Capstone Project - The Battle of the Neighborhoods (Week 2)

Table of Contents

1 Introduction: Business Problem

Fig 1-1. Amsterdam Centraal

In this project we will try to find an optimal location for opening a restaurant in **Amsterdam, Netherlands** near Central Railway Station. Our sole sole objective is to find location that is surrounded by least number of restaurants and is nearest to Central Railway Station which is the famous transport center in Europe. Since there are lots of restaurants in this location we will try to detect locations that are not already crowded with restaurants. We are also particularly interested in areas with no restaurants in vicinity. We would also prefer locations as close to Central Railway Station of Amsterdam as possible, assuming that first two conditions are met.

We will use our data science powers to generate a few most promising neighborhoods based on this criteria. Advantages of each area will then be clearly expressed so that best possible final location can be chosen by stakeholders.We will use our data science powers to generate a few most promising neighborhoods based on this criteria. Advantages of each area will then be clearly expressed so that best possible final location can be chosen by stakeholders.

Obviously, stakeholder would be very interested in accurate prediction location for

new ventures and for getting their objectives fulfilled i.e. Attract most tourists visiting Amsterdam through railway, for competitive advantage and business values.

2 Data

Based on definition of our problem, factors that will influence our decision are:

- number of existing restaurants in the neighborhood (any type of restaurant)
- distance of neighborhood from Central Railway Station of Amsterdam

We decided to use regularly spaced grid of locations, centered around Central Railway Station of Amsterdam, to define our neighborhoods.

Following data sources will be needed to extract/generate the required information:

- centers of candidate areas will be generated algorithmically and approximate addresses of centers of those areas will be obtained using **Google Maps API reverse geocoding**
- number of restaurants and their type and location in every neighborhood will be obtained using **Foursquare API**
- coordinate of Central Railway Station of Amsterdam will be obtained using **Google Maps A**
- **PI geocoding**

Steps for Foursquare API

1. Since we want to select addresses for a new restaurant, we can use Foursquare API to run a search query of food.
2. To get geographical coordinates of Amsterdam Centraal by geolocator (Nominatim).

```
address = 'Amsterdam Centraal, Netherlands'

geolocator = Nominatim(user_agent="amsterdam_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print("The geographical coordinates of Amsterdam Centraal is {}, {}.".format(latitude, longitude))
```

3. Get data in the form of json file and keep only columns include venue name and anything that is associated with location.

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

```
# keep only columns that include venue name, and anything that is associated with location
filtered_columns = ['name', 'categories'] + [col for col in dataframe.columns if col.startswith('location.')] + ['id']
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.columns]

dataframe_filtered.head(10)
```

Steps for Google Maps API reverse geocoding

In [22]:
```
google_api_key='AIzaSyB8Qd0k3UbmTiC4xoS4nGnM9ZvQYX9nVFk'
def get_address(api_key, latitude, longitude, verbose=False):
    try:
        url = 'https://maps.googleapis.com/maps/api/geocode/json?key={}&latlng={},{}'.format(api_key, latitude, longitude)
        response = requests.get(url).json()
        if verbose:
            print('Google Maps API JSON result =>', response)
        results = response['results']
        address = results[0]['formatted_address']
        return address
    except:
        return None

addr = get_address(google_api_key, centraal[0], centraal[1])
print('Reverse geocoding check')
print('-----------------------')
print('Address of [{}, {}] is: {}'.format(centraal[0], centraal[1], addr))
```

```
Reverse geocoding check
-----------------------
Address of [52.3791, 4.9003] is: Amsterdam Centraal, Stationsplein, 1012 AB Amsterdam, Netherlands
```

3 Methodology

Firstly, we can know the geographical coordinates of Amsterdam Centraal is 52.378901, 4.9005805. Using Foursquare API, we can get the total number of restaurants is 2156. To calculate the density of restaurant, hexagonal grid of cells can be created. I offset every other row, and adjust vertical row spacing so that every cell is equally distant from all its neighborhoods. Since around 4km in radius Amsterdam Centraal there are many restaurants so will try to find most optimal location for it. We would also prefer locations as close to Amsterdam Centraal as possible. We will divide area around Amsterdam Centraal into grids up to 6 km radius and try to find are with least number of restaurants in those grids. Let's create latitude & longitude coordinates for centroids of our candidate neighborhoods. We will create a grid of cells covering our area of interest which is approx. 12x12 kilometers centered around Amsterdam Centraal. Since the average number of restaurants in neighborhood is around 5.

