Battle of the Neighborhoods

AN ANALYSIS ON WHERE THE BEST PLACE TO PUT A NEW RESTAURANT IN DETROIT, MI

The Problem

We have a client that wants to open a new seafood restaurant in Detroit, Michigan. They said there appears to be a lack of the venue in question, so we'll analyze that using data that's available online.

This is an important problem since there are many small business owners who wonder what the best location for their new business would be but aren't sure where to start or how to find an adequate solution except their gut reaction. So using data science tools readily available to us, we'll collect data found on the internet and analyze it to find the best solution.

The Requirements

We’re required to follow these stipulations:

1) Our area of operations (AOE) in based around Campus Martius

2) We are to be find a location within half-a-mile (~800 meters) of our AOE

3) We need to collect location data using the Foursquare API

4) We’ll need to find a building for our client in the specified radius using data scraped from the web

Building the nearby_venues DataFrame

Here's a snippet of code that creates a Pandas DataFrame object. Our starting location was Campus Martius and we looked for venues in an 800 meter radius.

```
venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
nearby_venues =nearby_venues.loc[:, filtered_columns]

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

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues.head()
```

In this particular instance, we're only showing the head of the object which displays 5 values from the dataframe.

	name	categories	lat	lng
0	Avalon Cafe and Bakery	Café	42.332834	-83.047694
1	Campus Martius	Park	42.331575	-83.046598
2	Texas de Brazil	Steakhouse	42.332293	-83.046711
3	Dime Store	American Restaurant	42.331039	-83.047734
4	Chase Bank	Bank	42.330676	-83.046648

Building the Restaurants DataFrame

Here we took the dataframe from before and created a new one called “restaurants” and populated it with values that are **only** classified as restaurants since we don’t care about parks, speakeasies, or theatres.

```
restaurants = nearby_venues[nearby_venues['categories'].str.contains("Restaurant")]
restaurants
```

	name	categories	lat	lng
3	Dime Store	American Restaurant	42.331039	-83.047734
16	The Standby	New American Restaurant	42.334439	-83.046009
19	Parc	American Restaurant	42.331564	-83.046700
23	Vicente's Cuban Cuisine	Cuban Restaurant	42.334436	-83.047193
24	Orchid Thai	Thai Restaurant	42.333579	-83.045220
28	Maru Sushi & Grill	Japanese Restaurant	42.330360	-83.048269
44	Townhouse Detroit	New American Restaurant	42.330305	-83.045361
45	Sweetwater Tavern	American Restaurant	42.331861	-83.041839
46	Caucus Club	American Restaurant	42.329436	-83.047551
47	Central Kitchen + Bar	American Restaurant	42.331518	-83.045962
53	Carnival Fresh Mex	Mexican Restaurant	42.330757	-83.047328
54	Which Wich Superior Sandwiches	Restaurant	42.330412	-83.047079
56	Jacoby's German Biergarten	German Restaurant	42.332132	-83.042073
62	The Apparatus Room	Restaurant	42.328113	-83.048299
65	Wala Urban	American Restaurant	42.330871	-83.050553
72	Bangkok Crossing	Thai Restaurant	42.330509	-83.046118
75	Go Sy Thai	Thai Restaurant	42.332993	-83.049187
91	Firebird Tavern	New American Restaurant	42.334701	-83.043345
93	The Golden Fleece	Greek Restaurant	42.335165	-83.042064
96	Fishbone's Rhythm Kitchen Cafe	Seafood Restaurant	42.334376	-83.043068
98	Pegasus Taverna	Greek Restaurant	42.335227	-83.041650
99	La Lanterna	Italian Restaurant	42.333013	-83.049096

Narrowing Down the Competition

```
seafood_restaurants = nearby_venues[nearby_venues['categories'].str.contains("Seafood")]
seafood_restaurants
```

	name	categories	lat	lng
96	Fishbone's Rhythm Kitchen Cafe	Seafood Restaurant	42.334376	-83.043068

