In [ ]:

The Battle of Neighborhood 2

Introduction:

This project is to take a glimpse at my hometown, Kavala, Greece. This project may be interesting for property developers to develop new shops in my hometown.

Such activity will increase the number of yearly visitors and of course the GDP of Kavala.

Data:

There will be used two datasets with PostalCode and Neighborhoods based on their coordinates, latitude and longtude, of Kavala city from online sources.

Some of the data have to be collected manually as some of the neightbohoods are not easily identified. As an ex-"local" resident, I am able to identify them via Google maps.

Methodology:

Skills from all prevous weeks lab activities plus some Python programming skills and knowledge of IBM cloud to help the project.

The notebook will be created in IBM Watson Studio.

GitHub will be used to share the notebook and commit it to the Master branch.

We will be relying on the Foursquare API to retrieve all venues of each neighborhoods, then group by each neighborhoods and count how many venues exist. Then filter them on the top 100 that are within a radius of 2000 meters.

In [3]:
```
# Import the required libraries
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

!conda install -c conda-forge geopy --yes # uncomment this line if you haven't
completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and l
ongitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas d
ataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

# for webscraping import Beautiful Soup
from bs4 import BeautifulSoup

import xml

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you
 haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

```
Solving environment: done

## Package Plan ##

  environment location: /opt/conda/envs/Python36

  added / updated specs:
    - geopy


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    geographiclib-1.50         |             py_0          34 KB  conda-forge
    geopy-1.20.0               |             py_0          57 KB  conda-forge
    certifi-2019.11.28         |           py36_0         149 KB  conda-forge
    openssl-1.1.1d             |       h516909a_0         2.1 MB  conda-forge
    ca-certificates-2019.11.28 |       hecc5488_0         145 KB  conda-forge
    ------------------------------------------------------------
                                           Total:         2.5 MB

The following NEW packages will be INSTALLED:

    geographiclib:   1.50-py_0         conda-forge
    geopy:           1.20.0-py_0       conda-forge

The following packages will be UPDATED:

    ca-certificates: 2019.11.27-0                 --> 2019.11.28-hecc5488_0
conda-forge
    certifi:         2019.11.28-py36_0            --> 2019.11.28-py36_0
conda-forge

The following packages will be DOWNGRADED:

    openssl:         1.1.1d-h7b6447c_3            --> 1.1.1d-h516909a_0
conda-forge


Downloading and Extracting Packages
geographiclib-1.50   | 34 KB     | ##################################### | 10
0%
geopy-1.20.0         | 57 KB     | ##################################### | 10
0%
certifi-2019.11.28   | 149 KB    | ##################################### | 10
0%
openssl-1.1.1d       | 2.1 MB    | ##################################### | 10
0%
ca-certificates-2019 | 145 KB    | ##################################### | 10
0%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Solving environment: done

## Package Plan ##
```

```
  environment location: /opt/conda/envs/Python36

  added / updated specs:
    - folium=0.5.0


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    vincent-0.4.4              |             py_1          28 KB  conda-forge
    folium-0.5.0               |             py_0          45 KB  conda-forge
    altair-4.0.1               |             py_0         575 KB  conda-forge
    branca-0.3.1               |             py_0          25 KB  conda-forge
    ------------------------------------------------------------
                                           Total:         673 KB

The following NEW packages will be INSTALLED:

    altair:  4.0.1-py_0 conda-forge
    branca:  0.3.1-py_0 conda-forge
    folium:  0.5.0-py_0 conda-forge
    vincent: 0.4.4-py_1 conda-forge


Downloading and Extracting Packages
vincent-0.4.4        | 28 KB     | ##################################### | 10
0%
folium-0.5.0         | 45 KB     | ##################################### | 10
0%
altair-4.0.1         | 575 KB    | ##################################### | 10
0%
branca-0.3.1         | 25 KB     | ##################################### | 10
0%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Libraries imported.
```

In [4]:
```
address = 'Kavala, Greece'

geolocator = Nominatim()
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of the City of Kavala are {}, {}.'.format(la
titude, longitude))
```

