B.E CSE VI Q BATCH

MACHINE LEARNING PROJECT

TEAM MEMBERS:

NAME – REGNO

VISHUNUPRIYA N – 2019103599

NAVVYA L – 2019103548

ISHWARYA RANI M – 2019103527

DATE: 23-05-2022

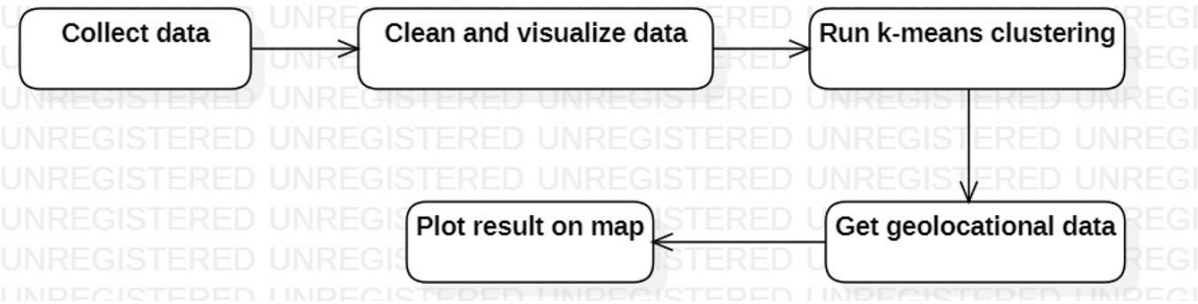
TITLE OF THE PROJECT –

HOUSING EXPLORATORY ANALYSIS OF GEOLOCATIONAL DATA

Problem Statement

This project involves the use of K-Means Clustering to find the best accommodation for students in a city, by classifying accommodation for incoming students on the basis of their preferences on amenities, budget and proximity to the location.

Overall Architecture Diagram



Modules

1. DATA COLLECTION

    Fetch Datasets from the relevant locations

2. DATA CLEANING

    Clean the Datasets to prepare them for analysis (via Pandas).

    Visualise the data using boxplots (Using Matplotlib /Seaborn /Pandas)

3. CLUSTER THE LOCATIONS

    Fetch Geolocational Data from the Foursquare API (REST APIs).

    Use K-Means Clustering to cluster the locations (Using ScikitLearn).

4. PLOT RESULTS ON MAP

    Present findings on a map. (Using Folium/Seaborn)

Implementation details

DATASET USED: food_coded.csv

food_coded.csv ×

1 to 10 of 125 entries Filter

GPA	Gender	breakfast	calories_chicken	calories_day	calories_scone	coffee	comfort_food	comfort_food_reasons	comfort_food_reasons_coded	cook	comfort_
2.4	2	1	430	nan	315	1	none	we dont have comfort	9	2	9
3.654	1	1	610	3	420	2	chocolate, chips, ice cream	Stress, bored, anger	1	3	1
3.3	1	1	720	4	420	2	frozen yogurt, pizza, fast food	stress, sadness	1	1	1
3.2	1	1	430	3	420	2	Pizza, Mac and cheese, ice	Boredom	2	2	2

Module 1: DATA COLLECTION

HOUSING EXPLORATORY ANALYSIS OF GEOLOCATIONAL DATA

DATA COLLECTION

```
[ ] import pandas as pd
    data=pd.read_csv("food_coded.csv")
```

```
data
```

	GPA	Gender	breakfast	calories_chicken	calories_day	calories_scone	coffee	comfort_food	comfort_food_reasons	comfort_food_reasons_coded	...
0	2.4	2	1	430	NaN	315.0	1	none	we dont have comfort	9.0	...
1	3.654	1	1	610	3.0	420.0	2	chocolate, chips, ice cream	Stress, bored, anger	1.0	...
2	3.3	1	1	720	4.0	420.0	2	frozen yogurt, pizza, fast food	stress, sadness	1.0	...
3	3.2	1	1	430	3.0	420.0	2	Pizza, Mac and cheese, ice cream	Boredom	2.0	...
4	3.5	1	1	720	2.0	420.0	2	Ice cream, chocolate, chips	Stress, boredom, cravings	1.0	...
...	...	...	...	...	...	...	...	...	...	...	...
120	3.5	1	1	610	4.0	420.0	2	wine. mac and cheese, pizza, ice cream	boredom and sadness	NaN	...
121	3	1	1	265	2.0	315.0	2	Pizza / Wings / Cheesecake	Loneliness / Homesick / Sadness	NaN	...
122	3.882	1	1	720	NaN	420.0	1	rice, potato, seaweed soup	sadness	NaN	...
123	3	2	1	720	4.0	420.0	1	Mac n Cheese, Lasagna, Pizza	happiness, they are some of my favorite foods	NaN	...
124	3.9	1	1	430	NaN	315.0	2	Chocolates, pizza, and Ritz.	hormones, Premenstrual syndrome.	NaN	...

125 rows × 61 columns

Module 2: DATA CLEANING AND VISUALIZATION

DATA CLEANING

```
[ ] data.columns
```

```
Index(['GPA', 'Gender', 'breakfast', 'calories_chicken', 'calories_day',
       'calories_scone', 'coffee', 'comfort_food', 'comfort_food_reasons',
       'comfort_food_reasons_coded', 'cook', 'comfort_food_reasons_coded.1',
       'cuisine', 'diet_current', 'diet_current_coded', 'drink',
       'eating_changes', 'eating_changes_coded', 'eating_changes_coded1',
       'eating_out', 'employment', 'ethnic_food', 'exercise',
       'father_education', 'father_profession', 'fav_cuisine',
       'fav_cuisine_coded', 'fav_food', 'food_childhood', 'fries', 'fruit_day',
       'grade_level', 'greek_food', 'healthy_feeling', 'healthy_meal',
       'ideal_diet', 'ideal_diet_coded', 'income', 'indian_food',
       'italian_food', 'life_rewarding', 'marital_status',
       'meals_dinner_friend', 'mother_education', 'mother_profession',
       'nutritional_check', 'on_off_campus', 'parents_cook', 'pay_meal_out',
       'persian_food', 'self_perception_weight', 'soup', 'sports', 'thai_food',
       'tortilla_calories', 'turkey_calories', 'type_sports', 'veggies_day',
       'vitamins', 'waffle_calories', 'weight'],
      dtype='object')
```

```
[ ] column=['cook','eating_out','employment','ethnic_food', 'exercise','fruit_day','income','on_off_campus','pay_meal_out','sports','veggies_day']
```

```
[ ] d=data[column]
```

```
d
```

