IBM Data Science Capstone Project

BATTLE OF NEIGHBORHOOD RESTAURANTS IN PARIS

Philippe BOTTIER | Applied Data Science Capstone by IBM |

Introduction

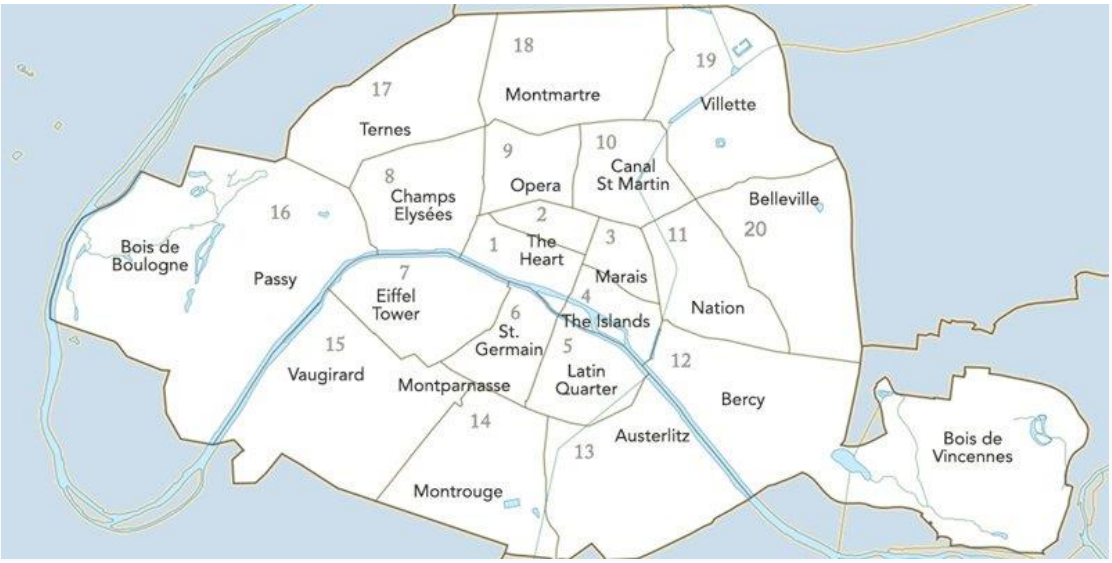
DESCRIPTION AND DISCUSSION OF THE CONTEXT

Paris is the capital and most populous city of France, with an estimated population of 2,175,601 residents as of 2018, in an area of more than 105 square kilometres (41 square miles). Since the 17th century, Paris has been one of Europe's major centres of finance, diplomacy, commerce, fashion, science, arts and gastronomy

The city of Paris is divided into twenty administrative districts called arrondissements. The twenty arrondissements are arranged in the form of a clockwise spiral (often likened to a snail shell), starting from the middle of the city, with the first on the Right Bank (north bank) of the Seine.

In French, notably on street signs, the number is often given in Roman numerals. For example, the Eiffel Tower belongs to the VIIe arrondissement while Gare de l'Est is in the Xe arrondissement.

Each Parisian arrondissement has four neighborhoods, so a total of 80, which constitute the highest level of public administration in Paris.



Paris received 38 million visitors in 2019, measured by hotel stays, with the largest numbers of foreign visitors coming from the United States, the United Kingdom, Germany, and China. It was ranked as the second most visited travel destination in the world in 2019, after Bangkok and just ahead of London.

As a resident in a suburb near the capital, I chose the city of Paris to lead my project.

BUSINESS PROBLEM

The problem to be solved is totally imaginary.

My company, specialized in data analysis, was contacted by the Tourist Office of the city of Paris to provide them with an analysis that would allow them to advise tourists on the types of restaurants they could find during their visits to the Paris neighborhoods.

This analysis could be carried out using an unsupervised learning model that would reproduce on the map of Paris the groupings of neighborhoods according to the types of restaurants that are most represented there.

Data

To carry out this project, I needed the following data :

- List of Parisian districts (arrondissements in French) as well as the list of Parisian neighborhoods
- Geo-coordinates of the districts in Paris
- Top venues of districts

1. Data, in GeoJSON format, are available on the web site : https://opendata.paris.fr/explore/dataset/arrondissements/export/?location=13,48.85156,2.32327

I used the jason library to import the files

```
import json
geo_borough = json.load(open("arrondissements.geojson")) # Paris arrondissements
geo_neighbourhood = json.load(open("quartier_paris.geojson")) # Paris neighborhoods
```

The GeoJSON format is used to represent data of a geographic type. In this object, transformed into a dictionary under python, there are two elements: the type and the information (named features).

In each object of the features list, we also have different objects types :

```
geo_borough["features"][0].keys()
```

⇨ dict_keys(['type', 'geometry', 'properties'])

In the properties field, there are various useful information, including the longitude and latitude coordinates of the center of the borough (or of the neighbourhood).

```
geo_borough["features"][0]['properties']
{'n_sq_co': 750001537,
 'perimetre': 4519.26364836,                    => borough perimeter
 'l_ar': '3ème Ardt',                           => borough number
 'surface': 1170882.82818778,
 'geom_x_y': [48.86287238, 2.3600009859],       => latitude, longitude
 'n_sq_ar': 750000003,
 'l_aroff': 'Temple',                          => borough name
 'c_arinsee': 75103,                           => borough postal code
        'c_ar': 3}
```

```
geo_neighbourhood["features"][0]['properties']
{'n_sq_qu': 750000015,
 'n_sq_ar': 750000004,
```

```
 'geom_x_y': [48.851585175, 2.36476795387],=>neighbourhood lat & l
on
 'c_qu': 15,                                  =>neighbourhood number
 'surface': 487264.93707154,
 'l_qu': 'Arsenal',                           => neighbourhood name
 'perimetre': 2878.55965556,
 'c_quinsee': 7510403,
'c_ar': 4}                                    => borough number
```

For the rest of the project, with the Panda library imported previously, I have created 2 DataFrames with the useful information :

```
# Borough Dataframe
bouroughDF = pd.DataFrame({
    "BoroughNumber"    : [bor["properties"]["c_ar"] for bor in geo_borough["features"]]
   ,"Borough"          : [bor["properties"]["l_aroff"] for bor in geo_borough["features"]]
   ,"Latitude"   : [bor["properties"]["geom_x_y"][0] for bor in geo_borough["features"]]
   ,"Longitude" : [bor["properties"]["geom_x_y"][1] for bor in geo_borough["features"]]
   })
```

```
bouroughDF
```

	BoroughNumber	Borough	Latitude	Longitude
0	3	Temple	48.862872	2.360001
1	7	Palais-Bourbon	48.856174	2.312188
2	13	Gobelins	48.828388	2.362272
3	17	Batignolles-Monceau	48.887327	2.306777
4	20	Ménilmontant	48.863461	2.401188

⇨

```
# Neighbourhood Dataframe
neighbourhoodDF = pd.DataFrame({
   "BoroughNumber"  : [neigh["properties"]["c_ar"] for neigh in geo_neighbourhood["features"]]
   ,"NeighNumber"   : [neigh["properties"]['c_qu'] for neigh in geo_neighbourhood["features"]]
   ,"Neighbourhood" : [neigh["properties"]["l_qu"] for neigh in geo_neighbourhood["features"]]
})
```

```
neighbourhoodDF
```

	BoroughNumber	NeighNumber	Neighbourhood
0	4	15	Arsenal
1	5	18	Jardin-des-Plantes
2	10	39	Porte-Saint-Martin
3	11	43	Roquette
4	12	46	Picpus

⇨

After cleaning, merging, combining the values of the data frames, I obtain the following data frame for the rest of the study:

```
#concat neighborhood
postalCodesDF_combined = borough_neighbourhoodDF.groupby(['BoroughNumber', 'Bor
ough','Latitude','Longitude'])['Neighbourhood'].apply(lambda x: "%s" % ', '.join(x))
#convert into dataframe
postalCodesDF_combined = postalCodesDF_combined.to_frame().reset_index()
postalCodesDF_combined
```

	BoroughNumber	Borough	Latitude	Longitude	Neighbourhood
0	1	Louvre	48.862563	2.336443	Palais-Royal, Saint-Germain-l'Auxerrois, Halle...
1	2	Bourse	48.868279	2.342803	Vivienne, Mail, Bonne-Nouvelle, Gaillon
2	3	Temple	48.862872	2.360001	Sainte-Avoie, Arts-et-Métiers, Archives, Enfan...
3	4	Hôtel-de-Ville	48.854341	2.357630	Arsenal, Saint-Gervais, Saint-Merri, Notre-Dame
4	5	Panthéon	48.844443	2.350715	Jardin-des-Plantes, Sorbonne, Saint-Victor, Va...

2. The geographical coordinates of Paris were extracted using GeoPy libray in Python. The geographic coordinates will be used to draw the map of Paris with arrondissements colored according to the types of restaurants that are present.

```
address = 'PARIS'
geolocator = Nominatim(user_agent="to_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
```

```
longitude = location.longitude

print('The geograpical coordinate of PARIS are {}, {}.'.format(latitude, longitude))
```

⇨ The geograpical coordinate of PARIS are 48.8566969, 2.3514616

3. Top venues data will be obtained from Foursquare through an API :

   Venue data was extracted using the Foursquare API. These data made it possible to study the different types of restaurants located in the Parisian districts.
   These data were then used to give the conclusions of the project.

Methodology
Results
Discussion
Conclusion
References Acknowledgement & Sources

REFERENCES

Source : https://en.wikipedia.org/wiki/Arrondissements_of_Paris

Source : https://www.parisinsidersguide.com/paris-neighborhoods.html

Source : https://fr.wikipedia.org/wiki/Liste_des_quartiers_administratifs_de_Paris

LINKS:

The notebook with the code for this project, as well as the report, can be found in my github repository.