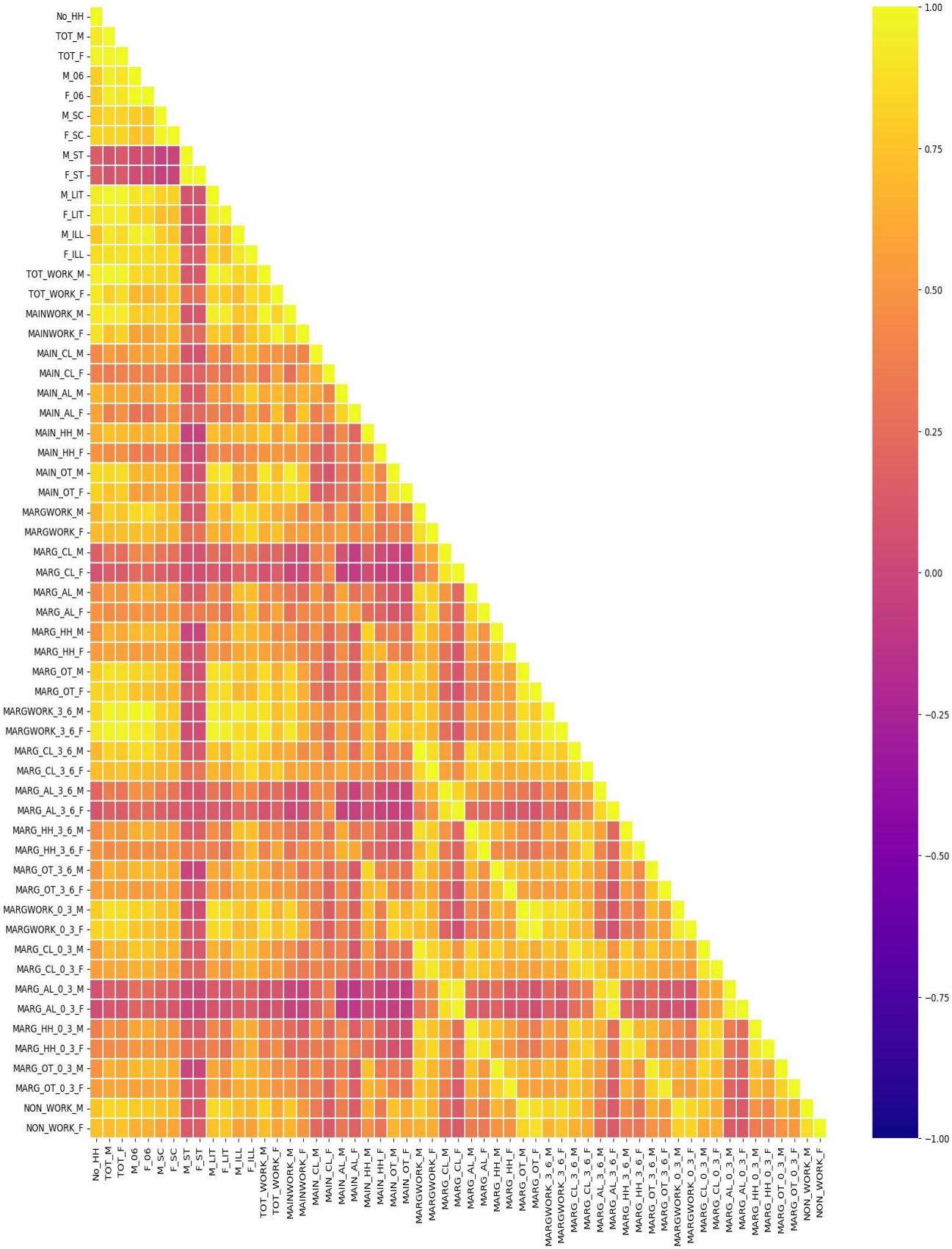

FIGURE 2.6 : Heatmap based on the scaled data

Many of the off diagonal elements in this heatmap are bright yellow in colour which according to the colour code used indicates the existence of very strong positive correlation among the corresponding pairs of original variables.

