Professional Data Science Certificate

Capstone Project _ Week 2 _ 03/29/2020

1. Introduction (1/4)

When customers choose to visit a restaurant, they are looking for a memorable gastronomical experience. This experience depends on different parameters:

- quality, freshness, and taste of food;
- décor, ambiance and hygiene level;
- staff courtesy and prompt service…

Soliciting customer feedback can give to the restaurant owners an insight into what customers like about the restaurant and what they expect.

Based on comments and opinions from customers, restaurant owners can work on customer loyalty.

1. Introduction (2/4)

Unfortunately, from a research published by Shephyken ('Create A Customer Service Culture):

43% of customers who were surveyed stated that they don't complain or leave feedback because they don't think 'that the business cares.' Out of these same customers, 81% admitted that they would be willing to give feedback if they are assured that they would get a fast response.

Now that it has been established that customer feedback is extremely important, the next is to understand how to gather this feedback.

1. Introduction (3/4)

What are the different methods of collecting customer feedback ?

1. Face-to-Face Feedback: When the waitstaff directly asks the customer for feedback.
2. Feedback Forms: Distributing printed forms among customers to obtain their feedback.
3. Mobile POS Feedback System: Using a tablet connected to POS to collect customer feedback based on what the customer had ordered and how their particular experience was.
4. Social Media Feedback: Responding to comments on different social media apps and platforms like Zomato is important for online reputation and understanding the customer feedback in the restaurant.
5. Online Customer Feedback: Another technique to obtain customer feedback is to have an online survey form which the customers can fill at their leisure.

1. Introduction (4/4)

What are the different methods of collecting customer feedback ?

6. Focus Groups: Collecting true customer feedback by creating focus groups of target audience and interviewing them. This method of customer feedback collection should be carried out periodically.
7. Community Groups And Discussion Boards: Be a part of different social media groups and communities to understand what the customers are saying about a specific restaurant.
8. Web Analytics: Use POS and analyze the performance of various dishes and different feedbacks collected from customers to understand what they like about you as a whole.

Key point: behind all these different methods, one key point is machine learning algorithm choice and efficiency: this project demonstrates importance of this point.

2. Data (1/5)

Machine algorithms are tested and selected on different datasets (2 datasets in this project).

These datasets are obtained from a few restaurant owners who accept to collect and share data.
Data collection method: no question is asked to customers. Instead of that, the restaurant owners who participate to this study, record 3 elements for each of their customers:

- Is this customer a regular one ?
- How much time is measured between customer' s arrival in the restaurant and the real lunch/dinner start time ? (fitting with waiting time from the customer)
- What is the bill total for this lunch/dinner ?

These datasets merge two populations among the customers:
Regular customers;
Customers who visited once a restaurant, and didn't come back.

2. Data (2/5)

One point that could be discussed in this study is about the small number of parameters taken into account in what a feedback directly received from customers could really collect: the 2 parameters studied are:

- Waiting time (= WAITING in the datasets, expressed in minutes);
- Bill total (= BILL in the datasets, expressed in euros).

Actually, there are different significant advantages of this data collection type with only 2 parameters:

- Time and money will always stay very important parameters, or key parameters, even if relations with regular customers are built thanks to many other factors;
- These are parameters that are collected by restaurant owners themselves (no issue of missing data, no staff involvement in this collecting task, continuous data, factual data);
- In this project context (or for a potential demonstration by a company to restaurant owners about company skills in machine learning), a study with only 2 parameters can allow to visualize easily machine learning choice and efficiency as a key point for every analysis.

2. Data (3/5)

Data collected for regular customers, in 6 restaurants in Paris, France:

Restaurant Id	Minimal bill (€)	Maximal bill (€)	Minimal waiting time (mn)	Maximal waiting time (mn)
1	17	19	1	5
2	20	21	5	6
3	22	23	6	7
4	24	25	7	8.5
5	26	27	8.5	10
6	28	29	10	11.5

Data collected for passing customers, in 5 restaurants in Paris, France:

Restaurant Id	Minimal bill (€)	Maximal bill (€)	Mean waiting time (mn)
7	35	39	27
8	40	44	41
9	45	49	54
10	50	54	74
11	55	59	100

2. Data _ Data generating code (4/5)

```
# Data_Generation.py
# Version: 03/29/2020


#---- Module import --
import random
import pandas as pnd


#---- Data Features------

# Come back customers
comeback = [[17,19,1,5],[20,21,5,6],[22,23,6,7],[24,25,7,8.5],[26,27,8.5,10],[28,29,10,11.5]]


# Don't come back customers

#Case 1 to build customers1.csv :
#dont = [[40,44,41],[45,49,54],[50,54,74],[55,59,100]]

#Case 2 to build customers2.csv:
dont = [[35,39,27],[40,44,41],[45,49,54],[50,54,74],[55,59,100]]
```

