1. Introduction: COFFEE ETHNIC IN DA NANG

In such a large and rich of coffee culture city like Da Nang, Viet Nam, it will be competitive to start up coffee business. In this case my contractor is a humble Vietnamese 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).

2. Orientation

First of all we need to collect Data of all coffee shops in Da Nang including their name, id, location (address, latitude, longitude) then pick up the "hot" neighbor where locates most of the venue. 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, **"O Day Roi!"**(Meaning "Here It Is!" in Vietnamese)

---

3. Execution steps

We import all the tools we need.

```
import requests # library to handle requests
import pandas as pd # library for data analsysis
import numpy as np
!pip install folium
import folium
```

Apply your credential ID on [FourSquare]

```
CLIENT_ID = 'your Foursquare ID' # your Foursquare ID
CLIENT_SECRET = 'your Foursquare Secret' # your Foursquare Secret
VERSION = '20180604'
LIMIT = 40
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

Get requests near Da Nang city.

import requests

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

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

Transform data into json then request geocode.

```
d = data.json()["response"]
d.keys()
dict_keys(['suggestedFilters', 'geocode', 'headerLocation', 'headerFullLocation',
'headerLocationGranularity', 'query', 'totalResults', 'suggestedBounds', 'groups'])
d["headerLocationGranularity"], d["headerLocation"], d["headerFullLocation"]
('city', 'Da Nang', 'Da Nang')
d["suggestedBounds"], d["totalResults"]
({'ne': {'lat': 16.076950003892705, 'lng': 108.22982675100442},
  'sw': {'lat': 16.059727142671775, 'lng': 108.21333222890732}},
 41)
d["geocode"]
{'cc': 'VN',
 'center': {'lat': 16.06778, 'lng': 108.22083},
 'displayString': 'Da Nang, Thành Phố Đà Nẵng, Vietnam',
```

```
'geometry': {'bounds': {'ne': {'lat': 16.11072, 'lng': 108.276871},
  'sw': {'lat': 15.982205, 'lng': 108.141861}}},
'longId': '72057594039511928',
'slug': 'turan-vietnam',
'what': '',
'where': 'da nang'}
```

We start creating group including information which is recommended.

```
d["groups"][0].keys()
dict_keys(['type', 'name', 'items'])
d["groups"][0]["type"], d["groups"][0]["name"]
('Recommended Places', 'recommended')
```

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

```
items = d["groups"][0]["items"]
print("number of items: %i" % len(items))
items[0]
number of items: 41
{'reasons': {'count': 0,
  'items': [{'reasonName': 'globalInteractionReason',
    'summary': 'This spot is popular',
    'type': 'general'}]},
 'referralId': 'e-5-5a26a41a31ac6c676705e94c-0',
 'venue': {'categories': [{'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/coffeeshop_',
    'suffix': '.png'},
   'id': '4bf58dd8d48988d1e0931735',
   'name': 'Coffee Shop',
   'pluralName': 'Coffee Shops',
   'primary': True,
   'shortName': 'Coffee Shop'}],
 'id': '5a26a41a31ac6c676705e94c',
 'location': {'address': '39',
  'cc': 'VN',
  'city': 'Đà Nẵng',
  'country': 'Việt Nam',
  'crossStreet': 'Nguyễn Thái Học',
  'formattedAddress': ['39 (Nguyễn Thái Học)',
   'Đà Nẵng',
   'Thành Phố Đà Nẵng',
   'Việt Nam'],
  'labeledLatLngs': [{'label': 'display',
    'lat': 16.068062879455486,
    'lng': 108.22351222071603}],
  'lat': 16.068062879455486,
  'lng': 108.22351222071603,
  'postalCode': '551105',
  'state': 'Thành Phố Đà Nẵng'},
```

```
  'name': 'cộng cà phê',
  'photos': {'count': 0, 'groups': []}}}
```

Testing by calling an item

From the output we can identify necessary factors of what we will use later to consider the probability of launching our up-to-coming location.

Based on that we start to organize what we have got.

```
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': '39', 'crossStreet': 'Nguyễn Thái Học', 'lat': 16.068062879455486, 'lng': 108.22351222071603, 'labeledLatLngs': [{'label': 'display', 'lat': 16.0
{'address': '17 Phan Đình Phùng', 'lat': 16.070069923328905, 'lng': 108.2234060288238, 'labeledLatLngs': [{'label': 'display', 'lat': 16.070069923328905, 'lng
{'address': '37-39 Pasteur', 'lat': 16.069698707615007, 'lng': 108.21804495108044, 'labeledLatLngs': [{'label': 'display', 'lat': 16.069698707615007, 'lng':
{'address': '96-98 Bạch Đằng', 'lat': 16.069014884488677, 'lng': 108.22497088340684, 'labeledLatLngs': [{'label': 'display', 'lat': 16.069014884488677, 'lng'
{'address': '58', 'crossStreet': 'Hoàng Văn Thụ. Hải Châu', 'lat': 16.06328567930564, 'lng': 108.22068421206521, 'labeledLatLngs': [{'label': 'display', 'lat
{'address': '186 Bach Dang', 'crossStreet': 'Tran Quoc Toan', 'lat': 16.06608485359888, 'lng': 108.2243266366388, 'labeledLatLngs': [{'label': 'display', 'la
{'address': '134 Phan Chau Trinh', 'lat': 16.063847858621482, 'lng': 108.21989220791667, 'labeledLatLngs': [{'label': 'display', 'lat': 16.063847858621482, '
{'lat': 16.066373, 'lng': 108.220971, 'labeledLatLngs': [{'label': 'display', 'lat': 16.066373, 'lng': 108.220971}], 'cc': 'VN', 'country': 'Việt Nam', 'form
{'address': '9 Quang Trung', 'crossStreet': 'Tran Phu', 'lat': 16.07534538869973, 'lng': 108.22224273760446, 'labeledLatLngs': [{'label': 'display', 'lat': 1
{'address': '146 Bach Dang St', 'lat': 16.06702, 'lng': 108.22479, 'labeledLatLngs': [{'label': 'display', 'lat': 16.06702, 'lng': 108.22479}], 'postalCode':
{'address': '79 Le Hong Phong', 'lat': 16.064163367156592, 'lng': 108.21928138013452, 'labeledLatLngs': [{'label': 'display', 'lat': 16.064163367156592, 'lng
{'address': 'Lô A4 - 2 Nguyễn Văn Linh', 'lat': 16.06079455709096, 'lng': 108.22346985340118, 'labeledLatLngs': [{'label': 'display', 'lat': 16.0607945570909
{'address': '180A Bach Dang Street, Hai Chau 1 Ward, Hai Chau District', 'crossStreet': 'Tran Quoc Toan', 'lat': 16.066268104610995, 'lng': 108.2244957989686
{'address': '1 Nguyen Van Linh', 'lat': 16.06064813996494, 'lng': 108.2169811846068, 'labeledLatLngs': [{'label': 'display', 'lat': 16.06064813996494, 'lng'
{'address': '32 Cô Giang, Phước Ninh, Hải Châu, Da Nang', 'lat': 16.06163998148517, 'lng': 108.22152256965637, 'labeledLatLngs': [{'label': 'display', 'lat':
{'address': '101 Trrần Phú', 'lat': 16.06909601096204, 'lng': 108.22396568330407, 'labeledLatLngs': [{'label': 'display', 'lat': 16.06909601096204, 'lng': 10
{'address': 'Le Loi', 'crossStreet': 'Le Loi', 'lat': 16.072559472418, 'lng': 108.22037004752178, 'labeledLatLngs': [{'label': 'display', 'lat': 16.072559472
{'address': 'Lê Đình Dương, Hải Châu, Da Nang', 'lat': 16.061322248017326, 'lng': 108.21981112190863, 'labeledLatLngs': [{'label': 'display', 'lat': 16.06132
{'address': '88', 'crossStreet': 'Nguyễn Chí Thanh', 'lat': 16.07085130717108, 'lng': 108.22122981692984, 'labeledLatLngs': [{'label': 'display', 'lat': 16.0
{'address': '74 Bach Dang', 'crossStreet': 'Phan Dinh Phung', 'lat': 16.0699462321021, 'lng': 108.22496141517442, 'labeledLatLngs': [{'label': 'display', 'la
{'address': '50 Bạch Đằng', 'lat': 16.073066369057457, 'lng': 108.22466120300325, 'labeledLatLngs': [{'label': 'display', 'lat': 16.073066369057457, 'lng': 1
{'address': 'Ông Ích Khiêm', 'crossStreet': 'Hung Vuong', 'lat': 16.066948386992838, 'lng': 108.21408197991173, 'labeledLatLngs': [{'label': 'display', 'lat'
{'address': 'Quang Trung', 'lat': 16.07486200847801, 'lng': 108.22006134589006, 'labeledLatLngs': [{'label': 'display', 'lat': 16.07486200847801, 'lng': 108.
{'lat': 16.068096, 'lng': 108.222755, 'labeledLatLngs': [{'label': 'display', 'lat': 16.068096, 'lng': 108.222755}], 'cc': 'VN', 'city': 'Hải Châu 1', 'state
{'address': '27 Tran Quoc Toan', 'lat': 16.066176343719146, 'lng': 108.22300314903259, 'labeledLatLngs': [{'label': 'display', 'lat': 16.066176343719146, 'ln
{'address': '102 Bạch Đằng, Thạch Thang, Hải Châu, Da Nang', 'lat': 16.068962391291606, 'lng': 108.2249717078671, 'labeledLatLngs': [{'label': 'display', 'la
{'address': '91', 'crossStreet': 'Lê Hồng Phong', 'lat': 16.064031894461532, 'lng': 108.2169015693665, 'labeledLatLngs': [{'label': 'display', 'lat': 16.064
{'address': '80A Nguyễn Chí Thanh', 'lat': 16.0725, 'lng': 108.2208, 'labeledLatLngs': [{'label': 'display', 'lat': 16.0725, 'lng': 108.2208}], 'cc': 'VN', '
{'address': 'Le Hong Phong - Da Nang', 'lat': 16.06367059719038, 'lng': 108.22347738827133, 'labeledLatLngs': [{'label': 'display', 'lat': 16.06367059719038,
{'address': '223 Trần Phú, phường Phước Ninh, quận Hải Châu', 'lat': 16.063622075206023, 'lng': 108.22350606322289, 'labeledLatLngs': [{'label': 'display', '
{'address': 'Bach Dang', 'crossStreet': 'under Han River Bridge', 'lat': 16.072871723022736, 'lng': 108.2251196519525, 'labeledLatLngs': [{'label': 'display'
{'address': '143 Le Loi', 'lat': 16.073552306534197, 'lng': 108.219980167977, 'labeledLatLngs': [{'label': 'display', 'lat': 16.073552306534197, 'lng': 108.2
{'address': '156 Le Dinh Duong', 'crossStreet': 'Trieu Nu Vuong', 'lat': 16.062463396498668, 'lng': 108.21678225824697, 'labeledLatLngs': [{'label': 'display
{'lat': 16.060794905109915, 'lng': 108.22040768373152, 'labeledLatLngs': [{'label': 'display', 'lat': 16.060794905109915, 'lng': 108.22040768373152}], 'cc':
{'address': '59-61 Nguyen Van Linh, Hai Chau', 'lat': 16.06065, 'lng': 108.22036, 'labeledLatLngs': [{'label': 'display', 'lat': 16.06065, 'lng': 108.22036}]
{'address': '04 Quang Trung', 'lat': 16.075100872070294, 'lng': 108.22139809463792, 'labeledLatLngs': [{'label': 'display', 'lat': 16.075100872070294, 'lng':
{'lat': 16.07550910076423, 'lng': 108.21971266035493, 'labeledLatLngs': [{'label': 'display', 'lat': 16.07550910076423, 'lng': 108.21971266035493}], 'cc': 'V
{'address': 'Nguyễn Văn Linh', 'lat': 16.060564393091386, 'lng': 108.2170831688409, 'labeledLatLngs': [{'label': 'display', 'lat': 16.060564393091386, 'lng':
{'lat': 16.06051, 'lng': 108.21709, 'labeledLatLngs': [{'label': 'display', 'lat': 16.06051, 'lng': 108.21709}], 'cc': 'VN', 'country': 'Việt Nam', 'formatte
{'address': '69 Nguyễn Chí Thanh', 'lat': 16.07616714656448, 'lng': 108.22046560241405, 'labeledLatLngs': [{'label': 'display', 'lat': 16.07616714656448, 'ln
{'address': '339 Tran Hung Dao', 'lat': 16.071583, 'lng': 108.229077, 'labeledLatLngs': [{'label': 'display', 'lat': 16.071583, 'lng': 108.229077}], 'cc': 'V

