1 - Introduction

Business Problem & Interested Stakeholders

Where to open a restaurant in Lisbon?

This is the question that we will help to answer, understanding the actual configuration and possibilities using available data. Lisbon is the capital city of Portugal and one of the most beautiful and charismatic cities of Europe. It has a vibrant life with lots of people coming from suburbs to work and many tourists. It is also well known by having a great climate and of course the great food. Those who try to open a new restaurant face a great competition and challenge. All the help they can have is for sure wellcome.

This kind of knowledge can have a lot of interested stakeholders:

- Someone who wants to open a new restaurant and know the competition.
- Local authorities who want to better develop their municipalities.
- Real estate agents who want to get additional knowledge for their clients.
- Other intereste stakeholdes like students, foodies or just curious people.

2 - Data

On this section we will gather all the information needed to answer the question of this problem.

Since the new administrative reordering on 2013, Lisbon is divided into 24 borgoughs (freguesias in Portuguese). We will use this administrative division for our sutdy.

For this we need the following data:

- Coordinates to search from using the Foursquare API

> Using the wikipedia page in which we can get the borough's coordinates: https://pt.wikipedia.org/wiki/Lisboa (https://pt.wikipedia.org/wiki/Lisboa)

- Boroughs shapes to match with the venues location

> Define in which borough the venue belongs, this information does not come from Foursquare API
>
> https://geodados-cml.hub.arcgis.com/datasets/freguesias-2012?geometry=-9.315%2C38.702%2C-8.988%2C38.749&page=2&selectedAttribute=Shape__Length (https://geodados-cml.hub.arcgis.com/datasets/freguesias-2012?geometry=-9.315%2C38.702%2C-8.988%2C38.749&page=2&selectedAttribute=Shape__Length)

- Aditional information by borough (population, population density, visiting population)

> It will be helpful for the final anaysis, use also wikipedia page: https://pt.wikipedia.org/wiki/Lista_de_freguesias_de_Lisboa (https://pt.wikipedia.org/wiki/Lista_de_freguesias_de_Lisboa)

- Venues information

> Use the Foursquare API to get explore the venues location and other relevant information

Data Clean

And we get our clean data set with all the borough information.

In [7]:
```
df_lisbon = pd.merge(df_new, df_json, on="Borough")
df_lisbon.head()
df_lisbon['Area(km2)'] = df_lisbon['Area(km2)'].astype(float)

df_lisbon["Density(per km2)"] = round(df_lisbon["Population"] / df_lisbon["Area(km2)"],1)


column_names = ['Borough', 'Population', 'Area(km2)', 'Density(per km2)', 'Lat', 'Lon','Geometry', 'Centroid_Lat', 'Centroid_Lon']

df_lisbon = df_lisbon.reindex(columns=column_names)
df_lisbon
```

Out[7]:

	Borough	Population	Area(km2)	Density(per km2)	Lat	Lon	Geometry	Centroid_Lat	Centroid_Lon
0	Ajuda	15617	2.88	5422.6	38.707500	-9.198333	POLYGON ((-9.19303 38.71619, -9.19295 38.71368...	38.712179	-9.198651
1	Alcântara	13943	5.07	2750.1	38.706389	-9.174167	POLYGON ((-9.17812 38.72376, -9.17800 38.72373...	38.709598	-9.183388
2	Alvalade	31813	5.34	5957.5	38.746944	-9.136111	POLYGON ((-9.12857 38.76302, -9.12859 38.76279...	38.753888	-9.146497
3	Areeiro	20131	1.72	11704.1	38.740278	-9.128056	POLYGON ((-9.12374 38.73866, -9.12477 38.73829...	38.741380	-9.133525
4	Arroios	31653	2.13	14860.6	38.728889	-9.138889	POLYGON ((-9.13306 38.73019, -9.13311 38.73008...	38.727657	-9.137940
5	Avenidas Novas	21625	2.99	7232.4	38.738889	-9.145833	POLYGON ((-9.14758 38.74494, -9.14752 38.74464...	38.737465	-9.152328
6	Beato	12737	2.48	5135.9	38.734722	-9.105833	POLYGON ((-9.12374 38.73866, -9.12323 38.73894...	38.731054	-9.110093
7	Belém	16528	10.43	1584.7	38.700000	-9.200000	POLYGON ((-9.20591 38.71526, -9.20620 38.71482...	38.696207	-9.213136
8	Benfica	36985	8.02	4611.6	38.751111	-9.202222	POLYGON ((-9.19402 38.75724, -9.19439 38.75657...	38.737746	-9.196363
9	Campo de Ourique	22120	1.65	13406.1	38.715278	-9.166944	POLYGON ((-9.15764 38.72246, -9.15735 38.72230...	38.718542	-9.165753
10	Campolide	15460	2.77	5581.2	38.726389	-9.163333	POLYGON ((-9.16667 38.73939, -9.16642 38.73922...	38.731158	-9.167270
11	Carnide	19218	3.69	5208.1	38.760833	-9.183611	POLYGON ((-9.17522 38.78027, -9.17522 38.78025...	38.765200	-9.186734
12	Estrela	20128	4.60	4375.7	38.713333	-9.160000	POLYGON ((-9.16081 38.71555, -9.16075 38.71554...	38.703768	-9.163458
13	Lumiar	45605	6.57	6941.4	38.765278	-9.158611	POLYGON ((-9.16682 38.78379, -9.16624 38.78332...	38.769795	-9.163491
14	Marvila	37793	7.12	5308.0	38.745278	-9.104167	POLYGON ((-9.12567 38.76252, -9.12532 38.76243...	38.747934	-9.110954
15	Misericórdia	13044	2.19	5956.2	38.711389	-9.148056	POLYGON ((-9.14552 38.71556, -9.14510 38.71549...	38.705446	-9.146674
16	Olivais	33788	8.09	4176.5	38.773611	-9.117500	POLYGON ((-9.13429 38.78492, -9.13393 38.78485...	38.771035	-9.124650
17	Parque das Nações	21025	5.44	3864.9	38.768056	-9.093889	POLYGON ((-9.09570 38.79676, -9.09569 38.79676...	38.774806	-9.095478
18	Penha de França	27967	2.71	10319.9	38.730000	-9.131667	POLYGON ((-9.12640 38.73774, -9.12602 38.73707...	38.726432	-9.120539
19	Santa Clara	22480	3.36	6690.5	38.785278	-9.145000	POLYGON ((-9.14846 38.79474, -9.14832 38.79471...	38.785163	-9.152169
20	Santa Maria Maior	12822	3.01	4259.8	38.712778	-9.135556	POLYGON ((-9.13504 38.71978, -9.13499 38.71978...	38.708311	-9.132404
21	Santo António	11836	1.49	7943.6	38.724167	-9.145000	POLYGON ((-9.14492 38.72825, -9.14405 38.72731...	38.721345	-9.148808
22	São Domingos de Benfica	33043	4.29	7702.3	38.743611	-9.170000	POLYGON ((-9.16970 38.75732, -9.16978 38.75707...	38.746451	-9.176373
23	São Vicente	15339	1.99	7708.0	38.719444	-9.126389	POLYGON ((-9.12715 38.72365, -9.12646 38.72317...	38.716395	-9.121721

