# CAPSTONE PROJECT –

# BATTLE OF THE NEIGHBORHOODS

# PROBLEM AND BACKGROUND DISCUSSION

- For one city, it has a large area. Which area is more profitable to open a restaurant? Which area is more suitable to start your new business. You will feel worried about these questions when you are in a large city. I will use the former argument in the above query for this assignment and say that I have a client looking to open a restaurant in Detroit, Michigan and they want to know the best place for that. Let's presume this restaurant will be a restaurant that serves Greek food.
- Detroit has a wide variety of downtown restaurants and we'll do this analysis with the aid of Foursquare and their comprehensive location dataset about the city and its locations. Through opening this new Greek food restaurant, this will allow people to choose from a wider range while dining downtown, creating revenue for themselves, nearby locations, and the city as a whole.

# DATA USED

- I will use the Foursquare API to request location data which I will use similarly to how I used it in the assignments in New York and Toronto. I'll start by specifying the initial location (Campus Martius) and setting our 800 m radius, and then exploring the neighborhood for restaurants similar. I am also going to want to look for a location that has a reasonable amount of foot traffic and is decently far enough away from other sestern restaurants so that the market is not over-saturated in one area. They will take the sporting arenas, casinos, and other entertainment venues into account as it will produce the most foot traffic. Main streets are suitable for passenger cars and foot traffic, but are likely to be more costly to rent out.

# BUILDING THE DATA FRAME (NEARBY)

- 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]

| | name | categories | lat | lng |
|---|---|---|---|---|
| 0 | Campus Martius | Park | 42.331575 | -83.046598 |
| 1 | Avalon Cafe and Bakery | Café | 42.332834 | -83.047694 |
| 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 |
| 5 | Lafayette Coney Island | Hot Dog Joint | 42.331683 | -83.048780 |
| 6 | Bon Bon Bon | Dessert Shop | 42.330548 | -83.047914 |
| 7 | Grand Trunk Pub | Pub | 42.330403 | -83.045943 |
| 8 | Athens Souvlaki Detroit | Diner | 42.330360 | -83.048183 |
| 9 | Detroit Water Ice Factory | Ice Cream Shop | 42.332575 | -83.047436 |
| 10 | Vault of Midnight | Comic Shop | 42.334135 | -83.046621 |
| 11 | Bad Luck Bar | Speakeasy | 42.332914 | -83.049132 |
| 12 | Guardian Building | Building | 42.329497 | -83.046083 |
| 13 | The Belt | Art Gallery | 42.334386 | -83.046087 |
| 14 | Parc | American Restaurant | 42.331564 | -83.046700 |
| 15 | Dessert Oasis Coffee | Coffee Shop | 42.332840 | -83.049165 |
| 16 | Roasting Plant Detroit | Coffee Shop | 42.331354 | -83.046049 |
| 17 | The Standby | New American Restaurant | 42.334439 | -83.046009 |
| 18 | Shake Shack | Burger Joint | 42.330955 | -83.046371 |
| 19 | 7 Greens | Salad Place | 42.334094 | -83.046444 |

# BUILDING THE DATA FRAME (RESTAURANT)

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

| | name | categories | lat | lng |
|---|---|---|---|---|
| 3 | Dime Store | American Restaurant | 42.331039 | -83.047734 |
| 14 | Parc | American Restaurant | 42.331564 | -83.046700 |
| 17 | The Standby | New American Restaurant | 42.334439 | -83.046009 |
| 20 | Orchid Thai | Thai Restaurant | 42.333579 | -83.045220 |
| 22 | Vicente's Cuban Cuisine | Cuban Restaurant | 42.334436 | -83.047193 |
| 27 | Maru Sushi & Grill | Japanese Restaurant | 42.330360 | -83.048269 |
| 42 | Freshii | Restaurant | 42.331569 | -83.047713 |
| 45 | Central Kitchen + Bar | American Restaurant | 42.331518 | -83.045962 |
| 46 | Sweetwater Tavern | American Restaurant | 42.331861 | -83.041839 |
| 47 | Go Sy Thai | Thai Restaurant | 42.332993 | -83.049187 |
| 48 | Townhouse Detroit | New American Restaurant | 42.330305 | -83.045361 |
| 55 | Jacoby's German Biergarten | German Restaurant | 42.332132 | -83.042073 |
| 57 | Which Wich Superior Sandwiches | Restaurant | 42.330412 | -83.047079 |
| 61 | Carnival Fresh Mex | Mexican Restaurant | 42.330757 | -83.047328 |
| 62 | Caucus Club | American Restaurant | 42.329436 | -83.047551 |
| 64 | La Lanterna | Italian Restaurant | 42.333013 | -83.049096 |
| 66 | San Morello | Italian Restaurant | 42.334145 | -83.048628 |
| 67 | The Apparatus Room | Restaurant | 42.328113 | -83.048299 |
| 68 | Bangkok Crossing | Thai Restaurant | 42.330509 | -83.046118 |
| 86 | The Monarch Club | New American Restaurant | 42.335455 | -83.048829 |
| 92 | Firebird Tavern | New American Restaurant | 42.334701 | -83.043345 |

