My Capstone Project Restaurant Business in Istanbul

JUNE 5

Restaurant Business In Istanbul

Introduction

Istanbul is the largest city in Turkey, constituting the country's economic, cultural and historical center. The population of almost 15 million lives in Istanbul's vast area of 5,343 square kilometers. Istanbul is situated at the Bosphorus – one of the world's busiest waterways – in northwestern Turkey, between the Sea of Marmara and the Black Sea. Its commercial and historical center lies in Europe, while a third of its population lives on the Asian side.

Foreign investors in Turkey often choose to open a restaurant or any other business establishment that can be linked to tourism. Turkey is an increasingly popular tourist destination and a large number of visitors will always represent a good clientele for a restaurant. Much as the country, the Turkish cuisine is a blend between different cultures - the Asian, Caucasian, Middle Eastern, Mediterranean and Balkan cuisines have shaped and influenced the Turkish food.

Business Problem

In recent months and years, the restaurant industry has seen its share of bankruptcies, including, but not limited to, filings from Bertucci's, Logan's Roadhouse, Real Mex Restaurants, fast casual Noon Mediterranean, Romano's Macaroni Grill, Scotty's Brewhouse, Ruby's Diner, and Iron Chef Jose Garces. Bankruptcies aren't anything new in the industry, but they've been proliferating at a rapid pace, compared to more sporadic declarations 10 years ago.

If you are an investor and decided to open a new restaurant, you should know that location is one of the most important decisions you need to make. In this project, I will analyze restaurant locations and try to provide some valuable recommendation to the foreign or internal investors who want to enter restaurant business in Istanbul.

Data

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

- Number of existing restaurants in every Borough from Foursquare API

- Latitude and Longitude values of 39 boroughes in Istanbul from Github (*https://gist.github.com/ismailbaskin/2492196*)

Methodology

In this project we will direct our efforts on detecting areas of Istanbul that have low restaurant density, particularly those with low number of Italian restaurants. We will limit our analysis to area ~6km around city center.

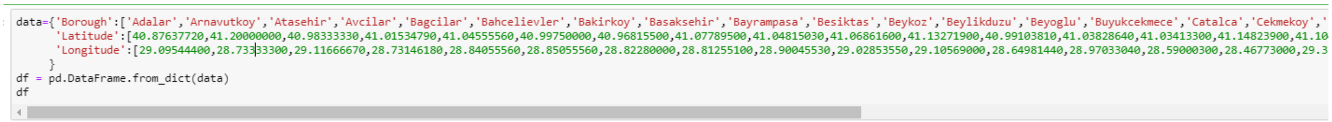
In first step we have collected the location data of 39 Boroughes in Istanbul from the Github. Since the format of the data that I found at that website was not ready to use, I have prepared the location dataframe manualy by using the original data. I have also identified Italian restaurants (according to Foursquare categorization).

Second step in our analysis will be calculation and exploration of 'restaurant density' across different areas of Istanbul