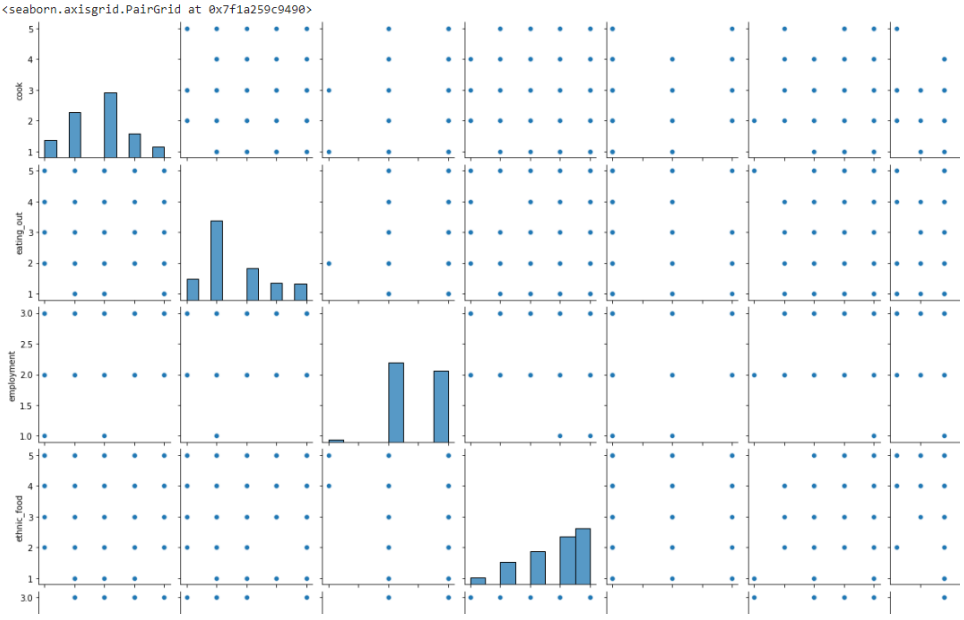
	cook	eating_out	employment	ethnic_food	exercise	fruit_day	income	on_off_campus	pay_meal_out	sports	veggies_day
0	2.0	3	3.0	1	1.0	5	5.0	1.0	2	1.0	5
1	3.0	2	2.0	4	1.0	4	4.0	1.0	4	1.0	4
2	1.0	2	3.0	5	2.0	5	6.0	2.0	3	2.0	5
3	2.0	2	3.0	5	3.0	4	6.0	1.0	2	2.0	3
4	1.0	2	2.0	4	1.0	4	6.0	1.0	4	1.0	4
...	...	...	...	...	...	...	...	...	...	...	...
120	3.0	2	1.0	4	2.0	5	4.0	3.0	4	1.0	5
121	3.0	4	3.0	3	2.0	4	2.0	1.0	4	NaN	5
122	3.0	3	3.0	5	2.0	4	2.0	1.0	4	2.0	4
123	3.0	5	2.0	2	1.0	5	4.0	1.0	3	2.0	3
124	NaN	1	2.0	3	2.0	3	5.0	1.0	3	2.0	4

125 rows × 11 columns

DATA EXPLORATION AND VISUALIZATION

```
[ ] import seaborn as sns
    sns.pairplot(d)
```

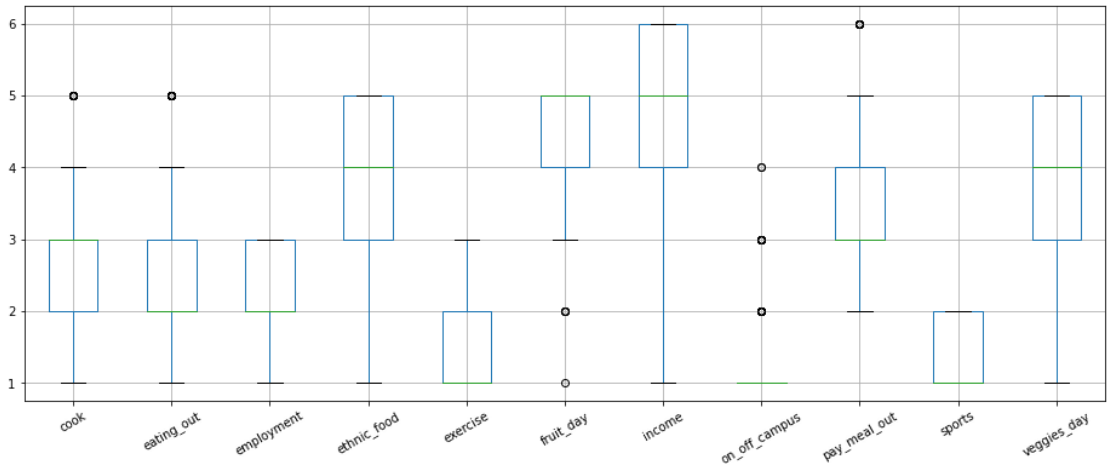
```
<seaborn.axisgrid.PairGrid at 0x7f1a259c9490>
```



BOXPLOT OF DATASET

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
% matplotlib inline
ax=d.boxplot(figsize=(16,6))
ax.set_xticklabels(ax.get_xticklabels(),rotation=30)
```

```
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecationWarning: Creating an ndarray
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
[Text(0, 0, 'cook'),
 Text(0, 0, 'eating_out'),
 Text(0, 0, 'employment'),
 Text(0, 0, 'ethnic_food'),
 Text(0, 0, 'exercise'),
 Text(0, 0, 'fruit_day'),
 Text(0, 0, 'income'),
 Text(0, 0, 'on_off_campus'),
 Text(0, 0, 'pay_meal_out'),
 Text(0, 0, 'sports'),
 Text(0, 0, 'veggies_day')]
```




