

Untitled

November 30, 2020

0.1 1. Introduction: COFFEE PLACE IN THESSALONIKI

In such a large and rich of coffee culture city like Thessaloniki, Greece, it will be competitive to start up coffee business. In this case my contractor is a humble local man who has contacted me to give advises and draw up essential lines of business prediction and back-up plans (but this part we will just discuss about predicting hot spot).

0.2 2. Orientation

First of all we need to collect Data of all coffee shops in Thessaloniki including their name, id, location (address, latitude, longitude) then pick the “hot” neighborhood where locates most of the venues. In order to asset Data we use FourSquare and apply folium for visualizing a particular neighbor in which that we will observe customer “traffic” and predict an appropriate location of new coffee shop in town. In this case you will find its temporary name on the folium map, “Oriste!”(Meaning “Here It Is!” in Greek)

0.3 3. Execution steps

We import all the tools we need.

```
[1]: import requests # library to handle requests
import pandas as pd # library for data analysis
import numpy as np
!pip install folium
import folium
```

```
Requirement already satisfied: folium in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (0.5.0)
Requirement already satisfied: branca in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from folium)
(0.4.1)
Requirement already satisfied: jinja2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from folium)
(2.11.2)
Requirement already satisfied: requests in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from folium)
(2.24.0)
Requirement already satisfied: six in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from folium)
(1.15.0)
```

Requirement already satisfied: MarkupSafe>=0.23 in
 /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
 jinja2->folium) (1.1.1)
 Requirement already satisfied: idna<3,>=2.5 in
 /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
 requests->folium) (2.10)
 Requirement already satisfied: certifi>=2017.4.17 in
 /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
 requests->folium) (2020.6.20)
 Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in
 /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
 requests->folium) (1.25.11)
 Requirement already satisfied: chardet<4,>=3.0.2 in
 /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
 requests->folium) (3.0.4)

Apply your credential ID n Foursquare

```
[6]: CLIENT_ID = '4XV4MUEEX00QGF55PBPI4VCIUN3YTWEHKKHQGL0T11QXGE2NA'
CLIENT_SECRET = 'APIDDD1AYMIJ4CVDNVYNJSZRQWZ4VEAHW2WEBDQ4VX31NQ4Q'
VERSION = 20180604
LIMIT = 40
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET: '+ CLIENT_SECRET)
```

Your credentails:

CLIENT_ID: 4XV4MUEEX00QGF55PBPI4VCIUN3YTWEHKKHQGL0T11QXGE2NA

CLIENT_SECRET: APIDDD1AYMIJ4CVDNVYNJSZRQWZ4VEAHW2WEBDQ4VX31NQ4Q

Get requests near Thessaloniki city

```
[12]: import requests

request_parameters = {
    "client_id": CLIENT_ID,
    "client_secret": CLIENT_SECRET,
    "v": '20180605',
    "section": "coffee",
    "near": "Thessaloniki",
    "radius": 1000,
    "limit": 50}

data = requests.get("https://api.foursquare.com/v2/venues/explore",
    ↪params=request_parameters)
```

Transform data into json then request geocode

```
[18]: d = data.json()["response"]
      d.keys()
```

```
[18]: dict_keys(['suggestedFilters', 'geocode', 'headerLocation',
               'headerFullLocation', 'headerLocationGranularity', 'query', 'totalResults',
               'suggestedBounds', 'groups'])
```

```
[20]: d["headerLocationGranularity"], d["headerLocation"], d["headerFullLocation"]
```

```
[20]: ('city', 'Thessaloníki', 'Thessaloníki')
```

```
[22]: d["suggestedBounds"], d["totalResults"]
```

```
[22]: ({'ne': {'lat': 40.65073224977761, 'lng': 22.941163388173987},
       'sw': {'lat': 40.635032627617214, 'lng': 22.918850791039336}},
      39)
```

```
[23]: d["geocode"]
```

```
[23]: {'what': '',
      'where': 'thessaloniki',
      'center': {'lat': 40.64361, 'lng': 22.93086},
      'displayString': 'Thessaloníki, Central Macedonia, Greece',
      'cc': 'GR',
      'geometry': {'bounds': {'ne': {'lat': 40.71540595456036,
                                     'lng': 23.023440674362327},
                     'sw': {'lat': 40.52794000897531, 'lng': 22.848796876532834}}},
      'slug': 'thessaloniki-greece',
      'longId': '72057594038662013'}
```

