Restaurant Venues Clustering in Chennai City using Foursquare & Population data

IBM Data Science Professional Certificate – Final Capstone project (The Battle of Neighborhoods)

By

Prabhakaran Elangovan

Contents

1. Introduction ... 3
1.1 Business Case ... 3
2. Data Requirements ... 3
2.1 Chennai city Population data ... 3
2.2 Venue Master Data ... 4
2.3. Additional Data ... 4
3. Methodology ... 5
Step 1: Data preparation ... 5
Step 2: Exploratory Data Analysis ... 7
Step 3: Feature Selection ... 9
Step 4: Modeling ... 9
4. Results ... 10
5. Discussion ... 11
5.1 Cluster 1 Analysis ... 11
5.2 Cluster 2 analysis ... 12
5.3 Cluster 3 Analysis ... 13
6. Conclusion ... 13

1. Introduction

1.1 Business Case

Chennai is one of the emerging market places in India next to Delhi and Mumbai. It is also a hub for industries, due to the availability of port for easy exports and import. The city's economy is supported majorly by urban population with high per capita income working in various sectors. It ranks 2nd in GDP contribution among cities and 10th in GDP per capita. This provides ample opportunities for budding entrepreneurs, especially in organized food retail sector with the recent gigantic growth of food delivery partners like Swiggy, Zomato, etc. As the city is developing for a really long period, there comes a concern for entrepreneurs in finding out the right place for their restaurant business. The objective of the project is to harness the power of data & Data Science and to derive insights about various locations. The project is aimed at reducing the burden of entrepreneurs to make better and profitable decisions while finalizing the venue to invest.

2. Data Requirements

This section will discuss about the publicly available data that are required in building a solution to our business case. The business case requires the public venues listed in each area along with the population data for each area. The data needs are met by accessing data from two popular public data sources.

- Population data – Primary Census Abstract (PCA) data released by Government of India partnered with the Chennai Metro Corporation department.
- Venue master list – Venue list for each area in Chennai as available in Foursquare application.

2.1 Chennai city Population data

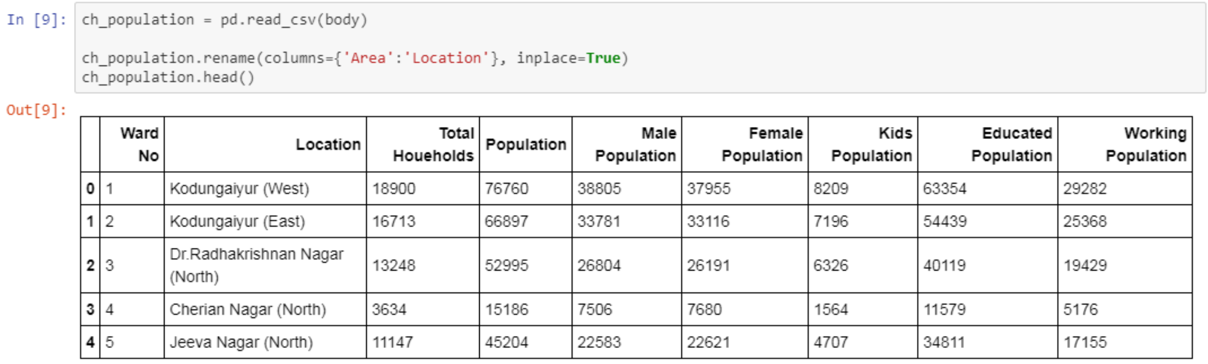
The population data provided by Government of India includes multiple details that are irrelevant to the task at hand. Hence, the following details were utilized for the project:

- Total Households – Total number of households/houses.
- Total Population – Total population
- Male Population – Total male population
- Female Population – Total female population
- Kids Population – Total population of kids aged between 0-6
- Educated Population – Total number of educated persons
- Working Population – Total number of working persons

Data Source: http://censusindia.gov.in/pca/pcadata/Houselisting-housing-TM.html

The data is available for download in the above given link. As stated earlier, the population data contained multiple other details which are either irrelevant or not of much help in this context.

Also, the data does not cover complete demography of the population; few important details like age-wise classification, sector-wise workers cover only Agricultural, Household industries and does not cover other sector workers. Though, if available these details can help in improving our end solution it is not mandatory in this context. Snapshot of the excel file read into a pandas DataFrame is provided below for reference.



```
In [9]: ch_population = pd.read_csv(body)

        ch_population.rename(columns={'Area':'Location'}, inplace=True)
        ch_population.head()
```

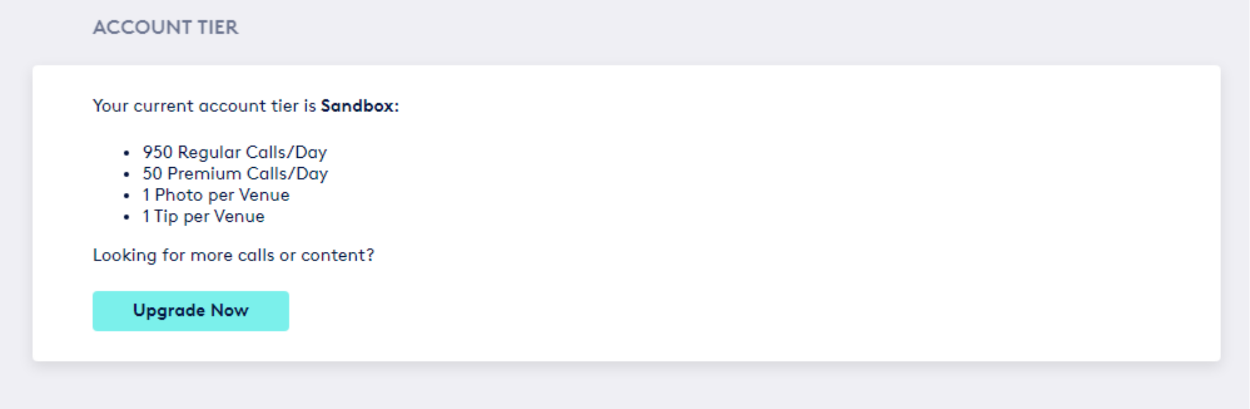
Out[9]:

	Ward No	Location	Total Houeholds	Population	Male Population	Female Population	Kids Population	Educated Population	Working Population
0	1	Kodungaiyur (West)	18900	76760	38805	37955	8209	63354	29282
1	2	Kodungaiyur (East)	16713	66897	33781	33116	7196	54439	25368
2	3	Dr.Radhakrishnan Nagar (North)	13248	52995	26804	26191	6326	40119	19429
3	4	Cherian Nagar (North)	3634	15186	7506	7680	1564	11579	5176
4	5	Jeeva Nagar (North)	11147	45204	22583	22621	4707	34811	17155

2.2 Venue Master Data

The list of venues available in each neighborhood is gathered from Foursquare via API available in their Developer Portal. Foursquare provides geographical data service to its users and the data creation and maintenance of valid information is crowd-sourced. Hence, their venues database available under a specific city is exhaustive and useful for business cases such as these.

Foursquare enables developers to access their data by touching different endpoints using API. This is free of cost; however there is a limit on the number of API calls that can be made in a single day for the free account (screenshot below).