3 - Methodology

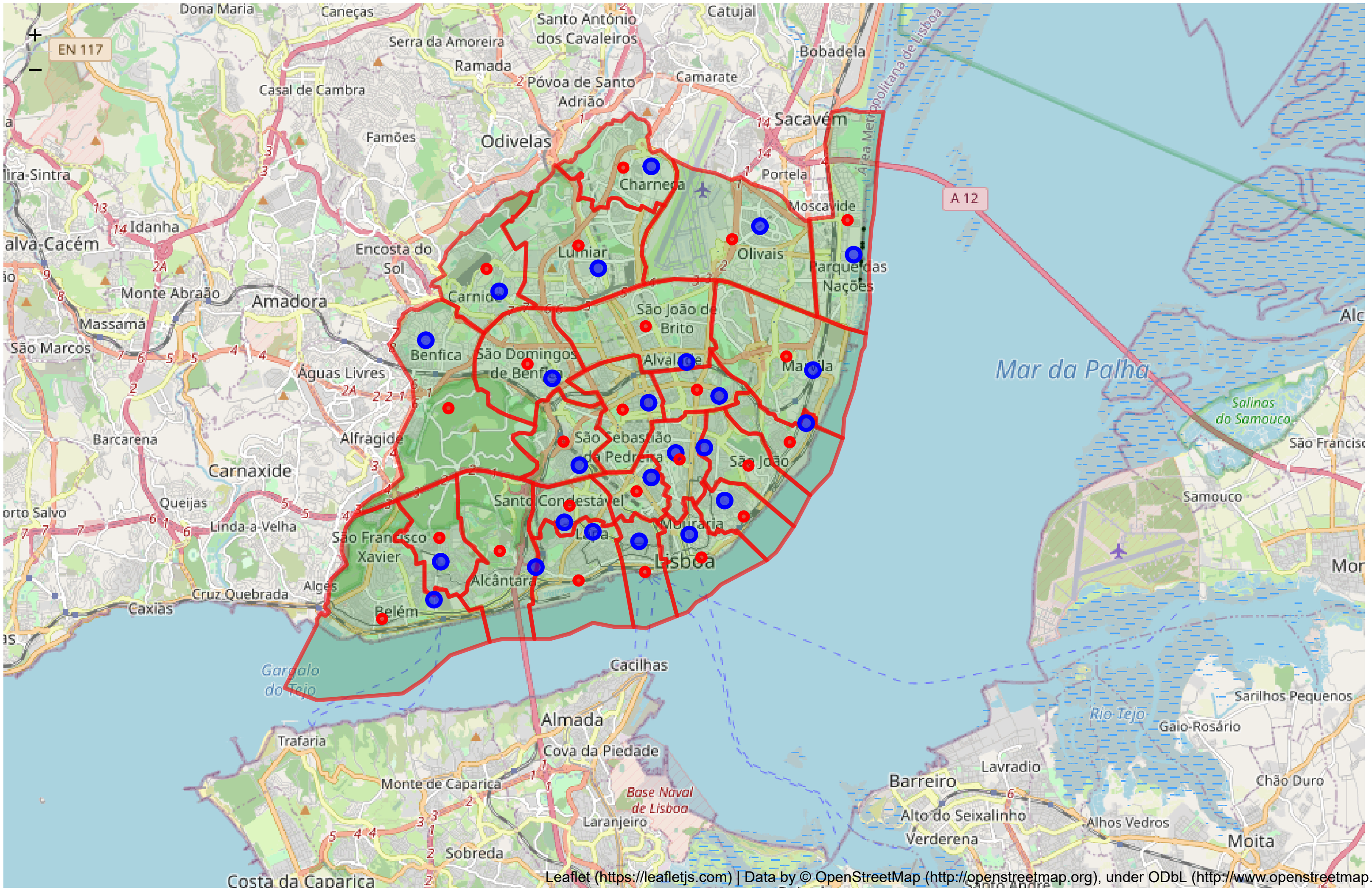
Data Exploration

Plot the boroughs and the two possible point of search.