We start creating group including information which is recommended.

```
[25]: d["groups"][0].keys()
```

```
[25]: dict_keys(['type', 'name', 'items'])
```

```
[26]: d["groups"][0]["type"], d["groups"][0]["name"]
```

```
[26]: ('Recommended Places', 'recommended')
```

Creating items of objects coffee shop and their attributes - id, address, name, etc

```
[28]: items = d["groups"][0]["items"]
      print("number of items: %i" % len(items))
      items[0]
```

number of items: 39

```
[28]: {'reasons': {'count': 0,
  'items': [{'summary': 'This spot is popular',
    'type': 'general',
    'reasonName': 'globalInteractionReason'}]},
'venue': {'id': '4f8cf843e4b04bd7c54fe648',
'name': 'Εστιατόριο',
'location': {'address': 'Π.Σ. Σ',
'lat': 40.640941349567335,
'lng': 22.928243694085044,
'labeledLatLngs': [{'label': 'display',
'lat': 40.640941349567335,
'lng': 22.928243694085044}]},
'postalCode': '546 27',
'cc': 'GR',
'city': 'Θεσσαλονίκη',
'state': 'Θεσσαλονίκη',
'country': 'Ελλάδα',
'formattedAddress': ['Π.Σ. Σ',
'546 27 Θεσσαλονίκη',
'Ελλάδα']},
'categories': [{'id': '53d6c1b0e4b02351e88a83e8',
'name': 'Bougatsa Shop',
'pluralName': 'Bougatsa Shops',
'shortName': 'Bougatsa Shops',
'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/snacks_',
'suffix': '.png'},
'primary': True}],
'photos': {'count': 0, 'groups': []}},
'referralId': 'e-5-4f8cf843e4b04bd7c54fe648-0'}
```

```
[30]: df_raw = []
for item in items:
    venue = item["venue"]
    categories, uid, name, location = venue["categories"], venue["id"], venue["name"], venue["location"]
    print(location)
    assert len(categories) == 1
    shortname = categories[0]["shortName"]
    address = ''
    if hasattr(location, 'address'):
        address = location['address']
    if not "postalCode" in location:
        continue
    postalcode = location["postalCode"]
    lat = location["lat"]
    lng = location["lng"]
    datarow = (uid, name, shortname, address, postalcode, lat, lng)
```

```

df_raw.append(datarow)
df = pd.DataFrame(df_raw, columns=["uid", "name", "shortname", "address", "postalcode", "lat", "lng"])
print("found %i cafes" % len(df))
df.head()