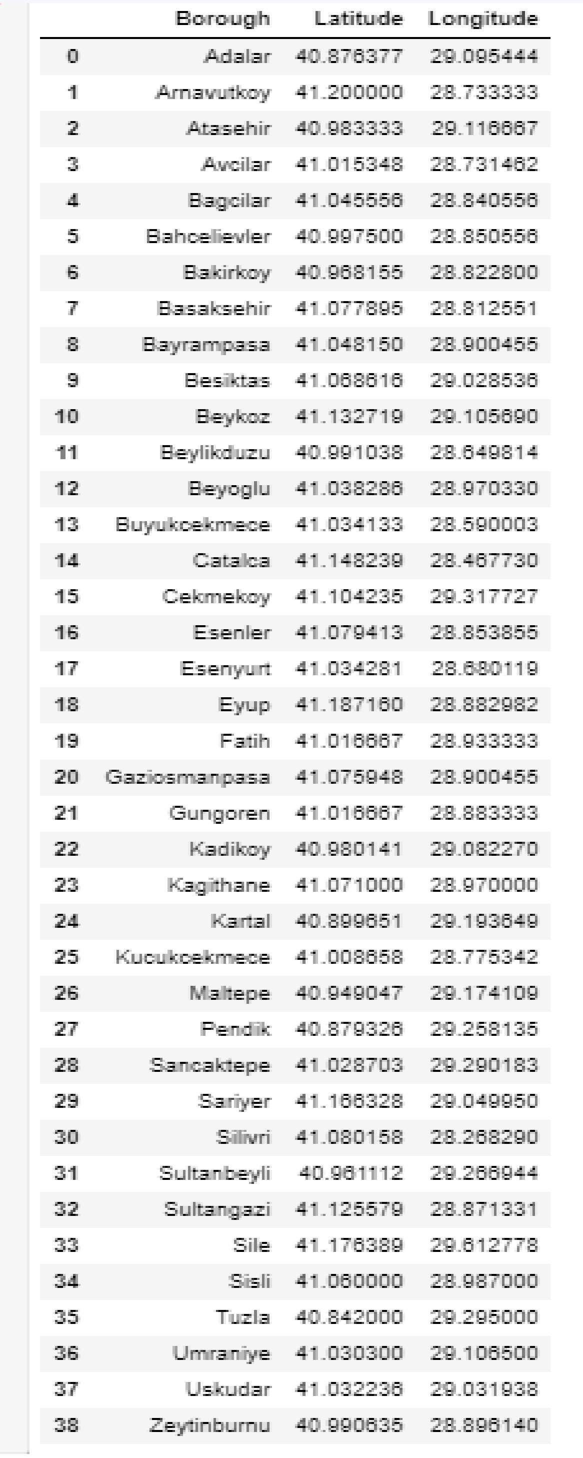
In third and final step we will focus on most promising areas and within those create clusters of locations that meet some basic requirements established in discussion with stakeholders: we will take into consideration locations with no more than two restaurants in radius of 250 meters, and we want locations without Italian restaurants in radius of 400 meters. We will present map of all such locations but also create clusters (using k-means clustering) of those locations to identify general zones / neighborhoods / addresses which should be a starting point for final 'street level' exploration and search for optimal venue location by stakeholders.

Location Data

Firstly, I got Location information of 39 boroughs of Istanbul from the github (https://gist.github.com/ismailbaskin/2492196) and converted them to a dataframe



