

Restaurants in Porto District - Portugal

Objective

I want to open a new Restaurant in Porto District with the district code in Portugal of 4. So I want to decide which is the best municipality place to open the new restaurant. My criteria is based on the amount of restaurants in each municipality vs. population density in each municipality.

Data

I collected the data from the PORDATA (www.pordata.pt) website in Portugal and using Pandas I was able to select all the municipalities of Porto District. Following dataframe:

```
In [81]: #creates a source for the Beautiful Soup
df = pd.read_excel(r'pordata.xlsx') # Taken from www.pordata.pt
df = df[['Region', 'Municipality', 'Density']]
dfporto = df[df['Region'] == 'Porto']
dfporto.head(10)
```

	Region	Municipality	Density
24	Porto	Arouca	63.7
25	Porto	Espinho	1401.2
26	Porto	Gondomar	1255.5
27	Porto	Maia	1654.8
28	Porto	Matosinhos	2788.6
29	Porto	Oliveira de Azeméis	410.8
30	Porto	Paredes	549.2
31	Porto	Porto	5189.2
32	Porto	Póvoa de Varzim	759.6
33	Porto	Santa Maria da Feira	641.9

After that I need the Postal Code for each Municipality in order to use Foursquare as well as the Folium package. The following code:

```
In [82]: # All districts and municipalities of Portugal
code_data = pd.read_csv("concelhos.csv")["nome_concelho", "cod_distrito", "cod_concelho"]
code_data.columns = ['Municipality', 'DCode', 'MCode']
code_data.head(10)
```

	Municipality	DCode	MCode
0	Castelo de Paiva	1	6
1	Espinho	1	7
2	Estarreja	1	8
3	Santa Maria da Feira	1	9
4	Ílhavo	1	10
5	Mealhada	1	11
6	Murtosa	1	12
7	Oliveira de Azeméis	1	13
8	Oliveira do Bairro	1	14
9	Ovar	1	15

The raw data was taken from http://centralldedados.pt/codigos_postais/.

Later on I added the Latitude and Longitude using the following code:

```
In [85]: # Read latitude and longitude
coord_data = pd.read_csv("lat_long_f.csv", sep = ';', decimal=',')[[
coord_data.columns = ['Region', 'Municipality', 'CPCode', 'Latitude'
coord_porto = coord_data[coord_data['Region'] == 'Porto']
coord_porto.head()
```

Out[85]:

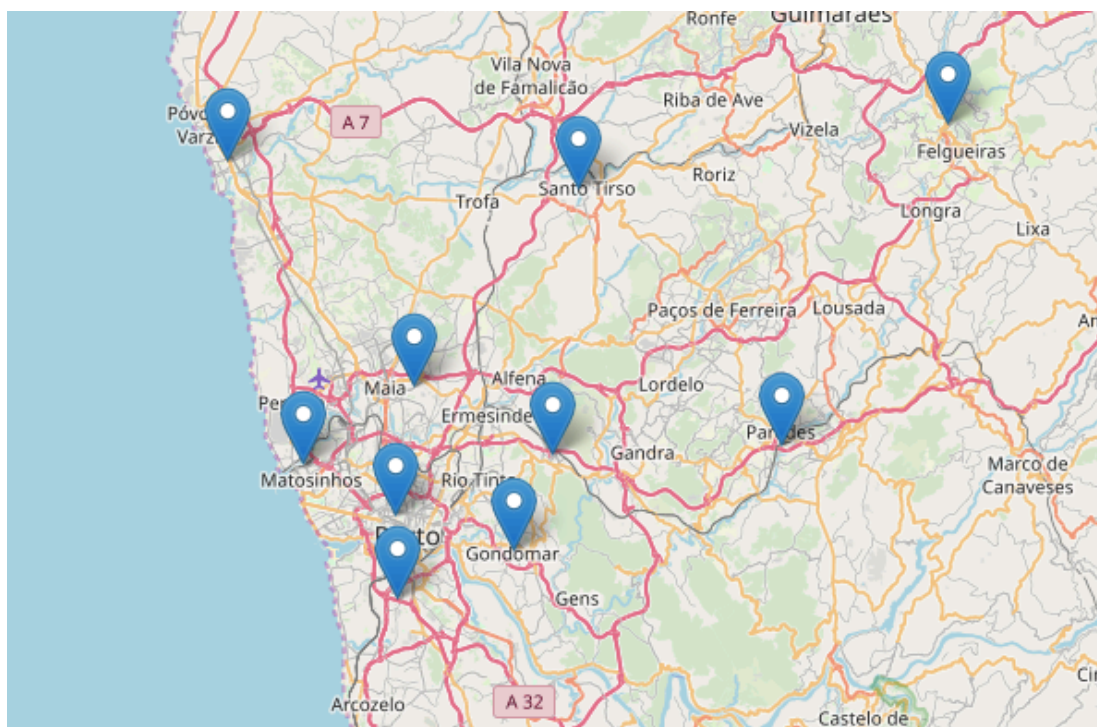
	Region	Municipality	CPCode	Latitude	Longitude
183068	Porto	Cruzeiro	4615	41.322937	-8.144780
183069	Porto	Carvalheiras	4615	41.153729	-8.606222
183070	Porto	Campo	4615	41.322937	-8.144780
183071	Porto	Aboim	4600	41.330636	-8.071072
183072	Porto	Amarante	4600	41.262439	-8.082573