In [8]:

```python
latitude = 38.73124000000007
longitude = -9.162549999999953
venues_map = folium.Map(location=[latitude, longitude], zoom_start=13) # generate map centred around the Conrad Hotel


folium.GeoJson(lisbon_shape,
    style_function=lambda x: {
        'color' : 'red',
        'opacity': 0.6,
        'fillColor' : 'green',
        }).add_to(venues_map)


for lat, lng, label in zip(df_lisbon.Lat, df_lisbon.Lon, df_lisbon.Borough):
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        color='blue',
        popup=label,
        fill = True,
        fill_color='blue',
        fill_opacity=0.6
    ).add_to(venues_map)

for lat, lng, label in zip(df_lisbon.Centroid_Lat, df_lisbon.Centroid_Lon, df_lisbon.Borough):
    folium.Circle(
        [lat, lng],
        radius=100,
        color='red',
        popup=label,
        fill = True,
        fill_color='red',
        fill_opacity=0.6
    ).add_to(venues_map)



# display map
venues_map
```

Out[8]:



Use the Foursquare API

From previous map the centroid coordinates are a better location to choose from when request the venues from Foursquare API. They can lead to better distribution, making less probable that a venue is not requested and also less duplicates. Let's get the venues using thees coordinates as starting point.

In [20]:
```
from shapely.geometry import shape, Point

rest_list = []

for ind1, rest in rest_unique.iterrows():
    point = Point(rest[["Venue Longitude"]].item(), rest[["Venue Latitude"]].item())
    for ind2, borough in df_lisbon.iterrows():
        polygon = shape(borough[["Geometry"]].item())
        if (polygon.contains(point)):
            frame = {'Borough': borough[["Borough"]].item(),'Borough Latitude': borough[["Centroid_Lat"]].item(),
                     'Borough Longitude': borough[["Centroid_Lon"]].item(),'Venue': rest[["Venue"]].item(), 'Venue Latitude': rest[["Venue La
titude"]].item(), 'Venue Longitude': rest[["Venue Longitude"]].item(),
                     'Venue Category': rest[["Venue Category"]].item()
                    }
            rest_list.append(frame)

col = ['Borough', 'Borough Latitude', 'Borough Longitude', 'Venue', 'Venue Latitude', 'Venue Longitude', 'Venue Category']
lisbon_restaurants_unique = pd.DataFrame(rest_list, columns = col)
lisbon_restaurants_unique.head()
```

Out[20]:

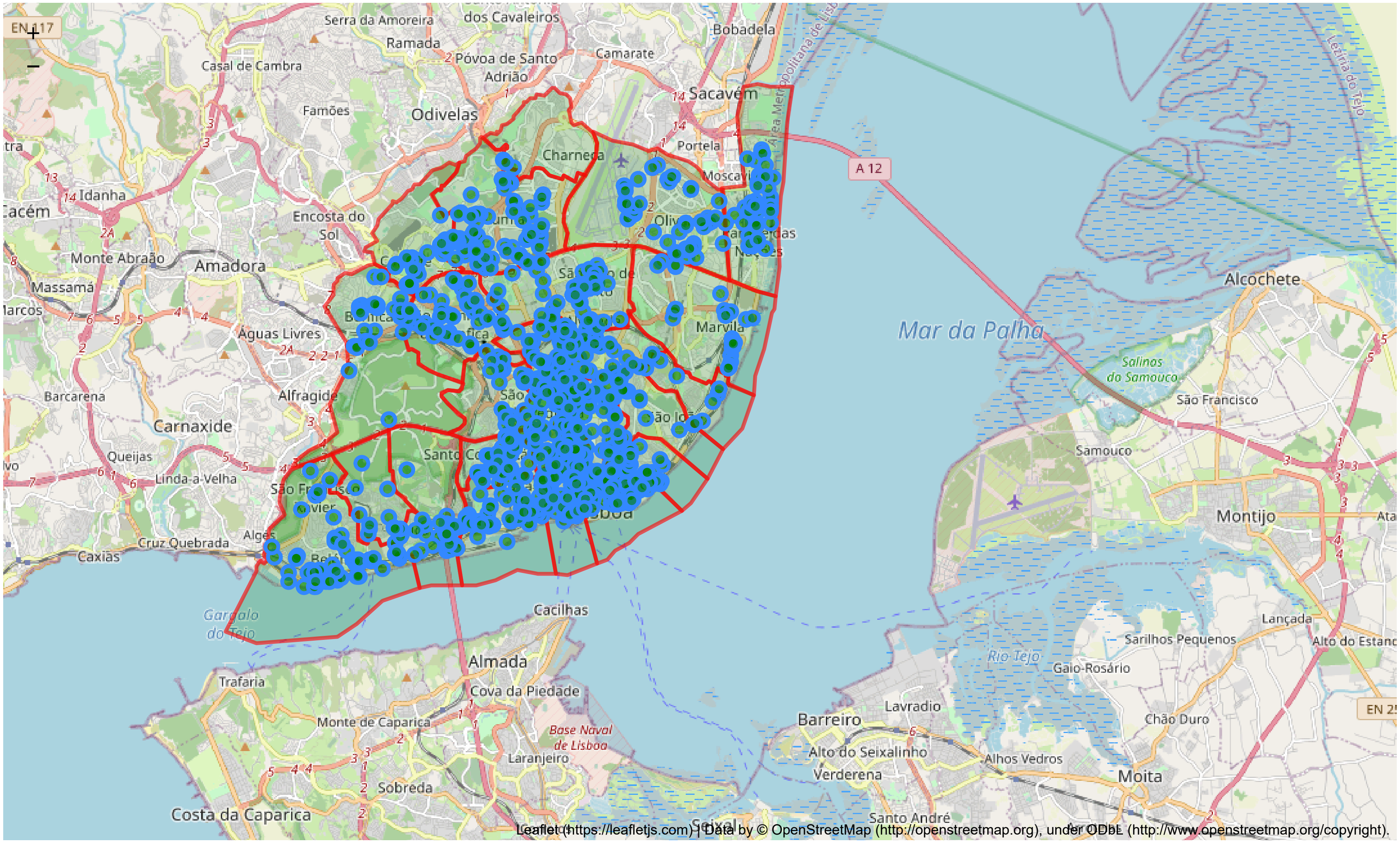
	Borough	Borough Latitude	Borough Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Misericórdia	38.705446	-9.146674	100 Maneiras	38.714712	-9.144826	Restaurant
1	Misericórdia	38.705446	-9.146674	100 Montaditos	38.707056	-9.147182	Tapas Restaurant
2	Parque das Nações	38.774806	-9.095478	100 Montaditos	38.772738	-9.092169	Tapas Restaurant
3	Carnide	38.765200	-9.186734	100% Hamburgueria	38.761268	-9.179715	Burger Joint
4	Areeiro	38.741380	-9.133525	1001 Nights Iranian Restaurant	38.745068	-9.140761	Persian Restaurant

