Market Segmentation using Data Analysis and Machine Learning

Analyzing the Ed-Tech Market in India using Segmentation analysis to find a feasible strategy to enter the market

Presented By

Team-Andre

Andre Isaac Nazareth

Swapnil Biswas

Ketan Chandra

Gangisetty Venkata Naga Sai Krishna Vamsi

Overview

EdTech (a combination of "education" and "technology") refers to hardware and software designed to enhance teacher-led learning in classrooms and improve students' education outcomes.

Edtech is the practice of introducing IT tools into the classroom to create a more engaging, inclusive and individualized learning experience.

Classrooms are no longer the typical chalk-blackboard type of arrangement that they used to be. We now have interactive smartboards, online classes from the comfort of your home and whatnot. While the industry was already moving in this direction, the pandemic forced us to adapt even faster than ever.

In-classroom tablets, interactive projection screens and whiteboards, online content delivery, and MOOCs are all examples of EdTech.

AI and ML have always been an essential part of EdTech. A typical example of the use of AI in EdTech would be how the questions get tougher and tougher if a student keeps on correctly solving the questions. Grammarly, a popular writing software relies on AI and Natural Language Processing (NLP) to function.

While EdTech is still in its early stages of development, it has a huge scope and could definitely be one of the next big things in the future.

Introduction

In this project, we have used AI and ML techniques and algorithms to determine which would be the best Indian states to start an edtech venture, based on factors such as the percentage of literate population and the percentage of households with access to internet.

We have analyzed the Ed-Tech Market in India using Segmentation analysis and have come up with a feasible strategy to enter the market, targeting the segments most likely to use the product in accordance with the Innovation Adoption Life Cycle.

To arrive at this conclusion, we considered various factors including but not limited to Geographic, Demographic, Psychographic and Behavioral factors.

What is Market Segmentation?

In marketing, market segmentation is the process of dividing a broad consumer or business market, normally consisting of existing and potential customers, into sub-groups of consumers (known as segments) based on some type of shared characteristics.

Types of market segmentation

1. Demographic segmentation

Demographic is one of the simplest and most commonly used forms of segmentation. Demographic segmentation sorts a market by elements such as age, education, income, family size, race, gender, occupation, and nationality.

2. Geographic segmentation

Geographic segmentation creates different target customer groups based on geographical boundaries. The needs, preferences, and interests of a potential

customer may be different from another customer living in a different geographical area. So it is important to understand the climates and geographic regions of customer groups as that can help determine where to sell and advertise, as well as where to expand.

3. Firmographic Segmentation

Firmographic Segmentation is similar to demographic segmentation, except that demographic look at individuals while firmographics looks at organizations. Factors like company size, number of employees would come into play when determining the difference between addressing a small business from addressing an enterprise corporation.

4. Behavioral Segmentation

Dividing the market based on behaviors and decision-making patterns such as purchase, consumption, lifestyle, and usage comes under behavioral segmentation. Segmenting markets based on purchase behaviors enables marketers to develop a more targeted approach because you can focus on what you know they, and are therefore more likely to buy.

Data Source

Our dataset was obtained from kaggle (www.kaggle.com) which is a subsidiary of Google LLC and an online community of data scientists and machine learning practitioners.

Data Collection was not an easy task for this project. The demographic and geographic data were freely available but data on behavioural and psychographic were not. After days of research and exploration we concluded that this data was not available freely and extensive surveys would be needed to be conducted in order to obtain them.

What is Data Pre-processing?

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put it in a formatted way. So for this, we use data preprocessing tasks.

Why do we need Data Preprocessing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data

- Splitting dataset into training and test set
- Feature scaling

What is Exploratory Data Analysis?

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

It is a good practice to understand the data first and try to gather as many insights from it. EDA is all about making sense of data in hand,before getting them dirty with it.

Getting Started

Importing the Libraries

```
[ ]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     import plotly.express as px
     from sklearn.preprocessing import LabelEncoder
     from sklearn.preprocessing import StandardScaler
```

Reading our Dataset

```
[ ]  df=pd.read_csv('dataset.csv')
     df.head()
```

	District code	State name	District name	Population	Male	Female	Literate	Male_Literate	Female_Literate	SC	...
0	1	JAMMU AND KASHMIR	Kupwara	870354	474190	396164	439654	282823	156831	1048	...
1	2	JAMMU AND KASHMIR	Badgam	753745	398041	355704	335649	207741	127908	368	...
2	3	JAMMU AND KASHMIR	Leh(Ladakh)	133487	78971	54516	93770	62834	30936	488	...
3	4	JAMMU AND KASHMIR	Kargil	140802	77785	63017	86236	56301	29935	18	...
4	5	JAMMU AND KASHMIR	Punch	476835	251899	224936	261724	163333	98391	556	...

5 rows × 118 columns

Checking for Null Values and Statistics

```
[ ] df.isna().sum()
```

```
District code               0
State name                  0
District name               0
Population                  0
Male                        0
Female                      0
Literate                    0
Households_with_Internet    0
Households_with_Computer    0
Rural_Households            0
Urban_Households            0
Households                  0
Age_Group_0_29              0
Age_Group_30_49             0
Age_Group_50                0
Age not stated              0
dtype: int64
```

```
[ ] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 640 entries, 0 to 639
Data columns (total 16 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   District code             640 non-null    int64 
 1   State name                640 non-null    object
 2   District name             640 non-null    object
 3   Population                640 non-null    int64 
 4   Male                      640 non-null    int64 
 5   Female                    640 non-null    int64 
 6   Literate                  640 non-null    int64 
 7   Households_with_Internet  640 non-null    int64 
 8   Households_with_Computer  640 non-null    int64 
 9   Rural_Households          640 non-null    int64 
 10  Urban_Households          640 non-null    int64 
 11  Households                640 non-null    int64 
 12  Age_Group_0_29            640 non-null    int64 
 13  Age_Group_30_49           640 non-null    int64 
 14  Age_Group_50              640 non-null    int64 
 15  Age not stated            640 non-null    int64 
dtypes: int64(14), object(2)
memory usage: 80.1+ KB
```

```
[ ] df.describe()
```

	District code	Population	Male	Female	Literate	Households_with_Internet
count	640.000000	6.400000e+02	6.400000e+02	6.400000e+02	6.400000e+02	640.000000
mean	320.500000	1.891961e+06	9.738598e+05	9.181011e+05	1.193186e+06	12044.564062
std	184.896367	1.544380e+06	8.007785e+05	7.449864e+05	1.068583e+06	33573.205832
min	1.000000	8.004000e+03	4.414000e+03	3.590000e+03	4.436000e+03	17.000000
25%	160.750000	8.178610e+05	4.171682e+05	4.017458e+05	4.825982e+05	1406.750000
50%	320.500000	1.557367e+06	7.986815e+05	7.589200e+05	9.573465e+05	3489.000000
75%	480.250000	2.583551e+06	1.338604e+06	1.264277e+06	1.602260e+06	8820.500000
max	640.000000	1.106015e+07	5.865078e+06	5.195070e+06	8.227161e+06	430880.000000

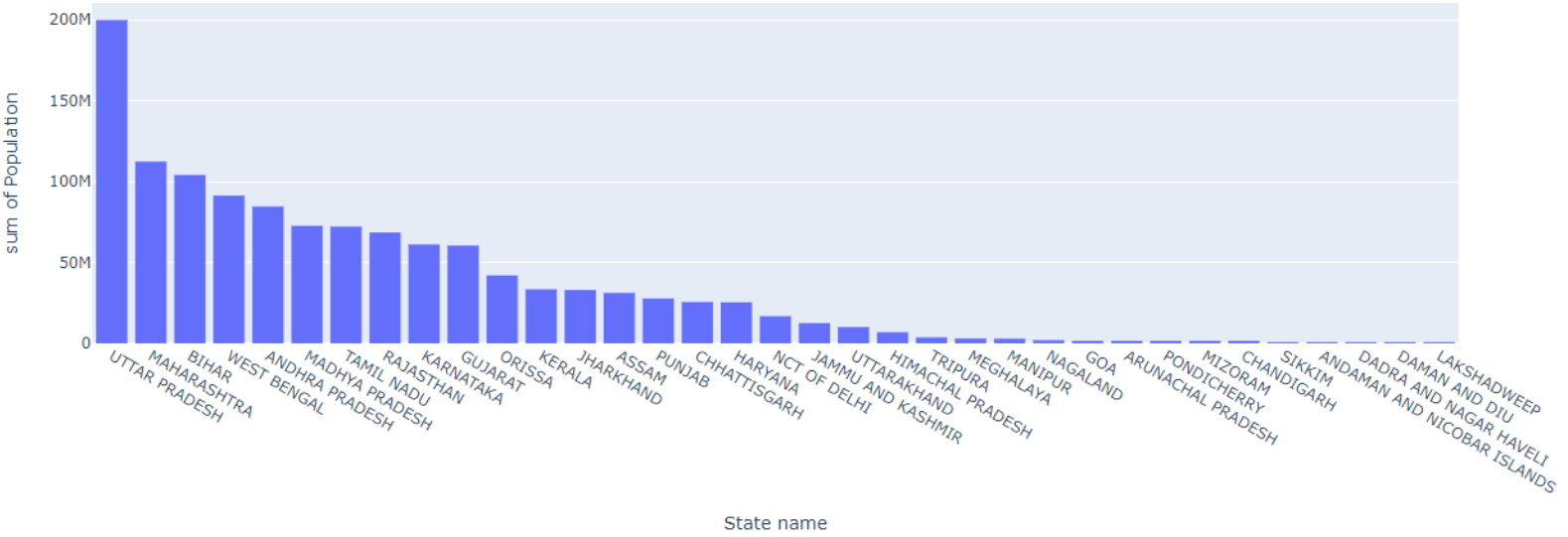
STATE WISE POPULATION ANALYSIS

In the past two years Indian edtech start-ups rose to the fore when the Indian government shut down educational institute amidst a deadly COVID-19 pandemic.

Despite having hundreds of bright students, smaller Indian cities and towns still lack quality education avenues.

With this project we are trying to bridge the gap for innovative edtech solutions in smaller town and cities.