```
/opt/conda/envs/Python36/lib/python3.6/site-packages/ipykernel/__main__.py:3:
DeprecationWarning: Using Nominatim with the default "geopy/1.20.0" `user_age
nt` is strongly discouraged, as it violates Nominatim's ToS https://operation
s.osmfoundation.org/policies/nominatim/ and may possibly cause 403 and 429 HT
TP errors. Please specify a custom `user_agent` with `Nominatim(user_agent="m
y-application")` or by overriding the default `user_agent`: `geopy.geocoders.
options.default_user_agent = "my-application"`. In geopy 2.0 this will become
an exception.
  app.launch_new_instance()

The geograpical coordinate of the City of Kavala are 40.9369224, 24.4122766.
```

In [5]:
```
# create map of Kavala using latitude and longitude values
map_kavala = folium.Map(location=[latitude, longitude], zoom_start=10)
map_kavala
```

Out[5]:

Upload Table with postal code, address and region

In [23]:
```
# create a list to store neighborhood data
neighborhoodList = ['Agia Varvara','Agios Athanasios','Agios Ioannis','Agios L
oukas','Dexameni','Kalamitsa', 'Kentro (Centre)','Panagia','Perigiali','Potamo
udia','Profitis Ilias',
                    'Timios Stavros','Vyronas']
```

In [24]:
```
# create a new DataFrame from the list
kl_df = pd.DataFrame({"Neighborhood": neighborhoodList})

kl_df.head()
```

Out[24]:

	Neighborhood
0	Agia Varvara
1	Agios Athanasios
2	Agios Ioannis
3	Agios Loukas
4	Dexameni

In [25]:
```
# print the number of rows of the dataframe
kl_df.shape
```

Out[25]: (13, 1)

Get the geographical coordinates

In [36]:
```
# create a list to store coordinates as not all the areas in my hometown can h
ave correct coordinates

latitude_kavala = [ 40.9376048,40.8170169,40.9371466,40.9333244,40.9397263,40.
922226, 40.9359791,40.934301,40.9368829,40.9417546,40.9418333,40.9402566,40.93
5112 ]
longtitude_kavala = [ 24.4191961, 24.2955306, 24.3998248, 24.3817282, 24.39347
75, 24.383695, 24.4085043, 24.414428, 24.4057799, 24.4022739, 24.4047204, 24.4
158105, 24.3919673 ]
```

In [42]:
```
# add the coordinates into the kl_df dataframe to populate the coordinates int
o Latitude and Longitude
kl_df = pd.DataFrame({"Neighborhood": neighborhoodList,"Latitude": latitude_ka
vala , "Longitude": longtitude_kavala })
kl_df.head()
```

Out[42]:

	Neighborhood	Latitude	Longitude
0	Agia Varvara	40.937605	24.419196
1	Agios Athanasios	40.817017	24.295531
2	Agios Ioannis	40.937147	24.399825
3	Agios Loukas	40.933324	24.381728
4	Dexameni	40.939726	24.393477

In [43]:
```
# save the DataFrame as CSV file
kl_df.to_csv("kl_df.csv", index=False)
```

Create a map of Kavala and include the above neighbohoods

In [44]:
```
# get the coordinates of Kuala Lumpur
address = 'Kavala, Greece'

