NEW VETERINARY CLINIC IN MADRID

The Battle of the Neighbourhoods
Applied Data Science Capstone by IBM/Coursera

Eva Porras

Madrid, September 2019

1. Introduction

For a veterinarian who decides to open a new veterinarian clinic in Madrid, Spain it is very important to know the distribution of pets and veterinarian clinics in the neighbourhoods of Madrid to find an optimal location for the new facility.

We will analyze the number of pets per neighborhood and the number of veterinarians to detect those neighborhoods with the highest ratios of pets per veterinarian (i.e. with more needing of locating a new facility).

On the other hand, it is also important to check the ascending trend of the number of pets in Madrid in the last five years and analyze those boroughs with highest increase ratios.

With these two criteria we will cluster the neighbourhoods to detect the most promising neighborhoods to install a new veterinary clinic (and the most "saturated" neighbourhood to avoid).

2. Data acquisition and cleaning

Based on the definition of the problem, factors that will influence our decision are:

- increase in the number of pets in the last 5 years

- number of pets in the neighbourhood,

- number of veterinary clinics in the neighbourhood

2.1. Name and location of the boroughs and neighbourhoods of Madrid

City Hall Public Data web publishes the street guide including numbering of all urban premises (206,866 premises). We will extract the name of the boroughs, neighbourhoods and their location.

In [3]:
```
# From the street directoty of the city, it is extracted the list of boroughs, neighbourhoods and location
df=pd.read_csv(project.get_file('CALLEJERO_VIGENTE_NUMERACIONES_201908.csv'), sep=';', encoding = "ISO-8859-1")
df.head(5)
```

Out[3]:

	Codigo de numero	Codigo de via	Clase de la via	Partícula de la vía	Nombre de la vía	Literal de numeracion	Codigo de distrito	Nombre del distrito	Codigo de barrio	Nombre del barrio	Seccion censal	Codigo postal	Secci carter
0	31031089	31001337	AUTOVIA		A-1	KM.001000EN	8	FUENCARRAL-EL PARDO	6	VALVERDE	166	28050	ND
1	31031088	31001337	AUTOVIA		A-1	KM.001000SA	16	HORTALEZA	6	VALDEFUENTES	119	28050	ND
2	31031091	31001337	AUTOVIA		A-1	KM.001100EN	8	FUENCARRAL-EL PARDO	6	VALVERDE	171	28050	ND
3	31031090	31001337	AUTOVIA		A-1	KM.001100SA	16	HORTALEZA	6	VALDEFUENTES	125	28050	ND
4	31031093	31001337	AUTOVIA		A-1	KM.001200EN	8	FUENCARRAL-EL PARDO	6	VALVERDE	167	28050	ND

The dataframe shape is 206866 rows and 26 columns.

In [4]:
```
# it is a very big dataframe
df.shape
```

Out[4]: (206866, 26)

From the dataframe we extract only the 6 required columns, we group the dataframe by neighbourhoods and transform coordinates to decimal format to finally obtain the name of the boroughs, neighbourhoods and their location.

In [9]:
```python
#grouping by neighbourhood and reordering columns
df=df.groupby(['Neighbourhood']).mean()
df=df.reset_index()
df=df.join(codigos_distrito.set_index('Borough Code'), on='Borough Code')
columnas=df.columns.tolist()
columnas= columnas[1:2] + columnas[-1:]+ columnas[2:3]+columnas[0:1]+columnas[3:5]
df=df[columnas]
df=df.sort_values(['Borough Code', 'Neighbourhood Code'])
df=df.reset_index(drop=True)
df
```

Out[9]:

	Borough Code	Borough	Neighbourhood Code	Neighbourhood	Longitude	Latitude
0	1	CENTRO	1	PALACIO	-3.711270	40.414430
1	1	CENTRO	2	EMBAJADORES	-3.702885	40.409738
2	1	CENTRO	3	CORTES	-3.697530	40.414064
3	1	CENTRO	4	JUSTICIA	-3.697507	40.423651
4	1	CENTRO	5	UNIVERSIDAD	-3.706039	40.425119
5	1	CENTRO	6	SOL	-3.704920	40.416991
6	2	ARGANZUELA	1	IMPERIAL	-3.718203	40.407663
7	2	ARGANZUELA	2	ACACIAS	-3.706246	40.401900
8	2	ARGANZUELA	3	CHOPERA	-3.698228	40.395881
9	2	ARGANZUELA	4	LEGAZPI	-3.689785	40.388806

In [10]:
```python
# calculate number of boroughs and neighbourhoods
print('Madrid has {} boroughs and {} neighbourhoods.'.format(
        len(df['Borough'].unique()),
        df.shape[0]
    )
)
```

```
Madrid has 21 boroughs and 131 neighbourhoods.
```

Once we have the name of the boroughs, neighbourhoods and their location, we can plot the map of the neighbourhoods of the city of Madrid.

