Capstone Project

FINAL REPORT

Stéphane Degeye | IBM Data Science Course | 02-Mar-2020

Introduction

This project aims to determine the best location for opening a Lounge Bar in Brussels.

Choosing the best place for your business is valuable:

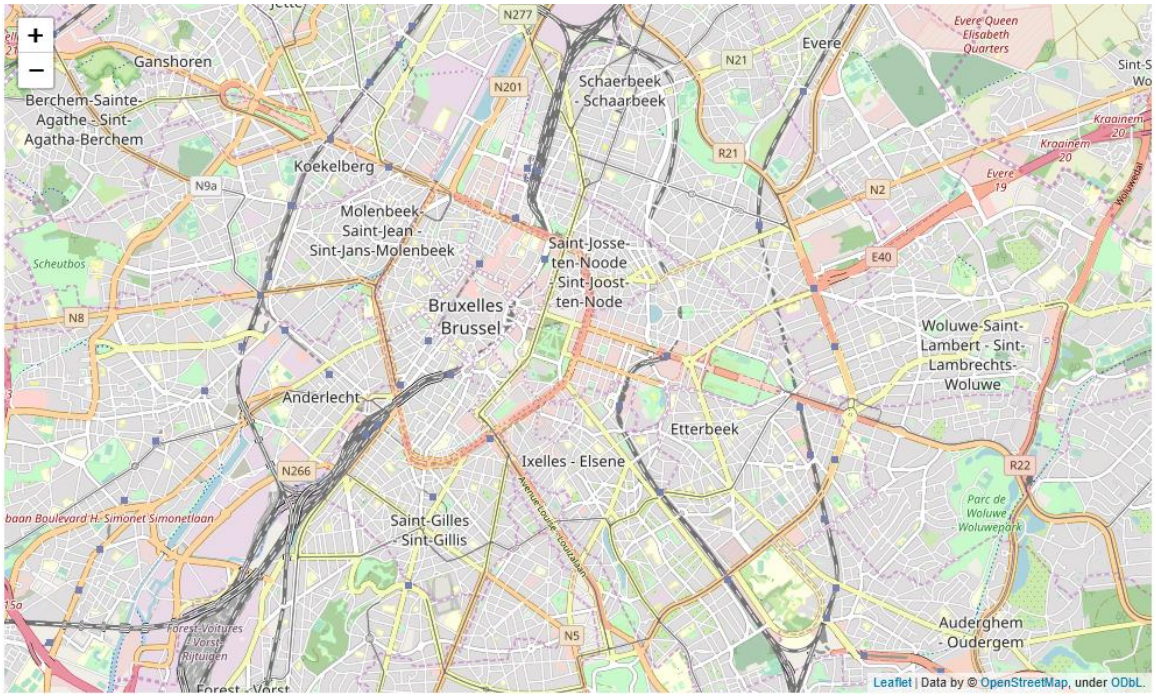
- Accessibility and parking are essential
- Competing business is sometimes a good thing
- Flow from other existing businesses (Bars, Restaurants,…)
- What are the potential customers (Schools, Factories,…)
- To answer these questions, a structural approach is necessary
- Data sources and Spatial analysis can help

We can extend this strategy to other projects. Of course in some cases, it will be necessary to cross-check the data taking into account different factors such as standard of living, crime, …

First, we must determine a reference point for our investigations.

For this purpose, the reference point will be the city of Brussels:

Coordinates: [50.8436709 , 4.3674366933879565]



Data Source

Foursquare API will provide us with the data necessary for our project.

In order to apply the different criteria discussed in the introduction, we will explore the following data from Foursquare:

- Bars
- Parking
- Schools
- Restaurants
- Metro Stations

We will locally save these data in different datasets files (csv).

Some notes about data extraction:

- Unfortunately, the data obtained from Foursquare is limited to 100 records despite a limit set to 500 during the function call
- Searches sometimes return records that are unrelated to the request, so a data wrangling step is needed (e.g.: a query on “parking” returns venues related to Hotels

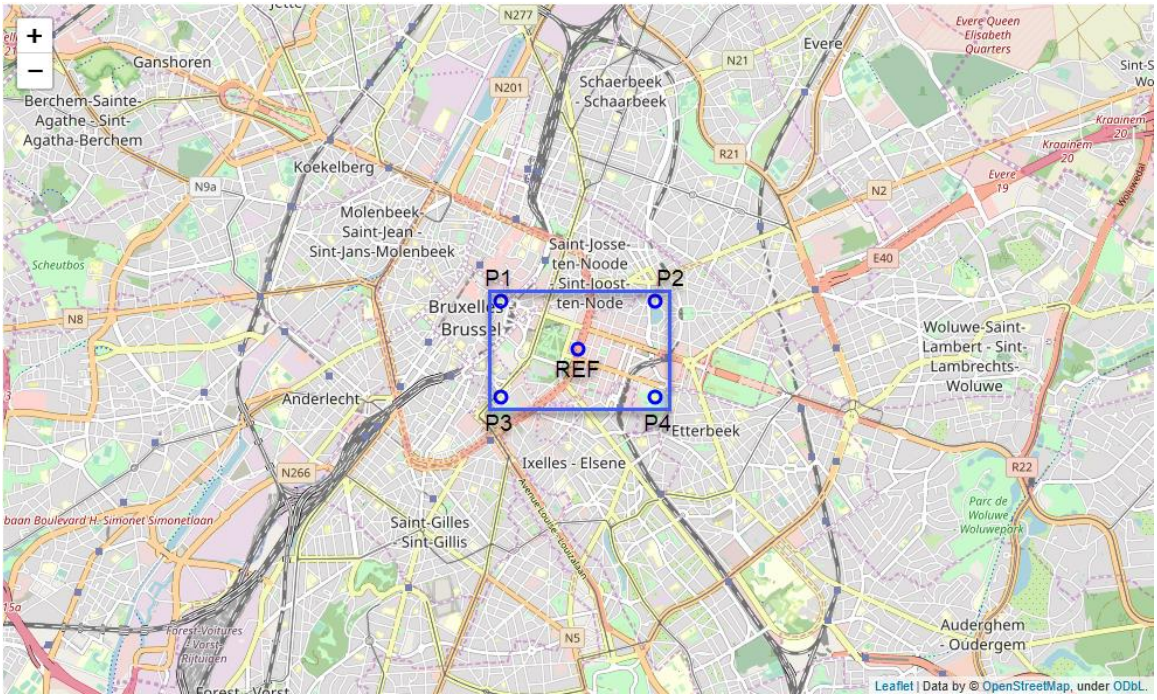
A workaround to this Foursquare limitation is to explore data, not only from the Reference Coordinate but also from other ones.

At this effect, I have defined a function for offsetting Latitude/Longitude Coordinate.

