Capstone Project - The Battle Of Neighborhoods (Week 2)

BY: BRIANA MOSES

Scenario:

My family and I decided to go on a trip somewhere within the United States. After a long decision, we narrowed our search between Chicago, Illinois, and Seattle, Washington. After careful consideration, we had to develop a solution to break the tie between the two cities. Desserts! Since my family and I crave baked goods, we believed it would be great to stay in a hotel with a plethora of bakeries nearby.

- For my capstone project, I will be picking a scenario where my family and I will go on vacation.

- To solve this dilemma, I will use Foursquare location data on each hotel location to find the best city to have our vacation.

- Since my family is a fan of Four Seasons Hotels, we decided to stay at a Four Seasons Hotel for either city.

Business Problem:

Between the two locations, which city has the most bakeries nearby their respected Four Seasons Hotel?

Data:

The data source that I will be using to execute my solution would be Foursquare API. To narrow our search, I will be using Foursquare location data to pick the best place to have our trip where there would be various bakeries nearby. To use Foursquare location data, I would need to: make calls to the Foursquare API, define a query to search for local bakeries, determine the necessary information into a filtered data frame, and create a folium map to help visualize the bakeries nearby the hotels.

<u>Website</u>: https://foursquare.com/

Methodology:

My capstone project's main component would be to locate the number of bakeries in two locations and compare them. To prepare this data, I will need to construct a URL to request the Foursquare API to explore bakeries in the area and get the number of bakeries around each location. To visualize the results more efficiently, I will be creating a Folium map to view my data.

Step 1: Import Libraries

```python
import pandas as pd
import numpy as np
import random
import requests
!pip install geopy
from geopy.geocoders import Nominatim
from pandas.io.json import json_normalize
! pip install folium==0.5.0
import folium

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

Step 2:Define Foursquare credentials to access location datA

```python
CLIENT_ID = 'WFARCWCLP0ZX30QI2V1RCKDTQUJFI3SEW2J525OBCHUG0LEY'
CLIENT_SECRET = '1YM24G13U4YVGLOTZGHABWKDBZD4VVFBH3RQNJCIQLGBPN31'
ACCESS_TOKEN = '1SDRI00RGVCNBPFHHIOAHDBMPPDVPPQKP2PXQXPVJPC35QKQ'
VERSION = '20180604'
LIMIT = 50
print('Your credentials:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentials:
CLIENT_ID: WFARCWCLP0ZX30QI2V1RCKDTQUJFI3SEW2J525OBCHUG0LEY
CLIENT_SECRET:1YM24G13U4YVGLOTZGHABWKDBZD4VVFBH3RQNJCIQLGBPN31
```

Step 3: Define a user-agent to access a geocoder for both Four Seasons hotels

```
address = '99 Union St, Seattle, WA 98101'

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

47.607375700000006 -122.3391602000244

```
address = '120 E Delaware Pl, Chicago, IL 60611'

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

41.899607149999994 -87.6248798647693

Step 4: Define a search query for bakeries in Seattle & Chicago

```python
search_query = 'Bakery'
radius = 1500
print(search_query + ' in Seattle .... OK!')
```

```
Bakery in Seattle .... OK!
```

```python
search_query = 'Bakery'
radius = 1500
print(search_query + ' in Chicago .... OK!')
```

```
Bakery in Chicago .... OK!
```

Step 5: Define Foursquare API Url

```
url = 'https://api.foursquare.com/v2/venues/search?client_id={}&client_secret={}&ll={},{}&oauth_token={}&v={}&query={}&radius=
{}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude,ACCESS_TOKEN, VERSION, search_query, radius, LIMIT)
url
```