1. Usage of the datasets shared by the 11 restaurant owners in Paris.

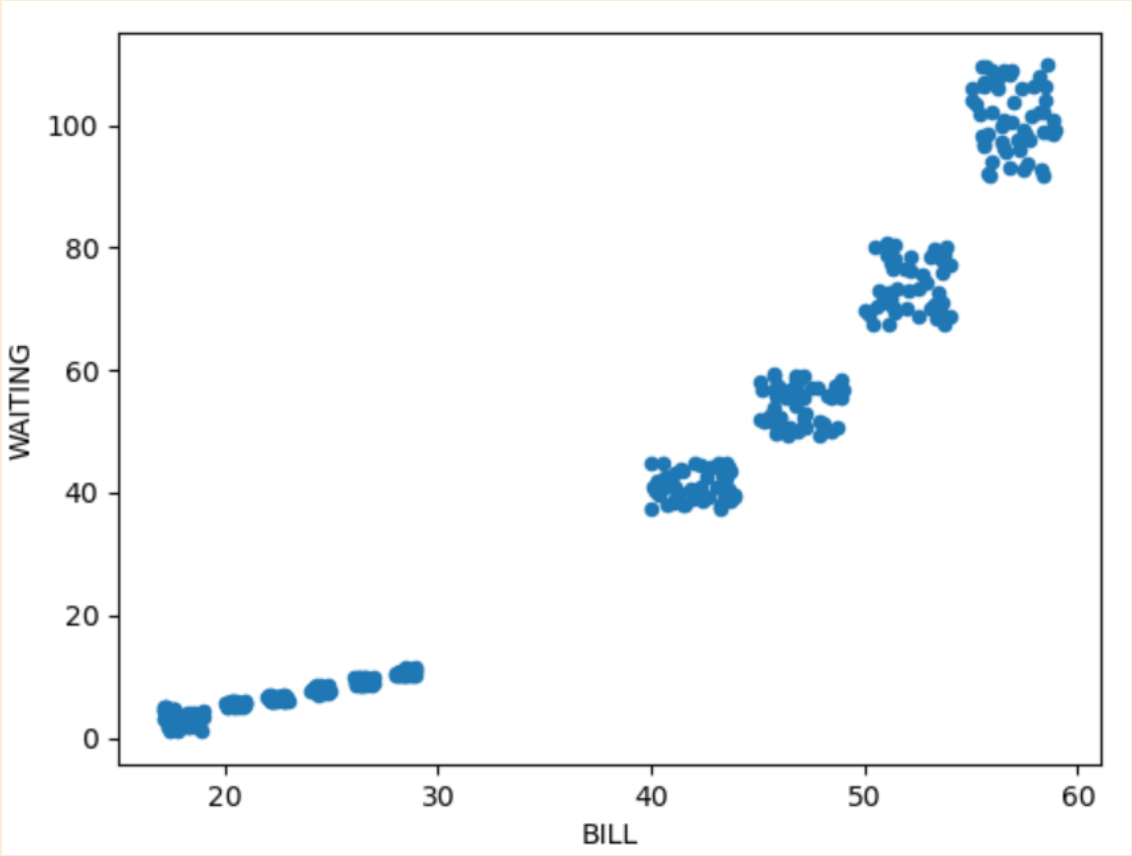
```
# Data generating [BILL, WAITING]
nombreObservations = 200

# Regular customers
cust = []
random.seed()
for iteration in range(nombreObservations):
    cust = random.choice(comeback)
    bill = round(random.uniform(cust[0], cust[1]),2)
    waiting = round(random.uniform(cust[2], cust[3]),2)
    cust.append([bill,waiting)

# Passing customers
nocust = []
random.seed()
for iteration in range(nombreObservations):
    nocust = random.choice(dont)
    bill = round(random.uniform(nocust[0], nocust[1]),2)
    minwaiting = dont[2] / 1.10
    maxwaiting = dont[2] * 1.10
    waiting = round(random.uniform(minwaiting, maxwaiting),2)
    nocust.append([bill,waiting])
```

2. Data generating for regular and passing customers.

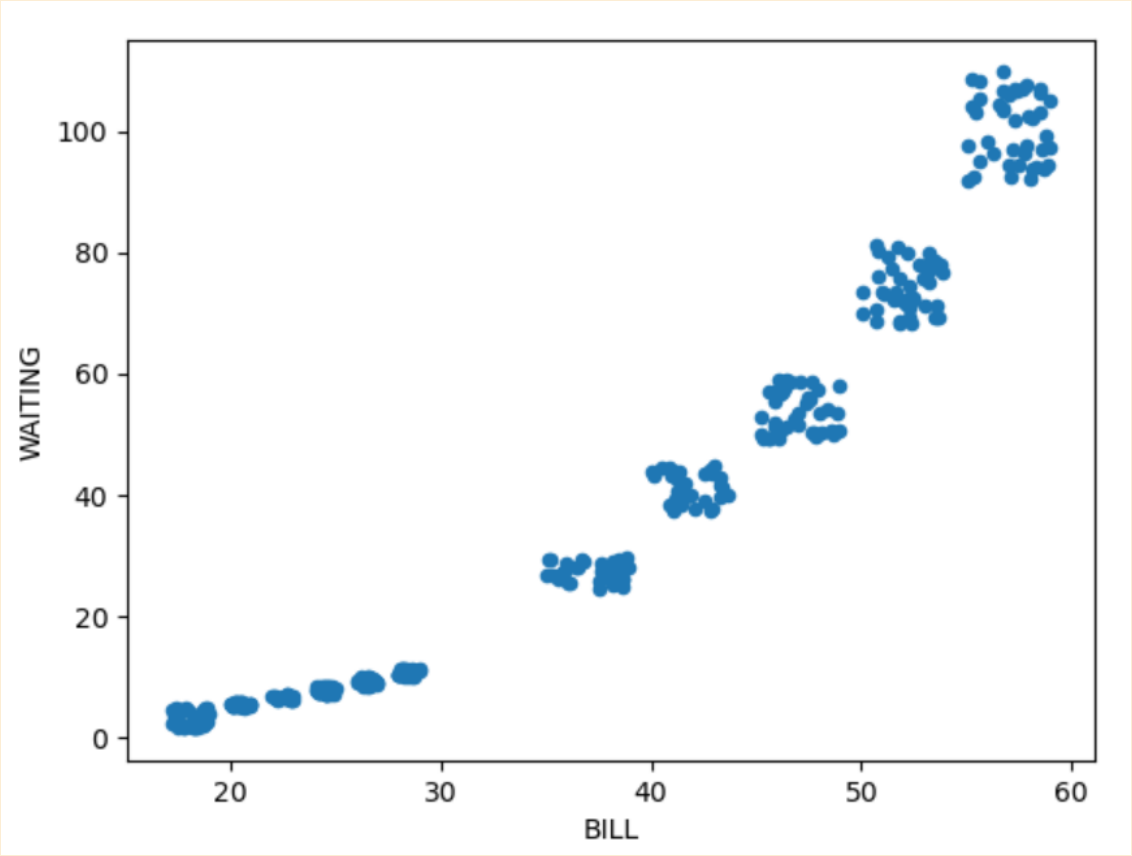
3. Data merging and saving in file.

```
# Final data generating
customers = cust+nocust
print(customers)

# Data mixing
random.shuffle(customers)

# Saving in a file
dataFrame = pnd.DataFrame(customers)
#dataFrame.to_csv("C:/ xxx /PycharmProjects/Clustering/datas/customers1.csv", index=False, header=False)
dataFrame.to_csv("C:/ xxx /PycharmProjects/Clustering/datas/customers2.csv", index=False,header=False)
```

2. Data plotting (5/5)



Dataset #1: ‘customers1.csv’



Dataset #2: ‘customers2.csv’

3. Methodology

Reminder: Project Purpose:
Design of a computer application for restaurant owners that allows them to predict if a customer will come back to their restaurants: this could allow them to manage better future potential investments: in a first step, a demonstration application is designed to be presented to the most frequented restaurants in each of the 20 Paris districts. (See later in chapter 6: Searching for the most frequented restaurants in Paris, using Foursquare.)

Implementation of a clustering algorithm with the following steps:

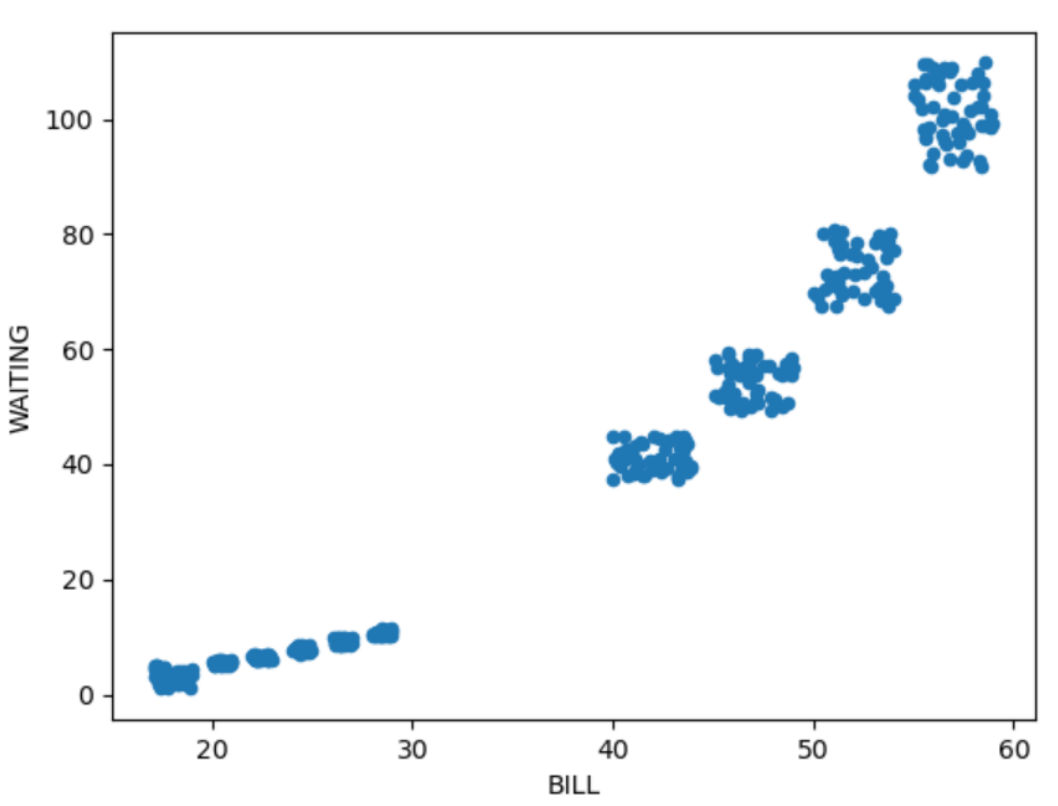
- Learning on data (search for the 2 clusters (regular customers / passing customers);
- Prediction with the algorithm settings found from the first step;
- Graphical plots of the clusters and centroids, for the learning data ;
- Test on other data, and discussion about classifying errors.

2 datasets are used, to compare efficiency (based on classifying error estimation) using 2 different clustering algorithms:

- K-Means
- Gaussian Mixture Models (GMM)

4. Results (1/7)

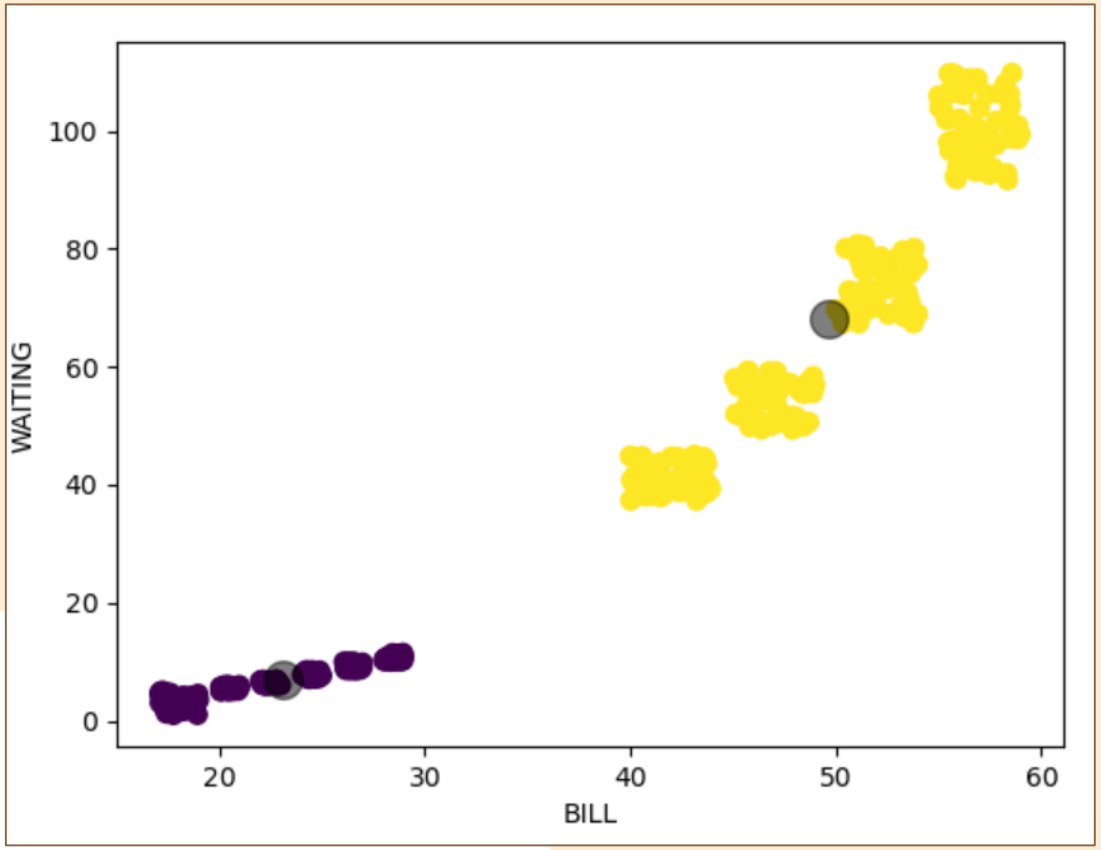
Step 1. Data plotting of the first dataset (customers1.csv)




```
# Version: 03/29/2020
#-------------------------------------------------------------------------------

# Module import
import pandas as pnd
import matplotlib.pyplot as plt

#-------------------------------------------------------------------------------
# K-Mean algorithm tested on a first data file

# Data loading
customers = pnd.read_csv("C:/ XXX /PycharmProjects/Clustering/datas/customers1.csv", names=['BILL','WAITING'], header=None)

# Graphical plotting
customers.plot.scatter(x="BILL",y="WAITING")
plt.show()
```

4. Results (2/7)

Step 2. K-Means algorithm implementation:
-Learning on dataset
-Classification: visualization of clustering result and plotting of the centroids




```
# Using K-Mean algorithm
from sklearn.cluster import KMeans
modele=KMeans(n_clusters=2)
modele.fit(customers)

# Predictions
predictions_kmeans = modele.predict(customers)

# Clustering plot
plt.scatter(customers.BILL, customers.WAITING, c=predictions_kmeans, s=50, cmap='viridis')
plt.xlabel("BILL")
plt.ylabel("WAITING")

# Centroid plotting
centers = modele.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5)
plt.show()
```

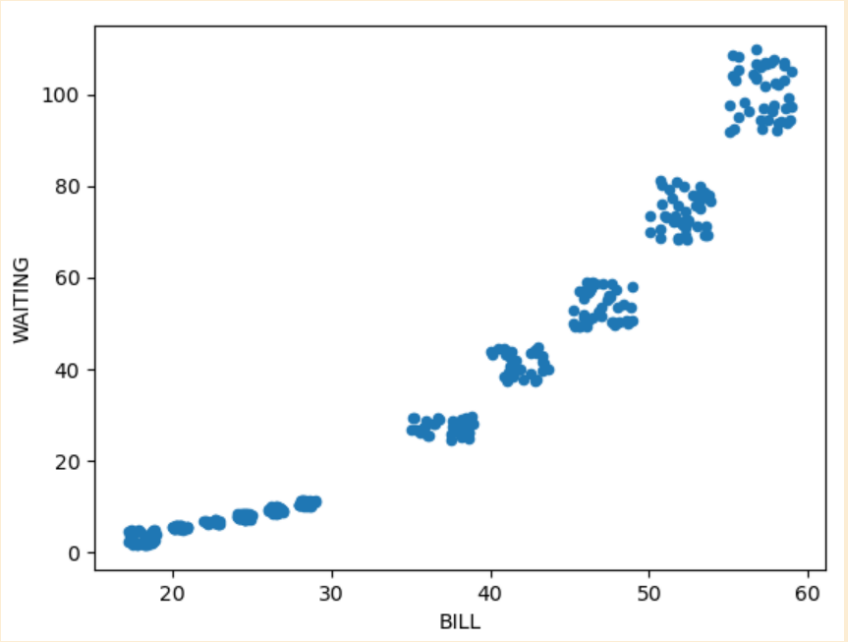
4. Results (3/7)

Step 3. Test/prediction on other data

```
# What is the cluster id for the customers who could come back ?
comeback = [[26.98,8.75]]
numCluster = modele.predict(comeback)
print("Cluster id for the customers who could come back: "+ str(numCluster))

# What is the cluster if for the customers who couldn't come back ?
wont = [[55.7,102.16]]
numCluster = modele.predict(wont)
print("Cluster id for the customers who couldn't come back: " + str(numCluster))

# Code to be adjusted according to the 2 previous results:
comeback = [[26.98,8.75]]
numCluster = modele.predict(comeback)
if int(numCluster)==1:
    print("This customer couldn't come back ...")
else:
    print("This customer could come back ... ")

wont = [[55.7,102.16]]
numCluster = modele.predict(wont)
if int(numCluster)==1:
    print("This customer couldn't come back ...")
else:
    print("This customer could come back ... ")
```

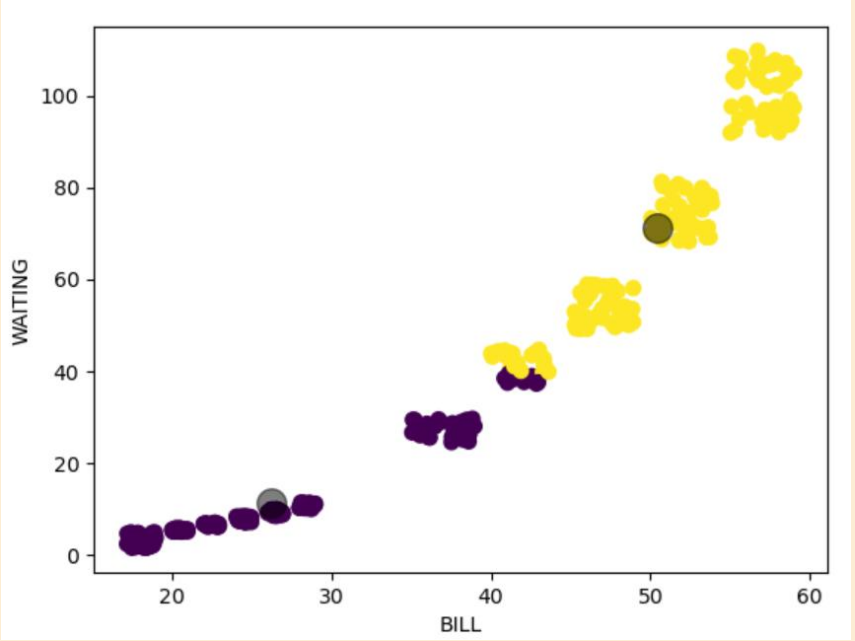
4. Results (4/7)

Step 4. Data plotting of the second dataset (customers2.csv)



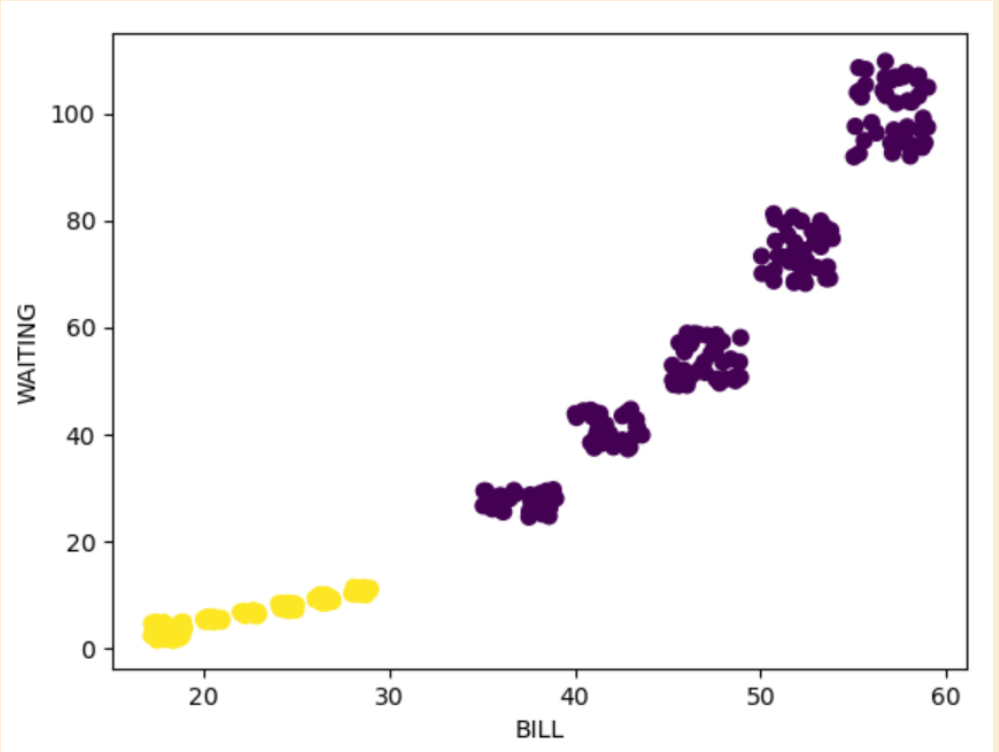
Step 5. K-Means algorithm implementation:
-Learning on dataset
-Classification: visualization of clustering result and plotting of the centroids
-Visualization of classification errors

```
# Data loading
customers = pnd.read_csv("C:/Users/nxa13794/PycharmProjects/Clustering/datas/customers2.csv", names=['BILL','WAITING'], header=None)

# Graphical plotting
customers.plot.scatter(x="BILL",y="WAITING")
plt.show()

# Using K-Mean algorithm
from sklearn.cluster import KMeans
modele=KMeans(n_clusters=2)
modele.fit(customers)

# Predictions
predictions_kmeans = modele.predict(customers)

# Clustering plot
plt.scatter(customers.BILL, customers.WAITING, c=predictions_kmeans, s=50, cmap='viridis')
plt.xlabel("BILL")
plt.ylabel("WAITING")

# Centroid plotting
centers = modele.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5)
plt.show()
```



4. Results (5/7)

Step 6. Gaussian Mixture Model (GMM) algorithm implementation:
-Learning on dataset
-Classification: visualization of clustering result and plotting of the centroids
-No classification error

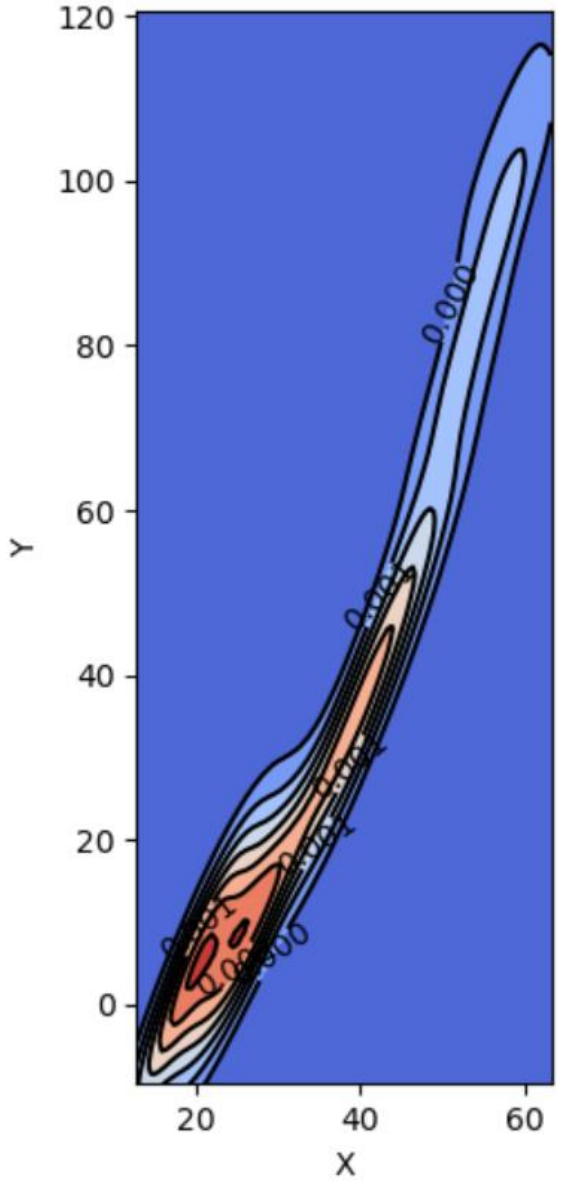



```
#-------------------------------------------------------------------------------
#---- Gaussian Mixture Model (GMM) -----------
from sklearn import mixture

# Clustering algorithm (2 clusters to be designed)
gmm = mixture.GaussianMixture(n_components=2)

# Learning
gmm.fit(customers)

# Clustering
clusters = gmm.predict(customers)

# Cluster plotting
plt.scatter(customers.BILL, customers.WAITING, c=clusters, s=40, cmap='viridis');
plt.xlabel("BILL")
plt.ylabel("WAITING")
plt.show()
```

4. Results (6/7)

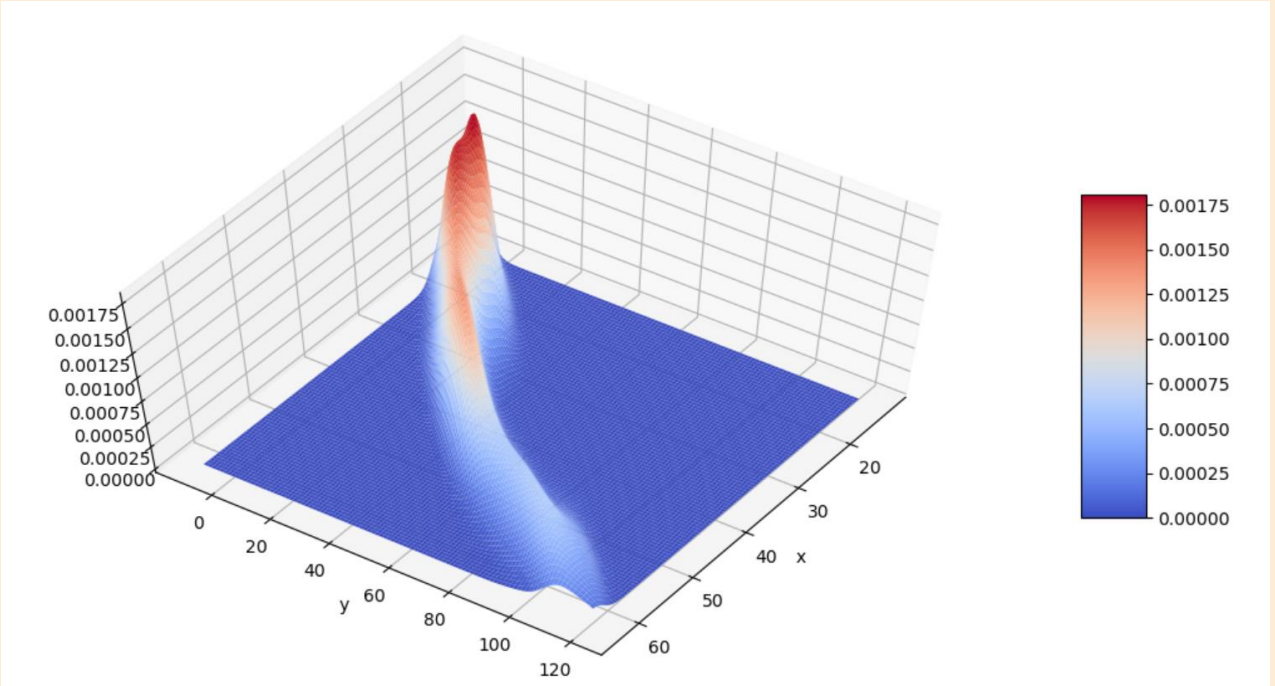
Step 7. Gaussian Mixture Model (GMM): 2D plots

```
import pandas as pnd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as st

customers = pnd.read_csv("C:/ XXX /PycharmProjects/Clustering/datas/customers.csv", names=['BILL','WAITING'], header=None)

n_components = 2

# Extract x and y
x = customers.BILL
y = customers.WAITING
# Define the borders
deltaX = (max(x) - min(x))/10
deltaY = (max(y) - min(y))/10
xmin = min(x) - deltaX
xmax = max(x) + deltaX
ymin = min(y) - deltaY
ymax = max(y) + deltaY
print(xmin, xmax, ymin, ymax)
# Create meshgrid
xx, yy = np.mgrid[xmin:xmax:100j, ymin:ymax:100j]

positions = np.vstack([xx.ravel(), yy.ravel()])
values = np.vstack([x, y])
kernel = st.gaussian_kde(values)
f = np.reshape(kernel(positions).T, xx.shape)

fig = plt.figure(figsize=(8,8))
ax = fig.gca()
ax.set_xlim(xmin, xmax)
ax.set_ylim(ymin, ymax)
cfset = ax.contourf(xx, yy, f, cmap='coolwarm')
ax.imshow(np.rot90(f), cmap='coolwarm', extent=[xmin, xmax, ymin, ymax])
cset = ax.contour(xx, yy, f, colors='k')
ax.clabel(cset, inline=1, fontsize=10)
ax.set_xlabel('X')
ax.set_ylabel('Y')
plt.show()
```



4. Results (7/7)

Step 8. Gaussian Mixture Model (GMM): 3D plots




```
from mpl_toolkits.mplot3d import axes3d, Axes3D
fig = plt.figure(figsize=(13, 7))
ax = plt.axes(projection='3d')
surf = ax.plot_surface(xx, yy, f, rstride=1, cstride=1, cmap='coolwarm', edgecolor='none')
ax.set_xlabel('x')
ax.set_ylabel('y')
fig.colorbar(surf, shrink=0.5, aspect=5) # add color bar indicating the PDF
ax.view_init(60, 35)
plt.show()
```

5. Discussion

This project is fitting only with a preliminary study of a demonstrator for the computer application targeted.
As next steps, it would be needed:

- To enlarge the datasets;
- To potentially take into account additional parameters, besides waiting time and total bill: even if time and money are key parameters, another one could be highlighted;
- To test other algorithms, to go still further in fine tuning them.

Promotion actions near restaurant owners are also very important in order to find the funds to continue studies, tests and finalization of this computer application

6. Searching for the most frequented restaurants _ Foursquare (1/5)

Problem, Problem Solving and tools used:

- Search for the relative coordinates of the 20 districts in Paris (package used: Geopy);
- Searching for the most famous venues in Paris (application used: Foursquare);
- Selecting of the top 5 restaurants listed in each district;
- Next step would be to propose a demonstration of the machine learning application, to the restaurant owners.

6. Searching for the most frequented restaurants _ Foursquare (2/5)

Step 1. Search for the relative coordinates of the 20 districts in Paris (package used: Geopy)

In [16]:
```
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

def get_latitude_longitude(address=""):
    if not address:
        return None, None

    geolocator = Nominatim(user_agent="paris_explorer")
    location = geolocator.geocode(address)
    latitude = location.latitude
    longitude = location.longitude
    return (latitude, longitude)

def get_latitude_longitude_paris_fr():
    address = 'Paris, FR'
    return get_latitude_longitude(address)

latitude, longitude = get_latitude_longitude_paris_fr()
print('The geograpical coordinate of Paris, FR are {}, {}.'.format(latitude, longitude))
```

```
The geograpical coordinate of Paris, FR are 48.8566969, 2.3514616.
```

address	latitude	longitude
75001, FR	48.863554	2.338856
75002, FR	48.867418	2.344256
75003, FR	48.862607	2.360211
75004, FR	48.856004	2.357028
75005, FR	48.852752	2.346343
75006, FR	48.853537	2.343370
75007, FR	48.855913	2.313839
75008, FR	48.872385	2.312707
75009, FR	48.877355	2.336856
75010, FR	48.879201	2.354391
75011, FR	48.855630	2.370806
75012, FR	48.839734	2.380054
75013, FR	48.826997	2.353396
75014, FR	48.828590	2.307541
75015, FR	48.838461	2.315728
75016, FR	48.855031	2.273958
75017, FR	48.883508	2.304923
75018, FR	48.893074	2.343881
75019, FR	48.878076	2.376198
75020, FR	48.857126	2.409257

6. Searching for the most frequented restaurants _ Foursquare (3/5)

Step 2. Request of top of the venues in Paris districts: 'Get request: requests.get(url)' and study of the categories

```
LIMIT = 150 # limit of number of venues returned by Foursquare API
radius = 750 # define radius

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

url # display URL
```

```
import requests # library to handle requests

results = requests.get(url).json()
results
```

```
# function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

6. Searching for the most frequented restaurants _ Foursquare (4/5)

Step 3. Conversion JSON file into pandas dataframe

```
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe

venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
nearby_venues =nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues.head()
```

	name	categories	lat	lng
0	Jardin du Palais Royal	Garden	48.864941	2.337728
1	Palais Royal	Historic Site	48.863236	2.337127
2	Comédie-Française	Theater	48.863088	2.336612
3	Place du Palais Royal	Plaza	48.862523	2.336688
4	Christian Louboutin	Shoe Store	48.862697	2.340757

6. Searching for the most frequented restaurants _ Foursquare (5/5)

Step 4. List of the most frequented venues in each district in Paris, in another dataframe.

Paris district id	Top 1	Top 2	Top 3	Top 4	Top 5
1	French Restaurant	Hotel	Café	Plaza	Japanese Restaurant
2	French Restaurant	Bakery	Cocktail Bar	Wine Bar	Bistro
3	French Restaurant	Coffee Shop	Burger Joint	Café	Bistro
4	French Restaurant	Pastry Shop	Hotel	Wine Bar	Gourmet Shop
5	French Restaurant	Indie Movie Theater	Hotel	Café	Bookstore
6	French Restaurant	Bookstore	Café	Creperie	Hotel
7	French Restaurant	Hotel	Café	Plaza	Italian Restaurant
8	French Restaurant	Hotel	Bakery	Thai Restaurant	Theater
9	French Restaurant	Hotel	Bakery	Cocktail Bar	Bistro
10	French Restaurant	Hotel	Indian Restaurant	Restaurant	Japanese Restaurant
11	French Restaurant	Bar	Pizza Place	Italian Restaurant	Bistro
12	French Restaurant	Hotel	Beer Bar	Coffee Shop	Restaurant
13	French Restaurant	Vietnamese Restaurant	Hotel	Bar	Bakery
14	Café	Bakery	Supermarket	Hotel	Grocery Store
15	Hotel	French Restaurant	Pizza Place	Dessert Shop	Coffee Shop
16	French Restaurant	Bakery	Italian Restaurant	Japanese Restaurant	Plaza
17	Hotel	French Restaurant	Italian Restaurant	Bar	Bistro
18	French Restaurant	Bar	Hotel	Café	Pizza Place
19	French Restaurant	Bar	Park	Café	Restaurant
20	Hotel	Supermarket	Tram Station	French Restaurant	Pizza Place

Next step: how to use this result ?

- Visit the districts in which there are more restaurants and more frequented;
- Save time avoiding the districts such as 14 and 20;
- Locate the restaurants in the top 5 or top 10;
- Visit them to promote the computer application demonstrator, newly designed to help restaurant owners to make customers loyal

7. Conclusion

Key points:

Machine learning implementation ROI highly depends on efforts put in all the steps:

- Data collecting (need to collect data from all possible sources, concerns about data volume, privacy, …);
- Data cleaning (data missing, inconsistency, outliers, …);
- Machine Learning algorithm choice and implementation.

So far, human expertise is needed, whatever the analysis: our intelligence is not reproduced yet by the computers.

About 80% of a machine learning project deal with data collecting and cleaning.

Link to project code:
https://gist.github.com/corinne3/7443030b58f0429f628e4fb7070a2408

'To write it, it took three months; to conceive it three minutes; to collect the data in it, all my life.'
_ F. Scott Fitzgerald in 'This Side of Paradise'

8. References (1/3)

Background and problem:

'How To Obtain Customer Feedback For Your Restaurant Using Different Methods':
https://www.posist.com/restaurant-times/restro-gyaan/customers-want-conduct-surveys-collect-customer-feedback-restaurant.html

Guest blog: rebuilding the foundations of customer support in the new world of software as a service
https://hyken.com/customer-care/guest-blog-rebuilding-foundations-customer-support-new-world-software-service/

Data collection:

'How long is too long to wait for a table at a restaurant ?':
https://www.phoenixnewtimes.com/restaurants/how-long-is-too-long-to-wait-for-a-table-at-a-restaurant-6509473

French and Americans are poles apart... when it comes to time spent eating:
https://www.thelocal.fr/20180313/french-spend-twice-as-long-eating-and-drinking-as-americans

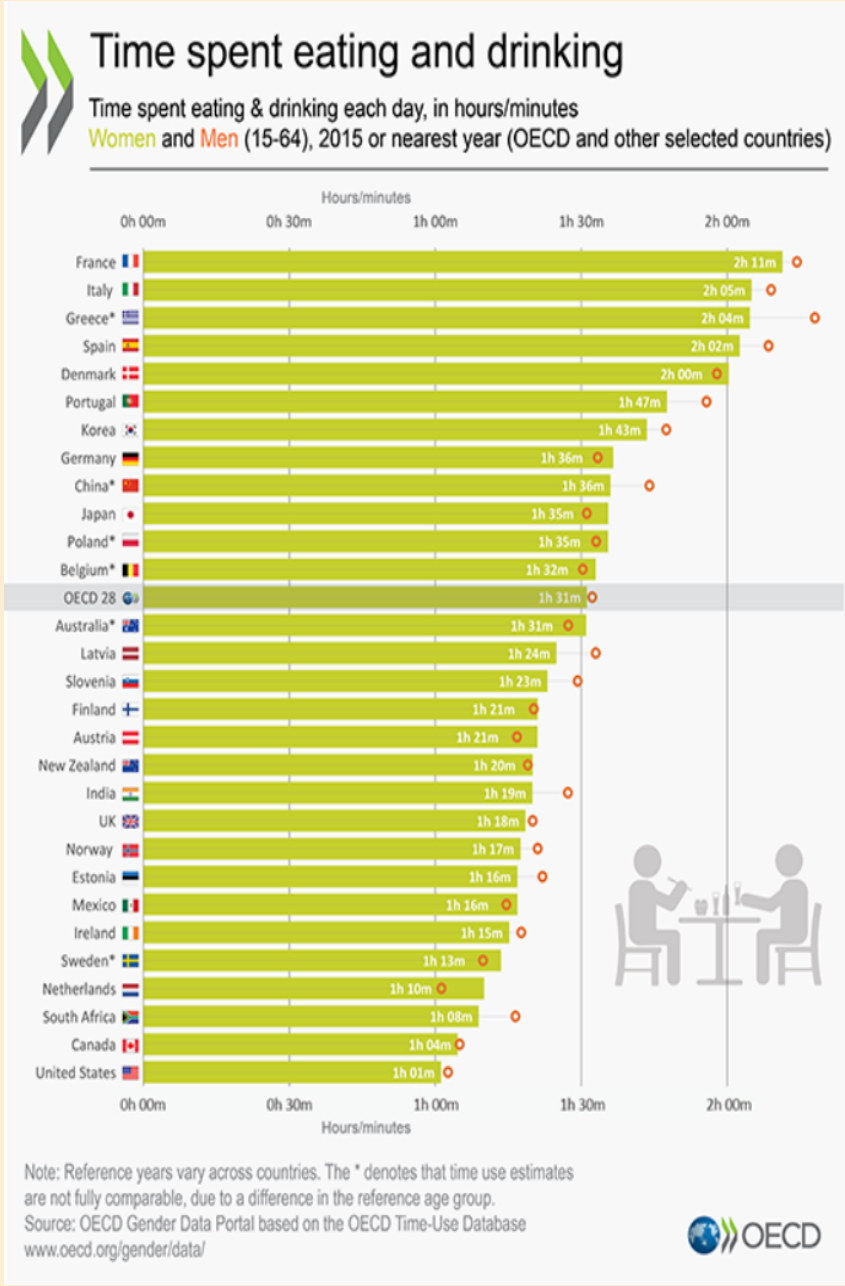
OECD Gender Data Portal:
https://www.oecd.org/gender/balancing-paid-work-unpaid-work-and-leisure.htm

Cost of Living in France:
https://www.numbeo.com/cost-of-living/country_result.jsp?country=France

8. References (2/3)

Plot from OECD site:

OECD Gender Data Portal:
https://www.oecd.org/gender/balancing-paid-work-unpaid-work-and-leisure.htm

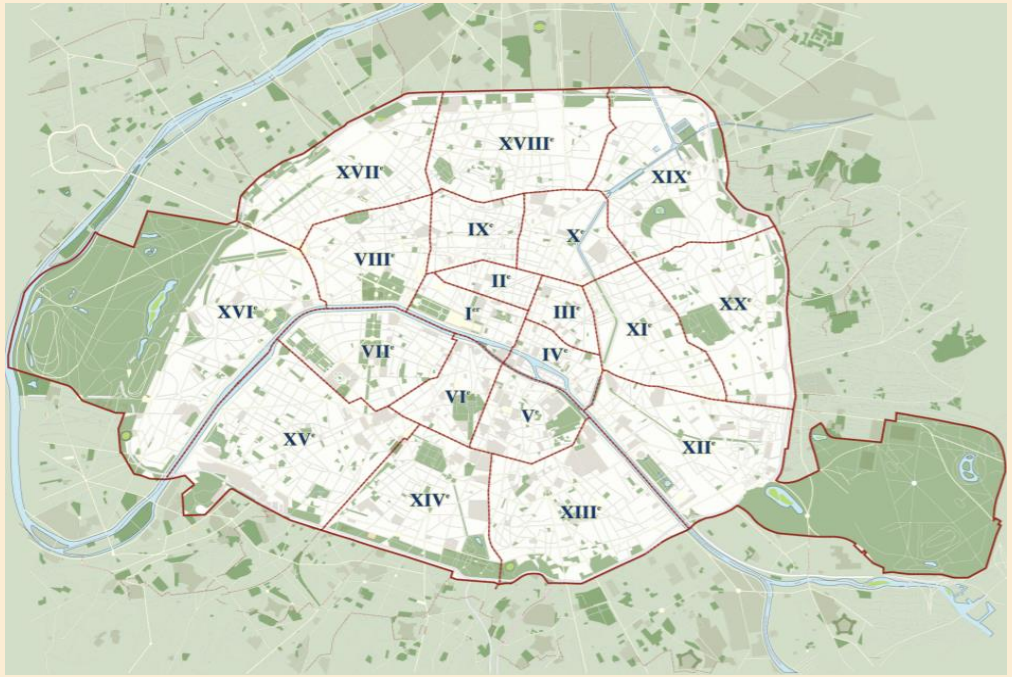


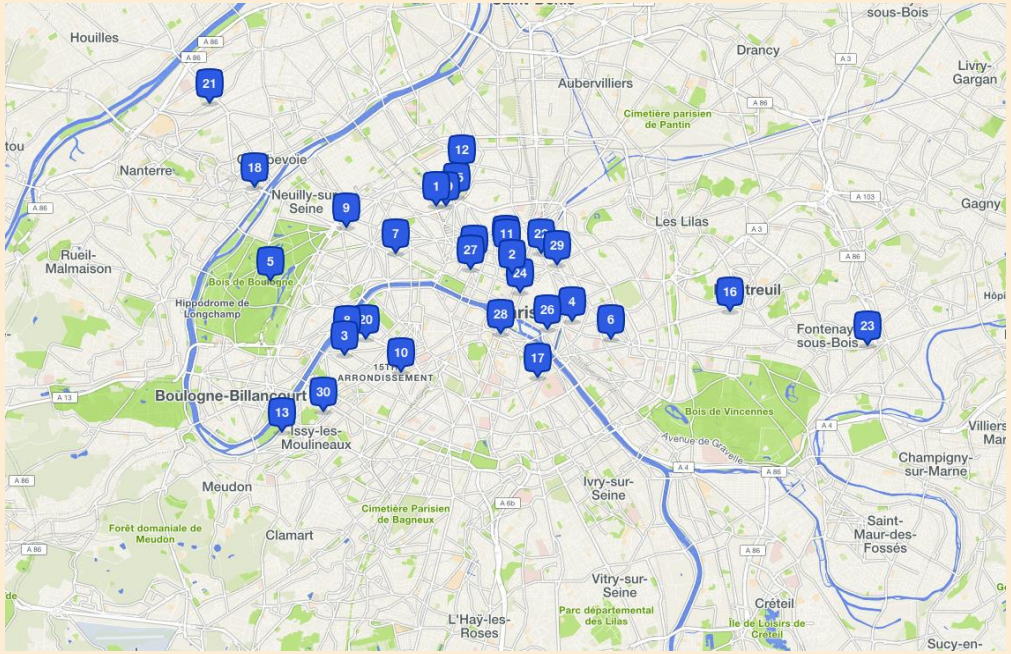
8. References (3/3)

List of Paris districts and population density:
https://fr.wikipedia.org/wiki/Arrondissements_de_Paris

Paris maps:
https://parismap360.com/paris-arrondissement-map#.XfVpqtEo91l

Paris density population:
https://www.insee.fr/fr/statistiques?collection=119





THANK YOU!

Capstone Project _ Week 2 _ 03/29/2020