Market niche for a vegetarian/vegan eatery in Madrid, Spain

Josias Belem Neto
josiasbelemneto@gmail.com
Capstone Project
Coursera IBM Data Science Certificate
April/2021

Introduction

Over the last few years the whole world has turn its attention to environmental issues, more than ever before. Every lifestyle choice we make affect the environment differently and in different degrees. We can clearly see signs our planet is suffering due to our impact on it. In certain areas we have more options to reduce this impact, such as eating habits, others not as much.

Searching for food with a lower environmental impact, together with the awareness of animal well-being and a more natural, less industrial, source of nutrients has been leading millions of people in the world to adopt a diet refereed as vegan or vegetarian. The offer for this type of diet is still limited but we can clearly see everywhere in the world a wave of these products being offered more and more often at supermarkets and restaurants.

The main objective of this study is to analyse how a possible eatery classified as vegan/vegetarian can find a possible market in any location on the planet. The city of Madrid, Spain was used as an example in this study. Since the coding created is general, it can be easily modified to find other possible market niches in any city, for example, a Mexican restaurant in Moscow or a Japanese restaurant in New Delhi.

Data

The API provider FourSquare was used to located market niches. FourSquare offers data based on location, keyword, user, among others. In this project the location one was used, this way more data could be collected for each area, then it was compared to find trends or gaps available in the market.

The API request made to FourSquare returns a json file with all the information about each venue; such as id, name, categories, latitude, longitude, address, crossing street, post code, etc. For this project id, categories and coordinates were used. Areas equally distributed around the city were created, then each venue retrieved was allocated the closest area.

With the data collected in a single data frame Pandas library was used manipulate the data to pinpoint where the restaurants and the vegetarian restaurants were located. Then further exploration of the data was carried out to draw more information from the dataset, such as market share and market niches.

Methodology Collecting Data

FourSquare only offers a limited number of searches per day with the basic profile (950) and each of these searches also have a limit on the amount of results returned (50-100). FourSquare uses the name of the place or its coordinates, latitude and longitude, as a parameter of each request.

The solution found for this problem was to create a grid over the area/city. This grid contains points equally distributed with each containing coordinates (latitude and longitude) and an area number. For this case study a grid of 14x14 points were created, distancing 500 meters between them. Here is the iteration process to create this dataset.

```
In [7]: r_lat=40000# radius of Earth for any given latitude span
        r_lon=r_lat*math.cos(math.radians(lat_searched))  # radius of Earth for given latitude span
        deg_lat=r_lat/360 # distance for each degree latitude
        deg_lon=r_lon/360 # distance for each degree longitude
        size=distance_btw_points*total_points # size of square in kms
        coordinates=[] # a list to store the results from the iteration
        area_num=0  # area number starting with 0
        lat_init=lat_searched-(size/2)/deg_lat# because the given value is in the center
        lon_init=lon_searched-(size/2)/deg_lon# an initial value needs to be calculated for the iteration
        for i in (np.linspace(lat_init,lat_init+(size/deg_lat),total_points)):
            for j in (np.linspace(lon_init,lon_init+(size/deg_lon),total_points)):
                coordinates.append([area_num,i,j])
                area_num=area_num+1

        coordinates=pd.DataFrame(coordinates)
        coordinates.columns=['area_num','lat','lon']
        coordinates.head()
```

Out[7]:

	area_num	lat	lon
0	0	40.38308	-3.748129
1	1	40.38308	-3.741797
2	2	40.38308	-3.735464
3	3	40.38308	-3.729131

Fig. 1

Below part of central Madrid after the grid was applied over it.

Fig. 2

Once this list was created a loop was carried out every coordinate, feeding these numbers (latitude and longitude) to FourSquare to request information. A single dataset was created with all the information fetched.

```
In [10]: ACCESS_TOKEN='NLUYF143PWP4CDMRREWYED2XX00SN4KWPA0UQ5LKAIZHVECD'
         CLIENT_ID='0HIRPSJC2WXNYWGK1EDTLMTINENVEL2NEQBR5GQWTCM3WGD4'
         CLIENT_SECRET='IMYBDXPYRBJ4ZVGWFVKSDL5QDSCRZJS4S2RBHTN2DKM1ALMB'
         LIMIT=200
         VERSION='20191220' # chosen date
```

The radius of search around each coordinate point will depend on how distance the waypoints are from each other. Radius will be equal to half distance between waypoints and because we need to cover the spaces between the circles created, we divide this value by 0.85 (sin 45 degrees).

```
In [11]: radius =1000*(distance_btw_points/2)/0.85 # radius in meters
         radius

Out[11]: 294.11764705882354

In [13]: dataframe=pd.DataFrame() # creating an empty dataframe to store all the results

In [14]: for area, lat,lon in zip(coordinates['area_num'],coordinates['lat'],coordinates['lon']):
             print('searching area: ', area )
             url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&ll={},{}&v={}&radius={}&limit={}'.format(
             CLIENT_ID, CLIENT_SECRET, lat, lon, VERSION, radius, LIMIT)
             results = requests.get(url).json()
             number_searches=number_searches+1
             try:
                 items = results['response']['groups'][0]['items']# getting the relevant part
                 items_df = pd.json_normalize(items)
                 items_df['area_num']=area # assigning the area number to the search
                 dataframe=pd.concat([dataframe,items_df])
             except:
                 print('area with problem: ', station)
         searching area:  205
```

