IBM Data Science Capstone Project : Find a suitable locations to open a luxury asian restaurant in Paris

Sanchez William

Introduction :

In order to complete this Capstone project, I am going to carry out a fictitious project where I will help a future restaurant owner to find out in which neighbourhood, borough of Paris, it would be interesting to set up his restaurant. Indeed, this entrepreneur, restaurateur has an ambitious project. He wants to introduce luxury asian food with new, delicious recipes and at an affordable price.

Indeed, Paris is a very touristy city and he is convinced that he will be able to spread the asian food culture throughout the world. Paris is a multicultural city recognised throughout the world.

After having carried out an Business Plan and in particular a market study, this entrepreneur wishes to establish himself in a neighbourhood where asian cuisine is popular. Thus, my study is a key step for this entrepreneur, because the success of his project depends on my estimation of the location of his future restaurant.

Business Problem :

The aim of this Capstone project is to help an entrepreneur find the best location in Paris, France, so that his restaurant can be best established and his project can be as viable and profitable as possible. On a larger scale, this project may help stakeholders wanting to create asian restaurants.

Data :

We are going to need geographic location data for the city of Paris, France. We will use a database including postcodes, borough in order to better use the Foursquare API and to determine the most popular venues categories.

In order to solve this problem we will need this data :

3.1 Paris Neighbourhood Data

JSON data : https://www.data.gouv.fr/fr/datasets/r/e88c6fda-1d09-42a0-a069-606d3259114e

The JSON file has data about all the neighbourhoods in France, we limit it to Paris.

postal_code : Postal codes for France

nom_comm : Name of Neighbourhoods in France

nom_dept : Name of the boroughs, equivalent to towns in France

geo_point_2d : Tuple containing the latitude and longitude of the Neighbourhoods.

3.2 Foursquare API Data

The data retrieved from Foursquare contained information of venues within a specified distance of the longitude and latitude of the postcodes. The information obtained per venue as follows:

Neighbourhood : Name of the Neighbourhood

Neighbourhood Latitude : Latitude of the Neighbourhood

Neighbourhood Longitude : Longitude of the Neighbourhood

Venue : Name of the Venue

Venue Latitude : Latitude of Venue

Venue Longitude : Longitude of Venue

Venue Category : Category of Venue

Methodology :

Firstly, To collect data for Paris, I download the JSON file containg all the postal codes of France from https://www.data.gouv.fr/fr/datasets/r/e88c6fda-1d09-42a0-a069-606d3259114e.

I am interested in the fields part of the France dataset, which contains all the data on the neighbourhoods, postcodes, latitude and longitude of Paris. Then I take a subset of the France dataset which contains the characters 'Paris'.

```
df_paris['geo_point_2d'][0]
```

```
']:  [48.87689616237872, 2.337460241388529]
```

However for the latitude and longitude data I had to separate the variable geo_point_2d as follows:

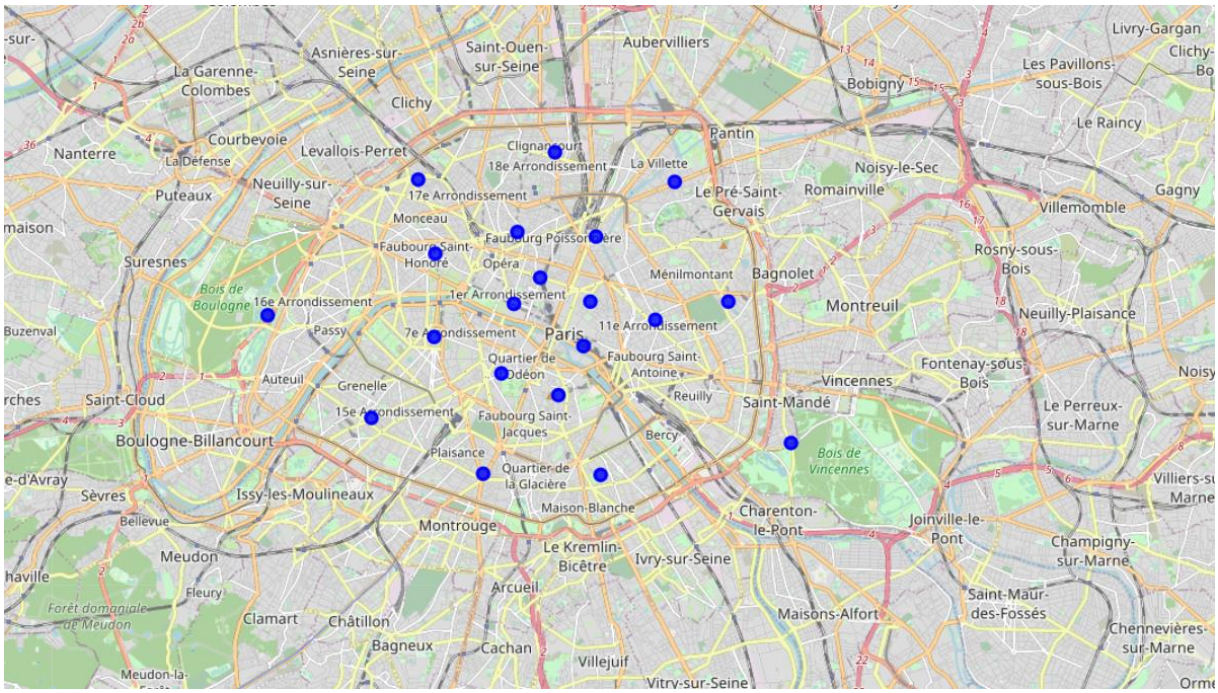
```
lat_lng = df_paris['geo_point_2d'].astype('str') # we have to be sure that

# Latitude
lat = lat_lng.apply(lambda x: x.split(',')[0])
lat = lat.apply(lambda x: x.lstrip('['))

# Longitude
lng = lat_lng.apply(lambda x: x.split(',')[1])
lng = lng.apply(lambda x: x.rstrip(']'))
```