After clean and join all the needed dataframes I achieved the final Dataframe that allow me to plot using Folium:

Out[86]:

	Municipality	MCode	Latitude	Longitude
0	Gondomar	4	41.139991	-8.532800
1	Maia	6	41.232430	-8.606012
2	Matosinhos	8	41.188501	-8.689159
3	Paredes	10	41.198988	-8.333568
4	Porto	12	41.160364	-8.619520
5	Santo Tirso	14	41.342284	-8.484229
6	Trofa	18	41.379333	-8.209239
7	Valongo	15	41.194495	-8.503505
8	Vila Nova de Gaia	17	41.113921	-8.619381
9	Vila do Conde	16	41.358067	-8.744970

Folium Map:



Foursquare

And finally I call Foursquare, using the code:

```
In [90]: CLIENT_ID = 'ZC4GEQ0R3WUBDVREK3PBHNT003HVJFKF1TFYQPIQEINCPREG'
CLIENT_SECRET = 'EWPJRXRGNTDSJKXCLT2XAA4SLIJLC0IERAF4YHNM3MKFYRC'
VERSION = '20180604'

url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    neighborhood_latitude,
    neighborhood_longitude,
    radius,
    LIMIT)
results = requests.get(url).json()

venues = results['response']['groups'][0]['items']
nearby_venues = json_normalize(venues)

# 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(20)
```

Foursquare Results:

	name	categories	lat	lng
0	BeerDrive Caffè	[[{'id': '4bf58dd8d48988d116941735', 'name': 'B...	41.140675	-8.530712
1	Confeitaria Avenida	[[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...	41.138011	-8.532795
2	O Monte	[[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...	41.142131	-8.536163
3	Basho Sushi House	[[{'id': '4bf58dd8d48988d1d2941735', 'name': 'S...	41.143862	-8.535312
4	Monte Crasto	[[{'id': '4bf58dd8d48988d165941735', 'name': 'S...	41.142088	-8.536096
5	Tip-Off	[[{'id': '4bf58dd8d48988d116941735', 'name': 'B...	41.140522	-8.531692
6	Madureiras Souto	[[{'id': '4bf58dd8d48988d1ce941735', 'name': 'S...	41.135916	-8.530946
7	Restaurante O Brasinhas	[[{'id': '4bf58dd8d48988d1c4941735', 'name': 'R...	41.137065	-8.531881
8	Cantinho Avenida	[[{'id': '4bf58dd8d48988d1e0931735', 'name': 'C...	41.136619	-8.533706
9	Marpão	[[{'id': '4bf58dd8d48988d16a941735', 'name': 'B...	41.136524	-8.531875

Conclusion

After the Capstone I was able to implement the Python knowledge that I acquire during the course as well as Foursquare API. I lacked time to finish the data analysis required to do a good choice for the best place for my restaurant.