```
'https://api.foursquare.com/v2/venues/search?client_id=WFARCWCLP0ZX30QI2V1RCKDTQUJFI3SEW2J525OBCHUG0LEY&client_secret=1YM24G13
U4YVGLOTZGHABWKDBZD4VVFBH3RQNJCIQLGBPN31&ll=47.607375700000006,-122.3391602000244&oauth_token=1SDRI00RGVCNBPFHHIOAHDBMPPDVPPQK
P2PXQXPVJPC35QKQ&v=20180604&query=Bakery&radius=1500&limit=50'
```

Step 6.1: Send a get request to examine results(Seattle)

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

{'meta': {'code': 200, 'requestId': '603719b2720e82269232e58c'},
 'notifications': [{'type': 'notificationTray', 'item': {'unreadCount': 0}}],
 'response': {'venues': [{'id': '47f245d1f964a520974e1fe3',
    'name': 'Three Girls Bakery',
    'location': {'address': '1514 Pike Pl',
     'crossStreet': 'in Post Alley',
     'lat': 47.60906474579556,
     'lng': -122.34091319298555,
     'labeledLatLngs': [{'label': 'display',
       'lat': 47.60906474579556,
       'lng': -122.34091319298555},
      {'label': 'entrance', 'lat': 47.609094, 'lng': -122.340702}],
     'distance': 229,
     'postalCode': '98101',
     'cc': 'US',
     'city': 'Seattle',
     'state': 'WA',
     'country': 'United States',
```

# Step 6.2: Send a get request to examine results (Chicago)

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

```
{'meta': {'code': 200, 'requestId': '603719b4f53fc41dc49766f6'},
 'notifications': [{'type': 'notificationTray', 'item': {'unreadCount': 0}}],
 'response': {'venues': [{'id': '49f752d2f964a520816c1fe3',
    'name': 'Corner Bakery Cafe',
    'location': {'address': '676 N St. Clair',
     'lat': 41.894285,
     'lng': -87.623079,
     'labeledLatLngs': [{'label': 'display',
       'lat': 41.894285,
       'lng': -87.623079}],
     'distance': 610,
     'postalCode': '60611',
     'cc': 'US',
     'city': 'Chicago',
     'state': 'IL',
     'country': 'United States',
```

Step 7.1: specify json search and transform data into dataframe (Seattle)

```
venues = results['response']['venues']

dataframe = json_normalize(venues)
dataframe.head()
```

```
/opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages/ipykernel/__main__.py:3: FutureWarning: pandas.io.json.js
on_normalize is deprecated, use pandas.json_normalize instead
  app.launch_new_instance()
```

	id	name	categories	referralId	hasPerk	location.address	location.crossStreet	location.lat
0	47f245d1f964a520974e1fe3	Three Girls Bakery	[{'id': '4bf58dd8d48988d16a941735', 'name': 'B...	v-1614223794	False	1514 Pike Pl	in Post Alley	47.609065
1	4b9d10c1f964a5208f8c36e3	Pike Place Bakery	[{'id': '4bf58dd8d48988d16a941735', 'name': 'B...	v-1614223794	False	1501 Pike Pl	at Pike Place Market	47.608835
2	4c77edff93ef236afa74aa0f	Alaskan Sourdough Bakery	[{'id': '4bf58dd8d48988d16a941735', 'name': 'B...	v-1614223794	False	1301 Alaskan Way	NaN	47.606162
3	427ff980f964a520d8211fe3	Macrina Bakery	[{'id': '4bf58dd8d48988d16a941735', 'name': 'B...	v-1614223794	False	2408 1st Ave	btwn Wall & Battery	47.614406
4	4ca7b9f4f47ea143a9d46d21	Fuji Bakery	[{'id': '4bf58dd8d48988d16a941735', 'name': 'B...	v-1614223794	False	526 S King St	NaN	47.598356

Step 7.2: specify json search and transform data into dataframe (Chicago)