```
[ ] d.shape
```

```
(125, 11)
```

```
[ ] s=d.dropna()
```

```
[ ] pip install minisom
```

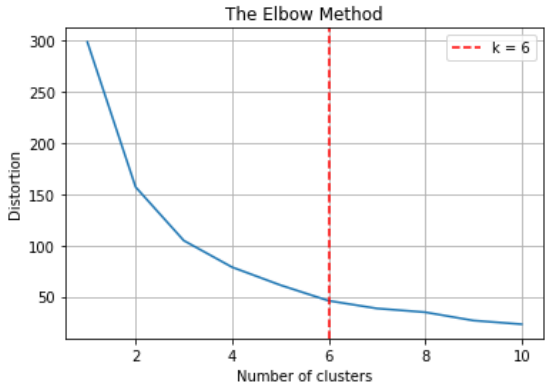
```
Collecting minisom
  Downloading MiniSom-2.3.0.tar.gz (8.8 kB)
Building wheels for collected packages: minisom
  Building wheel for minisom (setup.py) ... done
  Created wheel for minisom: filename=MiniSom-2.3.0-py3-none-any.whl size=9018 sha256=ac31bb6ea11c7ac3b8caf832f541200862d9af7968042852d94d775a20821e21
  Stored in directory: /root/.cache/pip/wheels/d4/ca/4a/488772b0399fec45ff53132ed14c948dec4b30deee3a532f80
Successfully built minisom
Installing collected packages: minisom
Successfully installed minisom-2.3.0
```

Module 3 – CLUSTER THE LOCATIONS

RUNNING KMEANS CLUSTRING ON THE DATA

```
[ ]  ## for data
     import numpy as np
     import pandas as pd
     ## for plotting
     import matplotlib.pyplot as plt
     import seaborn as sns
     ## for geospatial
     import folium
     import geopy
     ## for machine learning
     from sklearn import preprocessing, cluster
     import scipy
     ## for deep learning
     import minisom
```

```
[ ]  f=['cook','income']
     X = s[f]
     max_k = 10
     ## iterations
     distortions = []
     for i in range(1, max_k+1):
         if len(X) >= i:
            model = cluster.KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
            model.fit(X)
            distortions.append(model.inertia_)
     ## best k: the lowest derivative
     k = [i*100 for i in np.diff(distortions,2)].index(min([i*100 for i
          in np.diff(distortions,2)]))
     ## plot
     fig, ax = plt.subplots()
     ax.plot(range(1, len(distortions)+1), distortions)
     ax.axvline(k, ls='--', color="red", label="k = "+str(k))
     ax.set(title='The Elbow Method', xlabel='Number of clusters',
            ylabel="Distortion")
     ax.legend()
     ax.grid(True)
     plt.show()
```



GET GEOLOCATIONAL DATA

```
from pandas import json_normalize
import folium
from geopy.geocoders import Nominatim
import requests
CLIENT_ID = "KTCJJ2YZ2143QHEZ2JAQS4FJIO5DLSDO0YN4YBXPMI5NKTEF" # your Foursquare ID
CLIENT_SECRET = "KNG2LO22BPLHN1E3OAHWLYQ5PQBN14XYZMEMAS0CPJEJKOTR" # your Foursquare Secret
VERSION = '20200316'
LIMIT = 10000
```

```
[ ] url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        13.011139208115479, 80.23544117310388,
        30000,
        LIMIT)
```

```
[ ] results = requests.get(url).json()
    results
```


```
            'lat': 12.797460584175763,
            'lng': 80.24848325999668,
            'postalCode': '603 112',
            'state': 'Tamil Nadu'},
           'name': "Vivanta by Taj - Fisherman's Cove",
           'photos': {'count': 0, 'groups': []}}},
         {'reasons': {'count': 0,
           'items': [{'reasonName': 'globalInteractionReason',
             'summary': 'This spot is popular',
             'type': 'general'}]},
          'referralId': 'e-0-4c4be5d5712ac928bb628b6d-99',
          'venue': {'categories': [{'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/italian_',
              'suffix': '.png'},
             'id': '4bf58dd8d48988d110941735',
             'name': 'Italian Restaurant',
             'pluralName': 'Italian Restaurants',
             'primary': True,
             'shortName': 'Italian'}],
           'id': '4c4be5d5712ac928bb628b6d',
           'location': {'address': '14 L Jey Avenue, Akkarai',
            'cc': 'IN',
            'city': 'Chennai',
            'country': 'India',
            'crossStreet': 'East Coast Road',
            'distance': 13017,
            'formattedAddress': ['14 L Jey Avenue, Akkarai (East Coast Road)',
             'Chennai 600119',
             'Tamil Nadu',
             'India'],
            'labeledLatLngs': [{'label': 'display',
              'lat': 12.89536435369259,
              'lng': 80.25232523858824}],
            'lat': 12.89536435369259,
            'lng': 80.25232523858824,
            'postalCode': '600119',
            'state': 'Tamil Nadu'},
           'name': 'Kipling Cafe',
           'photos': {'count': 0, 'groups': []}}}],
       'name': 'recommended',
       'type': 'Recommended Places'}],
     'headerFullLocation': 'Chennai',
     'headerLocation': 'Chennai',
```

```
'headerLocationGranularity': 'city',
'queryRefinements': {'refinements': [{'query': 'Food'},
  {'query': 'Nightlife'},
  {'query': 'Coffee'},
  {'query': 'Shops'},
  {'query': 'Arts'},
  {'query': 'Outdoors'}],
 'target': {'params': {'ll': '13.011139,80.235441', 'radius': '30000'},
  'type': 'path',
  'url': '/venue/explore'}},
'suggestedBounds': {'ne': {'lat': 13.28113947811575,
  'lng': 80.51203859759471},
 'sw': {'lat': 12.741138938115208, 'lng': 79.95884374861305}},
'suggestedFilters': {'filters': [{'key': 'openNow', 'name': 'Open now'}],
 'header': 'Tap to show:'},
'totalResults': 151}}
```

```
[ ]  venues = results['response']['groups'][0]['items']
     nearby_venues = json_normalize(venues)
```