Map of Madrid Neighbourhoods

In [11]:
```python
address = 'Madrid'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Madrid are {}, {}.'.format(latitude, longitude))
```

```
The geograpical coordinate of Madrid are 40.4167047, -3.7035825.
```

In [12]:
```python
# create map of Madrid using latitude and longitude values
map_madrid = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighbourhood in zip(df['Latitude'], df['Longitude'], df['Borough'], df['Neighbourhood']):
    label = '{}, {}'.format(neighbourhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_madrid)

map_madrid
```

2.2. Number of pets increase and distribution per neighbourhood

City Hall Public Data web publishes the number of dogs and cats per borough of the last 5 years.

In [20]:
```
# In the file "Censo animales.csv" we have the total number of pets per Borough and the incr
pets=pd.read_csv(project.get_file('Censo animales .csv'), sep=';', encoding = "ISO-8859-1")
names = list(pets.columns)
names[0]='Year'
names[1]='Borough_code'
names[2]='Borough'
names[3]='Dogs'
names[4]='Cats'
pets.columns = names
pets['Total_pets']=pets['Dogs']+pets['Cats']

pets.head(30)
```

Out[20]:

	Year	Borough_code	Borough	Dogs	Cats	Total_pets
0	2018	2	ARGANZUELA	10622	4458	15080
1	2018	21	BARAJAS	5203	1300	6503
2	2018	11	CARABANCHEL	20265	5524	25789
3	2018	1	CENTRO	15881	8186	24067
4	2018	5	CHAMARTÍN	11417	3601	15018
5	2018	7	CHAMBERÍ	13615	4087	17702
6	2018	15	CIUDAD LINEAL	17375	7226	24601
7	2018	8	FUENCARRAL-EL PARDO	17645	5558	23203
8	2018	16	HORTALEZA	15965	7797	23762
9	2018	10	LATINA	19282	7990	27272

We extract the increase of the number of total pets per borough and year and calculate the increase in the last 5 years.

In [49]:
```python
#pivot table to analyse the number of pets per borough and year
pets=pets.pivot_table('Total_pets',['Borough_code','Borough'],'Year')
pets=pets.rename_axis(None, axis=1).reset_index()

# calculation of the increase in the number of pets per borough in the last five years
pets['Inc_5_y (%)']=(pets[2018]/pets[2014]-1)*100
pets=pets.round({'Inc_5_y (%)': 1})


pets
```

Out[49]:

	Borough_code	Borough	2014	2015	2016	2017	2018	Inc_5_y (%)
0	1	CENTRO	16745	17045	21634	22527	24067	43.7
1	2	ARGANZUELA	11448	11728	13793	14476	15080	31.7
2	3	RETIRO	10118	10274	10244	10622	10728	6.0
3	4	SALAMANCA	14735	15019	16133	17002	17832	21.0
4	5	CHAMARTÍN	14735	15019	14568	15017	15018	1.9
5	6	TETUÁN	14479	14706	15851	16434	17241	19.1
6	7	CHAMBERÍ	15771	16115	16440	17228	17702	12.2
7	8	FUENCARRAL-EL PARDO	20823	21124	22259	23062	23203	11.4
8	9	MONCLOA-ARAVACA	13946	14270	14968	15514	15915	14.1
9	10	LATINA	20639	21124	22297	23606	27272	32.1
10	11	CARABANCHEL	19885	20428	23003	24227	25789	29.7
11	12	USERA	12288	12633	14037	14470	14894	21.2

We obtain that the number of pets in Madrid is around 370,000 and that the increase in the las 5 years is the 22%.

In [24]:
```python
print('There are {} registered pets in Madrid.'.format(
         pets['Total_pets'].sum()
     )
)

print('The increase of the number of pets in the last 5 years in Madrid is {}%.'.format(
         round(pets['Inc_5_y (%)'].mean(),1)
     )
)
```

```
There are 370018 registered pets in Madrid.
The increase of the number of pets in the last 5 years in Madrid is 22.0%.
```

On the other hand, since the number of pets is published per borough, we use the proportion of population to distribute the number of pets per neighbourhood. For that, we extract the population per borough and per neighbourhood from the City Hall Public Data web:

In [25]:
```python
# as we have the number of pets per borough, we use the proportion of population to distribute the number of pets
# per neighbourhood. First we download the population file from the City Hall Public web
population=pd.read_csv(project.get_file('Rango_Edades_Seccion_201908.csv'), sep=';', encoding = "ISO-8859-1")
names = list(population.columns)
names[0]='Borough_code'
names[1]='Borough'
names[2]='Neighbourhood_code'
names[3]='Neighbourhood'
names[8]='SpanishMen'
names[9]='SpanishWomen'
names[10]='OtherMen'
names[11]='OtherWomen'
population.columns = names
population=population.fillna(0)
population['Total_Pop_Neighbourhood']=population['SpanishMen']+population['SpanishWomen']+population['OtherMen']+population['OtherWomen']
population.head()
```