found 6 cafes

	uid	name	shortname	address	postalcode	lat	lng
0	5a26a41a31ac6c676705e94c	cộng cà phê	Coffee Shop		551105	16.068063	108.223512
1	5627c637498e21ca8f40f4be	Leaf Drinks and More	Coffee Shop		550000	16.067020	108.224790
2	55ff7cf0498e203b2932d5fd	Namunamu Cafe 180	Café		550000	16.066268	108.224496
3	51bd1671498e58791c2ea92d	Freeman Coffee	Coffee Shop		550000	16.070851	108.221230
4	4c25fdd8f1272d7f33ac85c5	Highlands Coffee	Café		60999	16.069946	108.224961

As we can see that there are many coffee shops without address we need to execute hasattr() to determine if each object (coffeeshop) has a attribute (address).

Next step we will execute a very important part - get coordinates of Da Nang and create folium map which will help visualize what we have got from data.

```
DaNang_center = d["geocode"]["center"]
DaNang_center
{'lat': 16.06778, 'lng': 108.22083}
from folium import plugins


map_DaNang = folium.Map(location=[16.06778, 108.22083], zoom_start=14)

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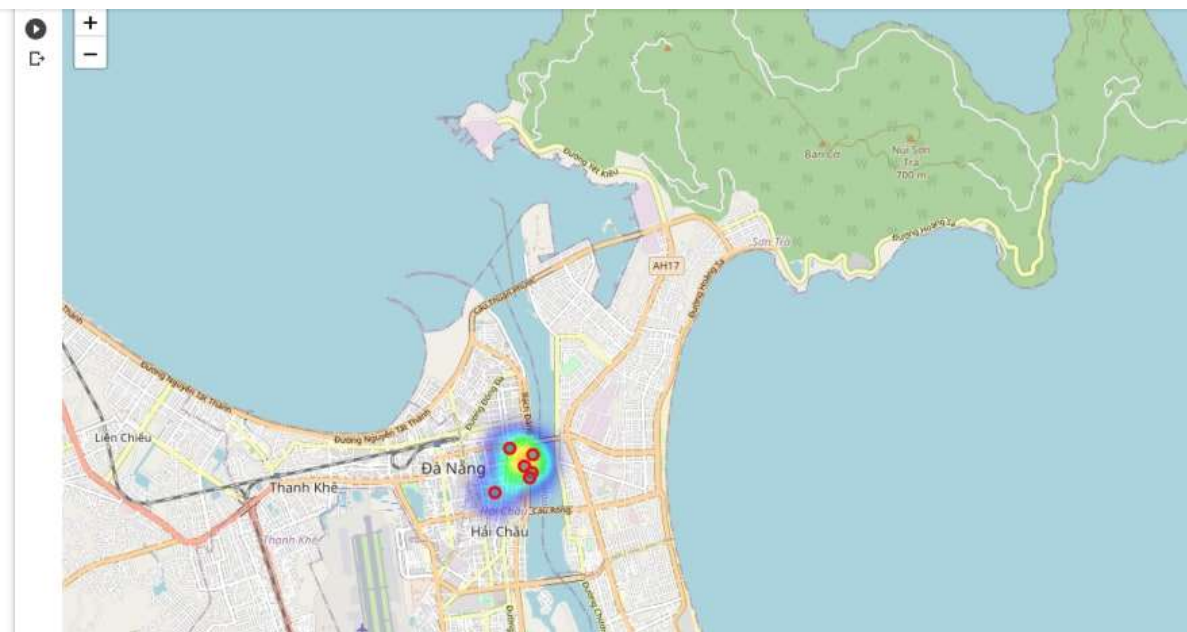
```
        [row["lat"], row["lng"]],
        radius=5,
        popup=label,
        color='red',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_DaNang)

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

map_DaNang
```

Here is our beautiful Da Nang with little red dots presenting different items.

By spotting the clusters of items we can see which neighborhood has density of coffee business.

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 competivity of business.

```
lat = 16.06778
lng = 108.22083
map_DaNang = folium.Map(location=[lat, lng], zoom_start=17)
add_markers(df)
folium.CircleMarker(
    [lat, lng],
    radius=15,
    popup="O Day Roi!",
    color='green',
    fill=True,
    fill_color='#3186cc',
    fill_opacity=0.7,
    parse_html=False).add_to(map_DaNang)
map_DaNang
```

Look at the blue buble, here we find out that it will locate about the crossroad between Pham Hong Thai street and Nguyen Chi Thanh street. As I know this neighborhood is safe and right at the center, it's off to the hot spot of night life and on the passing-by path of high school students in the area.

Wish You All The Best !!!