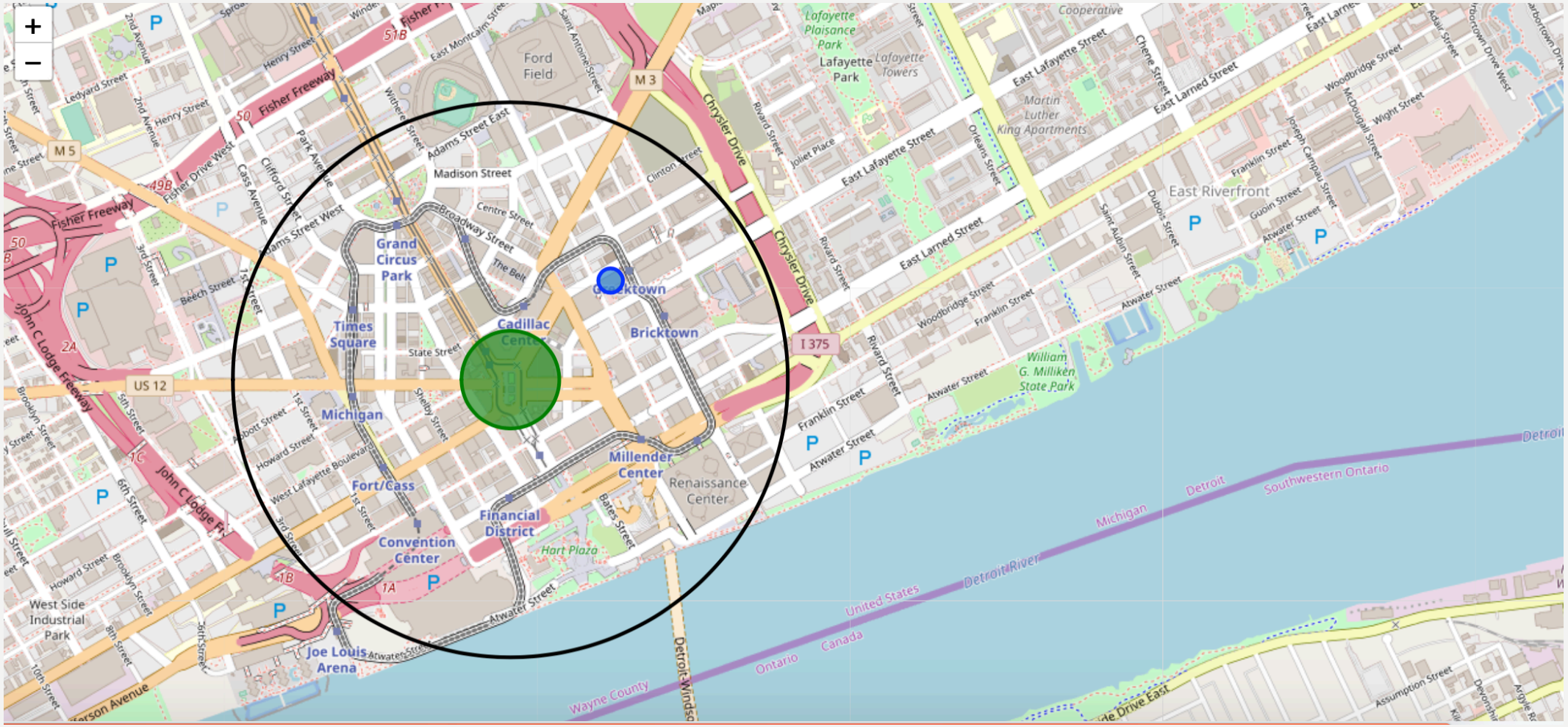
# NARROWING DOWN GREEK RESTAURANT

- I am creating datat frame to hold only restaurants that are Greek food related.

- greek_restaurants = nearby_venues[nearby_venues['categories'].str.contains("Greek")]
- greek_restaurants

| | name | categories | lat | lng |
|---|---|---|---|---|
| **98** | Pegasus Taverna | Greek Restaurant | 42.335227 | -83.04165 |

# MAKE A MAP


Ford Field
Grand Circus Park
Madison Street
Broadway Street
Centre Street
Adams Street East
Adams Street West
Times Square
Cadillac Center
Bricktown
State Street
Michigan
Fort/Cass
Financial District
Convention Center
Millender Center
Renaissance Center
Hart Plaza
Joe Louis Arena
West Side Industrial Park
East Riverfront
William G. Milliken State Park
Lafayette Park
Lafayette Plaisance Park
Martin Luther King Apartments
Fisher Freeway
John C Lodge Freeway
Chrysler Drive
East Lafayette Street
East Larned Street
Atwater Street
Franklin Street
Woodbridge Street
Michigan
Detroit
Southwestern Ontario
United States
Canada
Ontario
Detroit River
Wayne County
M 3
M 5
I 375
US 12

# BUILDINGS ORGANIZATION

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

| | Address | Building Type |
|---|---|---|
| **0** | 1442 Brush St | Office |
| **1** | 415 Clifford St | OfficeRetail |
| **2** | 1452 Brush St | RetailRestaurantOffice |
| **3** | 1442 Brush St | OfficeRetail |
| **4** | 2310 Park Ave | Office |