Then I will create two pandas dataset, including latitutes and longitudes data and add it to the paris data frame, paris_f.

Afterwards, thanks to the Folium library, I will visualise the map of Paris with the different neighbourhoods :



Then we'll get the venues data using Foursquare. And we're going to use the Foursquare API to get the first 100 venues within a radius of 500. Then we're going to add them to a new dataset, paris_venues.

```
LIMIT = 100
def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            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 = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

Then we are going to analyse the districts of Paris. They will be grouped the venues dataset by the different districts of Paris.

We are going to check the number of unique venues but and if there are Asian restaurants.

Afterwards, we are going to do a one-hot encoding in order to encode our dataset of venues, to encode our categorical variables in order to facilitate the k-means algorithm. We will look at the average of the arrivals by neighbourhood.

```
onehot = pd.get_dummies(paris_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
onehot['Neighborhoods'] = paris_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [onehot.columns[-1]] + list(onehot.columns[:-1])
onehot = onehot[fixed_columns]

print(onehot.shape)
onehot.head()
```

```
paris_grouped = onehot.groupby(["Neighborhoods"]).mean().reset_index()

print(paris_grouped.shape)
paris_grouped
```

Before creating the different clusters, we will check the number of Asian restaurants in Paris. Indeed, my project was based on African restaurants, but there were not enough to use the k-means algorithm.

So we are going to check the districts with Asian restaurants.

Then, we will check the top 10 restaurants in Paris by neighbourhood.

```
num_top_venues = 10
indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhoods']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))
```

```
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhoods'] = paris_grouped['Neighborhoods']

for ind in np.arange(paris_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(paris_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.head(20)
```

Finally we're going to create our 3 clusters and do the k-means clustering and look at the k-means_labels

```
# set number of clusters
kclusters = 3

to_clustering = asian.drop(["Neighborhoods"], 1)

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

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

Then we will visualize the 3 clusters :

```
# 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(asian_merged['Neighborhood Latitude'], asian_merged['Neighborhood Longitude'], asian_merged['Neighborhood'], asian_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' - Cluster ' + str(cluster))
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

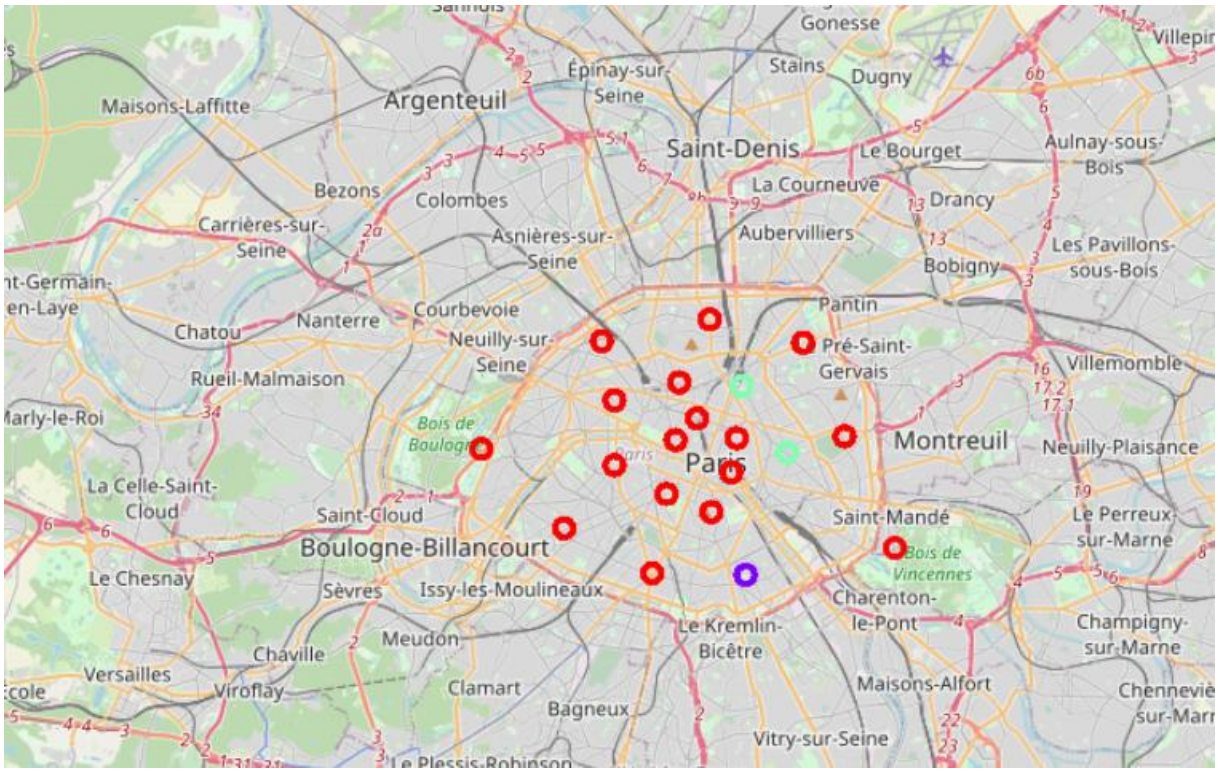
And finally examin the 3 clusters :