	id	name	categories	referralId	hasPerk	location.address	location.lat	location.lng	location.labeledLat
0	49f752d2f964a520816c1fe3	Corner Bakery Cafe	[{'id': '4bf58dd8d48988d16a941735', 'name': 'B...	v-1614223796	False	676 N St. Clair	41.894285	-87.623079	[{'label': 'display', 'la 41.894285, 'lng':...
1	4b003368f964a520743b22e3	Oak Tree Restaurant & Bakery	[{'id': '4bf58dd8d48988d14e941735', 'name': 'A...	v-1614223796	False	900 N Michigan Ave	41.899469	-87.624131	[{'label': 'display', 'la 41.89946945812373
2	4bca6e0168f976b06b7d5f83	Sacred Grounds Bakery & Café	[{'id': '4bf58dd8d48988d1e0931735', 'name': 'C...	v-1614223796	False	845 N Michigan Ave	41.897964	-87.623046	[{'label': 'display', 'la 41.89796390791032
3	5cb22ae4666116002cd42017	3 Greens Coffee & Bakery	[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...	v-1614223796	False	900 N Michigan Ave	41.899590	-87.624855	[{'label': 'display', 'la 41.89959, 'lng': ...
4	4a50c752f964a52023b01fe3	Corner Bakery Cafe	[{'id': '4bf58dd8d48988d16a941735', 'name': 'B...	v-1614223796	False	360 N Michigan Ave, ENTRANCE ON WACKER DRIVE	41.888143	-87.625123	[{'label': 'display', 'la 41.88814347156145

Step 8.1: Filtering dataframe to show relevant data (seattle)

```python
filtered_columns = ['name', 'categories'] + [col for col in dataframe.columns if col.startswith('location.')] + ['id']
dataframe_filtered = dataframe.loc[:, filtered_columns]

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']


dataframe_filtered['categories'] = dataframe_filtered.apply(get_category_type, axis=1)

dataframe_filtered.columns = [column.split('.')[-1] for column in dataframe_filtered.columns]

dataframe_filtered.head()
```

	name	categories	address	crossStreet	lat	lng	labeledLatLngs	distance	postalCode	cc	city	state	country	f
0	Three Girls Bakery	Bakery	1514 Pike Pl	in Post Alley	47.609065	-122.340913	[{'label': 'display', 'lat': 47.60906474579556...	229	98101	US	Seattle	WA	United States	['F S
1	Pike Place Bakery	Bakery	1501 Pike Pl	at Pike Place Market	47.608835	-122.340498	[{'label': 'display', 'lat': 47.60883450562949...	190	98101	US	Seattle	WA	United States	['F S

Step 8.2: Filtering dataframe to show relevant data (Chicago)

```python
filtered_columns = ['name', 'categories'] + [col for col in dataframe.columns if col.startswith('location.')] + ['id']
dataframe_filtered = dataframe.loc[:, filtered_columns]

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']


dataframe_filtered['categories'] = dataframe_filtered.apply(get_category_type, axis=1)

dataframe_filtered.columns = [column.split('.')[-1] for column in dataframe_filtered.columns]

dataframe_filtered.head()
```

	name	categories	address	lat	lng	labeledLatLngs	distance	postalCode	cc	city	state	country	formattedAddress	cro
0	Corner Bakery Cafe	Bakery	676 N St. Clair	41.894285	-87.623079	[{'label': 'display', 'lat': 41.894285, 'lng':...	610	60611	US	Chicago	IL	United States	[676 N St. Clair, Chicago, IL 60611]	NaN
1	Oak Tree Restaurant & Bakery	American Restaurant	900 N Michigan Ave	41.899469	-87.624131	[{'label': 'display', 'lat': 41.89946945812373...	63	60611	US	Chicago	IL	United States	[900 N Michigan Ave (at Walton), Chicago, IL 6...	at V

Step 9.1: List all found bakeries for each location (Seattle)

```
dataframe_filtered.name
```

```
0                  Three Girls Bakery
1                   Pike Place Bakery
2            Alaskan Sourdough Bakery
3                     Macrina Bakery
4                        Fuji Bakery
5             Il Fornaio Bakery & Café
6                      Dahlia Bakery
7                 Sugar Bakery + Cafe
8            Specialty's Café & Bakery
9               Grand Central Bakery
10                  Sun Bakery & Deli
11     The Barque Brontes Bakery Café
12                   Three Dog Bakery
13                            Bakery
14               Seattle Bagel Bakery
15                 Little Rae's Bakery
16                 Very French Bakery
17                   M's Bakery Piper
```

```
18                   Bullseye Bakery
19                  post deli & bakery
20                 Askatu Bakery Cafe
21             Uptown Espresso & Bakery
22                Opaline Cafe & Bakery
23                Wild Rye Cafe Bakery
24                    85C Bakery Cafe
25                  Holy Bee's Bakery
26               Seattle Bagel Bakery
27                Yummy House Bakery
28                   Standard Bakery
29                          Le Panier
30                          Le Chatel
31            Sisters European Snacks
32                   Uptown Espresso
33                      Panera Bread
Name: name, dtype: object
```