Fig. 3

Below it's a screenshot of the initial data frame created before any modifications, it contained a total of 3559 rows.

```
In [170]: dataframe.head()
```

Out[170]:

	referralId	reasons.count	reasons.items	venue.id	venue.name	venue.location.lat	venue.location.lng	venue.location.labele
0	e-0-4d8ba29cace98cfae1285a9a-0	0.0	[{'summary': 'This spot is popular', 'type': '...	4d8ba29cace98cfae1285a9a	Polideportivo La Mina	40.381581	-3.746746	[{'label': 'di 40.3815812
1	e-0-4d82625c4bbaa0939a0bc1ac-1	0.0	[{'summary': 'This spot is popular', 'type': '...	4d82625c4bbaa0939a0bc1ac	Pastelería-Panadería La Gallega	40.384049	-3.746432	[{'label': 'di 40.3840490
2	e-0-4c82a9c72f1c236ad1e13b43-2	0.0	[{'summary': 'This spot is popular', 'type': '...	4c82a9c72f1c236ad1e13b43	Devinums	40.382068	-3.748143	[{'label': 'di 40.3820677
3	e-0-4cf1865b7bf3b60c9a4e607f-3	0.0	[{'summary': 'This spot is popular', 'type': '...	4cf1865b7bf3b60c9a4e607f	Pulpería Caracoleria	40.383472	-3.747245	[{'label': 'di 40.3834722

Fig. 4

The next step was to clean up the data frame and correct formatting issues to make the information ready to use. Also, during the loop search the same venue was picked up more than once because the circles used for an individual search overlap each other. Then these duplicate values (201) were then removed from the initial dataset and the final set contained 3358 rows. The criteria used was to keep the venues closest to each area coordinates. As a final step columns that were not relevant to the study were removed and the dataset was ready to be explored.

	name	categories	lat	lng	id	distance	area_num
0	Polideportivo La Mina	Athletics & Sports	40.381581	-3.746746	4d8ba29cace98cfae1285a9a	203.0	224
1	Pastelería-Panadería La Gallega	Bakery	40.384049	-3.746432	4d82625c4bbaa0939a0bc1ac	179.0	224
2	Devinums	Tapas Restaurant	40.382068	-3.748143	4c82a9c72f1c236ad1e13b43	112.0	224
3	Pulperia Caracoleria	Seafood Restaurant	40.383472	-3.747245	4cf1865b7bf3b60c9a4e607f	86.0	224
4	La Sala Live!	Concert Hall	40.383423	-3.747477	4b7c5023f964a520a78b2fe3	67.0	224

Fig. 5

Analysing the Data

Different analyses were carried out on the dataset using Pandas' features to easily filter, slice, gather information from large datasets; such as the methods 'groupby', 'sort_values', 'count' 'value_counts', amongst others. These were used in conjunction with Folium maps to display the results in a way that is very easy to grasp the meaning of these analyses.

With the initial dataset created, cleaned and with the relevant columns filtered a first analysis was carried out to locate the eateries in the dataset. FourSquare has many categories for eateries besides restaurants. So a list with all these 'extra categories' was first created to be used in a filter. Here is a part of this list called 'categories_list':

```
In [92]: categories_list=[ 'Coffee Shop','Pub','Eastern European Restaurant','Fast Food Restaurant', 'Creperie', 'Brewery',  'Diner', 'Br
         'Gastropub', 'Café', 'Middle Eastern Restaurant','Bakery','Churrascaria' ,'Falafel Restaurant', 'Argentinian Restaurant'
         'Moroccan Restaurant', 'Mediterranean Restaurant','Thai Restaurant', 'Pizza Place', 'Sushi Restaurant',
        'Japanese Restaurant','Persian Restaurant', 'Fish & Chips Shop','Bar', 'Brasserie', 'Indian Restaurant',
         'Italian Restaurant', 'Portuguese Restaurant',  'Lebanese Restaurant',
          'Pastry Shop', 'Halal Restaurant',  'Korean Restaurant','Modern European Restaurant','Chinese Restaurant',
          'Burger Joint', 'Greek Restaurant','Turkish Restaurant', 'Caribbean Restaurant', 'Spanish Restaurant',
          'Polish Restaurant','Tapas Restaurant', 'American Restaurant', 'Seafood Restaurant',
```

Fig. 6

This filter was applied to the initial data frame resulting in a data set with all the eateries in Madrid called 'eateries_df'.

```
In [93]: eateries_df=dataframe_df[dataframe_df['categories'].isin(categories_list)].reset_index()
         eateries_df.drop('index',axis=1,inplace=True)
```

