LastCapstoneProject

February 22, 2021

1 Final Capstone Project: Top of the cities

This notebook show how I using capstone project to help my friend to find a suitable place to open up a hotel.

1.1 1. Business problem

My colleage wants to open up a hotel in populated city in Vietnam. But he lacks of information about the current situation of existing hotels to detect where the best place to build a hotel is. This project aims at finding an appropriate city to open it.This project can be used by anyone who wishes to open a hotel or any other retail business in Vietnam.

1.2 2. Data requirements

The data needed for this project would be the name of the cities in Vietnam and their respective latitudes and longitudes, also the name, id , latitude and longitude of the respective venues in and around the cities of Vietnam. Such data are provided freely from this website https://simplemaps.com/data/vn-cities, where the data is available in the form of a csv file and also the name, id , latitude and longitude of the respective venues.

1.2.1 Packages that need to be installed for working the data

- Geopy
- Geocoder
- Folium

Libraries needed for this project

- numpy
- pandas
- random
- requests
- matplotlib
- KMeans
- folium
- json_normalize
- Nominatim

First we must install necessary libraries:

[1]:
```
!pip install geopy
```

```
Collecting geopy
  Downloading geopy-2.1.0-py3-none-any.whl (112 kB)
Collecting geographiclib<2,>=1.49
  Downloading geographiclib-1.50-py3-none-any.whl (38 kB)
Installing collected packages: geographiclib, geopy
Successfully installed geographiclib-1.50 geopy-2.1.0
```

[2]:
```
! pip install geocoder
```

```
Collecting geocoder
  Downloading geocoder-1.38.1-py2.py3-none-any.whl (98 kB)
Requirement already satisfied: future in c:\programdata\anaconda3\lib\site-
packages (from geocoder) (0.18.2)
Requirement already satisfied: click in c:\programdata\anaconda3\lib\site-
packages (from geocoder) (7.1.2)
Collecting ratelim
  Downloading ratelim-0.1.6-py2.py3-none-any.whl (4.0 kB)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-packages
(from geocoder) (1.15.0)
Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-
packages (from geocoder) (2.24.0)
Requirement already satisfied: decorator in c:\programdata\anaconda3\lib\site-
packages (from ratelim->geocoder) (4.4.2)
Requirement already satisfied: certifi>=2017.4.17 in
c:\programdata\anaconda3\lib\site-packages (from requests->geocoder) (2020.6.20)
Requirement already satisfied: chardet<4,>=3.0.2 in
c:\programdata\anaconda3\lib\site-packages (from requests->geocoder) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in
c:\programdata\anaconda3\lib\site-packages (from requests->geocoder) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\programdata\anaconda3\lib\site-packages (from requests->geocoder) (1.25.11)
Installing collected packages: ratelim, geocoder
Successfully installed geocoder-1.38.1 ratelim-0.1.6
```

[3]:
```
!pip install folium
```

```
Requirement already satisfied: folium in c:\programdata\anaconda3\lib\site-
packages (0.12.1)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-
packages (from folium) (1.19.2)
Requirement already satisfied: jinja2>=2.9 in c:\programdata\anaconda3\lib\site-
packages (from folium) (2.11.2)
Requirement already satisfied: branca>=0.3.0 in
c:\programdata\anaconda3\lib\site-packages (from folium) (0.4.2)
Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-
packages (from folium) (2.24.0)
Requirement already satisfied: MarkupSafe>=0.23 in
```

```
c:\programdata\anaconda3\lib\site-packages (from jinja2>=2.9->folium) (1.1.1)
Requirement already satisfied: chardet<4,>=3.0.2 in
c:\programdata\anaconda3\lib\site-packages (from requests->folium) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
c:\programdata\anaconda3\lib\site-packages (from requests->folium) (2020.6.20)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\programdata\anaconda3\lib\site-packages (from requests->folium) (1.25.11)
Requirement already satisfied: idna<3,>=2.5 in
c:\programdata\anaconda3\lib\site-packages (from requests->folium) (2.10)
```

Import necessary libraries:

[4]:
```
import pandas as pd
import numpy as np
import random
import requests
from pandas.io.json import json_normalize
import matplotlib.cm as cm
import matplotlib.colors as colors
from sklearn.cluster import KMeans
import folium
from geopy.geocoders import Nominatim
```

Create dataframe of cities of Vietnam:

[24]:
```
data_file= 'D://vn.csv'
df=pd.read_csv(data_file, sep=',')
df.head()
```

