IBM Data Science Capstone Project
The Battle of Neighbourhoods
Final Report

Introduction/Business Problem

Paris, capital of France, is one of the most important and influential cities in the world. In terms of tourism, Paris is the second most visited city in Europe after London. The capital of France seems to have been designed specifically for the enjoyment of its visitors. Its streets, squares, buildings, gardens, and monuments beckon tourists to return, and indeed, many do.

Tourists from everywhere of the world visit Paris every day in two ways basically: with travel agencies providing standard "must see" guided tours or plan themselves with the help of websites or travel app full of impersonalised recommendation. Time is short and time is money. How to make the trip as personalised as possible so make the days in Paris as profitable as possible? Customised travel services are usually very expensive and not accessible to everyone then they always rely on experimented human guide expertise which can be asynchronized in terms of POI (Points of Interests) data.

This project will try to give access to everyone the possibility to customise his trip in Paris based on his personal interests. With the help of data visualisation, he can easily make his ideas on which Parisian arrondissement to be visited on priority with a deeper understanding on points of interests in different categories of arrondissement.

Data section

To accomplish this project two important data sets are necessary.

- Geo-Coordinate Data:

For this project we will use dataset from opendata.paris.fr for the arrondissements of Paris.

- Points of Interests Data:

We will need data about different venues across all of Paris and connect each venue to its respective arrondissement. To gain this information, we will use Foursquare API.

Methodology

1. Import necessary libraries to Jupiter Notebook

```python
# Import libraries
import numpy as np
import json
import pandas as pd
import requests
!conda install -c conda-forge geopy --yes
from geopy.geocoders import Nominatim
import requests
from pandas.io.json import json_normalize
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors
from bs4 import BeautifulSoup
from sklearn.cluster import KMeans
import folium
print('Libraries imported.')
```

2. Data Collection - Download and load geo-coordinate data to Pandas Dataframe format.

```python
paris = pd.read_csv('https://raw.githubusercontent.com/BolinF77/IBM_DS_Capstone/main/paris_arrondissements.csv')
paris
```

	CAR	NAME	NSQAR	CAR.1	CARINSEE	LAR	NSQCO	SURFACE	PERIMETRE	Geometry_X	Geometry_Y
0	3	Temple	750000003	3	3	3eme Ardt	750001537	1170882828	4519264	48.862872	2.360001
1	19	Buttes-Chaumont	750000019	19	19	19eme Ardt	750001537	6792651129	11253182	48.887076	2.384821
2	14	Observatoire	750000014	14	14	14eme Ardt	750001537	5614877309	10317483	48.829245	2.326542
3	10	Entrepot	750000010	10	10	10eme Ardt	750001537	2891739442	6739375	48.876130	2.360728
4	12	Reuilly	750000012	12	12	12eme Ardt	750001537	16314782637	24089666	48.834974	2.421325
5	16	Passy	750000016	16	16	16eme Ardt	750001537	16372542129	17416110	48.860392	2.261971
6	11	Popincourt	750000011	11	11	11eme Ardt	750001537	3665441552	8282012	48.859059	2.380058
7	2	Bourse	750000002	2	2	2eme Ardt	750001537	991153745	4554104	48.868279	2.342803
8	4	Hotel-de-Ville	750000004	4	4	4eme Ardt	750001537	1600585632	5420908	48.854341	2.357630
9	17	Batignolles-Monceau	750000017	17	17	17eme Ardt	750001537	5668834504	10775580	48.887327	2.306777
10	18	Buttes-Montmartre	750000018	18	18	18eme Ardt	750001537	5996051308	9916464	48.892569	2.348161
11	1	Louvre	750000001	1	1	1er Ardt	750001537	1824612860	6054937	48.862563	2.336443
12	5	Pantheon	750000005	5	5	5eme Ardt	750001537	2539374623	6239195	48.844443	2.350715
13	7	Palais-Bourbon	750000007	7	7	7eme Ardt	750001537	4090057185	8099425	48.856174	2.312188
14	20	Menilmontant	750000020	20	20	20eme Ardt	750001537	5983446037	10704940	48.863461	2.401188
15	8	elysee	750000008	8	8	8eme Ardt	750001537	3880036397	7880533	48.872721	2.312554
16	9	Opera	750000009	9	9	9eme Ardt	750001537	2178303275	6471588	48.877164	2.337458
17	13	Gobelins	750000013	13	13	13eme Ardt	750001537	7149311091	11546547	48.828388	2.362272
18	15	Vaugirard	750000015	15	15	15eme Ardt	750001537	8494994081	13678798	48.840085	2.292826
19	6	Luxembourg	750000006	6	6	6eme Ardt	750001537	2153095586	6483687	48.849130	2.332898

