Clusterting the City of Chennai

AKASH TK

August 14, 2020

1. Introduction

1.1 Background

Chennai is a beautiful coastal city situated in the eastern coast of India. It has many merits and heritages under its belt, one of them being the second oldest corporation in the world (next to London). Chennai being famous for its tradition, culture, heritage, and architcture attracts large number of tourist every day. On top of this, the job opportunities that Chennai offers attracts a lot of migrants particularly from towns and villages of the southern states of India. For such migrants, the city of herigate, culture, and opportunity could easily turn into a nightmare due to its size, population, and complexity. Most people who migrate to chennai, find it expensive to survive. Given this fact, we can clearly see the impact of a wrong decision about the choice of residence or restaurant or hospital on such migrant. Even a slightly inaccurate decision would burn a huge hole in their pocket. This is where a some guidance and wisdom could be of great help. With a little guidance and information, such migrants would be able to make the right choice, that

best fits their needs.

1.2 Problem

To help such migrants in making wise decisions, we needs to collect details on the essential accessories and their distribution across the city. With this details, we can split the city into various zones based and highlight the density of distribution of essential accessories across these zones. Such a distribution would guide such migrants in selecting the right place of residence.

1.3 Interest

This project would be of interest particularly for people who migrate to Chennai. As the migrants count is very large, this project has the potential to impact a large nuimber of people. On top of this, those who are planning to start a new business can also leverage this project and its data to get insights on competetions in various zones/regions of the city. This way, they could also choose the right place for getting started with their business.

2. Data acquisition and cleaning

2.1 Data Requirement

To make this project a reality, we would need a huge amount of location data. Location data is the data that describes a location by specifying its location (coordinates) and describes its neighbourhood. We would

need data on various accessories locations in and around Chennai.

2.2 Data Source

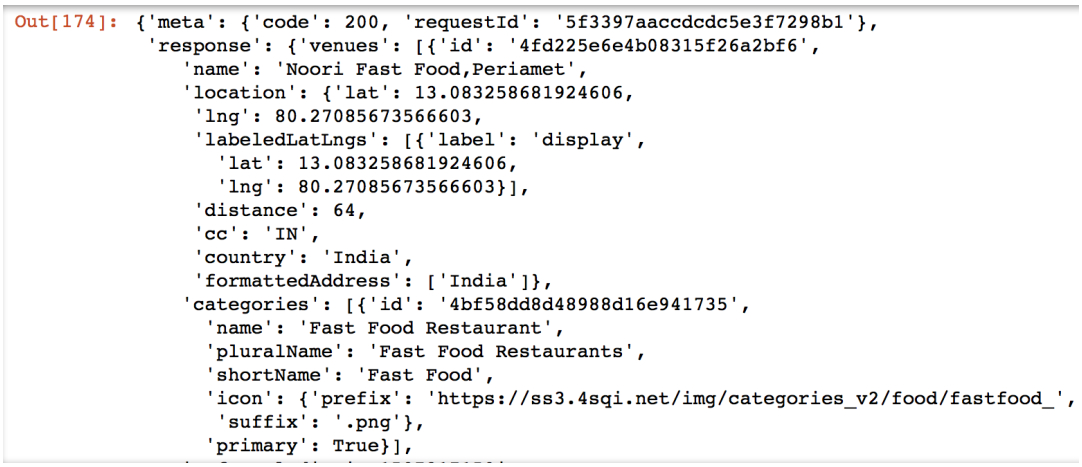
Location data can be obtained from various location data providers like Google. We would be using the APIs of one such provider called 'Foursquare'. Foursquare has several APIs for different types of location data. We would be using the 'search' API to search for essential accessories distribution across the city. As we need to focus on multiple accessories (like food, transportation, health care, shopping, ATMs, schools etc) we would be making different API calls. For example, we would be using the below API to get the list of Restaurants in and around chennai:

```
url ='https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&v={}&query={}&radius={}
url
```

```
'https://api.foursquare.com/v2/venues/search?client_id=GGBJJGIIN3PC[redacted]ient_s
=KMF[redacted]4&ll=13.0827,80.2707&v=20180604&query=food&radius=5000000&limit=
```

Data

The output of this API would be a JSON with numerous JSON objects as shown below:



```
Out[174]: {'meta': {'code': 200, 'requestId': '5f3397aaccdcdc5e3f7298b1'},
           'response': {'venues': [{'id': '4fd225e6e4b08315f26a2bf6',
              'name': 'Noori Fast Food,Periamet',
              'location': {'lat': 13.083258681924606,
               'lng': 80.27085673566603,
               'labeledLatLngs': [{'label': 'display',
                 'lat': 13.083258681924606,
                 'lng': 80.27085673566603}],
               'distance': 64,
               'cc': 'IN',
               'country': 'India',
               'formattedAddress': ['India']},
              'categories': [{'id': '4bf58dd8d48988d16e941735',
                'name': 'Fast Food Restaurant',
                'pluralName': 'Fast Food Restaurants',
                'shortName': 'Fast Food',
                'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/fastfood_',
                 'suffix': '.png'},
                'primary': True}],
```

Each JSON object represents a restaurant and has details about the restaurant like its location, distance from the center of the city, category etc.