[24]:

	city	lat	lng	country	iso2	admin_name	capital	population	population_proper
0	Ho Chi Minh City	10.8167	106.6333	Vietnam	VN	Hồ Chí Minh	admin	13312000.0	7431000.0
1	Hanoi	21.0245	105.8412	Vietnam	VN	Hà Nội	primary	7785000.0	7785000.0
2	Haiphong	20.8000	106.6667	Vietnam	VN	Hải Phòng	admin	2103500.0	2103500.0
3	Cần Thơ	10.0333	105.7833	Vietnam	VN	Cần Thơ	admin	1237300.0	1237300.0
4	Biên Hòa	10.9575	106.8426	Vietnam	VN	Đồng Nai	admin	1104000.0	1104000.0

[28]:
```
df.shape
```

[28]:
```
(64, 9)
```

1.3 3. Data pre-processing

Drop all row in the dataframe which have Nan value in population:

[27]:
```
df.dropna(subset=["population"], axis=0, inplace=True)
```

[29]:
```
df.rename(columns = {'lat':'Latitude','lng':'Longitude','admin_name':
 ↪'Region'},inplace=True)
df.head()
```

[29]:
```
                city  Latitude  Longitude  country iso2       Region  capital  \
0  Ho Chi Minh City   10.8167   106.6333  Vietnam   VN  Hồ Chí Minh    admin
1             Hanoi   21.0245   105.8412  Vietnam   VN       Hà Nội  primary
2          Haiphong   20.8000   106.6667  Vietnam   VN    Hải Phòng    admin
3           Cần Thơ   10.0333   105.7833  Vietnam   VN      Cần Thơ    admin
4          Biên Hòa   10.9575   106.8426  Vietnam   VN     Đồng Nai    admin

   population  population_proper
0  13312000.0          7431000.0
1   7785000.0          7785000.0
2   2103500.0          2103500.0
3   1237300.0          1237300.0
4   1104000.0          1104000.0
```

Removing cities having population less than 10000:

[30]:
```
length = df.shape[0]
for i in range(0,length):
    if (df['population'][i]<10000):
        df.drop([i],axis=0, inplace= True)
df.shape
```

[30]:
```
(64, 9)
```

No cities removed from this step!

Showing the capital of Vietnam and the capital of its respective region

[31]:
```
tempdf = df
tempdf = tempdf.groupby('capital')
tempdf.get_group('primary')
```

[31]:
```
    city  Latitude  Longitude  country iso2  Region  capital  population  \
1  Hanoi   21.0245   105.8412  Vietnam   VN  Hà Nội  primary   7785000.0

   population_proper
1          7785000.0
```

Showing capitals of the respectve regions of Vietnam

[32]:
```
tempdf.get_group('admin')
```

[32]:

	city	Latitude	Longitude	country	iso2	Region
0	Ho Chi Minh City	10.8167	106.6333	Vietnam	VN	Hồ Chí Minh
2	Haiphong	20.8000	106.6667	Vietnam	VN	Hải Phòng
3	Cần Thơ	10.0333	105.7833	Vietnam	VN	Cần Thơ
4	Biên Hòa	10.9575	106.8426	Vietnam	VN	Đồng Nai
6	Bắc Ninh	21.1861	106.0763	Vietnam	VN	Bắc Ninh
7	Hải Dương	20.9411	106.3331	Vietnam	VN	Hải Dương
8	Vinh	18.6733	105.6922	Vietnam	VN	Nghệ An
9	Huế	16.4667	107.5833	Vietnam	VN	Thừa Thiên-Huế
10	Thanh Hóa	19.8075	105.7764	Vietnam	VN	Thanh Hóa
11	Nha Trang	12.2500	109.1833	Vietnam	VN	Khánh Hòa
12	Nam Định	20.4200	106.1683	Vietnam	VN	Nam Định
13	Buôn Ma Thuột	12.6667	108.0500	Vietnam	VN	Đắk Lắk
14	Thái Nguyên	21.6000	105.8500	Vietnam	VN	Thái Nguyên
15	Vũng Tàu	10.4042	107.1417	Vietnam	VN	Bà Rịa-Vũng Tàu
16	Cà Mau	9.1833	105.1500	Vietnam	VN	Cà Mau
17	Quy Nhơn	13.7765	109.2237	Vietnam	VN	Bình Định
18	Sóc Trăng	9.6000	105.9719	Vietnam	VN	Sóc Trăng
19	Long Xuyên	10.3686	105.4234	Vietnam	VN	An Giang
20	Việt Trì	21.3228	105.4019	Vietnam	VN	Phú Thọ
21	Thái Bình	20.4461	106.3422	Vietnam	VN	Thái Bình
22	Quảng Ngãi	15.1206	108.7922	Vietnam	VN	Quảng Ngãi
24	Rạch Giá	10.0125	105.0808	Vietnam	VN	Kiến Giang
25	Thủ Dầu Một	11.0042	106.6583	Vietnam	VN	Bình Dương
26	Tuy Hòa	13.0875	109.3106	Vietnam	VN	Phú Yên
27	Bạc Liêu	9.2833	105.7167	Vietnam	VN	Bạc Liêu
29	Phan Thiết	10.9375	108.1583	Vietnam	VN	Bình Thuận
31	Phan Rang-Tháp Chàm	11.5643	108.9886	Vietnam	VN	Ninh Thuận
32	Hạ Long	20.9500	107.0833	Vietnam	VN	Quảng Ninh
33	Hà Tĩnh	18.3428	105.9058	Vietnam	VN	Hà Tĩnh
34	Đồng Hới	17.4833	106.6000	Vietnam	VN	Quảng Bình
37	Cao Lãnh	10.4603	105.6331	Vietnam	VN	Đồng Tháp
38	Lạng Sơn	21.8478	106.7578	Vietnam	VN	Lạng Sơn
40	Pleiku	13.9833	108.0000	Vietnam	VN	Gia Lai
41	Tân An	10.5322	106.4042	Vietnam	VN	Long An
42	Trà Vinh	9.9369	106.3411	Vietnam	VN	Trà Vinh
43	Ninh Bình	20.2539	105.9750	Vietnam	VN	Ninh Bình
44	Tây Ninh	11.3131	106.0963	Vietnam	VN	Tây Ninh
46	Mỹ Tho	10.3500	106.3500	Vietnam	VN	Tiền Giang
48	Hòa Bình	20.8172	105.3375	Vietnam	VN	Hòa Bình
49	Vĩnh Long	10.2550	105.9753	Vietnam	VN	Vĩnh Long
50	Vị Thanh	9.7833	105.4708	Vietnam	VN	Hậu Giang
51	Yên Bái	21.7000	104.8667	Vietnam	VN	Yên Bái
54	Lào Cai	22.4194	103.9950	Vietnam	VN	Lào Cai
55	Bến Tre	10.2333	106.3833	Vietnam	VN	Bến Tre

56	Bắc Giang	21.2731	106.1947	Vietnam	VN	Bắc Giang
57	Cao Bằng	22.6731	106.2500	Vietnam	VN	Cao Bằng
59	Hà Giang	22.8233	104.9836	Vietnam	VN	Hà Giang
60	Tuyên Quang	21.8281	105.2156	Vietnam	VN	Tuyên Quang
61	Bắc Kạn	22.1333	105.8333	Vietnam	VN	Bắc Kạn
62	Sơn La	21.3270	103.9141	Vietnam	VN	Sơn La
63	Đông Hà	16.8056	107.0906	Vietnam	VN	Quảng Trị

	capital	population	population_proper
0	admin	13312000.0	7431000.0
2	admin	2103500.0	2103500.0
3	admin	1237300.0	1237300.0
4	admin	1104000.0	1104000.0
6	admin	520000.0	520000.0
7	admin	507469.0	507469.0
8	admin	490000.0	490000.0
9	admin	455230.0	455230.0
10	admin	393294.0	393294.0
11	admin	392279.0	392279.0
12	admin	352108.0	352108.0
13	admin	340000.0	340000.0
14	admin	330000.0	330000.0
15	admin	327000.0	327000.0
16	admin	315270.0	315270.0
17	admin	311000.0	311000.0
18	admin	300000.0	173922.0
19	admin	278658.0	278658.0
20	admin	277539.0	277539.0
21	admin	268167.0	268167.0
22	admin	260252.0	260252.0
24	admin	250660.0	250660.0
25	admin	244277.0	244277.0
26	admin	242840.0	242840.0
27	admin	225000.0	150000.0
29	admin	205333.0	205333.0
31	admin	179773.0	91520.0
32	admin	172915.0	148066.0
33	admin	165396.0	165396.0
34	admin	160325.0	160325.0
37	admin	149837.0	149837.0
38	admin	148000.0	148000.0
40	admin	142900.0	114225.0
41	admin	137498.0	64801.0
42	admin	131360.0	131360.0
43	admin	130517.0	130517.0
44	admin	126370.0	126370.0
46	admin	124143.0	122310.0

```
48   admin    121309.0          121309.0
49   admin    103314.0          103314.0
50   admin     97200.0           97200.0
51   admin     96540.0           96540.0
54   admin     67206.0           36502.0
55   admin     59442.0           59442.0
56   admin     53728.0           53728.0
57   admin     41112.0           41112.0
59   admin     38362.0           32690.0
60   admin     36430.0           36430.0
61   admin     29227.0           29227.0
62   admin     19054.0           19054.0
63   admin     17662.0           17662.0
```

