DP0701EN-2-2-1-Foursquare-API-py-v1.0

December 18, 2018

Learning FourSquare API with Python

0.1 Introduction

In this lab, you will learn in details how to make calls to the Foursquare API for different purposes. You will learn how to construct a URL to send a request to the API to search for a specific type of venues, to explore a particular venue, to explore a Foursquare user, to explore a geographical location, and to get trending venues around a location. Also, you will learn how to use the visualization library, Folium, to visualize the results.

0.2 Table of Contents

0.2.1 Import necessary Libraries

In [1]:
```
!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim # module to convert an address into latitude and l
import requests # library to handle requests
import pandas as pd # library for data analsysis
import numpy as np # library to handle data in a vectorized manner
import random # library for random number generation

# libraries for displaying images
from IPython.display import Image
from IPython.core.display import HTML

# tranforming json file into a pandas dataframe library
from pandas.io.json import json_normalize

!conda install -c conda-forge folium=0.5.0 --yes
import folium # plotting library
```

```
        print('Folium installed')
        print('Libraries imported.')
```

```
Solving environment: done

## Package Plan ##

  environment location: /home/jupyterlab/conda

  added / updated specs:
    - geopy


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    geopy-1.18.1               |             py_0          51 KB  conda-forge
    conda-4.5.12               |        py36_1000         653 KB  conda-forge
    geographiclib-1.49         |             py_0          32 KB  conda-forge
    ------------------------------------------------------------
                                           Total:         737 KB

The following NEW packages will be INSTALLED:

    geographiclib: 1.49-py_0     conda-forge
    geopy:         1.18.1-py_0   conda-forge

The following packages will be UPDATED:

    conda:         4.5.12-py36_0             --> 4.5.12-py36_1000 conda-forge


Downloading and Extracting Packages
geopy-1.18.1         | 51 KB     | ##################################### | 100%
conda-4.5.12         | 653 KB    | ##################################### | 100%
geographiclib-1.49   | 32 KB     | ##################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Solving environment: done

## Package Plan ##

  environment location: /home/jupyterlab/conda

  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
    branca-0.3.1               |             py_0          25 KB  conda-forge
    altair-2.3.0               |        py36_1001         533 KB  conda-forge
    pandas-0.23.4              |   py36hf8a1672_0        27.8 MB  conda-forge
    folium-0.5.0               |             py_0          45 KB  conda-forge
    ------------------------------------------------------------
                                           Total:        28.4 MB

The following NEW packages will be INSTALLED:

    altair:  2.3.0-py36_1001       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

The following packages will be UPDATED:

    pandas:  0.23.4-py37h04863e7_0             --> 0.23.4-py36hf8a1672_0 conda-forge


Downloading and Extracting Packages
vincent-0.4.4        | 28 KB     | ##################################### | 100%
branca-0.3.1         | 25 KB     | ##################################### | 100%
altair-2.3.0         | 533 KB    | ##################################### | 100%
pandas-0.23.4        | 27.8 MB   | ##################################### | 100%
folium-0.5.0         | 45 KB     | ##################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Folium installed
Libraries imported.
```

0.2.2 Define Foursquare Credentials and Version

Make sure that you have created a Foursquare developer account and have your credentials handy

```
In [2]: CLIENT_ID = 'AUDP4MRVSHCP2RDWBPDWJSOSBUFMHCYPOMHYOWO2ESKEBM4D' # your Foursquare ID
        CLIENT_SECRET = 'QRDC34SWNOZARTH3I4MBOKPNIOYIPUAOYRO1BILPQID3AIS1' # your Foursquare Sec
        VERSION = '20180604'
```

```
        LIMIT = 30
        print('Your credentails:')
        print('CLIENT_ID: ' + CLIENT_ID)
        print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: AUDP4MRVSHCP2RDWBPDWJSOSBUFMHCYPOMHYOWO2ESKEBM4D
CLIENT_SECRET:QRDC34SWNOZARTH3I4MBOKPNIOYIPUAOYRO1BILPQID3AIS1
```

Let's again assume that you are staying at the Conrad hotel. So let's start by converting the Contrad Hotel's address to its latitude and longitude coordinates.

```
In [3]: address = '102 North End Ave, New York, NY'

        geolocator = Nominatim()
        location = geolocator.geocode(address)
        latitude = location.latitude
        longitude = location.longitude
        print(latitude, longitude)
```

```
/home/jupyterlab/conda/lib/python3.6/site-packages/ipykernel_launcher.py:3: DeprecationWarning:
  This is separate from the ipykernel package so we can avoid doing imports until


40.7149555 -74.0153365
```

0.3 1. Search for a specific venue category

`https://api.foursquare.com/v2/venues/`**search**`?client_id=`**CLIENT_ID**`&client_secret=`**CLIENT_SEC**

Now, let's assume that it is lunch time, and you are craving Italian food. So, let's define a query to search for Italian food that is within 500 metres from the Conrad Hotel.

```
In [6]: search_query = 'Italian'
        radius = 500
        print(search_query + ' .... OK!')
```

```
Italian ... OK!
```

Define the corresponding URL

```
In [7]: url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{
        url
```

```
Out[7]: 'https://api.foursquare.com/v2/venues/search?client_id=AUDP4MRVSHCP2RDWBPDWJSOSBUFMHCYPO
```