Out[9]:

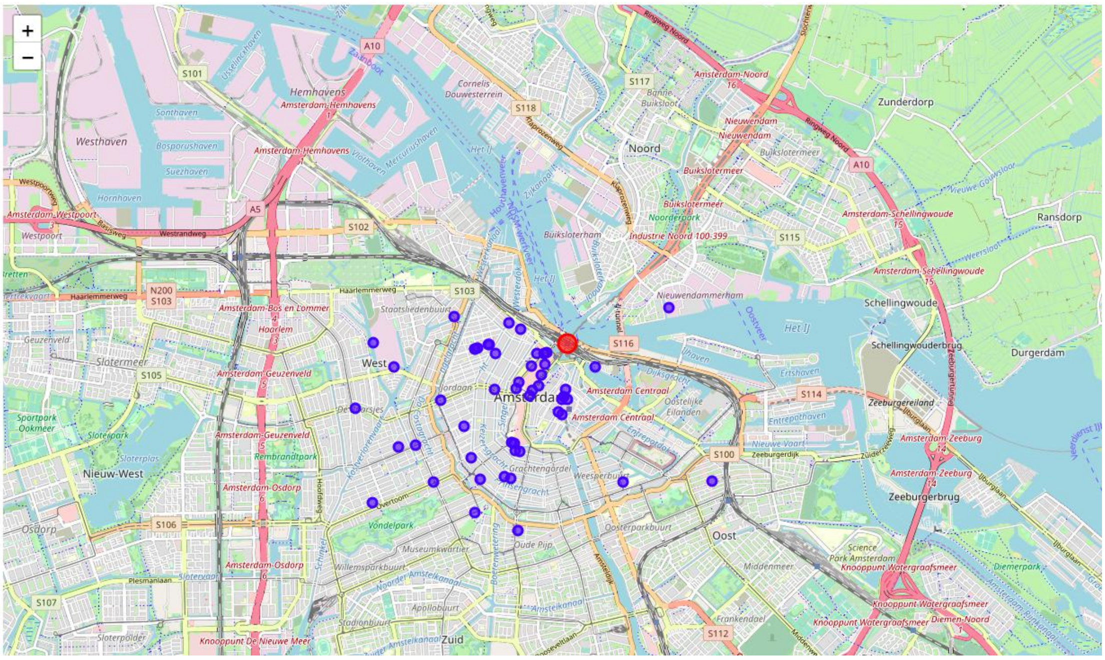

Fig 3-1. Map representation of restaurants in 6 km radius of Amsterdam Centraal.

Out[19]:

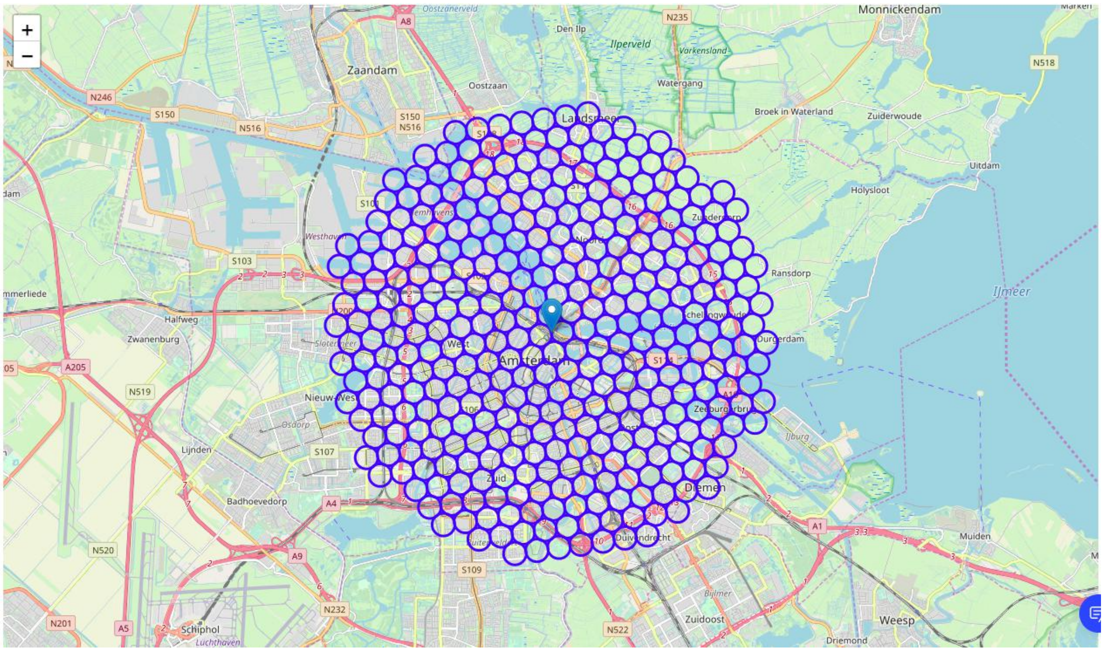

Fig 3-2. Hexagonal grid of cells centered Amsterdam Centraal.

Out[22]:

	Address	Latitude	Longitude	X	Y	Distance from center
0	NO ADDRESS	52.326262	4.885943	-188356.674341	5.845635e+06	5992.495307
1	NO ADDRESS	52.327012	4.894610	-187756.674341	5.845635e+06	5840.376700
2	NO ADDRESS	52.327761	4.903278	-187156.674341	5.845635e+06	5747.173218
3	NO ADDRESS	52.328509	4.911946	-186556.674341	5.845635e+06	5715.767665
4	NO ADDRESS	52.329257	4.920615	-185956.674341	5.845635e+06	5747.173218
5	NO ADDRESS	52.330004	4.929283	-185356.674341	5.845635e+06	5840.376700
6	NO ADDRESS	52.330750	4.937953	-184756.674341	5.845635e+06	5992.495307
7	NO ADDRESS	52.329736	4.871882	-189256.674341	5.846155e+06	5855.766389
8	NO ADDRESS	52.330486	4.880549	-188656.674341	5.846155e+06	5604.462508
9	NO ADDRESS	52.331236	4.889217	-188056.674341	5.846155e+06	5408.326913

Table 3-1. Geographical coordinates of each cells.

Now that we have our location candidates, we use Foursquare API to get information on restaurants in each neighborhood. We're interested in venues in 'food' category, but only those that are proper restaurants - coffee shops, pizza places, bakeries etc. are not direct competitors so we don't care about those. So, we will include in our list only venues that have 'restaurant' in category name.

4 Analysis

Performing some basic explanatory data analysis and derive some additional info from our raw data. First let's count the number of restaurants in every area candidate: Around 300m radius of neighborhoods.

Out[28]:

	Address	Latitude	Longitude	X	Y	Distance from center	Restaurants in area
0	NO ADDRESS	52.326262	4.885943	-188356.674341	5.845635e+06	5992.495307	7
1	NO ADDRESS	52.327012	4.894610	-187756.674341	5.845635e+06	5840.376700	1
2	NO ADDRESS	52.327761	4.903278	-187156.674341	5.845635e+06	5747.173218	0
3	NO ADDRESS	52.328509	4.911946	-186556.674341	5.845635e+06	5715.767665	1
4	NO ADDRESS	52.329257	4.920615	-185956.674341	5.845635e+06	5747.173218	0
5	NO ADDRESS	52.330004	4.929283	-185356.674341	5.845635e+06	5840.376700	3
6	NO ADDRESS	52.330750	4.937953	-184756.674341	5.845635e+06	5992.495307	3
7	NO ADDRESS	52.329736	4.871882	-189256.674341	5.846155e+06	5855.766389	4
8	NO ADDRESS	52.330486	4.880549	-188656.674341	5.846155e+06	5604.462508	13
9	NO ADDRESS	52.331236	4.889217	-188056.674341	5.846155e+06	5408.326913	6

Table 4-1. Number of restaurants in each neighborhood.

Preferable locations found for setting up restaurant near Amsterdam Centraal which are nearest and least surrounded by other restaurants.

Out[32]:

	Address	Latitude	Longitude	X	Y	Distance from center	Restaurants in area
0	NO ADDRESS	52.387173	4.885160	-187456.674341	5.852390e+06	1374.772708	0
1	NO ADDRESS	52.387923	4.893839	-186856.674341	5.852390e+06	1081.665383	0
2	NO ADDRESS	52.388673	4.902519	-186256.674341	5.852390e+06	1081.665383	0
3	NO ADDRESS	52.389422	4.911199	-185656.674341	5.852390e+06	1374.772708	0

Table 4-2. Neighborhood candidates with limits: distance not larger than 1.5 km and no restaurant.