2.3 Cleaning

As there is a lot of details that we wouldn't be using in our project, we would have to clean the data and extract only the required details. This is the most important and time consuming step of the entire project. As the data is in JSON format, we should use appropriate technique to pull the required information from the API response.

Once the appropriate data is obtained using the JSON Object keys, we can store them in seperate lists to be used later for plotting

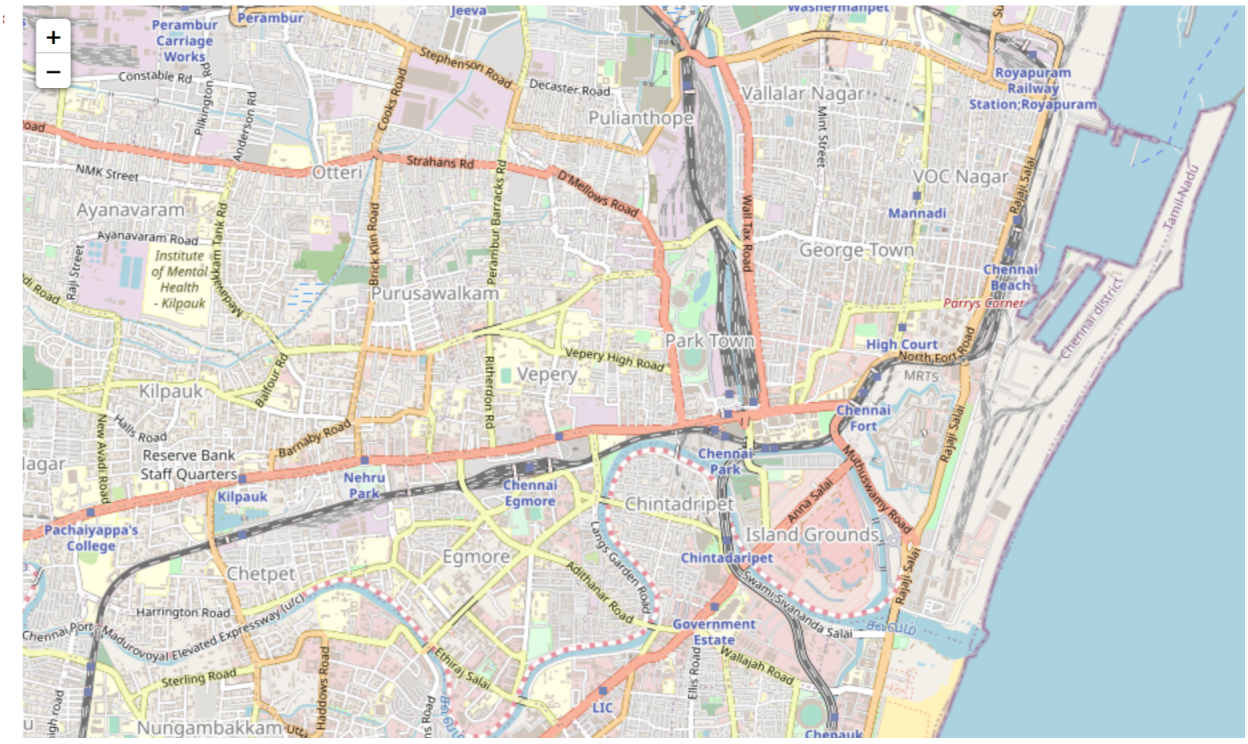
3. EDA

As the data used in this project is primarily the coordinates, name of the place and the category, data munging was not extensively needed. However, we had to create a map of Chennai and add these points as feature group to the map. This provided with opportunity to visualize the map everytime the locations were plotted.

Map of Chennai

The first in EDA is to plot the map of chennai. Folium is used for this purpose as this is an open source and easy to use module of python. The map of Chennai looked as below:

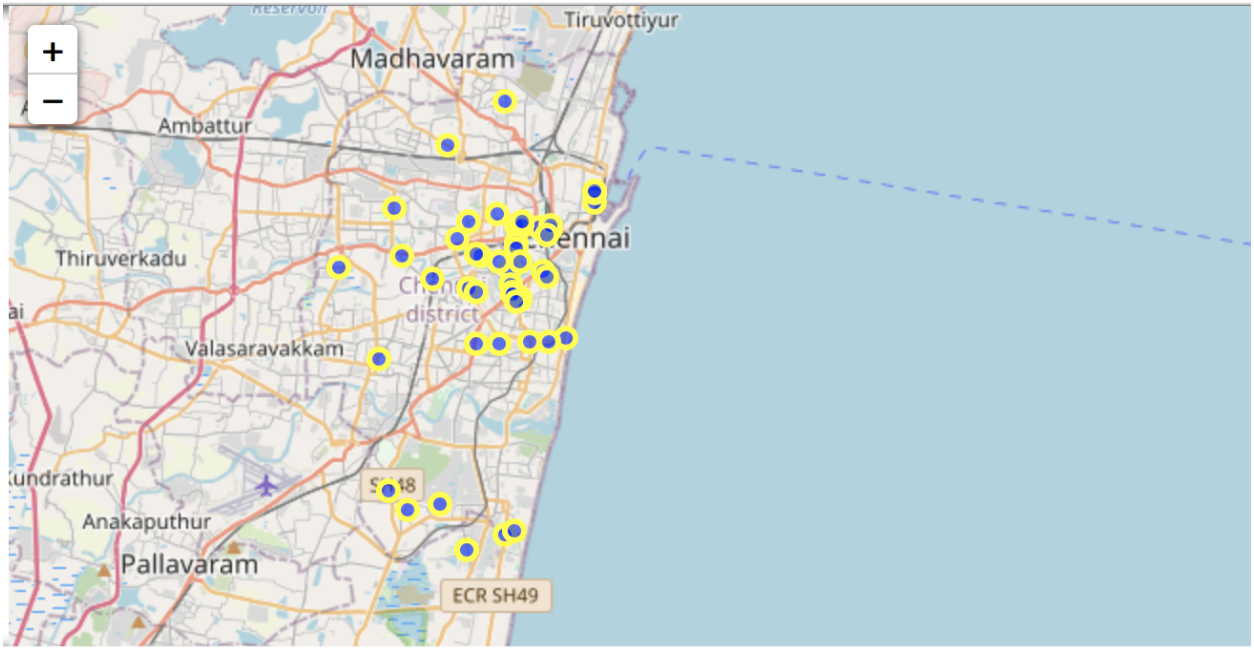
```
chennai_map = folium.Map(location=[13.0827,80.2707],zoom_start=14)

chennai_map
```



Folium provides the option to input the coordinates of the location and the zoom level. This makes it easy to get the right amount of focus on the city. As we can see, we have got the map of the city of chennai as the first step.