Out[25]:

	Borough_code	Borough	Neighbourhood_code	Neighbourhood	COD_BARRIO	COD_DIST_SECCION	COD_SECCION	COD_EDAD_INT	SpanishMen	SpanishWomen	OtherMen	OtherWomen	Total_Pop_Neighbourhood
0	1	CENTRO	101	PALACIO	1	1001	1	0	3.0	2.0	0.0	1.0	6.0
1	1	CENTRO	101	PALACIO	1	1001	1	1	3.0	1.0	1.0	2.0	7.0
2	1	CENTRO	101	PALACIO	1	1001	1	2	1.0	1.0	1.0	0.0	3.0
3	1	CENTRO	101	PALACIO	1	1001	1	3	3.0	3.0	1.0	1.0	8.0
4	1	CENTRO	101	PALACIO	1	1001	1	4	2.0	3.0	1.0	1.0	7.0

The population per neighbourhood and per borough is extracted from the population dataframe:

In [26]:
```
# the population per neighbourhood
pop_neigh=population.pivot_table('Total_Pop_Neighbourhood',['Borough_code','Borough','Neighbourhood_code','Neighbourhood'],aggfunc='sum')
pop_neigh=pop_neigh.rename_axis(None, axis=1).reset_index()
pop_neigh
```

4	1	CENTRO	105	UNIVERSIDAD	32991.0
5	1	CENTRO	106	SOL	7508.0
6	2	ARGANZUELA	201	IMPERIAL	22907.0
7	2	ARGANZUELA	202	ACACIAS	36958.0
8	2	ARGANZUELA	203	CHOPERA	20208.0
9	2	ARGANZUELA	204	LEGAZPI	19784.0
10	2	ARGANZUELA	205	DELICIAS	28155.0
11	2	ARGANZUELA	206	PALOS DE MOGUER	26171.0
12	2	ARGANZUELA	207	ATOCHA	1194.0
13	3	RETIRO	301	PACIFICO	33789.0
14	3	RETIRO	302	ADELFAS	18825.0
15	3	RETIRO	303	ESTRELLA	23327.0

In [27]:
```
# the population per borough
pop_bor=population.pivot_table('Total_Pop_Neighbourhood',['Borough_code'],aggfunc='sum')
pop_bor=pop_bor.rename_axis(None, axis=1).reset_index()
pop_bor.rename(columns={'Total_Pop_Neighbourhood': 'Total_Pop_Bor'}, inplace=True)
pop_bor
```

Out[27]:

	Borough_code	Total_Pop_Bor
0	1	139037.0
1	2	155377.0
2	3	120239.0
3	4	147273.0
4	5	147435.0
5	6	160421.0
6	7	140289.0
7	8	249159.0
8	9	120849.0

Finally, we use the proportion of population to distribute the number of pets per neighbourhood and obtain the final pets dataframe including Borough, coordinates, total number of pets and the increase in the number of pets in the last five years.

In [29]:
```
# distribution of pets per neighbourhood
pets=pop_neigh.set_index('Borough_code').join(pets.set_index('Borough_code') )
pets=pets.rename_axis(None, axis=1).reset_index()
pets.rename(columns={'Total_pets': 'Total_Pets_Borough'}, inplace=True)
pets['Total_Pets_Neighbourhood']=pets['Total_Pets_Borough']/pets['Total_Pop_Bor']*pets['Total_Pop_Neighbourhood']
pets=pets.round({'Total_Pets_Neighbourhood': 0})
pets=pets.drop('Borough',axis=1)
pets=pets.set_index('Neighbourhood').join(df.set_index('Neighbourhood') )
pets=pets.rename_axis(None, axis=1).reset_index()
pets=pets.iloc[:, [0,9,11,12,7,5]]

pets
```

Out[29]:

	Neighbourhood	Borough	Longitude	Latitude	Total_Pets_Neighbourhood	Inc_5_y (%)
0	PALACIO	CENTRO	-3.711270	40.414430	4055.0	43.7
1	EMBAJADORES	CENTRO	-3.702885	40.409738	8070.0	43.7
2	CORTES	CENTRO	-3.697530	40.414064	1857.0	43.7
3	JUSTICIA	CENTRO	-3.697507	40.423651	3074.0	43.7
4	UNIVERSIDAD	CENTRO	-3.706039	40.425119	5711.0	43.7
5	SOL	CENTRO	-3.704920	40.416991	1300.0	43.7
6	IMPERIAL	ARGANZUELA	-3.718203	40.407663	2223.0	31.7

2.3. Number of veterinary clinic in every neighbourhood

City Hall Public Data web publishes the active venues in Madrid (163,251 venues).