STATE POPULATION ANALYSIS



We have performed a state population analysis and the majority of the states with high populations have a high number of literate people. This makes sense as the literate population is a sample subset of the entire population. Listing some of these states: -

1) **Uttar Pradesh**
2) **Maharashtra**
3) **West Bengal**
4) **Bihar**
5) **Tamil Nadu**
6) **Andhra Pradesh**

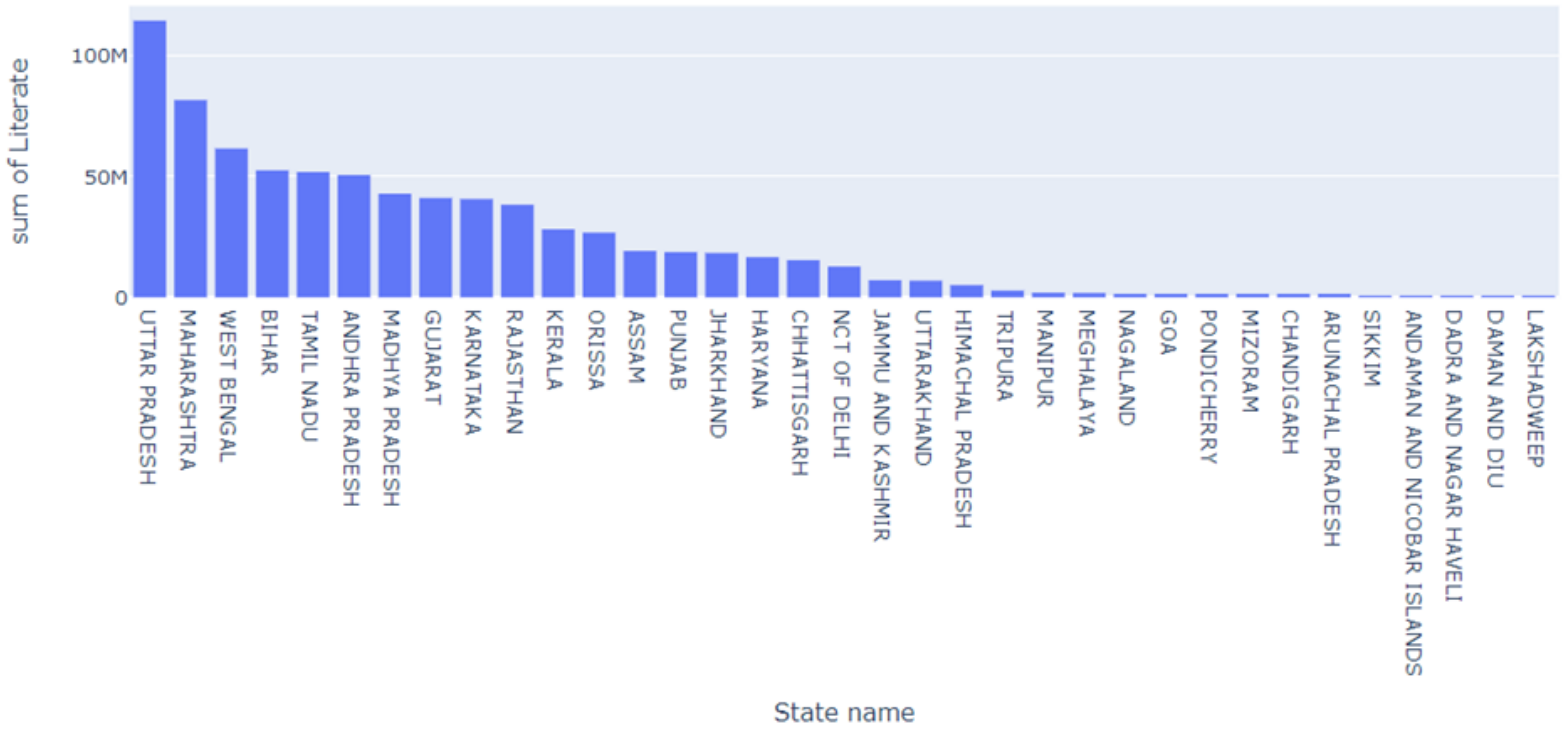
7) **Madhya Pradesh**

8) **Gujarat**

9) **Karnataka**

10) **Rajasthan**

LITERATE POPULATION ANALYSIS

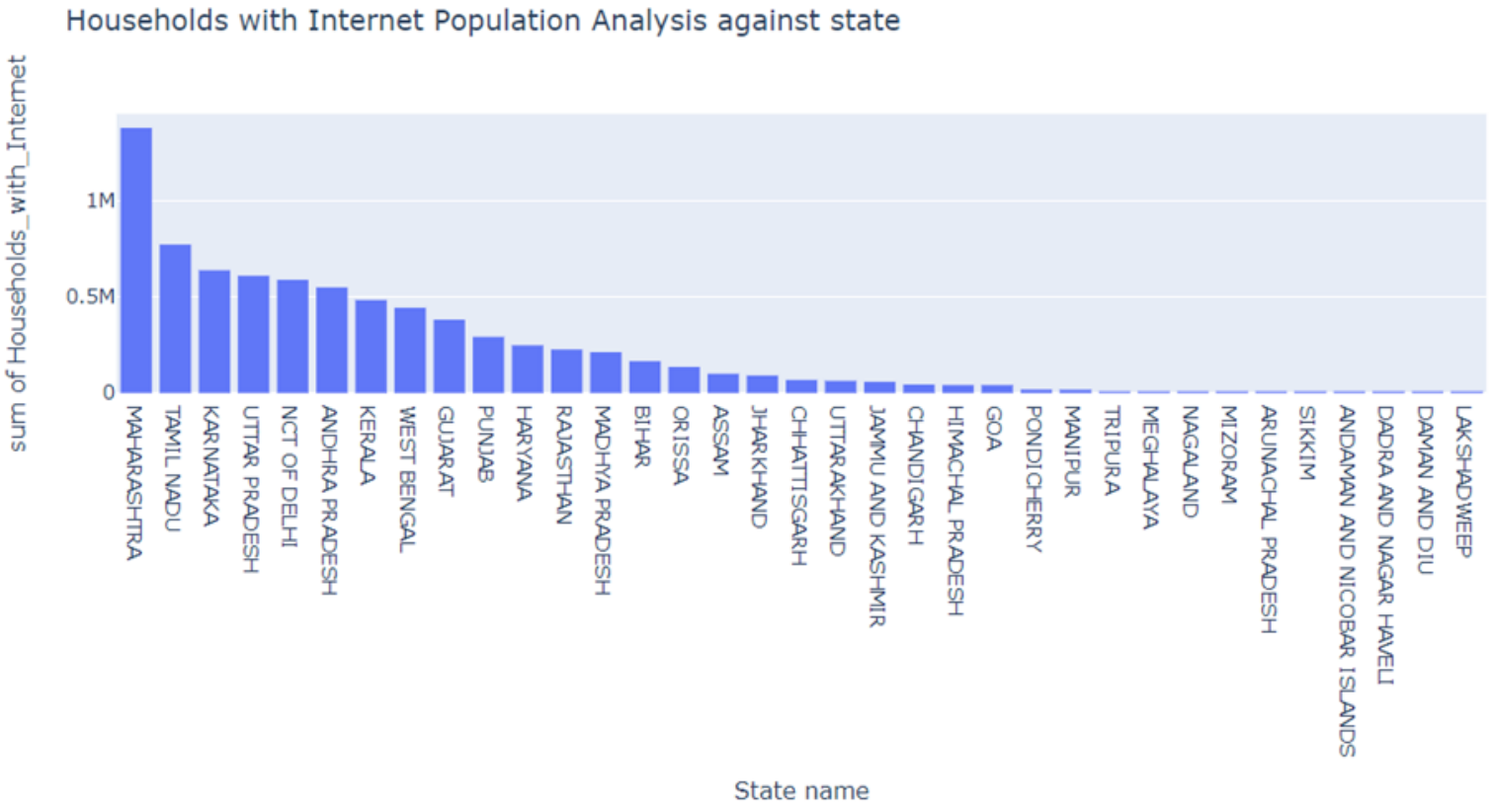


We have performed a literate population analysis from which we can come to the conclusion that majority of the states with high population have high number of literate people are: -

1. **Uttar Pradesh**
2. **Maharashtra**
3. **West Bengal**
4. **Bihar**

5. **Tamil Nadu**
6. **Andhra Pradesh**
7. **Madhya Pradesh**
8. **Gujarat**
9. **Karnataka**
10. **Rajasthan**

HOUSEHOLD WITH INTERNET POPULATION ANALYSIS



We have performed households with internet population analysis from which we can conclude that Maharashtra has the highest number of people with access to the internet. Listing The other population dense states with access to the internet: -

1. **Maharashtra**
2. **Tamil Nadu**
3. **Karnataka**

4. **Uttar Pradesh**
5. **NCT OF Delhi**
6. **Andhra Pradesh**
7. **Kerala**
8. **West Bengal**
9. **Gujarat**

These states would be the main target for the edtech start-ups

DISTRICT WISE POPULATION ANALYSIS

Using the conclusions drawn from the state population analysis we will be exploring which districts have maximum scope for educational technology in India.