```

```

{'address': 'Π.Σ. Σ ', 'lat': 40.640941349567335, 'lng': 22.928243694085044, 'labeledLatLngs': [{'label': 'display', 'lat': 40.640941349567335, 'lng': 22.928243694085044}], 'postalCode': '546 27', 'cc': 'GR', 'city': 'Θ ', 'state': 'Θ ', 'country': 'E ', 'formattedAddress': ['Π.Σ. Σ ', '546 27 Θ ', 'Θ ', 'E ']}
{'lat': 40.64036571242544, 'lng': 22.933995822487688, 'labeledLatLngs': [{'label': 'display', 'lat': 40.64036571242544, 'lng': 22.933995822487688}], 'cc': 'GR', 'country': 'E ', 'formattedAddress': ['E ']}
{'address': 'Λ ', 'lat': 40.642369604339585, 'lng': 22.934978090042275, 'labeledLatLngs': [{'label': 'display', 'lat': 40.642369604339585, 'lng': 22.934978090042275}], 'cc': 'GR', 'city': 'Θ ', 'state': 'Θ ', 'country': 'E ', 'formattedAddress': ['Λ ', 'Θ ', 'Θ ', 'E ']}
{'address': '26 0 28', 'crossStreet': 'Α ', 'lat': 40.639664583213325, 'lng': 22.931559106916918, 'labeledLatLngs': [{'label': 'display', 'lat': 40.639664583213325, 'lng': 22.931559106916918}], 'cc': 'GR', 'city': 'Thessalonika', 'country': 'E ', 'formattedAddress': ['26 0 28 (Α )', 'Thessalonika', 'E ']}
{'address': 'Φ 2-4', 'lat': 40.63821775903948, 'lng': 22.934232055055453, 'labeledLatLngs': [{'label': 'display', 'lat': 40.63821775903948, 'lng': 22.934232055055453}], 'postalCode': '546 26', 'cc': 'GR', 'city': 'Θ ', 'state': 'Θ ', 'country': 'E ', 'formattedAddress': ['Φ 2-4', '546 26 Θ ', 'Θ ', 'E ']}
{'address': 'Σ 8', 'crossStreet': 'Λ 3', 'lat': 40.635768430179496, 'lng': 22.93572908518955, 'labeledLatLngs': [{'label': 'display', 'lat': 40.635768430179496, 'lng': 22.93572908518955}], 'postalCode': '546 25', 'cc': 'GR', 'city': 'Θ ', 'state': 'Θ ', 'country': 'E ', 'formattedAddress': ['Σ 8 (Λ 3)', '546 25 Θ ', 'Θ ', 'E ']}
{'lat': 40.63840987763911, 'lng': 22.935703927591877, 'labeledLatLngs': [{'label': 'display', 'lat': 40.63840987763911, 'lng': 22.935703927591877}], 'postalCode': '546 26', 'cc': 'GR', 'city': 'Θ ', 'state': 'Θ ', 'country': 'E ', 'formattedAddress': ['546 26 Θ ', 'Θ ', 'E ']}
{'address': '3, Salaminos str.', 'crossStreet': 'Karatassou', 'lat': 40.63574624680632, 'lng': 22.93547939241569, 'labeledLatLngs': [{'label': 'display', 'lat': 40.63574624680632, 'lng': 22.93547939241569}], 'postalCode': '546 26', 'cc': 'GR', 'city': 'Θ ', 'state': 'Θ ', 'country': 'E ', 'formattedAddress': ['3, Salaminos str. (Karatassou)', '546 26 Θ ', 'Θ ', 'E ']}

```

```

{'address': 'Δ 17', 'crossStreet': 'Φ ', 'lat': 40.640104, 'lng':
22.933211, 'labeledLatLngs': [{'label': 'display', 'lat': 40.640104, 'lng':
22.933211}], 'postalCode': '546 26', 'cc': 'GR', 'city': 'Θ ', 'state':
'Θ ', 'country': 'E ', 'formattedAddress': ['Δ 17
(Φ )', '546 26 Θ ', 'Θ ', 'E ']}
{'address': 'Frangon str', 'crossStreet': 'btwn Bradouna and Moskof str', 'lat':
40.63864992338333, 'lng': 22.934979151781555, 'labeledLatLngs': [{'label':
'display', 'lat': 40.63864992338333, 'lng': 22.934979151781555}], 'postalCode':
'546 26', 'cc': 'GR', 'city': 'Θ ', 'state': 'Θ ', 'country':
'E ', 'formattedAddress': ['Frangon str (btwn Bradouna and Moskof str)',
'546 26 Θ ', 'Θ ', 'E ']}
{'address': 'Π 17', 'lat': 40.63757350954775, 'lng':
22.934311101247218, 'labeledLatLngs': [{'label': 'display', 'lat':
40.63757350954775, 'lng': 22.934311101247218}], 'cc': 'GR', 'city':
'Θ ', 'state': 'Θ ', 'country': 'E ', 'formattedAddress':
['Π 17', 'Θ ', 'Θ ', 'E ']}
{'lat': 40.64603267310816, 'lng': 22.920536879488584, 'labeledLatLngs':
[{'label': 'display', 'lat': 40.64603267310816, 'lng': 22.920536879488584}],
'cc': 'GR', 'country': 'E ', 'formattedAddress': ['E ']}
{'address': 'E 29', 'crossStreet': 'A ', 'lat': 40.6388620739888,
'lng': 22.938494478634716, 'labeledLatLngs': [{'label': 'display', 'lat':
40.6388620739888, 'lng': 22.938494478634716}], 'postalCode': '546 30', 'cc':
'GR', 'city': 'Θ ', 'state': 'Θ ', 'country': 'E ',
'formattedAddress': ['E 29 (A )', '546 30 Θ ',
Θ ', 'E ']}
{'address': 'B 7', 'crossStreet': 'B ', 'lat': 40.63767459638246,
'lng': 22.93934839943559, 'labeledLatLngs': [{'label': 'display', 'lat':
40.63767459638246, 'lng': 22.93934839943559}], 'postalCode': '546 25', 'cc':
'GR', 'city': 'Θ ', 'state': 'Θ ', 'country': 'E ',
'formattedAddress': ['B 7 (B )', '546 25 Θ ', 'Θ ',
'E ']}
{'address': 'A Π 69', 'lat': 40.64840509004396, 'lng':
22.930304652326917, 'labeledLatLngs': [{'label': 'display', 'lat':
40.64840509004396, 'lng': 22.930304652326917}], 'postalCode': '546 29', 'cc':
'GR', 'city': 'Θ ', 'state': 'Θ ', 'country': 'E ',
'formattedAddress': ['A Π 69', '546 29 Θ ', 'Θ ',
'E ']}
{'lat': 40.640254079697634, 'lng': 22.94014917921332, 'labeledLatLngs':
[{'label': 'display', 'lat': 40.640254079697634, 'lng': 22.94014917921332}],
'cc': 'GR', 'country': 'E ', 'formattedAddress': ['E ']}
{'lat': 40.64001221742726, 'lng': 22.932550801798392, 'labeledLatLngs':
[{'label': 'display', 'lat': 40.64001221742726, 'lng': 22.932550801798392}],
'cc': 'GR', 'country': 'E ', 'formattedAddress': ['E ']}
{'address': 'M 9', 'lat': 40.641275766138406, 'lng':
22.933640450717032, 'labeledLatLngs': [{'label': 'display', 'lat':
40.641275766138406, 'lng': 22.933640450717032}], 'postalCode': '546 29', 'cc':
'GR', 'city': 'Θ ', 'state': 'Θ ', 'country': 'E ',
'formattedAddress': ['M 9', '546 29 Θ ', 'Θ ',