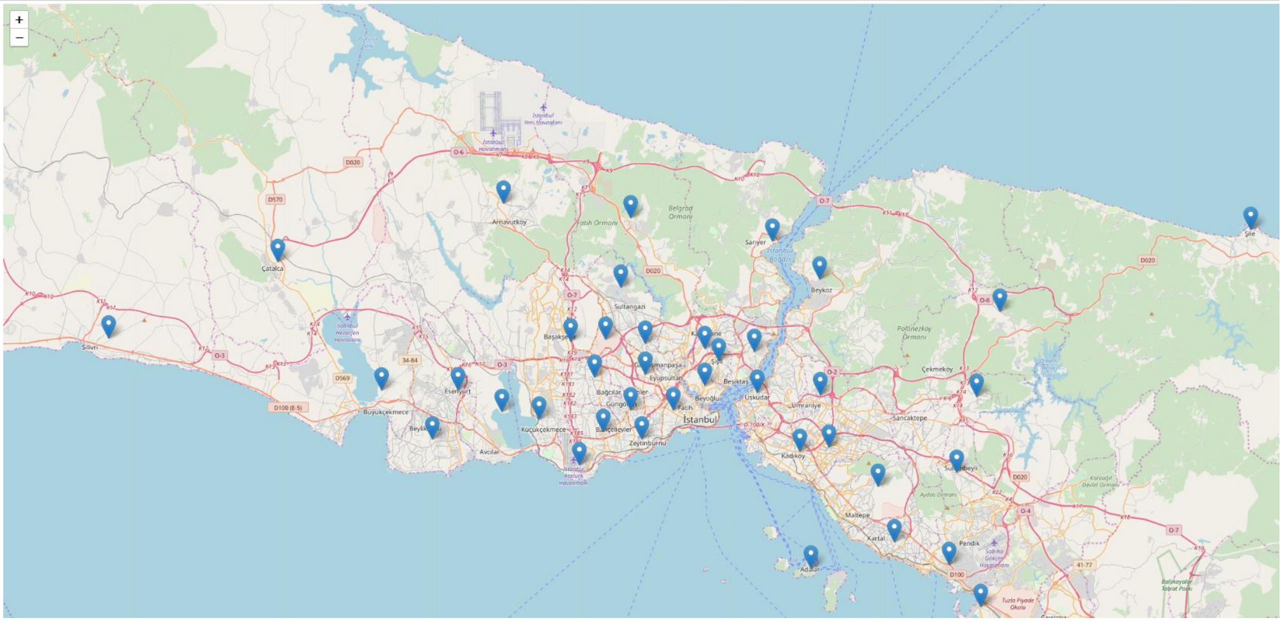
```
data={'Borough':['Adalar','Arnavutkoy','Atasehir','Avcilar','Bagcilar','Bahcelievler','Bakirkoy','Basaksehir','Bayrampasa','Besiktas','Beykoz','Beylikduzu','Beyoglu','Buyukcekmece','Catalca','Cekmekoy','
      'Latitude':[40.87637720,41.20000000,40.98333330,41.01534790,41.04555560,40.99750000,40.96815500,41.07789500,41.04815030,41.06861600,41.13271900,40.99103810,41.03828640,41.03413300,41.14823900,41.10
      'Longitude':[29.09544400,28.73333300,29.11666670,28.73146180,28.84055560,28.85055560,28.82280000,28.81255100,28.90045530,29.02853550,29.10569000,28.64981440,28.97033040,28.59000300,28.46773000,29.3
     }
df = pd.DataFrame.from_dict(data)
df
```



	Borough	Latitude	Longitude
0	Adalar	40.876377	29.095444
1	Arnavutkoy	41.200000	28.733333
2	Atasehir	40.983333	29.116667
3	Avcilar	41.015348	28.731462
4	Bagcilar	41.045556	28.840556
5	Bahcelievler	40.997500	28.850556
6	Bakirkoy	40.968155	28.822800
7	Basaksehir	41.077895	28.812551
8	Bayrampasa	41.048150	28.900455
9	Besiktas	41.068616	29.028536
10	Beykoz	41.132719	29.105690
11	Beylikduzu	40.991038	28.649814
12	Beyoglu	41.038286	28.970330
13	Buyukcekmece	41.034133	28.590003
14	Catalca	41.148239	28.467730
15	Cekmekoy	41.104235	29.317727
16	Esenler	41.079413	28.853855
17	Esenyurt	41.034281	28.680119
18	Eyup	41.187160	28.882982
19	Fatih	41.016667	28.933333
20	Gaziosmanpasa	41.075948	28.900455
21	Gungoren	41.016667	28.883333
22	Kadikoy	40.980141	29.082270
23	Kagithane	41.071000	28.970000
24	Kartal	40.899651	29.193649
25	Kucukcekmece	41.008658	28.775342
26	Maltepe	40.949047	29.174109
27	Pendik	40.879326	29.258135
28	Sancaktepe	41.028703	29.290183
29	Sariyer	41.166328	29.049950
30	Silivri	41.080158	28.268290
31	Sultanbeyli	40.961112	29.266944
32	Sultangazi	41.125579	28.871331
33	Sile	41.176389	29.612778
34	Sisli	41.060000	28.987000
35	Tuzla	40.842000	29.295000
36	Umraniye	41.030300	29.106500
37	Uskudar	41.032236	29.031938
38	Zeytinburnu	40.990635	28.896140