Let's take a look at this dataframe. Now we are ready to extract some useful information from it.

```
In [94]: eateries_df.head()
```

Out[94]:

	name	categories	lat	lng	id	distance	area_num
0	Pastelería-Panadería La Gallega	Bakery	40.384049	-3.746432	4d82625c4bbaa0939a0bc1ac	179.0	0
1	Devinums	Tapas Restaurant	40.382068	-3.748143	4c82a9c72f1c236ad1e13b43	112.0	0
2	Pulperia Caracoleria	Seafood Restaurant	40.383472	-3.747245	4cf1865b7bf3b60c9a4e607f	86.0	0
3	Mulligan's	Pub	40.382348	-3.747841	4d8f7a12fa943704f0a11bc6	85.0	0
4	Cafeteria La Joya	Bakery	40.380701	-3.748496	4d20a69a5acaa35d5287c935	266.0	0

Fig. 7

From this data frame, a quick glimpse at the most frequent (10) categories of eateries in Madrid in percentage using the method 'value_counts'.

```
In [99]: 100*eateries_df['categories'].value_counts(normalize=True).head(10)
```

```
Out[99]: Spanish Restaurant          16.446701
         Restaurant                  11.573604
         Bar                          7.664975
         Tapas Restaurant             7.411168
         Café                         5.126904
         Coffee Shop                  4.365482
         Bakery                       3.857868
         Italian Restaurant           3.401015
         Pizza Place                  2.893401
         Mediterranean Restaurant     2.690355
         Name: categories, dtype: float64
```

Fig. 8

Also, from the main dataset, a new one called ' veg_df ' was created just for the vegetarian eateries.

```
In [102]: vegetarian_df=eateries_df[eateries_df['categories'].str.contains('Vegetarian')]
```

```
In [103]: vegetarian_df.head()
```

Out[103]:

	name	categories	lat	lng	id	distance	area_num
437	El Triángulo de las Verduras	Vegetarian / Vegan Restaurant	40.413453	-3.728349	4e0397de45ddb464557678c5	173.0	93
455	Gauranga Trascendental Food	Vegetarian / Vegan Restaurant	40.412892	-3.713486	519e20e4498e0bb2bb4de285	270.0	95
479	Viva Burger	Vegetarian / Vegan Restaurant	40.412630	-3.711683	4b9d4647f964a520319f36e3	148.0	96
484	pura vida vegan bar	Vegetarian / Vegan Restaurant	40.410208	-3.709029	57adcfbc498e0d546e3f1bdb	221.0	96
516	La Encomienda	Vegetarian / Vegan Restaurant	40.410672	-3.705982	57a09a99498edffb1cba9db1	237.0	97

Fig. 9

As mentioned, Folium was used to display the information extracted from the data frames, here the location of each vegetarian eatery in Madrid.

```
In [106]: veg_map=folium.Map(location=[lat_searched, lon_searched], zoom_start=13)
          sw = vegetarian_df[['lat', 'lng']].min().values.tolist()
          ne = vegetarian_df[['lat', 'lng']].max().values.tolist()
          veg_map.fit_bounds([sw, ne])
          for lat,lon,name in zip(vegetarian_df['lat'],vegetarian_df['lng'],vegetarian_df['name']):
              folium.CircleMarker(
                  [lat,lon],
                  radius=10,
                  color='black',
                  fill = True,
                  fill_color = 'green',
                  fill_opacity = 0.5
                  ).add_to(veg_map)

          veg_map
```

Out[106]:

Fig.10

The information was then grouped in each area and two new data frames were created: one for the number of eateries per area ('eateries_area_df') to locate the busier dinning zones in the city and another for number of vegetarian eateries per area (eateries_area_df). A glimpse of these datasets are pictured below:

Eateries in each area

Let's explore a bit more and see how many eateries are located in each area:

```
In [107]: eateries_area_df=pd.DataFrame(eateries_df.groupby(by='area_num')['name'].count())
          eateries_area_df.reset_index(inplace=True)

In [108]: eateries_area_df.columns=['area_num','num_eateries']

In [109]: eateries_area_df=pd.merge(eateries_area_df,coordinates,on='area_num', how='inner')

In [110]: eateries_area_df.sort_values(by='num_eateries', ascending=False).head(20)
```

Out[110]:

	area_num	num_eateries	lat	lon
84	98	59	40.412009	-3.697467
147	175	46	40.436116	-3.684802
111	131	43	40.421651	-3.678469

Fig. 11

Vegetarian Eateries in each area

Let's now see in which areas the vegetarian/vegan eateries are located:

```
In [112]: veg_area_df=pd.DataFrame(vegetarian_df['area_num'].value_counts())
          veg_area_df.reset_index(inplace=True)

In [113]: veg_area_df.columns=['area_num','num_veg']

In [114]: veg_area_df=pd.merge(veg_area_df,coordinates,on='area_num', how='inner')
```

Taking a glimpse at the stations with most number of vegetarian/vegan places we have:

```
In [115]: veg_area_df.head(5)
```