geolocator = Nominatim(user_agent="my-application")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Kavala, Greece {}, {}.'.format(latitude,
longitude))
```

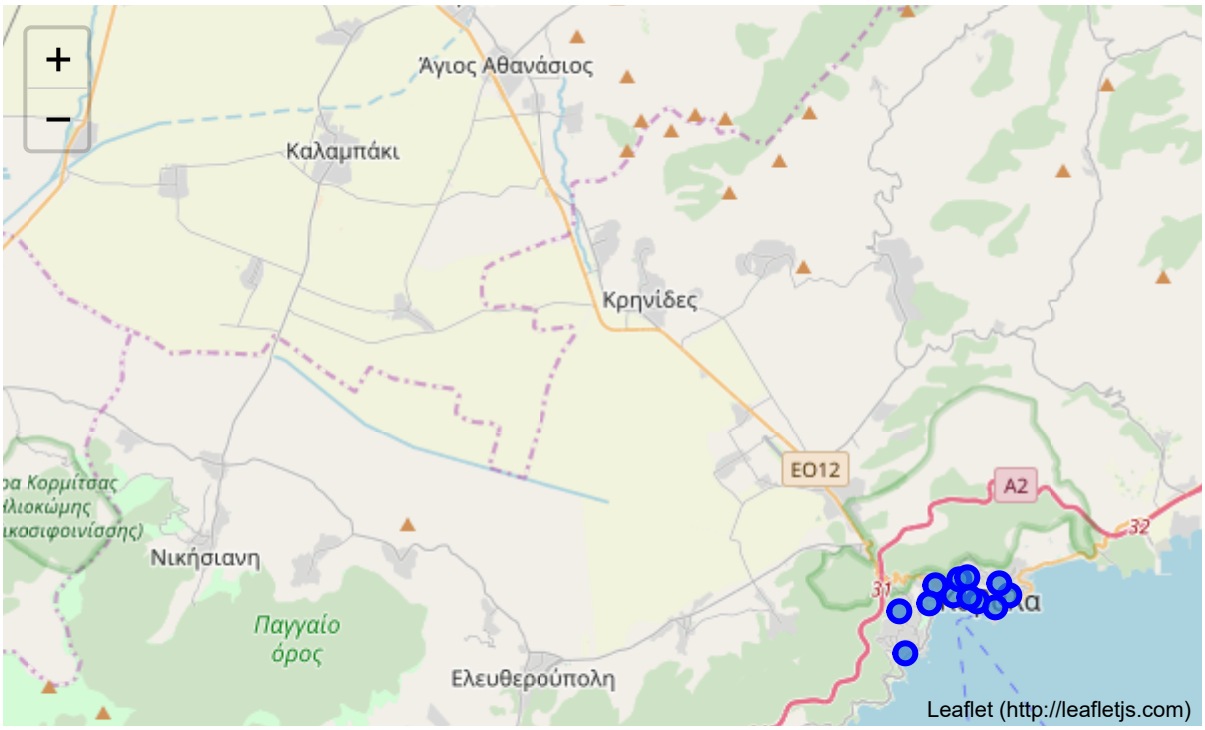
```
The geograpical coordinate of Kavala, Greece 40.9369224, 24.4122766.
```

In [45]:
```
# create map of Kavala using latitude and longitude values
map_kl = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, neighborhood in zip(kl_df['Latitude'], kl_df['Longitude'], kl_df
['Neighborhood']):
    label = '{}'.format(neighborhood)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7).add_to(map_kl)

map_kl
```

Out[45]:



In [46]:
```
# save the map as HTML file
map_kl.save('map_kl.html')
```

Use the Foursquare API to explore the neighborhoods

In [47]:
```
# define Foursquare Credentials and Version
CLIENT_ID = 'LCND3VF5E5UGOHCUDOMRQV5FAI5V3ELTXRSUEXYK312CQH14' # your Foursqua
re ID
CLIENT_SECRET = '3MVWMRVPRRFDPJ4PZ5KOI325IAHM4F0R00HYYSGKWDU1XBER' # your Four
square Secret
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: LCND3VF5E5UGOHCUDOMRQV5FAI5V3ELTXRSUEXYK312CQH14
CLIENT_SECRET:3MVWMRVPRRFDPJ4PZ5KOI325IAHM4F0R00HYYSGKWDU1XBER
```

Now, let's get the top 100 venues that are within a radius of 2000 meters.

In [48]:
```
radius = 2000
LIMIT = 100

venues = []

for lat, long, neighborhood in zip(kl_df['Latitude'], kl_df['Longitude'], kl_d
f['Neighborhood']):

    # create the API request URL
    url = "https://api.foursquare.com/v2/venues/explore?client_id={}&client_se
cret={}&v={}&ll={},{}&radius={}&limit={}".format(
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        lat,
        long,
        radius,
        LIMIT)