```
[ ] nearby_venues
```

	referralId	reasons.count	reasons.items	venue.id	venue.name	venue.location.address	venue.location.crossStreet	venue.location.lat	venue.location.lng	venue.location.labeledLatLngs	...
0	e-0-503f4face4b05b14135984e9-0	0	[{'summary': 'This spot is popular', 'type': '...	503f4face4b05b14135984e9	Ottimo Cucina Italiana, ITC Grand Chola	#63 Mount Rd.	Guindy	13.010444	80.220938	[{'label': 'display', 'lat': 13.01044420875714...	...
1	e-0-4d848e465ad3a0932c8dd1fd-1	0	[{'summary': 'This spot is popular', 'type': '...	4d848e465ad3a0932c8dd1fd	ITC Grand Chola	#63 Mount Road, Guindy	Mount Rd	13.010440	80.220669	[{'label': 'display', 'lat': 13.01043960430814...	...
2	e-0-4f72a31ee4b053123f1acd68-2	0	[{'summary': 'This spot is popular', 'type': '...	4f72a31ee4b053123f1acd68	Park Hyatt Chennai	39 Velachery Road Near Raj Bhavan	NaN	13.010554	80.223461	[{'label': 'display', 'lat': 13.01055407430340...	...
3	e-0-4fc30399e4b07ac9fd39a644-3	0	[{'summary': 'This spot is popular', 'type': '...	4fc30399e4b07ac9fd39a644	The Flying Elephant	39 Vellachery Rd	Sardar Patel Rd	13.010472	80.223536	[{'label': 'display', 'lat': 13.01047216830747...	...
4	e-0-50fbfed3d86c1bb70c07680c-4	0	[{'summary': 'This spot is popular', 'type': '...	50fbfed3d86c1bb70c07680c	Luxe Cinemas	Phoenix Market City	Velachery	12.991041	80.216962	[{'label': 'display', 'lat': 12.99104145412169...	...
...	...	...	...	...	...	...	...	...	...	...	...
95	e-0-4bd91a3211dcc928c4fff833-95	0	[{'summary': 'This spot is popular', 'type': '...	4bd91a3211dcc928c4fff833	MGM Beach Resort	East Coast Road	NaN	12.825891	80.246869	[{'label': 'display', 'lat': 12.82589076703705...	...
96	e-0-56cbd3d9cd10af0dc3d732ae-96	0	[{'summary': 'This spot is popular', 'type': '...	56cbd3d9cd10af0dc3d732ae	Sangeetha Drive-in Restaurant	Bangalore Trunk Road	Thirumazhisai	13.045672	80.068746	[{'label': 'display', 'lat': 13.04567212598119...	...
97	e-0-55158708498e237bfc673002-97	0	[{'summary': 'This spot is popular', 'type': '...	55158708498e237bfc673002	Barbeque Nation	No 11, Ground Floor, Ramaniyam Isha,Blk 1, Raj...	OMR	12.943957	80.237865	[{'label': 'display', 'lat': 12.94395700230097...	...
98	e-0-4eae35c2b8f765aba1a5d723-98	0	[{'summary': 'This spot is popular', 'type':	4eae35c2b8f765aba1a5d723	Vivanta by Taj - Fisherman's	Covelong Beach,	Kancheepuram Dist.	12.797461	80.248483	[{'label': 'display', 'lat': 12.79746058417576...	...

ADDING 2 MORE COLUMNS - RESTAURANT AND OTHERS

1.RESTAURANT - Number of restaurant in the radius of 20km

2.OTHERS - Number of Gyms, Parks, etc in the radius of 20km

```
[ ]  resta=[]
     oth=[]
     for lat,long in zip(nearby_venues['venue.location.lat'],nearby_venues['venue.location.lng']):
         url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
           CLIENT_ID,
           CLIENT_SECRET,
           VERSION,
           lat,long,
           1000,
           100)
         res = requests.get(url).json()
         venue = res['response']['groups'][0]['items']
         nearby_venue = json_normalize(venue)
         df=nearby_venue['venue.categories']

         g=[]
         for i in range(0,df.size):
           g.append(df[i][0]['icon']['prefix'].find('food'))
         co=0
         for i in g:
           if i>1:
             co+=1
         resta.append(co)
         oth.append(len(g)-co)
```

```
[ ] nearby_venues['restaurant']=resta
    nearby_venues['others']=oth
    nearby_venues
```

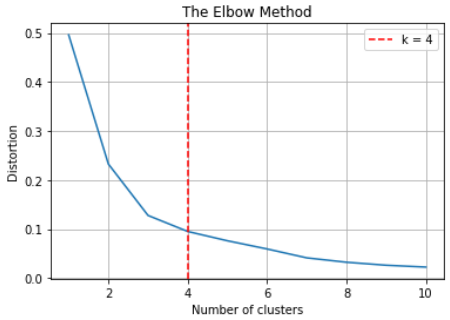
	referralId	reasons.count	reasons.items	venue.id	venue.name	venue.location.address	venue.location.crossStreet	v
0	e-0-503f4face4b05b14135984e9-0	0	[{'summary': 'This spot is popular', 'type': '...	503f4face4b05b14135984e9	Ottimo Cucina Italiana, ITC Grand Chola	#63 Mount Rd.	Guindy	
1	e-0-4d848e465ad3a0932c8dd1fd-1	0	[{'summary': 'This spot is popular', 'type': '...	4d848e465ad3a0932c8dd1fd	ITC Grand Chola	#63 Mount Road, Guindy	Mount Rd	
2	e-0-4f72a31ee4b053123f1acd68-2	0	[{'summary': 'This spot is popular', 'type': '...	4f72a31ee4b053123f1acd68	Park Hyatt Chennai	39 Velachery Road Near Raj Bhavan	NaN	
3	e-0-4fc30399e4b07ac9fd39a644-3	0	[{'summary': 'This spot is popular', 'type': '...	4fc30399e4b07ac9fd39a644	The Flying Elephant	39 Vellachery Rd	Sardar Patel Rd	
4	e-0-50fbfed3d86c1bb70c07680c-4	0	[{'summary': 'This spot is popular', 'type': '...	50fbfed3d86c1bb70c07680c	Luxe Cinemas	Phoenix Market City	Velachery	
...	...	...	...	...	...	...	...	
95	e-0-4bd91a3211dcc928c4fff833-95	0	[{'summary': 'This spot is popular', 'type': '...	4bd91a3211dcc928c4fff833	MGM Beach Resort	East Coast Road	NaN	
96	e-0-56cbd3d9cd10af0dc3d732ae-96	0	[{'summary': 'This spot is popular', 'type': '...	56cbd3d9cd10af0dc3d732ae	Sangeetha Drive-in Restaurant	Bangalore Trunk Road	Thirumazhisai	
97	e-0-55158708498e237bfc673002-97	0	[{'summary': 'This spot is popular', 'type': '...	55158708498e237bfc673002	Barbeque Nation	No 11, Ground Floor, Ramaniyam Isha,Blk 1, Raj...	OMR	

CHANGING THE COLUMN NAME

```
[ ] lat=nearby_venues['venue.location.lat']
    long=nearby_venues['venue.location.lng']
```

RUNNING KMEANS CLUSTERING ON THE DATASET, WITH THE OPTIMAL K VALUE USING ELBOW METHOD