Out[115]:

	area_num	num_veg	lat	lon
0	127	2	40.421651	-3.703800
1	128	2	40.421651	-3.697467
2	96	2	40.412009	-3.710133

Fig. 12

A further analysis was done to compare the proportion (in percentage) of the number of vegetarian eateries to the total number of eateries in each area. Initially the data sets were combined into 'comparison_df':

```
[154]: comparison_df=pd.merge(eateries_area_df,veg_area_df,on=['area_num','lat','lon'], how='outer')
       comparison_df.fillna(0, inplace=True)
       comparison_df=comparison_df[['area_num','lat','lon','num_eateries','num_veg']]
       comparison_df.head()
```

[154]:

	area_num	lat	lon	num_eateries	num_veg
0	0	40.38308	-3.748129	5	0.0
1	1	40.38308	-3.741797	3	0.0
2	2	40.38308	-3.735464	1	0.0
3	3	40.38308	-3.729131	4	0.0
4	4	40.38308	-3.722798	1	0.0

Fig. 13

Then an extra column was added to show the percentage of vegetarian eateries, when sorted the 'vegetarian hot spots' in Madrid could be seen.

```
In [151]: comparison_df.head()
```

Out[151]:

	area_num	lat	lon	num_eateries	num_veg	veg_share
79	93	40.412009	-3.729131	7	1.0	14.29
155	187	40.440937	-3.703800	8	1.0	12.50
99	116	40.416830	-3.678469	8	1.0	12.50
83	97	40.412009	-3.703800	12	1.0	8.33
118	140	40.426473	-3.716466	13	1.0	7.69

Fig. 14

The final step was to see where the business opportunities are located. A new column was created ('opportunity') to hold the result of the relationship between number of eateries ('num_eateries') and the number of vegetarian eateries ('num_veg'). However the number of eateries in many areas is zero, so the value of 1 was added to every row in the column 'num_veg' to make the division possible, then it was removed from the result. The final dataset was sorted by descending order using the column 'opportunity'.

```
In [167]: comparison_df['opportunity']=(comparison_df['num_eateries']/(comparison_df['num_veg']+1))-1
          comparison_df.sort_values(by='opportunity',ascending=False,inplace=True)
          comparison_df.reset_index(inplace=True)

In [168]: comparison_df.head()
```

Out[168]:

	index	area_num	lat	lon	num_eateries	num_veg	veg_share	opportunity
0	84	98	40.412009	-3.697467	59	0.0	0.0	58.0
1	147	175	40.436116	-3.684802	46	0.0	0.0	45.0
2	111	131	40.421651	-3.678469	43	0.0	0.0	42.0
3	130	155	40.431294	-3.716466	42	0.0	0.0	41.0
4	180	219	40.450580	-3.691134	41	0.0	0.0	40.0

Fig. 15

Results

Distribution of eateries in the Madrid. Each dot represents one venue. The eateries seem evenly distributed with some localised agglomeration.