    # make the GET request
    results = requests.get(url).json()["response"]['groups'][0]['items']

    # return only relevant information for each nearby venue
    for venue in results:
        venues.append((
            neighborhood,
            lat,
            long,
            venue['venue']['name'],
            venue['venue']['location']['lat'],
            venue['venue']['location']['lng'],
            venue['venue']['categories'][0]['name']))
```

In [49]:
```
# convert the venues list into a new DataFrame
venues_df = pd.DataFrame(venues)

# define the column names
venues_df.columns = ['Neighborhood', 'Latitude', 'Longitude', 'VenueName', 'Ve
nueLatitude', 'VenueLongitude', 'VenueCategory']

print(venues_df.shape)
venues_df.head()
```

```
(1077, 7)
```

Out[49]:

	Neighborhood	Latitude	Longitude	VenueName	VenueLatitude	VenueLongitude	VenueCateg
0	Agia Varvara	40.937605	24.419196	Dèrelicte	40.936769	24.413643	Cocktail
1	Agia Varvara	40.937605	24.419196	Kavala Castle (Κάστρο Καβάλας)	40.934365	24.415439	Ca
2	Agia Varvara	40.937605	24.419196	The Anthemion House	40.937637	24.416433	H
3	Agia Varvara	40.937605	24.419196	Espresso & 'tails	40.936802	24.413589	Cocktail
4	Agia Varvara	40.937605	24.419196	Μπουγάτσα το Ανώτερο	40.940580	24.418637	Bougatsa S

Let's check how many venues were returned for each neighorhood

In [50]:
```
venues_df.groupby(["Neighborhood"]).count()
```

Out[50]:

Neighborhood	Latitude	Longitude	VenueName	VenueLatitude	VenueLongitude	VenueCategory
Agia Varvara	99	99	99	99	99	99
Agios Athanasios	16	16	16	16	16	16
Agios Ioannis	100	100	100	100	100	100
Agios Loukas	47	47	47	47	47	47
Dexameni	100	100	100	100	100	100
Kalamitsa	40	40	40	40	40	40
Kentro (Centre)	94	94	94	94	94	94
Panagia	93	93	93	93	93	93
Perigiali	96	96	96	96	96	96
Potamoudia	97	97	97	97	97	97
Profitis Ilias	95	95	95	95	95	95
Timios Stavros	100	100	100	100	100	100
Vyronas	100	100	100	100	100	100

Let's find out how many unique categories can be curated from all the returned venues

In [52]:
```
print('There are {} uniques categories.'.format(len(venues_df['VenueCategory']
.unique())))
```

```
There are 62 uniques categories.
```

In [53]:
```
# print out the list of categories
venues_df['VenueCategory'].unique()[:50]
```

Out[53]:
```
array(['Cocktail Bar', 'Castle', 'Hotel', 'Bougatsa Shop',
       'Tsipouro Restaurant', 'Greek Restaurant', 'Café', 'Bar',
       'Taverna', 'Church', 'Dessert Shop', 'Fish Taverna',
       'Grilled Meat Restaurant', 'Comfort Food Restaurant',
       'Coffee Shop', 'Ouzeri', 'History Museum', 'Meze Restaurant',
       'Historic Site', 'Eastern European Restaurant', 'Gift Shop',
       'Sports Bar', 'Bakery', 'Cosmetics Shop', 'Plaza', 'Lounge',
       'Clothing Store', 'Bistro', 'Garden', 'Waterfront', 'Gym', 'Park',
       'Pool', 'Supermarket', 'Fast Food Restaurant', 'Movie Theater',
       'Soccer Field', 'Souvlaki Shop', 'Burger Joint', 'Nightclub',
       'Beach', 'Dance Studio', 'Snack Place', 'Betting Shop',
       'Basketball Stadium', 'Theater', 'Convenience Store',
       'Soccer Stadium', 'Beach Bar', 'Restaurant'], dtype=object)
```

In [54]:
```
# check if the results contain "Shopping Mall"
"Neighborhood" in venues_df['VenueCategory'].unique()
```

Out[54]:
```
False
```

Analyze Each Neighborhood

In [56]:
```
# one hot encoding
kl_onehot = pd.get_dummies(venues_df[['VenueCategory']], prefix="", prefix_sep
="")