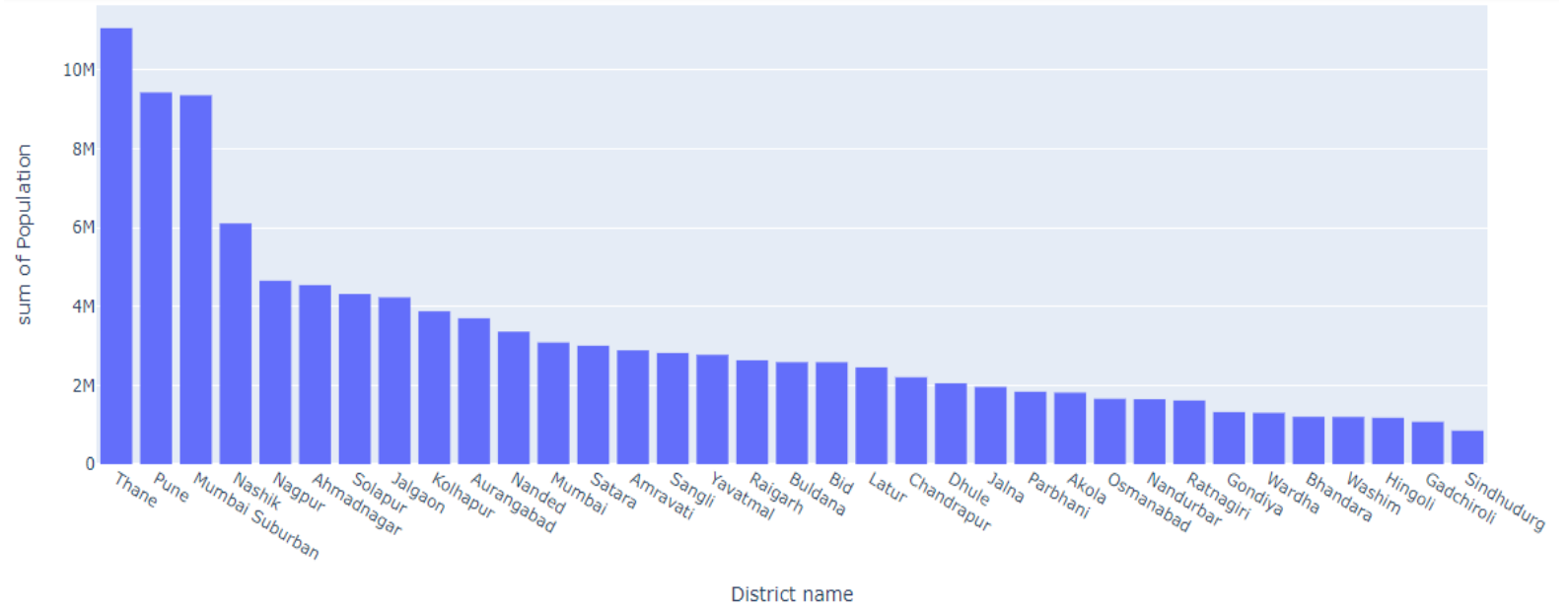
Defining a function to plot our histograms for district wise analysis of each state

```
[ ]  def district(state):
         temp=px.histogram(state,x="District name",y="Population",title='Population against Districts')
         temp.update_layout(barmode='stack', xaxis={'categoryorder':'total descending'})
         temp.show()

         temp=px.histogram(state,x="District name",y="Literate",title='Literate Population against Districts')
         temp.update_layout(barmode='stack', xaxis={'categoryorder':'total descending'})
         temp.show()

         temp=px.histogram(state,x="District name",y="Households_with_Internet")
         temp.update_layout(barmode='stack', xaxis={'categoryorder':'total descending'})
         temp.show()
```