Ploting the cities on the map of Vietnam using folium:

[33]:
```
address = 'Vietnam'
geolocator = Nominatim(user_agent="vn_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Vietnam are {}, {}.'.format(latitude,␣
 ↪longitude))
```

```
The geograpical coordinate of Vietnam are 13.2904027, 108.4265113.
```

[34]:
```
vn_map = folium.Map(location=[latitude, longitude], zoom_start=6)
for lat, lng, city in zip(df['Latitude'], df['Longitude'], df['city']):
    label = '{}'.format(city)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='white',
        fill=True,
        fill_color='orange',
        fill_opacity=0.7,
        parse_html=False).add_to(vn_map)

vn_map
```

[34]:
```
<folium.folium.Map at 0x1f43c5c5280>
```

Now using Foursquare to get information about venues around the cities:

[35]:
```
CLIENT_ID = '2VTMKEILEXMUMNYJZ3TNOCZZDOLI1ME01L1SPYESIXFJE1EC'
CLIENT_SECRET = 'MK5QKWSCBGVWNAGTWZCDODETEFXAVUPI1F0ITN2AA5MTEYH1'
VERSION = '20180605'
```

```
LIMIT = 100
def getNearbyVenues(names, latitudes, longitudes, radius=10000):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

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

        # make the GET request
        results = requests.get(url).json()["response"]['groups'][0]['items']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['id'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name']) for v in results])

    vn_venues = pd.DataFrame([item for venue_list in venues_list for item in␣
↪venue_list])
    vn_venues.columns = ['city',
                  'Lat',
                  'Long',
                  'Venue',
                  'Venue Id',
                  'Venue Lat',
                  'Venue Long',
                  'Venue Category']

    return(vn_venues)
```

[36]:
```
vn_venues = getNearbyVenues(names=df['city'],
                                    latitudes=df['Latitude'],
```

```
                                        longitudes=df['Longitude']
                                    )
```

```
Ho Chi Minh City
Hanoi
Haiphong
Cần Thơ
Biên Hòa
Quảng Hà
Bắc Ninh
Hải Dương
Vinh
Huế
Thanh Hóa
Nha Trang
Nam Định
Buôn Ma Thuột
Thái Nguyên
Vũng Tàu
Cà Mau
Quy Nhơn
Sóc Trăng
Long Xuyên
Việt Trì
Thái Bình
Quảng Ngãi
Ấp Đa Lợi
Rạch Giá
Thủ Dầu Một
Tuy Hòa
Bạc Liêu
Sa Đéc
Phan Thiết
Sơn Tây
Phan Rang-Tháp Chàm
Hạ Long
Hà Tĩnh
Đồng Hới
Châu Đốc
Cẩm Phả
Cao Lãnh
Lạng Sơn
Cam Ranh
Pleiku
Tân An
Trà Vinh
Ninh Bình
Tây Ninh
```

```
Cam Ranh
Mỹ Tho
Hội An
Hòa Bình
Vĩnh Long
Vị Thanh
Yên Bái
Quảng Trị
Phú Quốc
Lào Cai
Bến Tre
Bắc Giang
Cao Bằng
Bình Long
Hà Giang
Tuyên Quang
Bắc Kạn
Sơn La
Đông Hà
```

[37]:
```
vn_venues.head()
```

[37]:

	city	Lat	Long	Venue	Venue Id	Venue Lat	Venue Long	Venue Category
0	Ho Chi Minh City	10.8167	106.6333	Celadon City	4d76e815de42721edbab472b	10.802461	106.618018	Park
1	Ho Chi Minh City	10.8167	106.6333	Seventeen Coffee	4f2d36f9e4b0d411babf803b	10.825413	106.629618	Lounge
2	Ho Chi Minh City	10.8167	106.6333	Sân Golf Tân Sơn Nhất	540e8af5498e77fa82948544	10.829959	106.649974	Golf Course
3	Ho Chi Minh City	10.8167	106.6333	AEON Supermarket	54901356498e6c9bcb3c9cde	10.801783	106.618443	Supermarket
4	Ho Chi Minh City	10.8167	106.6333	IBIS SAIGON AIRPORT Hotel	581c6d1d4e1d9a7c32e3053f	10.813010	106.666183	Hotel

[38]:
```
# Checking number of venues returned for each city.
vn_venues.groupby('city').count()
```