And I visualized them

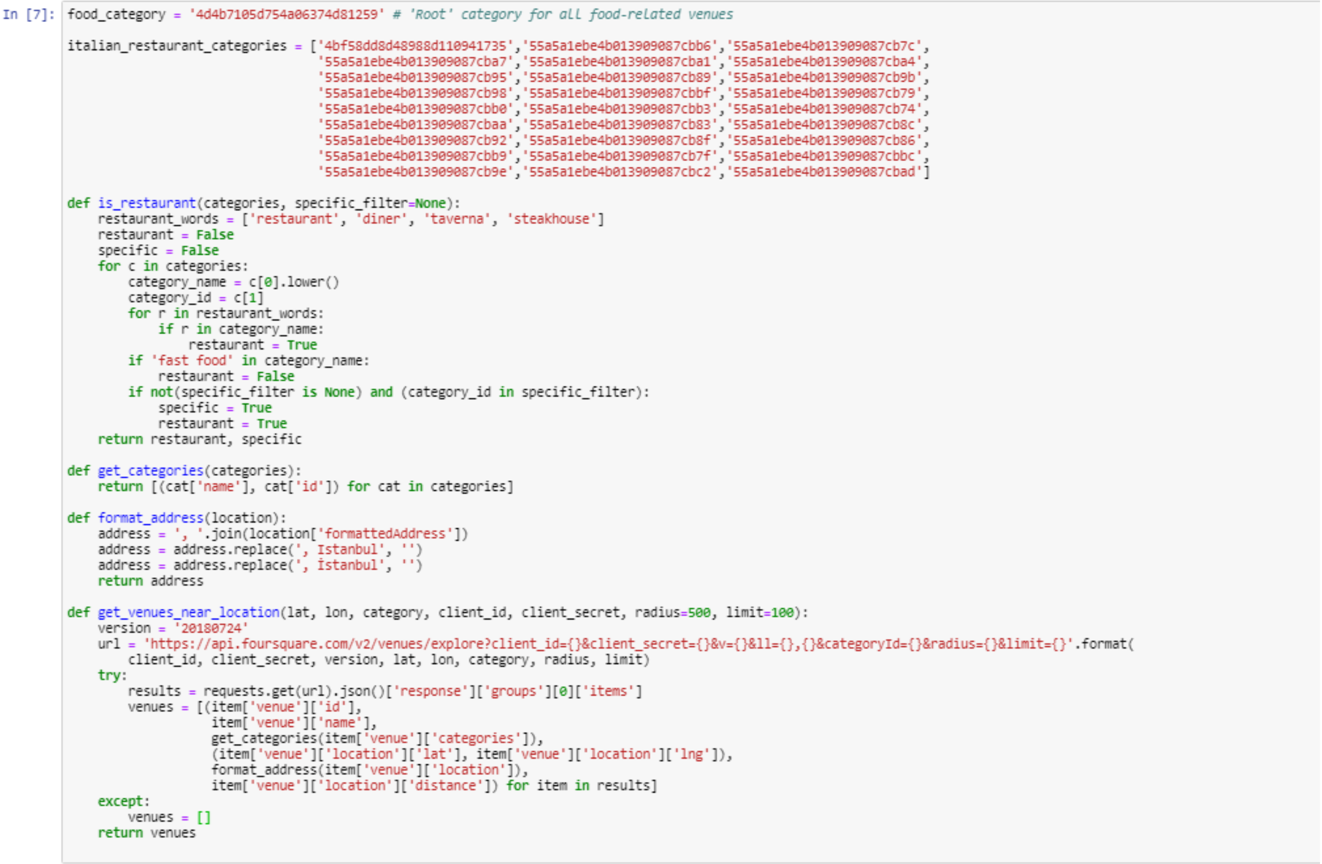
Out[24]:



Foursquare Data

Now that I have location candidates. Let's use Foursquare API to get info on restaurants in each neighborhood.

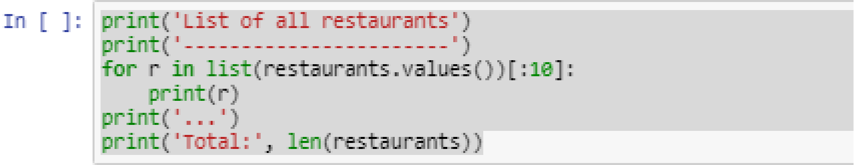
We're interested in venues in 'food' category, but only those that are proper restaurants - coffe shops, pizza places, bakeries etc. are not direct competitors so we don't care about those. So we will include in out list only venues that have 'restaurant' in category name, and we'll make sure to detect and include all the subcategories of specific 'Italian restaurant' category, as we need info on Italian restaurants in the neighborhood.