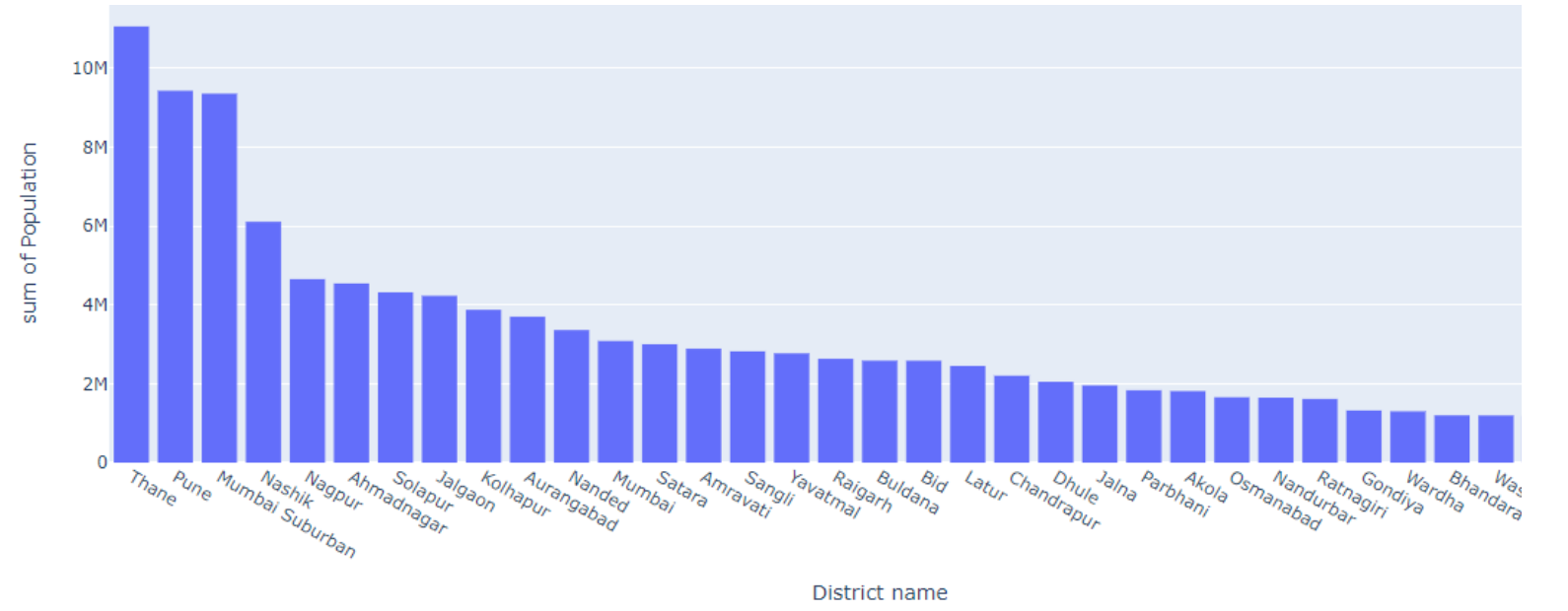
,

MAHARASHTRA DISTRICT ANALYSIS

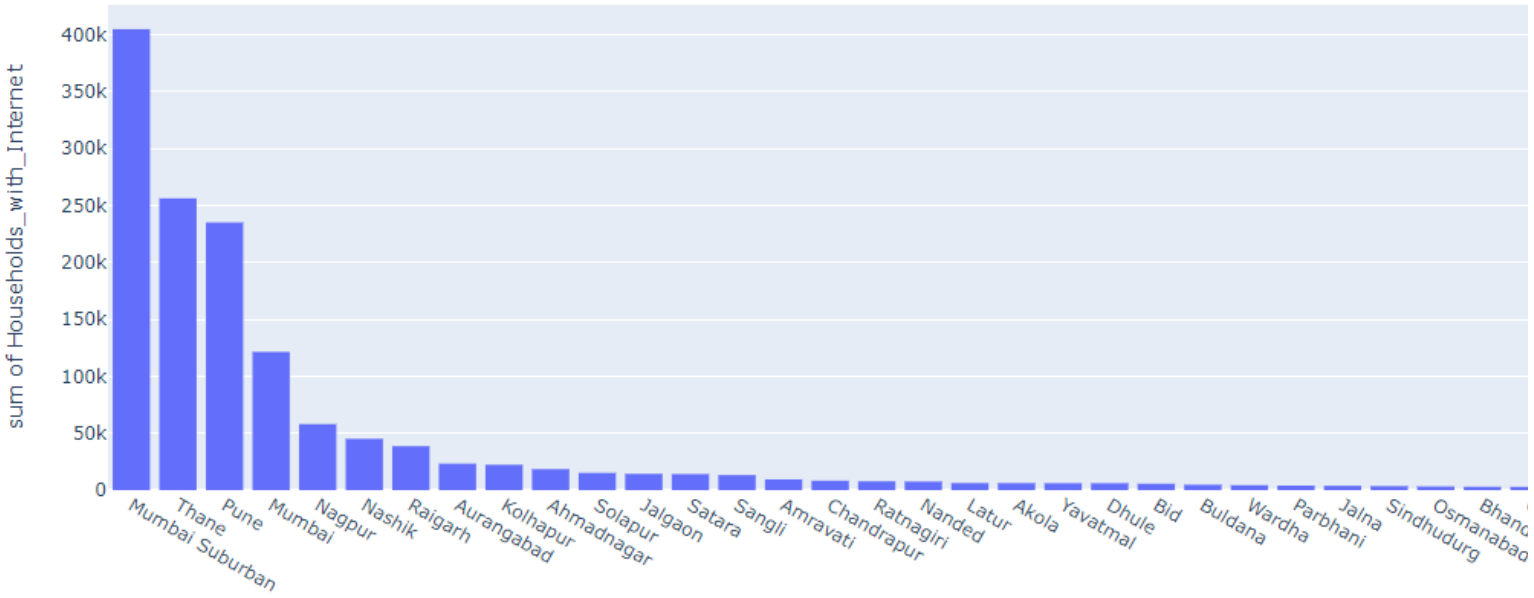
Population per district



Literate Population per district

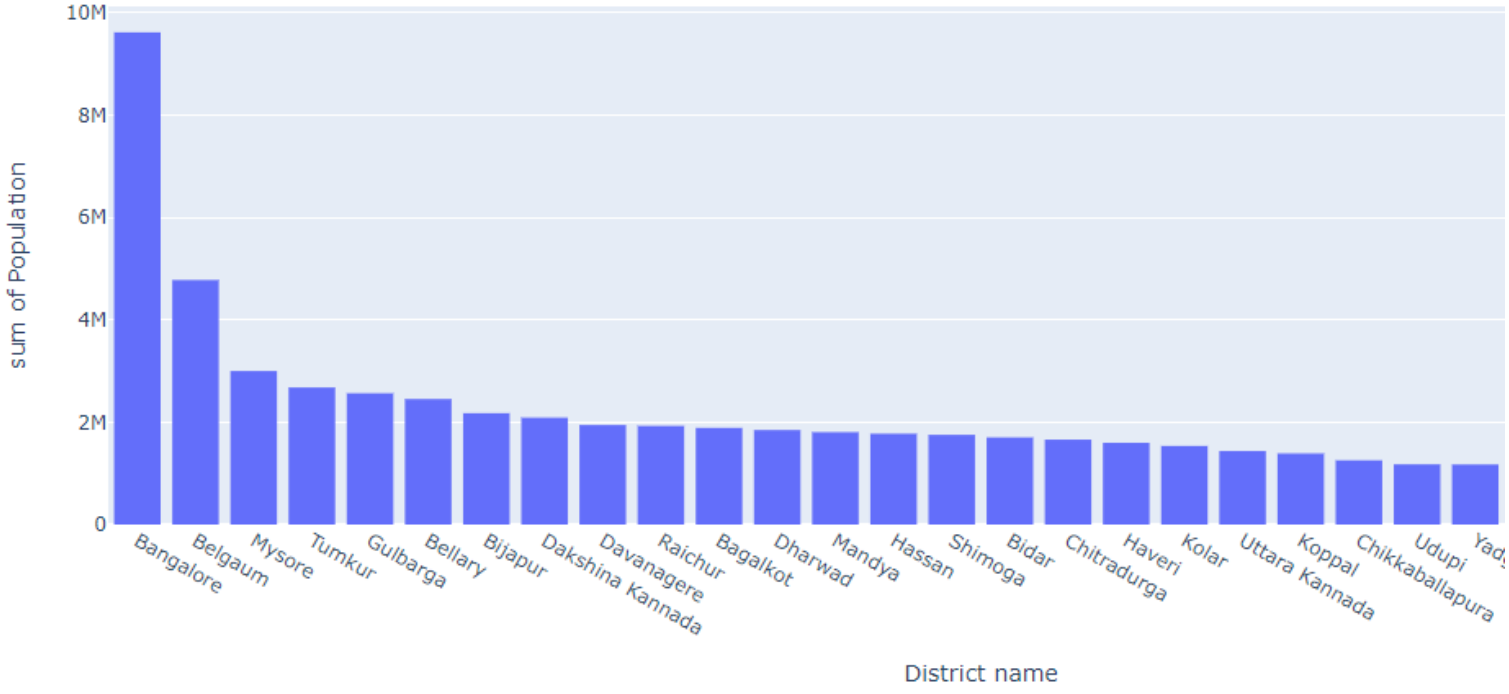


Household with Internet Access per District

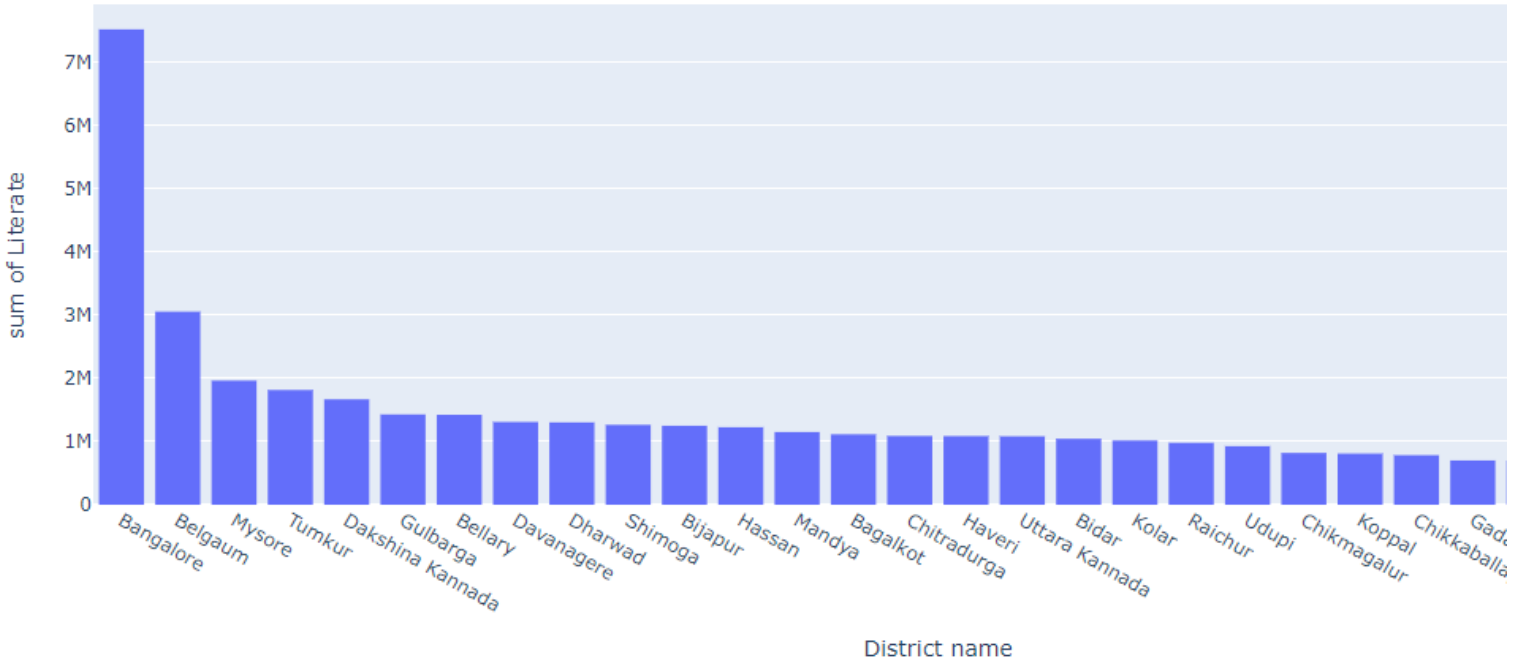


KARNATAKA DISTRICT ANALYSIS

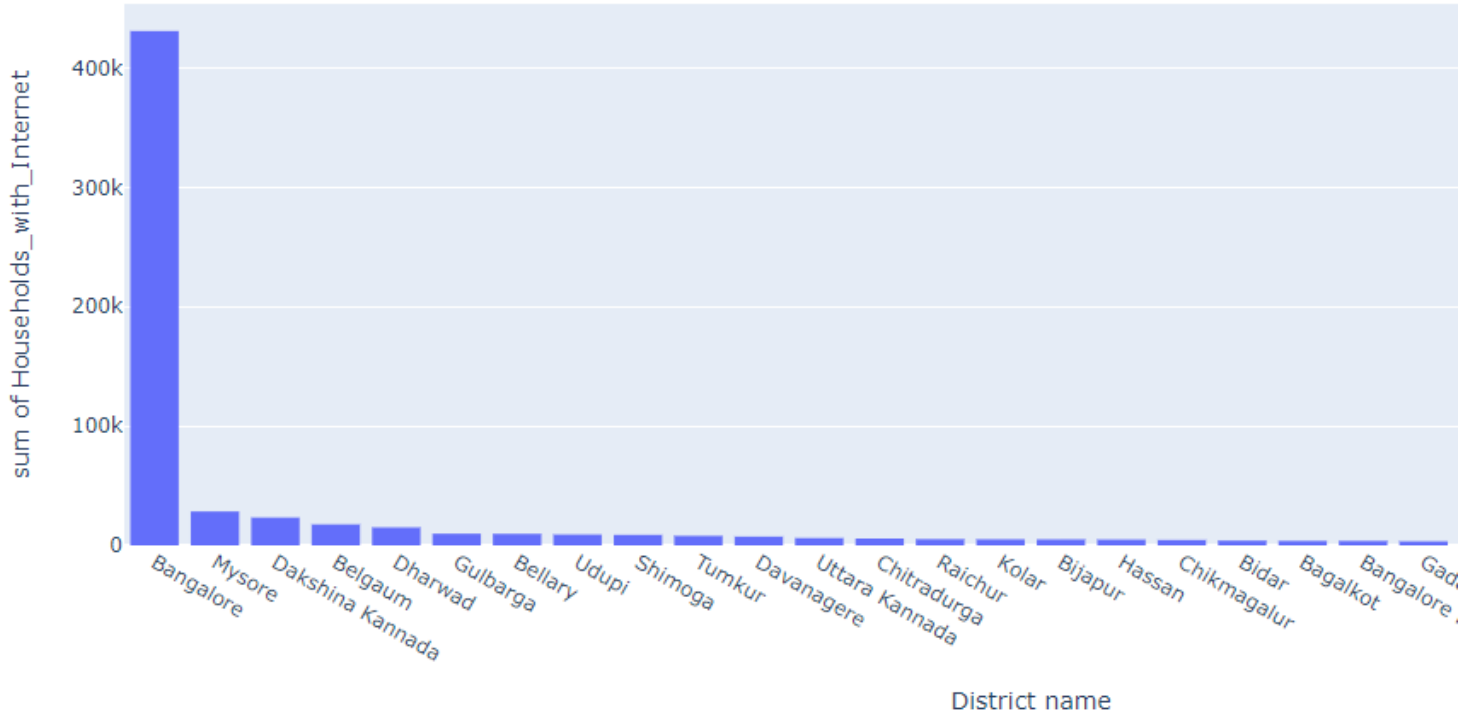
Population per district



Literate Population per district

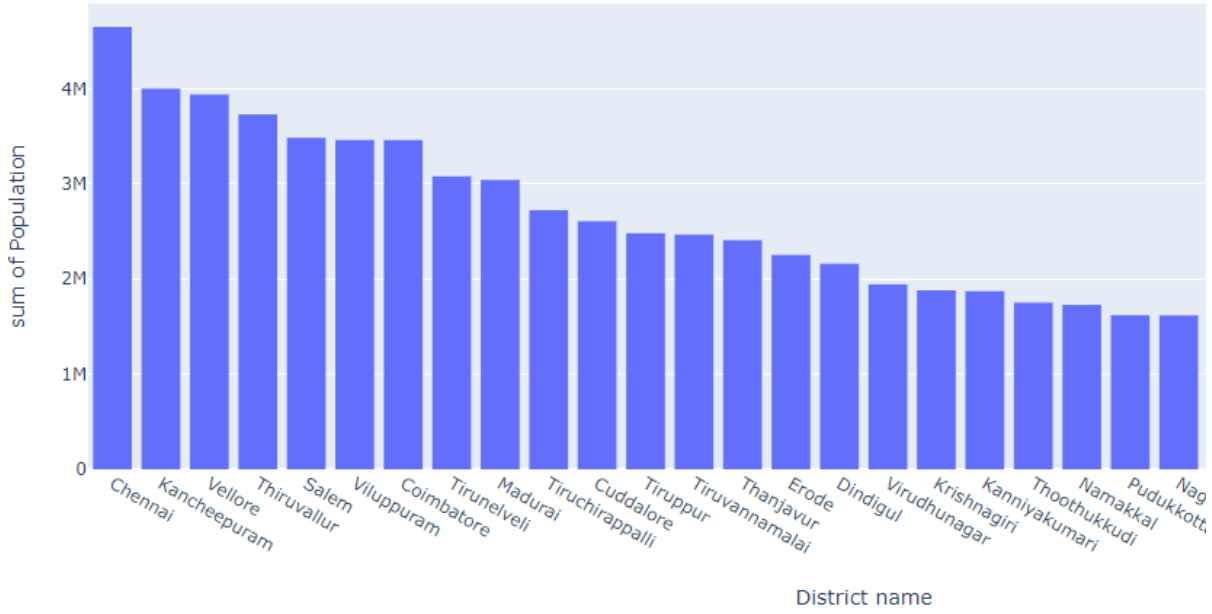


Household with Internet Access per District

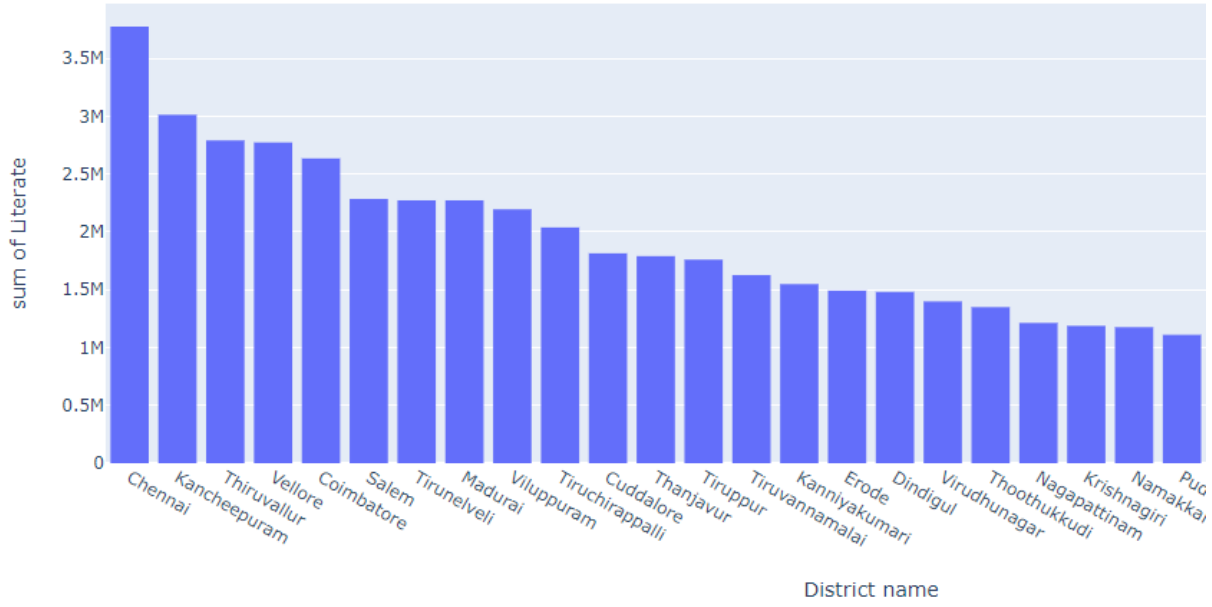


TAMIL NADU DISTRICT ANALYSIS

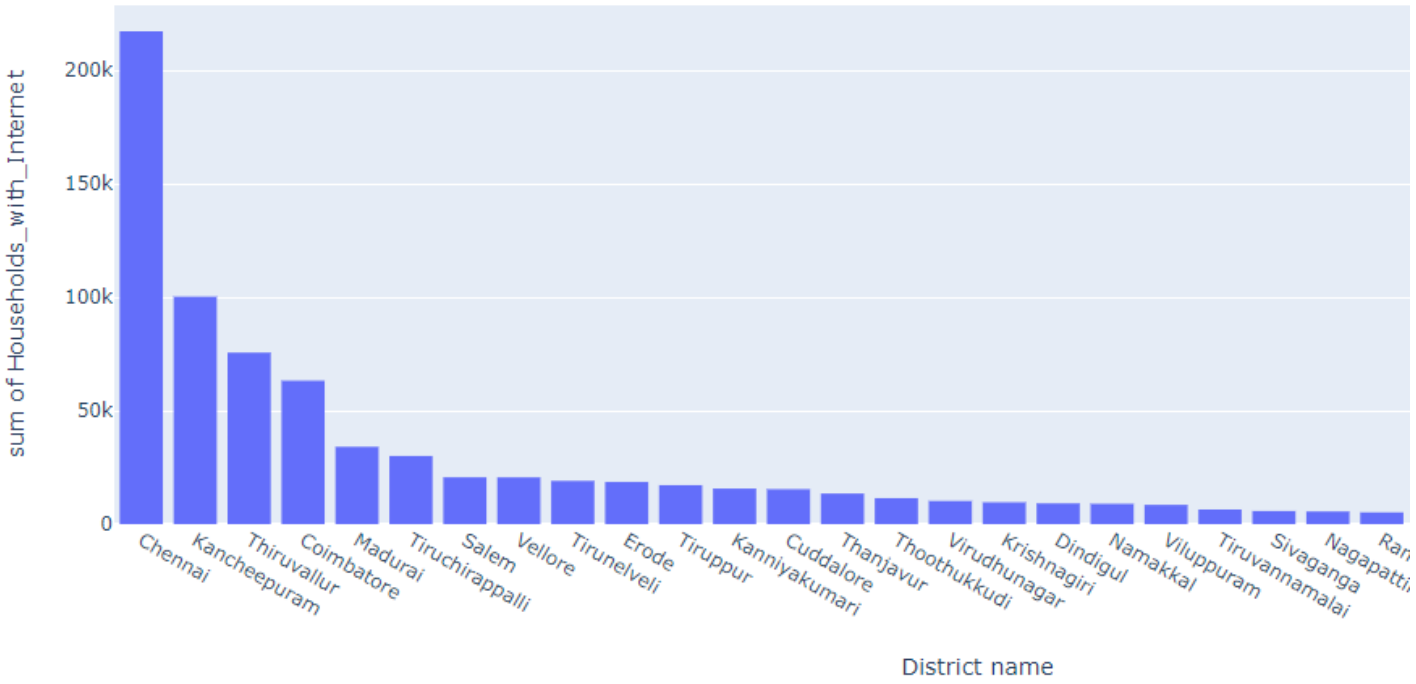
Population per district



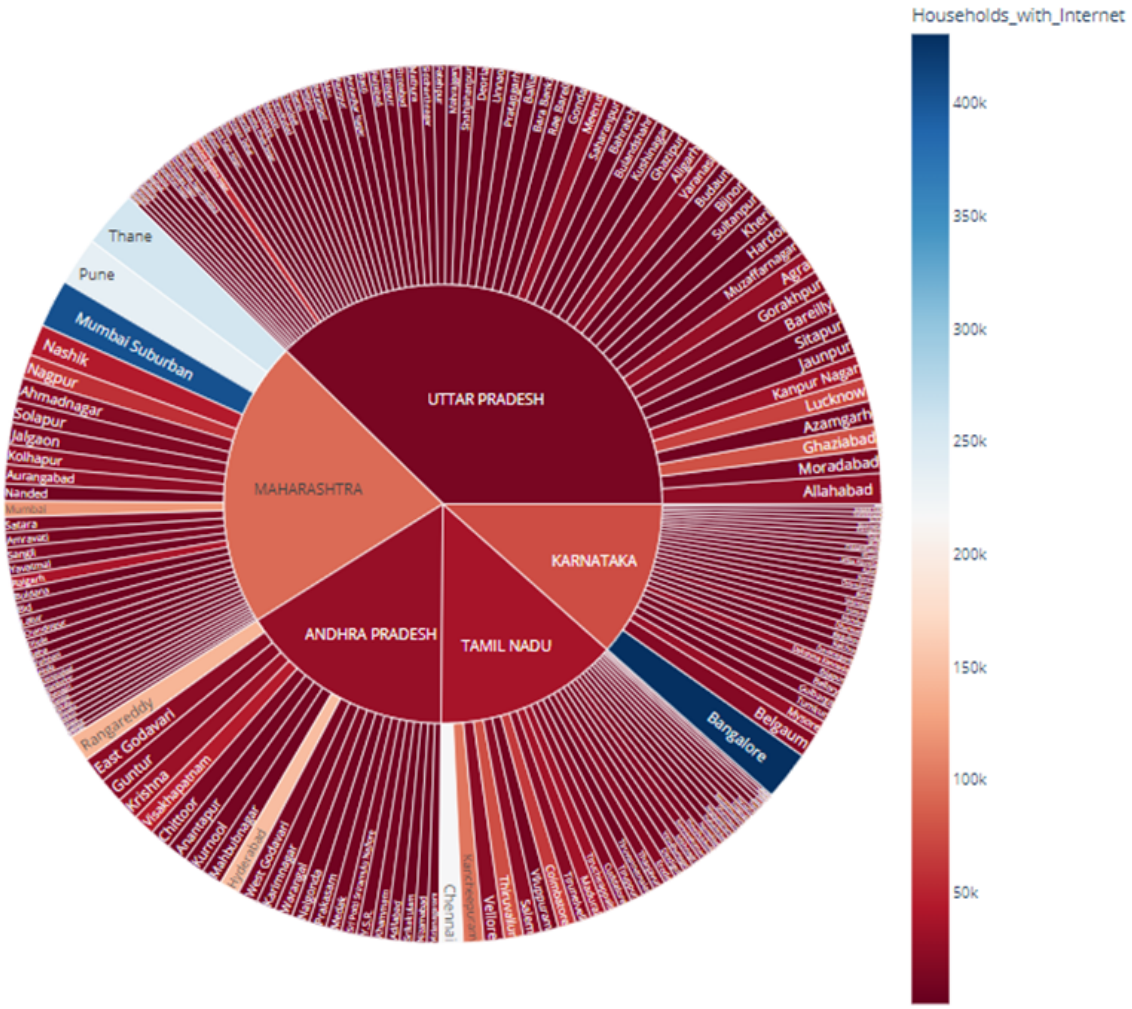
Literate Population per district



Household with Internet Access per District



Outcomes of District wise Analysis



In the above visualization we have shown the data of households with internet using sunburst visualization plot, from this we can observe that there are around 5 districts that can provide us with a great business opportunity. The places that have great business opportunities are Thane, Pune, Mumbai Suburban, Bangalore, Chennai.

In Thane there are around 256170 households with internet, in Pune there are around 235018 households with internet, in Mumbai there are around 404757 households with internet, in Bangalore there are around 430880 households with internet and in Chennai there are around 217368 households with internet. According to this, Bangalore has the highest population that is around 430880 of households with internet.