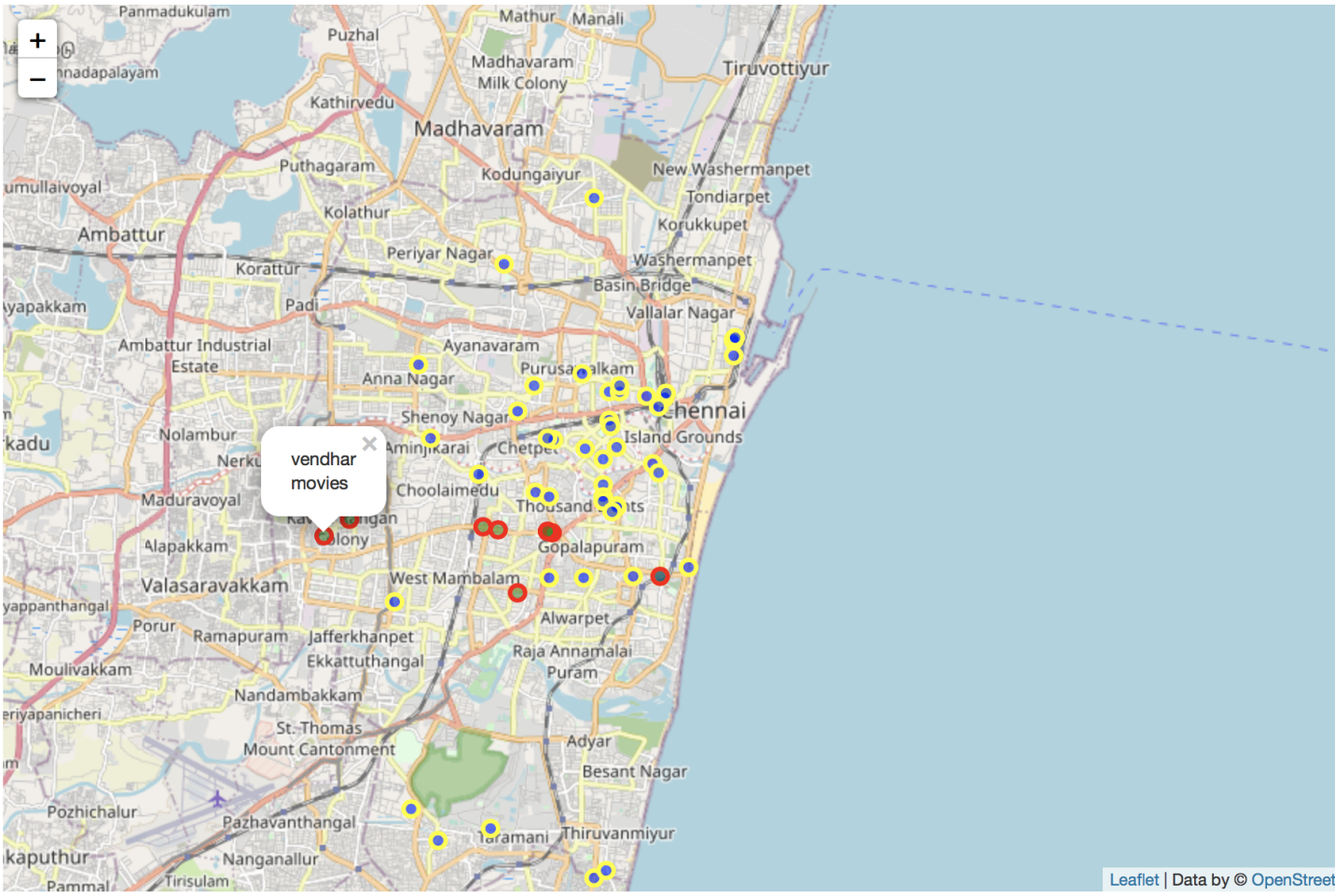
Plotting Restaurants

After the list of restaurants was obtained, they were marked on the map to get a visual on the distribution density of the restaurants.



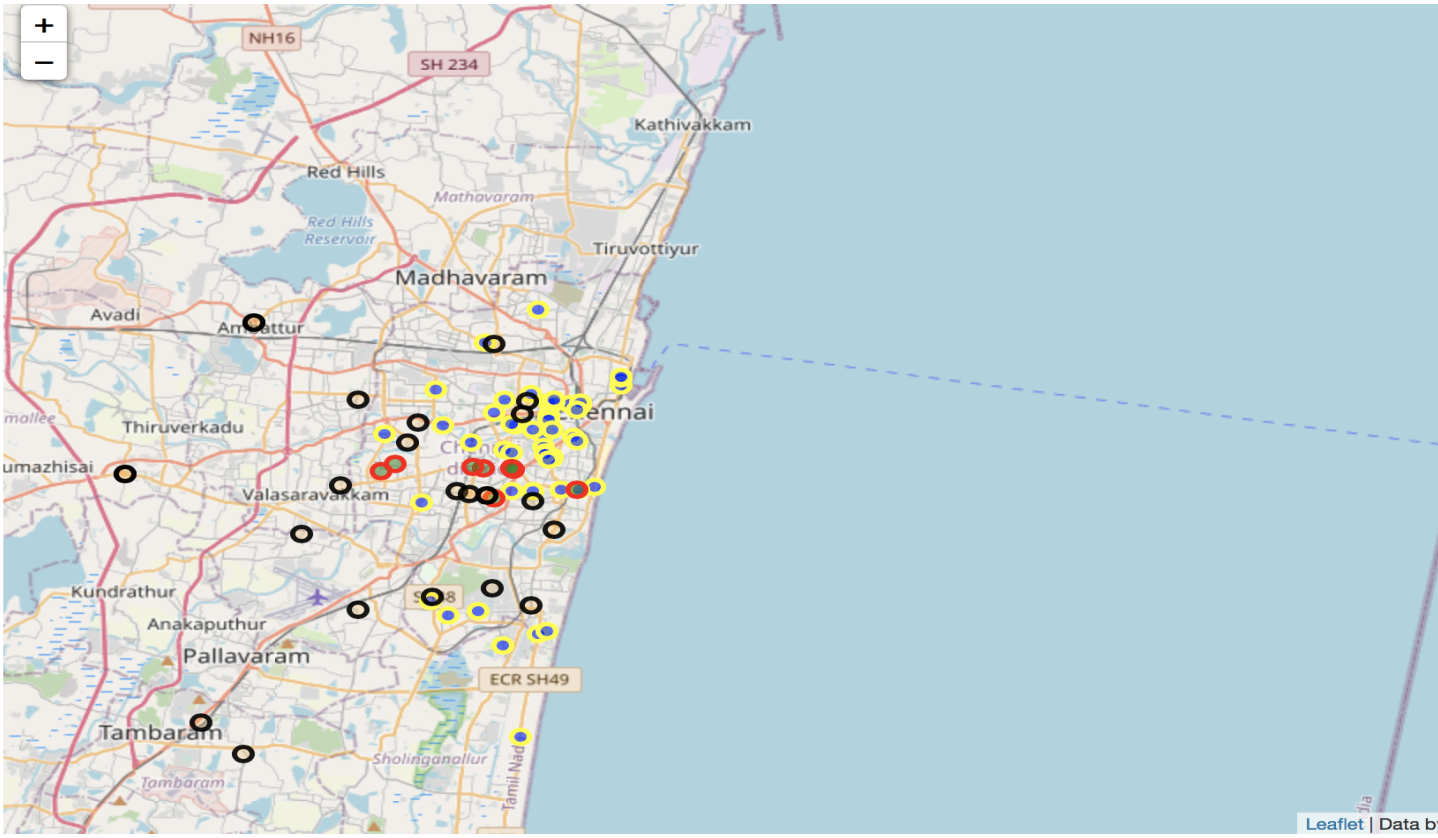
One of the biggest advantages with folium is the interactive nature of these maps. We could hover on these points, click on them to get details about these markers. This was any user can click on a point and get information about them.