ACCOUNT TIER

Your current account tier is **Sandbox:**

- 950 Regular Calls/Day
- 50 Premium Calls/Day
- 1 Photo per Venue
- 1 Tip per Venue

Looking for more calls or content?

2.3. Additional Data

The Foursquare API call requires the latitude and longitude of the area for which the venue data is required. It is to be noted that the population data does not contain the location co-ordinates; hence for this purpose a ‘geocoder’ python library is used. Though multiple other libraries were available for the same purpose, geocoder was more effective.

3. Methodology

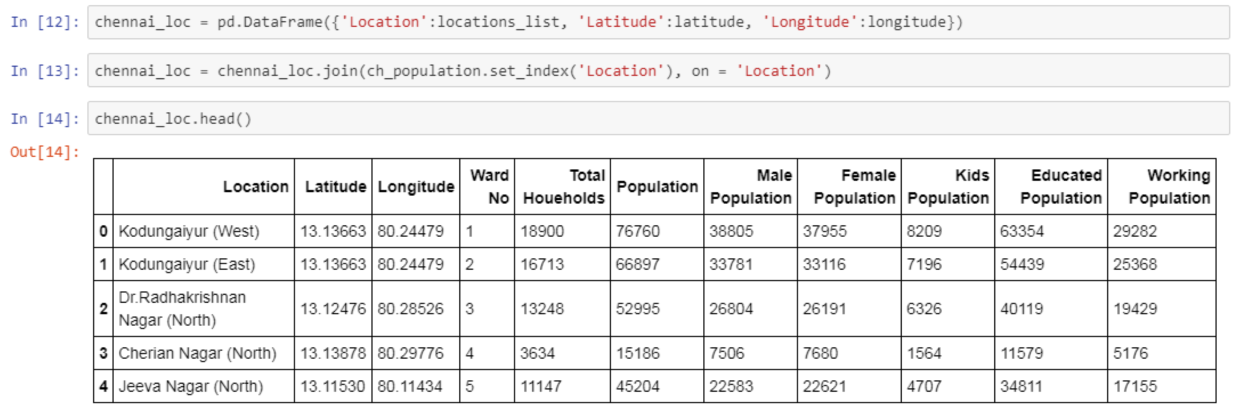
The problem statement is to segment locations in Chennai city to derive insights using the collected data. Of the various available Machine Learning algorithms, Clustering can be finalized as the Analytic approach to be followed for this project.

Class details are not available; hence this is an Unsupervised Machine Learning scenario. K-Means clustering is the widely used clustering algorithm for segmentation problems, the project also utilizes the same.

This section illustrates the steps involved in solving the problem:

Step 1: Data preparation

The requisite data is collected from various sources; the next step is to prepare the data in a usable way. The population data (collected from Govt portal) and the location coordinate data (collected using geocoder) are to merged/joined together into a single DataFrame as shown below:



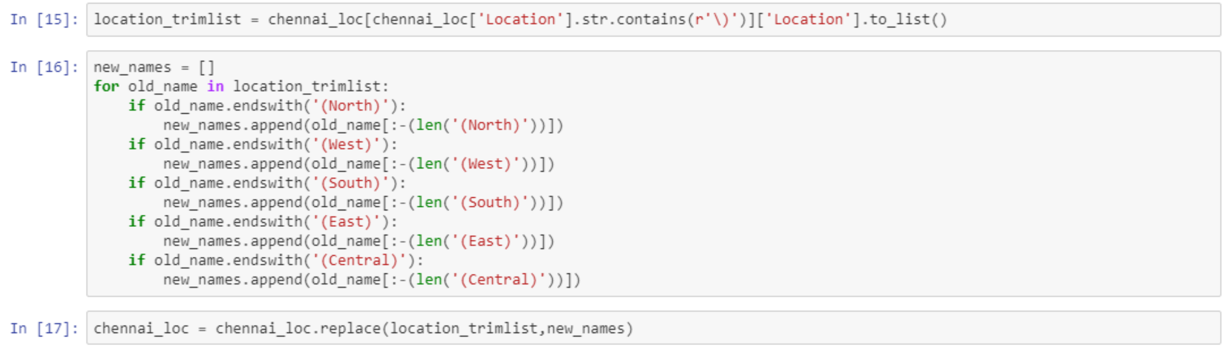
```
In [12]: chennai_loc = pd.DataFrame({'Location':locations_list, 'Latitude':latitude, 'Longitude':longitude})

In [13]: chennai_loc = chennai_loc.join(ch_population.set_index('Location'), on = 'Location')

In [14]: chennai_loc.head()
Out[14]:
```

	Location	Latitude	Longitude	Ward No	Total Houeholds	Population	Male Population	Female Population	Kids Population	Educated Population	Working Population
0	Kodungaiyur (West)	13.13663	80.24479	1	18900	76760	38805	37955	8209	63354	29282
1	Kodungaiyur (East)	13.13663	80.24479	2	16713	66897	33781	33116	7196	54439	25368
2	Dr.Radhakrishnan Nagar (North)	13.12476	80.28526	3	13248	52995	26804	26191	6326	40119	19429
3	Cherian Nagar (North)	13.13878	80.29776	4	3634	15186	7506	7680	1564	11579	5176
4	Jeeva Nagar (North)	13.11530	80.11434	5	11147	45204	22583	22621	4707	34811	17155

It is evident on observing the first two rows that there are few locations have the same location coordinates, thought the Location names are different. On analyzing further it is found that census data is split for few large locations for administrative purposes based on directions (North, South, etc.). In this context, if the data is used as-is the venues data derived from the Foursquare will be completely duplicates. Hence, it is decided that similar cases are to be found out and the duplicate rows are to be merged. Prior to merging of rows, the Location values containing the direction values are to be truncated and the direction details are to be removed.

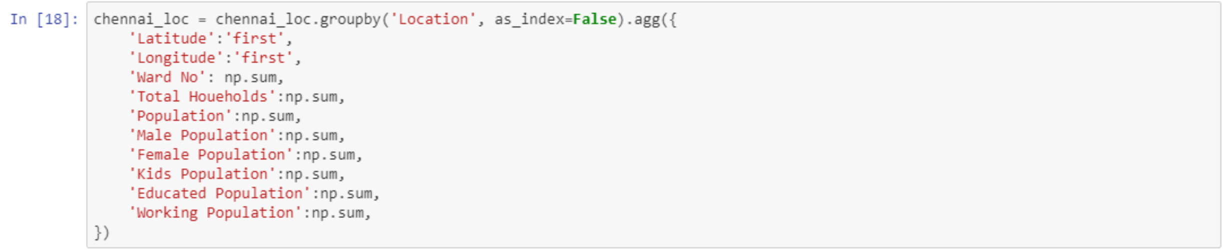
This step is achieved by checking for Location values that ends with a specific substring. The replace method is used to find the value to removed and replace with the new values. This ensured that the

Location data now contains some values that are repeated. The same is explained in the code that is shown below:



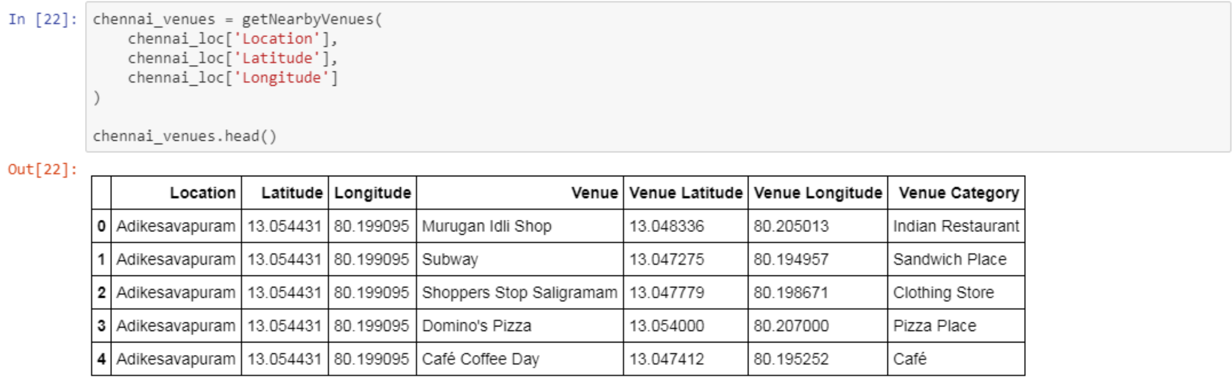
```
In [15]: location_trimlist = chennai_loc[chennai_loc['Location'].str.contains(r'\)')]['Location'].to_list()

In [16]: new_names = []
         for old_name in location_trimlist:
             if old_name.endswith('(North)'):
                 new_names.append(old_name[:-(len('(North)'))])
             if old_name.endswith('(West)'):
                 new_names.append(old_name[:-(len('(West)'))])
             if old_name.endswith('(South)'):
                 new_names.append(old_name[:-(len('(South)'))])
             if old_name.endswith('(East)'):
                 new_names.append(old_name[:-(len('(East)'))])
             if old_name.endswith('(Central)'):
                 new_names.append(old_name[:-(len('(Central)'))])

In [17]: chennai_loc = chennai_loc.replace(location_trimlist,new_names)
```

Once this is done, the groupby method is then used to group the table at Location level, care was taken to ensure that the latitude and longitude values are not merged/summed.



```
In [18]: chennai_loc = chennai_loc.groupby('Location', as_index=False).agg({
             'Latitude':'first',
             'Longitude':'first',
             'Ward No': np.sum,
             'Total Houeholds':np.sum,
             'Population':np.sum,
             'Male Population':np.sum,
             'Female Population':np.sum,
             'Kids Population':np.sum,
             'Educated Population':np.sum,
             'Working Population':np.sum,
         })
```

Once the demographics data is ready, the latitude and longitude values from this dataframe are passed to Foursquare API along with the credentials and other details in the required URI format. The data returned from Foursquare is stored in a different dataframe. The screen below explains the same.



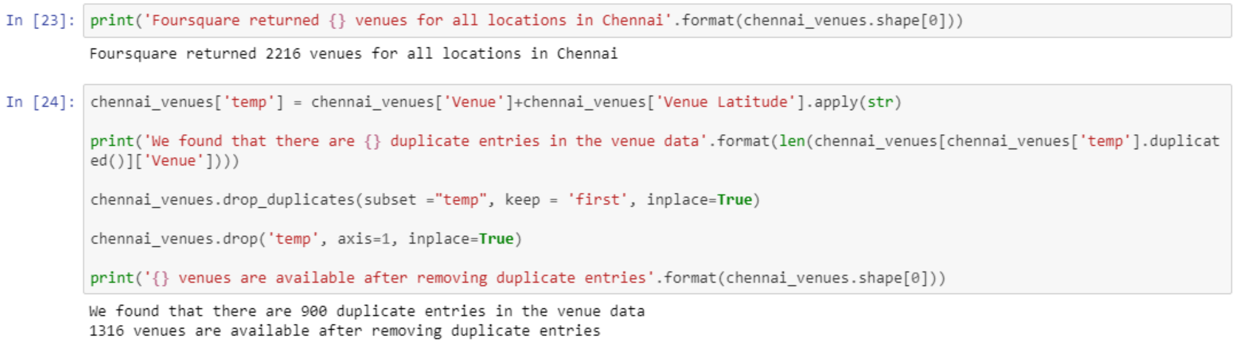
```
In [22]: chennai_venues = getNearbyVenues(
             chennai_loc['Location'],
             chennai_loc['Latitude'],
             chennai_loc['Longitude']
         )

         chennai_venues.head()
```

Out[22]:

	Location	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Adikesavapuram	13.054431	80.199095	Murugan Idli Shop	13.048336	80.205013	Indian Restaurant
1	Adikesavapuram	13.054431	80.199095	Subway	13.047275	80.194957	Sandwich Place
2	Adikesavapuram	13.054431	80.199095	Shoppers Stop Saligramam	13.047779	80.198671	Clothing Store
3	Adikesavapuram	13.054431	80.199095	Domino's Pizza	13.054000	80.207000	Pizza Place
4	Adikesavapuram	13.054431	80.199095	Café Coffee Day	13.047412	80.195252	Café

The venues within 1000 meters radius is pulled from Foursquare, subject to a maximum of 100 items return limit in a single request. The application returned 2216 venues for the list of locations passed. The results set contains many duplicate values at there were few venues that fall under 1000 meters radius of multiple location, as a result become duplicate entries.

To find the duplicate rows, a temp column called was added to the existing dataframe, values are the concatenation of Venues and Latitude. This implies that the rows are similar however present under different location. It is found that there are 900 duplicate rows available from the data returned by Foursquare. The duplicate values are removed using drop_duplicates method. Finally, a total of 1316 unique venues provided by Foursquare is selected.

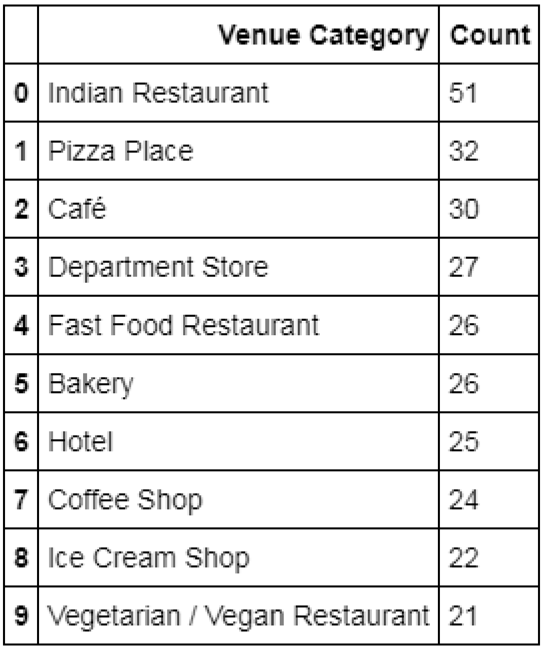


```
In [23]: print('Foursquare returned {} venues for all locations in Chennai'.format(chennai_venues.shape[0]))

         Foursquare returned 2216 venues for all locations in Chennai

In [24]: chennai_venues['temp'] = chennai_venues['Venue']+chennai_venues['Venue Latitude'].apply(str)

         print('We found that there are {} duplicate entries in the venue data'.format(len(chennai_venues[chennai_venues['temp'].duplicated()]['Venue'])))

         chennai_venues.drop_duplicates(subset ="temp", keep = 'first', inplace=True)

         chennai_venues.drop('temp', axis=1, inplace=True)

         print('{} venues are available after removing duplicate entries'.format(chennai_venues.shape[0]))

         We found that there are 900 duplicate entries in the venue data
         1316 venues are available after removing duplicate entries
```