```

```

'E ']]
{'address': 'Α Π 46', 'crossStreet': 'Δ Μ ', 'lat':
40.648156933133606, 'lng': 22.92916288928764, 'labeledLatLngs': [{'label':
'display', 'lat': 40.648156933133606, 'lng': 22.92916288928764}], 'postalCode':
'561 23', 'cc': 'GR', 'city': 'Θ ', 'state': 'Θ ', 'country':
'E ', 'formattedAddress': ['Α Π 46 (Δ Μ )', '561 23
Θ ', 'Θ ', 'E ']]
{'address': 'Β 20', 'crossStreet': 'Τ ', 'lat': 40.63813797710003,
'lng': 22.93857651947887, 'labeledLatLngs': [{'label': 'display', 'lat':
40.63813797710003, 'lng': 22.93857651947887}], 'cc': 'GR', 'city':
'Θ ', 'state': 'Θ ', 'country': 'E ', 'formattedAddress':
['Β 20 (Τ )', 'Θ ', 'Θ ', 'E ']]
{'address': 'Μ 26', 'crossStreet': 'Ν.Σ. Σ ', 'lat':
40.64466683163108, 'lng': 22.928521177576016, 'labeledLatLngs': [{'label':
'display', 'lat': 40.64466683163108, 'lng': 22.928521177576016}], 'postalCode':
'546 29', 'cc': 'GR', 'city': 'Θ ', 'state': 'Θ ', 'country':
'E ', 'formattedAddress': ['Μ 26 (Ν.Σ. Σ )', '546 29
Θ ', 'Θ ', 'E ']]
{'address': 'Γ 94', 'lat': 40.647209429034405, 'lng':
22.920441189682002, 'labeledLatLngs': [{'label': 'display', 'lat':
40.647209429034405, 'lng': 22.920441189682002}], 'postalCode': '546 27', 'cc':
'GR', 'city': 'Θ ', 'state': 'Θ ', 'country': 'E ',
'formattedAddress': ['Γ 94', '546 27 Θ ', 'Θ ',
'E ']]
{'address': 'Σ 10', 'lat': 40.63576102691674, 'lng': 22.935827913234743,
'labeledLatLngs': [{'label': 'display', 'lat': 40.63576102691674, 'lng':
22.935827913234743}], 'postalCode': '546 26', 'cc': 'GR', 'city': 'Θ ',
'state': 'Θ ', 'country': 'E ', 'formattedAddress': ['Σ
10', '546 26 Θ ', 'Θ ', 'E ']]
{'lat': 40.644447, 'lng': 22.929234, 'labeledLatLngs': [{'label': 'display',
'lat': 40.644447, 'lng': 22.929234}], 'postalCode': '546 27', 'cc': 'GR',
'city': 'Θ ', 'state': 'Θ ', 'country': 'E ',
'formattedAddress': ['546 27 Θ ', 'Θ ', 'E ']]
{'lat': 40.640888, 'lng': 22.928682, 'labeledLatLngs': [{'label': 'display',
'lat': 40.640888, 'lng': 22.928682}], 'postalCode': '546 27', 'cc': 'GR',
'city': 'Θ ', 'state': 'Θ ', 'country': 'E ',
'formattedAddress': ['546 27 Θ ', 'Θ ', 'E ']]
{'lat': 40.645077704103116, 'lng': 22.926292853425338, 'labeledLatLngs':
[{'label': 'display', 'lat': 40.645077704103116, 'lng': 22.926292853425338}],
'cc': 'GR', 'country': 'E ', 'formattedAddress': ['E ']]
{'lat': 40.6469814803752, 'lng': 22.935869192720556, 'labeledLatLngs':
[{'label': 'display', 'lat': 40.6469814803752, 'lng': 22.935869192720556}],
'cc': 'GR', 'country': 'E ', 'formattedAddress': ['E ']]
{'address': 'Φ ', 'lat': 40.638424, 'lng': 22.935581, 'labeledLatLngs':
[{'label': 'display', 'lat': 40.638424, 'lng': 22.935581}], 'postalCode': '546
26', 'cc': 'GR', 'city': 'Θ ', 'state': 'Θ ', 'country':
'E ', 'formattedAddress': ['Φ ', '546 26 Θ ', 'Θ ',
'E ']]