Plotting Movie Complexes



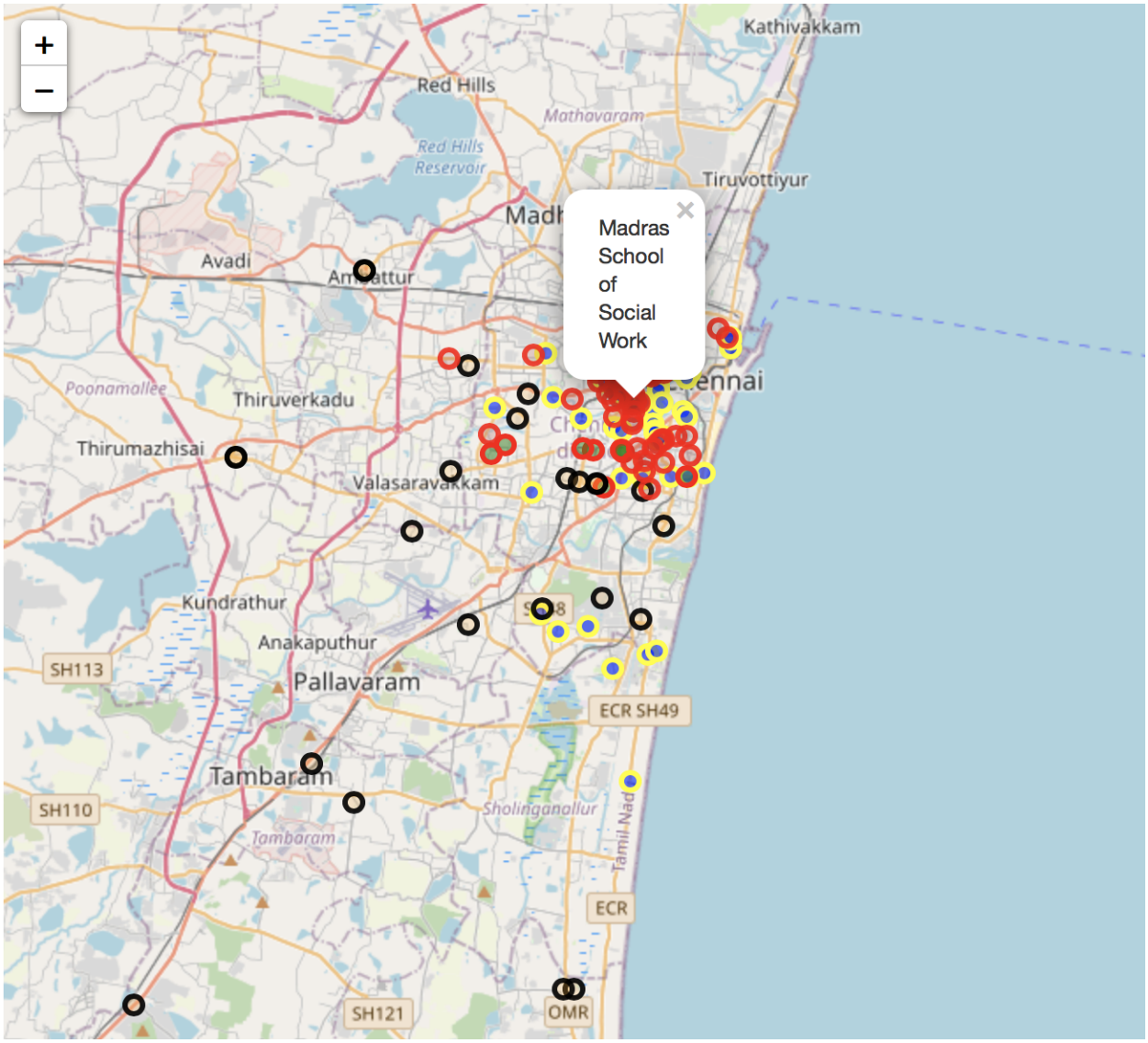
As it can be seen, red color is used to represent cinema complexes. We can also see that the name of the complex is added as an attribute to the location. This means, one could click on the point and get the name of the place. One can also show other interesting details on the graph along with the name of the venue. We can see that the movie complex count is small as compared to restaurants. However as this is not an essential accessory, users could give this least priority while looking for a place of residence.

Shopping places



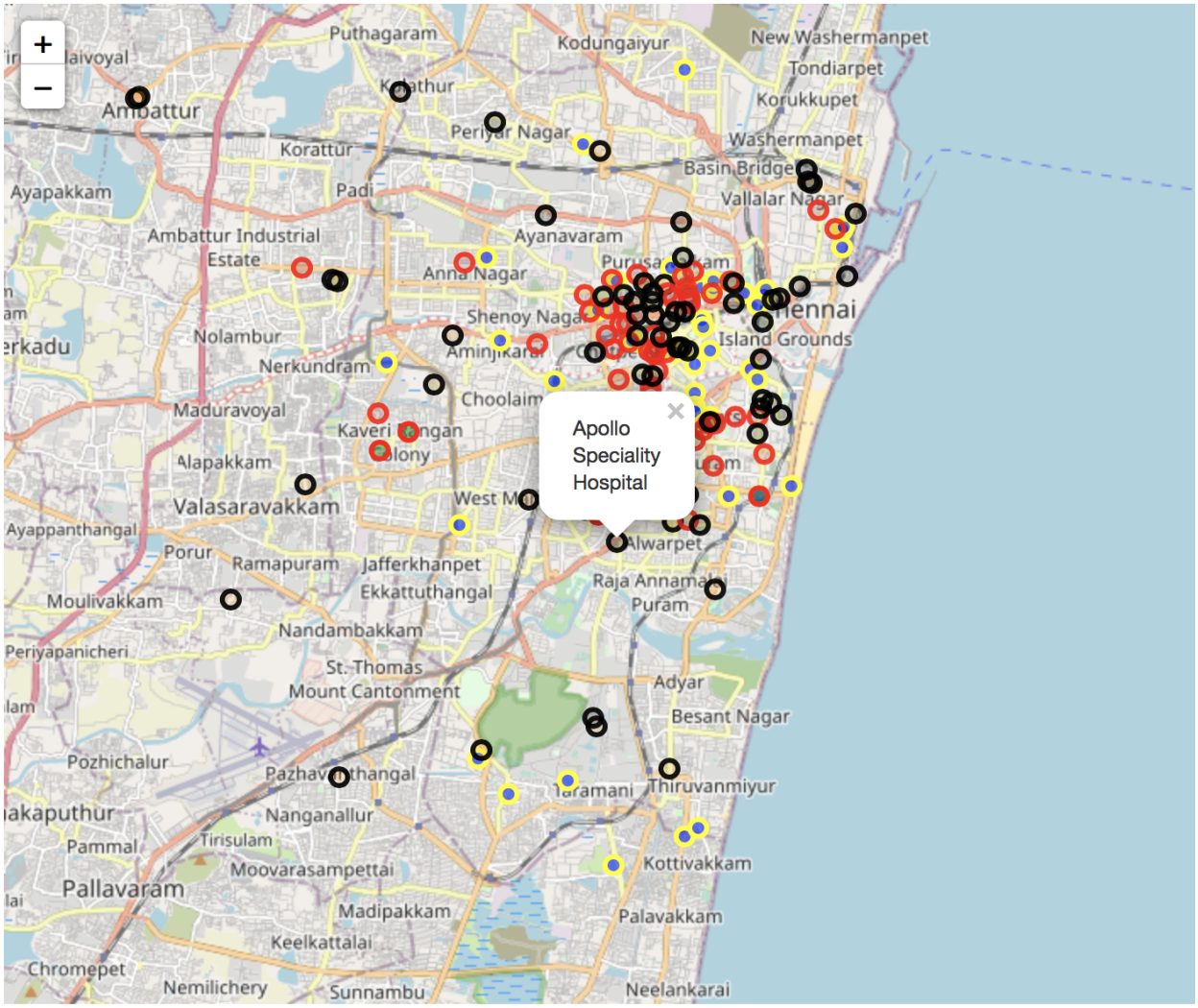
Shopping complexes are also added to the map and are highlighted using black circles. We note that number of shopping complexes is relatively less as compared to restaurants. This implies users should make sure atleast one shopping complex is present in their viscinity. However if its a user who is planning to start his/her own shopping complex, then they could prefer locations which has multiple restaurants but no shopping complex.