But before performing PCA it has to be checked whether these correlations are statistically significant enough so as to necessitate PCA. That check was performed by using **Bartlett's Sphericity Test** which is a statistical test of hypothesis.

Null Hypothesis (H_0) : The correlations are not significant.

Alternative Hypothesis (H_1) : The correlations are significant.

The level of significance (α) was assumed to be 5% or 0.05.

The p-value was computed using python and it was very close to 0.

Since p-value < α, we could reject the null hypothesis in favour of alternate hypothesis with evidence from the data and with 95% confidence level. So the correlations among the pairs of original variables are strong enough so as to necessitate PCA.

The adequacy of sample size was checked by performing the **KMO Test** and the kmo_model parameter was obtained as 0.8039, which is greater than 0.7 (the accepted threshold). So we could conclude that the sample size was adequate for PCA.

After these two tests were performed, the PCA was implemented using sklearn and using all the 57 features. The covariance matrix obtained from the scaled data is partly shown below.

```
Covariance matrix:
 [[1.00156495 0.91760364 0.97210871 ... 0.53769433 0.76357722 0.73684378]
 [0.91760364 1.00156495 0.98417823 ... 0.5891007  0.84621844 0.71718181]
 [0.97210871 0.98417823 1.00156495 ... 0.572748   0.82894851 0.74775097]
 ...
 [0.53769433 0.5891007  0.572748   ... 1.00156495 0.61052325 0.52191235]
 [0.76357722 0.84621844 0.82894851 ... 0.61052325 1.00156495 0.88228018]
 [0.73684378 0.71718181 0.74775097 ... 0.52191235 0.88228018 1.00156495]]
```

TABLE 2.8: Covariance matrix for the scaled dataset

This is a matrix of shape (57,57) and since it is a square matrix we can find its 57 Eigen values and 57 Eigen Vectors.

Each Eigen vector / each PC is a linear combination of the original variables. The coefficients or loadings which map the original variables to each PC are stored inside the corresponding Eigen Vector.

The Eigen values (which represent the explained variance of each Eigen vector / each PC) were as follows.

```
[31.814  7.869  4.153  3.669  2.207  1.938  1.176  0.751  0.617  0.528
  0.43   0.353  0.296  0.281  0.192  0.136  0.113  0.106  0.097  0.08
  0.058  0.044  0.038  0.03   0.027  0.023  0.015  0.011  0.009  0.008
  0.008  0.005  0.003  0.001  0.001  0.     0.     0.     0.     0.
  0.     0.     0.     0.     0.     0.     0.     0.     0.     0.
  0.     0.     0.     0.     0.     0.     0.   ]
```

From the above result the following conclusions can be drawn:

The first PC alone has the ability to differentiate between different observations in the dataset (explained variance / information content) which was possessed by more than 31 original variables taken together. So loosely it can be said that the information content in the first PC is equivalent to the information content in almost 32 original variables.

The second PC alone can do the task of almost 8 original variables in explaining the variance in the dataset / containing the information associated with the dataset, the third PC can do the task of more than 4 original variables and so on.

Along expected lines, the sum of the Eigen values or explained variances of all the PCs was a number very close to the total number of PCs / total number of original variables (57.0869). The physical interpretation of this is -- all the PCs taken together can explain the variance in the dataset as effectively as the 57 original variables does.