# add neighborhood column back to dataframe
kl_onehot['Neighborhoods'] = venues_df['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [kl_onehot.columns[-1]] + list(kl_onehot.columns[:-1])
kl_onehot = kl_onehot[fixed_columns]

print(kl_onehot.shape)
kl_onehot.head()
```

```
(1077, 63)
```

Out[56]:

	Neighborhoods	Bakery	Bar	Basketball Stadium	Beach	Beach Bar	Beer Bar	Betting Shop	Bistro	Bougatsa Shop	Bu
0	Agia Varvara	0	0	0	0	0	0	0	0	0	
1	Agia Varvara	0	0	0	0	0	0	0	0	0	
2	Agia Varvara	0	0	0	0	0	0	0	0	0	
3	Agia Varvara	0	0	0	0	0	0	0	0	0	
4	Agia Varvara	0	0	0	0	0	0	0	0	1	

Next, let's group rows by neighborhood and by taking the mean of the frequency of occurrence of each category

In [57]:
```
kl_grouped = kl_onehot.groupby(["Neighborhoods"]).mean().reset_index()

print(kl_grouped.shape)
kl_grouped
```

```
(13, 63)
```

Out[57]:

	Neighborhoods	Bakery	Bar	Basketball Stadium	Beach	Beach Bar	Beer Bar	Betting Shop	Bis
0	Agia Varvara	0.010101	0.050505	0.010101	0.030303	0.000	0.000000	0.010101	0.0101
1	Agios Athanasios	0.000000	0.000000	0.000000	0.125000	0.625	0.000000	0.000000	0.0000
2	Agios Ioannis	0.010000	0.040000	0.010000	0.020000	0.000	0.010000	0.000000	0.0100
3	Agios Loukas	0.000000	0.000000	0.021277	0.042553	0.000	0.021277	0.000000	0.0000
4	Dexameni	0.010000	0.040000	0.010000	0.020000	0.000	0.010000	0.000000	0.0100
5	Kalamitsa	0.050000	0.025000	0.025000	0.100000	0.050	0.025000	0.000000	0.0000
6	Kentro (Centre)	0.010638	0.042553	0.000000	0.010638	0.000	0.010638	0.000000	0.0106
7	Panagia	0.010753	0.043011	0.000000	0.032258	0.000	0.000000	0.000000	0.0107
8	Perigiali	0.010417	0.041667	0.000000	0.010417	0.000	0.010417	0.000000	0.0104
9	Potamoudia	0.010309	0.041237	0.000000	0.010309	0.000	0.010309	0.000000	0.0103
10	Profitis Ilias	0.010526	0.042105	0.000000	0.010526	0.000	0.010526	0.000000	0.0105
11	Timios Stavros	0.010000	0.060000	0.010000	0.030000	0.000	0.000000	0.010000	0.0100
12	Vyronas	0.010000	0.030000	0.010000	0.020000	0.000	0.010000	0.000000	0.0100

In [59]:
```
len(kl_grouped[kl_grouped["Bougatsa Shop"] > 0])
```

Out[59]:
```
10
```

Create a new DataFrame for Bougatsa Shop data only

In [61]:
```
kl_mall = kl_grouped[["Neighborhoods","Bougatsa Shop"]]

kl_mall.head()
```

Out[61]:

	Neighborhoods	Bougatsa Shop
0	Agia Varvara	0.030303
1	Agios Athanasios	0.000000
2	Agios Ioannis	0.030000
3	Agios Loukas	0.000000
4	Dexameni	0.020000

Cluster Neighborhoods

Run k-means to cluster the neighborhoods in Kavala into 3 clusters.

In [62]:
```
# set number of clusters
kclusters = 3

kl_clustering = kl_mall.drop(["Neighborhoods"], 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(kl_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

Out[62]:
```
array([2, 1, 2, 1, 0, 1, 2, 2, 2, 2], dtype=int32)
```

In [63]:
```
# create a new dataframe that includes the cluster as well as the top 10 venue
s for each neighborhood.
kl_merged = kl_mall.copy()

# add clustering labels
kl_merged["Cluster Labels"] = kmeans.labels_
```