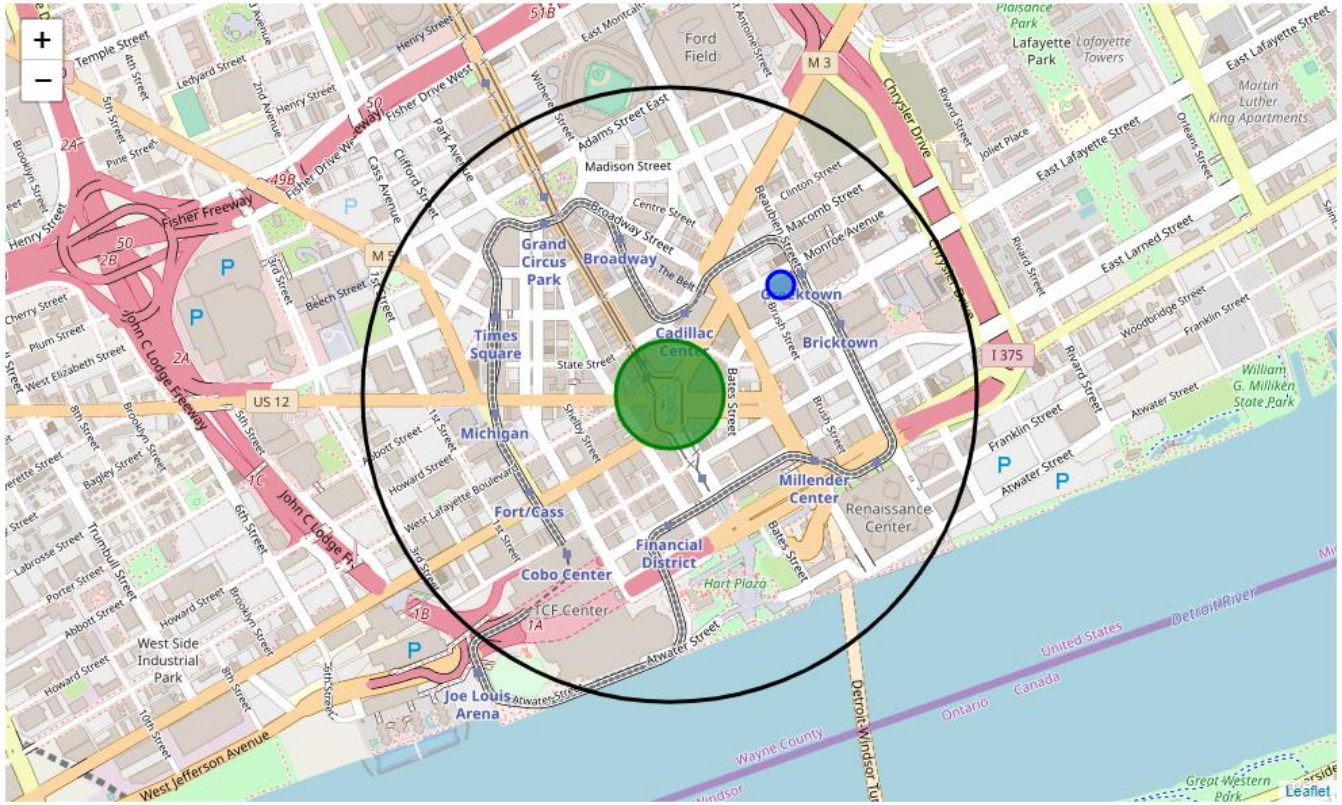
In this snippet, we're creating a dataframe to hold only restaurants that are seafood related. To our surprise we see there's only really one option. Not much in the way of competition, and that's an unexpected positive outcome for our client!

Make a Map

Here we built our map. The black outer circle is our 800 meter radius from Campus Martius.

The green circle is Campus Martius in its entireity.

The blue circle northeast of Campus Martius is our competitor, Fishbone's Café.



Let's Find a Building: The Ingredients

Since we have our radius marked already, let's find a building in the AOE that is for sale/rent that can house our client's new restaurant. We'll do this by scraping data from a website called *42floors.*