Schools



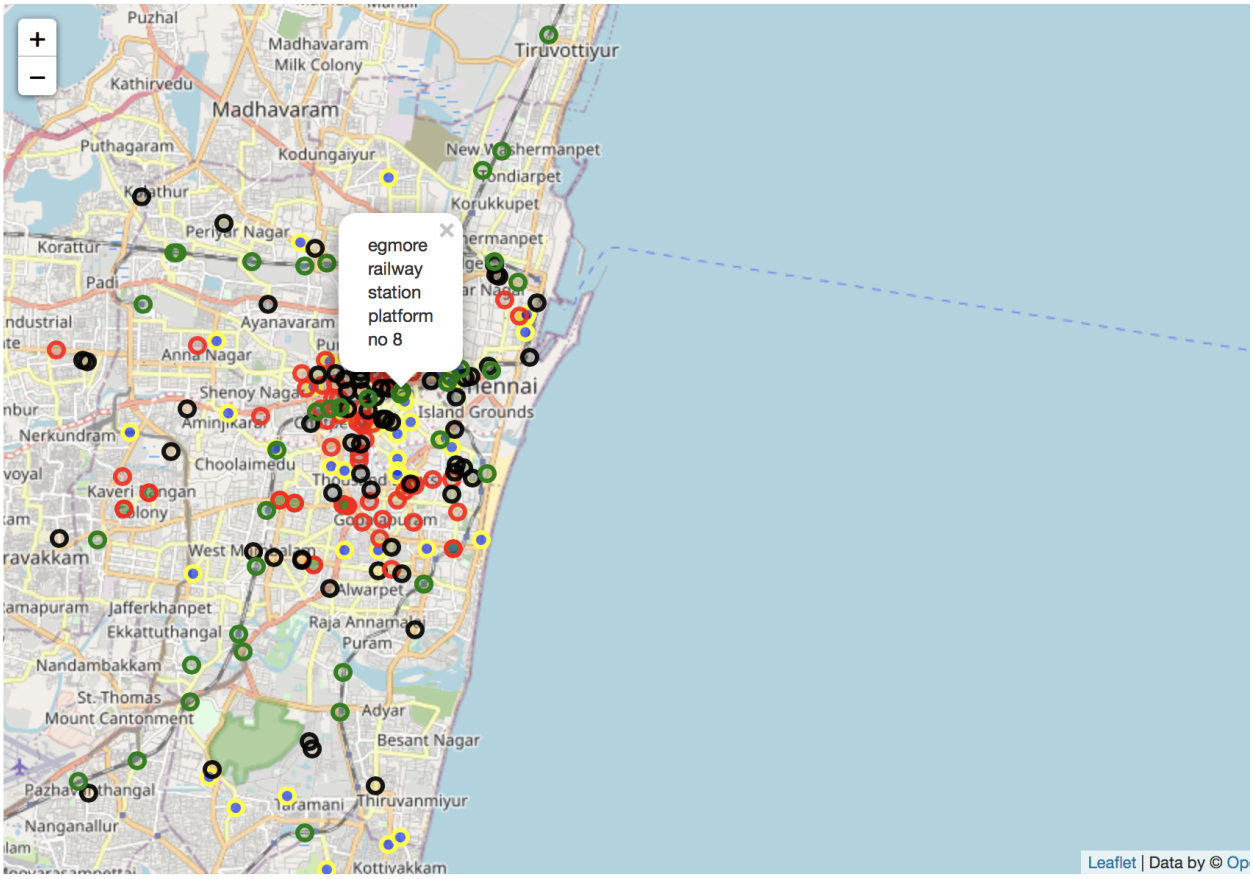
We also see that there is a lot of schools in Chennai. The distribution is schools is almost equal to that of restaurant. We can also infer that one is more likely to find a school anywhere in Chennai. As there are more schools, this can also be treated with relatively lesser priority while searching for the right location.

Hospitals



Hospitals in and around Chennai are plotted using a Black with orange color combination. We see that there is descent density of hospitals across the city. However its count is less when compared to restaurants or schools. Hence one should make sure to find a place near a hospital.

Railways Stations



Another important aspect to consider while looking for a residence is transportation. Chennai is truely well connected via rail routes and people can easily find a place close to a railway station as there is plenty of them. We can also see the same from the above image. Railway stations are plotted and are represented using Green color. As we can see there is atleast one railway station in most of the places. This means, commutation is not a big deal in Chennai and one can keep this at the bottom of their priorityn list.

4. Methodology

We would be using the technique of Clustering to automatically group the places into 5 different zones. The clustering is done based on the distance of the places from automatically, and randomly selected centroid points.

K-Means clustering is used to perform this as this is one of the simplest yet, effecient technique. This techinque has the following advantages:

1. Easy to Understand
2. Easy to implement
3. Easy to interprete

k-means clustering is a method of vector quantization that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. It is popular for cluster analysis in data mining. k-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. For instance, better Euclidean solutions can be found using k-medians and k-medoids.

The problem is computationally difficult (NP-hard); however, efficient heuristic algorithms converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both k-means and Gaussian mixture modeling. They both use cluster centers to model the data; however, k-means clustering tends to find clusters of comparable spatial extent, while the

expectation-maximization mechanism allows clusters to have different shapes.

The algorithm has a loose relationship to the k-nearest neighbor classifier, a popular machine learning technique for classification that is often confused with k-means due to the name. Applying the 1-nearest neighbor classifier to the cluster centers obtained by k-means classifies new data into the existing clusters. This is known as nearest centroid classifier.

Importing all the requiremed packages to get started with the process

```
import random # library for random number generation
import numpy as np # library for vectorized computation
import pandas as pd # library to process data as dataframes

import matplotlib.pyplot as plt # plotting library
# backend for rendering plots within the browser
%matplotlib inline

from sklearn.cluster import KMeans

print('Libraries imported.')
```

```
Libraries imported.
```