Send the GET Request and examine the results

```
In [9]: results = requests.get(url).json()
        results
```

```
Out[9]: {'meta': {'code': 200, 'requestId': '5c188f84f594df03f024d105'},
         'response': {'venues': [{'id': '4fa862b3e4b0ebff2f749f06',
            'name': "Harry's Italian Pizza Bar",
            'location': {'address': '225 Murray St',
             'lat': 40.71521779064671,
             'lng': -74.01473940209351,
             'labeledLatLngs': [{'label': 'display',
               'lat': 40.71521779064671,
               'lng': -74.01473940209351}],
             'distance': 58,
             'postalCode': '10282',
             'cc': 'US',
             'city': 'New York',
             'state': 'NY',
             'country': 'United States',
             'formattedAddress': ['225 Murray St',
              'New York, NY 10282',
              'United States']},
            'categories': [{'id': '4bf58dd8d48988d1ca941735',
              'name': 'Pizza Place',
              'pluralName': 'Pizza Places',
              'shortName': 'Pizza',
              'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/pizza_',
               'suffix': '.png'},
              'primary': True}],
            'delivery': {'id': '294544',
             'url': 'https://www.seamless.com/menu/harrys-italian-pizza-bar-225-murray-st-new-yo
             'provider': {'name': 'seamless',
              'icon': {'prefix': 'https://fastly.4sqi.net/img/general/cap/',
               'sizes': [40, 50],
               'name': '/delivery_provider_seamless_20180129.png'}}},
            'referralId': 'v-1545113476',
            'hasPerk': False},
           {'id': '4f3232e219836c91c7bfde94',
            'name': 'Conca Cucina Italian Restaurant',
            'location': {'address': '63 W Broadway',
             'lat': 40.71446,
             'lng': -74.010086,
             'labeledLatLngs': [{'label': 'display',
               'lat': 40.71446,
               'lng': -74.010086}],
             'distance': 446,
             'postalCode': '10007',
             'cc': 'US',
```

```
   'city': 'New York',
   'state': 'NY',
   'country': 'United States',
   'formattedAddress': ['63 W Broadway',
    'New York, NY 10007',
    'United States']},
  'categories': [{'id': '4d4b7105d754a06374d81259',
    'name': 'Food',
    'pluralName': 'Food',
    'shortName': 'Food',
    'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/default_',
     'suffix': '.png'},
    'primary': True}],
  'referralId': 'v-1545113476',
  'hasPerk': False},
 {'id': '3fd66200f964a520f4e41ee3',
  'name': 'Ecco',
  'location': {'address': '124 Chambers St',
   'crossStreet': 'btwn Church St & W Broadway',
   'lat': 40.71533713859952,
   'lng': -74.00884766217825,
   'labeledLatLngs': [{'label': 'display',
     'lat': 40.71533713859952,
     'lng': -74.00884766217825}],
   'distance': 549,
   'postalCode': '10007',
   'cc': 'US',
   'city': 'New York',
   'state': 'NY',
   'country': 'United States',
   'formattedAddress': ['124 Chambers St (btwn Church St & W Broadway)',
    'New York, NY 10007',
    'United States']},
  'categories': [{'id': '4bf58dd8d48988d110941735',
    'name': 'Italian Restaurant',
    'pluralName': 'Italian Restaurants',
    'shortName': 'Italian',
    'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/italian_',
     'suffix': '.png'},
    'primary': True}],
  'referralId': 'v-1545113476',
  'hasPerk': False}]}}
```


Get relevant part of JSON and transform it into a *pandas* dataframe

```
In [10]: # assign relevant part of JSON to venues
         venues = results['response']['venues']
```

```
# tranform venues into a dataframe
dataframe = json_normalize(venues)
dataframe.head()
```

Out[10]:

```
                                          categories delivery.id  \
0  [{'id': '4bf58dd8d48988d1ca941735', 'name': 'P...      294544
1  [{'id': '4d4b7105d754a06374d81259', 'name': 'F...         NaN
2  [{'id': '4bf58dd8d48988d110941735', 'name': 'I...         NaN

                delivery.provider.icon.name  \
0  /delivery_provider_seamless_20180129.png
1                                       NaN
2                                       NaN

              delivery.provider.icon.prefix delivery.provider.icon.sizes  \
0  https://fastly.4sqi.net/img/general/cap/                     [40, 50]
1                                       NaN                          NaN
2                                       NaN                          NaN

  delivery.provider.name                                       delivery.url  \
0               seamless  https://www.seamless.com/menu/harrys-italian-p...
1                    NaN                                                NaN
2                    NaN                                                NaN

   hasPerk                        id location.address      ...       \
0    False  4fa862b3e4b0ebff2f749f06    225 Murray St      ...
1    False  4f3232e219836c91c7bfde94    63 W Broadway      ...
2    False  3fd66200f964a520f4e41ee3  124 Chambers St      ...

          location.crossStreet location.distance  \
0                          NaN                58
1                          NaN               446
2  btwn Church St & W Broadway               549

                           location.formattedAddress  \
0  [225 Murray St, New York, NY 10282, United Sta...
1  [63 W Broadway, New York, NY 10007, United Sta...
2  [124 Chambers St (btwn Church St & W Broadway)...

                             location.labeledLatLngs  location.lat  \
0  [{'label': 'display', 'lat': 40.71521779064671...     40.715218
1  [{'label': 'display', 'lat': 40.71446, 'lng': ...     40.714460
2  [{'label': 'display', 'lat': 40.71533713859952...     40.715337

  location.lng location.postalCode  location.state  \
0   -74.014739               10282             NY
1   -74.010086               10007             NY
2   -74.008848               10007             NY
```

```
                              name    referralId
0        Harry's Italian Pizza Bar  v-1545113476
1  Conca Cucina Italian Restaurant  v-1545113476
2                             Ecco  v-1545113476

[3 rows x 23 columns]
```

Define information of interest and filter dataframe

```
In [12]: # keep only columns that include venue name, and anything that is associated with locat
         filtered_columns = ['name', 'categories'] + [col for col in dataframe.columns if col.st
         dataframe_filtered = dataframe.loc[:, filtered_columns]

         # function that extracts the category of the venue
         def get_category_type(row):
             try:
                 categories_list = row['categories']
             except:
                 categories_list = row['venue.categories']

             if len(categories_list) == 0:
                 return None
             else:
                 return categories_list[0]['name']

         # filter the category for each row
         dataframe_filtered['categories'] = dataframe_filtered.apply(get_category_type, axis=1)

         # clean column names by keeping only last term
         dataframe_filtered.columns = [column.split('.')[-1] for column in dataframe_filtered.co

         dataframe_filtered
```

```
Out[12]:                              name          categories          address  cc  \
         0        Harry's Italian Pizza Bar         Pizza Place    225 Murray St  US
         1  Conca Cucina Italian Restaurant                Food    63 W Broadway  US
         2                             Ecco  Italian Restaurant  124 Chambers St  US

                city        country                  crossStreet  distance  \
         0  New York  United States                          NaN        58
         1  New York  United States                          NaN       446
         2  New York  United States  btwn Church St & W Broadway       549

                                           formattedAddress  \
         0  [225 Murray St, New York, NY 10282, United Sta...
         1  [63 W Broadway, New York, NY 10007, United Sta...
         2  [124 Chambers St (btwn Church St & W Broadway)...
```

```
                                          labeledLatLngs        lat        lng  \
0  [{'label': 'display', 'lat': 40.71521779064671...  40.715218 -74.014739
1  [{'label': 'display', 'lat': 40.71446, 'lng': ...  40.714460 -74.010086
2  [{'label': 'display', 'lat': 40.71533713859952...  40.715337 -74.008848

  postalCode state                        id
0      10282    NY  4fa862b3e4b0ebff2f749f06
1      10007    NY  4f3232e219836c91c7bfde94
2      10007    NY  3fd66200f964a520f4e41ee3
```