In [7]:

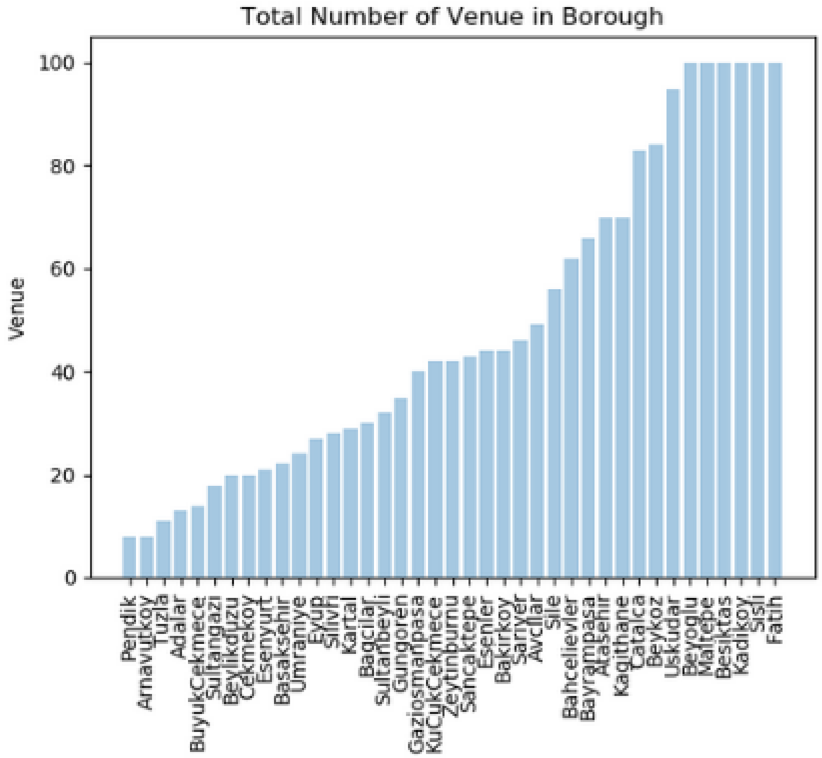
```
food_category = '4d4b7105d754a06374d81259' # 'Root' category for all food-related venues

italian_restaurant_categories = ['4bf58dd8d48988d110941735','55a5a1ebe4b013909087cbb6','55a5a1ebe4b013909087cb7c',
                                 '55a5a1ebe4b013909087cba7','55a5a1ebe4b013909087cba1','55a5a1ebe4b013909087cba4',
                                 '55a5a1ebe4b013909087cb95','55a5a1ebe4b013909087cb89','55a5a1ebe4b013909087cb9b',
                                 '55a5a1ebe4b013909087cb98','55a5a1ebe4b013909087cbbf','55a5a1ebe4b013909087cb79',
                                 '55a5a1ebe4b013909087cbb0','55a5a1ebe4b013909087cbb3','55a5a1ebe4b013909087cb74',
                                 '55a5a1ebe4b013909087cbaa','55a5a1ebe4b013909087cb83','55a5a1ebe4b013909087cb8c',
                                 '55a5a1ebe4b013909087cb92','55a5a1ebe4b013909087cb8f','55a5a1ebe4b013909087cb86',
                                 '55a5a1ebe4b013909087cbb9','55a5a1ebe4b013909087cb7f','55a5a1ebe4b013909087cbbc',
                                 '55a5a1ebe4b013909087cb9e','55a5a1ebe4b013909087cbc2','55a5a1ebe4b013909087cbad']

def is_restaurant(categories, specific_filter=None):
    restaurant_words = ['restaurant', 'diner', 'taverna', 'steakhouse']
    restaurant = False
    specific = False
    for c in categories:
        category_name = c[0].lower()
        category_id = c[1]
        for r in restaurant_words:
            if r in category_name:
                restaurant = True
        if 'fast food' in category_name:
            restaurant = False
        if not(specific_filter is None) and (category_id in specific_filter):
            specific = True
            restaurant = True
    return restaurant, specific

def get_categories(categories):
    return [(cat['name'], cat['id']) for cat in categories]

def format_address(location):
    address = ', '.join(location['formattedAddress'])
    address = address.replace(', Istanbul', '')
    address = address.replace(', İstanbul', '')
    return address

def get_venues_near_location(lat, lon, category, client_id, client_secret, radius=500, limit=100):
    version = '20180724'
    url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&v={}&ll={},{}&categoryId={}&radius={}&limit={}'.format(
        client_id, client_secret, version, lat, lon, category, radius, limit)
    try:
        results = requests.get(url).json()['response']['groups'][0]['items']
        venues = [(item['venue']['id'],
                   item['venue']['name'],
                   get_categories(item['venue']['categories']),
                   (item['venue']['location']['lat'], item['venue']['location']['lng']),
                   format_address(item['venue']['location']),
                   item['venue']['location']['distance']) for item in results]
    except:
        venues = []
    return venues
```