In [51]:
```python
# download venues from the City Hall Data Web
vets=pd.read_csv(project.get_file('OPEN DATA Locales-Epigrafes201907.csv'), sep=';', encoding = "ISO-8859-1", low_memory=False)
vets.head()
```

Out[51]:

	id_local	id_distrito_local	desc_distrito_local	id_barrio_local	desc_barrio_local	cod_barrio_local	id_seccion_censal_local	desc_seccion_censal_local	coordenada_x_local	coordenada
0	270315185	20	SAN BLAS-CANILLEJAS	2003	AMPOSTA	NaN	NaN	34	447255,56	447
1	270315223	15	CIUDAD LINEAL	1504	CONCEPCION	NaN	NaN	114	444279,58	447
2	270315300	11	CARABANCHEL	1105	PUERTA BONITA	NaN	NaN	119	0	
3	270315331	21	BARAJAS	2104	TIMON	NaN	NaN	30	450276,58	448
4	270315335	5	CHAMARTIN	503	CIUDAD JARDIN	NaN	NaN	45	442828,59	447

In [52]:
```python
vets.shape
```

Out[52]: (163251, 46)

We select only the neighbourhood and the activity columns and we count the number of veterinary clinics per neighbourhood.

In [31]:
```python
# counting the veterinary clinic per neighbourhood
vets.rename(columns={'desc_barrio_local': 'Neighbourhood','desc_division': 'Activity' }, inplace=True)
vets=vets[['Neighbourhood','Activity' ]]
vets.head()
vets=vets[vets['Activity'].str.contains("VETERINARIAS")==True]
vets=vets['Neighbourhood'].value_counts()
vets=pd.DataFrame(vets)
vets=vets.rename_axis(None, axis=1).reset_index()
names = list(vets.columns)
names[0]='Neighbourhood'
names[1]='Number_of_Vets'
vets.columns = names
vets
```

Out[31]:

	Neighbourhood	Number_of_Vets
0	GUINDALERA	13
1	CANILLAS	10
2	PUEBLO NUEVO	10
3	PACIFICO	9
4	ALUCHE	9
5	ARAVACA	7
6	VALVERDE	7
7	ACACIAS	7
8	PEÑA GRANDE	6

2.4. Final Dataframe

As final dataframe we have per each neighbourhood:

- Borough,
- Location: Latitude and Longitude,
- Total Number of pets,
- Increase of the number of pets in the last 5 years,
- Number of veterinarian clinics and
- Ratio pets/vet

In [32]:

```
# joining pets per neighbourhood + vets data
final_df=pets.set_index('Neighbourhood').join(vets.set_index('Neighbourhood') )
final_df=final_df.rename_axis(None, axis=1).reset_index()
final_df.head()
final_df=final_df.fillna(0)
final_df['pets/vet'] = 0
condition = final_df['Number_of_Vets'] > 0
final_df.loc[condition, 'pets/vet'] = final_df['Total_Pets_Neighbourhood']/final_df['Number_of_Vets']
final_df.loc[~condition, 'pets/vet'] = final_df['Total_Pets_Neighbourhood']
final_df=final_df.round({'pets/vet': 0})
final_df
```

Out[32]:

	Neighbourhood	Borough	Longitude	Latitude	Total_Pets_Neighbourhood	Inc_5_y (%)	Number_of_Vets	pets/vet
0	PALACIO	CENTRO	-3.711270	40.414430	4055.0	43.7	6.0	676.0
1	EMBAJADORES	CENTRO	-3.702885	40.409738	8070.0	43.7	5.0	1614.0
2	CORTES	CENTRO	-3.697530	40.414064	1857.0	43.7	3.0	619.0
3	JUSTICIA	CENTRO	-3.697507	40.423651	3074.0	43.7	1.0	3074.0
4	UNIVERSIDAD	CENTRO	-3.706039	40.425119	5711.0	43.7	5.0	1142.0
5	SOL	CENTRO	-3.704920	40.416991	1300.0	43.7	1.0	1300.0
6	IMPERIAL	ARGANZUELA	-3.718203	40.407663	2223.0	31.7	3.0	741.0
7	ACACIAS	ARGANZUELA	-3.706246	40.401900	3587.0	31.7	7.0	512.0
8	CHOPERA	ARGANZUELA	-3.698228	40.395881	1961.0	31.7	3.0	654.0
9	LEGAZPI	ARGANZUELA	-3.689785	40.388806	1920.0	31.7	2.0	960.0
10	DELICIAS	ARGANZUELA	-3.691106	40.396839	2733.0	31.7	2.0	1366.0

(Note: I have decided not to use the Foursquare API since not so much venues are registered in application yet. A checking of this is included in the code)

3. Clustering Model

Once we have obtained the Final Dataframe with the ratio of pets/vet and the increase in the last five years, we will cluster the neighbourhoods to detect the most promising neighborhoods to install a new veterinary clinic (and the most "saturated" neighbourhood to avoid).