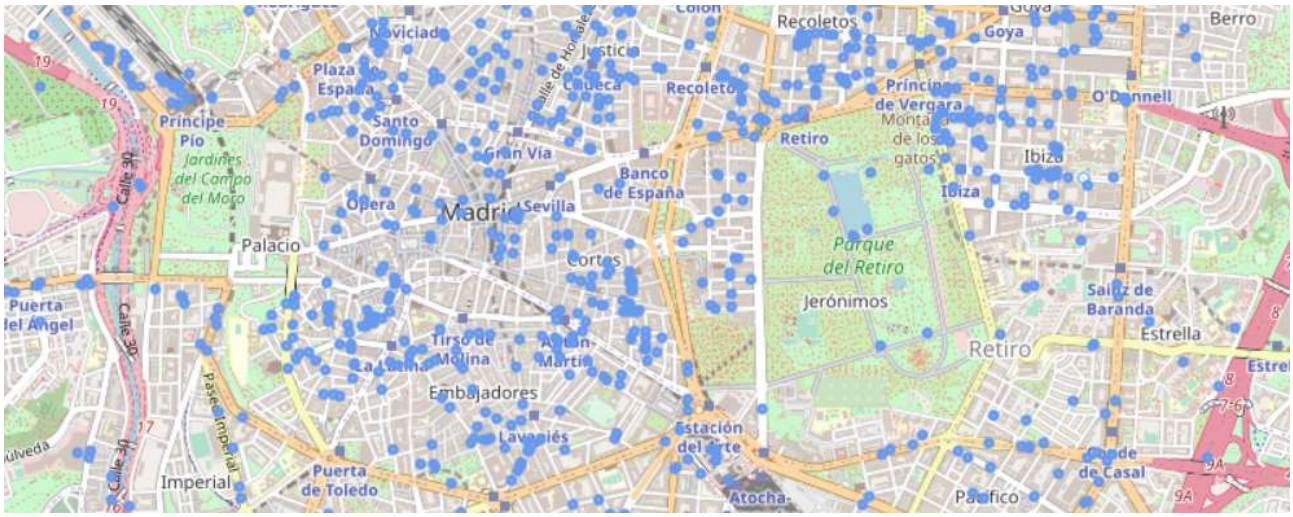


Fig. 16

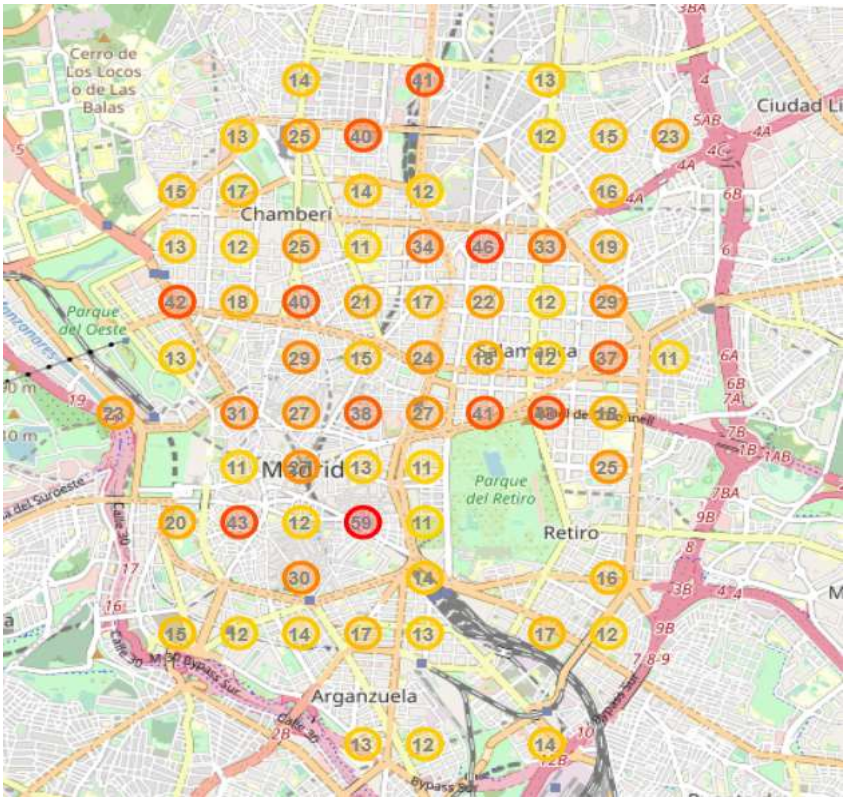


Fig. 17

Number of eateries per area searched. Each area is represented by a circle containing the number of eateries in the zone. The colour depicts the results, ranging from yellow (less dense) to red (more dense). The areas with less than 10 eateries were removed from the map to improve visibility.
This map shows clearly the higher concentration of eateries in certain zones.

Tables showing the top 10 categories under which the eateries are classified. It shows 'Spanish', 'Bar', 'Tapas' and 'Cafe' in the top results, unsurprisingly the habits of the population of Madrid.

Spanish Restaurant	324
Restaurant	228
Bar	151
Tapas Restaurant	146
Café	101
Coffee Shop	86
Bakery	76
Italian Restaurant	67
Pizza Place	57
Mediterranean Restaurant	53

Fig. 18
Numbers in absolute values

Spanish Restaurant	16.446701
Restaurant	11.573604
Bar	7.664975
Tapas Restaurant	7.411168
Café	5.126904
Coffee Shop	4.365482
Bakery	3.857868
Italian Restaurant	3.401015
Pizza Place	2.893401
Mediterranean Restaurant	2.690355

Fig. 19
Numbers in percentage of total

Spanish Restaurant	324
Restaurant	228
Tapas Restaurant	146
Italian Restaurant	67
Mediterranean Restaurant	53
Japanese Restaurant	38
Seafood Restaurant	36
Asian Restaurant	33
Chinese Restaurant	28
Mexican Restaurant	28
Fast Food Restaurant	25
Sushi Restaurant	17
Argentinian Restaurant	17
Vegetarian / Vegan Restaurant	16
American Restaurant	13
Indian Restaurant	12
Thai Restaurant	10
Peruvian Restaurant	10
Korean Restaurant	8
French Restaurant	7

Fig. 20

This table shows the total number of venues classified as restaurants and its distribution. Only the top 20 results are shown. The 'Vegetarian/Vegan' category was listed as the 14th most popular category.

Map showing the location of the vegetarian eateries in Madrid. Two results are not shown in the picture (located further away) to make the visualisation of the map better. Most of the venues are are located in small clusters in the city centre.