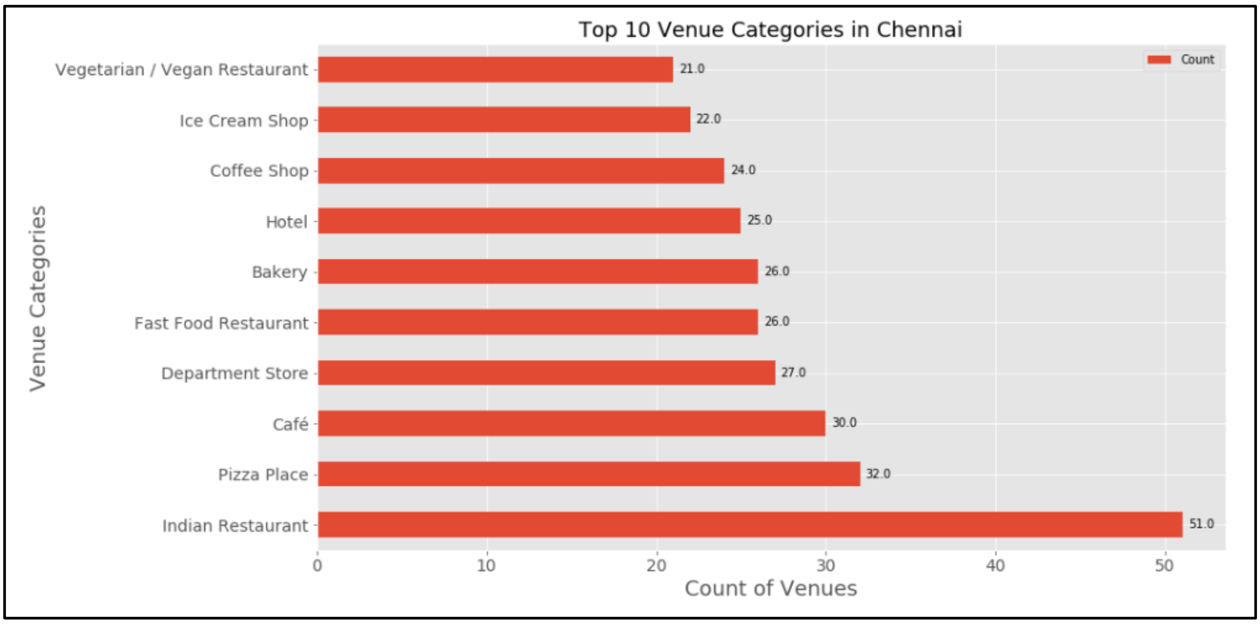
Step 2: Exploratory Data Analysis

Once, the data is cleaned and ready, the next step is to explore and find any trends that the data displays.

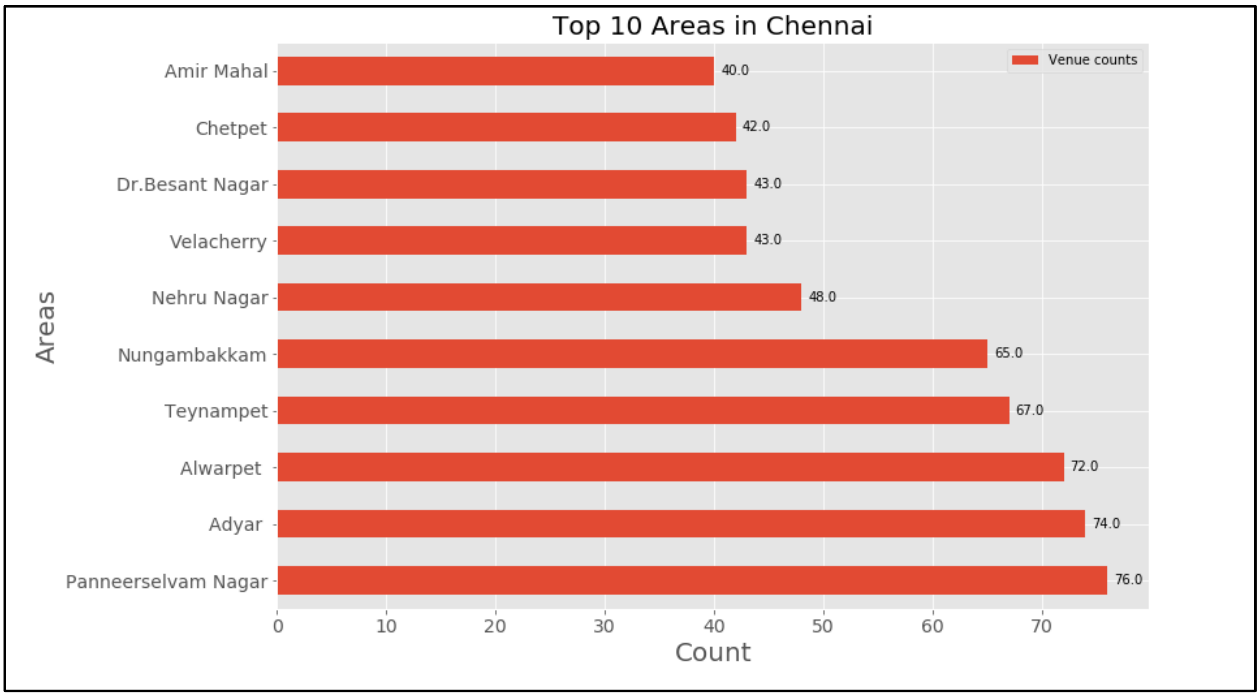
- Total number of Venue categories available in Chennai as per Foursquare is 171.
- Top 10 Venue Categories in the city are as follows:



	Venue Category	Count
0	Indian Restaurant	51
1	Pizza Place	32
2	Café	30
3	Department Store	27
4	Fast Food Restaurant	26
5	Bakery	26
6	Hotel	25
7	Coffee Shop	24
8	Ice Cream Shop	22
9	Vegetarian / Vegan Restaurant	21



- Top 10 locations that have most number of venues in the city.