In [21]:
```
df = lisbon_restaurants_unique.groupby('Borough').size().reset_index(name='Count')
df.sort_values(by='Count', ascending=False)
```

Out[21]:

	Borough	Count
5	Avenidas Novas	98
13	Lumiar	86
20	Santa Maria Maior	82
7	Belém	70
12	Estrela	69
21	Santo António	69
15	Misericórdia	68
2	Alvalade	65
22	São Domingos de Benfica	65
16	Olivais	64
4	Arroios	57
9	Campo de Ourique	56
17	Parque das Nações	56
11	Carnide	50
23	São Vicente	43
1	Alcântara	42
8	Benfica	33
14	Marvila	27
3	Areeiro	26
10	Campolide	14
0	Ajuda	12
18	Penha de França	9
6	Beato	9
19	Santa Clara	2

In [22]:
```
rest_map = folium.Map([latitude, longitude], zoom_start=12)

folium.GeoJson(lisbon_shape,
    style_function=lambda x: {
        'color' : 'red',
        'opacity': 0.6,
        'fillColor' : 'green',
        }).add_to(rest_map)

for label, lat, lng in zip(lisbon_restaurants_unique['Venue'],
                           lisbon_restaurants_unique['Venue Latitude'],
                           lisbon_restaurants_unique['Venue Longitude']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        location=[lat, lng],
        radius=5,
        popup=label,
        colur='red',
        fill=True,
        fill_color='green',
        fill_opacity=0.6,
        parse_html=False
    ).add_to(rest_map)



from IPython.display import display
display(rest_map)
```



There are some categories that are not restaurant, we must remove them. Those are not competition.

In [25]:
```
venueDF = lisbon_restaurants.groupby('Venue Category').size().reset_index(name='Counts')
venueDF.sort_values(by=['Counts'], ascending=False).head(10)
```

Out[25]:

	Venue Category	Counts
42	Portuguese Restaurant	249
44	Restaurant	149
32	Italian Restaurant	39
41	Pizza Place	37
8	Burger Joint	36
31	Indian Restaurant	29
47	Seafood Restaurant	29
5	BBQ Joint	28
52	Sushi Restaurant	27
34	Mediterranean Restaurant	27

Next, let's group rows by borough and by taking the mean of the frequency of occurrence of each category

In [27]:
```
lisbon_onehot = pd.get_dummies(lisbon_restaurants['Venue Category'])
lisbon_onehot.insert(loc=0, column='Borough', value=lisbon_restaurants['Borough'])
lisbon_grouped = lisbon_onehot.groupby('Borough').mean().reset_index()
lisbon_grouped.head()
```

Out[27]:

	Borough	African Restaurant	American Restaurant	Argentinian Restaurant	Asian Restaurant	Australian Restaurant	BBQ Joint	Brazilian Restaurant	Buffet	Burger Joint	Burrito Place	Cajun / Creole Restaurant	Cantonese Restaurant	Chinese Restaurant	Comfort Food Restaurant	Bo
0	Ajuda	0.0	0.0	0.000000	0.000000	0.0	0.181818	0.000000	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	0.00
1	Alcântara	0.0	0.0	0.032258	0.000000	0.0	0.032258	0.000000	0.0	0.000000	0.0	0.0	0.0	0.000000	0.032258	0.00
2	Alvalade	0.0	0.0	0.000000	0.037037	0.0	0.018519	0.018519	0.0	0.055556	0.0	0.0	0.0	0.055556	0.000000	0.01
3	Areeiro	0.0	0.0	0.047619	0.000000	0.0	0.047619	0.000000	0.0	0.095238	0.0	0.0	0.0	0.047619	0.000000	0.04
4	Arroios	0.0	0.0	0.000000	0.000000	0.0	0.021277	0.042553	0.0	0.042553	0.0	0.0	0.0	0.021277	0.000000	0.00

Organize by most common venue

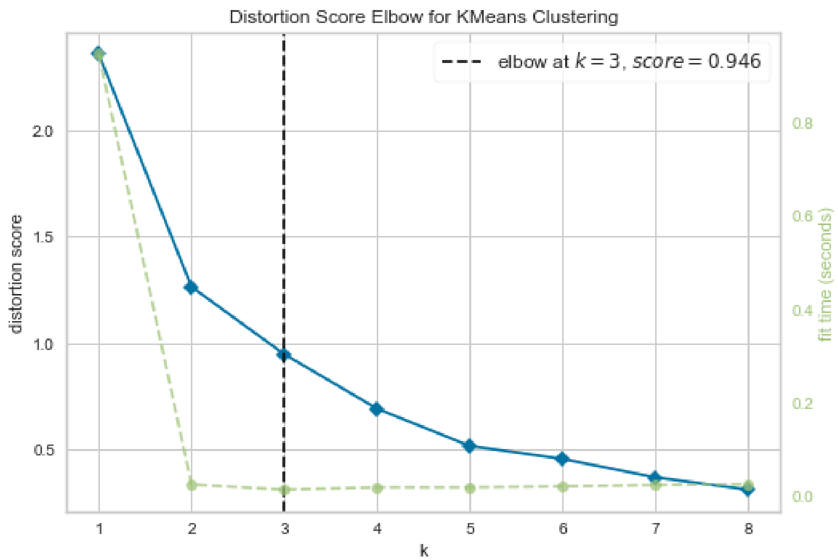
In [30]:
```
lisbon_venues_sorted.head()
```

Out[30]:

	Borough	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Ajuda	Restaurant	BBQ Joint	Portuguese Restaurant	Seafood Restaurant	African Restaurant	Ramen Restaurant	Italian Restaurant	Japanese Restaurant	Mediterranean Restaurant	Mexican Restaurant
1	Alcântara	Restaurant	Mediterranean Restaurant	Portuguese Restaurant	Pizza Place	Italian Restaurant	BBQ Joint	Comfort Food Restaurant	Seafood Restaurant	Argentinian Restaurant	Eastern European Restaurant
2	Alvalade	Portuguese Restaurant	Restaurant	Italian Restaurant	Chinese Restaurant	Burger Joint	Indian Restaurant	Pizza Place	Asian Restaurant	Sushi Restaurant	Fast Food Restaurant
3	Areeiro	Restaurant	Portuguese Restaurant	Italian Restaurant	Burger Joint	Chinese Restaurant	Indian Restaurant	Hot Dog Joint	Persian Restaurant	Empanada Restaurant	Mediterranean Restaurant
4	Arroios	Portuguese Restaurant	Indian Restaurant	Restaurant	Vegetarian / Vegan Restaurant	Japanese Restaurant	Brazilian Restaurant	Burger Joint	Seafood Restaurant	South Indian Restaurant	Pizza Place

Machine Learning - Clustering

In [31]:
```python
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer, SilhouetteVisualizer

lisbon_part_clustering = lisbon_grouped.drop('Borough', 1)

# Instantiate the clustering model and visualizer
model = KMeans()
visualizer = KElbowVisualizer(model, k=(1,9))

visualizer.fit(lisbon_part_clustering)        # Fit the data to the visualizer
visualizer.poof()    # Draw/show/poof the data
```



Out[31]: <AxesSubplot:title={'center':'Distortion Score Elbow for KMeans Clustering'}, xlabel='k', ylabel='distortion score'>

In [39]:
```python
# set number of clusters
kclusters = 4

lisbon_grouped_clustering = lisbon_grouped.drop('Borough', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(lisbon_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_
```

Out[39]:
```
array([0, 1, 2, 1, 2, 1, 0, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 1, 3, 0, 2, 2,
       1, 2])
```

In [41]:
```python
lisbon_venues_sorted.drop(columns=['Cluster Labels'], inplace=True)
lisbon_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
lisbon_venues_sorted.head()
```

Out[41]:

	Cluster Labels	Borough	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	0	Ajuda	Restaurant	BBQ Joint	Portuguese Restaurant	Seafood Restaurant	African Restaurant	Ramen Restaurant	Italian Restaurant	Japanese Restaurant	Mediterranean Restaurant	Mexican Restaurant
1	1	Alcântara	Restaurant	Mediterranean Restaurant	Portuguese Restaurant	Pizza Place	Italian Restaurant	BBQ Joint	Comfort Food Restaurant	Seafood Restaurant	Argentinian Restaurant	Eastern European Restaurant
2	2	Alvalade	Portuguese Restaurant	Restaurant	Italian Restaurant	Chinese Restaurant	Burger Joint	Indian Restaurant	Pizza Place	Asian Restaurant	Sushi Restaurant	Fast Food Restaurant
3	1	Areeiro	Restaurant	Portuguese Restaurant	Italian Restaurant	Burger Joint	Chinese Restaurant	Indian Restaurant	Hot Dog Joint	Persian Restaurant	Empanada Restaurant	Mediterranean Restaurant
4	2	Arroios	Portuguese Restaurant	Indian Restaurant	Restaurant	Vegetarian / Vegan Restaurant	Japanese Restaurant	Brazilian Restaurant	Burger Joint	Seafood Restaurant	South Indian Restaurant	Pizza Place

In [42]:
```python
lisbon_merged = df_lisbon.join(lisbon_venues_sorted.set_index('Borough'), on='Borough')
# lisbon_merged['Cluster Labels'] = lisbon_merged['Cluster Labels'].fillna(0)
# lisbon_merged['Cluster Labels'] = lisbon_merged['Cluster Labels'].astype(int)
lisbon_merged.drop(columns=['Geometry','Lat', 'Lon'], inplace=True)

lisbon_merged.head()
```

Out[42]:

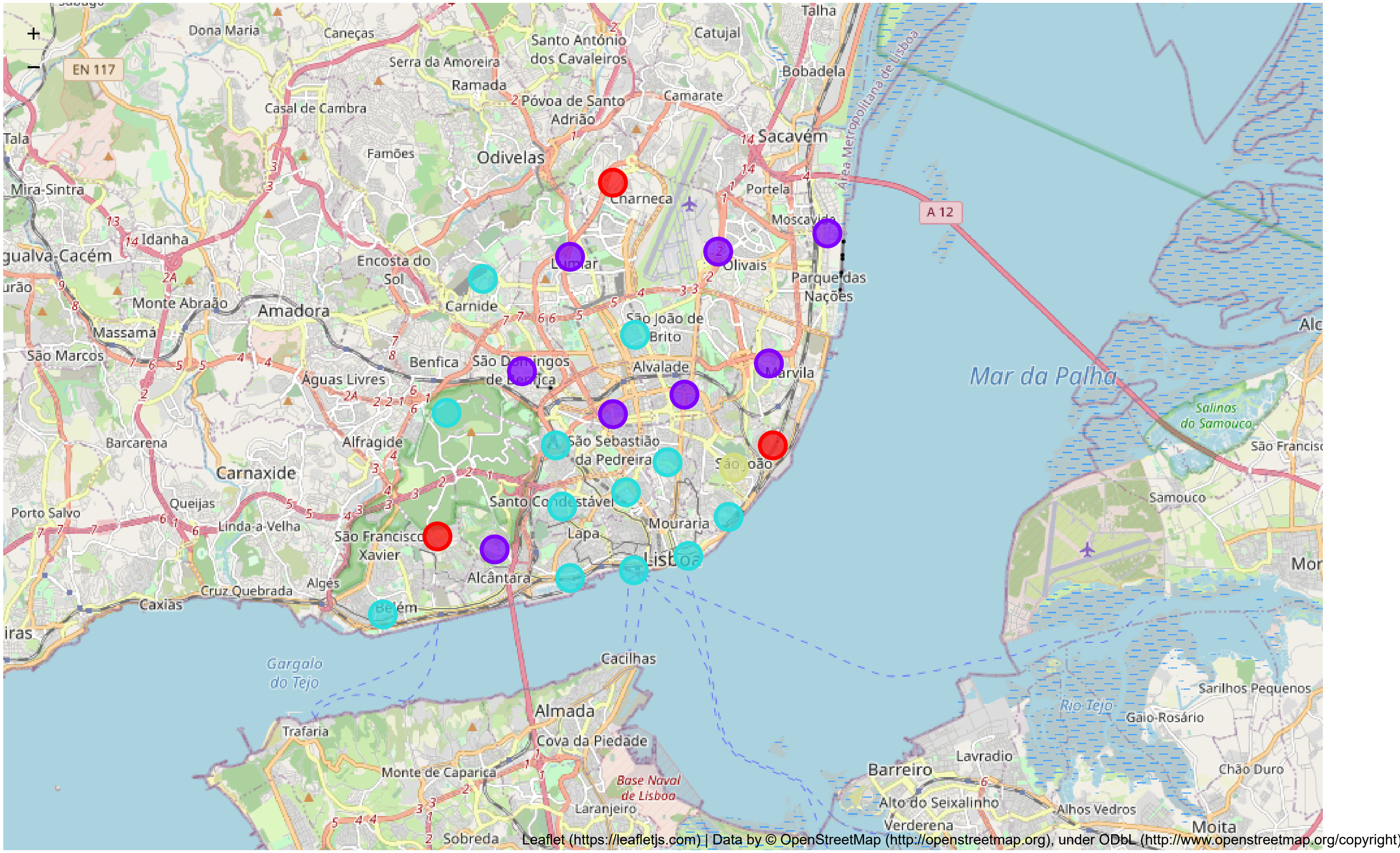
	Borough	Population	Area(km2)	Density(per km2)	Centroid_Lat	Centroid_Lon	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Co
0	Ajuda	15617	2.88	5422.6	38.712179	-9.198651	0	Restaurant	BBQ Joint	Portuguese Restaurant	Seafood Restaurant	African Restaurant	Ramen Restaurant	Italian Restaurant	Jap Rest
1	Alcântara	13943	5.07	2750.1	38.709598	-9.183388	1	Restaurant	Mediterranean Restaurant	Portuguese Restaurant	Pizza Place	Italian Restaurant	BBQ Joint	Comfort Food Restaurant	Se Rest
2	Alvalade	31813	5.34	5957.5	38.753888	-9.146497	2	Portuguese Restaurant	Restaurant	Italian Restaurant	Chinese Restaurant	Burger Joint	Indian Restaurant	Pizza Place	Rest
3	Areeiro	20131	1.72	11704.1	38.741380	-9.133525	1	Restaurant	Portuguese Restaurant	Italian Restaurant	Burger Joint	Chinese Restaurant	Indian Restaurant	Hot Dog Joint	P Rest
4	Arroios	31653	2.13	14860.6	38.727657	-9.137940	2	Portuguese Restaurant	Indian Restaurant	Restaurant	Vegetarian / Vegan Restaurant	Japanese Restaurant	Brazilian Restaurant	Burger Joint	Se Rest

4 - Result and Discussion

In [43]:

```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=12)



# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]
# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(lisbon_merged['Centroid_Lat'], lisbon_merged['Centroid_Lon'], lisbon_merged['Borough'], lisbon_merged['Clus
ter Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=10,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)


map_clusters
```

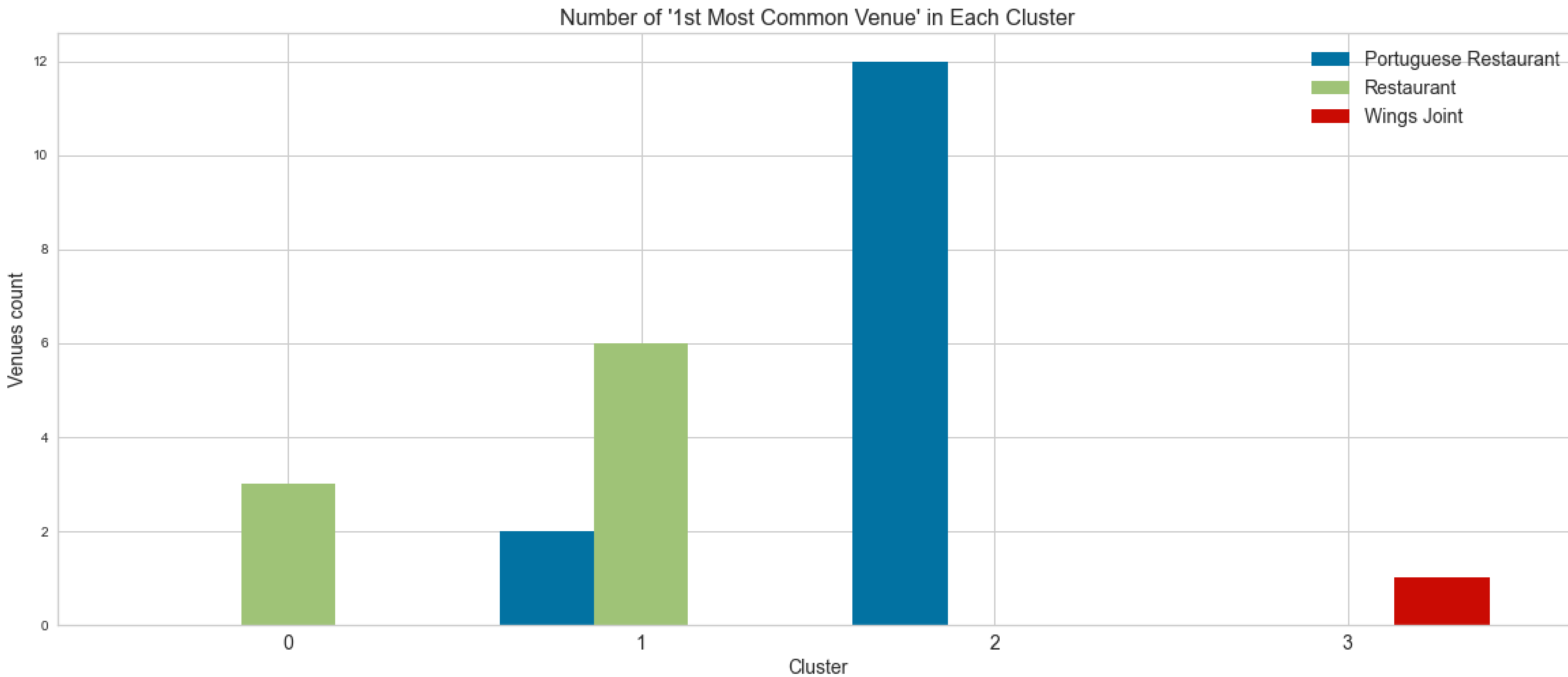
Out[43]:



Dona Maria
Caneças
Santo António dos Cavaleiros
Talha
Catujal
Bobadela
EN 117
Serra da Amoreira
Ramada
Casal de Cambra
Póvoa de Santo Adrião
Camarate
Sacavém
Tala
Famões
Odivelas
Mira-Sintra
Charneca
Portela
A 12
Moscavide
Idanha
Encosta do Sol
Olivais
Parque das Nações
Monte Abraão
Amadora
Carnide
Massamá
São João de Brito
São Marcos
Benfica
São Domingos de Benfica
Alvalade
Marvila
Águas Livres
Mar da Palha
Salinas do Samouco
Barcarena
Alfragide
São Sebastião da Pedreira
São João
São Francisco
Carnaxide
Queijas
Santo Condestável
Samouco
Porto Salvo
Linda-a-Velha
Mouraria
São Francisco Xavier
Lapa
Algés
Alcântara
Lisboa
Cruz Quebrada
Caxias
Belém
Gargalo do Tejo
Cacilhas
Sarilhos Pequenos
Rio Tejo
Gaio-Rosário
Almada
Trafaria
Cova da Piedade
Lavradio
Barreiro
Chão Duro
Monte de Caparica
Base Naval de Lisboa
Alto do Seixalinho
Alhos Vedros
Laranjeiro
Verderena
Moita
Sobreda
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [45]:
```
frame=cv_cluster.plot(kind='bar',figsize=(20,8),width = 0.8)

plt.legend(labels=cv_cluster.columns,fontsize= 14)
plt.title("Number of '1st Most Common Venue' in Each Cluster",fontsize= 16)
plt.xticks(fontsize=14)
plt.xticks(rotation=0)
plt.xlabel('Cluster', fontsize=14)
plt.ylabel('Venues count', fontsize=14)
```

Out[45]: Text(0, 0.5, 'Venues count')



```
In [46]: lisbon_merged.loc[lisbon_merged['Cluster Labels'] == 0, lisbon_merged.columns[[0] + list(range(4, lisbon_merged.shape[1]))]]
```

Out[46]:

	Borough	Centroid_Lat	Centroid_Lon	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Ajuda	38.712179	-9.198651	0	Restaurant	BBQ Joint	Portuguese Restaurant	Seafood Restaurant	African Restaurant	Ramen Restaurant	Italian Restaurant	Japanese Restaurant	Mediterranean Restaurant	Mexican Restaurant
6	Beato	38.731054	-9.110093	0	Restaurant	BBQ Joint	Portuguese Restaurant	African Restaurant	Italian Restaurant	Japanese Restaurant	Mediterranean Restaurant	Mexican Restaurant	Middle Eastern Restaurant	Modern European Restaurant
19	Santa Clara	38.785163	-9.152169	0	Restaurant	Ramen Restaurant	Indian Restaurant	Italian Restaurant	Japanese Restaurant	Mediterranean Restaurant	Mexican Restaurant	Middle Eastern Restaurant	Modern European Restaurant	Noodle House

```
In [47]: lisbon_merged.loc[lisbon_merged['Cluster Labels'] == 1, lisbon_merged.columns[[0] + list(range(4, lisbon_merged.shape[1]))]]
```

Out[47]:

	Borough	Centroid_Lat	Centroid_Lon	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Cor
1	Alcântara	38.709598	-9.183388	1	Restaurant	Mediterranean Restaurant	Portuguese Restaurant	Pizza Place	Italian Restaurant	BBQ Joint	Comfort Food Restaurant	Seafood Restaurant	Argentinian Restaurant	E Eur Rest
3	Areeiro	38.741380	-9.133525	1	Restaurant	Portuguese Restaurant	Italian Restaurant	Burger Joint	Chinese Restaurant	Indian Restaurant	Hot Dog Joint	Persian Restaurant	Empanada Restaurant	Mediterr Rest
5	Avenidas Novas	38.737465	-9.152328	1	Restaurant	Portuguese Restaurant	Italian Restaurant	Vegetarian / Vegan Restaurant	Japanese Restaurant	Steakhouse	Pizza Place	Indian Restaurant	Burger Joint	Rest
13	Lumiar	38.769795	-9.163491	1	Restaurant	Portuguese Restaurant	Pizza Place	Tapas Restaurant	Fast Food Restaurant	Sushi Restaurant	Chinese Restaurant	Mediterranean Restaurant	Brazilian Restaurant	Jap Rest
14	Marvila	38.747934	-9.110954	1	Restaurant	Portuguese Restaurant	Argentinian Restaurant	Fast Food Restaurant	Mediterranean Restaurant	Tapas Restaurant	Indian Restaurant	Buffet	Pizza Place	Cant Rest
16	Olivais	38.771035	-9.124650	1	Restaurant	Portuguese Restaurant	Chinese Restaurant	Fast Food Restaurant	Pizza Place	BBQ Joint	Salad Place	Food	Falafel Restaurant	Jap Rest
17	Parque das Nações	38.774806	-9.095478	1	Portuguese Restaurant	Burger Joint	Seafood Restaurant	Vegetarian / Vegan Restaurant	Pizza Place	Sushi Restaurant	Steakhouse	Restaurant	Italian Restaurant	Rest
22	São Domingos de Benfica	38.746451	-9.176373	1	Portuguese Restaurant	BBQ Joint	Burger Joint	Restaurant	Pizza Place	Chinese Restaurant	Italian Restaurant	Japanese Restaurant	Sushi Restaurant	