```

```

{'lat': 40.637684, 'lng': 22.934431, 'labeledLatLngs': [{'label': 'display',
'lat': 40.637684, 'lng': 22.934431}], 'postalCode': '546 26', 'cc': 'GR',
'city': 'Θ', 'state': 'Θ', 'country': 'E',
'formattedAddress': ['546 26 Θ', 'Θ', 'E']}
{'address': 'Π', 'lat': 40.63758669304457, 'lng': 22.934372134498275,
'labeledLatLngs': [{'label': 'display', 'lat': 40.63758669304457, 'lng':
22.934372134498275}], 'cc': 'GR', 'city': 'Θ', 'state': 'Θ',
'country': 'E', 'formattedAddress': ['Π', 'Θ',
Θ, 'E']}
{'lat': 40.6500186305885, 'lng': 22.93744644193629, 'labeledLatLngs': [{'label':
'display', 'lat': 40.6500186305885, 'lng': 22.93744644193629}], 'cc': 'GR',
'country': 'E', 'formattedAddress': ['E']}
{'lat': 40.638258, 'lng': 22.93634, 'labeledLatLngs': [{'label': 'display',
'lat': 40.638258, 'lng': 22.93634}], 'postalCode': '546 26', 'cc': 'GR', 'city':
'Θ', 'state': 'Θ', 'country': 'E', 'formattedAddress':
['546 26 Θ', 'Θ', 'E']}
{'address': 'Ο Δ 16', 'lat': 40.6375117534005, 'lng':
22.935467756587425, 'labeledLatLngs': [{'label': 'display', 'lat':
40.6375117534005, 'lng': 22.935467756587425}], 'postalCode': '546 26', 'cc':
'GR', 'city': 'Θ', 'state': 'Θ', 'country': 'E',
'formattedAddress': ['Ο Δ 16', '546 26 Θ', 'Θ',
'E']}
{'address': 'Α 43', 'crossStreet': 'E', 'lat':
40.64988829833071, 'lng': 22.925213892163807, 'labeledLatLngs': [{'label':
'display', 'lat': 40.64988829833071, 'lng': 22.925213892163807}], 'cc': 'GR',
'city': 'Α', 'state': 'Θ', 'country': 'E',
'formattedAddress': ['Α 43 (E)', 'Α', 'Θ',
'E']}
{'lat': 40.637257, 'lng': 22.936475, 'labeledLatLngs': [{'label': 'display',
'lat': 40.637257, 'lng': 22.936475}], 'postalCode': '546 26', 'cc': 'GR',
'city': 'Θ', 'state': 'Θ', 'country': 'E',
'formattedAddress': ['546 26 Θ', 'Θ', 'E']}
{'lat': 40.646435, 'lng': 22.920305, 'labeledLatLngs': [{'label': 'display',
'lat': 40.646435, 'lng': 22.920305}], 'postalCode': '546 27', 'cc': 'GR',
'city': 'Θ', 'state': 'Θ', 'country': 'E',
'formattedAddress': ['546 27 Θ', 'Θ', 'E']}
{'lat': 40.645885, 'lng': 22.919865, 'labeledLatLngs': [{'label': 'display',
'lat': 40.645885, 'lng': 22.919865}], 'cc': 'GR', 'country': 'E',
'formattedAddress': ['E']}
{'lat': 40.637007, 'lng': 22.938815, 'labeledLatLngs': [{'label': 'display',
'lat': 40.637007, 'lng': 22.938815}], 'postalCode': '546 25', 'cc': 'GR',
'city': 'Θ', 'state': 'Θ', 'country': 'E',
'formattedAddress': ['546 25 Θ', 'Θ', 'E']}
{'address': 'Γ 61', 'crossStreet': 'City Gate', 'lat':
40.64577451210042, 'lng': 22.92006779815583, 'labeledLatLngs': [{'label':
'display', 'lat': 40.64577451210042, 'lng': 22.92006779815583}], 'postalCode':
'546 27', 'cc': 'GR', 'city': 'Θ', 'state': 'Θ', 'country':
'E', 'formattedAddress': ['Γ 61 (City Gate)', '546 27