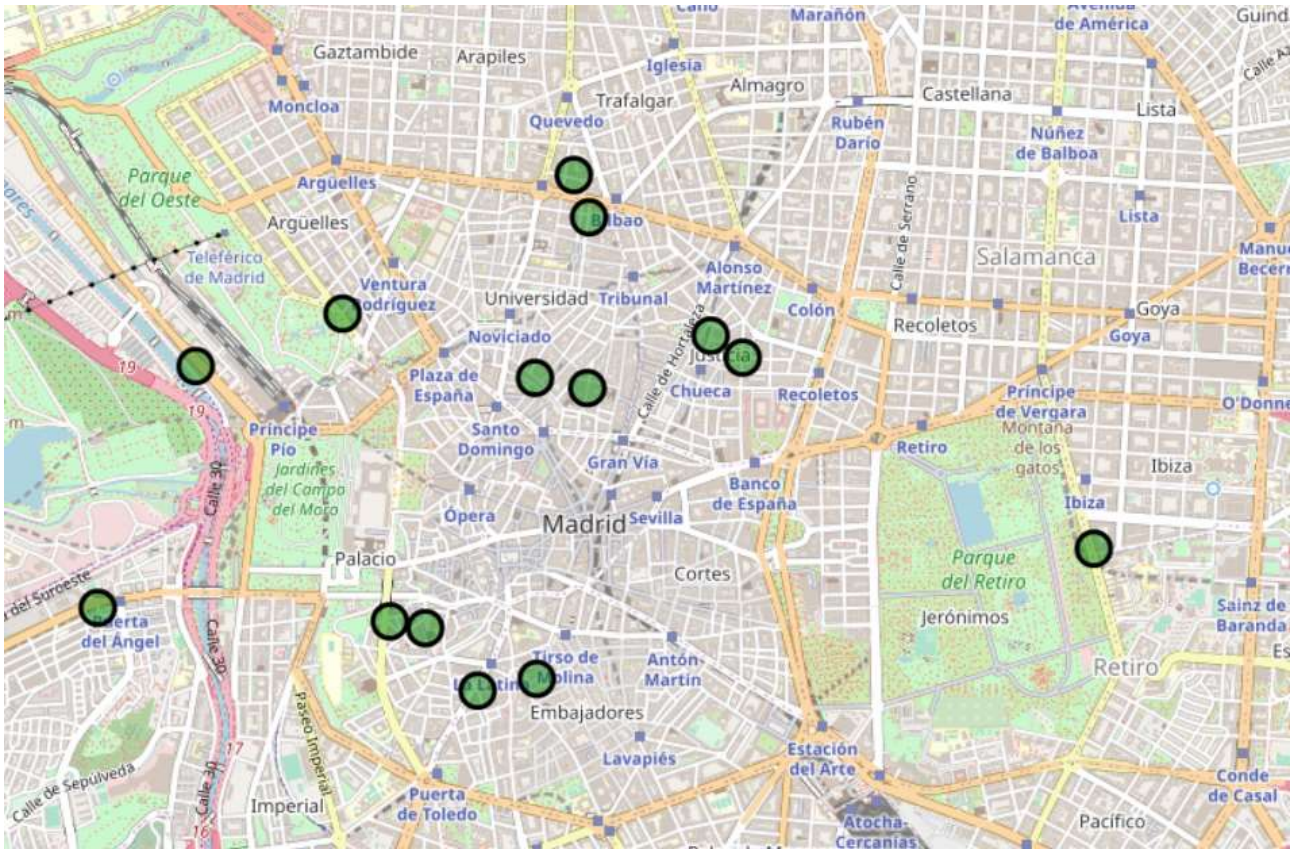


Fig. 21

Map showing the total number of vegetarian eateries in Madrid. Only areas with more than one venue were shown. All the other areas don't have any vegetarian venues, representing a possible market niche.