After importing the packages, we use clustering to cluster the data points into 5 different cluster.

```
k_means.fit(X)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=5, n_init=12, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

Its always nice to have a peek into our results. Lets looks at the some of the classifications and the centroids of these clusters.

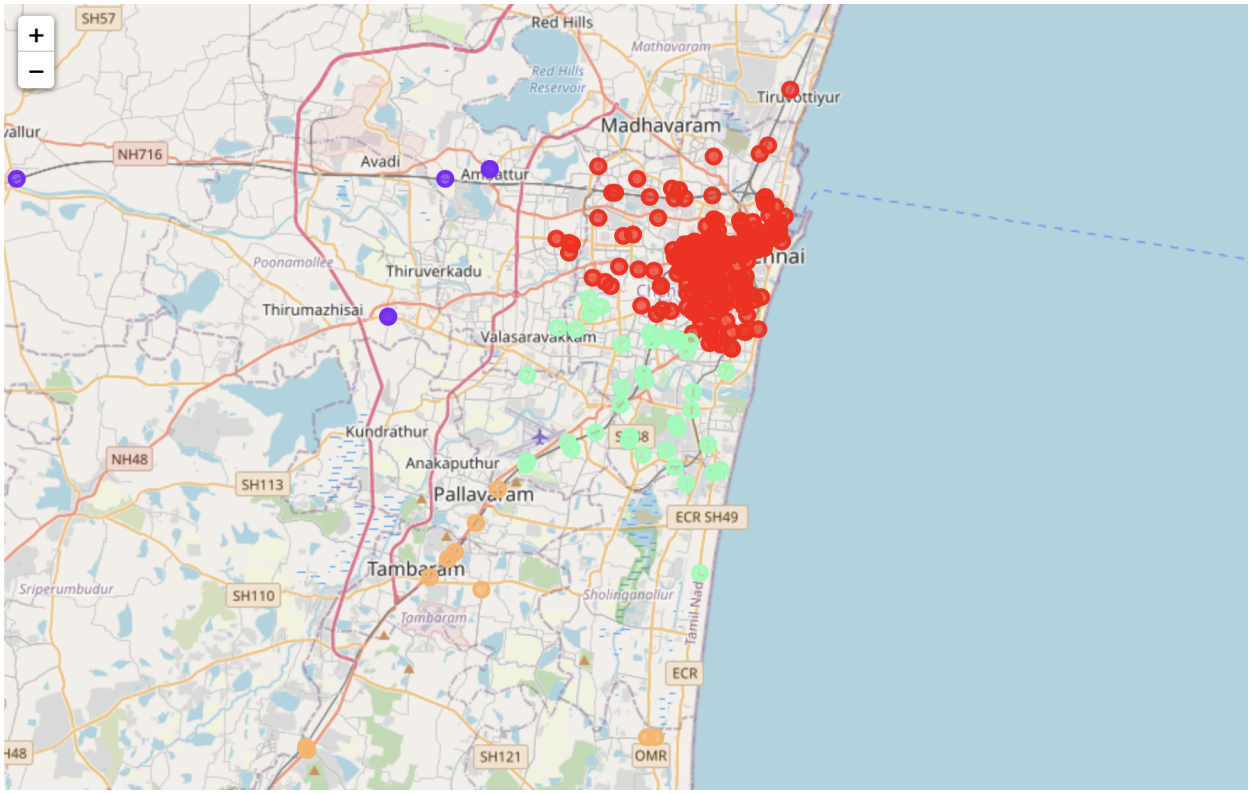
```
k_means_labels = k_means.labels_
k_means_labels
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0,
       0, 0, 0, 3, 0, 0, 3, 3, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 3,
       0, 0, 0, 0, 3, 3, 0, 0, 0, 0, 0, 3, 0, 3, 3, 0, 3, 0, 0, 0, 0, 3,
       3, 0, 0, 3, 1, 3, 3, 1, 3, 3, 1, 3, 1, 3, 3, 4, 4, 4, 4, 4, 2, 2,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 3, 0, 0, 4, 4, 3, 4, 0, 3, 3, 0,
       1, 3, 0, 4, 0, 0, 4, 3, 0, 0, 1, 3, 3, 0, 3, 3, 3, 3, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0], dtype=int32)
```

```
k_means_cluster_centers = k_means.cluster_centers_
k_means_cluster_centers
```

```
array([[13.07728064, 80.25766778],
       [13.09630326, 80.08874002],
       [13.63672295, 79.41909742],
       [13.01154391, 80.22247651],
       [12.90206558, 80.13839284]])
```

5. Result



Our analysis shows that although there is a great number of provisions are clustered around the north/central Chennai (in, Chennai Park, Egmore, Kipaul, Purusawalkam). However, the density of essential provisions decreases as we move towards Adyar, Mambalam, St. Thomas Mount. It becomes even lesser as we start moving down South and far West. Interestinggl, most of the IT firms in chennai are situated in and around the zones with second and third highest density of essential provisions. This means, if someone has an opportunity to work in an IT firm in Chennai, they can safely search for a residence in such zones. Another interesting observation is that such zones have covered all the basic requirements of the neighbourhood like food, ATM, hospitals etc. Its just that the options available to their

neighbourhood is limited.

As most part of Chennai has railway coverage, transportations remains equally distributed across all these zones and makes the commutation simple. As these low density zones has good number of ATMs, hospitals, and schools,it would be a move to get settled in such regions as they also turn out to be a cost effective solution.

6. Conclusion

The idea of this project was to cluster the city of Chennai into various zones based on the density of distribution of essential accessories to help people who are new to the city in choosing the right residential location.

We were able to achieve this by clustering the places into different zones based on distribution density. The interesting thing with this project is its data source. As this project uses APIs to perform the clustering, we can stay assured that this project would get updated with time automatically.