Step 9.2: List all found bakeries for each location(Chicago)

```
dataframe_filtered.name
0                    Corner Bakery Cafe
1            Oak Tree Restaurant & Bakery
2            Sacred Grounds Bakery & Café
3                3 Greens Coffee & Bakery
4                    Corner Bakery Cafe
5                     West Town Bakery
6                    Corner Bakery Cafe
7          Aigre Doux Restaraunt & Bakery
8                    Brown Sugar Bakery
9                      Hendricks Bakery
10                     3 Greens Market
11                        Jaffa Bakery
12       Hendrickx Belgian Bread Crafter
13                 Eva's Delicious Bakery
14                      Torry's Bakery
15         Warburtons Bakery and Cafe
16                        Jaffa Bakery
17                        Jaffa Bakery
18                         Au Bon Pain
Name: name, dtype: object
```

Step 10.1: Visualize data (Seattle, Washington)
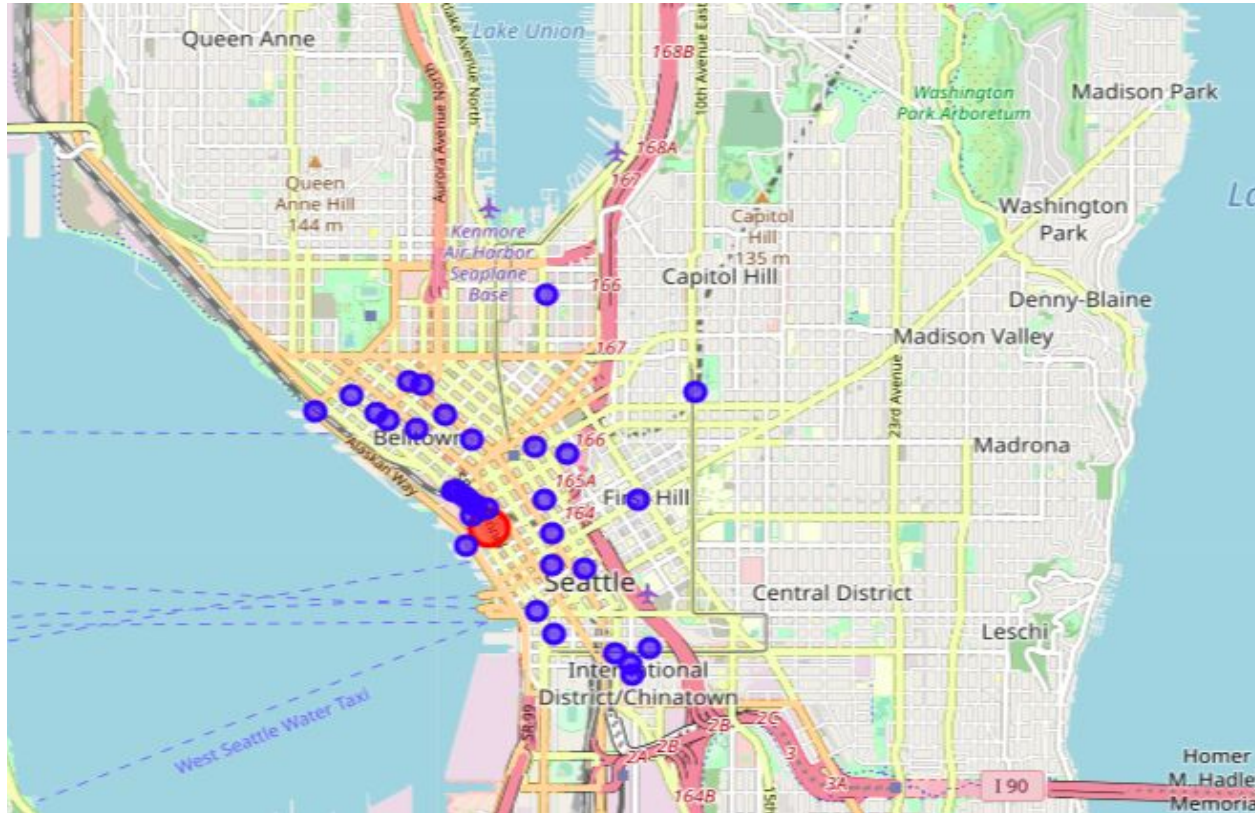
Step 10.2: Visualize data (Chicago, illinois)
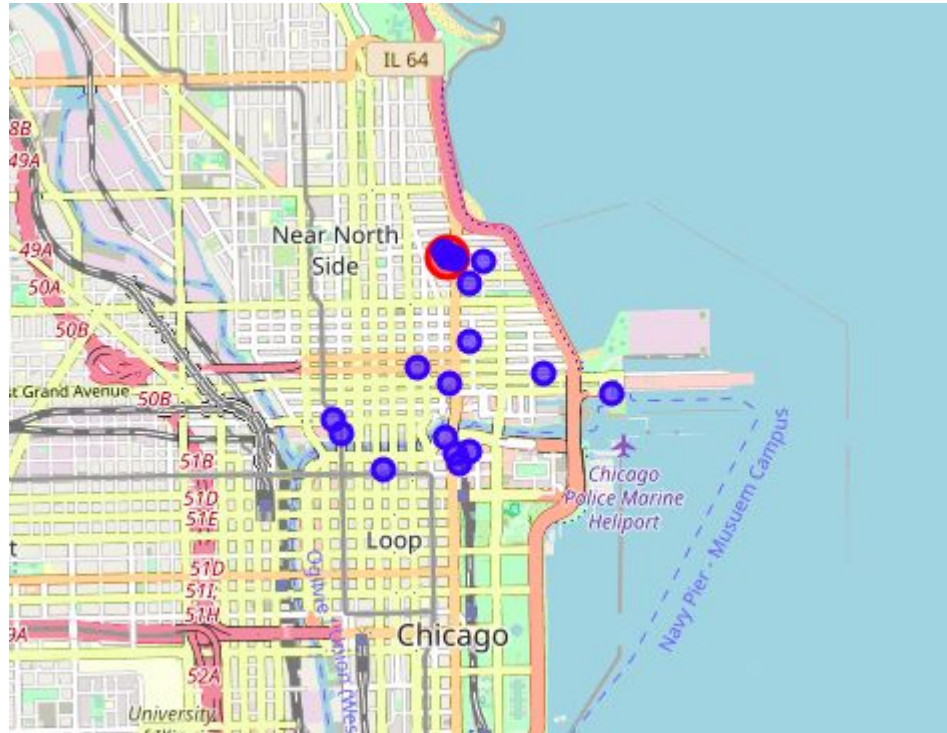
Results:

After filtering the panda dataframe for each Four Seasons hotel location, the results show that within a 1,500 radius for Four Seasons Seattle, there were 34 bakeries nearby. In contrast, Four Seasons Chicago had 19 bakeries nearby.

Discussion:

After applying extract data from Foursquare API to solve my business problem, I recognized that Seattle, Washington has the most bakeries near the Four Seasons hotel. To even further, I can access the overall ratings to all bakeries in the area. However, unfortunately, foursquare only allows a user to send a GET request to a venues' overall rating one at a time. Considering that each location encloses numerous bakery locations, retrieving all bakery ratings would be tedious and extensive.

Conclusion:

After analyzing the results, there are 34 bakeries nearby Four Seasons Seattle hotel and 19 near Four Seasons Chicago. Considering there are more bakeries near the hotel in Seattle, it would be in our families' best interest to have our vacation in <u>Seattle, Washington</u>, where more bakeries are available near our hotel.