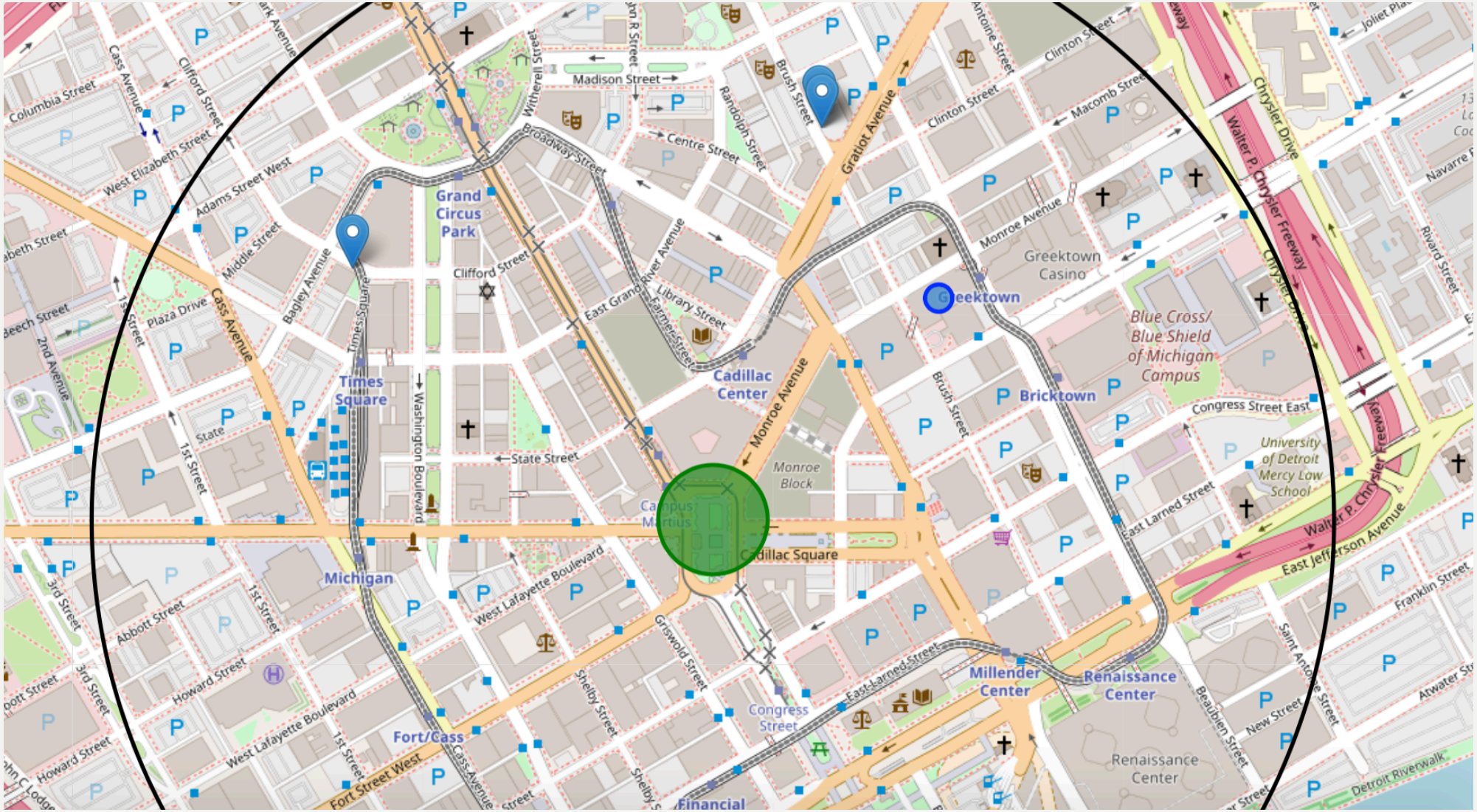
# LATS AND LONS COLLECTION

- I am trying to get lats and lons for each address

```
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)
```

```
42.33634192741168 -83.044893667659
42.334701571428575 -83.05227371428572
42.33643104751674 -83.04495319129133
42.33634192741168 -83.044893667659
```

# MAPPING MY DATA



# DISCUSSION & CONCLUSION

- There are a few different approaches that one may take to resolve this problem. That is the strategy that is most "brute-force." We collected the data relevant for Campus Martius and our defined radius as our starting place. From there we developed our map using the location data created which included seafood restaurants. To our surprise there was only one in the region that we had to take into account. Once the map was created, we then scraped building location data from a real estate website and crossed it using the geopy API to map latitudes and longitudes using Folium. From there we narrowed down our choices to include restaurant-appropriate buildings and selected our spot.
- As a whole, this project was a great learning experience on how to scrape data and use it efficiently to solve a business problem for business owners. In the future I would love to revisit this problem and use more advanced techniques to address it with relative ease. I would strongly recommend that any and all aspiring data scientists practice a problem like this to improve their own abilities.

# Thank You