Let us have a glimpse at the Eigen vectors / coefficients of the PCs.

```
[[ 0.15602058  0.16711763  0.16555318 ...  0.13219224  0.15037558
   0.1310662 ]
 [-0.12634653 -0.08967655 -0.10491237 ...  0.05081332 -0.06536455
  -0.07384742]
 [-0.00269025  0.05669762  0.03874947 ... -0.07871987  0.11182732
   0.1025525 ]
 ...
 [-0.         -0.17278849 -0.09520952 ...  0.00987322 -0.04362296
  -0.0207041 ]
 [-0.         -0.0116324  -0.0814326  ...  0.04647201 -0.17212428
   0.03763315]
 [ 0.          0.18260602  0.03874463 ... -0.00370238 -0.05681626
  -0.03729932]]
```

TABLE 2.9: Eigen Vectors / Coefficients of the Principal Components

The Eigen vectors represents a 57 X 57 matrix which contains the coefficients / loadings of each PC from each of the original variables.

Now in order to determine the optimum number of principal components (PCs) to be retained without losing much information / ability to explain the variance in the dataset, three different approaches were followed with appropriate threshold metrics.

ELBOW METHOD / SCREE PLOT APPROACH

The Eigen values / explained variance of each PC was plotted as follows.

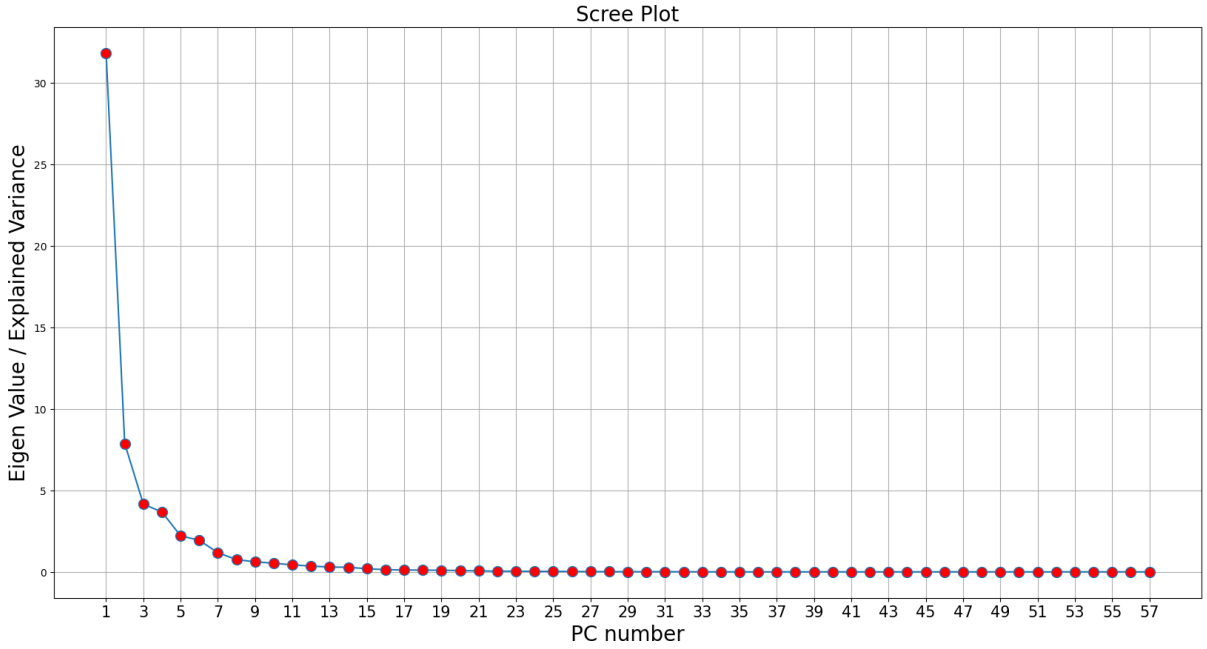

FIGURE 2.7 : Scree Plot of Eigen Values vs PC Number

From this graph we see that there is a bend / elbow / change in slope (curve flattens out) at around the 7th PC, which means that if we retain more than 7 PCs, the additional PCs do not add more explanatory power / information content to our model.

KAISER RULE APPROACH

According to this method we will retain only the PCs with Eigen values greater than 1 and discard the rest.

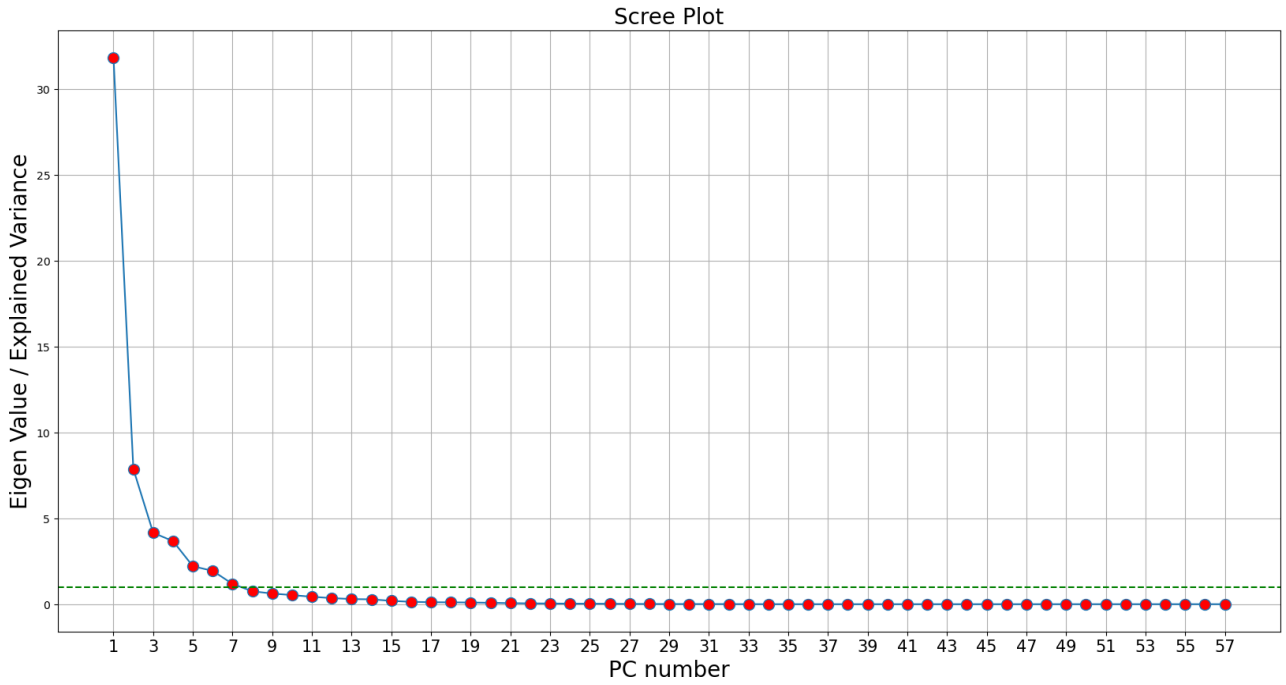

FIGURE 2.8 : Scree Plot for application of Kaiser rule

The horizontal line demarcates the PCs whose Eigen values are greater than 1 and less than 1 respectively. So only the PCs lying above the line were considered (total = 7) as their ability to explain the variance in the dataset was greater than equal to that of at least one original variable. The remaining PCs were discarded as they were not even as efficient as 1 original variable in explaining the variance in the dataset.

CUMULATIVE EXPLAINED VARIANCE RATIO APPROACH

The explained variance ratio for each PC was calculated as follows

Explained variance ratio = Eigen value / Sum of Eigen values of all PCs

The explained variance ratios of the PCs were as follows.

```
[0.55726063, 0.13784435, 0.07275295, 0.06426418, 0.03865049,
 0.03395169, 0.02060239, 0.01315764, 0.01080859, 0.00925395,
 0.00752912, 0.00619102, 0.00518772, 0.00492695, 0.00336593,
 0.00238693, 0.00198618, 0.00186207, 0.00170415, 0.00140318,
 0.0010091 , 0.00077765, 0.00066372, 0.00051912, 0.00047434,
 0.00041069, 0.00025418, 0.00019242, 0.00016317, 0.0001425 ,
 0.00013825, 0.00008804, 0.0000455 , 0.00001871, 0.0000125 ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        ])
```

So the first PC can explain 55.7 % of the variance in the dataset or in other words the first PC alone captures 55.7 % of the information content in the dataset.

The second PC explains 13.78 % of the variance in the data, the third PC explains 7.2 % and so on.

Now let us see the values of the cumulative explained variances.

```
[0.55726063, 0.69510499, 0.76785794, 0.83212212, 0.87077261,
 0.9047243 , 0.92532669, 0.93848433, 0.94929292, 0.95854687,
 0.96607599, 0.97226701, 0.97745473, 0.98238168, 0.98574761,
 0.98813454, 0.99012071, 0.99198278, 0.99368693, 0.99509011,
 0.99609921, 0.99687687, 0.99754058, 0.9980597 , 0.99853404,
 0.99894473, 0.99919891, 0.99939134, 0.9995545 , 0.99969701,
 0.99983525, 0.99992329, 0.9999688 , 0.9999875 , 1.        ,
 1.        , 1.        , 1.        , 1.        , 1.        ,
 1.        , 1.        , 1.        , 1.        , 1.        ,
 1.        , 1.        , 1.        , 1.        , 1.        ,
 1.        , 1.        , 1.        , 1.        , 1.        ,
 1.        , 1.        ])
```

These values can be interpreted as follows.

The first PC alone can explain 55.7 % of the variance in the data.

The first two PCs together can explain 69.5 % of the variance in the data.

The first three PCs together can explain 76.7 % of the variance in the data, and so on.

We will retain a number of PCs which together can explain at least 90 % of the variance in the data and drop the remaining PCs.

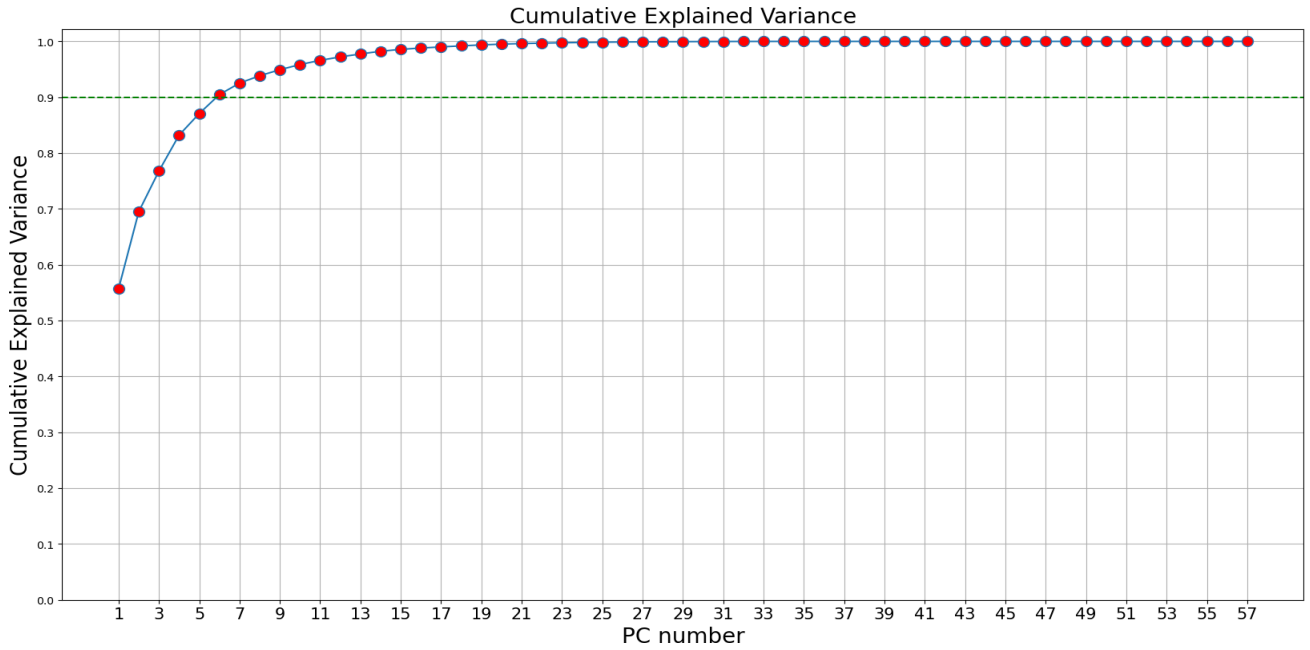

FIGURE 2.9 : Cumulative Explained Variance vs PC number

So if we want to retain at least 90% of the information in the dataset, we have to retain 7 PCs and drop the rest.

So the optimum number of principal components (PCs) is 7 and we achieve a dimensionality reduction from 57 to 7 through PCA and that too at the cost of less than 10 % of the information content in the dataset.

After deciding on the optimum number of PCs the PCA was repeated with no of PCs = 7.

Let us take a glimpse at the final transformed dataset in terms of the values of the top 7 PCs.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
0	-4.617263	0.138116	0.328545	1.543697	0.353737	-0.420947	-0.010386
1	-4.771662	-0.105865	0.244449	1.963214	-0.153884	0.417310	-0.023121
2	-5.964836	-0.294347	0.367394	0.619542	0.478199	0.276581	0.069555
3	-6.280796	-0.500384	0.212701	1.074515	0.300799	0.051158	-0.250539
4	-4.478566	0.894154	1.078277	0.535556	0.804065	0.341676	-0.092335
...	...	...	...	...	...	...	...
635	-6.262088	-0.854414	0.242575	1.174113	0.063815	-0.159470	-0.372274
636	-5.767714	-0.900436	0.168051	1.102774	0.055179	-0.156457	-0.511471
637	-6.294625	-0.638127	0.107483	1.368187	0.153746	0.141146	-0.344500
638	-6.223192	-0.672320	0.271326	1.143493	0.060440	-0.115682	-0.383034
639	-5.896236	-0.937170	0.349218	1.114861	0.149104	-0.154543	-0.384508

640 rows × 7 columns

TABLE 2.10 : PC SCORES for the selected PCs

These PCs are expected to be uncorrelated unlike the original 57 features. Let us verify that using a heatmap.

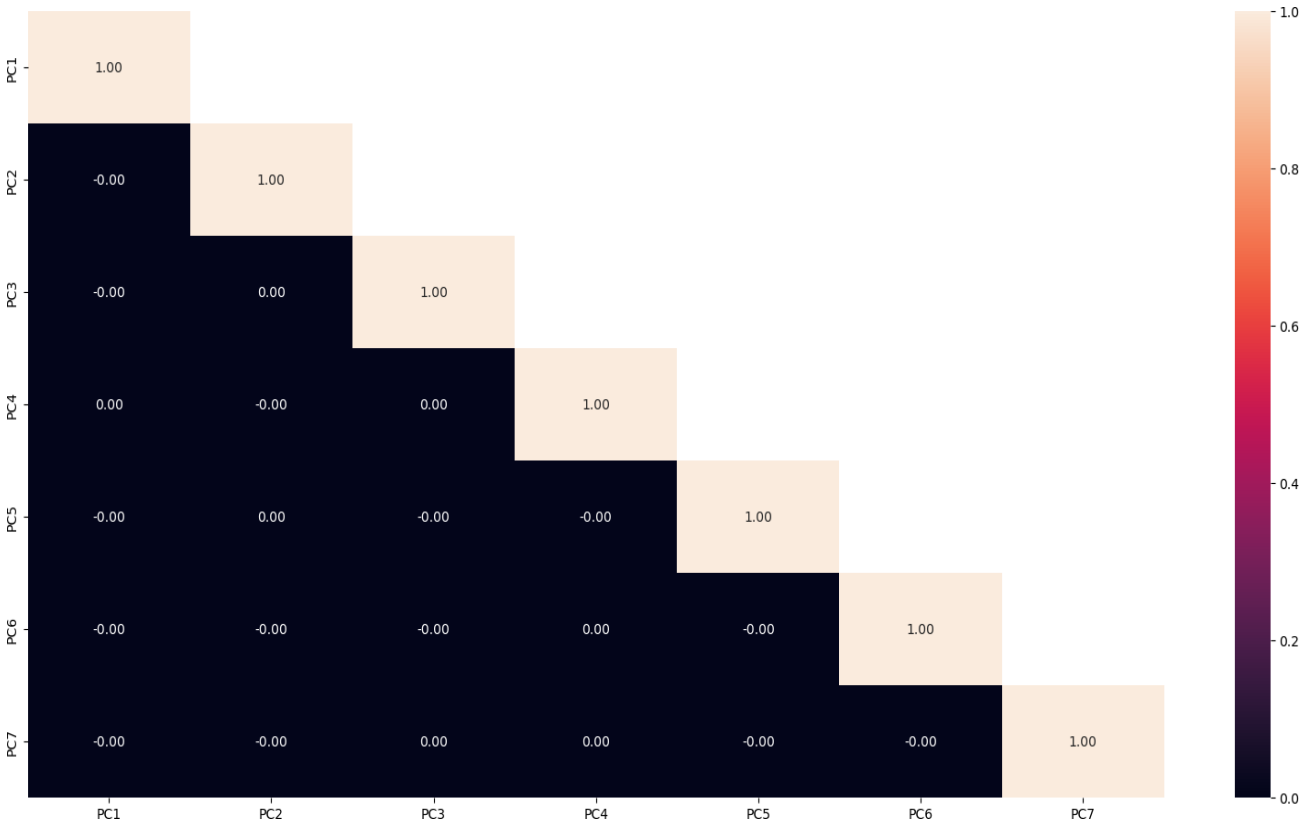

FIGURE 2.10 : Heatmap for the selected PCs

Therefore other than dimensionality reduction, PCA does another important task of getting rid of the correlations among the independent variables. This ensures that when the transformed data is being fed to a model, the same information is not supplied multiple times (multicollinearity).

The Eigen Values (explained variances) of the top 7 PCs are 31.814, 7.869, 4.153, 3.669, 2.207, 1.938, and 1.176.

Now in order to examine the influence of the original variables in each PC, the relevant loading / coefficient matrix was constructed using the Eigen vectors corresponding to the top 7 PCs.

	No_HH	TOT_M	TOT_F	M_06	F_06	M_SC	F_SC	M_ST	F_ST	M_LIT	F_LIT	M_ILL	F_ILL	TOT_WORK_M	TOT_WORK_F	I
PC1	0.16	0.17	0.17	0.16	0.16	0.15	0.15	0.03	0.03	0.16	0.15	0.16	0.17	0.16	0.15	
PC2	-0.13	-0.09	-0.10	-0.02	-0.02	-0.05	-0.05	0.03	0.03	-0.12	-0.15	-0.01	-0.01	-0.13	-0.09	
PC3	-0.00	0.06	0.04	0.06	0.05	0.00	-0.03	-0.12	-0.14	0.08	0.12	-0.02	-0.09	0.05	-0.06	
PC4	-0.13	-0.02	-0.07	0.01	0.01	0.01	-0.03	-0.22	-0.23	-0.04	-0.06	0.03	-0.08	-0.04	-0.23	
PC5	-0.01	-0.03	-0.01	-0.05	-0.04	-0.17	-0.16	0.43	0.44	-0.01	0.06	-0.10	-0.12	-0.02	-0.04	
PC6	0.00	-0.07	-0.04	-0.16	-0.15	-0.06	-0.04	0.22	0.23	-0.06	-0.05	-0.12	-0.03	-0.00	0.11	
PC7	-0.12	0.09	-0.00	0.17	0.17	-0.00	-0.08	0.41	0.36	0.05	-0.02	0.20	0.03	0.05	-0.12	

TABLE 2.11 : Coefficient / Loadings of the selected PCs

The loadings from the variables which contributed the most towards a particular PC were highlighted in the corresponding row by using a suitable python function. Based on the above relative importance of the different loadings from different variables let us discuss about the physical implications of the selected PCs.