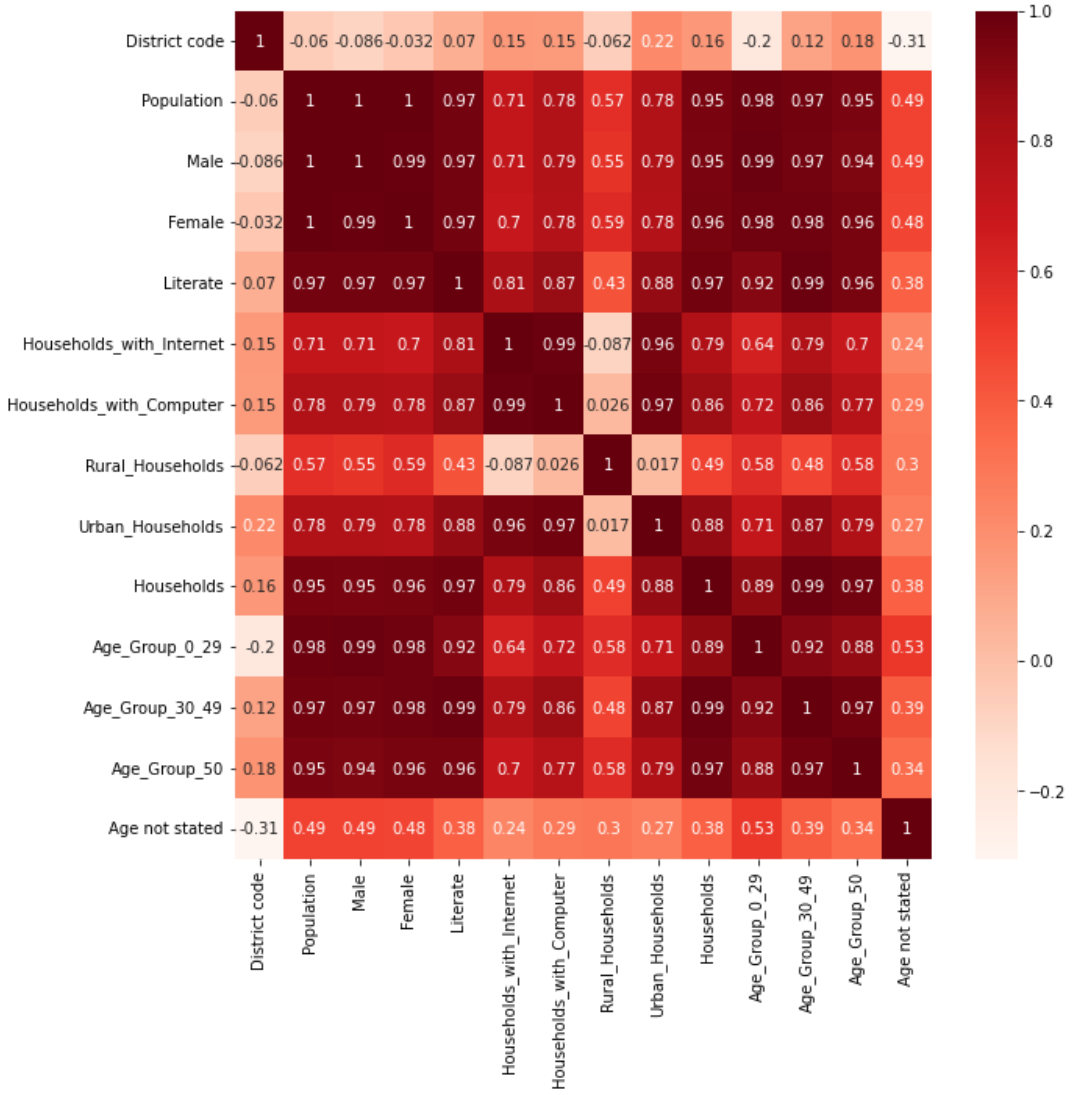
When coming to the state wide analysis Maharashtra has the highest population of households with internet.

Seaborn Heatmap

Seaborn is a library used for data visualisation based on the matplotlib library. It provides a high-level interface for drawing attractive and informative statistical graphics.

Heatmap is defined as a graphical representation of data using colours to visualise the value of the matrix. In this, to represent more common values or higher activities brighter colours basically reddish colours are used and to represent less common or activity values, darker colours are preferred. Heatmap is also defined by the name of the shading matrix. Heatmaps in Seaborn can be plotted by using the seaborn.heatmap() function.

```
plt.figure(figsize=(10,10))
sns.heatmap(data.corr(),cmap='Reds',annot=True)
```



We can see that the district code is giving us low correlation with all the attributes which makes sense. The district code is being used as a means of market segmentation based on geographic location only.

Label Encoding

Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering.

```
[ ]  le=LabelEncoder()

[ ]  data['State name'] = le.fit_transform(data['State name'])
     data['District name'] = le.fit_transform(data['District name'])
```

Feature Scaling

Feature scaling is a method used to normalise the range of independent variables or features of data. In data processing, it is also known as data normalisation and is generally performed during the data preprocessing step.

```
[ ]  data1=data
     scaler=StandardScaler()
     segmentation=scaler.fit_transform(data1)
     segmentation=pd.DataFrame(segmentation,columns=data1.columns)
```

Taking a look at the maximum values of our Scaled Data

```
[ ]  segmentation=pd.DataFrame(segmentation)
     print(segmentation.max())

     District code              1.128665
     State name                 1.043624
     District name              1.723006
     Population                 5.250251
     Male                       5.369545
     Female                     5.104793
     Literate                   5.575800
     Households_with_Internet   7.610601
     Households_with_Computer   7.588663
     Rural_Households           3.056791
     Urban_Households           6.509129
     Households                 5.788950
     Age_Group_0_29             5.054003
     Age_Group_30_49            5.675560
     Age_Group_50               4.475483
     Age not stated             5.391960
     dtype: float64
```

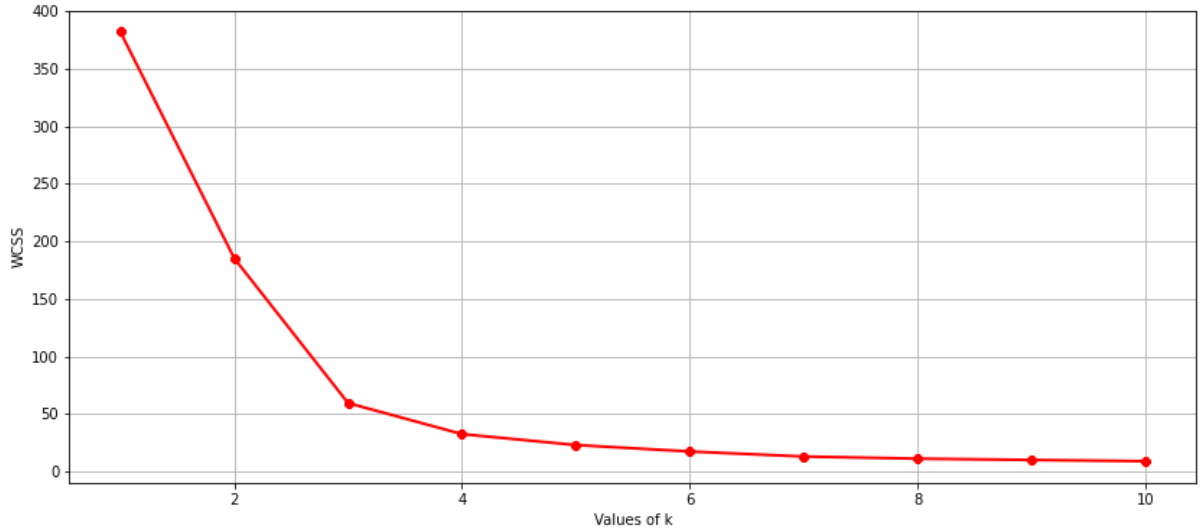
K-Means Clustering

k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

Using the Elbow Method for finding the appropriate value of K

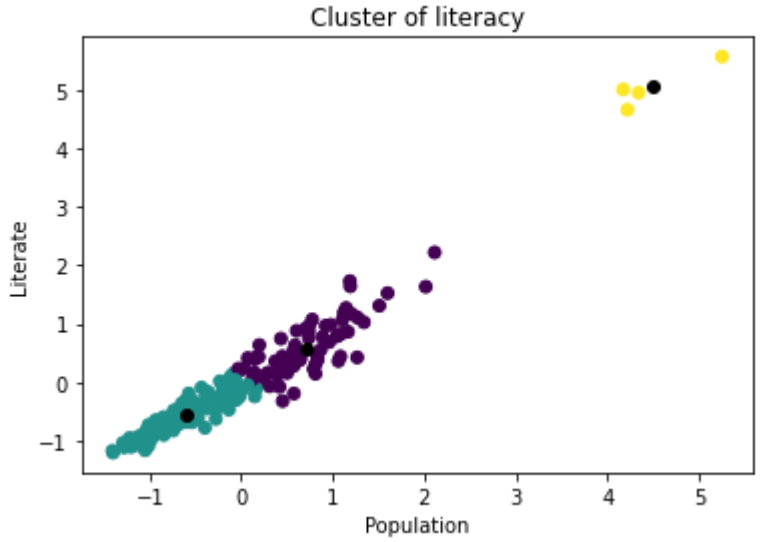
Total Population VS Literate Population

```
[ ]  X=segmentation.loc[:,["Population","Literate"]].values
     from sklearn.cluster import KMeans
     wcss = []
     for k in range(1, 11):
         kmeans=KMeans(n_clusters=k,init='k-means++')
         kmeans.fit(X)
         wcss.append(kmeans.inertia_)
     plt.figure(figsize=(14,6))
     plt.grid()
     plt.plot(range(1,11),wcss,linewidth=2,color='red',marker="8")
     plt.xlabel('Values of k')
     plt.ylabel('WCSS')
     plt.show()
```



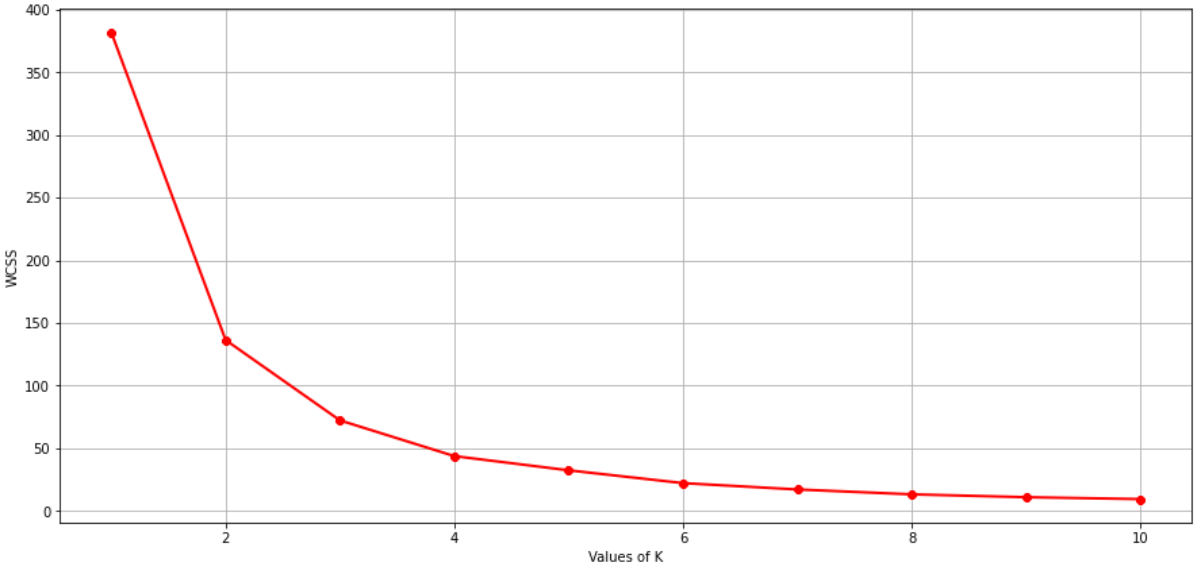

```
[ ]  kmeans=KMeans(n_clusters= 3)
     data['label1']=kmeans.fit_predict(X)
```

```
[ ]  plt.scatter(X[:,0],X[:,1],c=kmeans.labels_)
     plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black')
     plt.title('Cluster of literacy')
     plt.xlabel('Population')
     plt.ylabel('Literate')
     plt.show()
```