```



```

0      , 0      , 'E  ']]}
found 25 cafes

```

```

[30]:
uid
0  4f8cf843e4b04bd7c54fe648      E      Γ
1  56a2a05c498ed9b2b1614106      M
2  51694cb2e4b07d599519a366  The Blue Cup (The Blue Cup & Speakeasy)
3  5232ca05498edab594520a6f      Coffee INC
4  4b6de067f964a520f1972ce3      Mediterranean Palace Hotel

shortname address postalcode      lat      lng
0  Bougatsa Shops      546 27  40.640941  22.928244
1      Restaurant      546 26  40.638218  22.934232
2      Cocktail      546 25  40.635768  22.935729
3      Café      546 26  40.638410  22.935704
4      Hotel      546 26  40.635746  22.935479

```

```

[32]: Thessaloniki_center = d["geocode"]["center"]
Thessaloniki_center

```

```

[32]: {'lat': 40.64361, 'lng': 22.93086}

```

```

[34]: from folium import plugins

map_Thessaloniki = folium.Map(location=[40.64361, 22.93086], zoom_start=14)

def add_markers(df):
    for (j, row) in df.iterrows():
        label = folium.Popup(row["name"], parse_html=True)
        folium.CircleMarker(
            [row["lat"], row["lng"]],
            radius=5,
            popup=label,
            color='red',
            fill=True,
            fill_color='#3186cc',
            fill_opacity=0.7,
            parse_html=False).add_to(map_Thessaloniki)

add_markers(df)
hm_data = df[["lat", "lng"]].to_numpy().tolist()
map_Thessaloniki.add_child(plugins.HeatMap(hm_data))

map_Thessaloniki

```

```

[34]: <folium.folium.Map at 0x7f0a256ebdd8>

```

Above is our beautiful town Thessaloniki with little red dots presenting different items. By spotting the clusters of items we can see which neighborhood has density of coffee business.

0.4 3. Conclusion

We will need a location where we can catch out customers from “hot” location we have picked up from the map and stay in a certain distance so as to lessen the competitiveness of business.

```
[35]: lat = 40.64361
      lng = 22.93086
      map_Thessaloniki = folium.Map(location=[lat, lng], zoom_start=17)
      add_markers(df)
      folium.CircleMarker(
          [lat, lng],
          radius=15,
          popup="Oriste!",
          color='green',
          fill=True,
          fill_color='#3186cc',
          fill_opacity=0.7,
          parse_html=False).add_to(map_Thessaloniki)
      map_Thessaloniki
```

```
[35]: <folium.folium.Map at 0x7f0a257336a0>
```

Look at the blue bubble, here we find out that it will locate in the Train Station. As I know this neighborhood is safe and right at the center, and on the passing-by path of students and workers everyday.