3. Data wrangling – Rename the necessary columns and remove unnecessary columns

```
paris.rename(columns={'NAME': 'Neighborhood ', 'CAR': 'Arrondissement', 'Geometry_X': 'Latitude', 'Geometry_Y': 'Longitude',  'LAR': 'French_Name'}, inplace=True
paris.drop(['NSQAR','CAR.1','CARINSEE','NSQCO','SURFACE', 'PERIMETRE' ], axis=1, inplace=True)
paris
```

	Arrondissement	Neighborhood	French_Name	Latitude	Longitude
0	3	Temple	3eme Ardt	48.862872	2.360001
1	19	Buttes-Chaumont	19eme Ardt	48.887076	2.384821
2	14	Observatoire	14eme Ardt	48.829245	2.326542
3	10	Entrepot	10eme Ardt	48.876130	2.360728
4	12	Reuilly	12eme Ardt	48.834974	2.421325
5	16	Passy	16eme Ardt	48.860392	2.261971
6	11	Popincourt	11eme Ardt	48.859059	2.380058
7	2	Bourse	2eme Ardt	48.868279	2.342803
8	4	Hotel-de-Ville	4eme Ardt	48.854341	2.357630
9	17	Batignolles-Monceau	17eme Ardt	48.887327	2.306777
10	18	Buttes-Montmartre	18eme Ardt	48.892569	2.348161
11	1	Louvre	1er Ardt	48.862563	2.336443
12	5	Pantheon	5eme Ardt	48.844443	2.350715
13	7	Palais-Bourbon	7eme Ardt	48.856174	2.312188
14	20	Menilmontant	20eme Ardt	48.863461	2.401188
15	8	elysee	8eme Ardt	48.872721	2.312554
16	9	Opera	9eme Ardt	48.877164	2.337458
17	13	Gobelins	13eme Ardt	48.828388	2.362272
18	15	Vaugirard	15eme Ardt	48.840085	2.292826
19	6	Luxembourg	6eme Ardt	48.849130	2.332898