Let's look at restaurant statistics



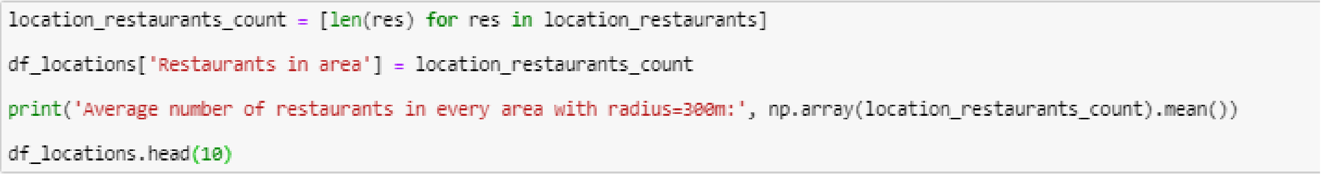
In [ ]:

```
print('List of all restaurants')
print('-----------------------')
for r in list(restaurants.values())[:10]:
    print(r)
print('...')
print('Total:', len(restaurants))
```



Analysis

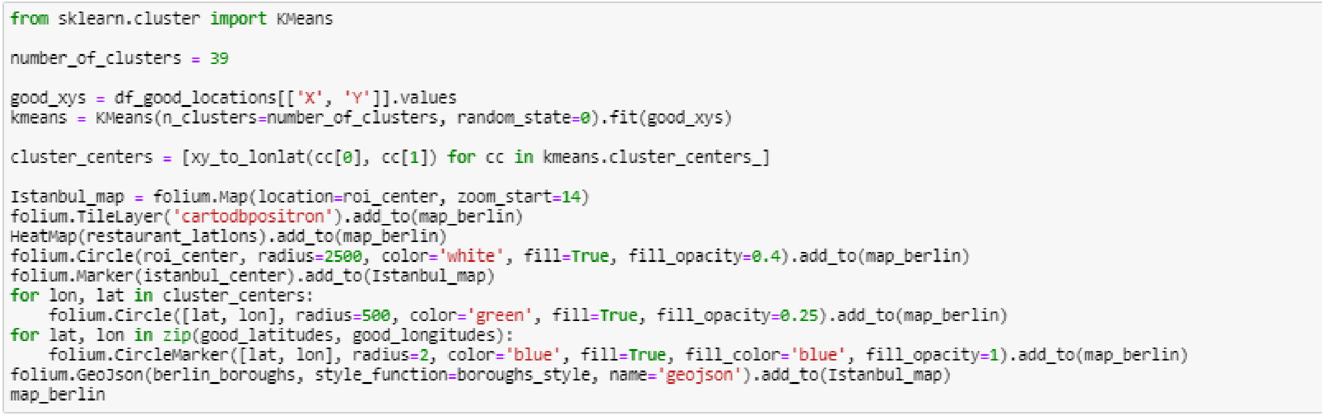
I have performed some basic explanatory data analysis and derive some additional info from my raw data. First let's count the number of restaurants in every area candidate:



```
location_restaurants_count = [len(res) for res in location_restaurants]

df_locations['Restaurants in area'] = location_restaurants_count

print('Average number of restaurants in every area with radius=300m:', np.array(location_restaurants_count).mean())

df_locations.head(10)
```

What we have now is a clear indication of zones with low number of restaurants in vicinity, and *no* Italian restaurants at all nearby.