PC1

This PC captures the maximum information in the dataset. It is influenced the most (loading = 0.17) by the variables like total no of males, total no of females, female illetarates, Marginal Cultivator Population 3-6 (Male), etc. Therefore this PC captures information related to the

overall population of different areas as well as population based on education level as well as occupation category.

PC2

This PC is influenced the most (loading = 0.27) by Marginal Cultivator Population (Male) and Marginal Agricultural Labourers 3-6 Male. So this PC talks mainly about male population occupied in the marginal occupations.

PC3

This PC also deals mostly with marginal profession occupants but it emphasises mostly on the female category unlike PC2 which focusses on the male category.

PC4

This PC deals with male population involved in marginal household industries and other marginal occupations.

PC5

This PC is mostly influenced by the Schedule Tribe category people (both male and female).

PC6

This PC is mostly about the female workers in main household industries, marginal household industries and marginal other industries.

PC7

This is mostly about the male population occupied in main cultivation.

LINEAR EQUATION FOR THE FIRST PRINCIPAL COMPONENT

Each PC is a linear combination of the original features. Using this concept and the loading / coefficient matrix we can frame a linear equation for PC1 in terms of the original variables as follows.

PC1 = (0.16) * No_HH + (0.17) * TOT_M + (0.17) * TOT_F + (0.16) * M_06 + (0.16) * F_06 + (0.15) * M_SC + (0.15) * F_SC + (0.03) * M_ST + (0.03) * F_ST + (0.16) * M_LIT + (0.15) * F_LIT + (0.16) * M_ILL + (0.17) * F_ILL + (0.16) * TOT_WORK_M + (0.15) * TOT_WORK_F + (0.15) * MAINWORK_M + (0.12) * MAINWORK_F + (0.1) * MAIN_CL_M + (0.07) * MAIN_CL_F + (0.11) * MAIN_AL_M + (0.07) * MAIN_AL_F + (0.13) * MAIN_HH_M + (0.08) * MAIN_HH_F + (0.12) * MAIN_OT_M + (0.11) * MAIN_OT_F + (0.16) * MARGWORK_M + (0.16) * MARGWORK_F + (0.08) * MARG_CL_M + (0.05) * MARG_CL_F + (0.13) * MARG_AL_M + (0.11) * MARG_AL_F + (0.14) * MARG_HH_M + (0.13) * MARG_HH_F + (0.16) * MARG_OT_M + (0.15) * MARG_OT_F + (0.16) * MARGWORK_3_6_M + (0.16) * MARGWORK_3_6_F + (0.17) * MARG_CL_3_6_M + (0.16) * MARG_CL_3_6_F + (0.09) * MARG_AL_3_6_M + (0.05) * MARG_AL_3_6_F + (0.13) * MARG_HH_3_6_M + (0.11) * MARG_HH_3_6_F + (0.14) * MARG_OT_3_6_M + (0.12) * MARG_OT_3_6_F + (0.15) * MARGWORK_0_3_M + (0.15) * MARGWORK_0_3_F + (0.15) * MARG_CL_0_3_M + (0.14) * MARG_CL_0_3_F + (0.05) * MARG_AL_0_3_M + (0.04) * MARG_AL_0_3_F + (0.12) * MARG_HH_0_3_M + (0.12) * MARG_HH_0_3_F + (0.14) * MARG_OT_0_3_M + (0.13) * MARG_OT_0_3_F + (0.15) * NON_WORK_M + (0.13) * NON_WORK_F