1452 Brush St - 3 Spaces Downtown Detroit, Detroit, MI Lease rate negotiable	Oct 09 27 days	3,500 - 8,250 sqft	Retail Restaurant Office
1442 Brush St - Floor 1 Downtown Detroit, Detroit, MI Lease rate negotiable and Plus utilities	Oct 09 27 days	2,800 sqft	Office Retail
2310 Park Ave - Floor 5 Downtown Detroit, Detroit, MI Lease rate negotiable	Oct 07 29 days	6,400 sqft	Office
1323 Broadway St - Unit 101 (Floor 1) Downtown Detroit, Detroit, MI Lease rate negotiable	Jun 14 144 days	3,180 sqft	Office
2233 Park Ave - Space Downtown Detroit, Detroit, MI Operating expenses: $3.00 and Partition allo...	Jun 05 153 days	5,000 sqft	Retail Restaurant
1529 Broadway St - Unit 2 Downtown Detroit, Detroit, MI This listing is a net lease lease.	May 24 165 days	7,000 sqft	Office
28 W Adams Ave - Unit 101 Downtown Detroit, Detroit, MI Kitchenette	May 17 172 days	1,130 sqft	Sublease Office

We're able to draw an area on a virtual map to specify an area we're looking for a property in. So I drew a crude 800 meter radius around Campus Martius and was able to wrangle about 7 different places available to us.

Let's Find a Building: The Soup

Now that we have a relative idea of some places we can get ahold of; we need to scrape this data, clean it, and turn it into a dataframe.

We accomplish this task by using BeautifulSoup and Pandas.

This is a snippet of the address table scraped from the website. Each element is part of an array called *table.*

```
table = soup.findAll('span', attrs={'class':'title block text-overflow text-blue'})
table
```

```
[<span class="title block text-overflow text-blue">
                1452 Brush St
                <span class="text-light">
                    -
                        3 Spaces
                </span>
 </span>, <span class="title block text-overflow text-blue">
                1442 Brush St
                <span class="text-light">
                    -
                        Floor 1
                </span>
 </span>, <span class="title block text-overflow text-blue">
                2310 Park Ave
                <span class="text-light">
                    -
                        Floor 5
                </span>
 </span>, <span class="title block text-overflow text-blue">
                1323 Broadway St
                <span class="text-light">
                    -
                        Unit 101 (Floor 1)
                </span>
 </span>, <span class="title block text-overflow text-blue">
                2233 Park Ave
                <span class="text-light">
                    -
                        Space
                </span>
 </span>, <span class="title block text-overflow text-blue">
                1529 Broadway St
                <span class="text-light">
                    -
                        Unit 2
                </span>
 </span>, <span class="title block text-overflow text-blue">
                28 W Adams Ave
                <span class="text-light">
                    -
                        Unit 101
                </span>
 </span>]
```

Let's Organize our Buildings

```
detroit_table = pd.DataFrame(list(zip(addresses, building_types)),
                columns =['Address', 'Building Type'])

detroit_table
```

	Address	Building Type
0	1452 Brush St	RetailRestaurantOffice
1	1442 Brush St	OfficeRetail
2	2310 Park Ave	Office
3	1323 Broadway St	Office
4	2233 Park Ave	RetailRestaurant
5	1529 Broadway St	Office
6	28 W Adams Ave	SubleaseOffice

We'll skip ahead to the part where we collected all the necessary data from the website and transformed it into the *detroit_table* dataframe object shown above.

Collect the Lats and Lons

In this portion of code, we use geopy to get our latitudes and longitudes for each address from the previous table.

For whatever reason the last address in our table was not able to retrieve latitudes or longitudes using geopy. Since we saw from the last table that it was a sublease office only, we're able to drop it anyway.

```
building_lats = []
building_lons = []

zipcode = '48226'
city = 'Detroit, MI'

geolocator = Nominatim(user_agent="detroit_app")

for _ in range(0,len(addresses)-1):
    place = addresses[_] + ' ' + city + ' ' + zipcode
    location = geolocator.geocode(place)
    building_lats.append(location.latitude)
    building_lons.append(location.longitude)
    print(location.latitude, location.longitude)
```

```
42.3364310475167 -83.0449531912913
42.3363419274117 -83.044893667659
42.3383745102041 -83.0539696326531
42.3348050816327 -83.0460905714286
42.3378896 -83.0538867
42.3358776 -83.0488959
```

Completing the Buildings Table

Here we have our completed dataframe that holds the addresses, building types, and coordinates for each location in question.

From this we can see that only two are really relevant for our needs as restaurant type buildings: 1452 Brush St and 2233 Park Ave.