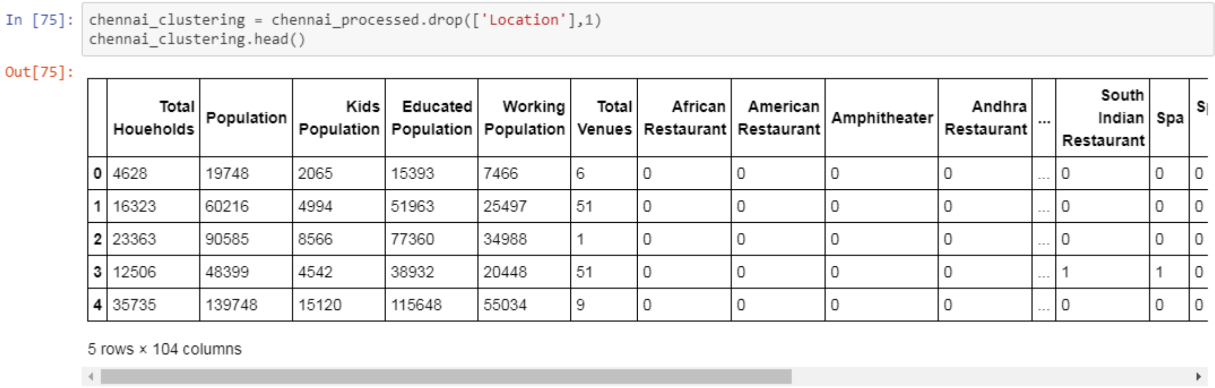
- The data is filtered such that it contains only ‘Restaurant’ related venue categories like any value containing restaurant in it or BBQ Joint or other similar categories. It is found that the city contains 418 such joints belonging to 71 different types of restaurants.

Step 3: Feature Selection

Though, Foursquare returned 1316 unique venues for exploration, not all venues or venue categories can contribute in solving the problem statement. As a part of Feature selection, only those venues which are either a restaurant or linked to entertainment spends like Multiplex, Shopping Malls, Beach, Sports venues, etc. are selected as the features that are to be fed into our clustering model.

Under population data, Male/Female population may not contribute in the expected manner, hence those values are excluded.

Apart from these, features such as Location coordinates of area and venue, ward no, venue names which does make sense to include as selected features are also removed.



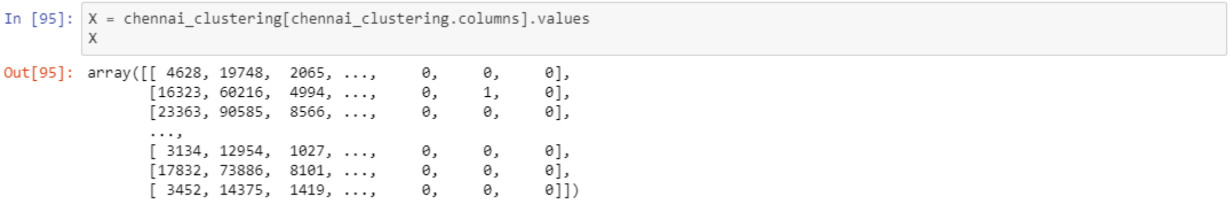
```
In [75]: chennai_clustering = chennai_processed.drop(['Location'],1)
         chennai_clustering.head()
Out[75]:
```

	Total Houeholds	Population	Kids Population	Educated Population	Working Population	Total Venues	African Restaurant	American Restaurant	Amphitheater	Andhra Restaurant	...	South Indian Restaurant	Spa	S
0	4628	19748	2065	15393	7466	6	0	0	0	0	...	0	0	0
1	16323	60216	4994	51963	25497	51	0	0	0	0	...	0	0	0
2	23363	90585	8566	77360	34988	1	0	0	0	0	...	0	0	0
3	12506	48399	4542	38932	20448	51	0	0	0	0	...	1	1	0
4	35735	139748	15120	115648	55034	9	0	0	0	0	...	0	0	0

5 rows × 104 columns

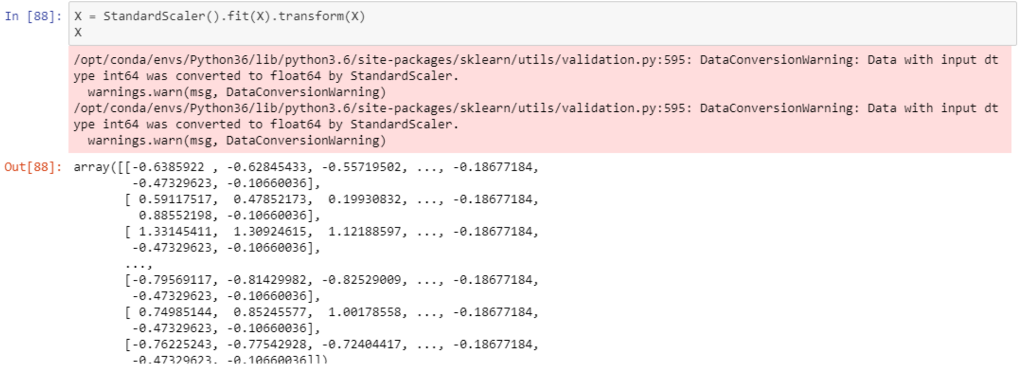
Step 4: Modeling

- The selected features are converted to numpy arrays and stored as X.



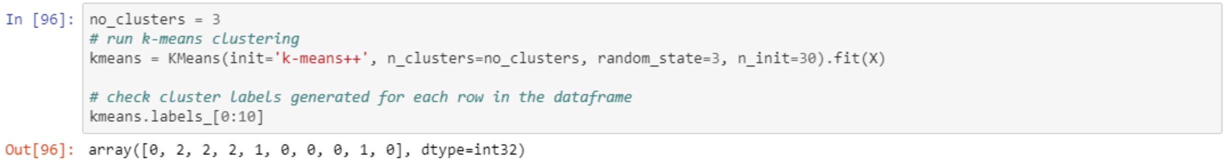
```
In [95]: X = chennai_clustering[chennai_clustering.columns].values
         X
Out[95]: array([[ 4628, 19748,  2065, ...,     0,     0,     0],
                [16323, 60216,  4994, ...,     0,     1,     0],
                [23363, 90585,  8566, ...,     0,     0,     0],
                ...,
                [ 3134, 12954,  1027, ...,     0,     0,     0],
                [17832, 73886,  8101, ...,     0,     0,     0],
                [ 3452, 14375,  1419, ...,     0,     0,     0]])
```

- The features are then standardized using StandardScaler from scikit learn preprocessing package



```
In [88]: X = StandardScaler().fit(X).transform(X)
         X
/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/utils/validation.py:595: DataConversionWarning: Data with input dt
ype int64 was converted to float64 by StandardScaler.
  warnings.warn(msg, DataConversionWarning)
/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/utils/validation.py:595: DataConversionWarning: Data with input dt
ype int64 was converted to float64 by StandardScaler.
  warnings.warn(msg, DataConversionWarning)
Out[88]: array([[-0.6385922 , -0.62845433, -0.55719502, ..., -0.18677184,
                 -0.47329623, -0.10660036],
                [ 0.59117517,  0.47852173,  0.19930832, ..., -0.18677184,
                  0.88552198, -0.10660036],
                [ 1.33145411,  1.30924615,  1.12188597, ..., -0.18677184,
                 -0.47329623, -0.10660036],
                ...,
                [-0.79569117, -0.81429982, -0.82529009, ..., -0.18677184,
                 -0.47329623, -0.10660036],
                [ 0.74985144,  0.85245577,  1.00178558, ..., -0.18677184,
                 -0.47329623, -0.10660036],
                [-0.76225243, -0.77542928, -0.72404417, ..., -0.18677184,
                 -0.47329623, -0.10660036]])
```