4. Get the latitude and longitude values of Paris with the help of geopy library

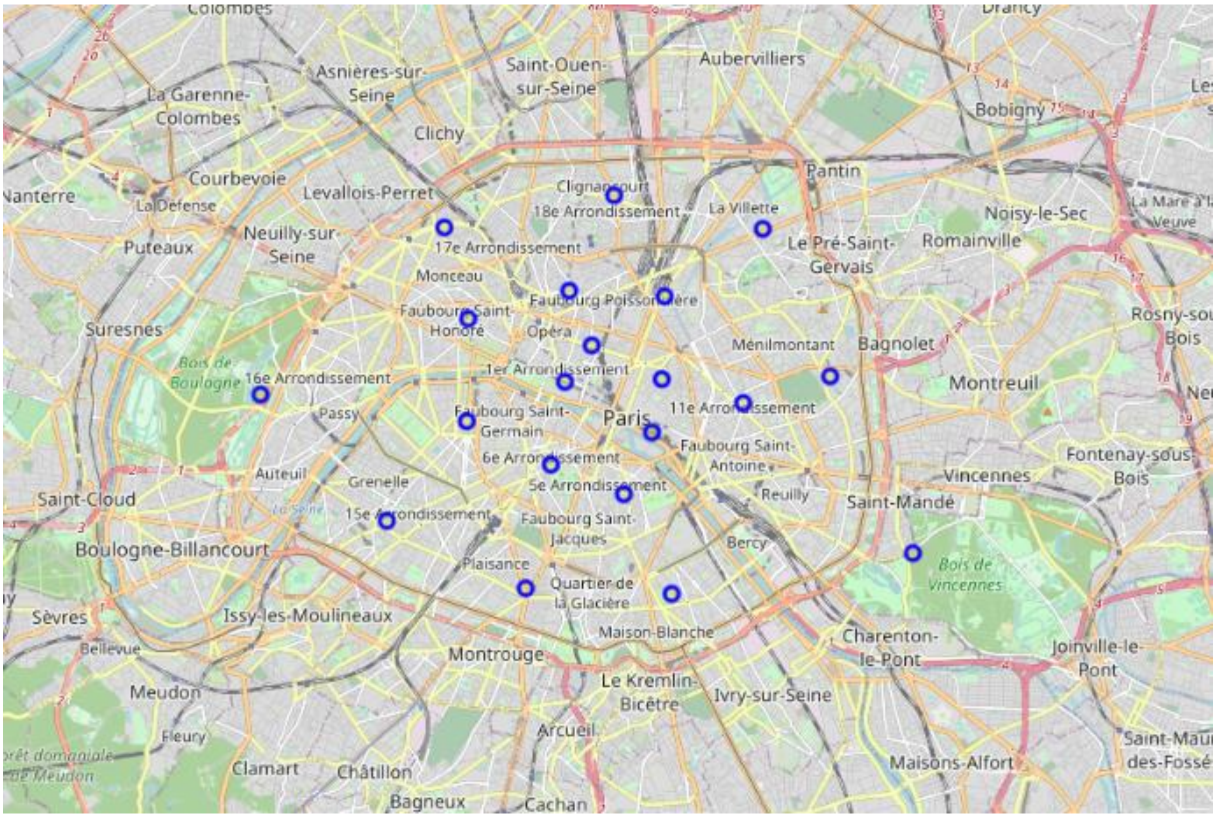
```
# Retrieve the Latitude and Longitude for Paris
from geopy.geocoders import Nominatim
address = 'Paris'
# Define the user_agent as Paris_explorer
geolocator = Nominatim(user_agent="Paris_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geographical coordinates of Paris France are {}, {}.'.format(latitude, longitude))
```

```
The geographical coordinates of Paris France are 48.8566969, 2.3514616.
```

5. Create map of Paris using the arrondissements latitude and longitude values

```
# create map of Paris using the above latitude and longitude values
map_paris = folium.Map(location=[latitude, longitude], zoom_start=12)
# add markers to map
for lat, lng, label in zip(paris['Latitude'], paris['Longitude'], paris['French_Name']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#e8dc54',
        fill_opacity=0.5,
        parse_html=False).add_to(map_paris)

map_paris
```



Until now, the preparation of the geo-coordinate data is finished. This first step is important as a foundation for further exploratory data analysis. For this project the ETL (Extract Transform Load) process is quite simple, and light weighted so no SQL commands are needed. For projects with voluminous data, we might have to request databases with SQL.

6. This step starts the query to Points of Interests (POI) data by using Foursquare API, after API authentication and request engineering cf. Jupiter Notebook for details, we explored all points of interests in all arrondissements in Paris

51	3eme Ardt	48.862872	2.360001	Archives Nationales	48.859975	2.357139	Historic Site
52	3eme Ardt	48.862872	2.360001	Happy Nouilles	48.864444	2.355294	Noodle House
53	3eme Ardt	48.862872	2.360001	Aēsop	48.858998	2.359470	Cosmetics Shop
54	3eme Ardt	48.862872	2.360001	La Perle	48.859985	2.360735	Bistro
55	3eme Ardt	48.862872	2.360001	Terres de Café	48.860580	2.355482	Coffee Shop
56	3eme Ardt	48.862872	2.360001	Loustic	48.863793	2.354707	Coffee Shop
57	3eme Ardt	48.862872	2.360001	La Chambre aux Confitures	48.858979	2.359230	Gourmet Shop
58	3eme Ardt	48.862872	2.360001	Galerie Perrotin	48.860726	2.365168	Art Gallery
59	3eme Ardt	48.862872	2.360001	Bien l'Épicerie	48.860179	2.360499	Health Food Store