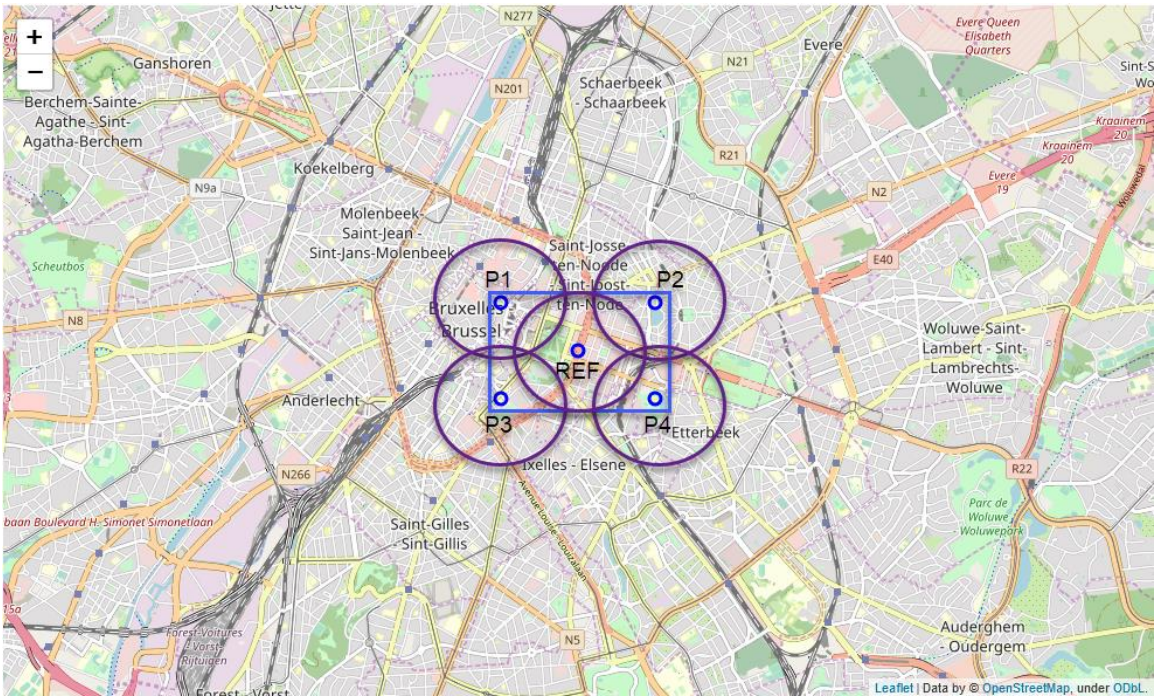
I consider Brussels coordinate as reference and I compute four additional Coordinates:

	Latitude	Longitude
0	50.839179	4.356056
1	50.848162	4.378818
2	50.839179	4.378818
3	50.848162	4.356056
4	50.843671	4.367437

Overview of these Points on the Map:



This will expand the field of data extraction from Foursquare and finally get more results:



Of course, I will remove the duplicated records

Overview of Datasets:

Query = "bar":

```
city_venues_Bar = pd.read_csv('city_venues_Bar.csv')
city_venues_Bar.drop(['Unnamed: 0'],axis=1,inplace=True)
city_venues_Bar.head()
```

	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Goupil le Fol	50.845403	4.352287	Bar
1	Le Dillens	50.831825	4.349135	Bar
2	Enjoy Brussels	50.838000	4.359300	Bar
3	Monk	50.850166	4.347425	Bar
4	Café Bizon	50.848532	4.347709	Bar

Query = "parking":

```
city_venues_Parking = pd.read_csv('city_venues_Parking.csv')
city_venues_Parking.drop(['Unnamed: 0'],axis=1,inplace=True)
city_venues_Parking.drop(city_venues_Parking.loc[city_venues_Parking['Venue Category'] != 'Parking'].index, inplace=True)
city_venues_Parking.head()
```

	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Parking Toison d'Or	50.837451	4.359342	Parking
1	Interparking	50.838096	4.352668	Parking
2	Parking Poelaert	50.837493	4.353179	Parking
3	Parking Boulevard de Waterloolaan	50.836655	4.357625	Parking
4	Parking Deux Portes	50.837973	4.359946	Parking

Query = "school":

```
city_venues_School = pd.read_csv('city_venues_School.csv')
city_venues_School.drop(['Unnamed: 0'],axis=1,inplace=True)
city_venues_School.head()
```

	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Alliance française de Bruxelles-Europe	50.842615	4.367413	Language School
1	Eurospeak	50.836444	4.360627	School
2	amira language school	50.839814	4.366747	School
3	Huis van het Nederlands	50.843246	4.347243	School
4	Sint-Jorisbasisschool	50.843541	4.349657	School

Query = "restaurant":

	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Genco	50.841465	4.353083	Italian Restaurant
1	COCO Donuts	50.841219	4.355315	Donut Shop
2	Chez Richard	50.841357	4.353860	Gastropub
3	Au Vieux Saint Martin	50.840984	4.354737	Belgian Restaurant
4	À l'Ombre de la Ville	50.839149	4.360859	Italian Restaurant

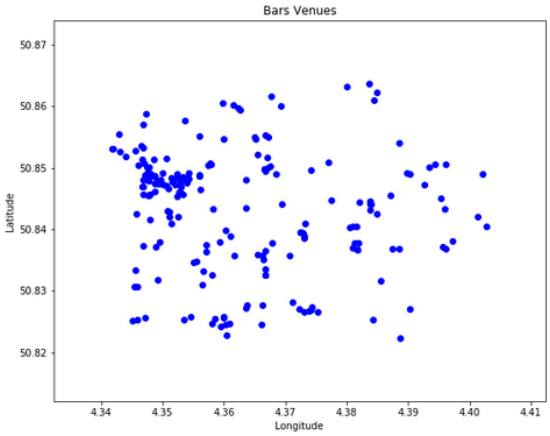
```
city_venues_Metro = pd.read_csv('city_venues_Metro.csv')
city_venues_Metro.drop(['Unnamed: 0'],axis=1,inplace=True)
city_venues_Metro.drop(city_venues_Metro.loc[city_venues_Metro['Venue Category'] != 'Metro Station'].index, inplace=True)
city_venues_Metro.head()
```

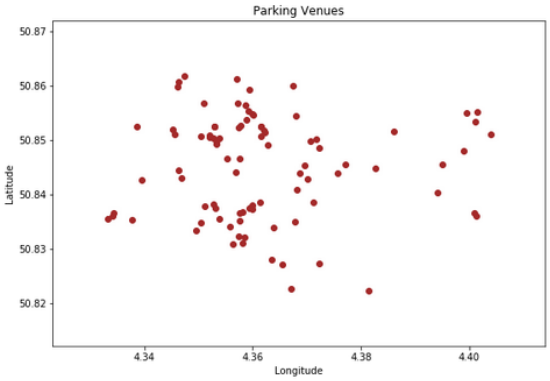
Query = "metro":

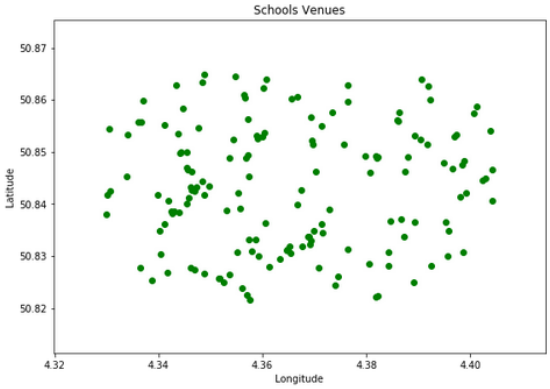
```
city_venues_Metro = pd.read_csv('city_venues_Metro.csv')
city_venues_Metro.drop(['Unnamed: 0'],axis=1,inplace=True)
city_venues_Metro.drop(city_venues_Metro.loc[city_venues_Metro['Venue Category'] != 'Metro Station'].index, inplace=True)
city_venues_Metro.head()
```

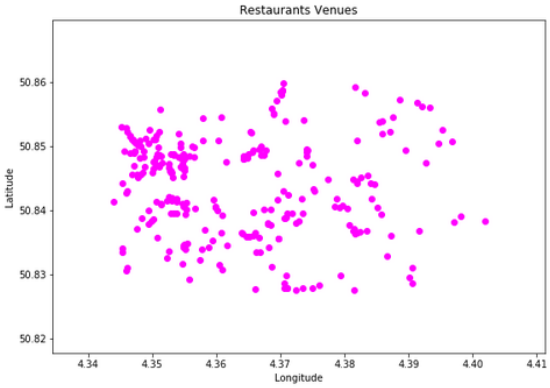
	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Parc / Park (STIB / MIVB) (Park (MIVB))	50.845634	4.362325	Metro Station
1	Naamsepoort / Porte de Namur (MIVB / STIB) (Na...	50.838246	4.361939	Metro Station
2	Trône / Troon (STIB / MIVB) (Troon (MIVB))	50.840669	4.366280	Metro Station
3	Louise / Louiza (STIB / MIVB) (Louiza (MIVB))	50.835620	4.355336	Metro Station
4	Centraal Station / Gare Centrale (MIVB / STIB)...	50.846440	4.358626	Metro Station

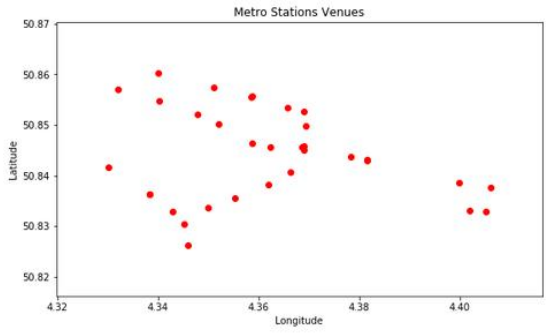
Spatial overview of the Data received from Foursquare:

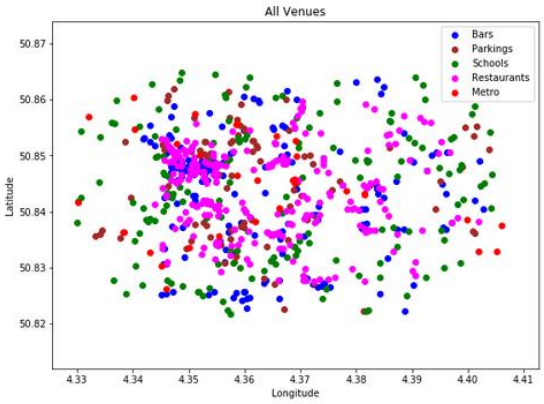








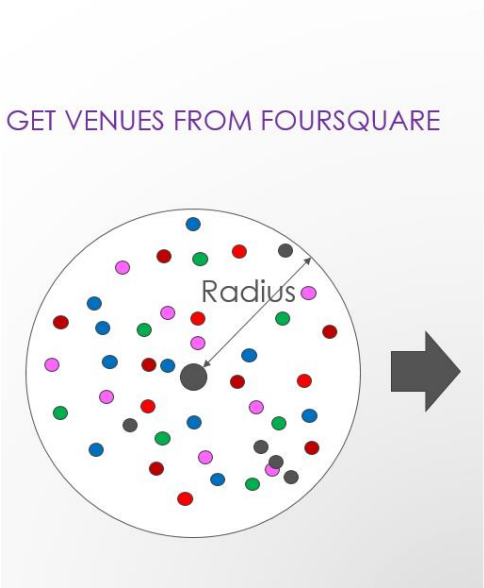


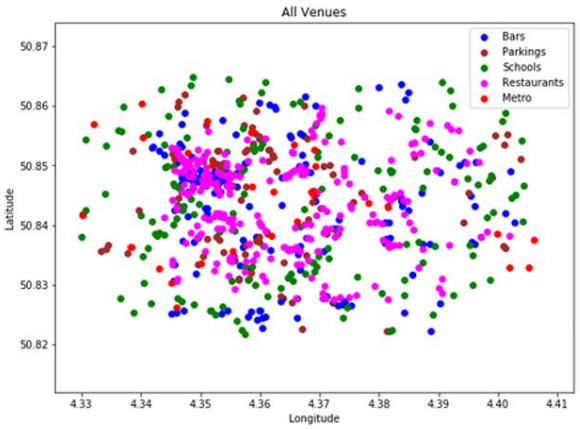


Methodology

A] GET POINTS OF INTEREST'S SPATIAL DATA

METHOD: EXPLORE VENUES FROM FOURSQUARE (API)





Define Foursquare Credentials and Version

```
CLIENT_ID = '[redacted]'
CLIENT_SECRET = '[redacted]'
VERSION = '20180605'
```

Define Function to explore

```
# Function for exploring Venues from FourSquare
def getNearbyVenues(name, lat, lng, qry,radius=2000,limit=500):

    venues_list=[]

    url = 'https://api.foursquare.com/v2/venues/explore'

    params = dict(client_id=CLIENT_ID, client_secret=CLIENT_SECRET,v=VERSION,
                  ll='{:f},{:f}'.format(lat,lng),
                  query=qry,
                  radius=radius,
                  limit=limit
                   )

    results = requests.get(url=url,params=params).json()["response"]['groups'][0]['items']

    # return only relevant information for each nearby venue
    venues_list.append([( v['venue']['name'],
                          v['venue']['location']['lat'],
                          v['venue']['location']['lng'],
                          v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])

    nearby_venues.columns = [ 'Venue',
                              'Venue Latitude',
                              'Venue Longitude',
                              'Venue Category']

    return(nearby_venues)
```

B] DATA FILTER ON DISTANCE AMONG “BARS” AND OTHER NEAREST VENUES

METHOD: GEOPY FUNCTION / DATAFRAME MANIPULATION

Since "Bars" Venues responds to several various highlighted arguments:

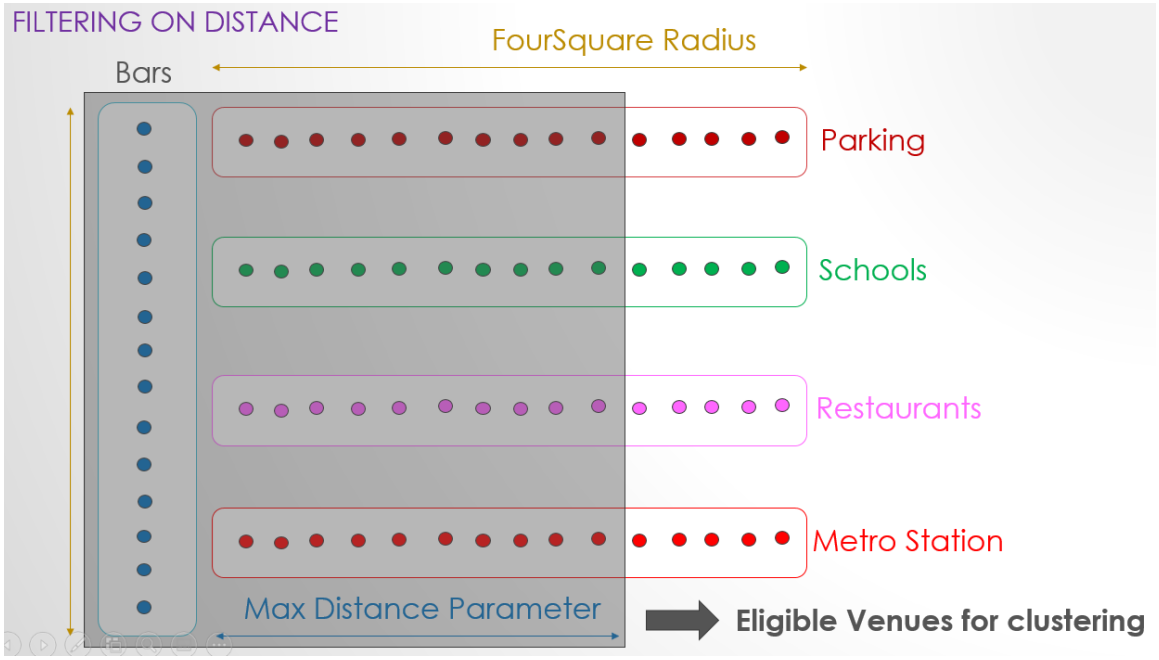
- Competing business
- Flow from other existing businesses

I will use it as a reference for filtering the other Venues Categories:

- For each “Bars” Venues, I compute all nearest Venues (Parking, Schools…)
- Based on predefined “Max Distance Parameters, I eliminate some of them

The resulting Data will contain all “Bars” Venues and all other Venues (for which distance to “Bars” Venues are lower or equal to a “MAX DISTANCE” parameter).

A first level filtering is done by the radius parameter (specified at Foursquare level) and a second level filtering is applied among “Bars” and the other POI (Points of Interest):



⇨ Note that all the "Bars" venues are kept in this process

I have applied geodesic () function (from GeoPy) to calculate the distance between two coordinates:

Data Reduction based on other Venues Max Distance's from Bar's Venues

```
# Function to get the value closest to another one and the distance between them
def nearest(row, geom_union, df1, df2, geom1_col='geometry', geom2_col='geometry', src_column=None):

    distances = []

    # Find the geometry that is closest
    nearest = df2[geom2_col] == nearest_points(row[geom1_col], geom_union)[1]

    # Get the corresponding value from df2 (matching is based on the geometry)
    value = df2[nearest][src_column].to_numpy()[0]

    start_point = row['geometry']
    end_point_series = df2[nearest]['geometry']
    end_point = end_point_series.iloc[0]

    dist_to_point = start_point.distance(end_point)

    start_point_np = np.array(start_point)
    end_point_np = np.array(end_point)

    start_point_np[[0, 1]] = start_point_np[[1, 0]]
    end_point_np[[0, 1]] = end_point_np[[1, 0]]

    dist_between_points = geodesic(start_point_np, end_point_np)

    distances.append(dist_between_points)

    return value,distances
```

And the result after function call is the following:

	Venue	Venue Latitude	Venue Longitude	Venue Category	geometry	centroid	nearest_id_Parking	dist_to_Parking	nearest_id_School	dist_to_School	nearest_id_Restau
1	Le Dillens	50.831825	4.349135	Bar	POINT (4.34914 50.83183)	POINT (4.34914 50.83183)	Interparking	0.175794	Athénée Royal Victor Horta	0.489978	La B
2	Enjoy Brussels	50.838000	4.359300	Bar	POINT (4.35930 50.83800)	POINT (4.35930 50.83800)	Parking Toison d'Or	0.061142	Eurospeak	0.196720	Chyl
3	Monk	50.850166	4.347425	Bar	POINT (4.34742 50.85017)	POINT (4.34742 50.85017)	Q-Park Dansaert	0.164022	Maria-Boodschaplyceum	0.146686	ABC Poisso
4	Café Bizon	50.848532	4.347709	Bar	POINT (4.34771 50.84853)	POINT (4.34771 50.84853)	Q-Park Dansaert	0.322795	Instituut Anneessens Funck	0.274556	Fanny
5	Bar des Amis	50.850042	4.347845	Bar	POINT (4.34784 50.85004)	POINT (4.34784 50.85004)	Q-Park Dansaert	0.195663	Maria-Boodschaplyceum	0.174745	Frites A
...	...	...	...	...	...	...	...	...	...	...	
202	Affligem Café	50.847755	4.348977	Pub	POINT (4.34898 50.84775)	POINT (4.34898 50.84775)	Interparking	0.342892	Instituut Anneessens Funck	0.253419	Yi
203	Gecko	50.848013	4.346808	Cocktail Bar	POINT (4.34681 50.84801)	POINT (4.34681 50.84801)	Q-Park Dansaert	0.355731	Athénée Léon Lepage (Atheneum Léon Lepage)	0.149666	Pu
205	Life Is Beautiful - Cocktail Bar	50.852547	4.343006	Cocktail Bar	POINT (4.34301 50.85255)	POINT (4.34301 50.85255)	Q-Park Lepage	0.175955	Mabo Basisschool	0.122050	
206	Quai des Bananes	50.847280	4.350161	Cocktail Bar	POINT (4.35016 50.84728)	POINT (4.35016 50.84728)	Interparking	0.380632	Sint-Joris Basisschool	0.348212	Manhattn's Bu
207	Au Laboureur	50.852756	4.345554	Pub	POINT (4.34555 50.85276)	POINT (4.34555 50.85276)	Q-Park Lepage	0.090910	Mabo Basisschool	0.148652	La M

125 rows × 14 columns

C] CLUSTERING

METHOD: DBSCAN ALGORITHM

Now, one interesting thing would be to discover the places where there is a minimum concentration of points of interest.

DBSCAN is very interesting in our case for the following reasons:

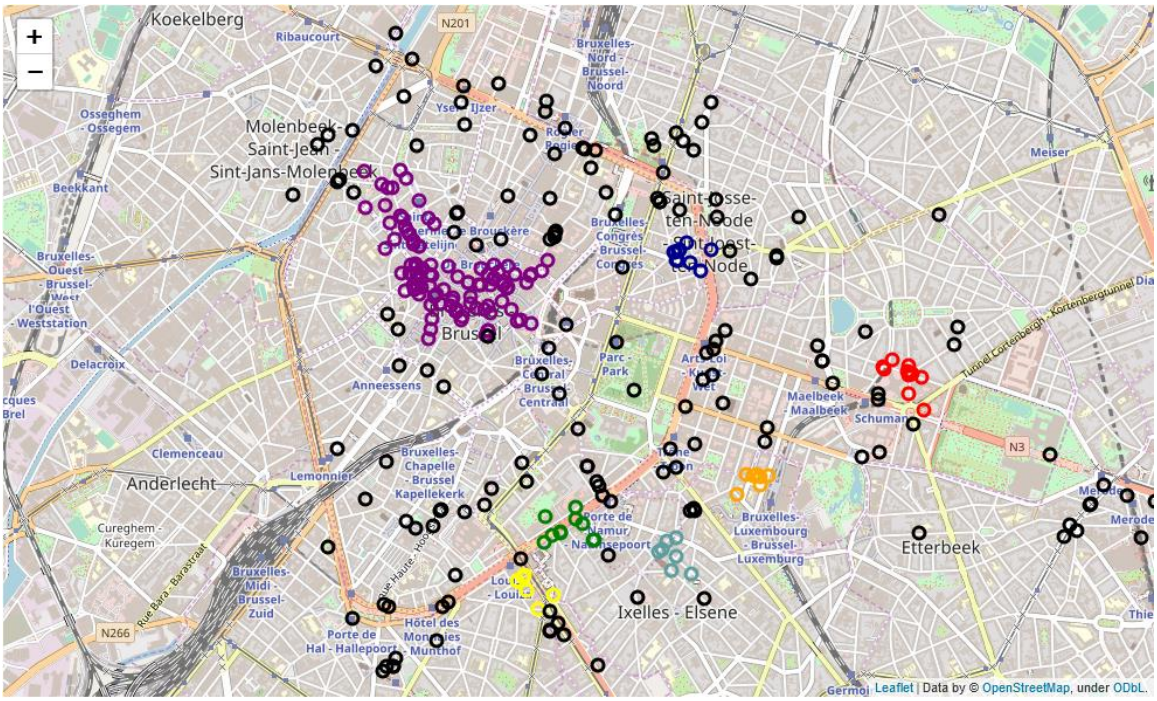
- Suitable for spatial data
- Relatively easy to implement (we have just to specify eps and min_samples param)
- Isolation of Outliers

I executed this algorithm with epsilon=0.15 (150 m) and min_samples = 7

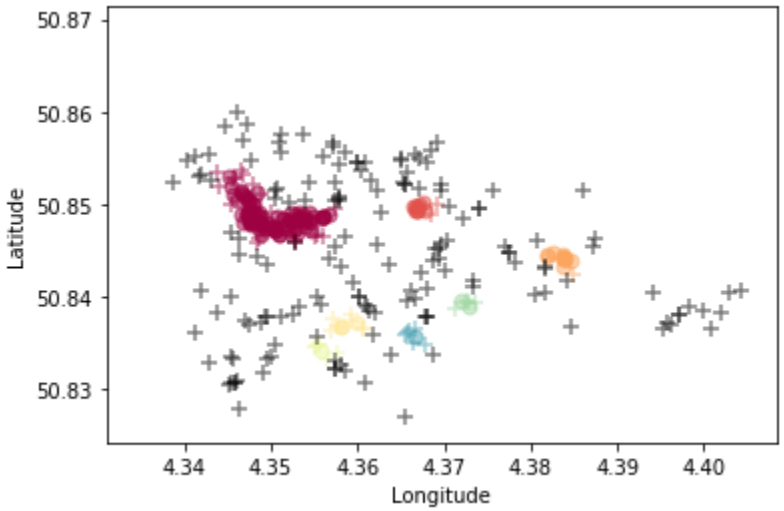
Results

Output of algorithm execution was:

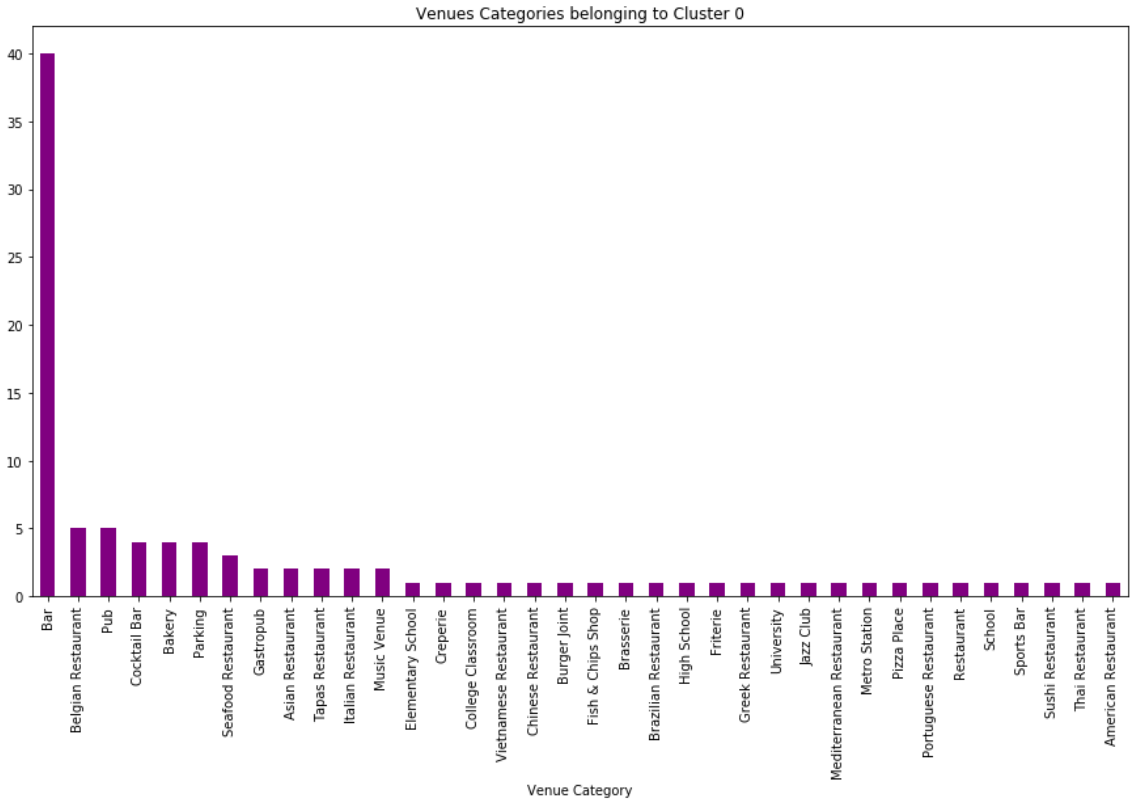
- 8 Clusters found (7 relevant clusters and 1 related to Outliers)
- Total of Outliers: 158

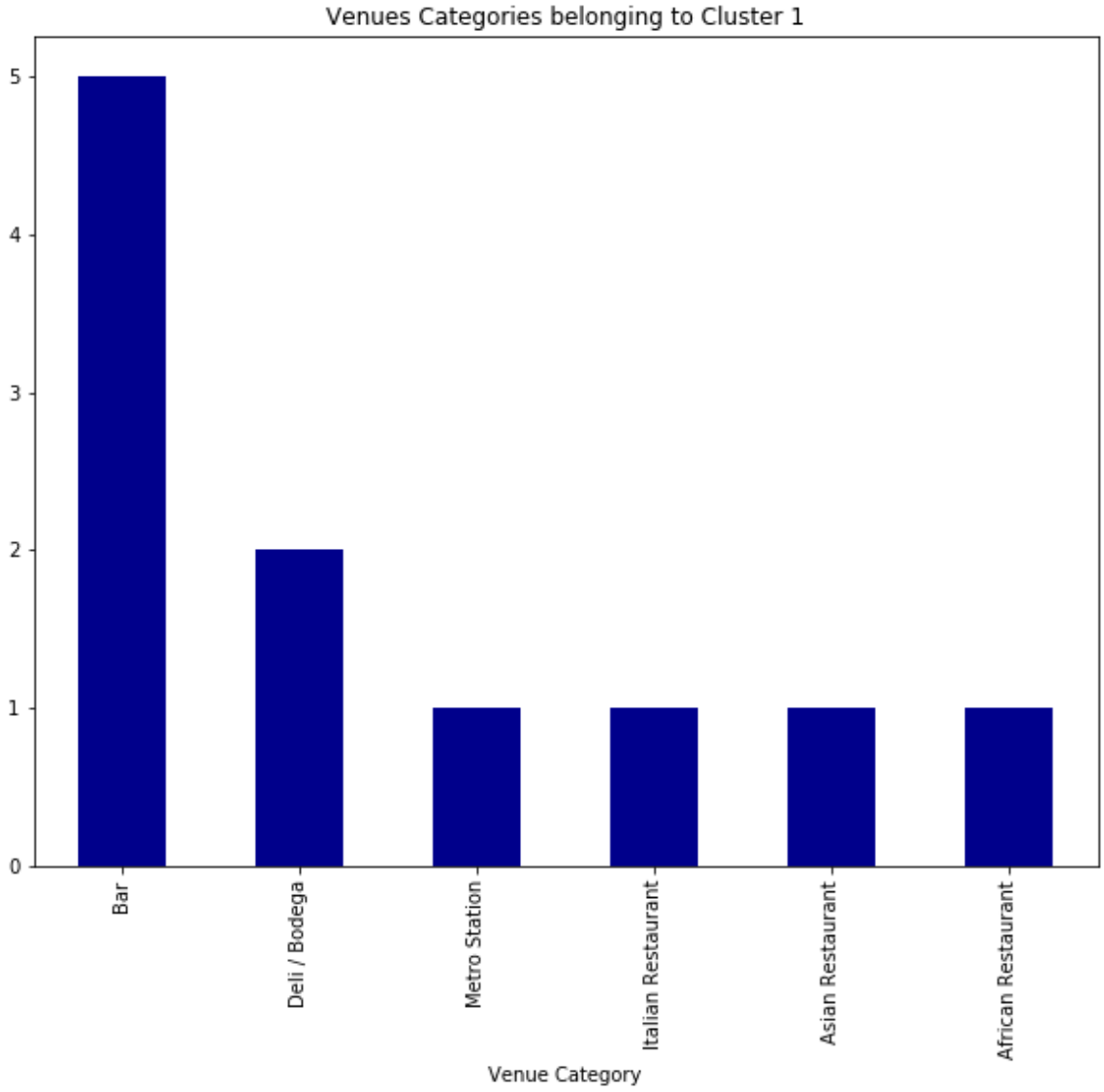


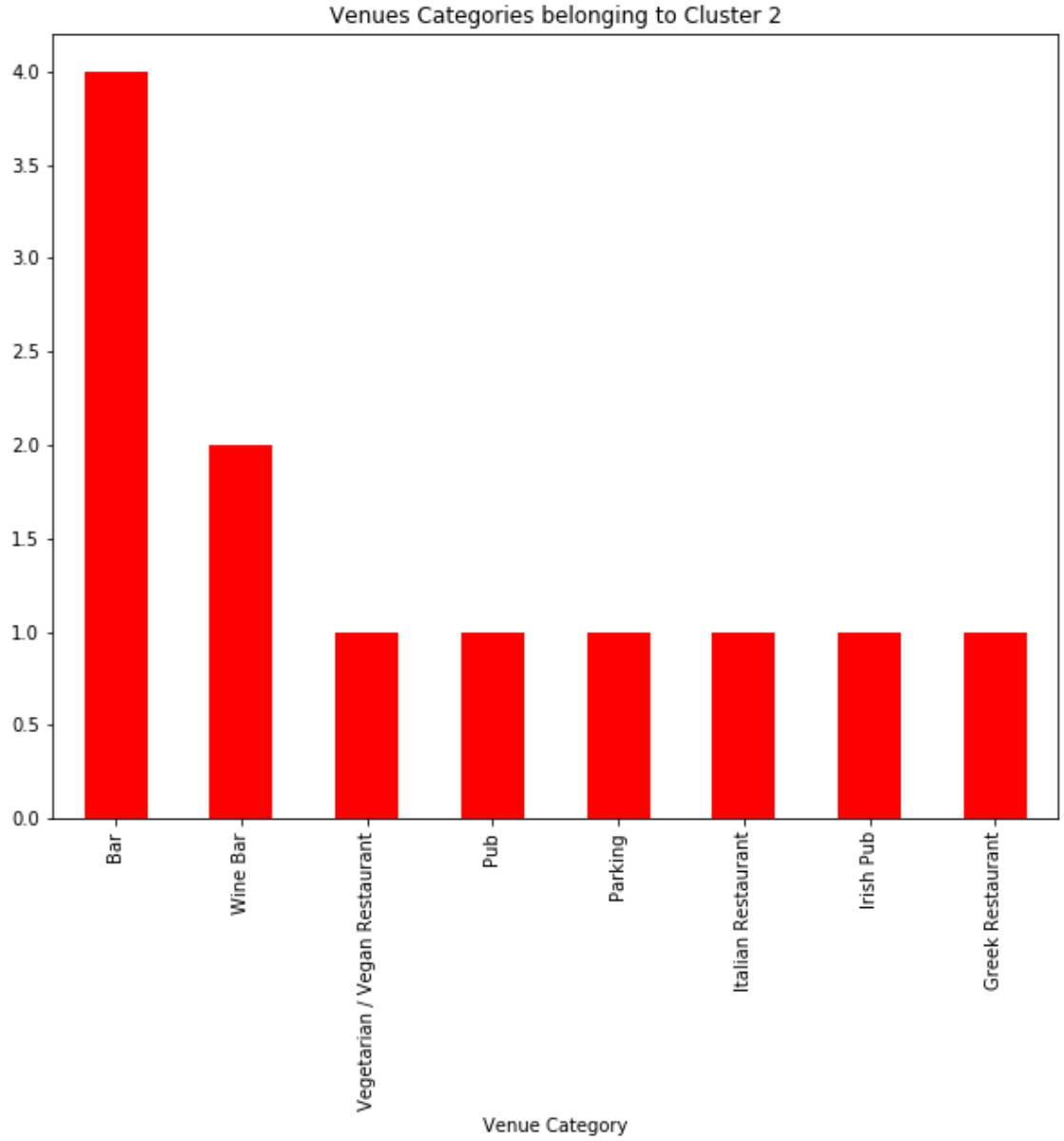
Discard of outliers (noise points) is a great functionality of DBSCAN algorithm:

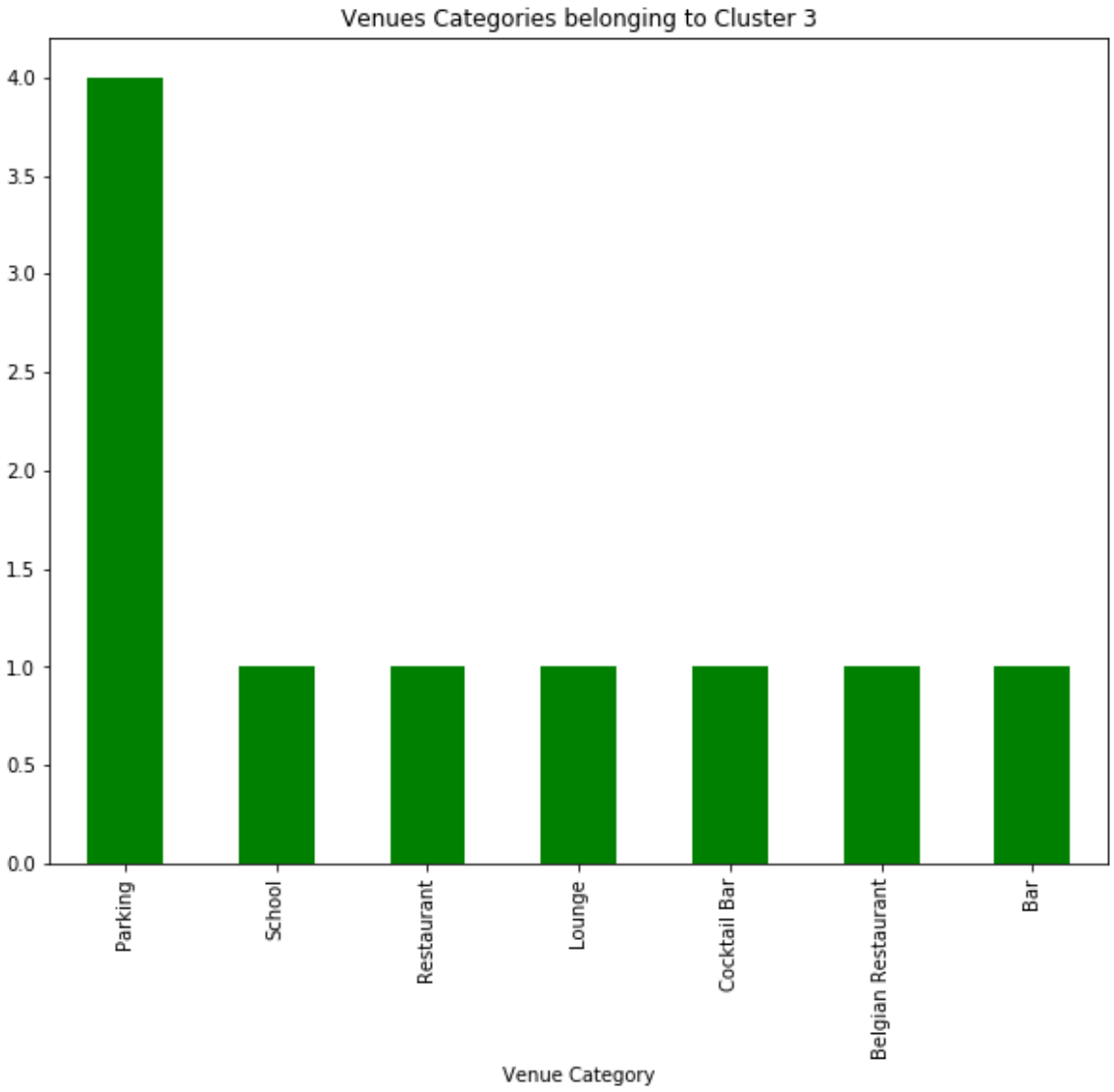


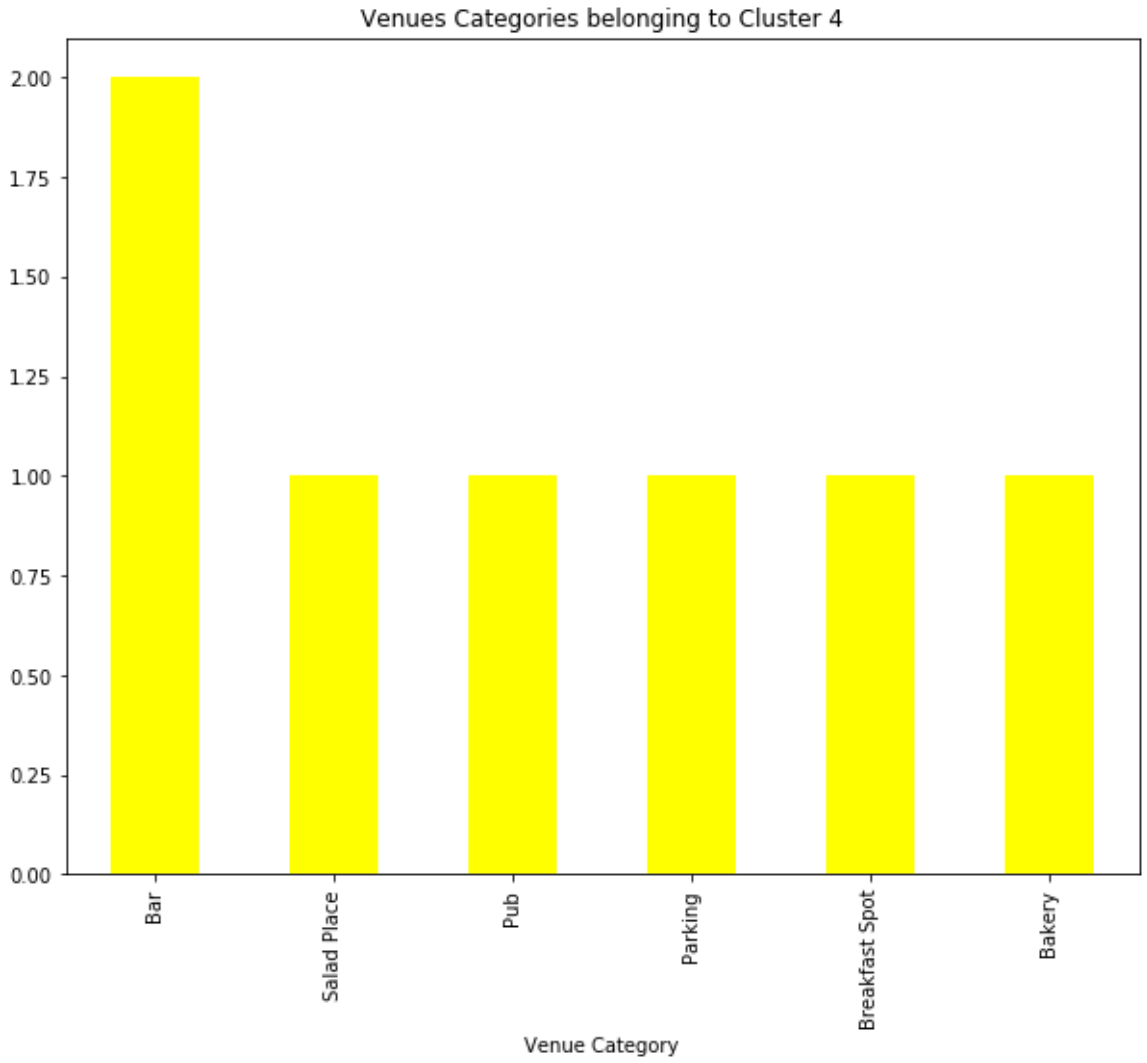
Overview of the different clusters (by Venues Category):

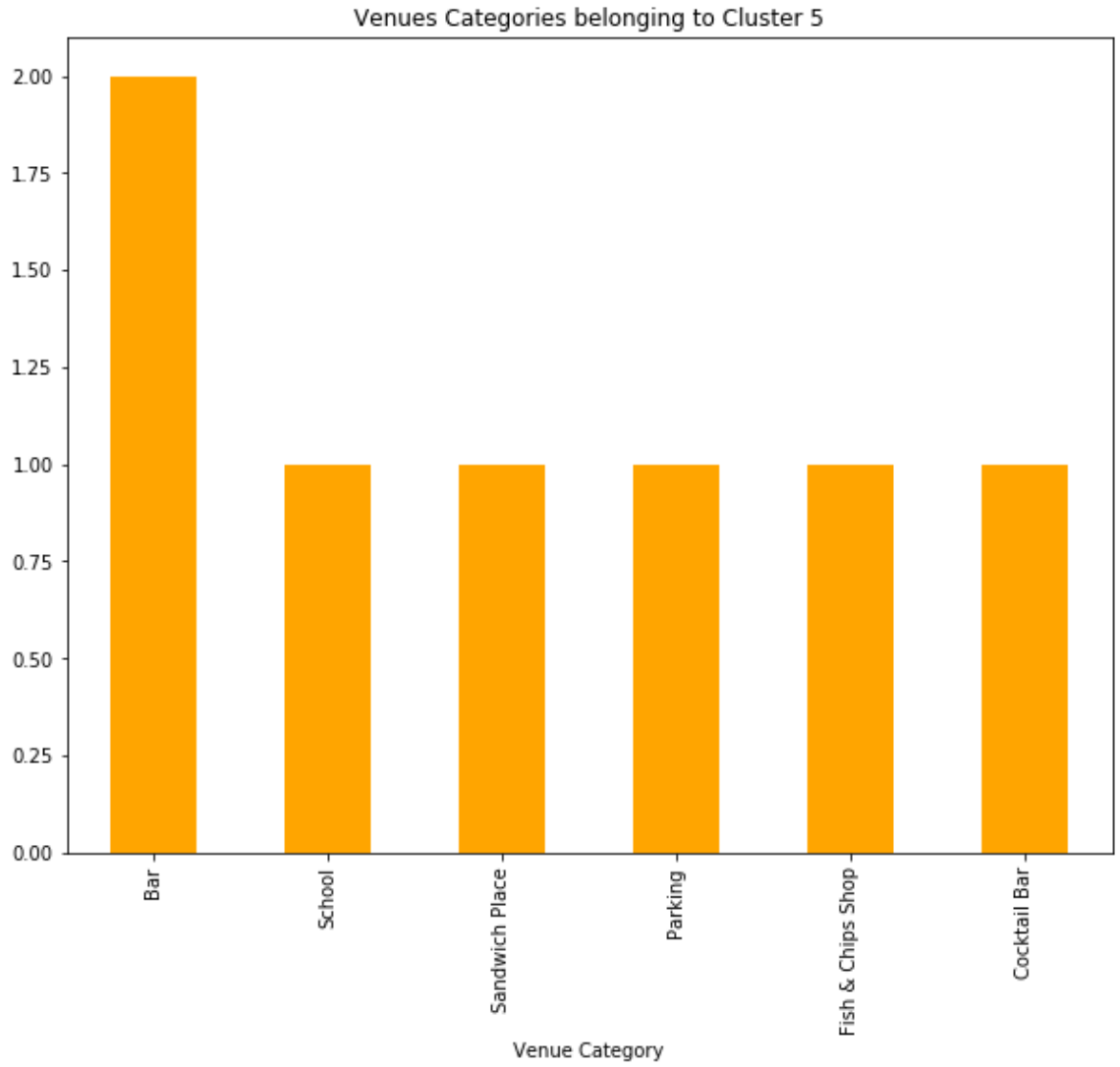


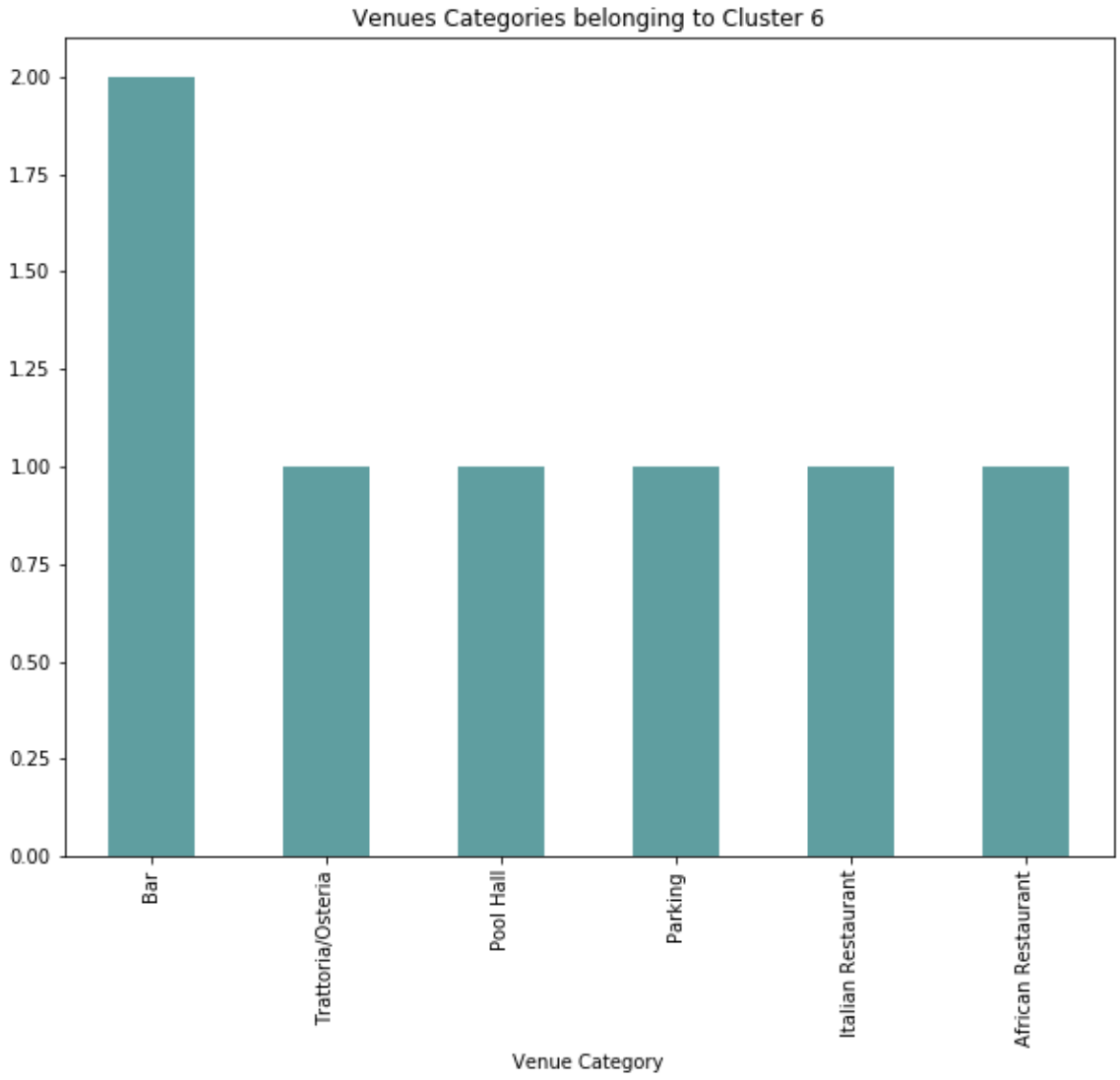

Venues Categories belonging to Cluster 1
5
4
3
2
1
0
Bar
Deli / Bodega
Metro Station
Italian Restaurant
Asian Restaurant
African Restaurant
Venue Category


Venues Categories belonging to Cluster 2
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Bar
Wine Bar
Vegetarian / Vegan Restaurant
Pub
Parking
Italian Restaurant
Irish Pub
Greek Restaurant
Venue Category


Venues Categories belonging to Cluster 3
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Parking
School
Restaurant
Lounge
Cocktail Bar
Belgian Restaurant
Bar
Venue Category


Venues Categories belonging to Cluster 4
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Bar
Salad Place
Pub
Parking
Breakfast Spot
Bakery
Venue Category


Venues Categories belonging to Cluster 5
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Bar
School
Sandwich Place
Parking
Fish & Chips Shop
Cocktail Bar
Venue Category



Discussion

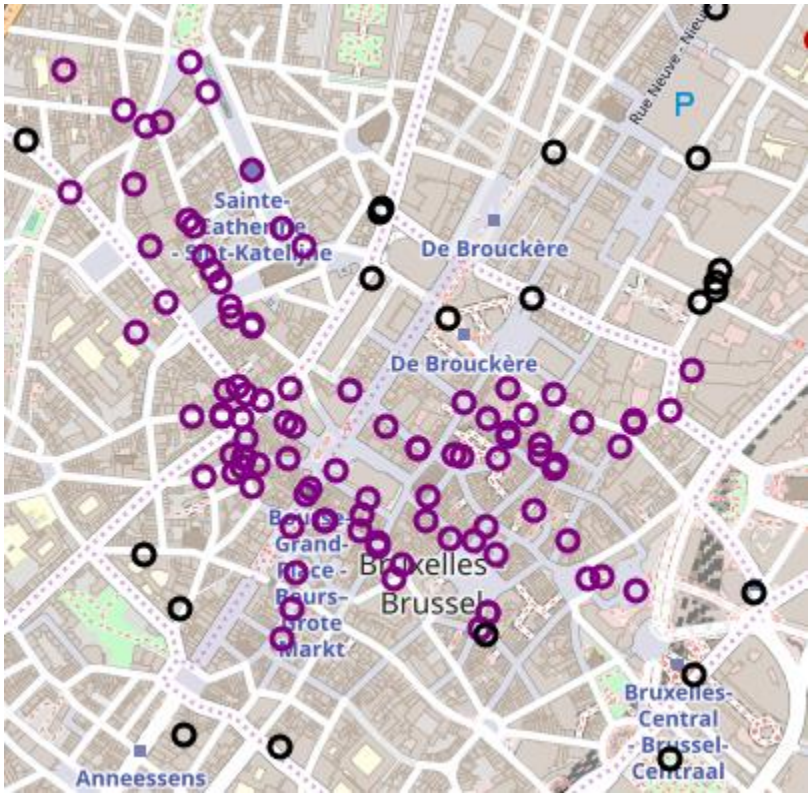
- ⇨ **Cluster 0**: Results show one main area (Cluster 0) and six other small ones: (This main area seems interesting given the diversity and concentration of POI)

- ⇨ **Cluster 3**: Parking, Lounge, Cocktail Bar, School and one Belgian Restaurant is a good distribution as well

- ⇨ **Cluster 5**: Parking, Bar, Cocktail Bar, School and one Sandwich Place is another good distribution

Conclusion

One of the challenges of the "New Lounge Bar" is to make yourself known to the public, so I recommend cluster 0 (Place Sainte-Catherine / Grand Place) as your first location:



Then in the future, if success is there, I recommend

- ⇨ Cluster 3 (Porte de Namur)
- ⇨ Cluster 5 (Place de Bruxelles Luxembourg)

to open another bar(s):

thank you for your attention,

Stéphane Degeye