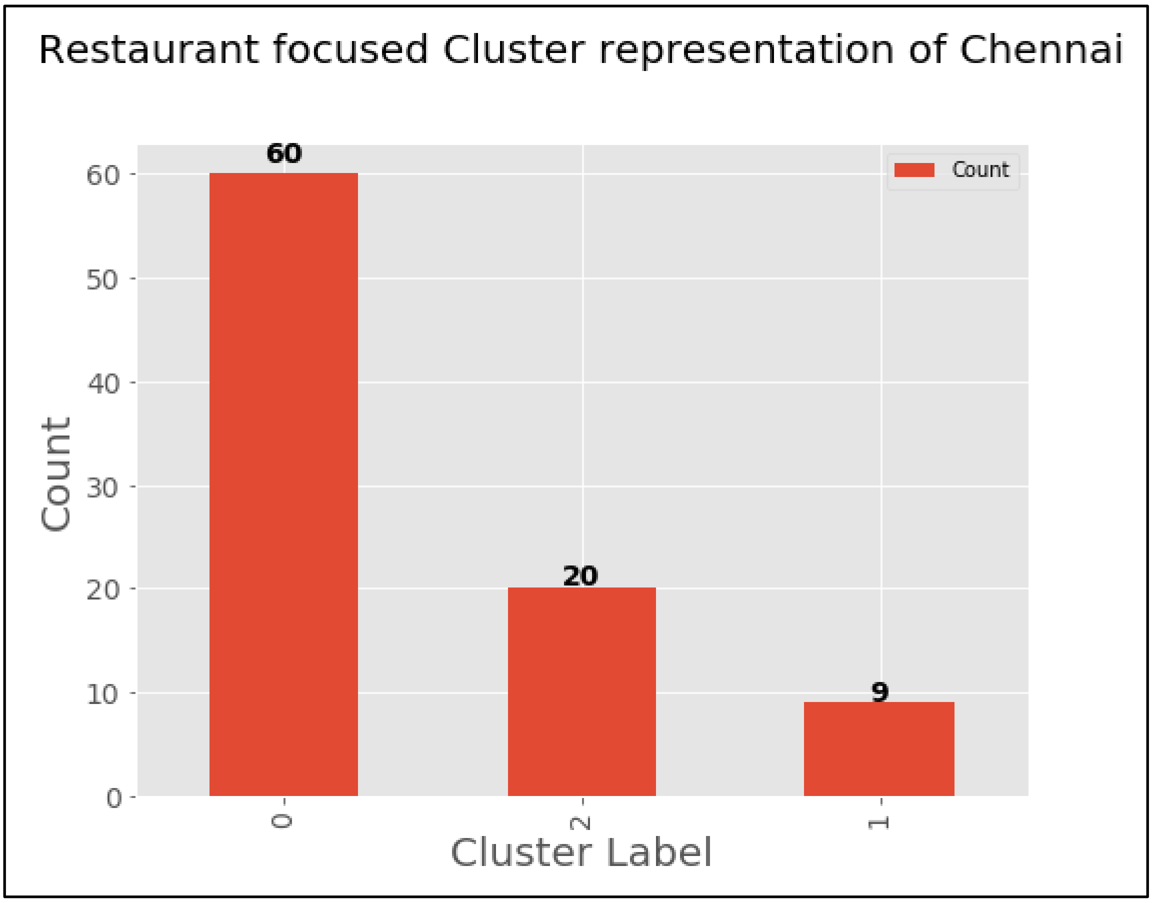
- Initialized the clusters as three, and k-Means clustering is done. n_init is set to 30 to run the algorithm that many times with different random centroids.



```
In [96]: no_clusters = 3
         # run k-means clustering
         kmeans = KMeans(init='k-means++', n_clusters=no_clusters, random_state=3, n_init=30).fit(X)

         # check cluster labels generated for each row in the dataframe
         kmeans.labels_[0:10]
Out[96]: array([0, 2, 2, 2, 1, 0, 0, 0, 1, 0], dtype=int32)
```

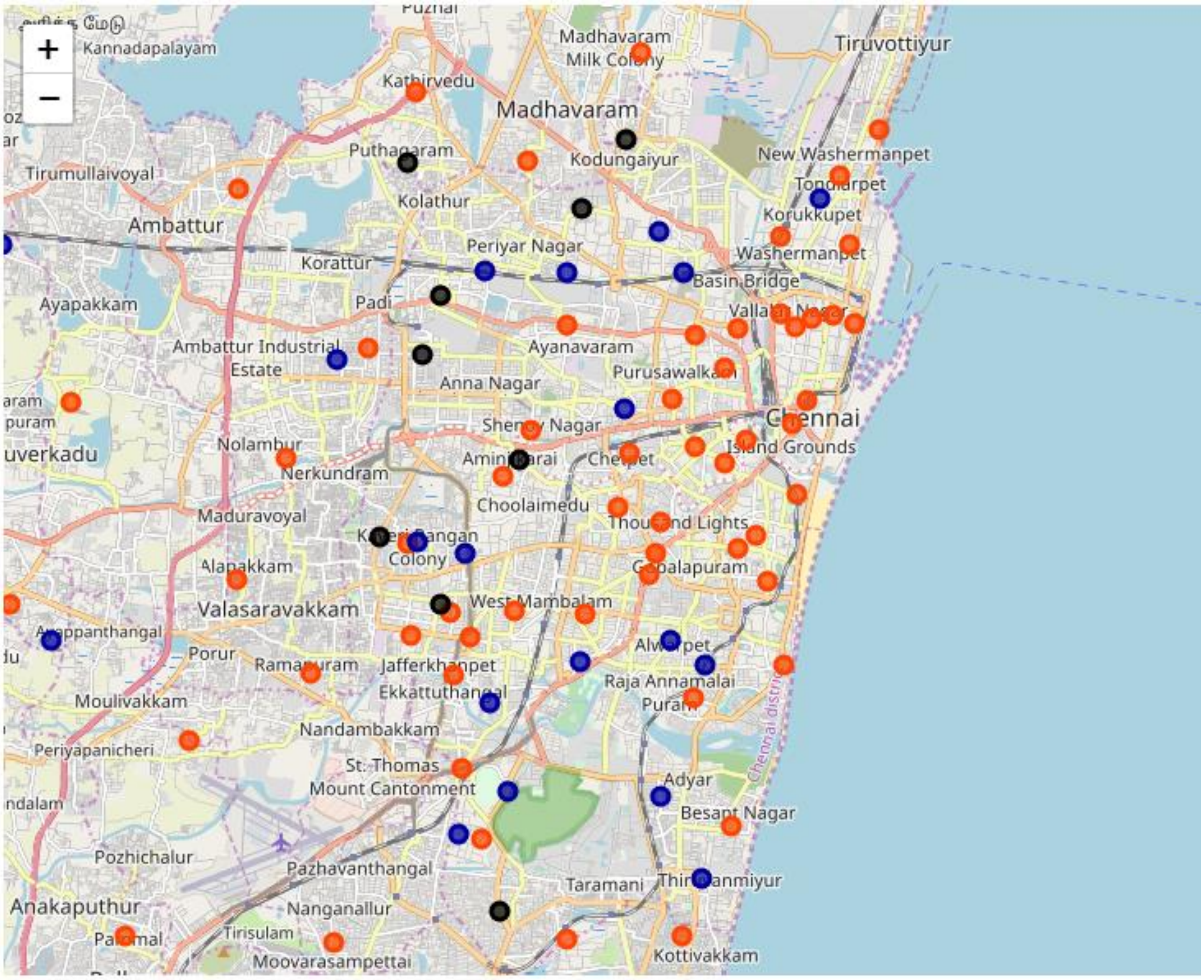
4. Results

The model classified 89 Chennai city locations into 3 different clusters as shown below:



- Cluster 1 – Labelled as 0 contains 60 localities
- Cluster 2 – Labelled as 1 contains 9 localities
- Cluster 3 – Labelled as 2 contains 20 localities

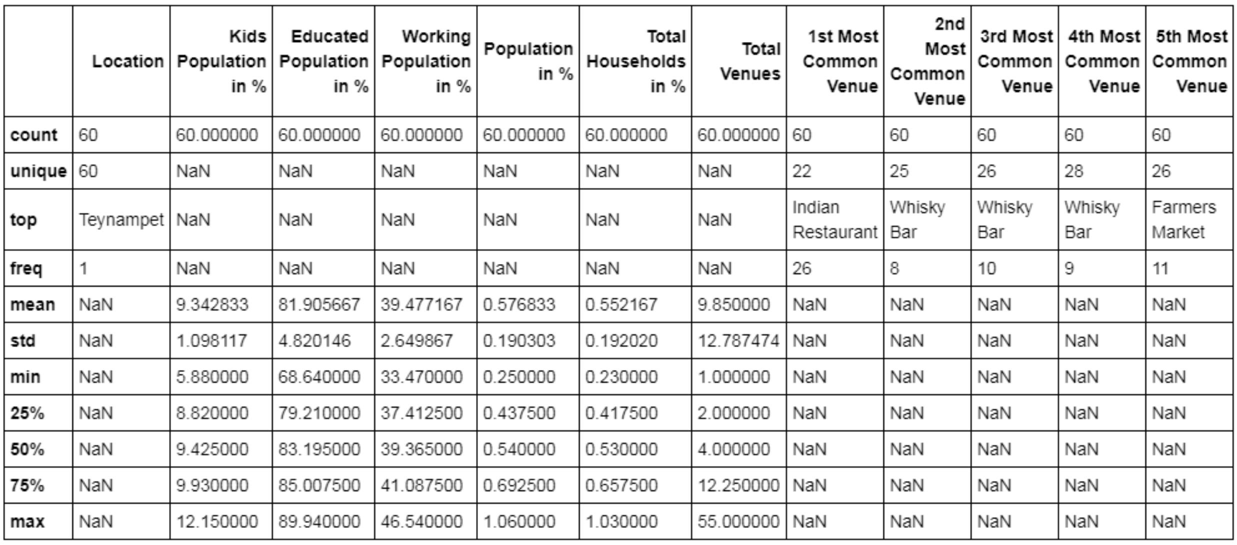
Below is the Folium map representation of the clusters spread across the city:



5. Discussion

5.1 Cluster 1 Analysis

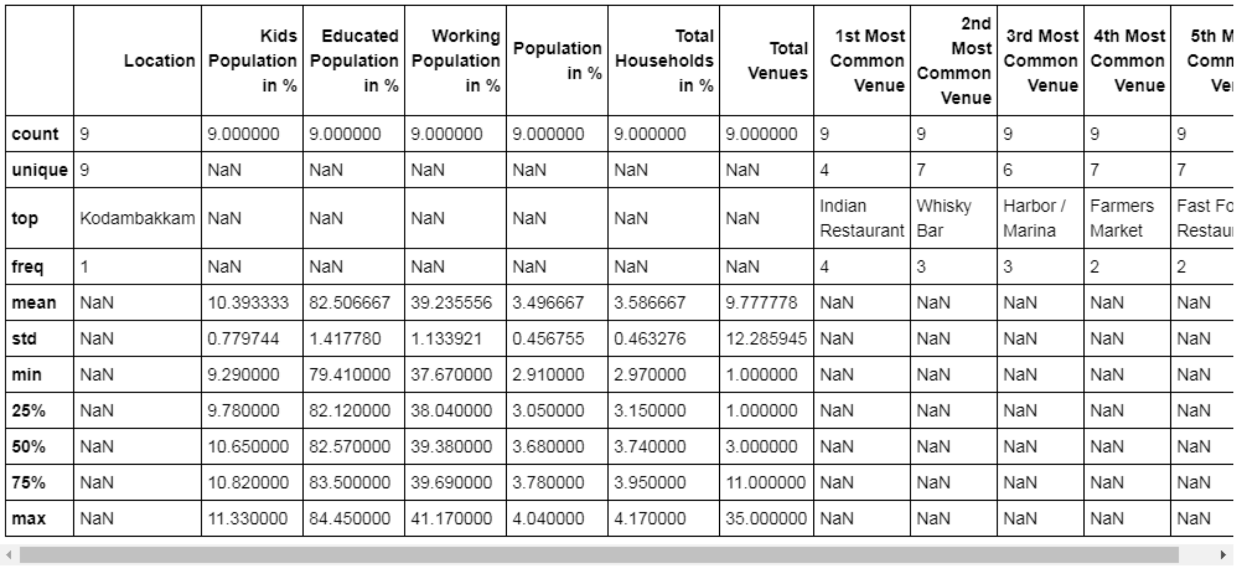
- This is the densest cluster of all three and contains 60 localities.
- The Cluster is sparsely populated with mean population of only 0.55%.
- The Total number of venues has a very high variance and ranges between 1 and 55.
- The cluster members have 'Whisky Bar' as either 2nd, 3rd, 4th common venue repeating to a total of 27 in these positions.
- Highest 1st common venue is 'Indian Restaurant', repeating 26 times in the spot.
- Has mean education population of 81%, indicating that the members contain educated groups in their locality slightly higher than other clusters.
- Below is the descriptive statistics data of the cluster.



	Location	Kids Population in %	Educated Population in %	Working Population in %	Population in %	Total Households in %	Total Venues	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
count	60	60.000000	60.000000	60.000000	60.000000	60.000000	60.000000	60	60	60	60	60
unique	60	NaN	NaN	NaN	NaN	NaN	NaN	22	25	26	28	26
top	Teynampet	NaN	NaN	NaN	NaN	NaN	NaN	Indian Restaurant	Whisky Bar	Whisky Bar	Whisky Bar	Farmers Market
freq	1	NaN	NaN	NaN	NaN	NaN	NaN	26	8	10	9	11
mean	NaN	9.342833	81.905667	39.477167	0.576833	0.552167	9.850000	NaN	NaN	NaN	NaN	NaN
std	NaN	1.098117	4.820146	2.649867	0.190303	0.192020	12.787474	NaN	NaN	NaN	NaN	NaN
min	NaN	5.880000	68.640000	33.470000	0.250000	0.230000	1.000000	NaN	NaN	NaN	NaN	NaN
25%	NaN	8.820000	79.210000	37.412500	0.437500	0.417500	2.000000	NaN	NaN	NaN	NaN	NaN
50%	NaN	9.425000	83.195000	39.365000	0.540000	0.530000	4.000000	NaN	NaN	NaN	NaN	NaN
75%	NaN	9.930000	85.007500	41.087500	0.692500	0.657500	12.250000	NaN	NaN	NaN	NaN	NaN
max	NaN	12.150000	89.940000	46.540000	1.060000	1.030000	55.000000	NaN	NaN	NaN	NaN	NaN