Let's visualize the Italian restaurants that are nearby

In [13]:
```
dataframe_filtered.name
```

Out[13]:
```
0          Harry's Italian Pizza Bar
1    Conca Cucina Italian Restaurant
2                               Ecco
Name: name, dtype: object
```

In [14]:
```
venues_map = folium.Map(location=[latitude, longitude], zoom_start=13) # generate map c

# add a red circle marker to represent the Conrad Hotel
folium.features.CircleMarker(
    [latitude, longitude],
    radius=10,
    color='red',
    popup='Conrad Hotel',
    fill = True,
    fill_color = 'red',
    fill_opacity = 0.6
).add_to(venues_map)

# add the Italian restaurants as blue circle markers
for lat, lng, label in zip(dataframe_filtered.lat, dataframe_filtered.lng, dataframe_fi
    folium.features.CircleMarker(
        [lat, lng],
        radius=5,
        color='blue',
        popup=label,
        fill = True,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(venues_map)

# display map
venues_map
```

Out[14]:
```
<folium.folium.Map at 0x7fb2540bf668>
```

0.4 2. Explore a Given Venue

https://api.foursquare.com/v2/venues/**VENUE_ID**?client_id=**CLIENT_ID**&client_secret=**CLIENT**

0.4.1 A. Let's explore the closest Italian restaurant -- *Harry's Italian Pizza Bar*

```
In [15]: venue_id = '4fa862b3e4b0ebff2f749f06' # ID of Harry's Italian Pizza Bar
         url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.form
         url
```

```
Out[15]: 'https://api.foursquare.com/v2/venues/4fa862b3e4b0ebff2f749f06?client_id=AUDP4MRVSHCP2R
```