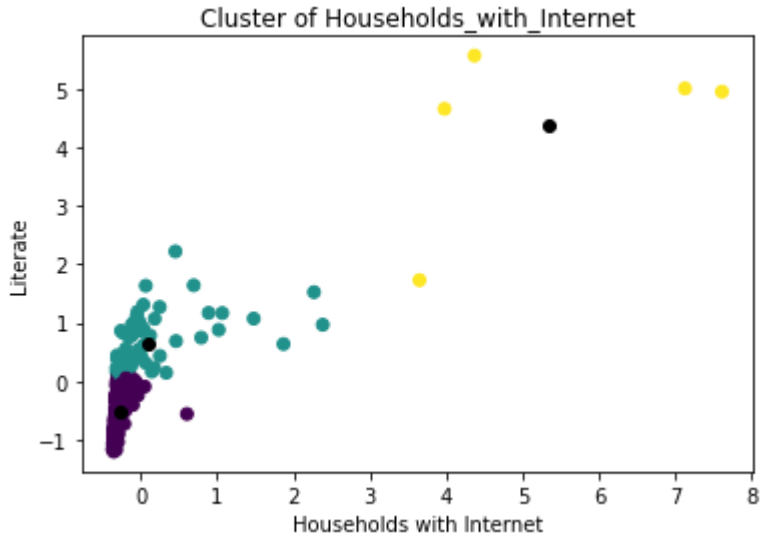
Household with Internet Population VS Literate Population

```
[ ]  X=segmentation.loc[:,["Households_with_Internet","Literate"]].values
     from sklearn.cluster import KMeans
     wcss=[]
     for k in range(1,11):
         kmeans=KMeans(n_clusters=k,init='k-means++')
         kmeans.fit(X)
         wcss.append(kmeans.inertia_)
     plt.figure(figsize=(15,7))
     plt.grid()
     plt.plot(range(1,11),wcss,linewidth=2,color='red',marker="8")
     plt.xlabel('Values of K')
     plt.ylabel('WCSS')
     plt.show()
```



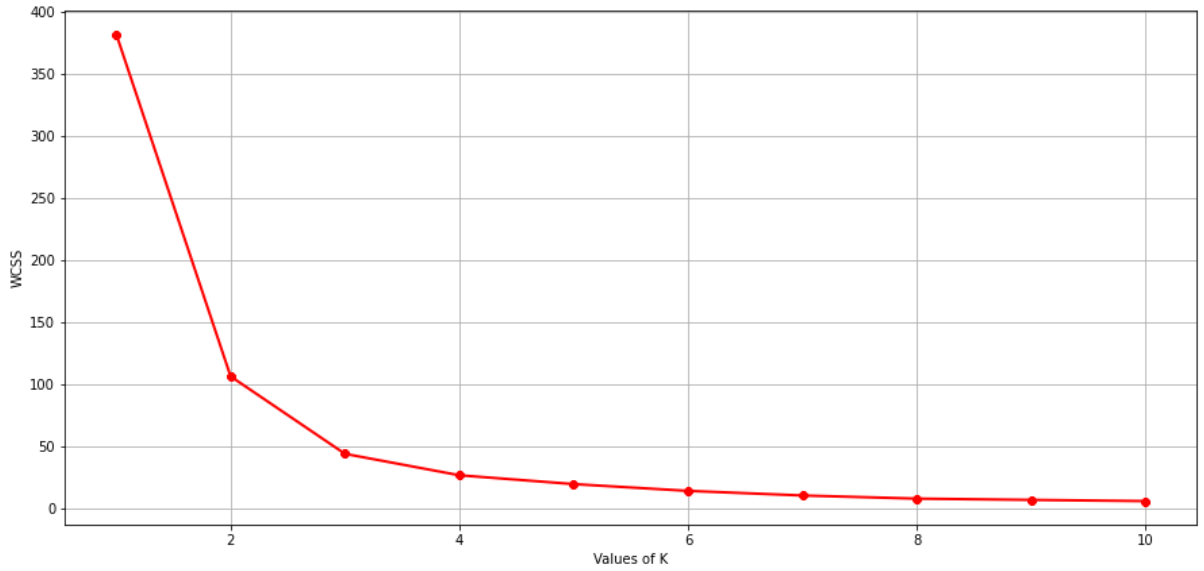
```
[ ]  kmeans=KMeans(n_clusters= 3)
     data['label2']=kmeans.fit_predict(X)
```

```
[ ]  plt.scatter(X[:,0],X[:,1],c=kmeans.labels_)
     plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black')
     plt.title('Cluster of Households_with_Internet')
     plt.xlabel('Households with Internet')
     plt.ylabel('Literate')
     plt.show()
```



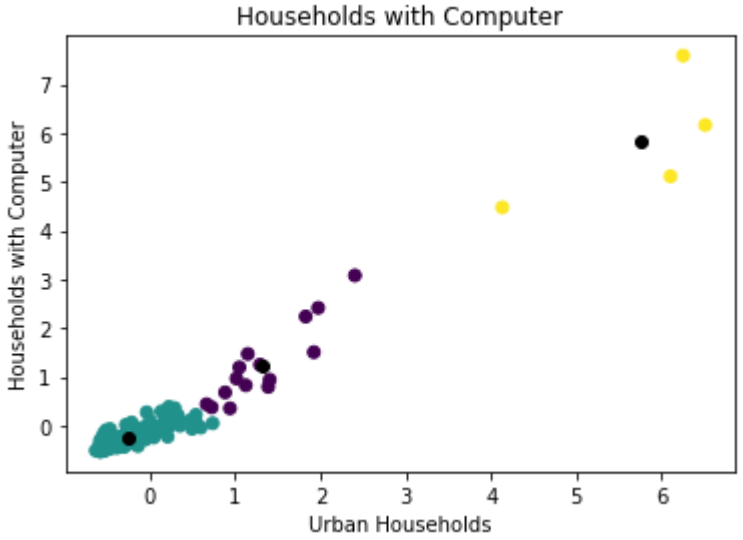
Urban Household Population VS Household with Computers

```
[ ]  X=segmentation.loc[:,["Urban_Households","Households_with_Computer"]].values
     from sklearn.cluster import KMeans
     wcss=[]
     for k in range(1,11):
         kmeans=KMeans(n_clusters=k,init='k-means++')
         kmeans.fit(X)
         wcss.append(kmeans.inertia_)
     plt.figure(figsize=(15,7))
     plt.grid()
     plt.plot(range(1,11),wcss,linewidth=2,color='red',marker="8")
     plt.xlabel('Values of K')
     plt.ylabel('WCSS')
     plt.show()
```




```
[ ]  kmeans=KMeans(n_clusters=3)
     data['label3']=kmeans.fit_predict(X)
```

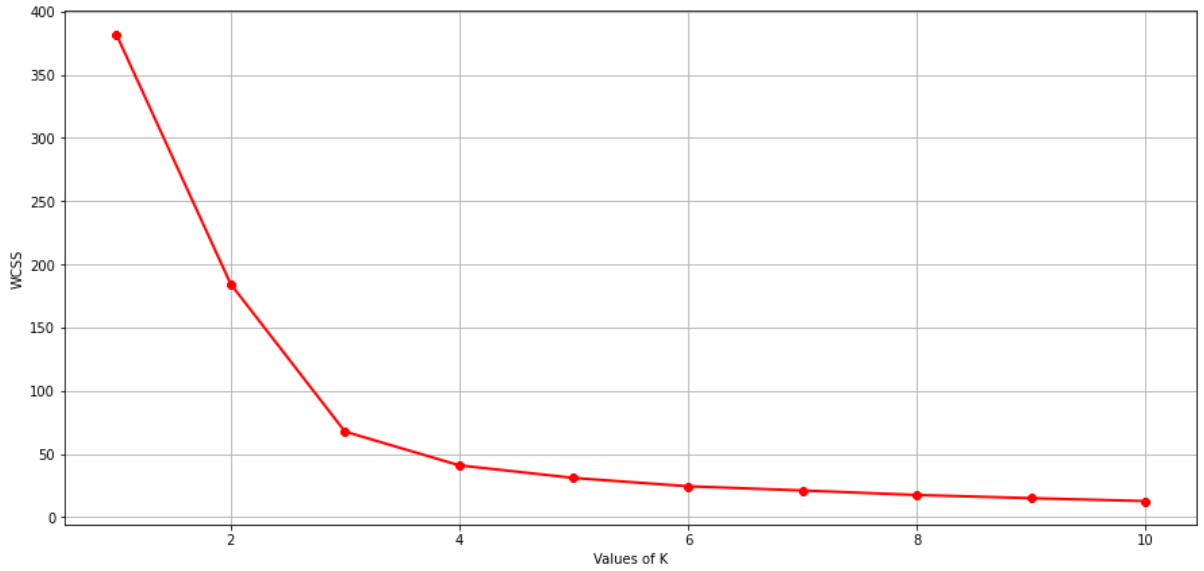
```
[ ]  plt.scatter(X[:,0],X[:,1],c=kmeans.labels_)
     plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black')
     plt.title('Households with Computer')
     plt.xlabel('Urban Households')
     plt.ylabel('Households with Computer')
     plt.show()
```