35]:
```
paris_venues.shape
```

35]: (1300, 7)

Check how many venues were returned for each neighborhood

36]:
```
paris_venues.groupby('French_Name').count()
```

36]:

French_Name	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
10eme Ardt	100	100	100	100	100	100
11eme Ardt	69	69	69	69	69	69
12eme Ardt	4	4	4	4	4	4
13eme Ardt	60	60	60	60	60	60
14eme Ardt	24	24	24	24	24	24
15eme Ardt	60	60	60	60	60	60
16eme Ardt	10	10	10	10	10	10
17eme Ardt	57	57	57	57	57	57
18eme Ardt	46	46	46	46	46	46
19eme Ardt	44	44	44	44	44	44
1er Ardt	100	100	100	100	100	100
20eme Ardt	45	45	45	45	45	45
2eme Ardt	100	100	100	100	100	100
3eme Ardt	100	100	100	100	100	100
4eme Ardt	100	100	100	100	100	100
5eme Ardt	86	86	86	86	86	86
6eme Ardt	53	53	53	53	53	53
7eme Ardt	100	100	100	100	100	100
8eme Ardt	42	42	42	42	42	42
9eme Ardt	100	100	100	100	100	100

According to the dataframe shape, we count 1300 venues POI in Paris and we list the number of venues by arrondissement.

7. Calculate the number of unique categories and encode venues in arrondissement by using one hot encoding method

```python
# Calculate how many unique categories there are.
print('There are {} unique venue categories.'.format(len(paris_venues['Venue Category'].unique())))
```

```
There are 199 unique venue categories.
```

```python
# Analyze each of the Neighborhoods from the results
# one hot encoding
paris_onehot = pd.get_dummies(paris_venues[['Venue Category']], prefix="", prefix_sep="")
# add neighborhood column back to dataframe
paris_onehot['Neighborhood'] = paris_venues['French_Name']
# move neighborhood column to the first column
fixed_columns = [paris_onehot.columns[-1]] + list(paris_onehot.columns[:-1])
paris_onehot = paris_onehot[fixed_columns]
paris_onehot
```

	Neighborhood	Afghan Restaurant	African Restaurant	American Restaurant	Antique Shop	Argentinian Restaurant	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	...	Udon Restaurant	University	Vegetarian / Vegan Restaurant	Venezuelan Restaurant	Vietnamese Restaurant	Wine Bar	Wine Shop	Women's Store	Zoo	Zoo Exhibit
0	3eme Ardt	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	3eme Ardt	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	3eme Ardt	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	3eme Ardt	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	3eme Ardt	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1295	6eme Ardt	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1296	6eme Ardt	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1297	6eme Ardt	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1298	6eme Ardt	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1299	6eme Ardt	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

1300 rows × 200 columns

```python
paris_grouped = paris_onehot.groupby('Neighborhood').mean().reset_index()
paris_grouped
```