```
[ ] f=['venue.location.lat','venue.location.lng']
    X = nearby_venues[f]
    max_k = 10
    ## iterations
    distortions = []
    for i in range(1, max_k+1):
        if len(X) >= i:
           model = cluster.KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
           model.fit(X)
           distortions.append(model.inertia_)
    ## best k: the lowest derivative
    k = [i*100 for i in np.diff(distortions,2)].index(min([i*100 for i
         in np.diff(distortions,2)]))
    ## plot
    fig, ax = plt.subplots()
    ax.plot(range(1, len(distortions)+1), distortions)
    ax.axvline(k, ls='--', color="red", label="k = "+str(k))
    ax.set(title='The Elbow Method', xlabel='Number of clusters',
           ylabel="Distortion")
    ax.legend()
    ax.grid(True)
    plt.show()
```




```
[ ] city = "Chennai"
    ## get location
    locator = geopy.geocoders.Nominatim(user_agent="MyCoder")
    location = locator.geocode(city)
    print(location)
    ## keep latitude and longitude only
    location = [location.latitude, location.longitude]
    print("[lat, long]:", location)
```

```
Chennai, Chennai District, Tamil Nadu, 600001, India
[lat, long]: [13.0836939, 80.270186]
```

```
[ ] nearby_venues.head()
```

	referralId	reasons.count	reasons.items	venue.id	venue.name	venue.location.address	venue.location.crossStreet	ve
0	e-0-503f4face4b05b14135984e9-0	0	[{'summary': 'This spot is popular', 'type': '...	503f4face4b05b14135984e9	Ottimo Cucina Italiana, ITC Grand Chola	#63 Mount Rd.	Guindy	
1	e-0-4d848e465ad3a0932c8dd1fd-1	0	[{'summary': 'This spot is popular', 'type': '...	4d848e465ad3a0932c8dd1fd	ITC Grand Chola	#63 Mount Road, Guindy	Mount Rd	
2	e-0-4f72a31ee4b053123f1acd68-2	0	[{'summary': 'This spot is popular', 'type': '...	4f72a31ee4b053123f1acd68	Park Hyatt Chennai	39 Velachery Road Near Raj Bhavan	NaN	
3	e-0-4fc30399e4b07ac9fd39a644-3	0	[{'summary': 'This spot is popular', 'type': '...	4fc30399e4b07ac9fd39a644	The Flying Elephant	39 Vellachery Rd	Sardar Patel Rd	
4	e-0-50fbfed3d86c1bb70c07680c-4	0	[{'summary': 'This spot is popular', 'type': '...	50fbfed3d86c1bb70c07680c	Luxe Cinemas	Phoenix Market City	Velachery	

5 rows × 24 columns

```
nearby_venues.columns
```

```
Index(['referralId', 'reasons.count', 'reasons.items', 'venue.id',
       'venue.name', 'venue.location.address', 'venue.location.crossStreet',
       'venue.location.lat', 'venue.location.lng',
       'venue.location.labeledLatLngs', 'venue.location.distance',
       'venue.location.postalCode', 'venue.location.cc', 'venue.location.city',
       'venue.location.state', 'venue.location.country',
       'venue.location.formattedAddress', 'venue.categories',
       'venue.photos.count', 'venue.photos.groups',
       'venue.location.neighborhood', 'venue.venuePage.id', 'restaurant',
       'others'],
      dtype='object')
```

DATA CLEANING PROCESS FOR EXTRACTING NECESSARY COLUMNS IN THE DATASET

```
n=nearby_venues.drop(['referralId', 'reasons.count', 'reasons.items', 'venue.id',
       'venue.name',
       'venue.location.labeledLatLngs', 'venue.location.distance',
       'venue.location.cc',
       'venue.categories', 'venue.photos.count', 'venue.photos.groups',
       'venue.location.crossStreet', 'venue.location.address','venue.location.city',
       'venue.location.state', 'venue.location.crossStreet',
       'venue.location.neighborhood', 'venue.venuePage.id',
       'venue.location.postalCode','venue.location.country'],axis=1)
```

```
n.columns
```

```
Index(['venue.location.lat', 'venue.location.lng',
       'venue.location.formattedAddress', 'restaurant', 'others'],
      dtype='object')
```

NEW DATASET

```
n
```

	venue.location.lat	venue.location.lng	venue.location.formattedAddress	restaurant	others
0	13.010444	80.220938	[#63 Mount Rd. (Guindy), Chennai 600 032, Tami...	19	14
1	13.010440	80.220669	[#63 Mount Road, Guindy (Mount Rd), Chennai 60...	19	14
2	13.010554	80.223461	[39 Velachery Road Near Raj Bhavan, Chennai 60...	20	17
3	13.010472	80.223536	[39 Vellachery Rd (Sardar Patel Rd), Chennai 6...	20	17
4	12.991041	80.216962	[Phoenix Market City (Velachery), Chennai, Tam...	37	14
...	...	...	...	...	...
95	12.825891	80.246869	[East Coast Road, Covelong, Tamil Nadu, India]	5	3
96	13.045672	80.068746	[Bangalore Trunk Road (Thirumazhisai), Chennai...	6	0
97	12.943957	80.237865	[No 11, Ground Floor, Ramaniyam Isha,Blk 1, Ra...	10	3
98	12.797461	80.248483	[Covelong Beach, (Kancheepuram Dist.), Chennai...	7	3
99	12.895364	80.252325	[14 L Jey Avenue, Akkarai (East Coast Road), C...	5	5

100 rows × 5 columns

DROPPING NAN VALUES FROM DATASET

```
[ ] n=n.dropna()
    n = n.rename(columns={'venue.location.lat': 'lat', 'venue.location.lng': 'long'})
    n
```

	lat	long	venue.location.formattedAddress	restaurant	others
0	13.010444	80.220938	[#63 Mount Rd. (Guindy), Chennai 600 032, Tami...	19	14
1	13.010440	80.220669	[#63 Mount Road, Guindy (Mount Rd), Chennai 60...	19	14
2	13.010554	80.223461	[39 Velachery Road Near Raj Bhavan, Chennai 60...	20	17
3	13.010472	80.223536	[39 Vellachery Rd (Sardar Patel Rd), Chennai 6...	20	17
4	12.991041	80.216962	[Phoenix Market City (Velachery), Chennai, Tam...	37	14
...	...	...	...	...	...
95	12.825891	80.246869	[East Coast Road, Covelong, Tamil Nadu, India]	5	3
96	13.045672	80.068746	[Bangalore Trunk Road (Thirumazhisai), Chennai...	6	0
97	12.943957	80.237865	[No 11, Ground Floor, Ramaniyam Isha,Blk 1, Ra...	10	3
98	12.797461	80.248483	[Covelong Beach, (Kancheepuram Dist.), Chennai...	7	3
99	12.895364	80.252325	[14 L Jey Avenue, Akkarai (East Coast Road), C...	5	5

100 rows × 5 columns

CONVERT EVERY ROW OF COLUMN 'venue.location.formattedAddress' FROM LIST TO STRING