The algorithm that we are going to use is K-means clustering. K-means can group data only unsupervised based on the similarity of the neighbourhoods to each other.

There are various types of clustering algorithms such as partitioning, hierarchical, or density based clustering. K-means is a type of partitioning clustering. It divides the data into k non-overlapping subsets or clusters without any cluster internal structure or labels. This means, it's an unsupervised algorithm. Objects within a cluster are very similar and objects across different clusters are very different or dissimilar.

3.1. Determination of the number of clusters K

In partitioning clustering, such as k-means clustering, the user has to specify the number of clusters k to be generated. We have used the elbow method to determine the optimal number of clusters.

Elbow method: the idea is compute k-means clustering for different values of k. For each k, calculate the inertia (sum of squared distances of samples to their closest cluster center) and plot the curve of inertia depending of the number of clusters k. The location of the elbow in the plot is considered as the appropriate number of clusters (i.e. after the elbow, adding another cluster does not improve the inertia very much).

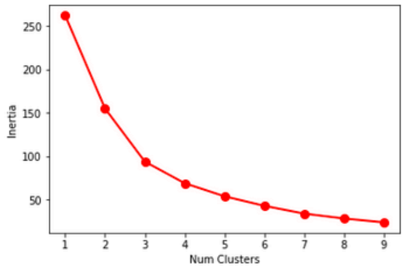
Application of Elbow Method to determine the optimal K number

In [34]:
```
# optimize number of clusters
ks = 10
inertia_clusters = list()

for i in range(1,ks):
        # Object KMeans
        kmeans=KMeans(n_clusters=i, random_state=0).fit(madrid_grouped_clustering)

        # Obtain inertia
        inertia_clusters.append([i, kmeans.inertia_])
```

In [53]:
```
import matplotlib.pyplot as plt
x, y = zip(*[inertia for inertia in inertia_clusters])
plt.plot(x, y, 'ro-', markersize=8, lw=2)
plt.xlabel('Num Clusters')
plt.ylabel('Inertia')
plt.show()
```



According to the plot, the optimal number of clusters is set in 4

According to the Elbow method, the optimal number of clusters is 4.

3.2. K-means clustering

Once we have determined the optimal number of clusters, we apply the k-meand clustering model to our Dataframe.

In [36]:
```
# set number of clusters
kclusters = 4

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

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:10]
```

Out[36]:
```
array([3, 3, 3, 0, 3, 3, 3, 3, 3, 3], dtype=int32)
```

In [37]:
```
# add clustering labels

madrid_grouped_sorted=final_df

madrid_grouped_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

madrid_grouped_sorted
```

Out[37]:

	Cluster Labels	Neighbourhood	Borough	Longitude	Latitude	Total_Pets_Neighbourhood	Inc_5_y (%)	Number_of_Vets	pets/vet
0	3	PALACIO	CENTRO	-3.711270	40.414430	4055.0	43.7	6.0	676.0
1	3	EMBAJADORES	CENTRO	-3.702885	40.409738	8070.0	43.7	5.0	1614.0
2	3	CORTES	CENTRO	-3.697530	40.414064	1857.0	43.7	3.0	619.0
3	0	JUSTICIA	CENTRO	-3.697507	40.423651	3074.0	43.7	1.0	3074.0
4	3	UNIVERSIDAD	CENTRO	-3.706039	40.425119	5711.0	43.7	5.0	1142.0
5	3	SOL	CENTRO	-3.704920	40.416991	1300.0	43.7	1.0	1300.0