In [64]:
```
kl_merged.rename(columns={"Neighborhoods": "Neighborhood"}, inplace=True)
kl_merged.head()
```

Out[64]:

	Neighborhood	Bougatsa Shop	Cluster Labels
0	Agia Varvara	0.030303	2
1	Agios Athanasios	0.000000	1
2	Agios Ioannis	0.030000	2
3	Agios Loukas	0.000000	1
4	Dexameni	0.020000	0

In [65]:
```
# merge kl_merged with kl_df to add latitude/longitude for each neighborhood
kl_merged = kl_merged.join(kl_df.set_index("Neighborhood"), on="Neighborhood")

print(kl_merged.shape)
kl_merged.head() # check the last columns!
```

```
(13, 5)
```

Out[65]:

	Neighborhood	Bougatsa Shop	Cluster Labels	Latitude	Longitude
0	Agia Varvara	0.030303	2	40.937605	24.419196
1	Agios Athanasios	0.000000	1	40.817017	24.295531
2	Agios Ioannis	0.030000	2	40.937147	24.399825
3	Agios Loukas	0.000000	1	40.933324	24.381728
4	Dexameni	0.020000	0	40.939726	24.393477

In [66]:
```
# sort the results by Cluster Labels
print(kl_merged.shape)
kl_merged.sort_values(["Cluster Labels"], inplace=True)
kl_merged
```

```
(13, 5)
```

Out[66]:

	Neighborhood	Bougatsa Shop	Cluster Labels	Latitude	Longitude
4	Dexameni	0.020000	0	40.939726	24.393477
12	Vyronas	0.020000	0	40.935112	24.391967
1	Agios Athanasios	0.000000	1	40.817017	24.295531
3	Agios Loukas	0.000000	1	40.933324	24.381728
5	Kalamitsa	0.000000	1	40.922226	24.383695
0	Agia Varvara	0.030303	2	40.937605	24.419196
2	Agios Ioannis	0.030000	2	40.937147	24.399825
6	Kentro (Centre)	0.031915	2	40.935979	24.408504
7	Panagia	0.032258	2	40.934301	24.414428
8	Perigiali	0.031250	2	40.936883	24.405780
9	Potamoudia	0.030928	2	40.941755	24.402274
10	Profitis Ilias	0.031579	2	40.941833	24.404720
11	Timios Stavros	0.030000	2	40.940257	24.415810