```
[ ] n['venue.location.formattedAddress']
```

```
0     [#63 Mount Rd. (Guindy), Chennai 600 032, Tami...
1     [#63 Mount Road, Guindy (Mount Rd), Chennai 60...
2     [39 Velachery Road Near Raj Bhavan, Chennai 60...
3     [39 Vellachery Rd (Sardar Patel Rd), Chennai 6...
4     [Phoenix Market City (Velachery), Chennai, Tam...
                            ...
95       [East Coast Road, Covelong, Tamil Nadu, India]
96    [Bangalore Trunk Road (Thirumazhisai), Chennai...
97    [No 11, Ground Floor, Ramaniyam Isha,Blk 1, Ra...
98    [Covelong Beach, (Kancheepuram Dist.), Chennai...
99    [14 L Jey Avenue, Akkarai (East Coast Road), C...
Name: venue.location.formattedAddress, Length: 100, dtype: object
```

```
[ ] spec_chars = ["[","]"]
    for char in spec_chars:
      n['venue.location.formattedAddress'] = n['venue.location.formattedAddress'].astype(str).str.replace(char, ' ')
    n
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: FutureWarning: The default value of regex will chan
  This is separate from the ipykernel package so we can avoid doing imports until
```

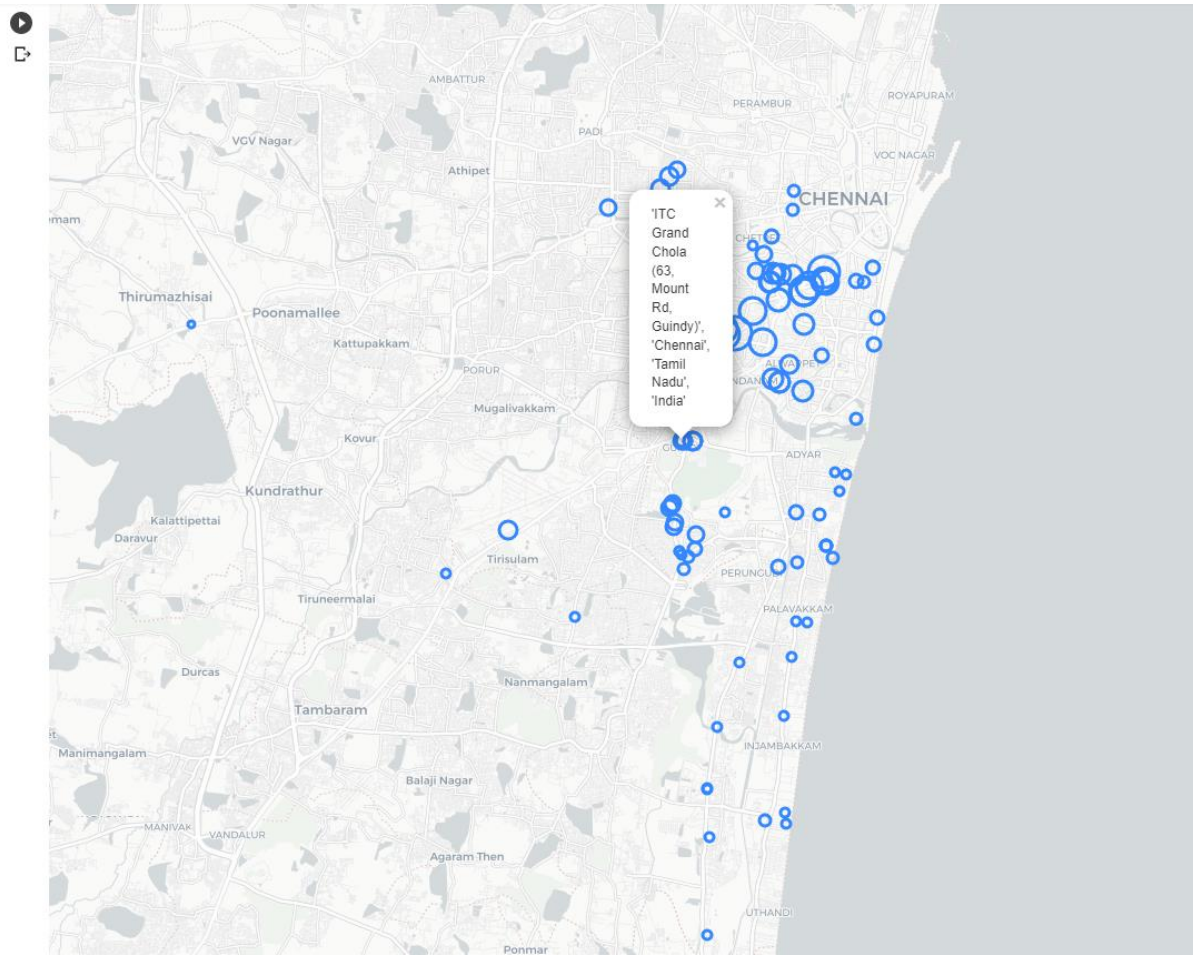
	lat	long	venue.location.formattedAddress	restaurant	others
0	13.010444	80.220938	'#63 Mount Rd. (Guindy)', 'Chennai 600 032', ...	19	14
1	13.010440	80.220669	'#63 Mount Road, Guindy (Mount Rd)', 'Chennai...	19	14
2	13.010554	80.223461	'39 Velachery Road Near Raj Bhavan', 'Chennai...	20	17
3	13.010472	80.223536	'39 Vellachery Rd (Sardar Patel Rd)', 'Chenna...	20	17
4	12.991041	80.216962	'Phoenix Market City (Velachery)', 'Chennai',...	37	14
...	...	...	...	...	...
95	12.825891	80.246869	'East Coast Road', 'Covelong', 'Tamil Nadu', ...	5	3
96	13.045672	80.068746	'Bangalore Trunk Road (Thirumazhisai)', 'Chen...	6	0
97	12.943957	80.237865	'No 11, Ground Floor, Ramaniyam Isha,Blk 1, R...	10	3
98	12.797461	80.248483	'Covelong Beach, (Kancheepuram Dist.)', 'Chen...	7	3
99	12.895364	80.252325	'14 L Jey Avenue, Akkarai (East Coast Road)',...	5	5

100 rows × 5 columns

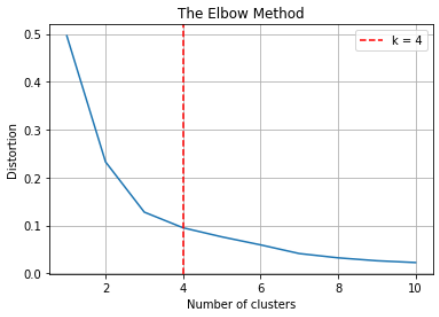
Module 4 – PLOT RESULTS ON MAP

PLOT THE CLUSTERED LOCATIONS ON A MAP