Send GET request for result

```
In [16]: result = requests.get(url).json()
         print(result['response']['venue'].keys())
         result['response']['venue']
```

```
dict_keys(['id', 'name', 'contact', 'location', 'canonicalUrl', 'categories', 'verified', 'stats
```

```
Out[16]: {'id': '4fa862b3e4b0ebff2f749f06',
          'name': "Harry's Italian Pizza Bar",
          'contact': {'phone': '2126081007', 'formattedPhone': '(212) 608-1007'},
          'location': {'address': '225 Murray St',
           'lat': 40.71521779064671,
           'lng': -74.01473940209351,
           'labeledLatLngs': [{'label': 'display',
             'lat': 40.71521779064671,
             'lng': -74.01473940209351}],
           'postalCode': '10282',
           'cc': 'US',
           'city': 'New York',
           'state': 'NY',
           'country': 'United States',
           'formattedAddress': ['225 Murray St',
            'New York, NY 10282',
            'United States']},
          'canonicalUrl': 'https://foursquare.com/v/harrys-italian-pizza-bar/4fa862b3e4b0ebff2f7
          'categories': [{'id': '4bf58dd8d48988d1ca941735',
            'name': 'Pizza Place',
            'pluralName': 'Pizza Places',
            'shortName': 'Pizza',
            'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/pizza_',
             'suffix': '.png'},
            'primary': True},
           {'id': '4bf58dd8d48988d110941735',
            'name': 'Italian Restaurant',
            'pluralName': 'Italian Restaurants',
```

```
  'shortName': 'Italian',
  'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/italian_',
   'suffix': '.png'}}],
'verified': False,
'stats': {'tipCount': 55},
'url': 'http://harrysitalian.com',
'price': {'tier': 2, 'message': 'Moderate', 'currency': '$'},
'hasMenu': True,
'likes': {'count': 117,
 'groups': [{'type': 'others', 'count': 117, 'items': []}],
 'summary': '117 Likes'},
'dislike': False,
'ok': False,
'rating': 7.2,
'ratingColor': 'C5DE35',
'ratingSignals': 206,
'delivery': {'id': '294544',
 'url': 'https://www.seamless.com/menu/harrys-italian-pizza-bar-225-murray-st-new-york
 'provider': {'name': 'seamless',
  'icon': {'prefix': 'https://fastly.4sqi.net/img/general/cap/',
   'sizes': [40, 50],
   'name': '/delivery_provider_seamless_20180129.png'}}},
'menu': {'type': 'Menu',
 'label': 'Menu',
 'anchor': 'View Menu',
 'url': 'https://foursquare.com/v/harrys-italian-pizza-bar/4fa862b3e4b0ebff2f749f06/me
 'mobileUrl': 'https://foursquare.com/v/4fa862b3e4b0ebff2f749f06/device_menu'},
'allowMenuUrlEdit': True,
'beenHere': {'count': 0,
 'unconfirmedCount': 0,
 'marked': False,
 'lastCheckinExpiredAt': 0},
'specials': {'count': 0, 'items': []},
'photos': {'count': 146,
 'groups': [{'type': 'checkin',
   'name': "Friends' check-in photos",
   'count': 0,
   'items': []},
  {'type': 'venue',
   'name': 'Venue photos',
   'count': 146,
   'items': [{'id': '4fad980de4b091b4626c3633',
     'createdAt': 1336776717,
     'source': {'name': 'Foursquare for Android',
      'url': 'https://foursquare.com/download/#/android'},
     'prefix': 'https://fastly.4sqi.net/img/general/',
     'suffix': '/ya1iQFI7pLjuIJp1PGDKlrZS3OJdHCF7tpILMmjv_2w.jpg',
     'width': 480,
```

```
     'height': 640,
     'user': {'id': '13676709',
      'firstName': 'Leony',
      'lastName': 'Naciri',
      'gender': 'none',
      'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
       'suffix': '/TOANFNGNMCHUDEUE.jpg'}},
     'visibility': 'public'}]}],
 'summary': '0 photos'},
'reasons': {'count': 1,
 'items': [{'summary': 'Lots of people like this place',
   'type': 'general',
   'reasonName': 'rawLikesReason'}]},
'hereNow': {'count': 0, 'summary': 'Nobody here', 'groups': []},
'createdAt': 1336435379,
'tips': {'count': 55,
 'groups': [{'type': 'others',
   'name': 'All tips',
   'count': 55,
   'items': [{'id': '53d27909498e0523841340b6',
     'createdAt': 1406302473,
     'text': "Harry's Italian Pizza bar is known for it's amazing pizza, but did you k
     'type': 'user',
     'canonicalUrl': 'https://foursquare.com/item/53d27909498e0523841340b6',
     'lang': 'en',
     'likes': {'count': 4,
      'groups': [{'type': 'others',
        'count': 4,
        'items': [{'id': '369426',
          'firstName': 'P.',
          'lastName': 'M.',
          'gender': 'male',
          'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
           'suffix': '/JPQYUWJKUTOH2OO4.jpg'}},
         {'id': '87587879',
          'firstName': 'Diane',
          'lastName': 'Danneels',
          'gender': 'female',
          'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
           'suffix': '/87587879-ESLRSZLQ2CBE2P4W.jpg'}},
         {'id': '87591341',
          'firstName': 'Tim',
          'lastName': 'Sheehan',
          'gender': 'male',
          'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
           'suffix': '/-Z4YK4VKEOJSVXIY1.jpg'}},
         {'id': '87473404',
          'firstName': 'TenantKing.com',
```

```
            'gender': 'none',
            'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
             'suffix': '/87473404-HI5DTBTK0HX401CA.png'},
            'type': 'page'}]}],
        'summary': '4 likes'},
       'logView': True,
       'agreeCount': 4,
       'disagreeCount': 0,
       'todo': {'count': 0},
       'user': {'id': '87473404',
        'firstName': 'TenantKing.com',
        'gender': 'none',
        'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
         'suffix': '/87473404-HI5DTBTK0HX401CA.png'},
        'type': 'page'}}]}]},
 'shortUrl': 'http://4sq.com/JNblHV',
 'timeZone': 'America/New_York',
 'listed': {'count': 50,
  'groups': [{'type': 'others',
    'name': 'Lists from other people',
    'count': 50,
    'items': [{'id': '4fa32fd0e4b04193744746b1',
      'name': 'Manhattan Haunts',
      'description': '',
      'type': 'others',
      'user': {'id': '24592223',
       'firstName': 'Becca',
       'lastName': 'McArthur',
       'gender': 'female',
       'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
        'suffix': '/24592223-RAW2UYM0GIB1U40K.jpg'}},
      'editable': False,
      'public': True,
      'collaborative': False,
      'url': '/becca_mcarthur/list/manhattan-haunts',
      'canonicalUrl': 'https://foursquare.com/becca_mcarthur/list/manhattan-haunts',
      'createdAt': 1336094672,
      'updatedAt': 1380845377,
      'photo': {'id': '4e8cc9461081e3b3544e12e5',
       'createdAt': 1317849414,
       'prefix': 'https://fastly.4sqi.net/img/general/',
       'suffix': '/0NLVU2HC1JF4DXIMKWUFW3QBUT31DC11EFNYYHMJG3NDWAPS.jpg',
       'width': 492,
       'height': 330,
       'user': {'id': '742542',
        'firstName': 'Time Out New York',
        'gender': 'none',
        'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
```

```
    'suffix': '/XXHKCBSQHBORZNSR.jpg'},
   'type': 'page'},
  'visibility': 'public'},
 'followers': {'count': 22},
 'listItems': {'count': 187,
  'items': [{'id': 'v4fa862b3e4b0ebff2f749f06',
    'createdAt': 1342934485}]}},
{'id': '4fae817be4b085f6b2a74d19',
 'name': 'USA NYC MAN FiDi',
 'description': 'Where to go for decent eats in the restaurant wasteland of Downto
 'type': 'others',
 'user': {'id': '12113441',
  'firstName': 'Kino',
  'gender': 'male',
  'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
   'suffix': '/12113441-K5HTHFLU2MUCM0CM.jpg'}},
 'editable': False,
 'public': True,
 'collaborative': False,
 'url': '/kinosfault/list/usa-nyc-man-fidi',
 'canonicalUrl': 'https://foursquare.com/kinosfault/list/usa-nyc-man-fidi',
 'createdAt': 1336836475,
 'updatedAt': 1536019882,
 'photo': {'id': '55984992498e13ba75e353bb',
  'createdAt': 1436043666,
  'prefix': 'https://fastly.4sqi.net/img/general/',
  'suffix': '/12113441_iOa6Uh-Xi8bhj2-gpzkkw8MKiAIs7RmOcz_RM7m8ink.jpg',
  'width': 540,
  'height': 960,
  'user': {'id': '12113441',
   'firstName': 'Kino',
   'gender': 'male',
   'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
    'suffix': '/12113441-K5HTHFLU2MUCM0CM.jpg'}},
  'visibility': 'public'},
 'followers': {'count': 20},
 'listItems': {'count': 272,
  'items': [{'id': 'v4fa862b3e4b0ebff2f749f06',
    'createdAt': 1373909433}]}},
{'id': '5266c68a498e7c667807fe09',
 'name': 'Foodie Love in NY - 02',
 'description': '',
 'type': 'others',
 'user': {'id': '547977',
  'firstName': 'WiLL',
  'gender': 'male',
  'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
   'suffix': '/-Q5NYGDMFDMOITQRR.jpg'}},
```

```
      'editable': False,
      'public': True,
      'collaborative': False,
      'url': '/sweetiewill/list/foodie-love-in-ny--02',
      'canonicalUrl': 'https://foursquare.com/sweetiewill/list/foodie-love-in-ny--02',
      'createdAt': 1382467210,
      'updatedAt': 1391995585,
      'followers': {'count': 7},
      'listItems': {'count': 200,
       'items': [{'id': 'v4fa862b3e4b0ebff2f749f06',
         'createdAt': 1386809936}]}},
     {'id': '4fddeff0e4b0e078037ac0d3',
      'name': 'NYC Resturants',
      'description': '',
      'type': 'others',
      'user': {'id': '21563126',
       'firstName': 'Richard',
       'lastName': 'Revilla',
       'gender': 'male',
       'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
        'suffix': '/21563126_v05J1KPw_SVj6Ehq9g8B9jeAGjFUMsU5QGl-NZ8inUQ7pKQm5bKplW37Em
      'editable': False,
      'public': True,
      'collaborative': True,
      'url': '/rickr7/list/nyc-resturants',
      'canonicalUrl': 'https://foursquare.com/rickr7/list/nyc-resturants',
      'createdAt': 1339944944,
      'updatedAt': 1544180338,
      'photo': {'id': '5072dd13e4b09145cdf782d1',
       'createdAt': 1349704979,
       'prefix': 'https://fastly.4sqi.net/img/general/',
       'suffix': '/208205_fGh2OuAZ9qJ4agbAA5wMVNOSIm9kNUlRtNwj1N-adqg.jpg',
       'width': 800,
       'height': 800,
       'user': {'id': '208205',
        'firstName': 'Thalia',
        'lastName': 'K',
        'gender': 'female',
        'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
         'suffix': '/SNOOLCAW2AGO4ZKD.jpg'}},
       'visibility': 'public'},
      'followers': {'count': 12},
      'listItems': {'count': 198,
       'items': [{'id': 't54ed3b13498e857fd7dbb6fc',
         'createdAt': 1514680908}]}}]}],
   'hours': {'status': 'Closed until 11:30 AM',
    'richStatus': {'entities': [], 'text': 'Closed until 11:30 AM'},
    'isOpen': False,
```

```
 'isLocalHoliday': False,
 'dayData': [],
 'timeframes': [{'days': 'MonWed, Sun',
   'includesToday': True,
   'open': [{'renderedTime': '11:30 AM11:00 PM'}],
   'segments': []},
  {'days': 'ThuSat',
   'open': [{'renderedTime': '11:30 AMMidnight'}],
   'segments': []}]},
'popular': {'isOpen': False,
 'isLocalHoliday': False,
 'timeframes': [{'days': 'Today',
   'includesToday': True,
   'open': [{'renderedTime': 'Noon2:00 PM'},
    {'renderedTime': '5:00 PM10:00 PM'}],
   'segments': []},
  {'days': 'WedThu',
   'open': [{'renderedTime': 'Noon2:00 PM'},
    {'renderedTime': '5:00 PM10:00 PM'}],
   'segments': []},
  {'days': 'Fri',
   'open': [{'renderedTime': 'Noon3:00 PM'},
    {'renderedTime': '5:00 PM11:00 PM'}],
   'segments': []},
  {'days': 'Sat',
   'open': [{'renderedTime': 'Noon11:00 PM'}],
   'segments': []},
  {'days': 'Sun',
   'open': [{'renderedTime': 'Noon3:00 PM'},
    {'renderedTime': '5:00 PM8:00 PM'}],
   'segments': []},
  {'days': 'Mon',
   'open': [{'renderedTime': 'Noon2:00 PM'},
    {'renderedTime': '6:00 PM8:00 PM'}],
   'segments': []}]},
'pageUpdates': {'count': 0, 'items': []},
'inbox': {'count': 0, 'items': []},
'attributes': {'groups': [{'type': 'price',
   'name': 'Price',
   'summary': '$$',
   'count': 1,
   'items': [{'displayName': 'Price', 'displayValue': '$$', 'priceTier': 2}]},
  {'type': 'payments',
   'name': 'Credit Cards',
   'summary': 'Credit Cards',
   'count': 7,
   'items': [{'displayName': 'Credit Cards',
     'displayValue': 'Yes (incl. American Express)'}]},
```

```
   {'type': 'outdoorSeating',
    'name': 'Outdoor Seating',
    'summary': 'Outdoor Seating',
    'count': 1,
    'items': [{'displayName': 'Outdoor Seating', 'displayValue': 'Yes'}]},
   {'type': 'serves',
    'name': 'Menus',
    'summary': 'Happy Hour, Brunch & more',
    'count': 8,
    'items': [{'displayName': 'Brunch', 'displayValue': 'Brunch'},
     {'displayName': 'Lunch', 'displayValue': 'Lunch'},
     {'displayName': 'Dinner', 'displayValue': 'Dinner'},
     {'displayName': 'Happy Hour', 'displayValue': 'Happy Hour'}]},
   {'type': 'drinks',
    'name': 'Drinks',
    'summary': 'Beer, Wine & Cocktails',
    'count': 5,
    'items': [{'displayName': 'Beer', 'displayValue': 'Beer'},
     {'displayName': 'Wine', 'displayValue': 'Wine'},
     {'displayName': 'Cocktails', 'displayValue': 'Cocktails'}]},
   {'type': 'diningOptions',
    'name': 'Dining Options',
    'summary': 'Delivery',
    'count': 5,
    'items': [{'displayName': 'Delivery', 'displayValue': 'Delivery'}]}]},
 'bestPhoto': {'id': '4fad980de4b091b4626c3633',
  'createdAt': 1336776717,
  'source': {'name': 'Foursquare for Android',
   'url': 'https://foursquare.com/download/#/android'},
  'prefix': 'https://fastly.4sqi.net/img/general/',
  'suffix': '/ya1iQFI7pLjuIJp1PGDKlrZS3OJdHCF7tpILMmjv_2w.jpg',
  'width': 480,
  'height': 640,
  'visibility': 'public'},
 'colors': {'highlightColor': {'photoId': '4fad980de4b091b4626c3633',
   'value': -13619152},
  'highlightTextColor': {'photoId': '4fad980de4b091b4626c3633', 'value': -1},
  'algoVersion': 3}}
```


0.4.2 B. Get the venue's overall rating

```
In [17]: try:
             print(result['response']['venue']['rating'])
         except:
             print('This venue has not been rated yet.')
```

```
7.2
```

That is not a very good rating. Let's check the rating of the second closest Italian restaurant.

```
In [18]: venue_id = '4f3232e219836c91c7bfde94' # ID of Conca Cucina Italian Restaurant
         url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.form

         result = requests.get(url).json()
         try:
             print(result['response']['venue']['rating'])
         except:
             print('This venue has not been rated yet.')
```

```
This venue has not been rated yet.
```

Since this restaurant has no ratings, let's check the third restaurant.

```
In [19]: venue_id = '3fd66200f964a520f4e41ee3' # ID of Ecco
         url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.form

         result = requests.get(url).json()
         try:
             print(result['response']['venue']['rating'])
         except:
             print('This venue has not been rated yet.')
```

```
7.8
```

Since this restaurant has a slightly better rating, let's explore it further.

0.4.3 C. Get the number of tips

```
In [20]: result['response']['venue']['tips']['count']
```

```
Out[20]: 16
```

0.4.4 D. Get the venue's tips

https://api.foursquare.com/v2/venues/**VENUE_ID**/tips?client_id=**CLIENT_ID**&client_secret=**CL**

Create URL and send GET request. Make sure to set limit to get all tips

```
In [21]: ## Ecco Tips
         limit = 15 # set limit to be greater than or equal to the total number of tips
         url = 'https://api.foursquare.com/v2/venues/{}/tips?client_id={}&client_secret={}&v={}&

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

```
Out[21]: {'meta': {'code': 200, 'requestId': '5c189099351e3d5f504e3fa8'},
          'response': {'tips': {'count': 16,
            'items': [{'id': '5ab1cb46c9a517174651d3fe',
              'createdAt': 1521601350,
              'text': 'A+ Italian food! Trust me on this: my moms side of the family is 100% Ita
              'type': 'user',
              'canonicalUrl': 'https://foursquare.com/item/5ab1cb46c9a517174651d3fe',
              'lang': 'en',
              'likes': {'count': 0, 'groups': []},
              'logView': True,
              'agreeCount': 2,
              'disagreeCount': 0,
              'todo': {'count': 0},
              'user': {'id': '484542633',
               'firstName': 'Nick',
               'lastName': 'El-Tawil',
               'gender': 'male',
               'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
                'suffix': '/484542633_mK2Yum7T_7Tn9fWpndidJsmw2Hof_6T5vJBKCHPLMK5OL-U5ZiJGj51iwB
              'authorInteractionType': 'liked'}]}}}
```

Get tips and list of associated features

```
In [22]: tips = results['response']['tips']['items']

         tip = results['response']['tips']['items'][0]
         tip.keys()

Out[22]: dict_keys(['id', 'createdAt', 'text', 'type', 'canonicalUrl', 'lang', 'likes', 'logView
```

Format column width and display all tips

```
In [23]: pd.set_option('display.max_colwidth', -1)

         tips_df = json_normalize(tips) # json normalize tips

         # columns to keep
         filtered_columns = ['text', 'agreeCount', 'disagreeCount', 'id', 'user.firstName', 'use
         tips_filtered = tips_df.loc[:, filtered_columns]

         # display tips
         tips_filtered

Out[23]:
         0  A+ Italian food! Trust me on this: my moms side of the family is 100% Italian. I was

            agreeCount  disagreeCount                        id user.firstName  \
         0  2           0             5ab1cb46c9a517174651d3fe  Nick
```

```
  user.lastName user.gender    user.id
0  El-Tawil      male        484542633
```

Now remember that because we are using a personal developer account, then we can access only 2 of the restaurant's tips, instead of all 15 tips.

0.5 3. Search a Foursquare User

`https://api.foursquare.com/v2/users/`**USER_ID**`?client_id=`**CLIENT_ID**`&client_secret=`**CLIENT_S**

0.5.1 Define URL, send GET request and display features associated with user

```
In [24]: user_id = '484542633' # user ID with most agree counts and complete profile

         url = 'https://api.foursquare.com/v2/users/{}?client_id={}&client_secret={}&v={}'.forma

         # send GET request
         results = requests.get(url).json()
         user_data = results['response']['user']

         # display features associated with user
         user_data.keys()

Out[24]: dict_keys(['id', 'firstName', 'lastName', 'gender', 'canonicalUrl', 'photo', 'friends',

In [25]: print('First Name: ' + user_data['firstName'])
         print('Last Name: ' + user_data['lastName'])
         print('Home City: ' + user_data['homeCity'])

First Name: Nick
Last Name: El-Tawil
Home City: New York, NY
```

How many tips has this user submitted?

```
In [26]: user_data['tips']

Out[26]: {'count': 246}
```

Wow! So it turns out that Nick is a very active Foursquare user, with more than 250 tips.

0.5.2 Get User's tips

```
In [27]: # define tips URL
         url = 'https://api.foursquare.com/v2/users/{}/tips?client_id={}&client_secret={}&v={}&l

         # send GET request and get user's tips
```

```
results = requests.get(url).json()
tips = results['response']['tips']['items']

# format column width
pd.set_option('display.max_colwidth', -1)

tips_df = json_normalize(tips)

# filter columns
filtered_columns = ['text', 'agreeCount', 'disagreeCount', 'id']
tips_filtered = tips_df.loc[:, filtered_columns]

# display user's tips
tips_filtered
```

Out[27]:

```
0   The best! Im especially fond of the salmon burger, but Ive had half of the menu and
1   I used to down a pint of chocolate like it was nothing back when I was bulking. Hig
2   They serve coffee!!!!!!
3   Im a fan. In fact, Im such a big fan, I want Taim to hire me to be their spokesman.
4   The linguine with clams is on point
5   Great for a quick, cheap lunch! Shorter lines than Chipotle too
6   Quick, cheap lunch that tastes good! Way shorter line than Chipotle, too.
7   Great for a quick, cheap bite. I stop by when Im in the neighborhood and dont have
8   Youre not a real New Yorker until youve shame-ordered Insomnia Cookies for delivery
9   Good for you yet still tasty! Clean green protein is my go-to after I hit the gym
10  Coffee game on point
11  This is the dive bar to end all other dive bars. Go here if you like cheap drinks!
12  Burger game strong
13  Great burgers & fries! Also, this place is exactly what its like when you go to a b
14  That guy looks familiar...

    agreeCount  disagreeCount                        id
0   1           0              5aec594b1f7440002c138612
1   1           0              5accc9f66fa81f196724807b
2   1           0              5accc98c0313204c9d7ec157
3   1           0              5accbf033abcaf09a24612a0
4   1           0              5accbe3a911fc423730f3ed3
5   1           0              5acbecb86fa81f1967e019b0
6   1           0              5acbec70a0215b732e264fe8
7   1           0              5acbec0c6fa81f1967dfeba3
8   1           0              5acbbd4eb1538e45373b07f5
9   1           0              5acbbcda01235808d5d6dc75
10  1           0              5acbbb1501235808d5d6525e
11  1           0              5ab576abea1e444f2abb051e
12  1           0              5ab575fb6bdee65f759da8c1
13  2           0              5ab5575d73fe2516ad8f363b
14  1           0              5ab5299635f98312029a53b7
```

Let's get the venue for the tip with the greatest number of agree counts

```
In [28]: tip_id = '5ab5575d73fe2516ad8f363b' # tip id

         # define URL
         url = 'http://api.foursquare.com/v2/tips/{}?client_id={}&client_secret={}&v={}'.format(

         # send GET Request and examine results
         result = requests.get(url).json()
         print(result['response']['tip']['venue']['name'])
         print(result['response']['tip']['venue']['location'])
```

```
Cowgirl
{'address': '519 Hudson St', 'crossStreet': 'at W 10th St', 'lat': 40.73375031678678, 'lng': -74
```

0.5.3 Get User's friends

```
In [29]: user_friends = json_normalize(user_data['friends']['groups'][0]['items'])
         user_friends
```

```
Out[29]: Empty DataFrame
         Columns: []
         Index: []
```

Interesting. Despite being very active, it turns out that Nick does not have any friends on Foursquare. This might definitely change in the future.

0.5.4 Retrieve the User's Profile Image

```
In [31]: user_data
```

```
Out[31]: {'id': '484542633',
          'firstName': 'Nick',
          'lastName': 'El-Tawil',
          'gender': 'male',
          'canonicalUrl': 'https://foursquare.com/nickeltawil',
          'photo': {'prefix': 'https://fastly.4sqi.net/img/user/',
           'suffix': '/484542633_mK2Yum7T_7Tn9fWpndidJsmw2Hof_6T5vJBKCHPLMK5OL-U5ZiJGj51iwBstcpD
          'friends': {'count': 0,
           'groups': [{'type': 'others',
             'name': 'Other friends',
             'count': 0,
             'items': []}]},
          'tips': {'count': 246},
          'homeCity': 'New York, NY',
          'bio': 'https://www.tawil.team/nick-el-tawil/',
          'contact': {},
          'photos': {'count': 0, 'items': []},
```

```
 'type': 'user',
 'mayorships': {'count': 0, 'items': []},
 'checkins': {'count': 1, 'items': []},
 'lists': {'count': 2,
  'groups': [{'type': 'created', 'count': 0, 'items': []},
   {'type': 'followed', 'count': 0, 'items': []},
   {'type': 'yours',
    'count': 2,
    'items': [{'id': '484542633/todos',
      'name': "Nick's Saved Places",
      'description': '',
      'type': 'todos',
      'editable': False,
      'public': True,
      'collaborative': False,
      'url': '/nickeltawil/list/todos',
      'canonicalUrl': 'https://foursquare.com/nickeltawil/list/todos',
      'listItems': {'count': 0}},
     {'id': '484542633/venuelikes',
      'name': 'Nicks Liked Places',
      'description': '',
      'type': 'likes',
      'editable': False,
      'public': True,
      'collaborative': False,
      'url': '/nickeltawil/list/venuelikes',
      'canonicalUrl': 'https://foursquare.com/nickeltawil/list/venuelikes',
      'listItems': {'count': 0}}]}]},
  'lenses': []}
```

In [32]:
```
# 1. grab prefix of photo
# 2. grab suffix of photo
# 3. concatenate them using the image size
Image(url='https://igx.4sqi.net/img/user/300x300/484542633_mK2Yum7T_7Tn9fWpndidJsmw2Hof
```

Out[32]:
```
<IPython.core.display.Image object>
```

0.6 4. Explore a location

https://api.foursquare.com/v2/venues/**explore**?client_id=**CLIENT_ID**&client_secret=**CLIENT_SE**

So, you just finished your gourmet dish at Ecco, and are just curious about the popular spots around the restaurant. In order to explore the area, let's start by getting the latitude and longitude values of Ecco Restaurant.

In [33]:
```
latitude = 40.715337
longitude = -74.008848
```

Define URL

```
In [34]: url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&ll={}
         url

Out[34]: 'https://api.foursquare.com/v2/venues/explore?client_id=AUDP4MRVSHCP2RDWBPDWJSOSBUFMHCY
```

Send GET request and examine results

```
In [35]: import requests

In [36]: results = requests.get(url).json()
         'There are {} around Ecco restaurant.'.format(len(results['response']['groups'][0]['ite

Out[36]: 'There are 30 around Ecco restaurant.'
```

Get relevant part of JSON

```
In [37]: items = results['response']['groups'][0]['items']
         items[0]

Out[37]: {'reasons': {'count': 0,
            'items': [{'summary': 'This spot is popular',
              'type': 'general',
              'reasonName': 'globalInteractionReason'}]},
           'venue': {'id': '4af5d65ff964a52091fd21e3',
            'name': 'Korin',
            'location': {'address': '57 Warren St',
             'crossStreet': 'Church St',
             'lat': 40.71482437714839,
             'lng': -74.00940425461492,
             'labeledLatLngs': [{'label': 'display',
               'lat': 40.71482437714839,
               'lng': -74.00940425461492}],
             'distance': 73,
             'postalCode': '10007',
             'cc': 'US',
             'neighborhood': 'Tribeca',
             'city': 'New York',
             'state': 'NY',
             'country': 'United States',
             'formattedAddress': ['57 Warren St (Church St)',
              'New York, NY 10007',
              'United States']},
            'categories': [{'id': '4bf58dd8d48988d1f8941735',
              'name': 'Furniture / Home Store',
              'pluralName': 'Furniture / Home Stores',
              'shortName': 'Furniture / Home',
              'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/shops/furniture_',
```

```
    'suffix': '.png'},
   'primary': True}],
 'photos': {'count': 0, 'groups': []},
 'venuePage': {'id': '33104775'}},
'referralId': 'e-0-4af5d65ff964a52091fd21e3-0'}
```

Process JSON and convert it to a clean dataframe

In [38]:
```
dataframe = json_normalize(items) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories'] + [col for col in dataframe.colum
dataframe_filtered = dataframe.loc[:, filtered_columns]

# filter the category for each row
dataframe_filtered['venue.categories'] = dataframe_filtered.apply(get_category_type, ax

# clean columns
dataframe_filtered.columns = [col.split('.')[-1] for col in dataframe_filtered.columns]

dataframe_filtered.head(10)
```

Out[38]:

	name	categories	address	cc	\
0	Korin	Furniture / Home Store	57 Warren St	US	
1	Chambers Street Wines	Wine Shop	148 Chambers St	US	
2	Takahachi Bakery	Bakery	25 Murray St	US	
3	Takahachi	Sushi Restaurant	145 Duane St	US	
4	Juice Press	Vegetarian / Vegan Restaurant	83 Murray St	US	
5	Nish Nsh	Falafel Restaurant	88 Reade St	US	
6	Mulberry & Vine	Café	73 Warren St	US	
7	Ten Over Ten	Nail Salon	112 Reade St	US	
8	Equinox Tribeca	Gym	54 Murray Street	US	
9	The Odeon	French Restaurant	145 W Broadway	US	

	city	country	crossStreet	distance	\
0	New York	United States	Church St	73	
1	New York	United States	btwn West Broadway & Hudson St	88	
2	New York	United States	at Church St	179	
3	New York	United States	btwn W Broadway & Church St	146	
4	New York	United States	btwn Greenwich St & W Broadway	202	
5	New York	United States	at Church St	97	
6	New York	United States	btwn W Broadway & Greenwich St	117	
7	New York	United States	NaN	68	
8	New York	United States	at W Broadway	159	
9	New York	United States	at Thomas St	195	

	formattedAddres
0	[57 Warren St (Church St), New York, NY 10007, United States]

```
1  [148 Chambers St (btwn West Broadway & Hudson St), New York, NY 10007, United States
2  [25 Murray St (at Church St), New York, NY 10007, United States]
3  [145 Duane St (btwn W Broadway & Church St), New York, NY 10013, United States]
4  [83 Murray St (btwn Greenwich St & W Broadway), New York, NY 10007, United States]
5  [88 Reade St (at Church St), New York, NY 10013, United States]
6  [73 Warren St (btwn W Broadway & Greenwich St), New York, NY 10007, United States]
7  [112 Reade St, New York, NY 10013, United States]
8  [54 Murray Street (at W Broadway), New York, NY 10007, United States]
9  [145 W Broadway (at Thomas St), New York, NY 10013, United States]

                                                                labeledLatLngs  \
0  [{'label': 'display', 'lat': 40.71482437714839, 'lng': -74.00940425461492}]
1  [{'label': 'display', 'lat': 40.715773063928374, 'lng': -74.00971823312332}]
2  [{'label': 'display', 'lat': 40.71372322632139, 'lng': -74.00873266967818}]
3  [{'label': 'display', 'lat': 40.71652647412374, 'lng': -74.00810108466207}]
4  [{'label': 'display', 'lat': 40.71478769908051, 'lng': -74.0111317502157}]
5  [{'label': 'display', 'lat': 40.71553710116416, 'lng': -74.00772452925565}]
6  [{'label': 'display', 'lat': 40.71517693966315, 'lng': -74.01022747778285}]
7  [{'label': 'display', 'lat': 40.71594125566931, 'lng': -74.00872053564494}]
8  [{'label': 'display', 'lat': 40.714057, 'lng': -74.009694}]
9  [{'label': 'display', 'lat': 40.71700358931347, 'lng': -74.00810144732247}]

         lat        lng neighborhood postalCode state  \
0  40.714824 -74.009404  Tribeca      10007      NY
1  40.715773 -74.009718  NaN          10007      NY
2  40.713723 -74.008733  NaN          10007      NY
3  40.716526 -74.008101  NaN          10013      NY
4  40.714788 -74.011132  NaN          10007      NY
5  40.715537 -74.007725  NaN          10013      NY
6  40.715177 -74.010227  NaN          10007      NY
7  40.715941 -74.008721  NaN          10013      NY
8  40.714057 -74.009694  NaN          10007      NY
9  40.717004 -74.008101  NaN          10013      NY

                         id
0  4af5d65ff964a52091fd21e3
1  4adcf23cf964a520cc6221e3
2  4c154c9a77cea593c401d260
3  4a8f2f39f964a520471420e3
4  54148bc6498ea7bb8c05b70a
5  50ba9119e4b071a4bae6dc10
6  5171b5cc011cef9833bbb787
7  4ce2a27cd58c60fc0fa1a76f
8  4a6e331af964a52031d41fe3
9  3fd66200f964a52083e61ee3
```

Let's visualize these items on the map around our location

```
In [39]: venues_map = folium.Map(location=[latitude, longitude], zoom_start=15) # generate map c


         # add Ecco as a red circle mark
         folium.features.CircleMarker(
             [latitude, longitude],
             radius=10,
             popup='Ecco',
             fill=True,
             color='red',
             fill_color='red',
             fill_opacity=0.6
             ).add_to(venues_map)


         # add popular spots to the map as blue circle markers
         for lat, lng, label in zip(dataframe_filtered.lat, dataframe_filtered.lng, dataframe_fi
             folium.features.CircleMarker(
                 [lat, lng],
                 radius=5,
                 popup=label,
                 fill=True,
                 color='blue',
                 fill_color='blue',
                 fill_opacity=0.6
                 ).add_to(venues_map)

         # display map
         venues_map

Out[39]: <folium.folium.Map at 0x7fb254010c18>
```

0.7 5. Explore Trending Venues

https://api.foursquare.com/v2/venues/**trending**?client_id=**CLIENT_ID**&client_secret=**CLIENT_S**

Now, instead of simply exploring the area around Ecco, you are interested in knowing the venues that are trending at the time you are done with your lunch, meaning the places with the highest foot traffic. So let's do that and get the trending venues around Ecco.

```
In [40]: # define URL
         url = 'https://api.foursquare.com/v2/venues/trending?client_id={}&client_secret={}&ll={

         # send GET request and get trending venues
         results = requests.get(url).json()
         results

Out[40]: {'meta': {'code': 200, 'requestId': '5c1891a51ed2194bd3db89c9'},
          'response': {'venues': []}}
```

0.7.1 Check if any venues are trending at this time

```
In [41]: if len(results['response']['venues']) == 0:
             trending_venues_df = 'No trending venues are available at the moment!'

         else:
             trending_venues = results['response']['venues']
             trending_venues_df = json_normalize(trending_venues)

             # filter columns
             columns_filtered = ['name', 'categories'] + ['location.distance', 'location.city',
             trending_venues_df = trending_venues_df.loc[:, columns_filtered]

             # filter the category for each row
             trending_venues_df['categories'] = trending_venues_df.apply(get_category_type, axis

In [42]: # display trending venues
         trending_venues_df

Out[42]: 'No trending venues are available at the moment!'
```

Now, depending on when you run the above code, you might get different venues since the venues with the highest foot traffic are fetched live.

0.7.2 Visualize trending venues

```
In [43]: if len(results['response']['venues']) == 0:
             venues_map = 'Cannot generate visual as no trending venues are available at the mom

         else:
             venues_map = folium.Map(location=[latitude, longitude], zoom_start=15) # generate m


             # add Ecco as a red circle mark
             folium.features.CircleMarker(
                 [latitude, longitude],
                 radius=10,
                 popup='Ecco',
                 fill=True,
                 color='red',
                 fill_color='red',
                 fill_opacity=0.6
             ).add_to(venues_map)


             # add the trending venues as blue circle markers
             for lat, lng, label in zip(trending_venues_df['location.lat'], trending_venues_df['
                 folium.features.CircleMarker(
                     [lat, lng],
```

```
                    radius=5,
                    poup=label,
                    fill=True,
                    color='blue',
                    fill_color='blue',
                    fill_opacity=0.6
                ).add_to(venues_map)
```

```
In [44]: # display map
         venues_map
```

```
Out[44]: 'Cannot generate visual as no trending venues are available at the moment!'
```

0.7.3 Thank you for completing this lab!

This notebook was created by Alex Aklson. I hope you found this lab interesting and educational. Feel free to contact me if you have any questions!

This notebook is part of a course on **Coursera** called *Applied Data Science Capstone*. If you accessed this notebook outside the course, you can take this course online by clicking here.