	Neighborhood	Afghan Restaurant	African Restaurant	American Restaurant	Antique Shop	Argentinian Restaurant	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	...	Udon Restaurant	University	Vegetarian / Vegan Restaurant	Venezuelan Restaurant	Vietnamese Restaurant	Wine Bar	Wine Shop	Women's Store	Zoo	Zoo Exhibit
0	10eme Ardt	0.000000	0.02	0.000000	0.00	0.00	0.000000	0.000000	0.000000	0.020000	...	0.00	0.000000	0.010000	0.00	0.000000	0.020000	0.02	0.000000	0.00	0.00
1	11eme Ardt	0.014493	0.00	0.000000	0.00	0.00	0.000000	0.014493	0.000000	0.028986	...	0.00	0.000000	0.014493	0.00	0.028986	0.028986	0.00	0.014493	0.00	0.00
2	12eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.000000	0.000000	0.000000	0.000000	...	0.00	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.25	0.25
3	13eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.000000	0.000000	0.000000	0.183333	...	0.00	0.000000	0.000000	0.00	0.233333	0.000000	0.00	0.000000	0.00	0.00
4	14eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.000000	0.000000	0.000000	0.000000	...	0.00	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.00	0.00
5	15eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.000000	0.000000	0.016667	0.000000	...	0.00	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.00	0.00
6	16eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.000000	0.100000	0.000000	0.000000	...	0.00	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.00	0.00
7	17eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.000000	0.017544	0.000000	0.017544	...	0.00	0.017544	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.00	0.00
8	18eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.000000	0.000000	0.000000	0.000000	...	0.00	0.000000	0.000000	0.00	0.043478	0.021739	0.00	0.000000	0.00	0.00
9	19eme Ardt	0.000000	0.00	0.022727	0.00	0.00	0.000000	0.000000	0.000000	0.000000	...	0.00	0.000000	0.000000	0.00	0.022727	0.000000	0.00	0.000000	0.00	0.00
10	1er Ardt	0.000000	0.00	0.000000	0.00	0.00	0.000000	0.030000	0.000000	0.000000	...	0.02	0.000000	0.000000	0.00	0.010000	0.010000	0.01	0.000000	0.00	0.00
11	20eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.000000	0.000000	0.000000	0.000000	...	0.00	0.000000	0.000000	0.00	0.000000	0.022222	0.00	0.000000	0.00	0.00
12	2eme Ardt	0.000000	0.00	0.010000	0.00	0.00	0.000000	0.000000	0.000000	0.010000	...	0.00	0.000000	0.000000	0.00	0.000000	0.040000	0.01	0.020000	0.00	0.00
13	3eme Ardt	0.000000	0.00	0.000000	0.00	0.01	0.050000	0.010000	0.000000	0.010000	...	0.00	0.000000	0.010000	0.00	0.020000	0.040000	0.02	0.000000	0.00	0.00
14	4eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.020000	0.010000	0.010000	0.000000	...	0.00	0.000000	0.000000	0.00	0.000000	0.020000	0.00	0.000000	0.00	0.00
15	5eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.000000	0.000000	0.000000	0.000000	...	0.00	0.000000	0.000000	0.00	0.023256	0.023256	0.00	0.000000	0.00	0.00
16	6eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.000000	0.018868	0.000000	0.000000	...	0.00	0.000000	0.000000	0.00	0.000000	0.018868	0.00	0.000000	0.00	0.00
17	7eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.000000	0.020000	0.000000	0.010000	...	0.00	0.000000	0.010000	0.00	0.000000	0.000000	0.00	0.000000	0.00	0.00
18	8eme Ardt	0.000000	0.00	0.000000	0.00	0.00	0.047619	0.023810	0.000000	0.000000	...	0.00	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.00	0.00
19	9eme Ardt	0.000000	0.01	0.000000	0.01	0.00	0.000000	0.000000	0.000000	0.000000	...	0.00	0.000000	0.020000	0.01	0.010000	0.030000	0.00	0.000000	0.00	0.00

20 rows × 200 columns

8. Find the top 10 venues' categories by arrondissement

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	10eme Ardt	French Restaurant	Coffee Shop	Bistro	Indian Restaurant	Café	Hotel	Pizza Place	Italian Restaurant	Asian Restaurant	Seafood Restaurant
1	11eme Ardt	French Restaurant	Supermarket	Restaurant	Café	Bakery	Pastry Shop	Italian Restaurant	Pizza Place	Wine Bar	Vietnamese Restaurant
2	12eme Ardt	Zoo Exhibit	Zoo	Monument / Landmark	Supermarket	Performing Arts Venue	Nightclub	Noodle House	Okonomiyaki Restaurant	Optical Shop	Outdoor Sculpture
3	13eme Ardt	Vietnamese Restaurant	Asian Restaurant	Chinese Restaurant	Thai Restaurant	French Restaurant	Juice Bar	Gourmet Shop	Creperie	Butcher	Bus Stop
4	14eme Ardt	French Restaurant	Food & Drink Shop	Hotel	Supermarket	Pizza Place	Bistro	Tea Room	Bakery	Brasserie	Fast Food Restaurant
5	15eme Ardt	Hotel	Italian Restaurant	French Restaurant	Coffee Shop	Bistro	Thai Restaurant	Brasserie	Supermarket	Indian Restaurant	Bakery
6	16eme Ardt	Lake	Plaza	Bus Station	Bus Stop	Art Museum	French Restaurant	Park	Boat or Ferry	Afghan Restaurant	Perfume Shop
7	17eme Ardt	French Restaurant	Hotel	Italian Restaurant	Bistro	Bakery	Japanese Restaurant	Café	Plaza	Restaurant	Asian Restaurant
8	18eme Ardt	French Restaurant	Bar	Café	Supermarket	Convenience Store	Coffee Shop	Restaurant	Vietnamese Restaurant	Sandwich Place	Beer Store
9	19eme Ardt	French Restaurant	Bar	Café	Seafood Restaurant	Beer Bar	Hotel	Supermarket	Bistro	Brewery	Creperie
10	1er Ardt	French Restaurant	Japanese Restaurant	Plaza	Hotel	Coffee Shop	Café	Italian Restaurant	Art Museum	Ramen Restaurant	Bistro
11	20eme Ardt	Bakery	Japanese Restaurant	French Restaurant	Bar	Plaza	Italian Restaurant	Café	Park	Bistro	Pizza Place
12	2eme Ardt	French Restaurant	Cocktail Bar	Wine Bar	Bakery	Plaza	Hotel	Japanese Restaurant	Pedestrian Plaza	Burger Joint	Salad Place
13	3eme Ardt	French Restaurant	Art Gallery	Cocktail Bar	Coffee Shop	Burger Joint	Wine Bar	Bakery	Bistro	Italian Restaurant	Sandwich Place
14	4eme Ardt	French Restaurant	Ice Cream Shop	Clothing Store	Pastry Shop	Hotel	Italian Restaurant	Plaza	Pedestrian Plaza	Cocktail Bar	Thai Restaurant
15	5eme Ardt	French Restaurant	Italian Restaurant	Science Museum	Hotel	Plaza	Bakery	Café	Greek Restaurant	Coffee Shop	Bar
16	6eme Ardt	French Restaurant	Bakery	Italian Restaurant	Cocktail Bar	Plaza	Ice Cream Shop	Bookstore	Seafood Restaurant	Fountain	Tailor Shop
17	7eme Ardt	Hotel	French Restaurant	Italian Restaurant	Plaza	Café	Cocktail Bar	History Museum	Coffee Shop	Gourmet Shop	Bistro
18	8eme Ardt	French Restaurant	Hotel	Spa	Art Gallery	Cocktail Bar	Bakery	Bar	Hotel Bar	Furniture / Home Store	Park
19	9eme Ardt	French Restaurant	Hotel	Bistro	Cocktail Bar	Bakery	Wine Bar	Restaurant	Lounge	Japanese Restaurant	Gym / Fitness Center

This top 10 venues' categories list is already very useful to provide deeper insights to tourists so they can already more structured ideas on where they may want to visit relating to their interests. To make easier and more visual, we want to use the K-means cluster method to let the data show us if there are clusters so tourists can combine different arrondissement when they make their visit path or when they want to choose and ideal hotel to stay in a more flexible and efficient way.

9. K-means clustering (unsupervised machine learning, K = 9)

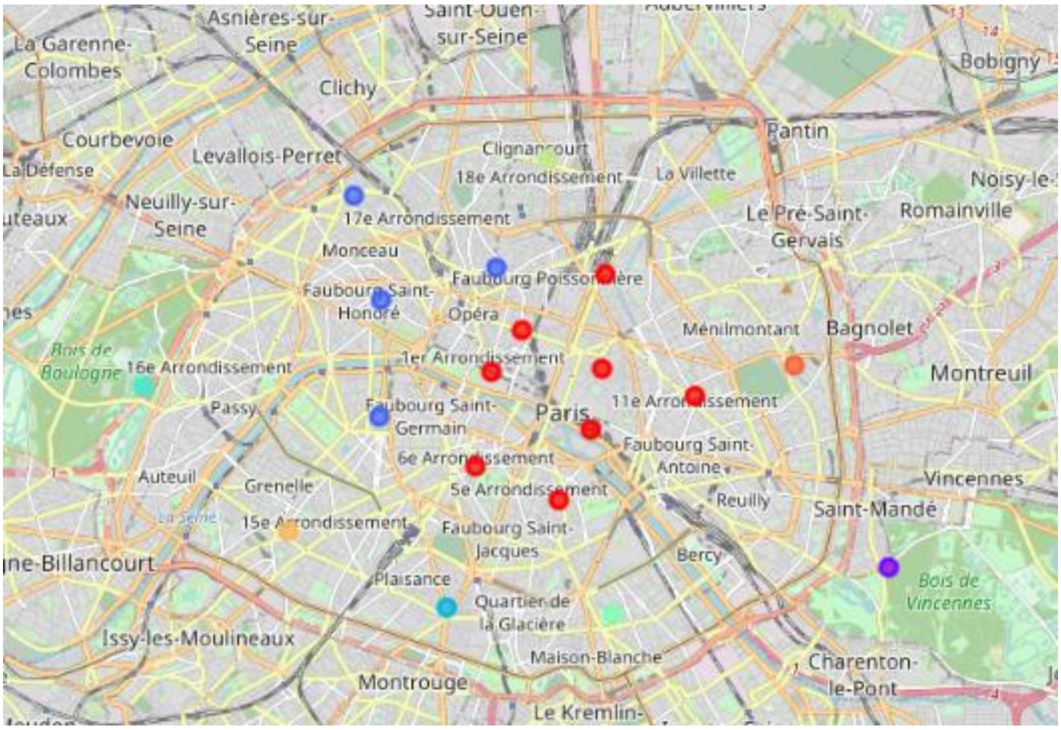
```
# set number of clusters
kclusters = 9
clustering_grouped_paris = paris_grouped.drop('Neighborhood', 1)
# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(clustering_grouped_paris)
# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:20]
```

```
array([0, 0, 1, 5, 3, 7, 4, 2, 6, 6, 0, 8, 0, 0, 0, 0, 0, 2, 2, 2])
```

```
# add clustering labels
paris_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
```

```
#new dataframe
paris_merged = paris_df
paris_merged = paris_merged.join(paris_venues_sorted.set_index("Neighborhood"), on = "French_Name")
paris_merged.head()
```

	Arrondissement	Neighborhood	French_Name	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	3	Temple	3eme Ardt	48.862872	2.360001	0	French Restaurant	Art Gallery	Cocktail Bar	Coffee Shop	Burger Joint	Wine Bar	Bakery	Bistro	Italian Restaurant	Sandwich Place
1	19	Buttes-Chaumont	19eme Ardt	48.887076	2.384821	6	French Restaurant	Bar	Café	Seafood Restaurant	Beer Bar	Hotel	Supermarket	Bistro	Brewery	Creperie
2	14	Observatoire	14eme Ardt	48.829245	2.326542	3	French Restaurant	Food & Drink Shop	Hotel	Supermarket	Pizza Place	Bistro	Tea Room	Bakery	Brasserie	Fast Food Restaurant
3	10	Entrepot	10eme Ardt	48.876130	2.360728	0	French Restaurant	Coffee Shop	Bistro	Indian Restaurant	Café	Hotel	Pizza Place	Italian Restaurant	Asian Restaurant	Seafood Restaurant
4	12	Reuilly	12eme Ardt	48.834974	2.421325	1	Zoo Exhibit	Zoo	Monument / Landmark	Supermarket	Performing Arts Venue	Nightclub	Noodle House	Okonomiyaki Restaurant	Optical Shop	Outdoor Sculpture



10. Cluster’s analysis

- Cluster 0 – Restaurant, Bar, Coffee and Ice-cream

```
paris_merged.loc[paris_merged["Cluster Labels"] == 0,
                 paris_merged.columns[[0] + [1] + list(range(5, paris_merged.shape[1]))]]
```

	Arrondissement	Neighborhood	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	3	Temple	0	French Restaurant	Art Gallery	Cocktail Bar	Coffee Shop	Burger Joint	Wine Bar	Bakery	Bistro	Italian Restaurant	Sandwich Place
3	10	Entrepot	0	French Restaurant	Coffee Shop	Bistro	Indian Restaurant	Café	Hotel	Pizza Place	Italian Restaurant	Asian Restaurant	Seafood Restaurant
6	11	Popincourt	0	French Restaurant	Supermarket	Restaurant	Café	Bakery	Pastry Shop	Italian Restaurant	Pizza Place	Wine Bar	Vietnamese Restaurant
7	2	Bourse	0	French Restaurant	Cocktail Bar	Wine Bar	Bakery	Plaza	Hotel	Japanese Restaurant	Pedestrian Plaza	Burger Joint	Salad Place
8	4	Hotel-de-Ville	0	French Restaurant	Ice Cream Shop	Clothing Store	Pastry Shop	Hotel	Italian Restaurant	Plaza	Pedestrian Plaza	Cocktail Bar	Thai Restaurant
11	1	Louvre	0	French Restaurant	Japanese Restaurant	Plaza	Hotel	Coffee Shop	Café	Italian Restaurant	Art Museum	Ramen Restaurant	Bistro
12	5	Pantheon	0	French Restaurant	Italian Restaurant	Science Museum	Hotel	Plaza	Bakery	Café	Greek Restaurant	Coffee Shop	Bar
19	6	Luxembourg	0	French Restaurant	Bakery	Italian Restaurant	Cocktail Bar	Plaza	Ice Cream Shop	Bookstore	Seafood Restaurant	Fountain	Tailor Shop

- Cluster 2 – Hotel, Restaurant, Bar

```
paris_merged.loc[paris_merged["Cluster Labels"] == 2,
                 paris_merged.columns[[0] + [1] + list(range(5, paris_merged.shape[1]))]]
```

	Arrondissement	Neighborhood	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
9	17	Batignolles-Monceau	2	French Restaurant	Hotel	Italian Restaurant	Bistro	Bakery	Japanese Restaurant	Café	Plaza	Restaurant	Asian Restaurant
13	7	Palais-Bourbon	2	Hotel	French Restaurant	Italian Restaurant	Plaza	Café	Cocktail Bar	History Museum	Coffee Shop	Gourmet Shop	Bistro
15	8	elysee	2	French Restaurant	Hotel	Spa	Art Gallery	Cocktail Bar	Bakery	Bar	Hotel Bar	Furniture / Home Store	Park
16	9	Opera	2	French Restaurant	Hotel	Bistro	Cocktail Bar	Bakery	Wine Bar	Restaurant	Lounge	Japanese Restaurant	Gym / Fitness Center

Discussion

According to the results of K-means clustering, we can see two obvious clusters which share the same categories of POI: Cluster 0 is a zone where tourists will have no difficulties to find restaurants especially good French restaurants as well as bars and ice-cream shops. Cluster 2 is an area where tourists will easily find hotels and European restaurants. As we setup the K=9 for 20 arrondissement, we can a fine granularity for clustering so besides there two obvious clusters each containing more than 3 arrondissements, we have also other “stand-alone” arrondissements with their unique top 10 POI categories. These “stand-alone” cluster can be also useful as a summary of the specialities and can help tourists to combine different clusters to plan their own customised trip. Tourists may also use the map of clusters to target a area to stay in the centre of many clusters they are interested in to facilitate their transport.

Conclusion

This project englobes and demonstrates basic data scientists’ skills in terms of data collection, data wrangling, data transformation (ETL) and Exploratory Data Analysis with unsupervised Machine Learning method: K-means. In addition, the data visualisation helps the illustration and explanation of findings. Tourists can use these data visualisations to better plan their trips with deeper understanding of points of interests in different clusters without the help of traditional travel agencies.