5.2 Cluster 2 analysis

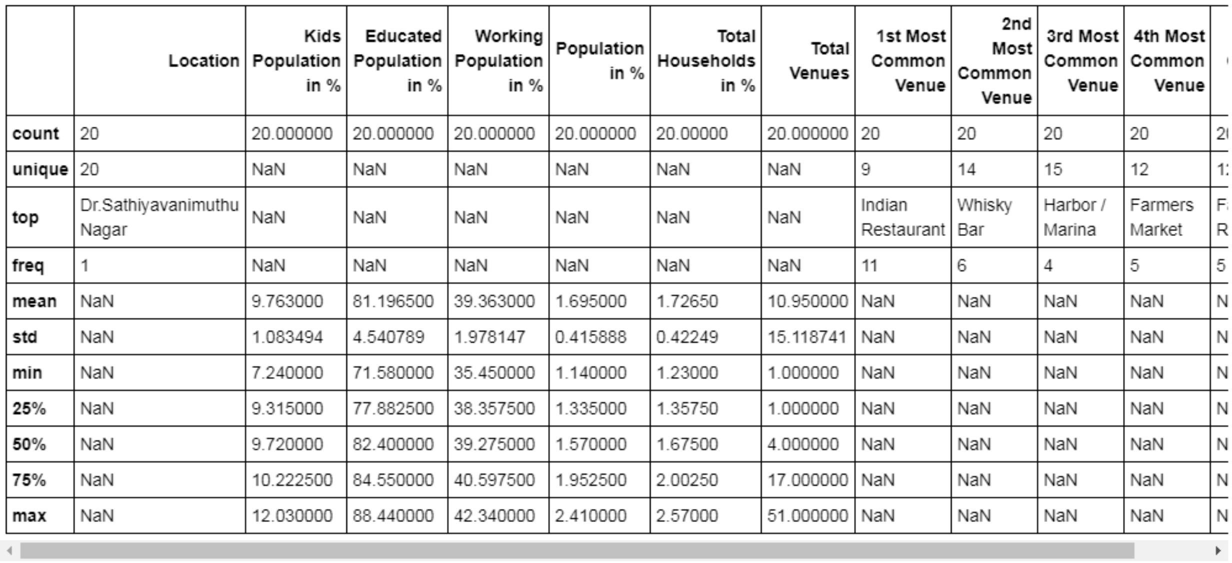
- The cluster contains 9 localities of Chennai city.
- The cluster is densely populated with higher population and households mean than the other 2 clusters, also standard deviation is less compared to others.
- Education population mean is higher in this cluster as well.
- 1st common venue is Indian Restaurant in this cluster as well.
- Kids population mean is higher than other clusters and the standard deviation is very low.
- Total venue variance is higher ranging between 1 and 35.



	Location	Kids Population in %	Educated Population in %	Working Population in %	Population in %	Total Households in %	Total Venues	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th M Comn Ve
count	9	9.000000	9.000000	9.000000	9.000000	9.000000	9.000000	9	9	9	9	9
unique	9	NaN	NaN	NaN	NaN	NaN	NaN	4	7	6	7	7
top	Kodambakkam	NaN	NaN	NaN	NaN	NaN	NaN	Indian Restaurant	Whisky Bar	Harbor / Marina	Farmers Market	Fast Fo Restau
freq	1	NaN	NaN	NaN	NaN	NaN	NaN	4	3	3	2	2
mean	NaN	10.393333	82.506667	39.235556	3.496667	3.586667	9.777778	NaN	NaN	NaN	NaN	NaN
std	NaN	0.779744	1.417780	1.133921	0.456755	0.463276	12.285945	NaN	NaN	NaN	NaN	NaN
min	NaN	9.290000	79.410000	37.670000	2.910000	2.970000	1.000000	NaN	NaN	NaN	NaN	NaN
25%	NaN	9.780000	82.120000	38.040000	3.050000	3.150000	1.000000	NaN	NaN	NaN	NaN	NaN
50%	NaN	10.650000	82.570000	39.380000	3.680000	3.740000	3.000000	NaN	NaN	NaN	NaN	NaN
75%	NaN	10.820000	83.500000	39.690000	3.780000	3.950000	11.000000	NaN	NaN	NaN	NaN	NaN
max	NaN	11.330000	84.450000	41.170000	4.040000	4.170000	35.000000	NaN	NaN	NaN	NaN	NaN

5.3 Cluster 3 Analysis

- The Cluster is moderately dense, having 20 members.
- The population and household mean is in between cluster 1 and cluster 2 with less standard deviation.
- High educated population mean like other clusters.



	Location	Kids Population in %	Educated Population in %	Working Population in %	Population in %	Total Households in %	Total Venues	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	
count	20	20.000000	20.000000	20.000000	20.000000	20.00000	20.000000	20	20	20	20	2
unique	20	NaN	NaN	NaN	NaN	NaN	NaN	9	14	15	12	1
top	Dr.Sathiyavanimuthu Nagar	NaN	NaN	NaN	NaN	NaN	NaN	Indian Restaurant	Whisky Bar	Harbor / Marina	Farmers Market	F R
freq	1	NaN	NaN	NaN	NaN	NaN	NaN	11	6	4	5	5
mean	NaN	9.763000	81.196500	39.363000	1.695000	1.72650	10.950000	NaN	NaN	NaN	NaN	N
std	NaN	1.083494	4.540789	1.978147	0.415888	0.42249	15.118741	NaN	NaN	NaN	NaN	N
min	NaN	7.240000	71.580000	35.450000	1.140000	1.23000	1.000000	NaN	NaN	NaN	NaN	N
25%	NaN	9.315000	77.882500	38.357500	1.335000	1.35750	1.000000	NaN	NaN	NaN	NaN	N
50%	NaN	9.720000	82.400000	39.275000	1.570000	1.67500	4.000000	NaN	NaN	NaN	NaN	N
75%	NaN	10.222500	84.550000	40.597500	1.952500	2.00250	17.000000	NaN	NaN	NaN	NaN	N
max	NaN	12.030000	88.440000	42.340000	2.410000	2.57000	51.000000	NaN	NaN	NaN	NaN	N

6. Conclusion

The objective of the location segmentation to assist in finding a suitable location to open a restaurant is achieved. Below are the findings of each cluster:

- Cluster 1 – Dense cluster, sparsely populated areas, high education population.
- Cluster 2 – Sparse cluster, densely populated areas, high kids population.
- Cluster 3 – Moderately dense cluster, moderately populated, high education population.

With the help of findings, we can assist any decision maker in arriving at the solution to the problem statement. Further, there is a scope for improvement in clustering of localities by trying k-Means with different k-values and also by trying other clustering methods such as hierarchical, DBSCAN, etc.

****End of Document****