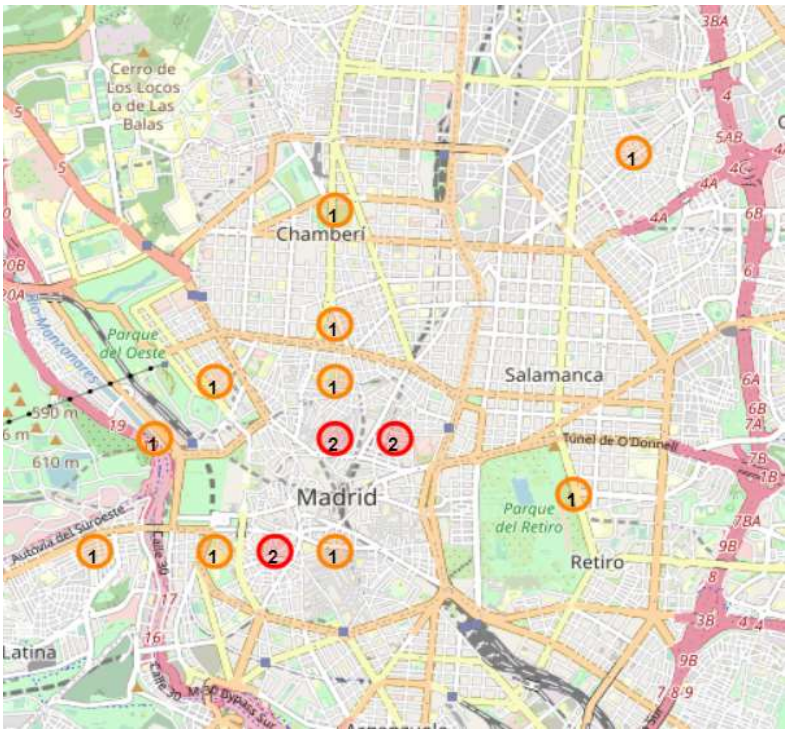
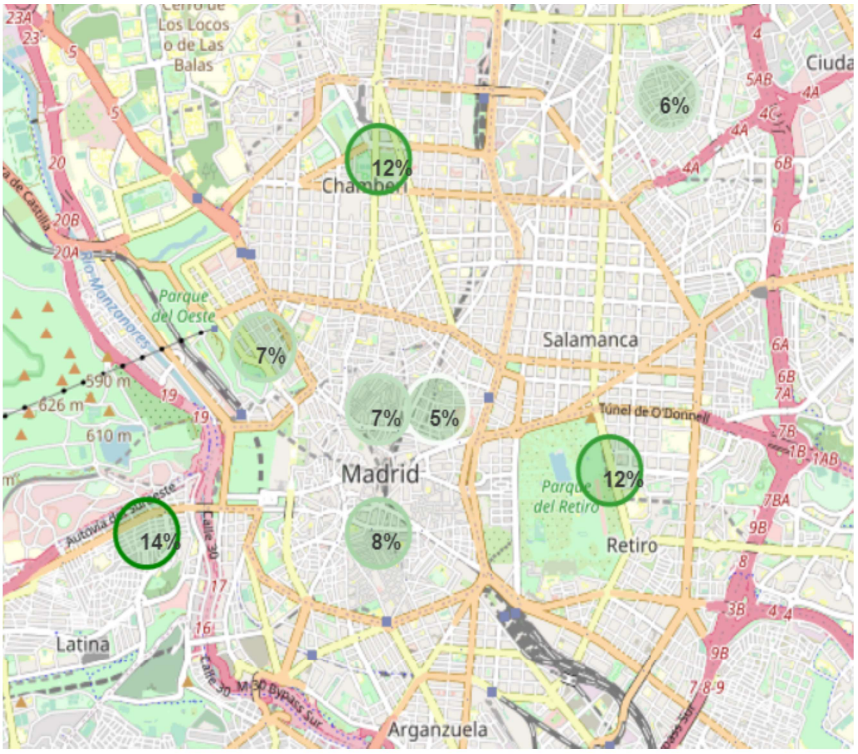


Fig. 22



Map showing the percentage of eateries classified as 'vegan' or 'vegetarian' compared to the total number of eateries in each area.
Only areas with 5% of more were depicted.

Fig.23

Map showing the top areas with the most business opportunities for a vegan/vegetarian place in Madrid (areas 1,2,3,4,5,8,9 and 10). The map below displays the other areas.

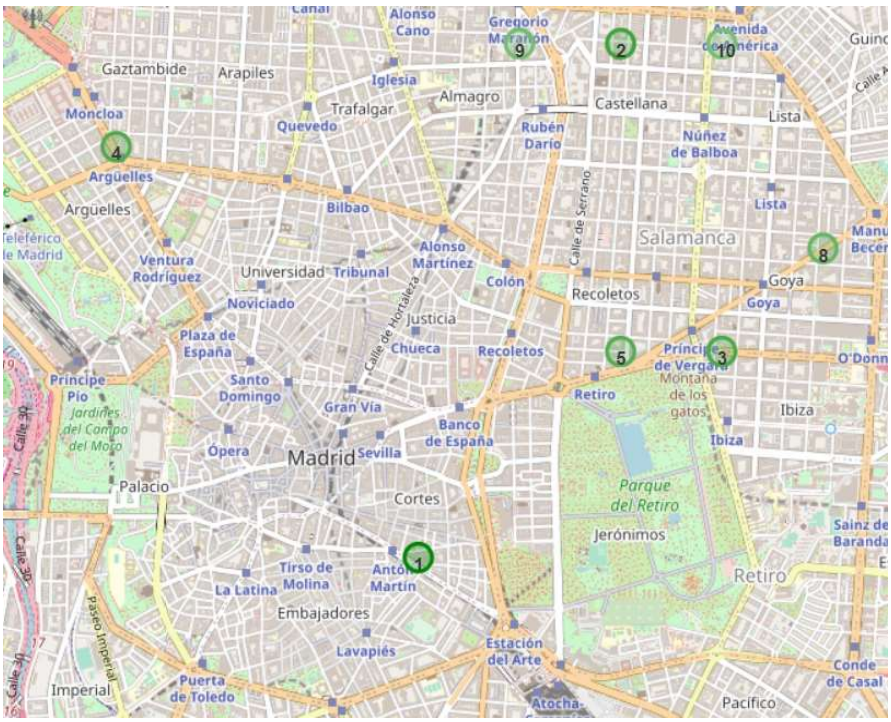


Fig.24

Map showing the top areas with the most business opportunities for a vegan/vegetarian place in Madrid (areas 2,4,6,7,9 and 10). The map above displays the other areas.

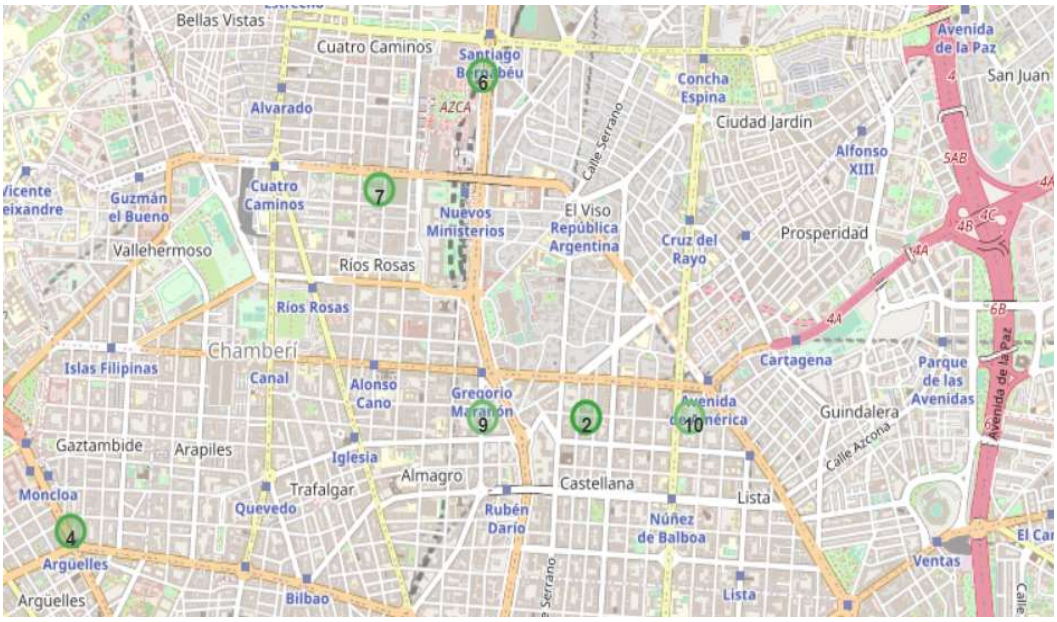


Fig.25

Discussion

Madrid is a vibrant city where the dinning culture is intertwined in its inhabitants' lifestyle. The initial finding in this study is that the eateries are spread quite evenly around the city, with some denser concentration in some central areas (fig. 16 & fig 17).

Spanish strong eating habits can be clearly seen exploring the top categories of eateries in the city (fig. 18 & 19), where 40% of eateries comprise traditional establishments such as Spanish restaurants, tapas bars, bars and cafes. This reinforces the accuracy of the data used in this study.

When the number of purely vegetarian/vegan venues were analysed only 16 eateries were located, placing this category in number 14th (fig. 20) of popularity in Madrid. But they are located in cluster in mostly central areas (fig. 21 & 22) giving rise to some vegetarian hotspots in the city (fig. 23), where this category of venue represents a larger proportion when compared to the total number of eateries.

Being a capital that embraces new trends very rapidly, a market niche is clearly seen in Madrid, where many areas that have lots of restaurants and no vegetarian venues. From the figures 24 and 25, these top 10 locations are in order of a possible market gap. The figures give us a quantitative analysis. But we can explore these areas a bit further and find some similarities amongst them:

- Anton Martin (area 1) : a central area with vibrant nightlife young population;
- Arguelles (area 4): an area very close to the university, with lots of young people;
- areas 2,9 and 10: many offices with business people eating out very often;
- areas 3,5 and 8: residential upscale area;
- areas 7 and 6: many offices with business people eating out very often.

Conclusion

The project started the main difficulty of the limitation of the free API requests from FourSquare. When using a place name as a parameter input the return would be very general and leave many areas of the city uncovered. To overcome this problem a loop was created to iterate over different coordinates in the city to gather all the data needed and combine it into one single data frame.

However this also represented a problem, which coordinates to use? Initially data for the underground stations was used to create the neighbourhoods coordinates in Madrid to gather data from FourSquare. However many gaps were left in the map and some areas were not evenly represented. The idea of the grid with equally spaced coordinates solved this problem and added a benefit to the code, expanding the its usage to other areas of the city or any other place.

Having lived in Madrid myself for a few years I do agree with the quantitative result found in this study; those are neighbourhoods/areas where a vegetarian place could see some potential market. This study should definitely be used in conjunction with other market analysis for future investments.

Very few lines of code were used to get to these conclusions. Simple yet very powerful this study could also be used in any place on the planet that is covered by FourSquare to find market niches which are not being fulfilled. Instead of a vegetarian eatery in Madrid a study of a burger place in Tokyo could be done or a gym in Cape Town. This could be achieved changing very few lines of code. A further analysis could be done, the development over time of a single category in a single place simply by changing the date parameter when requesting data to FourSquare. However I wanted to give this project one single scope to keep it simple to demonstrate it can be meaningful.