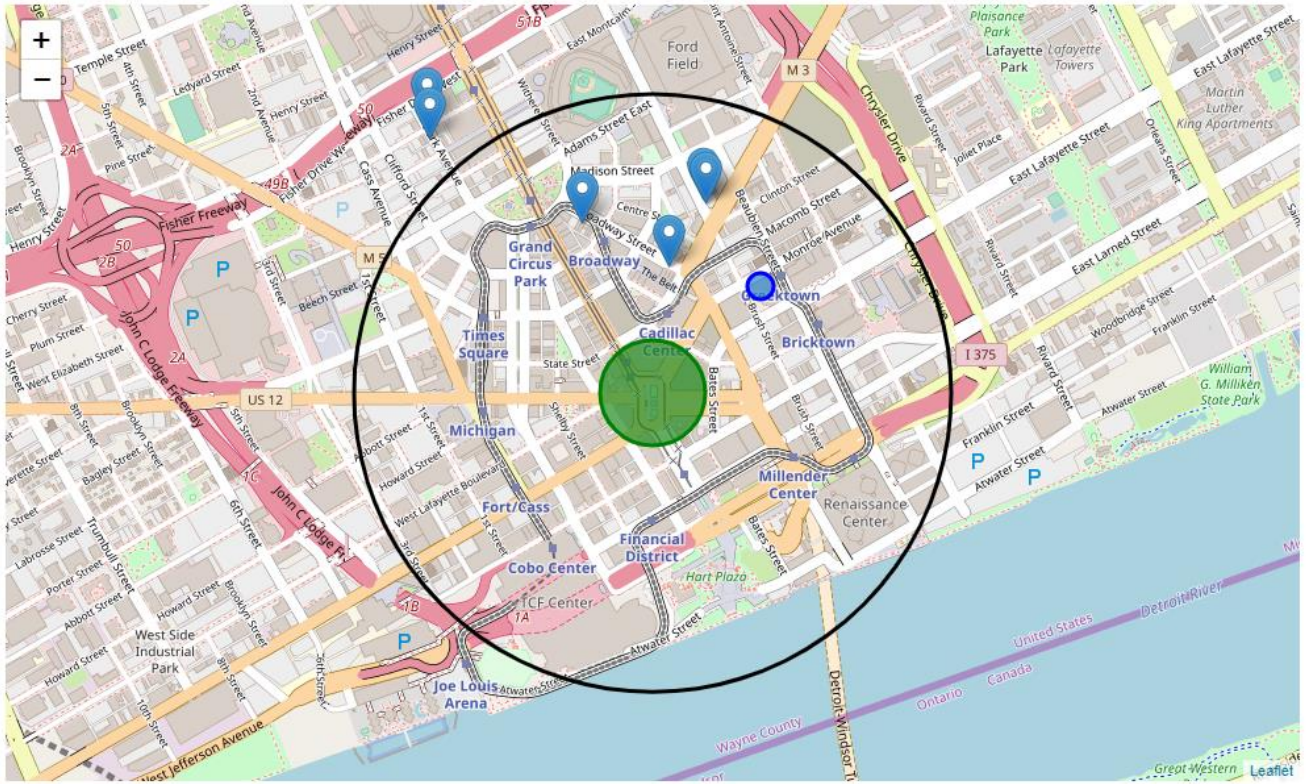
```
detroit_table['Latitude'] = building_lats
detroit_table['Longitude'] = building_lons

detroit_table
```

	Address	Building Type	Latitude	Longitude
0	1452 Brush St	RetailRestaurantOffice	42.336431	-83.044953
1	1442 Brush St	OfficeRetail	42.336342	-83.044894
2	2310 Park Ave	Office	42.338375	-83.053970
3	1323 Broadway St	Office	42.334805	-83.046091
4	2233 Park Ave	RetailRestaurant	42.337890	-83.053887
5	1529 Broadway St	Office	42.335878	-83.048896

Complete Map of Our Data

Here we see the same map as earlier, but now we have markers. Each marker is a location from the previous table.

2233 Park Ave is one of the two markers **outside** of the specified radius, while 1452 Brush is one of the two points just north-northwest of Fishbone's Café. This is our suggested location since it's the only building able to house a restaurant in our specified area.



Conclusion

Throughout this project we used various data science tools including: the Foursquare API, geopy, BeautifulSoup, and Pandas. Each tool had its own unique purpose that helped us to piece together the puzzle and answer a business question with relative ease. We were able to ask, collect, analyze, and answer a problem using data from the tools at our disposal.

This was a valuable learning experience and is highly recommended for users new to data science methodology.