Mama Mia Pizza

Battle of the Amsterdam Neighborhoods

Coursera Capstone Project for the IBM Data Science Certification

Wiljo Meijnhout

January 2020

Introduction & Business Problem

© pixabay

Introduction and background

This report is created as the final assignment for the Applied Data Science course from Coursera which results in an IBM Data Science Certification. The business problem and names I've defined, as part of the final assignment, is fictional but the data used to solve the problem is real data and can solve a real live business problem without any issues.

Business Problem

In Rome, Italy a famous franchise company called 'Mama Mia Pizza' want to spread their wings across the European capital cities, starting with Amsterdam in the Netherlands. Amsterdam City, according to Wikipedia has a population of 866,737 and is very popular amongst tourist from all over the world. According to DW.COM in 2018 19.000.000 tourists have visited Amsterdam. All these tourists have to stay of course in hotels, but as we all know also in one of the many Airbnb houses/rooms.

My client wants to '***Start a Mama Mia Pizza Restaurant in a neighborhood in Amsterdam, crowded of Airbnb hosts but where the restaurant market is less saturated compared to other neighborhoods'***.

In contrast to tourists who stay in a hotel an 'Airbnb tourist' have their dinner not at home but eat it i.e. in a restaurant. Pizza is a world famous-, well known, and an accessible dish. According to the Pizzajoint.com approximately 3 billion!! pizzas are sold in only the U.S. each year.

Data

Data Gathering

In order to solve my client's business problem a desk study was performed to gather the available data and to decide the Data Science approach. Airbnb itself does not provide any data at all according to their company policies, so that was a dead end. But the website insideairbnb.com does provide all the information we need. insideairbnb.com does scrape the Airbnb website in a smart way and represents the scraped data visually on their website and also make the data available via downloads in several formats.

Airbnb data

insideairbnb.com/amsterdam/ shows almost 20.000 Airbnb hosts across Amsterdam. (see fig.1). For computation power performance but also visuality, 20.000 Airbnb hosts in a dataset is too much and not needed at all to solve the business problem. Below is explained how we decreased this number to approximately 7000.

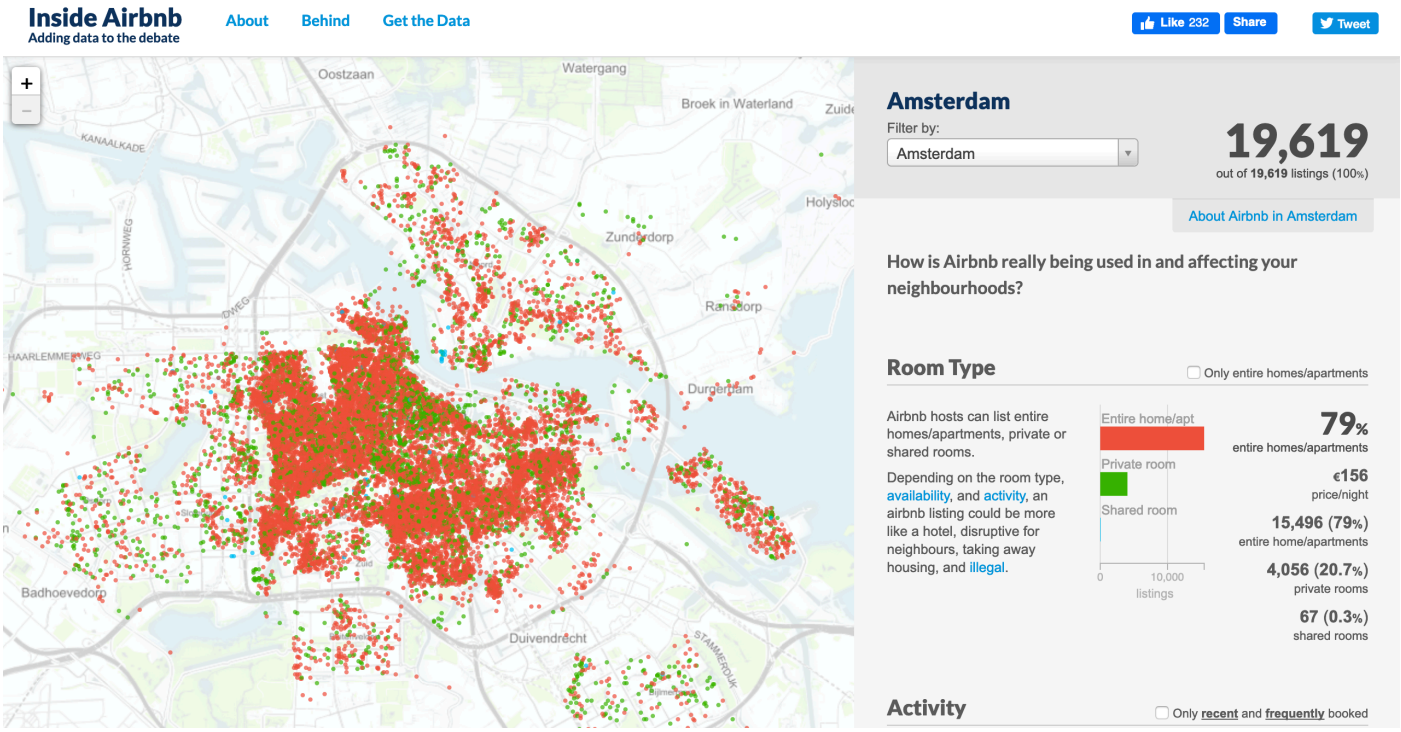


Figure 1 Screenshot inside Airbnb - Amsterdam

Restaurant data from FourSquare

Furthermore, neighborhood geographical data from the Amsterdam neighborhoods were needed. This data is available via the Amsterdam government data website: maps.amsterdam.nl. The dataset is downloaded as a geojson file for easy displaying on a Choropleth map.

To get an overview of currently settled restaurants in Amsterdam a 'nearby venues' lookup for every Amsterdam neighborhood was extracted from the FourSquare database using a developer account. foursquare.com/city-guide is a website to find the best places to eat, drink, shop, or visit in any city in the world. Access over 75 million short tips from local experts.

Approach / Data cleaning

Airbnb Data

Starting with the Airbnb data from InsideAirbnb I found almost 20.000 records. For performance- and visual reasons this needs to be cleaned up:

2404 hosts didn't have rating at all, meaning they were active. Let's remove them: (see fig.2)

```
In [7]: # Check for empty cells
        airbnb_data.isnull().sum()

Out[7]: id                                    0
        name                                 34
        host_id                               0
        host_name                           158
        neighbourhood_group               20025
        neighbourhood                         0
        latitude                              0
        longitude                             0
        room_type                             0
        price                                 0
        minimum_nights                        0
        number_of_reviews                     0
        last_review                        2404
        reviews_per_month                  2404
        calculated_host_listings_count        0
        availability_365                      0
        dtype: int64
```

Drop 2404 rows where never has been a review, assuming these are no active Airbnb hosts

Figure 2 2404 non active airbnb hosts

The dataset was still to large. Let's focus on the Price per night and see if we can clean up the outliers. (see fig.3)

Dataset is still too large and needs to be decreased

```
In [10]: # What about the distribution of 'price' ?

         # np.histogram returns 2 values
         count, bin_edges = np.histogram(airbnb_data['price'])

         print(count) # frequency count
         print(bin_edges) # bin ranges, default = 10 bins

         [17565    48     1     1     0     2     0     0     1     3]
         [   0.  900. 1800. 2700. 3600. 4500. 5400. 6300. 7200. 8100. 9000.]

In [11]: # 17000+ records left, so let's remove al the outliers above 400 euro per night
         indexNames = airbnb_data[airbnb_data['price'] > 400 ].index
         airbnb_data.drop(indexNames , inplace=True)
         airbnb_data.shape

Out[11]: (17253, 16)
```

Figure 3 Dataset still too large

We need to dive further into the price and check the distribution of it. Therefor a distribution plot was made (see fig.4). The majority is between 100 and 150 euro per night. Let's focus on the Airbnb hosts between 100 and 150 euro per night to decrease the dataset to a workable level of approximately 7000 records

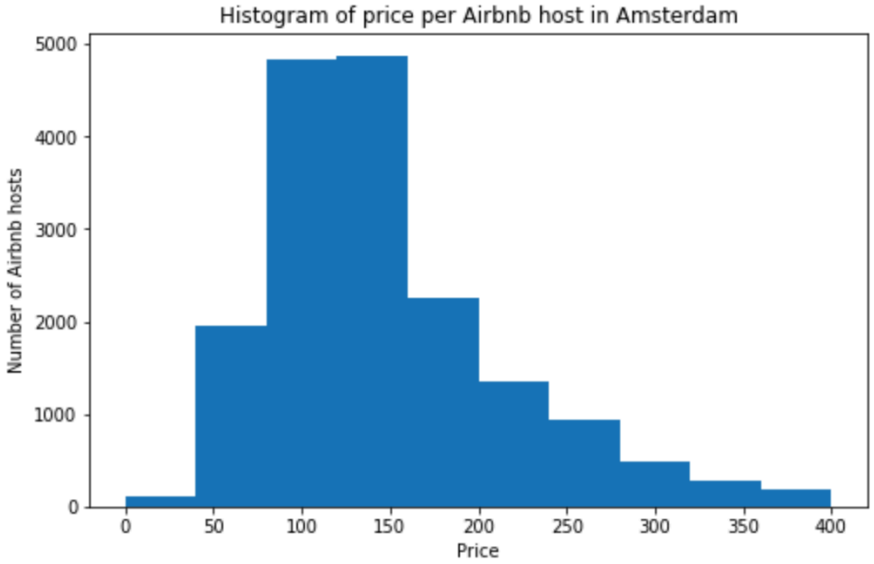
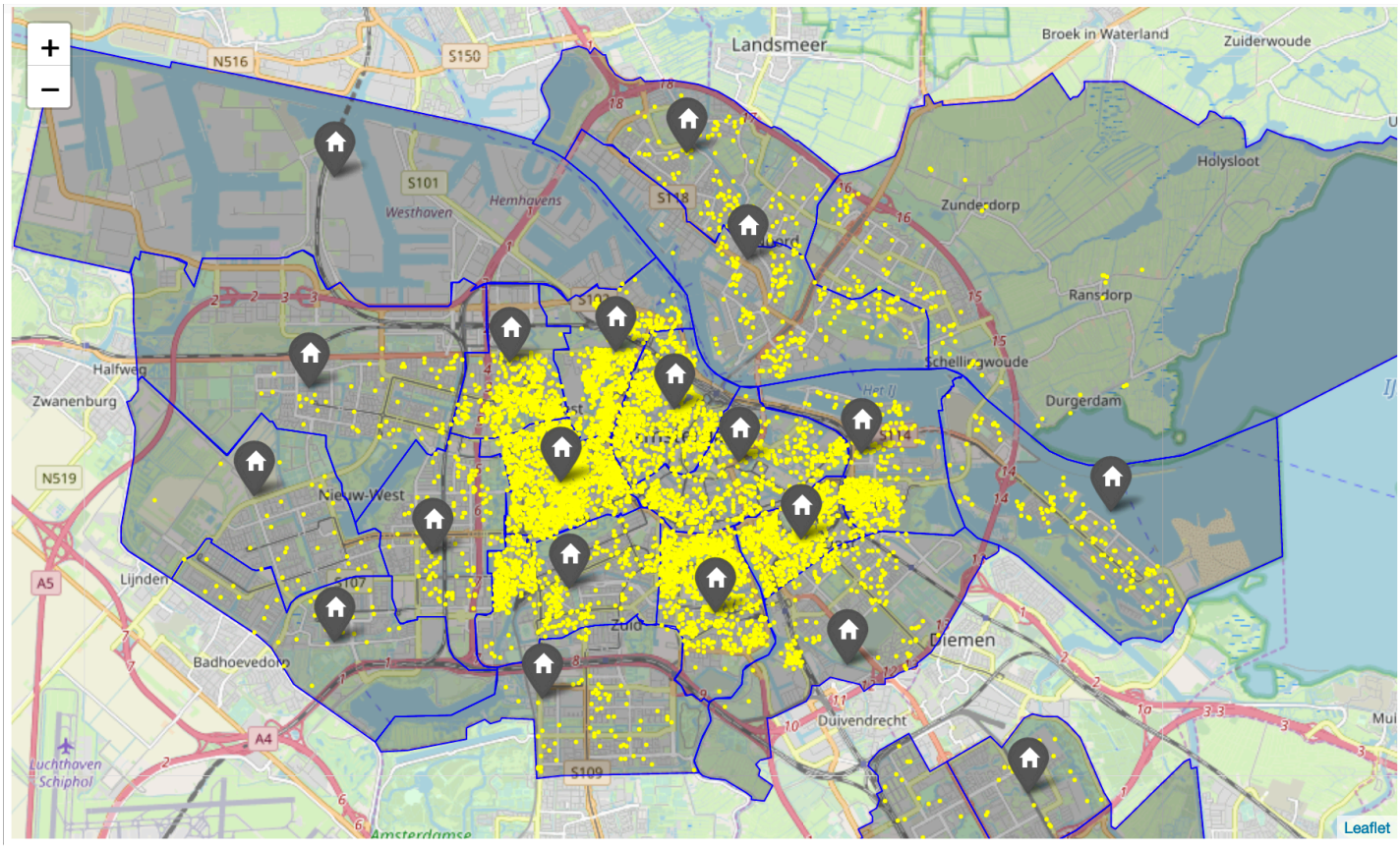


Figure 4 Histogram of price per Airbnb host in Amsterdam

Together with the geojson file with the neighborhood boundaries a map was created to plot all the ~7000 Airbnb hosts related to the neighborhoods where they reside. (see fig.5)



The yellow dots represent the Airbnb hosts, and the 'house' markers represent the Amsterdam Neigborhoods or Areas (Gebieden in Dutch)

Figure 5 Airbnb hosts plotted on the Amsterdam neighborhoods

FourSquare data cleaning

The export of the FourSquare database consists of 1889 venues (see fig.6)

```
In [24]: # How many venues were returned?
         print('{} venues were returned by Foursquare.'.format(amsterdam_venues.shape[0]))
         amsterdam_venues.head()

         1889 venues were returned by Foursquare.
```

Figure 6 Number of venues returned from FourSquare

These 1889 are all of the nearby venues, but we're interested in restaurants only. Before displaying the restaurants on a map a filter was written to extract only the Restaurants in Amsterdam. (see fig.7 and 8)

Create an array of all restaurants

```
In [28]: array = []
         search = 'Restaurant'
         for i in amsterdam_venues.venue_category :
             if search in i:
                 array.append(i)

         # and add Pizza place because this is also a restaurant
         array.append('Pizza Place')
```

Figure 7 Filter out all restaurants

Create a new dataset of all restaurants in Amsterdam

```
In [29]: amsterdam_restaurants = amsterdam_venues.loc[amsterdam_venues['venue_category'].isin(array)]
         amsterdam_restaurants.head()
```

Out[29]:

	neighborhood	neighborhood_latitude	neighborhood_longitude	venue	venue_latitude	venue_longitude	venue_category
4	Westpoort	52.411465	4.807319	KFC	52.427470	4.820170	Fast Food Restaurant
8	Westpoort	52.411465	4.807319	McDonald's	52.427396	4.820760	Fast Food Restaurant
19	Bijlmer-Oost	52.319564	4.976832	De Smeltkroes	52.322755	4.974752	South American Restaurant
22	Bijlmer-Oost	52.319564	4.976832	Pasta di Mamma	52.314779	4.955087	Italian Restaurant
24	Bijlmer-Oost	52.319564	4.976832	Margherita Tutta La Vita!	52.329054	4.955773	Pizza Place

Figure 8 New dataset with only restaurants in Amsterdam

And a plot of all restaurants together with the Airbnb hosts in the Amsterdam neighborhoods (see. fig.9)

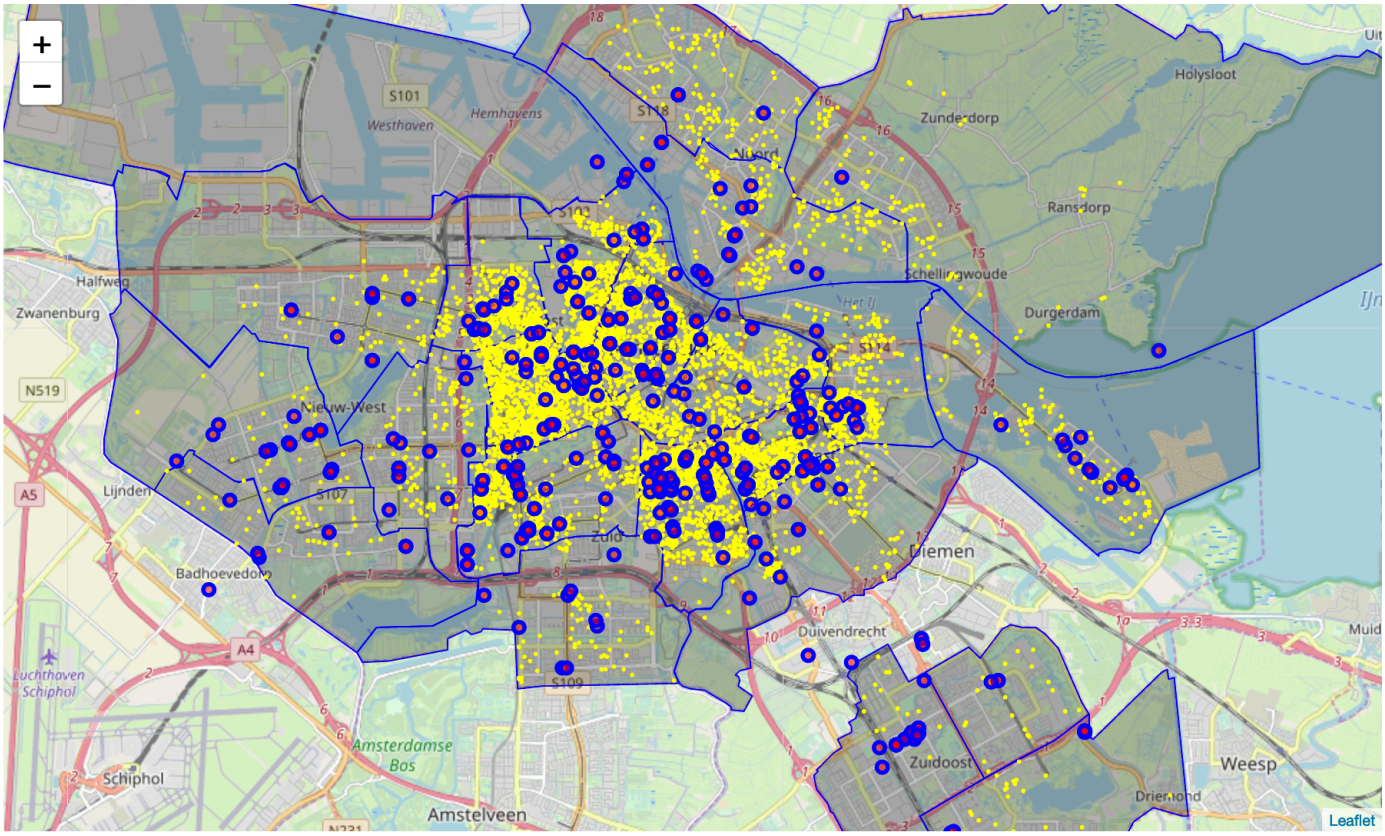


Figure 9 All restaurants plot on the map. Airbnb host in yellow, restaurants in blue

Methodology

Since we've plotted all the restaurants on the map a visual inspection did not answer the business problem. It looks like the restaurant should be opened somewhere in the center, not sure which neighborhood. We need more information before we can answer the business problem and need to analyze each neighborhood and cluster the restaurants.

To determine the number of clusters we use the Silhouette Coefficient Method. A higher Silhouette Coefficient score relates to a model with better-defined clusters. More information about the Silhouette Coefficient Score can be found on https://scikit-learn.org/stable/modules/clustering.html .

For this specific case the Silhouette Score was 3 (see fig.10)

```
In [33]: amsterdam_grouped_clustering = amsterdam_restaurant_grouped.drop('neighborhood', 1)

         for n_cluster in range(2, 10):
             kmeans = KMeans(n_clusters=n_cluster).fit(amsterdam_grouped_clustering)
             label = kmeans.labels_
             sil_coeff = silhouette_score(amsterdam_grouped_clustering, label, metric='euclidean')
             print("For n_clusters={}, The Silhouette Coefficient is {}".format(n_cluster, sil_coeff))

For n_clusters=2, The Silhouette Coefficient is 0.22844391429425503
For n_clusters=3, The Silhouette Coefficient is 0.25396818256927134
For n_clusters=4, The Silhouette Coefficient is 0.24056222210026956
For n_clusters=5, The Silhouette Coefficient is 0.21517349726221882
For n_clusters=6, The Silhouette Coefficient is 0.21888426417376594
For n_clusters=7, The Silhouette Coefficient is 0.1956549358008158
For n_clusters=8, The Silhouette Coefficient is 0.20301791030757166
For n_clusters=9, The Silhouette Coefficient is 0.19429071778259674
```

Figure 10 Silhouette Coefficient Score

Next we use K-means to generate the cluster labels and create a new dataframe based on the cluster labels (see fig.11)

```
In [35]: amsterdam_results = pd.DataFrame(kmeans.cluster_centers_)
         amsterdam_results.columns = amsterdam_grouped_clustering.columns
         amsterdam_results.index = ['cluster0','cluster1','cluster2']
         amsterdam_results['Total Sum'] = amsterdam_results.sum(axis = 1)
         amsterdam_results
```

Out[35]:

	American Restaurant	Argentinian Restaurant	Asian Restaurant	Australian Restaurant	Belgian Restaurant	Caribbean Restaurant	Chinese Restaurant	Con... Restau...
cluster0	0.100	-6.938894e-18	0.30	-5.551115e-17	-1.387779e-17	0.200	1.500	2.775558
cluster1	0.125	1.250000e-01	0.25	7.500000e-01	1.250000e-01	0.375	0.125	5.000000
cluster2	0.000	0.000000e+00	0.25	2.500000e-01	2.500000e-01	0.500	1.000	2.500000

Figure 11 Clustering results

The cluster with the lowest Total score will have our attention because this cluster is the less saturated cluster for restaurants and therefor the most suitable to solve our business problem.

To be able to plot the cluster results on a map together with the Airbnb data, the Results dataframe must be merged with the areas dataframe (see fig. 12)

```
In [40]: # Merge two dataframes
amsterdam_merged = df_areas

amsterdam_merged = amsterdam_merged.join(amsterdam_results_merged.set_index('neighborhood'), on='Gebied')

print(amsterdam_merged.shape)
amsterdam_merged
```

Figure 12 Merging the dataframes before plotting

The map will be discussed in the Result section of this report. After a visual inspection of the map we will create a new dataframe based on only the less saturated cluster. The other cluster are not on our interest anymore because they are too occupied to start a Mama Mia Pizza restaurant.

Next the Airbnb data will be integrated, and we will have a nice overview of the numbers of Airbnb hosts in all of the neighborhoods of the less saturated cluster

The neighborhood with the most Airbnb hosts in the 'winning' cluster will be appointed as the 'winning' neighborhood in the Battle of the Amsterdam Neighborhoods and will be the most suitable neighborhood for beginning a Mama Mia Pizza restaurant in Amsterdam.

Results

With a score of 13,70 cluster0 shows the less saturated restaurant market and is the most suitable place to begin a Mama Mia Pizza restaurant (see fig.13). But we need to dive further into detail because cluster0 consist of many neighborhoods (see fig.14)

```
In [35]: amsterdam_results = pd.DataFrame(kmeans.cluster_centers_)
         amsterdam_results.columns = amsterdam_grouped_clustering.columns
         amsterdam_results.index = ['cluster0','cluster1','cluster2']
         amsterdam_results['Total Sum'] = amsterdam_results.sum(axis = 1)
         amsterdam_results
```

Out[35]:

Satay estaurant	Scandinavian Restaurant	Seafood Restaurant	South American Restaurant	Southern / Soul Food Restaurant	Spanish Restaurant	Sushi Restaurant	Tapas Restaurant	Thai Restaurant	Turkish Restaurant	Vegetarian / Vegan Restaurant	Vietnamese Restaurant	Total Sum
0000e-01	0.20	0.200	0.200	-6.938894e-18	1.000000e-01	0.300	2.775558e-17	0.200	2.000	0.100	-6.938894e-18	13.70
8894e-18	0.00	0.875	0.125	-6.938894e-18	-6.938894e-18	0.125	5.000000e-01	1.125	0.375	1.125	1.250000e-01	25.75
0000e+00	0.25	0.750	0.250	2.500000e-01	0.000000e+00	0.000	2.500000e-01	1.250	1.500	0.500	0.000000e+00	25.75

Figure 13 Cluster scores

```
In [42]: amsterdam_merged[amsterdam_merged['Cluster_Labels'] == 0].reset_index(drop=True)
```

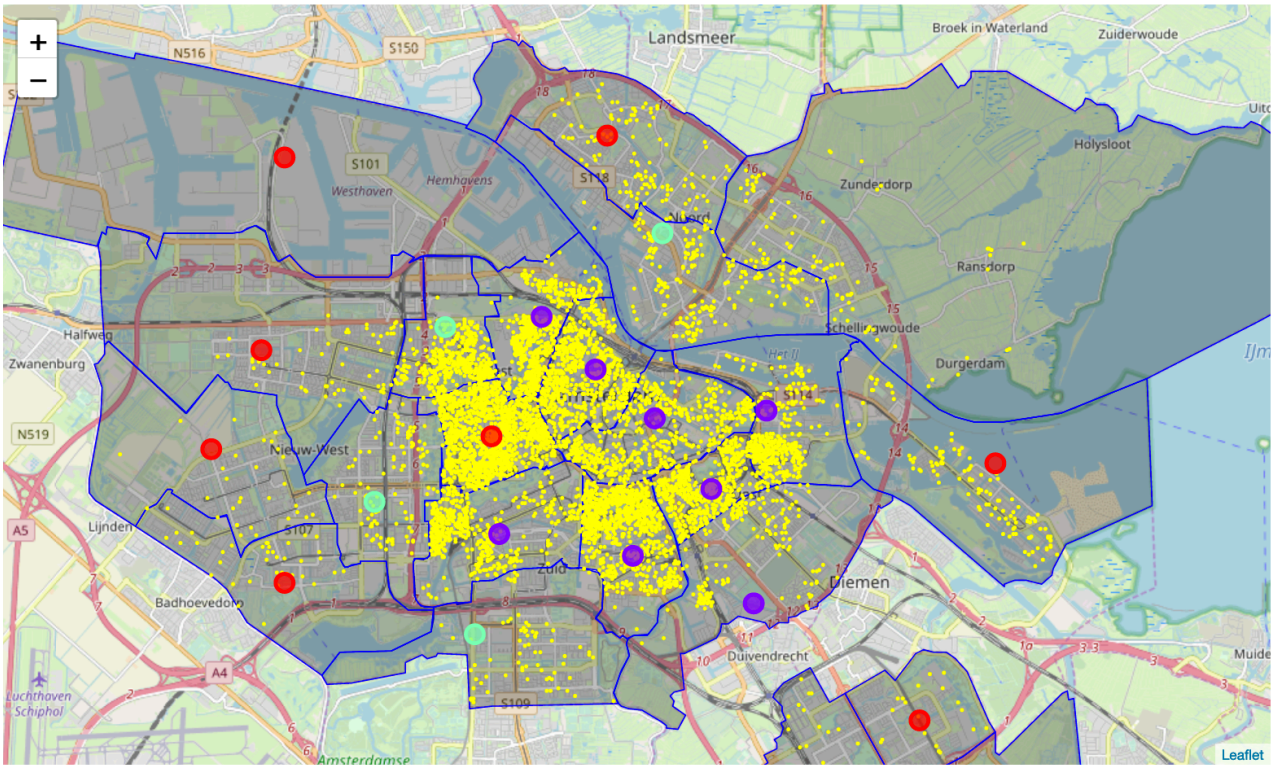
Out[42]:

	OBJECTNUMMER	Gebied_code	Gebied	Stadsdeel_code	Opp_m2	WKT_LNG_LAT	WKT_LAT_LNG	LNG	LAT	Unnamed: 9	Total
0	1	Dxxx	Westpoort	B	28991600	POLYGON((4.885861 52.39937,4.882702 52.401695,...	POLYGON((52.39937 4.885861,52.401695 4.882702,...	4.807319	52.411465	NaN	2
1	2	DX21	Bijlmer-Oost	T	4076020	POLYGON((4.996899 52.317636,4.996784 52.317753...	POLYGON((52.317636 4.996899,52.317753 4.996784...	4.976832	52.319564	NaN	20
2	3	DX06	Geuzenveld, Slotermeer, Sloterdijken	F	15357200	POLYGON((4.844813 52.385134,4.844798 52.385138...	POLYGON((52.385134 4.844813,52.385138 4.844798...	4.801156	52.379990	NaN	8
3	4	DX07	Osdorp	F	9174720	POLYGON((4.820437 52.351738,4.818066 52.356167...	POLYGON((52.351738 4.820437,52.356167 4.818066...	4.787646	52.363814	NaN	15
4	5	DX08	De Aker, Sloten, Nieuw-Sloten	F	7584560	POLYGON((4.84714 52.336896,4.847032 52.337685,...	POLYGON((52.336896 4.84714,52.337685 4.847032,...	4.807275	52.342135	NaN	21
5	8	DX17	Noord-West	N	8347620	POLYGON((4.930721 52.411601,4.930411 52.411936...	POLYGON((52.411601 4.930721,52.411936 4.930411...	4.893224	52.414954	NaN	8
6	10	DX05	Oud West, De Baarsjes	E	3327090	POLYGON((4.882097 52.362049,4.881886 52.362155...	POLYGON((52.362049 4.882097,52.362155 4.881886...	4.862335	52.365897	NaN	27
7	17	DX16	IJburg, Zeeburgereiland	M	13787000	POLYGON((5.039059 52.354569,5.038812 52.358098...	POLYGON((52.354569 5.039059,52.358098 5.038812...	4.997029	52.361586	NaN	15
8	22	DX20	Bijlmer-Centrum, Amstel III	T	8431710	POLYGON((4.971842 52.284355,4.970005 52.286362...	POLYGON((52.284355 4.971842,52.286362 4.970005...	4.950592	52.303143	NaN	16
9	23	DX22	Gaasperdam, Driemond	T	9605980	POLYGON((5.021546 52.302451,5.021464 52.303129...	POLYGON((52.302451 5.021546,52.303129 5.021464...	4.988479	52.304423	NaN	5

Figure 14 Cluster0 neighborhoods

And plot all the clusters on a map together with the Airbnb data (see fig. 15)

Out[41]:



Red markers: cluster0, Green markers: cluster1, Purple markers: cluster2

Figure 15 Map with Clusters and Airbnb data

Also, the Airbnb data needs to be cleaned up because we do not need the cluster1 and cluster2 data anymore (see fig.16)

In [44]:
```
# Drop all neighborhoods from cluster1 and cluster2
airbnb_cluster0.drop(airbnb_cluster0.loc[airbnb_cluster0['neighbourhood']=='Bos en Lommer'].index, inplace=True)
airbnb_cluster0.drop(airbnb_cluster0.loc[airbnb_cluster0['neighbourhood']=='Oud-Noord'].index, inplace=True)
airbnb_cluster0.drop(airbnb_cluster0.loc[airbnb_cluster0['neighbourhood']=='Slotervaart'].index, inplace=True)
airbnb_cluster0.drop(airbnb_cluster0.loc[airbnb_cluster0['neighbourhood']=='Oud-Zuid'].index, inplace=True)
airbnb_cluster0.drop(airbnb_cluster0.loc[airbnb_cluster0['neighbourhood']=='Westerpark'].index, inplace=True)
airbnb_cluster0.drop(airbnb_cluster0.loc[airbnb_cluster0['neighbourhood']=='Centrum-West'].index, inplace=True)
airbnb_cluster0.drop(airbnb_cluster0.loc[airbnb_cluster0['neighbourhood']=='Centrum-Oost'].index, inplace=True)
airbnb_cluster0.drop(airbnb_cluster0.loc[airbnb_cluster0['neighbourhood']=='Watergraafsmeer'].index, inplace=True)
airbnb_cluster0.drop(airbnb_cluster0.loc[airbnb_cluster0['neighbourhood']=='Buitenveldert - Zuidas'].index, inplace=Tru
airbnb_cluster0.drop(airbnb_cluster0.loc[airbnb_cluster0['neighbourhood']=='Zuid'].index, inplace=True)
airbnb_cluster0.drop(airbnb_cluster0.loc[airbnb_cluster0['neighbourhood']=='Oud-Oost'].index, inplace=True)
airbnb_cluster0.drop(airbnb_cluster0.loc[airbnb_cluster0['neighbourhood']=='De Pijp - Rivierenbuurt'].index, inplace=T
# And display it's shape
airbnb_cluster0.shape
```

Out[44]: (2245, 16)

Figure 16 Cleaning up Airbnb data by removing cluster1 and cluster2 neighborhoods

When grouping al information together and sort by the highest Airbnb hosts on top the following result appears (see fig.17:

In [47]:
```
airbnb_grouped = airbnb_cluster0.groupby('neighbourhood').count().reset_index()
```

In [49]:
```
# And display the airbnb data in relation to cluster0, sorted with the highest value on top
airbnb_grouped.sort_values(by=['id'], ascending=False)
```

Out[49]:

	neighbourhood	id	name	host_id	host_name	neighbourhood_group	latitude	longitude	room_type	price	minimum_nights	number_of_reviews
3	De Baarsjes - Oud-West	1351	1350	1351	1344	0	1351	1351	1351	1351	1351	1351
9	Oostelijk Havengebied - Indische Buurt	369	369	369	362	0	369	369	369	369	369	369
6	IJburg - Zeeburgereiland	123	123	123	123	0	123	123	123	123	123	123
8	Noord-West	122	122	122	121	0	122	122	122	122	122	122
7	Noord-Oost	88	88	88	88	0	88	88	88	88	88	88
5	Geuzenveld - Slotermeer	59	59	59	59	0	59	59	59	59	59	59
2	De Aker - Nieuw Sloten	40	40	40	40	0	40	40	40	40	40	40
10	Osdorp	29	29	29	29	0	29	29	29	29	29	29
4	Gaasperdam - Driemond	28	28	28	27	0	28	28	28	28	28	28
1	Bijlmer-Oost	21	21	21	21	0	21	21	21	21	21	21
0	Bijlmer-Centrum	15	15	15	15	0	15	15	15	15	15	15

Figure 17 Grouping all information together

Discussion

During my research I found that not all Foursquare information is plotted on the correct way. Maybe this has to do because Foursquare is not so popular in the Netherlands compared to the US. Probably in some cases wrong latitude and longitude coordinates were provided. As an example, 2 restaurants in the Noord-Oost area we're marked as 'IJburg, Zeeburger Eiland' and 'Oud-Noord' . It's also possible that the neighborhood information from Foursquare does not match the information provided by the Amsterdam city governance. Because the 'Noord-Oost' neighborhood data was not reliable I removed it. This has no influence on the results because it was too small.

Also, Airbnb does not provide any information directly. The site InsightAirbnb does of course and is scraping the Airbnb website in a smart way. I was not able to check if these figures are reliable, but even if the figures are reliable for 90% then it has no influence on the results.

I've also decreased the Airbnb dataset from about 20.000 to 7.000 to make it more workable due to computational power and visualization. Rendering 20.000+ Airbnb hosts, 500+ restaurants and 22 areas took a long time. I found 7.000 was better and did not influence the result at all.

I focused on restaurants in general, because this was the client request, but I would suggest further investigation on Pizza places itself. Although 'De Baarsjes – Oud-West' is not saturated with restaurants it's possible that there is a large percentage of Pizza places and it is advisable to move to the runner up 'Oostelijk Havengebied – Indische Buurt'.

Conclusion

Based on above investigation It's clear that the neighborhood 'De Baarsjes – Oud-West' is the most suitable neighborhood from Amsterdam to start a Mama Mia Pizza restaurant (see fig.17).

'De Baarsjes – Oud West' is part of clusterO, so the less saturated restaurant market and it has 1351 active Airbnb hosts.

The second-best neighborhood is 'Oostelijk Havengebied – Indische Buurt' with 369 airbnb hosts, on the other side of the center of Amsterdam. It's advisable to do further investigation to open another Mama Mia Pizza restaurant in the upcoming neighborhood.

Sources

https://en.wikipedia.org/wiki/Amsterdam

https://www.dw.com/en/how-amsterdam-is-fighting-mass-tourism/a-47806959

https://www.thepizzajoint.com/pizzafacts.html

http://data.insideairbnb.com/the-netherlands/north-holland/amsterdam/2019-12-07/visualisations/listings.csv

https://maps.amsterdam.nl/open_geodata/?k=204

https://foursquare.com/city-guide