4. Results

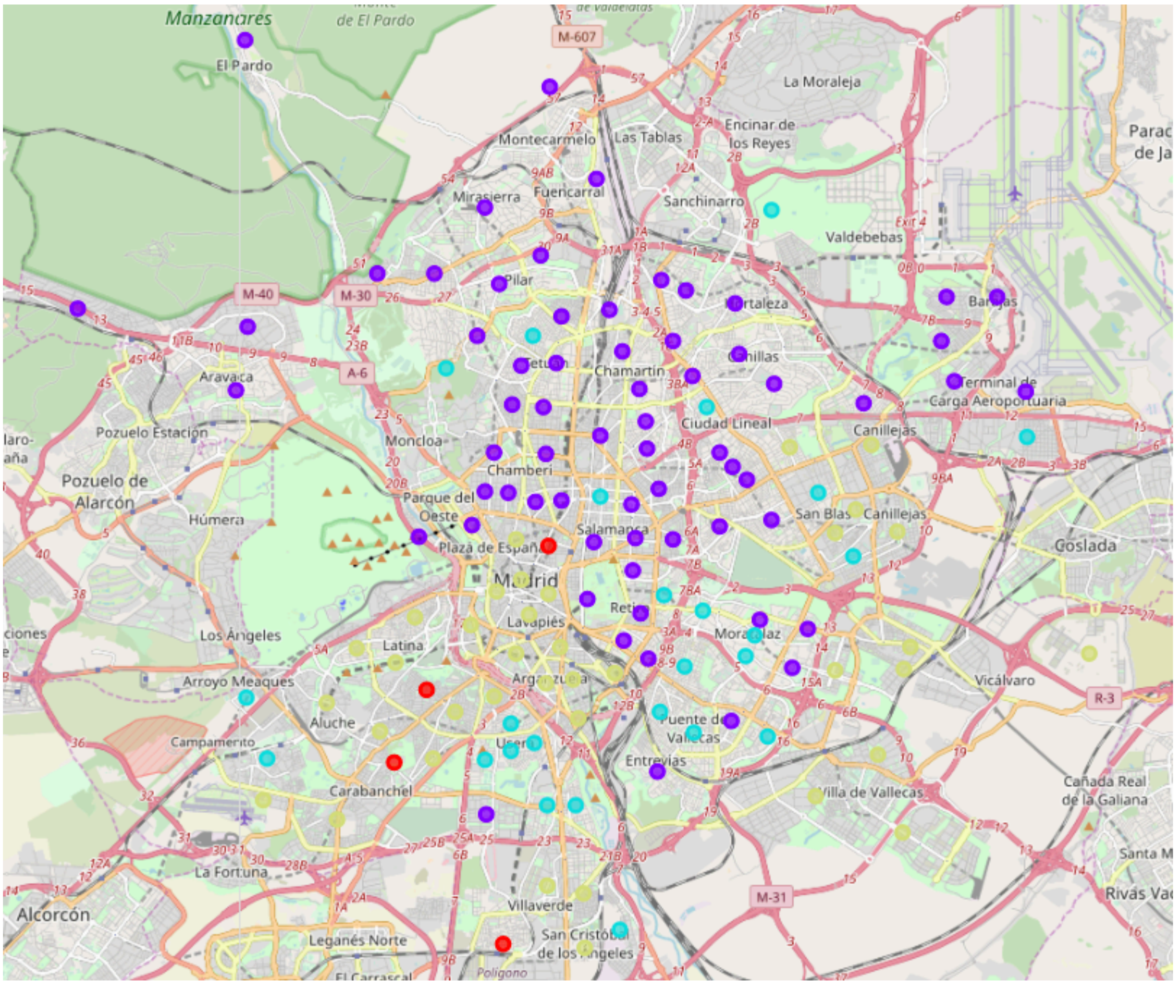
We plot the neighbourhoods in each cluster in the map of Madrid:

```
In [38]: # create map
         map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

         # 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(madrid_grouped_sorted['Latitude'], madrid_grouped_sorted['Longitude'],
             label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
             folium.CircleMarker(
                 [lat, lon],
                 radius=5,
                 popup=label,
                 color=rainbow[int(cluster)-1],
                 fill=True,
                 fill_color=rainbow[int(cluster)-1],
                 fill_opacity=0.7).add_to(map_clusters)

         map_clusters
```



The clustering model results in:

Cluster 0

Cluster 0

In [39]:
```
madrid_cluster_0=madrid_grouped_sorted.loc[madrid_grouped_sorted['Cluster Labels'] == 0, madrid_grouped_sorted.columns
[[1] + list(range(5, madrid_grouped_sorted.shape[1]))]]
madrid_cluster_0
```

Out[39]:

	Neighbourhood	Total_Pets_Neighbourhood	Inc_5_y (%)	Number_of_Vets	pets/vet
3	JUSTICIA	3074.0	43.7	1.0	3074.0
67	SAN ISIDRO	3966.0	29.7	1.0	3966.0
69	PUERTA BONITA	3555.0	29.7	1.0	3555.0
106	VILLAVERDE ALTO C.H.	4550.0	32.6	0.0	4550.0

In [40]:
```
madrid_cluster_0.describe()
```

Out[40]:

	Total_Pets_Neighbourhood	Inc_5_y (%)	Number_of_Vets	pets/vet
count	4.000000	4.000000	4.00	4.000000
mean	3786.250000	33.925000	0.75	3786.250000
std	626.205704	6.658516	0.50	626.205704
min	3074.000000	29.700000	0.00	3074.000000
25%	3434.750000	29.700000	0.75	3434.750000
50%	3760.500000	31.150000	1.00	3760.500000
75%	4112.000000	35.375000	1.00	4112.000000
max	4550.000000	43.700000	1.00	4550.000000

Cluster 1

Cluster 1

In [41]:
```
madrid_cluster_1=madrid_grouped_sorted.loc[madrid_grouped_sorted['Cluster Labels'] == 1, madrid_grouped_sorted.columns
[[1] + list(range(5, madrid_grouped_sorted.shape[1]))]]
madrid_cluster_1
```

Out[41]:

	Neighbourhood	Total_Pets_Neighbourhood	Inc_5_y (%)	Number_of_Vets	pets/vet
13	PACIFICO	3015.0	6.0	9.0	335.0
14	ADELFAS	1680.0	6.0	3.0	560.0
16	IBIZA	1947.0	6.0	3.0	649.0
17	LOS JERONIMOS	626.0	6.0	2.0	313.0
18	NIÑO JESUS	1380.0	6.0	2.0	690.0
19	RECOLETOS	1907.0	21.0	3.0	636.0
20	GOYA	3620.0	21.0	5.0	724.0
21	FUENTE DEL BERRO	2572.0	21.0	4.0	643.0
22	GUINDALERA	5100.0	21.0	13.0	392.0
23	LISTA	2553.0	21.0	4.0	638.0
25	EL VISO	1778.0	1.9	2.0	889.0
26	PROSPERIDAD	3755.0	1.9	4.0	939.0
27	CIUDAD JARDIN	1926.0	1.9	5.0	385.0
28	HISPANOAMERICA	3263.0	1.9	5.0	653.0
29	NUEVA ESPAÑA	2548.0	1.9	2.0	1274.0
30	CASTILLA	1747.0	1.9	3.0	582.0
31	BELLAS VISTAS	3190.0	19.1	3.0	1063.0
32	CUATRO CAMINOS	3761.0	19.1	6.0	627.0
33	CASTILLEJOS	2233.0	19.1	4.0	558.0
34	ALMENARA	2464.0	19.1	4.0	616.0
36	BERRUGUETE	2750.0	19.1	4.0	688.0
37	GAZTAMBIDE	2925.0	12.2	3.0	975.0
38	ARAPILES	3106.0	12.2	5.0	621.0
39	TRAFALGAR	3129.0	12.2	6.0	522.0
40	ALMAGRO	2501.0	12.2	3.0	834.0

In [42]:
```
madrid_cluster_1.describe()
```

Out[42]:

	Total_Pets_Neighbourhood	Inc_5_y (%)	Number_of_Vets	pets/vet
count	65.000000	65.000000	65.000000	65.000000
mean	2594.692308	13.269231	3.723077	693.138462
std	1521.523270	5.624026	2.534189	247.629188
min	176.000000	1.900000	0.000000	176.000000
25%	1733.000000	11.400000	2.000000	522.000000
50%	2548.000000	12.200000	4.000000	653.000000
75%	3263.000000	19.100000	5.000000	858.000000
max	7083.000000	21.200000	13.000000	1317.000000

Cluster 2

In [43]:
```
madrid_cluster_2=madrid_grouped_sorted.loc[madrid_grouped_sorted['Cluster Labels'] == 2, madrid_grouped_sorted.columns
[[1] + list(range(5, madrid_grouped_sorted.shape[1]))]]
madrid_cluster_2
```

Out[43]:

	Neighbourhood	Total_Pets_Neighbourhood	Inc_5_y (%)	Number_of_Vets	pets/vet
15	ESTRELLA	2081.0	6.0	1.0	2081.0
24	CASTELLANA	2080.0	21.0	1.0	2080.0
35	VALDEACEDERAS	2843.0	19.1	0.0	2843.0
53	CIUDAD UNIVERSITARIA	2134.0	14.1	1.0	2134.0
62	CAMPAMENTO	2232.0	32.1	1.0	2232.0
64	LAS AGUILAS	5887.0	32.1	3.0	1962.0
73	ORCASUR	1473.0	21.2	1.0	1473.0
74	SAN FERMIN	2514.0	21.2	1.0	2514.0
75	ALMENDRALES	2299.0	21.2	1.0	2299.0
76	MOSCARDO	2831.0	21.2	2.0	1416.0
77	ZOFIO	1501.0	21.2	1.0	1501.0
78	PRADOLONGO	1858.0	21.2	1.0	1858.0
80	SAN DIEGO	5180.0	18.8	3.0	1727.0
81	PALOMERAS BAJAS	4879.0	18.8	2.0	2440.0
82	PALOMERAS SURESTE	5164.0	18.8	3.0	1721.0
84	NUMANCIA	5765.0	18.8	2.0	2882.0
88	MEDIA LEGUA	1720.0	15.9	0.0	1720.0
89	FONTARRON	1665.0	15.9	1.0	1665.0
90	VINATEROS	1624.0	15.9	1.0	1624.0
96	SAN JUAN BAUTISTA	1419.0	19.2	0.0	1419.0
105	VALDEFUENTES	7490.0	11.9	3.0	2497.0
108	BUTARQUE	1923.0	32.6	1.0	1923.0
118	SIMANCAS	3306.0	25.7	2.0	1653.0
121	ARCOS	2858.0	25.7	2.0	1429.0
123	REJAS	1979.0	25.7	1.0	1979.0