Let us now **cluster** those locations to create **centers of zones containing good locations**. Those zones, their centers and addresses will be the final result of our analysis.



```
from sklearn.cluster import KMeans

number_of_clusters = 39

good_xys = df_good_locations[['X', 'Y']].values
kmeans = KMeans(n_clusters=number_of_clusters, random_state=0).fit(good_xys)

cluster_centers = [xy_to_lonlat(cc[0], cc[1]) for cc in kmeans.cluster_centers_]

Istanbul_map = folium.Map(location=roi_center, zoom_start=14)
folium.TileLayer('cartodbpositron').add_to(map_berlin)
HeatMap(restaurant_latlons).add_to(map_berlin)
folium.Circle(roi_center, radius=2500, color='white', fill=True, fill_opacity=0.4).add_to(map_berlin)
folium.Marker(istanbul_center).add_to(Istanbul_map)
for lon, lat in cluster_centers:
    folium.Circle([lat, lon], radius=500, color='green', fill=True, fill_opacity=0.25).add_to(map_berlin)
for lat, lon in zip(good_latitudes, good_longitudes):
    folium.CircleMarker([lat, lon], radius=2, color='blue', fill=True, fill_color='blue', fill_opacity=1).add_to(map_berlin)
folium.GeoJson(berlin_boroughs, style_function=boroughs_style, name='geojson').add_to(Istanbul_map)
map_berlin
```

Our clusters represent groupings of most of the candidate locations and cluster centers are placed nicely in the middle of the zones 'rich' with location candidates.

Addresses of those cluster centers will be a good starting point for exploring the neighborhoods to find the best possible location based on neighborhood specifics.

Results and Discussions

Our analysis shows that although there is a great number of restaurants in Istanbul, there are pockets of low restaurant density fairly close to city center. Highest concentration of restaurants was detected near Bosphorous, so we should focused our attention to those areas . Boroughs which offer a combination of popularity among tourists, closeness to city center, strong socio-economic dynamics a number of pockets of low restaurant density will give us better conclusions.

After directing our attention to this more narrow area of interest I first created a dense grid of location candidates (spaced 100m appart); those locations were then filtered so that those with more than two restaurants in radius of 250m and those with an Italian restaurant closer than 400m were removed.

Those location candidates were then clustered to create zones of interest which contain greatest number of location candidates. Addresses of centers of those zones were also generated using reverse geocoding to be used as markers/starting points for more detailed local analysis based on other factors.

Result of all this is 39 boroughes containing largest number of potential new restaurant locations based on number of and distance to existing venues - both restaurants in general and Italian restaurants particularly. This, of course, does not imply that those zones are actually optimal locations for a new restaurant! Purpose of this analysis was to only provide info on areas close to the Bosphorous, but not crowded with existing restaurants (particularly Italian) - 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.

Conlusion

Purpose of this project was to identify Istanbul areas close to center with low number of restaurants (particularly Italian restaurants) in order to aid stakeholders in narrowing down the search for optimal location for a new Italian restaurant. By calculating restaurant density distribution from Foursquare data we have first identified general boroughs that justify further analysis, and then generated extensive collection of locations which satisfy some basic requirements regarding existing nearby restaurants. Clustering of those locations was then performed in order to create major zones of interest (containing greatest number of potential locations) and addresses of those zone centers were created to be used as starting points for final exploration by stakeholders.

Final decission 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 , levels of noise / proximity to major roads, real estate availability, prices, social and economic dynamics of every neighborhood etc.