```
[ ]  x, y = "lat", "long"
     color = "restaurant"
     size = "others"
     popup = "venue.location.formattedAddress"
     data = n.copy()

     ## create color column
     lst_colors=["red","green","orange"]
     lst_elements = sorted(list(n[color].unique()))

     ## create size column (scaled)
     scaler = preprocessing.MinMaxScaler(feature_range=(3,15))
     data["size"] = scaler.fit_transform(
                    data[size].values.reshape(-1,1)).reshape(-1)

     ## initialize the map with the starting location
     map_ = folium.Map(location=location, tiles="cartodbpositron",
                       zoom_start=11)
     ## add points
     data.apply(lambda row: folium.CircleMarker(
                location=[row[x],row[y]],popup=row[popup],
                radius=row["size"]).add_to(map_), axis=1)
     ## add html legend


     ## plot the map
     map_
```




```
[ ]  X = n[["lat","long"]]
     max_k = 10
     ## iterations
     distortions = []
     for i in range(1, max_k+1):
         if len(X) >= i:
            model = cluster.KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
            model.fit(X)
            distortions.append(model.inertia_)
     ## best k: the lowest derivative
     k = [i*100 for i in np.diff(distortions,2)].index(min([i*100 for i in np.diff(distortions,2)]))
     ## plot
     fig, ax = plt.subplots()
     ax.plot(range(1, len(distortions)+1), distortions)
     ax.axvline(k, ls='--', color="red", label="k = "+str(k))
     ax.set(title='The Elbow Method', xlabel='Number of clusters',
            ylabel="Distortion")
     ax.legend()
     ax.grid(True)
     plt.show()
```



```
[ ] model = cluster.KMeans(n_clusters=k, init='k-means++')
    X = n[["lat","long"]]
    ## clustering
    dtf_X = X.copy()
    dtf_X["cluster"] = model.fit_predict(X)
    ## find real centroids
    closest, distances = scipy.cluster.vq.vq(model.cluster_centers_,
                         dtf_X.drop("cluster", axis=1).values)
    dtf_X["centroids"] = 0
    for i in closest:
        dtf_X["centroids"].iloc[i] = 1
    ## add clustering info to the original dataset
    n[["cluster","centroids"]] = dtf_X[["cluster","centroids"]]
    n
```

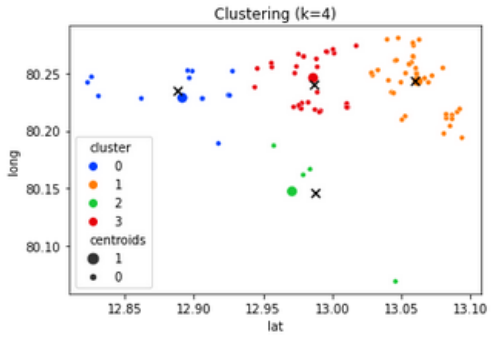
```
/usr/local/lib/python3.7/dist-packages/pandas/core/indexing.py:1732: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
  self._setitem_single_block(indexer, value, name)
```

	lat	long	venue.location.formattedAddress	restaurant	others	cluster	centroids
0	13.010440	80.220669	'#63 Mount Road, Guindy (Mount Rd)', 'Chennai...	17	11	3	0
1	13.010554	80.223461	'39 Velachery Road Near Raj Bhavan', 'Chennai...	19	15	3	0
2	13.010444	80.220938	'#63 Mount Rd. (Guindy)', 'Chennai 600 032', ...	17	11	3	0
3	13.028241	80.250240	'Chamiers Rd', 'India'	48	26	1	0
4	12.991041	80.216962	'Phoenix Market City (Velachery)', 'Chennai',...	40	13	3	0
...	...	...	...	...	...	...	...
95	13.045672	80.068746	'Bangalore Trunk Road (Thirumazhisai)', 'Chen...	6	0	2	0
96	12.830931	80.230223	'Chennai 603103', 'Tamil Nadu', 'India'	2	2	0	0
97	12.943957	80.237865	'No 11, Ground Floor, Ramaniyam Isha,Blk 1, R...	9	3	3	0
98	12.978902	80.161557	'Southern Trunk Road (Opposite Airport (MAA))...	9	16	2	0
99	12.925782	80.230684	'OMR', 'Chennai', 'Tamil Nadu', 'India'	21	1	0	0

100 rows × 7 columns

```
[ ] ## plot
    fig, ax = plt.subplots()
    sns.scatterplot(x="lat", y="long", data=n,
                    palette=sns.color_palette("bright",k),
                    hue='cluster', size="centroids", size_order=[1,0],
                    legend="brief", ax=ax).set_title('Clustering (k='+str(k)+')')
    th_centroids = model.cluster_centers_
    ax.scatter(th_centroids[:,0], th_centroids[:,1], s=50, c='black',
               marker="x")
```

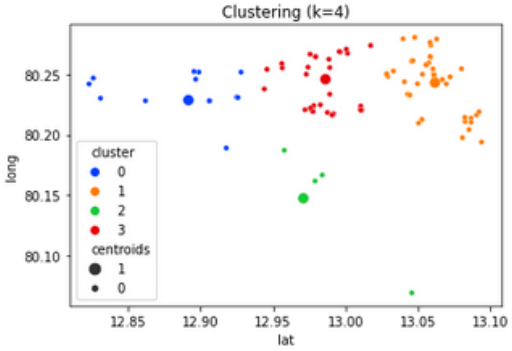
```
<matplotlib.collections.PathCollection at 0x7f6fc99e89d0>
```



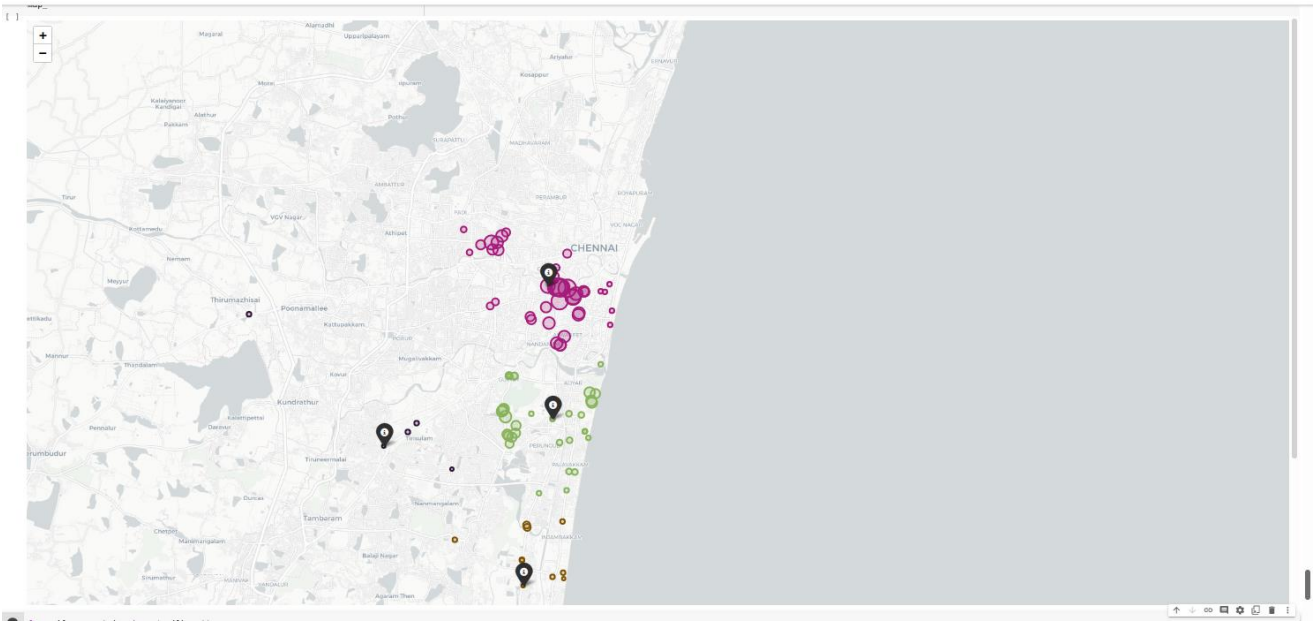
```
[ ]  model = cluster.AffinityPropagation()
```

```
[ ]  k = n["cluster"].nunique()
     sns.scatterplot(x="lat", y="long", data=n,
                     palette=sns.color_palette("bright",k),
                     hue='cluster', size="centroids", size_order=[1,0],
                     legend="brief").set_title('Clustering (k='+str(k)+')')
```

```
Text(0.5, 1.0, 'Clustering (k=4)')
```




```
x, y = "lat", "long"
color = "cluster"
size = "restaurant"
popup = "venue.location.formattedAddress"
marker = "centroids"
data = n.copy()
## create color column
lst_elements = sorted(list(n[color].unique()))
lst_colors = ['#%06X' % np.random.randint(0, 0xFFFFFF) for i in
              range(len(lst_elements))]
data["color"] = data[color].apply(lambda x:
                lst_colors[lst_elements.index(x)])
## create size column (scaled)
scaler = preprocessing.MinMaxScaler(feature_range=(3,15))
data["size"] = scaler.fit_transform(
               data[size].values.reshape(-1,1)).reshape(-1)
## initialize the map with the starting location
map_ = folium.Map(location=location, tiles="cartodbpositron",
                  zoom_start=11)
## add points
data.apply(lambda row: folium.CircleMarker(
           location=[row[x],row[y]],
           color=row["color"], fill=True,popup=row[popup],
           radius=row["size"]).add_to(map_), axis=1)
## add html legend
legend_html = """<div style="position:fixed; bottom:10px; left:10px; border:2px solid black; z-index:9999; font-size:14px;"> <b>"""+color+""":</b><br>"""
for i in lst_elements:
     legend_html = legend_html+""" <i class="fa fa-circle
     fa-1x" style="color:"""+lst_colors[lst_elements.index(i)]+"""">
     </i> """+str(i)+"""<br>"""
legend_html = legend_html+"""</div>"""
map_.get_root().html.add_child(folium.Element(legend_html))
## add centroids marker
lst_elements = sorted(list(n[marker].unique()))
data[data[marker]==1].apply(lambda row:
           folium.Marker(location=[row[x],row[y]],
           draggable=False,  popup=row[popup] ,
           icon=folium.Icon(color="black")).add_to(map_), axis=1)
## plot the map
map_
```




```
from sklearn.metrics import silhouette_score
g = X.copy()
print(g)
model.fit(g)
score = silhouette_score(g, model.labels_, metric='euclidean')
print('\nSilhouetter Score: %.3f' % score)
```

```
         lat       long
0   13.010440  80.220669
1   13.010554  80.223461
2   13.010444  80.220938
3   13.028241  80.250240
4   12.991041  80.216962
..        ...        ...
95  13.045672  80.068746
96  12.830931  80.230223
97  12.943957  80.237865
98  12.978902  80.161557
99  12.925782  80.230684

[100 rows x 2 columns]

Silhouetter Score: 0.544
```

Performance measures used

Silhouette Coefficient: ranges from -1 to +1

1: means clusters are well apart from each other and clearly distinguished

0: means clusters are indifferent

-1: means clusters are assigned in a wrong way

Silhouette score = (b-a)/max(a,b)

a- Average intra-cluster distance
b- Average inter-cluster distance

References

[1] Pereira-Martinez.D, Lopez Choa, V.Lizancos P, And Borges Pereira V.(2022). A geolocational collection strategy to asses housing in its social, environmental, and spatial aspects, Archi DOCT,17.29(9(2)).

[2] Sprido Spyrates, Demetris,Stathakis, Michael Lutz, Chizsa Tsimaraki. Using Foursquare place data for estimating build block use, March 2016.

[3] Joel Riberio, Tania Fontes, Carlos Soares, Joseluis Borges. Process Discovery on geolocational data, September 2019.