Out[33]:

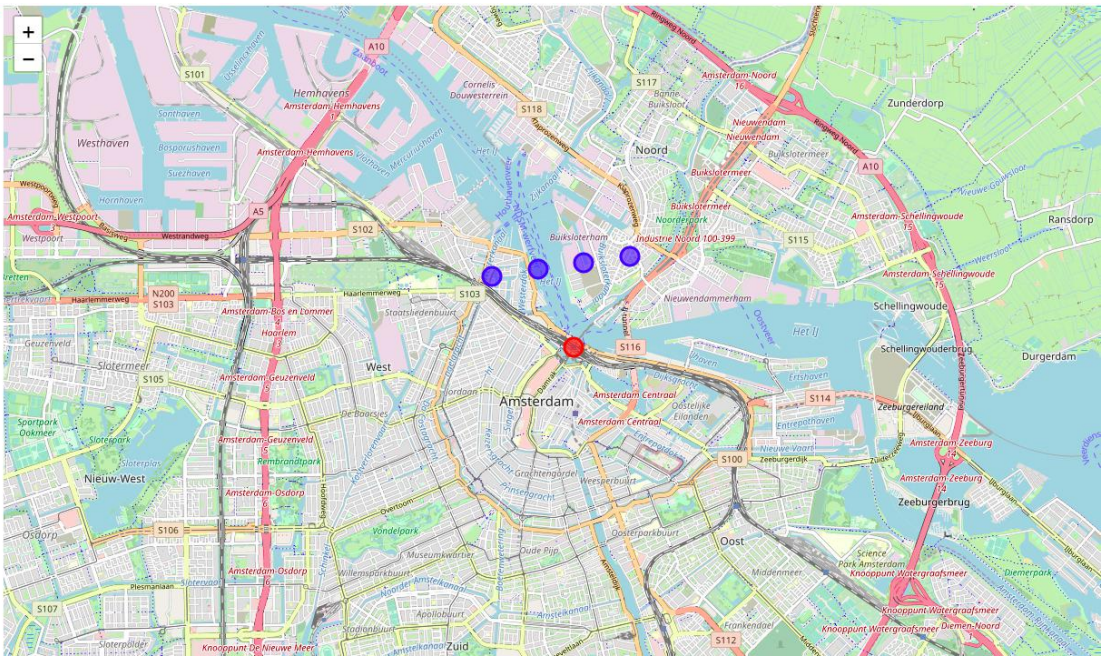

Fig 4-1. Map of Amsterdam with 4 neighborhood candidates

5 Results

Our analysis shows that there are many restaurants near Amsterdam Centraal and we were mainly concerned about finding best suitable location that is not already crowded with restaurants and is nearest to Amsterdam Centraal. Total number (~2156 in our initial area of interest which was 12x12km around Amsterdam Centraal), there are large pockets of low restaurant density fairly close to Amsterdam Centraal. Highest concentration of restaurants was detected north from Amsterdam Centraal, so we found out the neighborhoods with no restaurants and distance from Amsterdam Centraal less than 1.5 km and found out 4 such suitable neighborhoods.

Result of all these zones containing largest number of potential new restaurant location. Purpose of this analysis was to provide information on areas close to Amsterdam Centraal. It is entirely possible that there is a very good reason for small number of restaurants in any of those areas, reasons which would make them unsuitable for a new restaurant regardless of lack of competition in the area. Recommended zones should therefore be considered only as a starting point for more detailed analysis which could

eventually result in location which has not only no nearby competition but also other factors taken into account and all other relevant conditions met.

6 Conclusion

Purpose of this project was to identify Amsterdam neighborhoods close to Amsterdam Centraal with low number of restaurants in order to aid stakeholders in narrowing down the search for optimal location for a new restaurant. By calculating restaurant density distribution from Foursquare data we have first identified coordinates and neighborhoods that justify further analysis and then generated extensive collection of locations which satisfy some basic requirements regarding existing nearby restaurants. Final decision on optimal restaurant location will be made by stakeholders based on specific characteristics of neighborhoods and locations in every recommended zone, taking into consideration additional factors like attractiveness of each location (proximity to park or water), levels of noise / proximity to major roads, real estate availability, prices, social and economic dynamics of every neighborhood etc.