```
#Cluster 0
asian_merged.loc[asian_merged['Cluster Labels'] == 0]
```

	Neighborhood	Asian Restaurant	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
12	PARIS-2E-ARRONDISSEMENT	0.01	0	48.867903	2.344107	Cloud Cakes	48.865641	2.346302	Bakery
14	PARIS-4E-ARRONDISSEMENT	0.00	0	48.854228	2.357362	Quai d'Orléans	48.851596	2.354282	Trail
14	PARIS-4E-ARRONDISSEMENT	0.00	0	48.854228	2.357362	Galerie Azzedine Alaïa	48.857545	2.355217	Art Gallery
14	PARIS-4E-ARRONDISSEMENT	0.00	0	48.854228	2.357362	Les Nautes	48.852231	2.360306	Gastropub
14	PARIS-4E-ARRONDISSEMENT	0.00	0	48.854228	2.357362	Pitzman	48.855853	2.360295	Falafel Restaurant
...	...	...	...	...	...	...	...	...	...
10	PARIS-1ER-ARRONDISSEMENT	0.00	0	48.862630	2.336293	Pavillon des Sessions – Arts d'Afrique, d'Asie...	48.860724	2.332121	Art Museum
10	PARIS-1ER-ARRONDISSEMENT	0.00	0	48.862630	2.336293	Le Café Blanc	48.862719	2.339578	Bar
10	PARIS-1ER-ARRONDISSEMENT	0.00	0	48.862630	2.336293	Jardins du Carrousel (Jardin du Carrousel)	48.862056	2.332445	Garden
10	PARIS-1ER-ARRONDISSEMENT	0.00	0	48.862630	2.336293	Christian Louboutin	48.862697	2.340757	Shoe Store
10	PARIS-1ER-ARRONDISSEMENT	0.00	0	48.862630	2.336293	Colonnes de Buren	48.863618	2.336917	Sculpture Garden

This will allow me to determine the best location to set up the luxury Asian restaurant.

Results :



According to our k-means algorithm we defined 3 clusters for the city of Paris.

- Cluster 0 : Neighbourhoods of Paris with less asian restaurants, represented in red
- Cluster 1 : Neighbourhoods of Paris with most of asian restaurants, represented in purple
- Cluster 2 : Neighbourhoods of Paris with a reasonable number of Asian restaurants, represented in green

Discussions :

Explanation of which cluters you choose and therefore in which district you choose to locate the restaurant.

According to our results, the 13th arrondissement of Paris is the one with the most Asian restaurants. So you have to imagine that there will be competition if you want to set up an Asian restaurant in this region.

So, it seems more reasonable to set up your restaurant in the 10th or 11th arrondissement of Paris. Because the number of Asian restaurants is reasonable and will represent less competition.

Nevertheless, the analysis could be more effective. Indeed, as the entrepreneur wants to create a luxury Asian restaurant, other parameters would have to be taken into account in

the clustering, such as the average salary of the residents in this district, but also the rent in these districts. Indeed, the 10th and 11th arrondissements remain fairly poor neighbourhoods of Paris, compared to Paris 16th. Moreover, our clusters could be more precise with more visits received by the Foursquare API, as our dataset can still be provided.

Conclusion :

At the end of this project, we were able to identify the best neighbourhood in Paris to set up a luxury Asian restaurant. This, thanks to the identification of the problem and the processes of data extraction, data cleaning and because we had to prepare the data in order to make the best use of the k-means algorithm. The data visualization was used to give a recommendation to the actors of this project.