Age Group (0 to 29) VS Literate Population

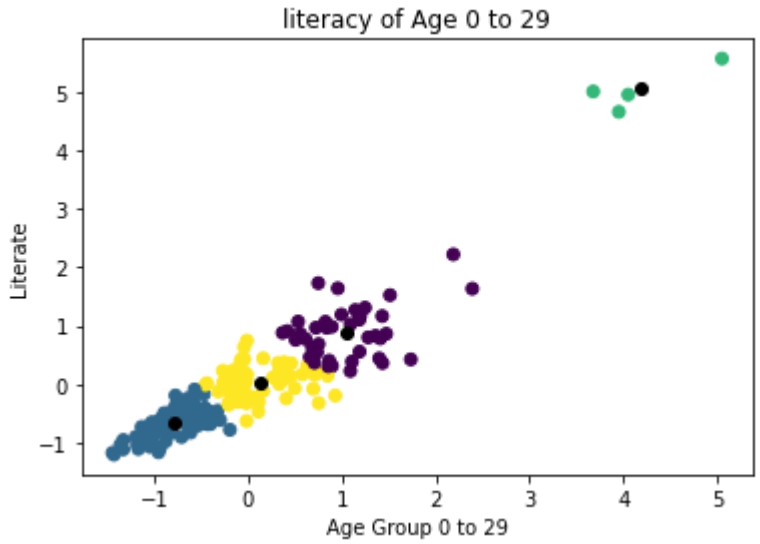
We would be targeting age with respect to the age group from 0 to 29 years of age as they would be our prime audience.

```
[ ]  X=segmentation.loc[:,["Age_Group_0_29","Literate"]].values
     from sklearn.cluster import KMeans
     wcss=[]
     for k in range(1,11):
         kmeans=KMeans(n_clusters=k,init='k-means++')
         kmeans.fit(X)
         wcss.append(kmeans.inertia_)
     plt.figure(figsize=(15,7))
     plt.grid()
     plt.plot(range(1,11),wcss,linewidth=2,color='red',marker="8")
     plt.xlabel('Values of K')
     plt.ylabel('WCSS')
     plt.show()
```




```
[ ]  kmeans=KMeans(n_clusters=4)
     data['label4']=kmeans.fit_predict(X)
```

```
[ ]  plt.scatter(X[:,0],X[:,1],c=kmeans.labels_)
     plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black')
     plt.title('literacy of Age 0 to 29')
     plt.xlabel('Age Group 0 to 29')
     plt.ylabel('Literate')
     plt.show()
```



3D Visualisation

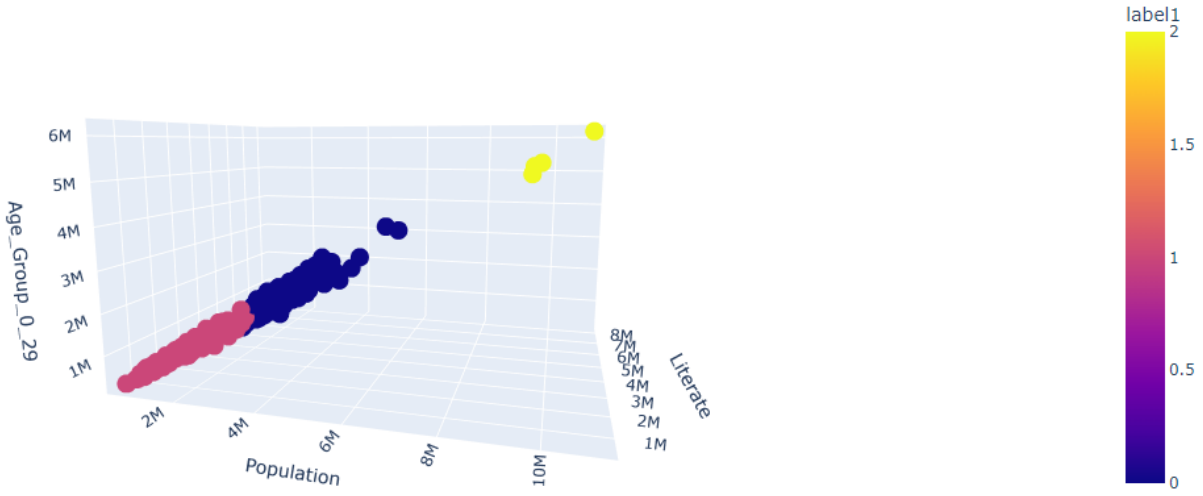
We would now be visualising our data in a three dimensional form such that we can see it via 3 scales or axes.
Visualisation of clusters of data points is very important. Various edges of the graph provide a quick view of the complex input data set.

It's not wise to serve all customers with the same product model, email, text message campaign, or ad. Customers have different needs. A one-size-for-all approach to business will generally result in less engagement, lower-click through rates, and ultimately fewer sales. Customer segmentation is the cure for this problem.

Finding an optimal number of unique customer groups will help us understand how our customers differ, and help you give them exactly what they want. Customer segmentation improves customer experience and boosts company revenue. That's why segmentation is a must if you want to surpass your competitors and get more customers.

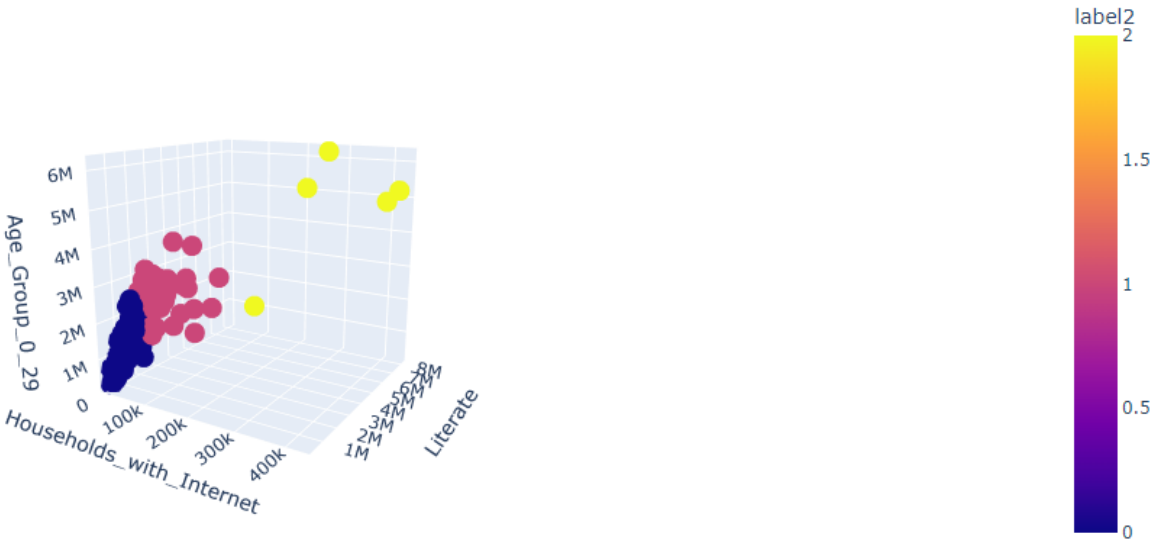
Population VS Literate Population VS Age (0 to 29)

```
[ ]  figure = px.scatter_3d(data,
                            color='label1',
                            x="Population",
                            y="Literate",
                            z="Age_Group_0_29",
                            category_orders = {"clusters": ["0","1","2","3","4"]}
                            )
     figure.update_layout()
     figure.show()
```



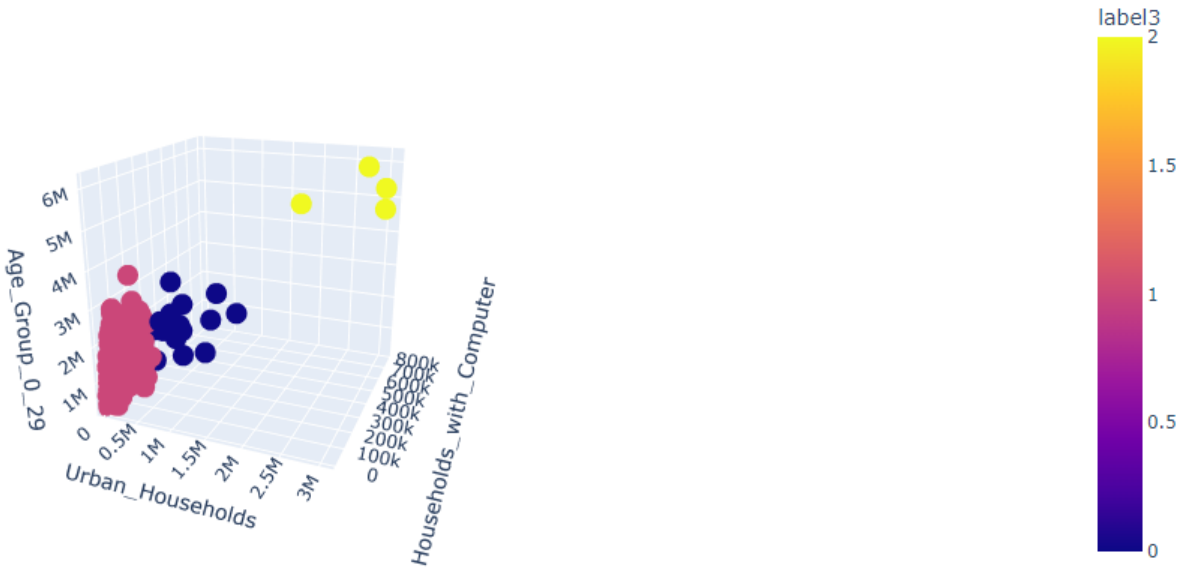
Household with Internet VS Literate Population VS Age (0 to 29)

```
[ ]  figure = px.scatter_3d(data,
                            color='label2',
                            x="Households_with_Internet",
                            y="Literate",
                            z="Age_Group_0_29",
                            category_orders = {"clusters": ["0","1","2","3","4"]}
                            )
     figure.update_layout()
     figure.show()
```



Urban Household Population VS Household with Computer VS Age (0 to 29)

```
[ ]  figure=px.scatter_3d(data,
                          color='label3',
                          x="Urban_Households",
                          y="Households_with_Computer",
                          z="Age_Group_0_29",
                          category_orders = {"clusters": ["0","1","2","3","4"]}
                          )
     figure.update_layout()
     figure.show()
```



Conclusion

- **Geographic Segment**

  Based on the data collected and analyzed we have established that Educational Technology will be applicable to students residing in these places:-

  1. Thane
  2. Pune
  3. Mumbai Suburban
  4. Bangalore
  5. Chennai

- **Demographic Segment**

  The Age group of 0-29 years of Age show to have the most demand for Educational Technologies in India and these Age Groups must be targeted.

- **Behavioral and Psychographic Segments**

  Extensive surveys must be conducted within the above mentioned 5 districts in order to understand the behavioral and psychographic nature of customers.

GitHub Link

https://github.com/ANHYDROUS-H2O/Feynn-Labs/tree/main/Team%20Andre