Open restaurant using the business density report

Sometimes, we will confuse where to open a restaurant while we are rich enough. So, can we find the way to do the assessments? Of course, yes. So, let's do it.

We will take **"Shanghai"** and **"Tokyo"** as our samples. Here is the information from wiki.

Shanghai is one of the four municipalities under the direct administration of the central government of China, the largest city in China by population, and the second most populous city proper in the world, with a population of more than 24 million as of 2017. It is a global financial centre and transport hub, with the world's busiest container port. Located in the Yangtze River Delta, it sits on the south edge of the estuary of the Yangtze in the middle portion of the East China coast. The municipality borders the provinces of Jiangsu and Zhejiang to the north, south and west, and is bounded to the east by the East China Sea.

Tokyo, officially **Tokyo Metropolis**, one of the 47 prefectures of Japan, has served as the Japanese capital since 1869.As of 2014 the Greater Tokyo Area ranked as the most populous metropolitan area in the world. The urban area houses the seat of the Emperor of Japan, of the Japanese government and of the National Diet. Tokyo forms part of the Kantō region on the southeastern side of Japan's main island, Honshu, and includes the Izu Islands and Ogasawara Islands.Tokyo was formerly named Edo when Shōgun Tokugawa Ieyasu made the city as his headquarters in 1603. It became the capital after Emperor Meiji moved his seat to the city from Kyoto in 1868; at that time Edo was renamed Tokyo. Tokyo Metropolis formed in 1943 from the merger of the former Tokyo Prefecture and the city of Tokyo.

So, next, we will find the useful data of the two cities.

We need to use **google map** to find the city's location data

We need to use the **foursquare** venues data to make samples

We need to use **DBSCAN** algorithm to make labels from the matrix sample which from foursquare venues data.

We need to use the **folium map** data to do the visualization

And we use **Python** as our program language.

Here we go.

Import all the libraries and functions

```
# -*- coding: utf-8 -*-
import os,sys
import urllib
import requests
import json
from urllib.request import urlopen
import pandas as pd
import numpy as np
import matplotlib.cm as cm
import matplotlib.colors as colors
from bs4 import BeautifulSoup
from geopy.geocoders import Nominatim
from sklearn.cluster import KMeans
from sklearn.cluster import DBSCAN
import folium
import math
```

```
# get the location of the city
def getlocation(address):
    geolocator = Nominatim()
    location = geolocator.geocode(address)
    latitude = 0.0
    latitude = location.latitude
    longitude =0.0
    longitude = location.longitude
    print('The geograpical coordinate of ',address,'are {}, {}.'.format(latitude, longitude))
    return latitude,longitude
```

```
# get the location of the city
def getlocation(address):
    geolocator = Nominatim()
    location = geolocator.geocode(address)
    latitude = 0.0
    latitude = location.latitude
    longitude =0.0
    longitude = location.longitude
    print('The geograpical coordinate of ',address,'are {}, {}.'.format(latitude, longitude))
    return latitude,longitude
```

```
    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Location','Venue','Latitude','Longitude','Category']
    return(nearby_venues)
```

```
# create cluster map
def clusterMap(kclusters,dfs):
    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(dfs['Latitude'], dfs['Longitude'], dfs['Venue'], dfs['Cluster Labels']):
        label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
        folium.CircleMarker(
            [lat, lon],
            radius=5,
            popup=label,
            color=rainbow[cluster-1],
            fill=True,
            fill_color=rainbow[cluster-1],
            fill_opacity=0.7).add_to(map_clusters)
```

```
# get the eps value of the DBSCAN algorithm
def epsilon(data, MinPts):
    m, n = np.shape(data)
    xMax = np.max(data, 0)
    xMin = np.min(data, 0)
    eps = ((np.prod(xMax - xMin) * MinPts * math.gamma(0.5 * n + 1)) / (m * math.sqrt(math.pi ** n))) ** (1.0 / n)
    return eps
```

```
# color the place in Map
def mapMarkers(map_name,dfs_data):  # add markers to map
    for lat, lng, label in zip(dfs_data['Latitude'], dfs_data['Longitude'], dfs_data['Venue']):
        label = folium.Popup(label, parse_html=True)
        folium.CircleMarker(
            [lat, lng],
            radius=5,
            popup=label,
            color='blue',
            fill=True,
            fill_color='#3186cc',
            fill_opacity=0.7).add_to(map_name)
```