In [48]:
```
lisbon_merged.loc[lisbon_merged['Cluster Labels'] == 2, lisbon_merged.columns[[0] + list(range(4, lisbon_merged.shape[1]))]]
```

Out[48]:

	Borough	Centroid_Lat	Centroid_Lon	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue
2	Alvalade	38.753888	-9.146497	2	Portuguese Restaurant	Restaurant	Italian Restaurant	Chinese Restaurant	Burger Joint	Indian Restaurant	Pizza Place	Asian Restaurant	Sushi Restaurant
4	Arroios	38.727657	-9.137940	2	Portuguese Restaurant	Indian Restaurant	Restaurant	Vegetarian / Vegan Restaurant	Japanese Restaurant	Brazilian Restaurant	Burger Joint	Seafood Restaurant	South Indian Restaurant
7	Belém	38.696207	-9.213136	2	Portuguese Restaurant	Restaurant	Mediterranean Restaurant	Sushi Restaurant	Seafood Restaurant	BBQ Joint	Food Truck	Burger Joint	Diner
8	Benfica	38.737746	-9.196363	2	Portuguese Restaurant	Restaurant	Seafood Restaurant	BBQ Joint	Burger Joint	Sushi Restaurant	Mediterranean Restaurant	Food Truck	Pizza Place
9	Campo de Ourique	38.718542	-9.165753	2	Portuguese Restaurant	Seafood Restaurant	Restaurant	Steakhouse	Japanese Restaurant	Dim Sum Restaurant	Italian Restaurant	Sushi Restaurant	Burger Joint
10	Campolide	38.731158	-9.167270	2	Portuguese Restaurant	Japanese Restaurant	Restaurant	Vegetarian / Vegan Restaurant	Brazilian Restaurant	African Restaurant	Ramen Restaurant	Italian Restaurant	Mediterranean Restaurant
11	Carnide	38.765200	-9.186734	2	Portuguese Restaurant	Restaurant	Italian Restaurant	Sushi Restaurant	Pizza Place	Burger Joint	Fast Food Restaurant	Seafood Restaurant	Dim Sum Restaurant
12	Estrela	38.703768	-9.163458	2	Portuguese Restaurant	Restaurant	Italian Restaurant	Seafood Restaurant	Mediterranean Restaurant	Vegetarian / Vegan Restaurant	Indian Restaurant	BBQ Joint	Tapas Restaurant
15	Misericórdia	38.705446	-9.146674	2	Portuguese Restaurant	Restaurant	Tapas Restaurant	Italian Restaurant	Vegetarian / Vegan Restaurant	Burger Joint	Peruvian Restaurant	Pizza Place	Food Court
20	Santa Maria Maior	38.708311	-9.132404	2	Portuguese Restaurant	Restaurant	Indian Restaurant	Mediterranean Restaurant	Tapas Restaurant	Peruvian Restaurant	Chinese Restaurant	Seafood Restaurant	Argentinian Restaurant
21	Santo António	38.721345	-9.148808	2	Portuguese Restaurant	Restaurant	Italian Restaurant	Steakhouse	Modern European Restaurant	Pizza Place	Ramen Restaurant	Himalayan Restaurant	Sushi Restaurant
23	São Vicente	38.716395	-9.121721	2	Portuguese Restaurant	Mediterranean Restaurant	Vegetarian / Vegan Restaurant	Indian Restaurant	Pizza Place	Seafood Restaurant	Food Stand	Comfort Food Restaurant	Restaurant

In [49]:
```
lisbon_merged.loc[lisbon_merged['Cluster Labels'] == 3, lisbon_merged.columns[[0] + list(range(4, lisbon_merged.shape[1]))]]
```

Out[49]:

	Borough	Centroid_Lat	Centroid_Lon	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
18	Penha de França	38.726432	-9.120539	3	Wings Joint	Indian Restaurant	Italian Restaurant	Middle Eastern Restaurant	Ramen Restaurant	Japanese Restaurant	Mediterranean Restaurant	Mexican Restaurant	Modern European Restaurant	Noodle House

5 - Conclusion

The analysis was done using Fousquare API and geographical information of Lisbon boroughs.

Information was acquired on different sources and different techniques: scraping web, importing geojson.

The information for the restaurants was obtained though Foursquare API.

Data cleaning was used to prepare data for machine learning algorithm.

A clustering algorithm, k-means was used to cluster the boroughs using the most common categories of restaurants.

Finally visualizing the clusters on Lisbon map and exploring a specific cluster can give our stakeholders some usefull insights.

More and insights can be obtained enriching the analysis with further developments.

6 - Next Developments

Next steps to develop the study:

Upgrading the amount of information, for example:

- Home sales/Rental prices
- People movement around the city
- Distance to market suppliers

Use a different Venue API with more data.

Try different machine learning algorithms.