In [44]:
```
madrid_cluster_2.describe()
```
Out[44]:

	Total_Pets_Neighbourhood	Inc_5_y (%)	Number_of_Vets	pets/vet
count	25.000000	25.000000	25.000000	25.000000
mean	2988.200000	20.612000	1.400000	1962.880000
std	1694.681927	6.135234	0.912871	432.878863
min	1419.000000	6.000000	0.000000	1416.000000
25%	1858.000000	18.800000	1.000000	1653.000000
50%	2232.000000	21.000000	1.000000	1923.000000
75%	3306.000000	21.200000	2.000000	2232.000000
max	7490.000000	32.600000	3.000000	2882.000000

Cluster 3

Cluster 3

In [45]:
```
madrid_cluster_3 = madrid_grouped_sorted.loc[madrid_grouped_sorted['Cluster Labels'] == 3, madrid_grouped_sorted.colum
ns[[1] + list(range(5, madrid_grouped_sorted.shape[1]))]]
madrid_cluster_3
```
Out[45]:

	Neighbourhood	Total_Pets_Neighbourhood	Inc_5_y (%)	Number_of_Vets	pets/vet
0	PALACIO	4055.0	43.7	6.0	676.0
1	EMBAJADORES	8070.0	43.7	5.0	1614.0
2	CORTES	1857.0	43.7	3.0	619.0
4	UNIVERSIDAD	5711.0	43.7	5.0	1142.0
5	SOL	1300.0	43.7	1.0	1300.0
6	IMPERIAL	2223.0	31.7	3.0	741.0
7	ACACIAS	3587.0	31.7	7.0	512.0
8	CHOPERA	1961.0	31.7	3.0	654.0
9	LEGAZPI	1920.0	31.7	2.0	960.0
10	DELICIAS	2733.0	31.7	2.0	1366.0
11	PALOS DE MOGUER	2540.0	31.7	5.0	508.0
12	ATOCHA	116.0	31.7	0.0	116.0
58	LOS CARMENES	1992.0	32.1	2.0	996.0
59	PUERTA DEL ANGEL	4765.0	32.1	5.0	953.0
60	LUCERO	4143.0	32.1	4.0	1036.0
61	ALUCHE	7590.0	32.1	9.0	843.0
63	CUATRO VIENTOS	664.0	32.1	1.0	664.0
65	COMILLAS	2278.0	29.7	3.0	759.0
66	OPAÑEL	3352.0	29.7	6.0	559.0
68	VISTA ALEGRE	4711.0	29.7	5.0	942.0
70	BUENAVISTA	4783.0	29.7	5.0	957.0

In [46]:
```
madrid_cluster_3.describe()
```
Out[46]:

	Total_Pets_Neighbourhood	Inc_5_y (%)	Number_of_Vets	pets/vet
count	37.000000	37.000000	37.000000	37.000000
mean	3013.972973	34.045946	3.189189	952.270270
std	1833.767683	7.651782	2.258783	396.125447
min	116.000000	25.700000	0.000000	116.000000
25%	1655.000000	29.700000	2.000000	676.000000
50%	2733.000000	31.700000	3.000000	953.000000
75%	4055.000000	32.600000	5.000000	1142.000000
max	8070.000000	53.200000	9.000000	1655.000000

5. Conclusions

According to the results for each cluster we have the following conclusions:

Cluster 0: The 4 neighbourhoods included in the cluster 0 are located in boroughs with a high increase in the number of pets in the last 5 years and with a high ratio pets/vets. The neighbourhoods included in this cluster are preferential to install a new veterinary clinic.

Cluster 1: The 65 neighbourhoods included in the cluster 1 are located in boroughs with a low increase in the number of pets in the last 5 years and with very low ratio pets/vets. The neighbourhoods included in this cluster are the worst to install a new veterinary clinic.

Cluster 2: The 25 neighbourhoods included in the cluster 2 are located in boroughs with a low increase in the number of pets in the last 5 years but with high ratio pets/vets. The neighbourhoods included in this cluster are not so good as the included in cluster 0 but are good location to install a new veterinary clinic yet.

Cluster 3: The 37 neighbourhoods included in cluster 3 are located in boroughs with a high increase in the number of pets in the last 5 years but with a low ratio pets/vets. The neighbourhoods included in this cluster are not good location to install a new veterinary clinic at the present but can be studied at the future to see how the increase evolves.

So, the decisions to consider are:

- **Best location** for the new Veterinary Clinic based in the analyzed criteria are the 4 neighbourhoods included in Cluster 0.
- The neighbourhoods included in the cluster 2 are not so good as the included in cluster 0 but are good location to install a new veterinary clinic yet.
- The neighbourhoods included in cluster 3 are not good location to install a new veterinary clinic at the present but can be studied at the future to see how the increase evolves.
- **Worst location** to install a new veterinary clinic are the neighbourhoods included in the cluster 1.