Get the location of Shanghai and Tokyo

```
address_sh = 'Shanghai,CN'
address_tk = 'Tokyo,JP'
#location_sh = getlocation(address_sh)
latitude_sh = 31.2253441
longitude_sh = 121.4888922
#location_tk = getlocation(address_tk)
latitude_tk = 35.6828387
longitude_tk = 139.7594549
print('The geograpical coordinate of ',address_sh,'are {} {} .'.format(latitude_sh,longitude_sh))
print('The geograpical coordinate of ',address_tk,'are {} {}.'.format(latitude_tk,longitude_sh))
```

```
The geograpical coordinate of  Shanghai,CN are 31.2253441 121.4888922 .
The geograpical coordinate of  Tokyo,JP are 35.6828387 121.4888922.
```

Make the map to display the two cities

```
map_all = folium.Map(location=[(latitude_sh+latitude_tk)/2, (longitude_sh+longitude_tk)/2], tiles='Stamen Terrain',zoom_start=5)

folium.Marker(location=[latitude_sh, longitude_sh], popup='Shanghai City').add_to(map_all)
folium.CircleMarker(location=[latitude_sh, longitude_sh], radius=10,
popup='Shanghai City', color='#3186cc',fill_color='#3186cc').add_to(map_all)

folium.Marker(location=[latitude_tk, longitude_tk], popup='Tokyo City').add_to(map_all)
folium.CircleMarker(location=[latitude_tk, longitude_tk], radius=10,
popup='Shanghai City', color='#3186cc',fill_color='#3186cc').add_to(map_all)
map_all
```

Define the ID of the foursquare API

```
CLIENT_ID = '3CYZT4F5XGUGM1JY1XP0B4XK0XTA2BPMXT03QR[redacted]' # your Foursquare ID
CLIENT_SECRET = 'C24OJ1Z0EEKAVUR005PSUFIFEGJKEKTSAC[redacted]' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version
QUERY = 'food'
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: 3CYZT4F5XGUGM1JY1XP0B4XK0XTA2BPMXT03QR[redacted]
CLIENT_SECRET:C24OJ1Z0EEKAVUR005PSUFIFEGJKEKTSAC[redacted]
```

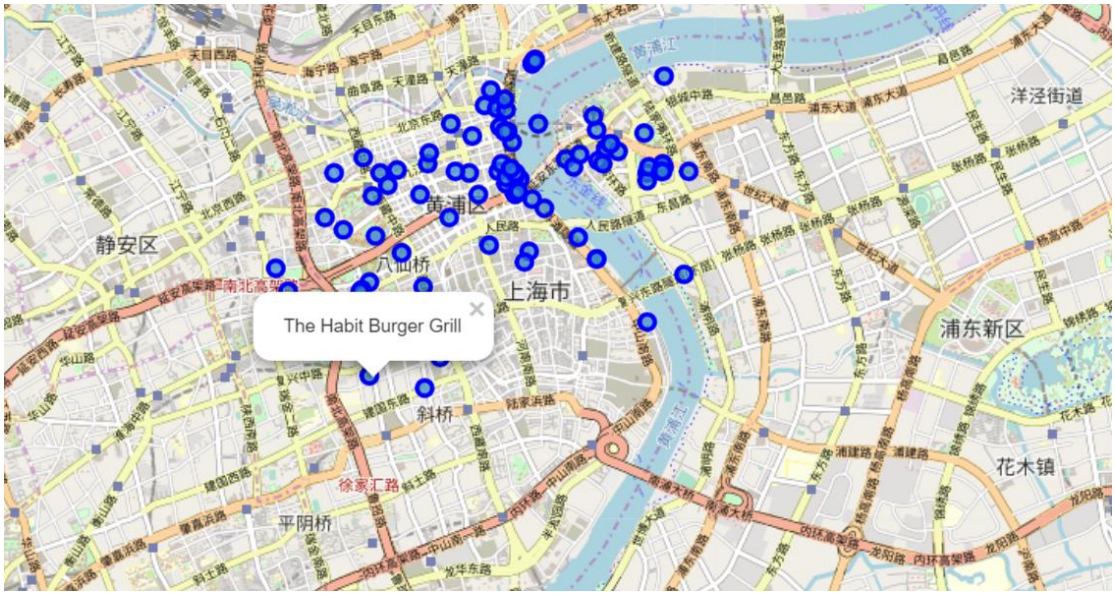
Get the venues of shanghai

```
dfs_sh = getNearbyVenues_new('Shanghai',latitude_sh,longitude_sh)
dfs_sh.head()
```

	Location	Venue	Latitude	Longitude	Category
0	Shanghai	Yu Garden (豫园)	31.228922	121.487982	Garden
1	Shanghai	CHAR Bar	31.228209	121.495593	Hotel Bar
2	Shanghai	Hotel Indigo Shanghai On The Bund (上海外灘英迪格酒店)	31.228193	121.495571	Hotel
3	Shanghai	City of God Temple (城隍庙)	31.227859	121.487536	Temple
4	Shanghai	Goodfellas	31.234878	121.486730	Italian Restaurant

Create the map of Shanghai with venues

```
# create map using latitude and longitude values
map_sh = folium.Map(location=[latitude_sh, longitude_sh], zoom_start=13)
mapMarkers(map_sh,dfs_sh)
map_sh
```



Get the location of each venues, then the dataframe as the sample matrix

```
X_dfs_sh=dfs_sh.drop(['Venue','Location','Category'],axis=1)
X_dfs_sh.head()
```

	Latitude	Longitude
0	31.228922	121.487982
1	31.228209	121.495593
2	31.228193	121.495571
3	31.227859	121.487536
4	31.234878	121.486730

Using the DBSCAN module to cluster

```
# Create the DBSCAN module
eps_temp = epsilon(X_dfs_sh,5)
ydbscan_sh = DBSCAN(eps=eps_temp,min_samples=5).fit(X_dfs_sh)
dfs_sh['Cluster Labels'] = ydbscan_sh.labels_
ydbscan_sh
ydbscan_sh.labels_
```

```
array([ 0, -1, -1, -1,  0,  0,  0,  0,  0, -1,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0, -1,  0,  0,  0,  0, -1,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, -1,  0,  0,  0,
        0,  0,  0,  0, -1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, -1, -1,  0, -1],
      dtype=int64)
```

Check the labels

```
dfs_sh.groupby('Cluster Labels').size().sort_values()
```

```
Cluster Labels
-1    11
 0    89
dtype: int64
```

```
business_density_of_shanghai = (89/100)
business_density_of_shanghai
```

```
0.89
```

We can get the business density of Shanghai is 0.89.

Marked in the map

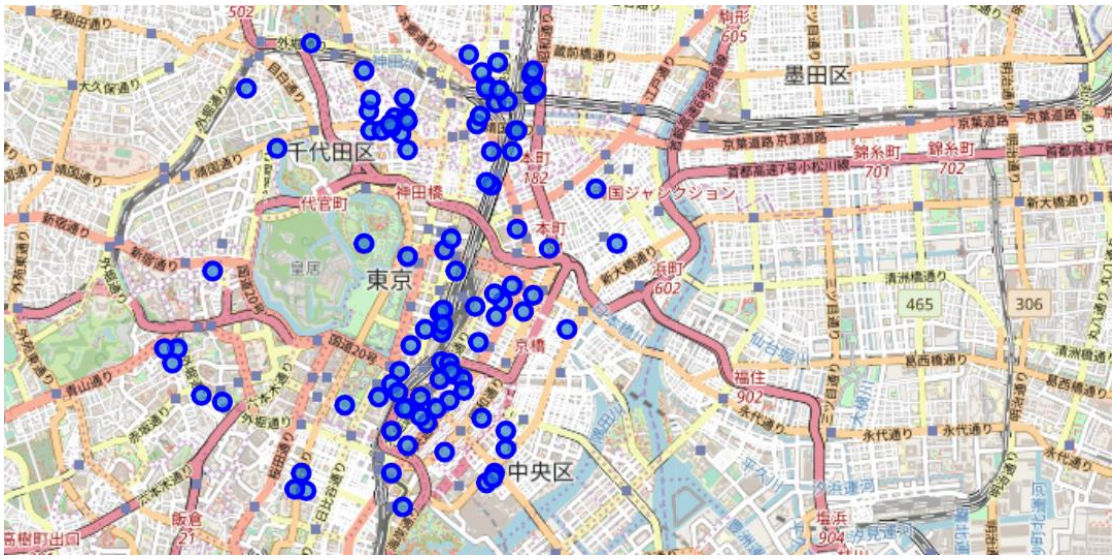
```
map_clusters = folium.Map(location=[latitude_sh, longitude_sh], zoom_start=13)
# set color scheme for the clusters
clusterMap(2,dfs_sh)
map_clusters
```

Repeat the actions for Tokyo city.

```
dfs_tk = getNearbyVenues_new('Tokyo', latitude_tk, longitude_tk)
dfs_tk.head()
```

	Location	Venue	Latitude	Longitude	Category
0	Tokyo	Palace Hotel Tokyo (パレスホテル東京)	35.684644	139.761302	Hotel
1	Tokyo	Imperial Palace East Garden (皇居東御苑)	35.685797	139.756662	Garden
2	Tokyo	KITTE Garden (屋上庭園 KITTEガーデン)	35.679806	139.764872	Garden
3	Tokyo	Aman Tokyo (アマン東京)	35.685236	139.765401	Hotel
4	Tokyo	Mitsubishi Ichigokan Museum (三菱一号館美術館)	35.678420	139.763260	Art Museum

```
# create map of Scarborough using latitude and longitude values
map_tk = folium.Map(location=[latitude_tk, longitude_tk], zoom_start=13)
mapMarkers(map_tk, dfs_tk)
map_tk
```



```
X_dfs_tk=dfs_tk.drop(['Venue','Location','Category'],axis=1)
X_dfs_tk.head()
```

	Latitude	Longitude
0	35.684644	139.761302
1	35.685797	139.756662
2	35.679806	139.764872
3	35.685236	139.765401
4	35.678420	139.763260

```
eps_temp = epsilon(X_dfs_tk,5)
ydbscan_tk = DBSCAN(eps=eps_temp,min_samples=5).fit(X_dfs_tk)
dfs_tk['Cluster Labels'] = ydbscan_tk.labels_
ydbscan_tk
ydbscan_tk.labels_
```

```
array([ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,
        0,  0,  1,  1,  0,  0,  0,  0,  0,  1,  0,  0,  1,  1,  1,  1,  0,
        1,  0,  0,  1,  0,  0,  0,  1,  0, -1, -1,  0,  0,  1,  0,  0,  0,
        0,  0,  0,  1,  0,  0, -1,  0, -1, -1,  0, -1, -1,  0,  0,  0,  0,
        0, -1, -1,  0, -1,  0, -1,  0,  0, -1, -1,  0,  0,  0, -1],
      dtype=int64)
```

```
dfs_tk.groupby('Cluster Labels').size().sort_values()
```

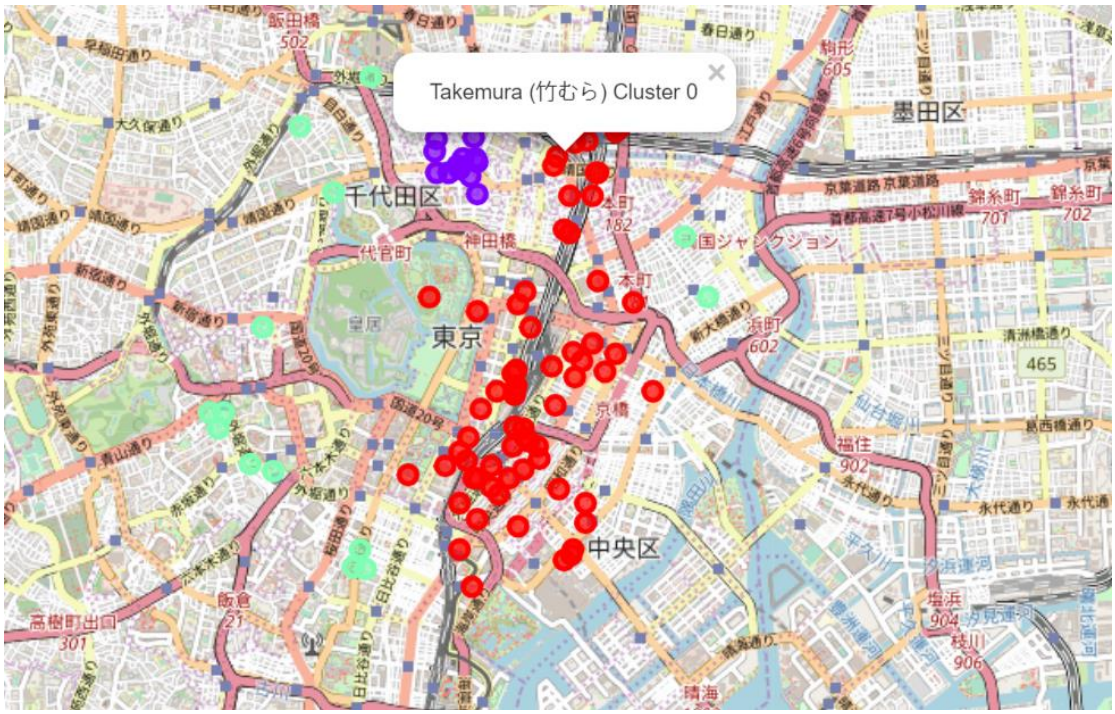
```
Cluster Labels
 1     13
-1     14
 0     73
dtype: int64
```

```
business_density_of_tokyo = ((100-14)/100)
business_density_of_tokyo
```

```
0.86
```

We can get the business density of Tokyo is 0.86.

```
map_clusters = folium.Map(location=[latitude_tk, longitude_tk], zoom_start=13)
# set color scheme for the clusters
clusterMap(3,dfs_tk)
map_clusters
```



Result:

From the labels, we can know the Shanghai's business density is 0.89 > Tokyo's business density 0.86. So, I will choose Shanghai to open a restaurant from the business density data.

Discussion

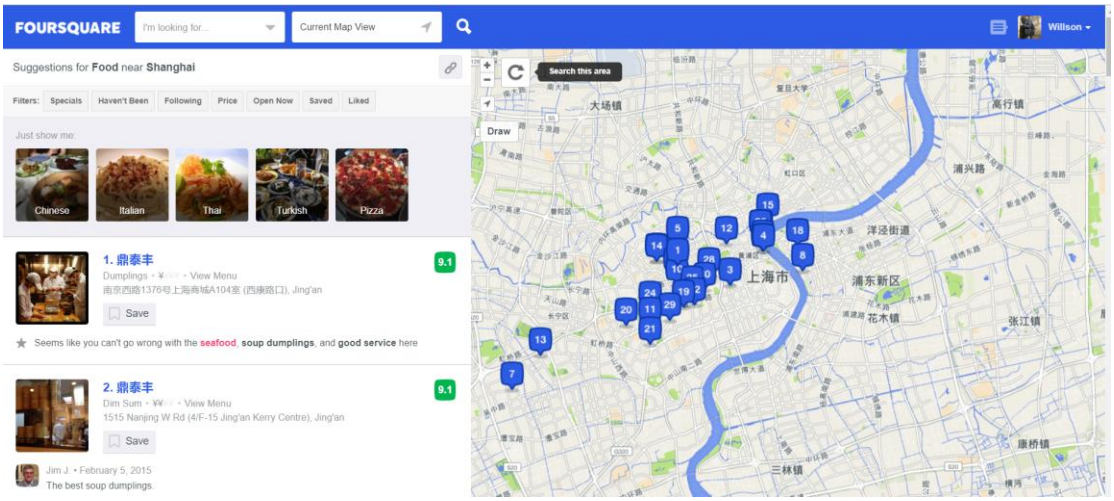
But business density is only one point of opening restaurant, we should consult more points such as population density, consumption index and so on. Because I can't get the data, so I only use business density to describe the problem.

Conclusion

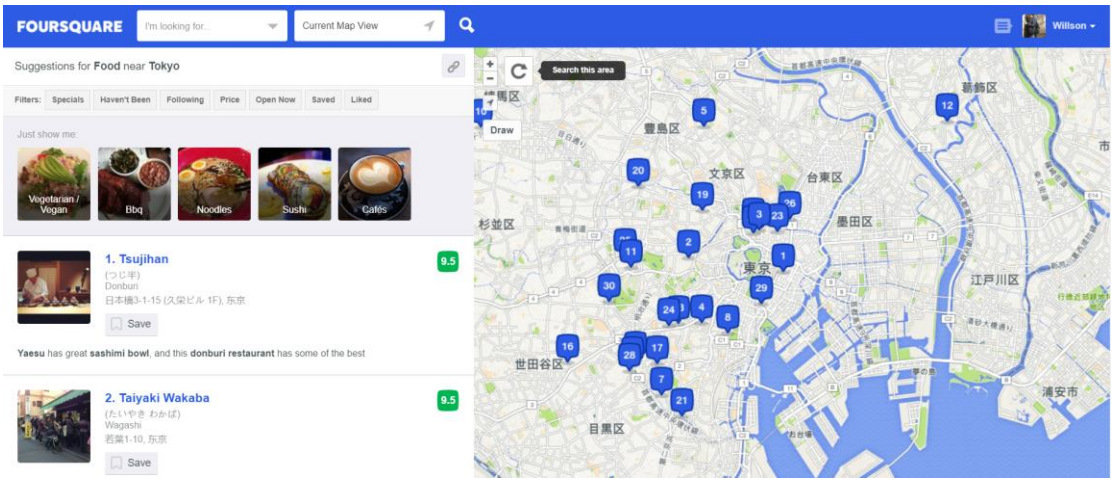
So, if base on the foursquare venues data, I will use business density to find the city to open a restaurant. So, I will open the restaurant in Shanghai City of China.

Remark:

1. I have put my whole scripts to git hub, here is the linkage: https://github.com/willsonluo/Coursera_Capstone/blob/master/Business%20Density%20Analyse/Business%20Density%20of%20Shanghai%20and%20Tokyo.ipynb
2. Shanghai information from wiki linkage: https://en.wikipedia.org/wiki/Shanghai
3. Tokyo information from wiki linkage: https://en.wikipedia.org/wiki/Tokyo
4. DBSCAN algorithm from wiki linkage: https://en.wikipedia.org/wiki/DBSCAN
5. Folium Map from GitHub linkage: https://github.com/python-visualization/folium
6. Shanghai food information from foursquare linkage: https://foursquare.com/explore?cat=food&mode=url&near=Shanghai%2C%20Shanghai%20Shi%2C%20China&nearGeoId=72057594039724172



7. Shanghai food information from foursquare linkage: https://foursquare.com/explore?cat=food&mode=url&near=Tokyo%2C%20T%C5%8Dky%C5%8D%2C%20Japan&nearGeoId=72057594039778083



Hope you will like it. Thank you.

--Willson Luo(罗崇锦) from China.
--2018/11/18