Finally, let's visualize the resulting clusters

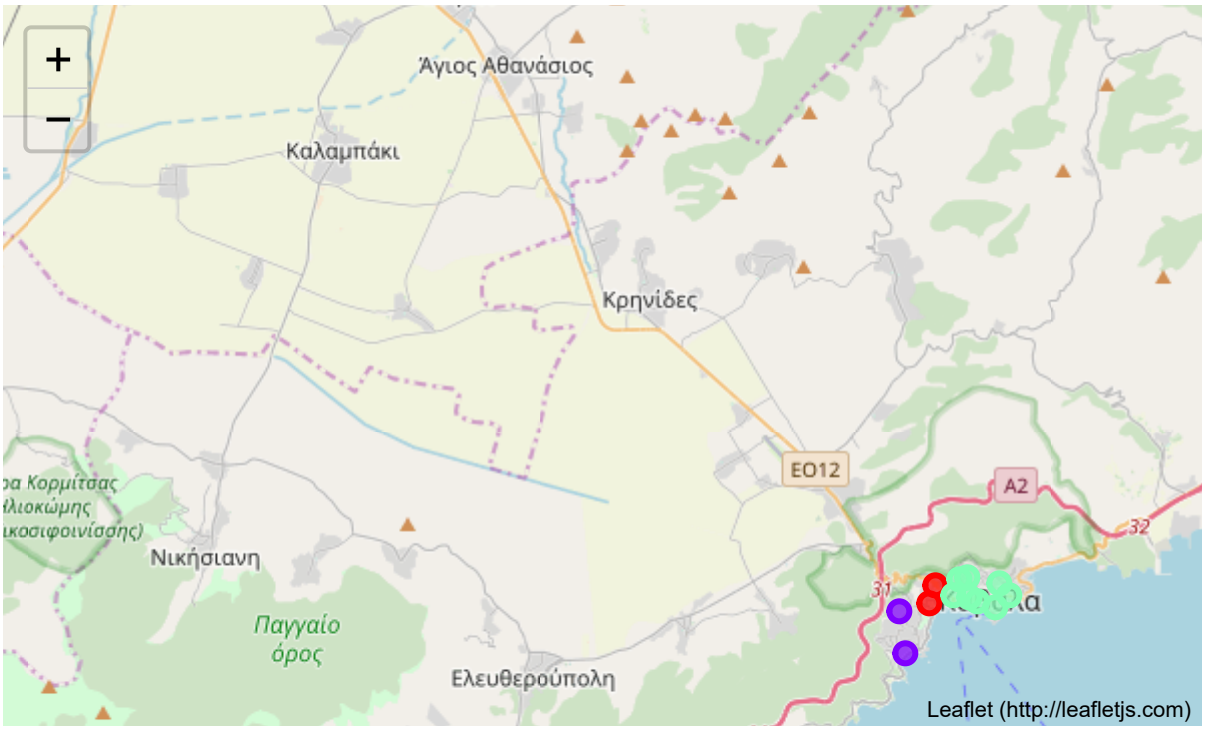
In [67]:
```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i+x+(i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(kl_merged['Latitude'], kl_merged['Longitude'
], kl_merged['Neighborhood'], kl_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' - Cluster ' + str(cluster), parse_html=T
rue)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

Out[67]:



In [68]:
```
# save the map as HTML file
map_clusters.save('map_clusters.html')
```

Examine Clusters

Cluster 0

In [69]:
```
kl_merged.loc[kl_merged['Cluster Labels'] == 0]
```

Out[69]:

	Neighborhood	Bougatsa Shop	Cluster Labels	Latitude	Longitude
4	Dexameni	0.02	0	40.939726	24.393477
12	Vyronas	0.02	0	40.935112	24.391967

Cluster 1

In [70]:
```
kl_merged.loc[kl_merged['Cluster Labels'] == 1]
```

Out[70]:

	Neighborhood	Bougatsa Shop	Cluster Labels	Latitude	Longitude
1	Agios Athanasios	0.0	1	40.817017	24.295531
3	Agios Loukas	0.0	1	40.933324	24.381728
5	Kalamitsa	0.0	1	40.922226	24.383695

Cluster 2

In [71]:
```
kl_merged.loc[kl_merged['Cluster Labels'] == 2]
```

Out[71]:

	Neighborhood	Bougatsa Shop	Cluster Labels	Latitude	Longitude
0	Agia Varvara	0.030303	2	40.937605	24.419196
2	Agios Ioannis	0.030000	2	40.937147	24.399825
6	Kentro (Centre)	0.031915	2	40.935979	24.408504
7	Panagia	0.032258	2	40.934301	24.414428
8	Perigiali	0.031250	2	40.936883	24.405780
9	Potamoudia	0.030928	2	40.941755	24.402274
10	Profitis Ilias	0.031579	2	40.941833	24.404720
11	Timios Stavros	0.030000	2	40.940257	24.415810

Observations:

Most of the Bougatsa shops are concentrated in the central area of Kavala city, with the highest number in cluster 2 and moderate number in cluster 0. On the other hand, cluster 1 has very low number to totally no bougatsa shops in the neighborhoods. This represents a great opportunity and high potential areas to open bougatsa shops as there is very little to none competition from existing shops. Meanwhile, bougatsa shops in cluster 2 are likely suffering from intense competition due to oversupply and high concentration of bougatsa shops. On the contrary, this shows that an oversupply of bougatsa shops mostly happened in the central area of the city, with the suburb areas still have very few bougatsa shops.

Therefore, this project recommends property developers to capitalize on these findings to open new bougatsa shops in neighborhoods in cluster 1 with little to none competition. Property developers with unique selling propositions can also open new bougatsa shops in neighborhoods in cluster 0 with moderate competition. Lastly, property developers are advised to avoid neighborhoods in cluster 2 which have high concentration of bougatsa shops and already suffering from intense competition.

In [ ]: