IBM Data Science Professional Certificated

Capstone Project

Final Report

Zinan Lin

May 21st, 2019

Table of Contents

Introduction

Background

Paris is the captain city of France, it is famous for its history, arts, and food. Paris has 20 arrondissements, which are regions that the city is separated into, like the picture [1] showed below. There are about 40,000 [2] restaurants in Paris and it is currently the city with the second most Michelin starred restaurants in the world [3]. I am a food lover and I am motivated to find out more about the fine-dining places in Paris.

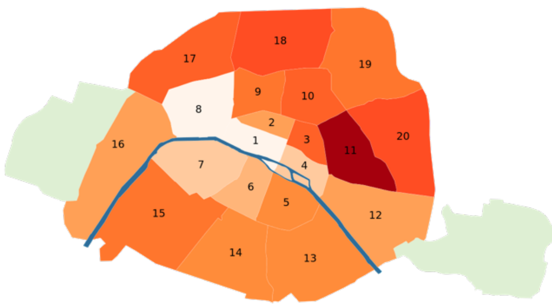


Fig. 1: The arrondissements of Paris

Problem Scope

What are the similarities in the arrondissements of Paris in terms of restaurants?
Are the restaurants in the central (downtown) areas more favored by clients?
Optionally, how does the number of Michelin Starred restaurants affect the way that that restaurants are clustered?

We will need to find out about the above problems with the support of datasets.

Audience

The audience of this problem are tourist, who planned to stay in Paris for vacation. Knowing which area/ areas are similar in terms of restaurants can help them deciding on where to live and where to eat.

This problem can also benefit restaurant owners/ businessmen who are interested in knowing the distribution of restaurants in Paris

Datasets

Data of Michelin Starred Restaurants

Source

List of all Michelin starred restaurants:
https://www.theupcoming.co.uk/2019/01/21/all-the-paris-michelin-star-restaurants-2019-on-a-map-and-full-list/

Description

Either using beautiful soup to scrape the web page or manually download the lines into an excel file. The data size is small (~100 rows, 2 columns) and the webpage is not very well organized, therefore it’s easy to manually create an excel file.

Example

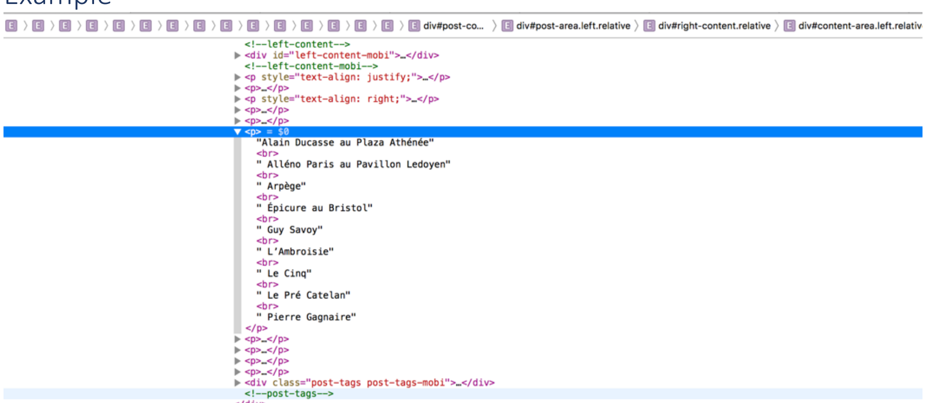
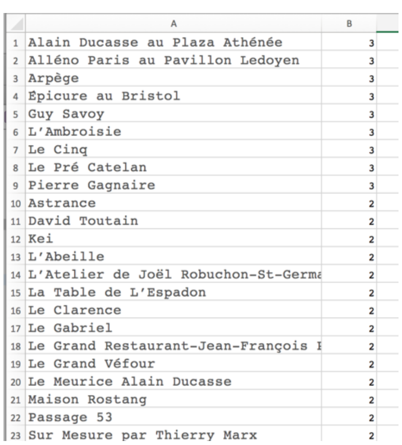


Fig 2. The website HTML, showing the labels of Michelin starred restaurants



	A	B
1	Alain Ducasse au Plaza Athénée	3
2	Alléno Paris au Pavillon Ledoyen	3
3	Arpège	3
4	Epicure au Bristol	3
5	Guy Savoy	3
6	L’Ambroisie	3
7	Le Cinq	3
8	Le Pré Catelan	3
9	Pierre Gagnaire	3
10	Astrance	2
11	David Toutain	2
12	Kei	2
13	L’Abeille	2
14	L’Atelier de Joël Robuchon-St-Germa	2
15	La Table de L’Espadon	2
16	Le Clarence	2
17	Le Gabriel	2
18	Le Grand Restaurant-Jean-François I	2
19	Le Grand Véfour	2
20	Le Meurice Alain Ducasse	2
21	Maison Rostang	2
22	Passage 53	2
23	Sur Mesure par Thierry Marx	2

Fig 3. Manually downloaded data

Data of arrondissements in Paris

Source

List of arrondissements in Paris
https://opendata.paris.fr/explore/dataset/arrondissements/export/

Description

The data is nicely formatted from the official website of Paris open Data. It supports download, so I will import the data as a csv file using pandas.

Example

1.2 Load the arrondissements data of Paris

In [965]:
```
df_paris_raw = pd.read_csv('arrondissements.csv', sep=';', header=0)
df_paris_raw = df_paris_raw[['C_AR', 'L_AROFF', 'Geometry X Y', 'Geometry']].rename(
    columns={'C_AR':'area_code', 'L_AROFF':'name', 'Geometry X Y':'coordinates'}).sort_values(by=['area_code']).reset_
df_paris_raw
```

Out[965]:

	area_code	name	coordinates	Geometry
0	1	Louvre	48.8625627018, 2.33644336205	{"type": "Polygon", "coordinates": [[[2.328007...
1	2	Bourse	48.8682792225, 2.34280254689	{"type": "Polygon", "coordinates": [[[2.351518...
2	3	Temple	48.86287238, 2.3600009859	{"type": "Polygon", "coordinates": [[[2.363828...
3	4	Hôtel-de-Ville	48.8543414263, 2.35762962032	{"type": "Polygon", "coordinates": [[[2.368512...
4	5	Panthéon	48.8444431505, 2.35071460958	{"type": "Polygon", "coordinates": [[[2.364433...
5	6	Luxembourg	48.8491303586, 2.33289799905	{"type": "Polygon", "coordinates": [[[2.344592...
6	7	Palais-Bourbon	48.8561744288, 2.31218769148	{"type": "Polygon", "coordinates": [[[2.320902...
7	8	Élysée	48.8727208374, 2.3125540224	{"type": "Polygon", "coordinates": [[[2.325836...
8	9	Opéra	48.8771635173, 2.33745754348	{"type": "Polygon", "coordinates": [[[2.339776...
9	10	Entrepôt	48.8761300365, 2.36072848785	{"type": "Polygon", "coordinates": [[[2.364685...
10	11	Popincourt	48.8590592213, 2.3800583082	{"type": "Polygon", "coordinates": [[[2.396236...
11	12	Reuilly	48.8349743815, 2.42132490078	{"type": "Polygon", "coordinates": [[[2.413879...
12	13	Gobelins	48.8283880317, 2.36227244042	{"type": "Polygon", "coordinates": [[[2.374913...
13	14	Observatoire	48.8292445005, 2.3265420442	{"type": "Polygon", "coordinates": [[[2.333806...
14	15	Vaugirard	48.8400853759, 2.29282582242	{"type": "Polygon", "coordinates": [[[2.299322...
15	16	Passy	48.8603921054, 2.26197078836	{"type": "Polygon", "coordinates": [[[2.274268...
16	17	Batignolles-Monceau	48.887326522, 2.30677699057	{"type": "Polygon", "coordinates": [[[2.295166...
17	18	Buttes-Montmartre	48.892569268, 2.34816051956	{"type": "Polygon", "coordinates": [[[2.365803...
18	19	Buttes-Chaumont	48.8870759966, 2.38482096015	{"type": "Polygon", "coordinates": [[[2.389428...
19	20	Ménilmontant	48.8634605789, 2.40118812928	{"type": "Polygon", "coordinates": [[[2.412765...

Fig 4. Code Snippet of importing arrondissement data as a data frame

Data of Restaurant Information

Source

Kaggle, TripAdvisor Restaurants Info for 31 Euro-Cities:
https://www.kaggle.com/damienbeneschi/krakow-ta-restaurans-data-raw

Description

This is a dataset consisting Ratings and reviews for restaurants across 31 European cities, it's 28.7MB. I will download it and import it as a CSV file. Later on, I will filter out only restaurants from Paris and use it to find out the details about restaurants I will be analyzing.

Example

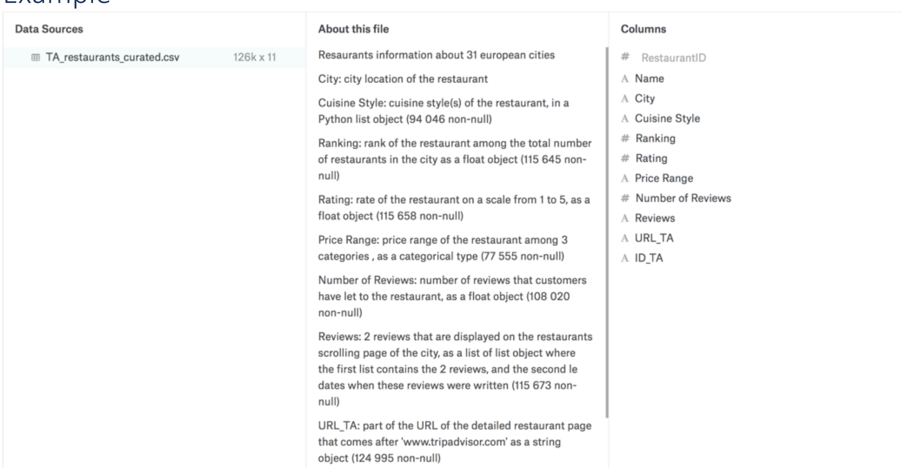


Fig 5, High level Overview of the data

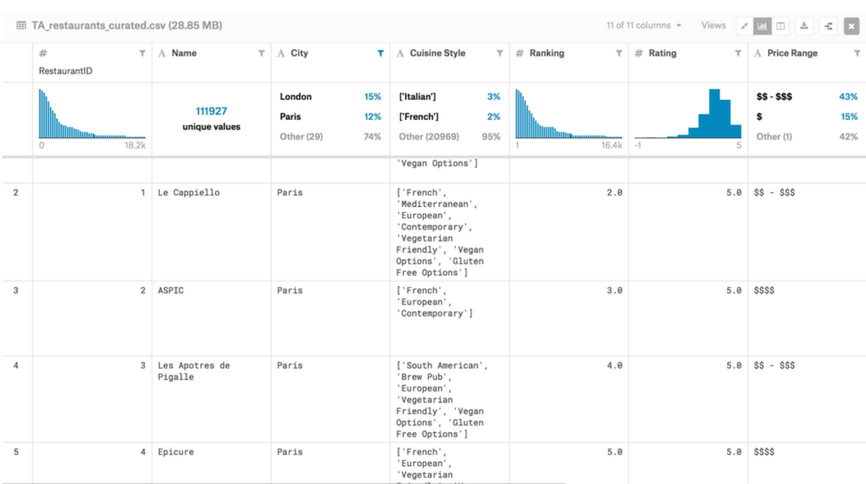


Fig 6, Example of the data

Data from Foursquare API

Source

Foursquare API:
https://foursquare.com/developers/apps

Description

Using Foursquare API and it's search endpoint, I will be able to get data on which arrondissement are restaurants from. As my current subscription, I can make 99,500 Regular Calls / Day, which should be more than enough for me as the total number of restaurants in Paris is about 40,000 [2].

Example

Documentation for search endpoint:
https://developer.foursquare.com/docs/api/venues/search

```
In [932]: def getNearbyVenues(names, latitudes, longitudes, radius=2300, LIMIT=150000):

              venues_list=[]
              food = '4d4b7105d754a06374d81259'  # catagory for food
              intent = 'browse'

              for name, lat, lng in zip(names, latitudes, longitudes):
                  print(name)

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

                  # make the GET request
                  results = requests.get(url).json()["response"]["venues"]

                  # return only relevant information for each nearby venue
                  venues_list.append([(
                      name,
                      lat,
                      lng,
                      v['name'],
                      v['location']['lat'],
                      v['location']['lng']
                  ) 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']

              return(nearby_venues)
```

Fig 7, Code snippet of the function that makes call to the API

Methodology

Data Collecting

I am working with four data frames, three of which are directly imported from files: they are df_michelin, df_paris and df_restaurants.

df_venues were from Foursqaure API, by specifying food = '4d4b7105d754a06374d81259', I was able to only get food category from the venues near the input coordinates --- the input coordinates are the coordinates of the arrondissements from df_paris.

Please look at the link to the notebook [4] for more about the code of data collection.

Data Wrangling

I started by cleaning the data, cut off unwanted columns in each data frame and drop NaN values where I need the label for future analysis.

Other than eliminating anomalies, data normalization is important as well. In df_restaurant, there are “ratings” and “number of reviews”. Using “rating” alone will not be good enough because, for example, a restaurant is rated 5 by 3 users is not necessarily better than a restaurant that is rated 4.5 by 30 users. Therefore, I will need to normalize the rating based on the number of reviewers.

I used Bayesian estimate of the weighted review as following:

calculate a Bayesian estimate of the weighted review, using $(WR) = (v \div (v + m)) \times R + (m \div (v + m)) \times C$

where:

R = average for the rating (mean)

v = number of reviews

m = minimum reviews required to be listed

C = the mean reviews across the whole report

Fig 8, Normalization equation [5]

Rating before the Normalization:

```
Out[789]: Text(0.5, 1.0, 'Review distribution before Normalization')
```

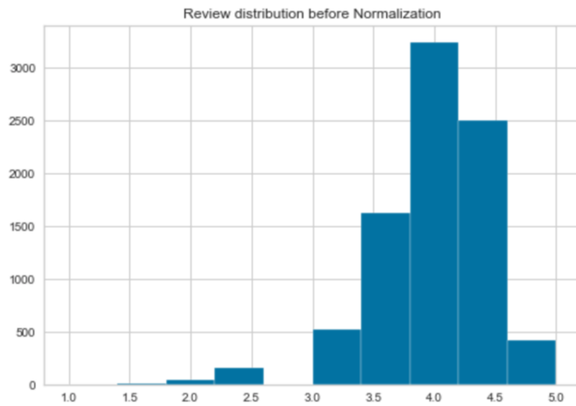


Fig 9, Rating Histogram before the Normalization

Rating after the Normalization:

```
Out[792]: Text(0.5, 1.0, 'Review Distribution after Normalization')
```

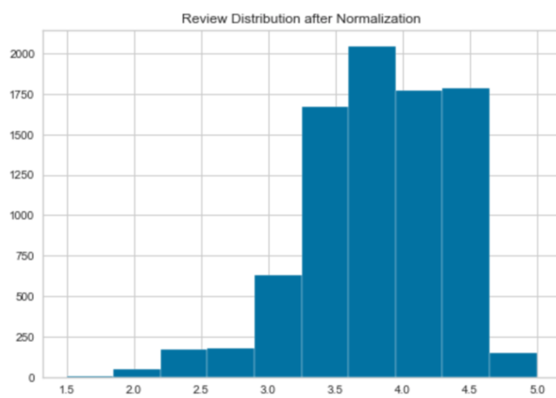


Fig 10, Rating Histogram after the Normalization

Distribution of the Number of Reviews:

```
Out[788]: <matplotlib.axes._subplots.AxesSubplot at 0x1a92305b00>
```

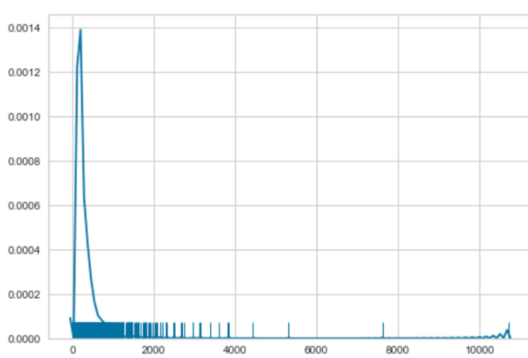


Fig 11, Distribution of the Number of Reviews

Data Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. [6]

In order to get a better understanding of the datasets, I employed some data visualization in the data exploring process.

For df_paris, I used folium to visualize the map with each arrondissement outlined:

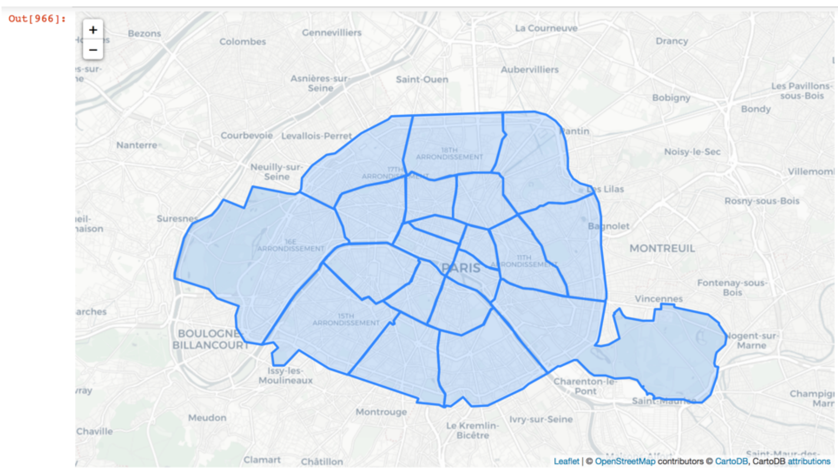


Fig 12, Map of Paris

For ratings of the restaurants, I plot rating with box plot in ascending order against each arrondissement to have an intuitive view of the data.

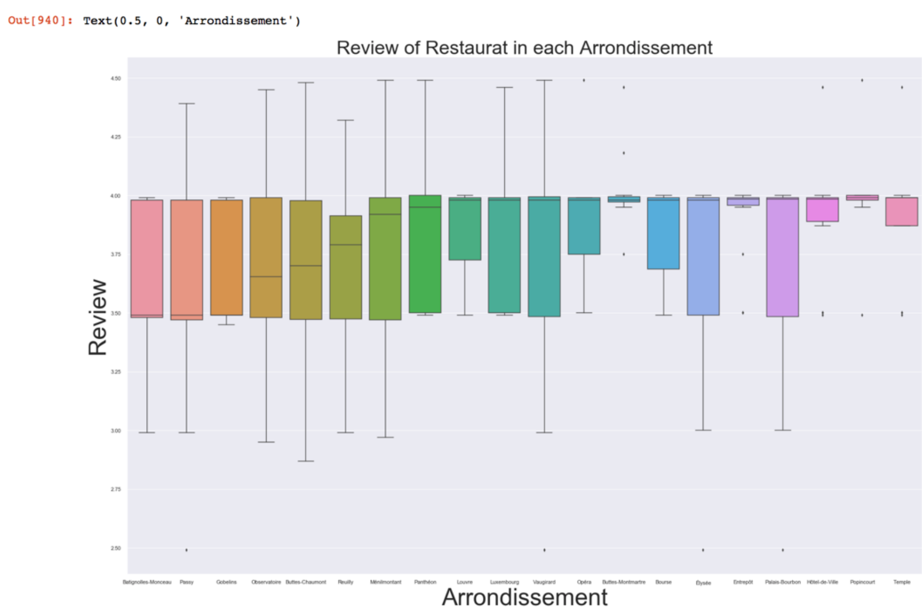


Fig 13, Box plot

For the percentage of the distribution of Michelin starred restaurants, I chose to use a pie plot. The result wasn't satisfying but I will cover the reason in the discussion section.

```
Out[942]: Text(0.5, 1.0, 'Percentage of Michelin starred restaurant')
```

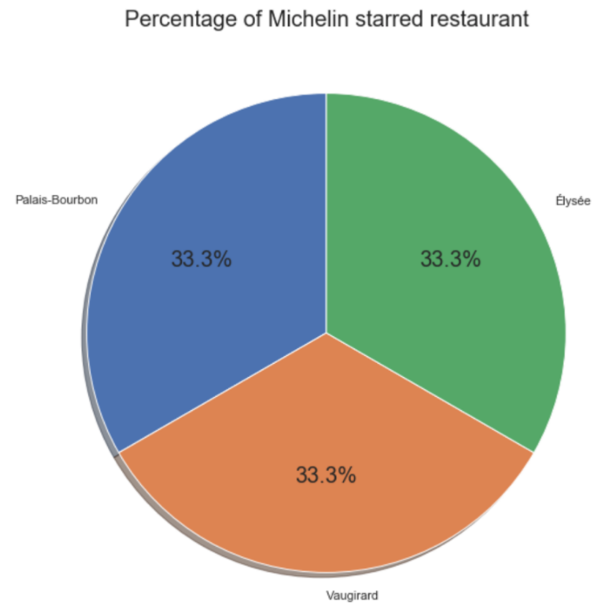


Fig 14, pie plot

Another Example of data visualization would be the plot for finding elbow point and visualize the clustered label on the map, which will be covered in the next section.

K-means Clustering

k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. [6]

In this project, I aim to utilize k mean method to cluster restaurant data geometrically to find similar groups in terms of their location.

I first use one-hot encoding to separate categorized fields in binary, such as price range and normalized rating. Then I use the elbow method to find the optimal k, the Elbow method is a method of interpretation and validation of consistency within cluster analysis designed to help finding the appropriate number of clusters in a dataset. [7]

The code I used for plotting the elbow point graph comes from KElbowVisualizer library. Details can be found on https://www.scikit-yb.org/en/latest/api/cluster/elbow.html.

Here is the plot:

Fig 15, elbow point plot

It is not a pretty plot and the distribution score does not provide an obvious K, I chose k=5 to move on.

After using scikit library for k-mean cluster and adding the label back into the data frame, I made the following plot overlaying with Fig. 12 to see how the restaurants are clustered:

Out[1011]:

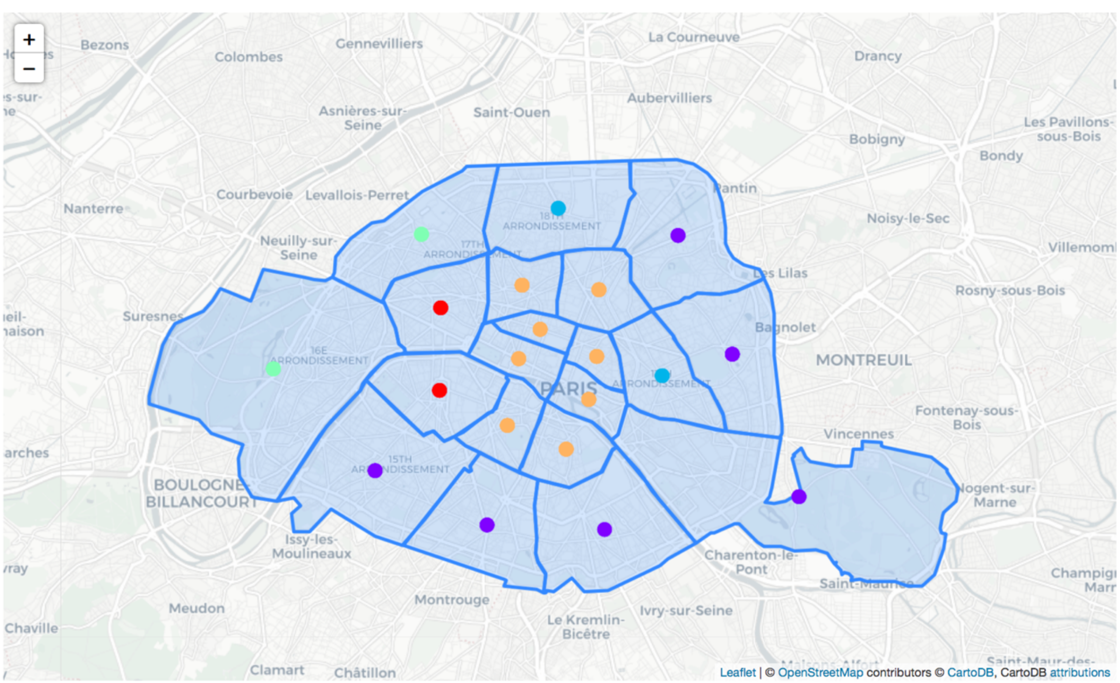


Fig 16, cluster label on Paris map

Results

The results are quite intuitive with the plot above (Fig 16), the restaurants near the central city are clustered together and the ones further away from the city are clustered together.

If we take a closer look at the clustered results, we can see that the central city cluster tends to have more expensive and highly rated restaurants than the ones further away from downtown:

------------ cluster 1 --------------

	Name	Price Range	Review
0	Le Nemours	Moderate	3.98
1	Le Nemours	Moderate	3.98
2	Buddha Bar	High	3.50
3	Buddha Bar	High	3.50
4	Le Fumoir	Moderate	3.99
5	Carette	Moderate	3.99
6	Carette	Moderate	3.99
7	Chez Francis	Moderate	3.00
8	Chez Francis	Moderate	3.00
9	Le Jules Verne	High	4.00
10	Le Jules Verne	High	4.00
11	Le Fouquet's	High	3.99
12	Le Fouquet's	High	3.99
13	Monsieur Bleu	High	3.99
14	Monsieur Bleu	High	3.99
15	Le Grand Corona	Moderate	2.49
16	Le Grand Corona	Moderate	2.49
17	Noura Marceau	Moderate	3.48
18	Noura Marceau	Moderate	3.48
19	Bouillon Pigalle	Moderate	3.95
20	Pink Mamma	Moderate	3.98

------------ cluster 2 --------------

	Name	Price Range	Review
0	Chez Prune	Moderate	3.98
1	Le Pavillon du Lac	Moderate	3.48
2	La Bellevilloise	Moderate	3.48
3	La Bellevilloise	Moderate	3.48
4	La Fontaine de Belleville	Moderate	2.87
5	Siseng	Moderate	4.48
6	Triplettes	Moderate	3.44
7	Triplettes	Moderate	3.44
8	Dong Huong	Low	3.97
9	Dong Huong	Low	3.97
10	Le Bastringue	Moderate	3.98
11	Moncoeur Belleville	Moderate	3.47
12	Moncoeur Belleville	Moderate	3.47
13	Tripletta	Moderate	3.92
14	Tripletta	Moderate	3.92
15	The Frog & British Library	Moderate	3.49
16	Chez Lili et Marcel	Moderate	3.99
17	Aux Cadrans	Moderate	3.98
18	Le Lakanal	Moderate	3.45
19	Tricotin	Low	3.49
20	Ober Mamma	Moderate	3.99
21	East Mamma	Moderate	4.49
22	Chez Prosper	Moderate	3.49
23	Chez Prosper	Moderate	3.49
24	Le Dalou	Moderate	2.99
25	Le Dalou	Moderate	2.99
26	Clamato	Moderate	3.98
27	Le Rey	Moderate	3.46
28	Aux Ours	Low	3.44
29	La Cave de Septime	Moderate	4.30
30	Melt	Moderate	4.46

------------ cluster 3 --------------

	Name	Price Range	Review
0	Bouillon Pigalle	Moderate	3.95
1	Pink Mamma	Moderate	3.98
2	La Maison Rose	Moderate	3.75
3	Mamma Primi	Moderate	3.98
4	Terminus Nord	Moderate	4.00
5	La Maison Bleue	Moderate	3.98
6	Le Hasard Ludique	Moderate	4.18
7	La Marmite	Moderate	3.98
8	Soul Kitchen	Moderate	4.46
9	Corso Trudaine	Moderate	3.97
10	L'As du Fallafel	Low	4.00
11	Les Philosophes	Moderate	4.00
12	Season	Moderate	3.98
13	Ober Mamma	Moderate	3.99
14	Le Voltigeur	Moderate	3.95
15	Carette	Moderate	3.99
16	Benedict	Moderate	3.99
17	Chez Janou	Moderate	4.00
18	Chez Prune	Moderate	3.98
19	East Mamma	Moderate	4.49
20	Les Marronniers	Moderate	3.49

------------ cluster 4 --------------

	Name	Price Range	Review
0	Bouillon Pigalle	Moderate	3.95
1	Pink Mamma	Moderate	3.98
2	Le Fouquet's	High	3.99
3	Al Ajami	High	3.48
4	Mamma Primi	Moderate	3.98
5	Triadou Haussmann	Moderate	3.49
6	Le Percier	Moderate	3.43
7	Le Deauville	Moderate	2.99
8	La Maison de l'Aubrac	Moderate	3.49
9	Carette	Moderate	3.99
10	Le Bistro Parisien	Moderate	2.99
11	Chez Ribe	Moderate	3.49
12	Le Bailli de Suffren	Moderate	2.49
13	L'Abreuvoir	Moderate	4.39
14	Chipotle Mexican Grill	Moderate	3.50
15	Le Malakoff	Moderate	3.49
16	La Rotonde de la Muette	Moderate	3.96
17	Le Flandrin	High	3.47
18	Auteuil Brasserie	Moderate	2.99
19	Scossa	Moderate	3.47
20	Schwartz's Deli	Moderate	3.99
21	Frog XVI	Moderate	3.98
22	Le Wilson	Moderate	3.49
23	Le Murat	High	3.48
24	Le Stella	Moderate	3.98
25	Il Cottage	Moderate	3.97

Fig 17, details of each cluster

To answer the following question:

What are the similarities in the arrondissements of Paris in terms of restaurants?

It seems like the distance to the centre/ downtown of the city has a linear relationship with regard to the characteristic of the restaurants.

Are the restaurants in the central (downtown) areas more favored by clients?

From our clustered ratings, yes.

Optionally, how does the number of Michelin Starred restaurants affect the way that that restaurants are clustered?
Non-conclusive, will need more data and analysis.

Discussion

There are a couple points to discussion for this project

1, the data are limited --- as Foursquare get detailed info about a restaurant is a premium endpoint, I only have limited called per day therefore I had to use the Kaggle dataset as an alternative. Resulting in limited resources.

2, the radius of which I search for venues (in this case, restaurants) are not rigorous since the actual geographical radius are unknown to me. I specified 2500 as a magic number, which can surely be improved.

3, the choice of k was not careful enough as I did not spend more time in investigation.

4, the extend of how much the number of Michelin starred restaurant affects the clustering was non-conclusive due to the small number of Michelin starred restaurant details I was able to get.

Conclusion

I am happy that I utilized the knowledge I learned throughout the course in this project. Although there are many imperfections, I am satisfied with the results and output of this project given the limited time frame. I look forward to expanding and carry on with this idea and hope to learn more about machine learning.

Good luck to you as well, my fellow classmates.

Reference

[1] https://en.wikipedia.org/wiki/Arrondissements_of_Paris
[2] https://www.quora.com/How-many-restaurants-are-in-Paris
[3] https://www.godsavethepoints.com/2019/02/15/13-most-michelin-starred-cities-in-the-world/
[4] https://github.com/linzinan/IBM-Data-Science-Certificate/blob/master/ParisCuisine.ipynb
[5] https://stackoverflow.com/questions/8542391/how-to-normalize-reviews-based-on-score
[6] https://www.tableau.com/learn/articles/data-visualization