[38]:

city	Lat	Long	Venue	Venue Id	Venue Lat	Venue Long
Biên Hòa	20	20	20	20	20	20
Buôn Ma Thuột	9	9	9	9	9	9
Bình Long	1	1	1	1	1	1
Bạc Liêu	4	4	4	4	4	4
Bắc Giang	9	9	9	9	9	9
…	…	…	…	…	…	…
Vị Thanh	4	4	4	4	4	4

```
Yên Bái          5     5      5         5          5          5
Đông Hà          6     6      6         6          6          6
Đồng Hới        14    14     14        14         14         14
Ấp Đa Lợi       66    66     66        66         66         66
```

```
               Venue Category
city
Biên Hòa                   20
Buôn Ma Thuột               9
Bình Long                   1
Bạc Liêu                    4
Bắc Giang                   9
…                         …
Vị Thanh                    4
Yên Bái                     5
Đông Hà                     6
Đồng Hới                   14
Ấp Đa Lợi                  66

[63 rows x 7 columns]
```

[40]:
```
# Using One-hot coding approach
# one hot encoding
vn_onehot = pd.get_dummies(vn_venues[['Venue Category']], prefix="",␣
↪prefix_sep="")

# add city column back to dataframe
vn_onehot['city'] = vn_venues['city']

# move city column to the first column
fixed_columns = [vn_onehot.columns[-1]] + list(vn_onehot.columns[:-1])
vn_onehot = vn_onehot[fixed_columns]

vn_onehot.head(5)
```

[40]:
```
              city  Afghan Restaurant  Airport  Airport Food Court  \
0  Ho Chi Minh City                  0        0                   0
1  Ho Chi Minh City                  0        0                   0
2  Ho Chi Minh City                  0        0                   0
3  Ho Chi Minh City                  0        0                   0
4  Ho Chi Minh City                  0        0                   0

   Airport Lounge  Airport Service  Airport Terminal  American Restaurant  \
0               0                0                 0                    0
1               0                0                 0                    0
2               0                0                 0                    0
3               0                0                 0                    0
```

```
4               0               0                0                     0

   Antique Shop  Art Gallery  …  Udon Restaurant  Ukrainian Restaurant  \
0             0            0  …                0                     0
1             0            0  …                0                     0
2             0            0  …                0                     0
3             0            0  …                0                     0
4             0            0  …                0                     0

   Vegetarian / Vegan Restaurant  Vietnamese Restaurant  Village  Vineyard  \
0                              0                      0        0         0
1                              0                      0        0         0
2                              0                      0        0         0
3                              0                      0        0         0
4                              0                      0        0         0

   Waterfall  Whisky Bar  Zoo  Zoo Exhibit
0          0           0    0            0
1          0           0    0            0
2          0           0    0            0
3          0           0    0            0
4          0           0    0            0

[5 rows x 197 columns]
```

[41]:
```
vn_group= vn_onehot.groupby('city').mean().reset_index()
vn_group.head(5)
```

[41]:
```
            city  Afghan Restaurant   Airport  Airport Food Court  \
0       Biên Hòa           0.000000  0.000000                 0.0
1  Buôn Ma Thuột           0.000000  0.111111                 0.0
2      Bình Long           0.000000  0.000000                 0.0
3       Bạc Liêu           0.000000  0.000000                 0.0
4      Bắc Giang           0.111111  0.000000                 0.0

   Airport Lounge  Airport Service  Airport Terminal  American Restaurant  \
0        0.000000              0.0               0.0                  0.0
1        0.111111              0.0               0.0                  0.0
2        0.000000              0.0               0.0                  0.0
3        0.000000              0.0               0.0                  0.0
4        0.000000              0.0               0.0                  0.0

   Antique Shop  Art Gallery  …  Udon Restaurant  Ukrainian Restaurant  \
0           0.0          0.0  …              0.0                   0.0
1           0.0          0.0  …              0.0                   0.0
2           0.0          0.0  …              0.0                   0.0
3           0.0          0.0  …              0.0                   0.0
```

```
4           0.0          0.0  …              0.0                    0.0

   Vegetarian / Vegan Restaurant  Vietnamese Restaurant  Village  Vineyard  \
0                            0.0               0.200000      0.0       0.0
1                            0.0               0.000000      0.0       0.0
2                            0.0               1.000000      0.0       0.0
3                            0.0               0.250000      0.0       0.0
4                            0.0               0.111111      0.0       0.0

   Waterfall  Whisky Bar  Zoo  Zoo Exhibit
0        0.0         0.0  0.0          0.0
1        0.0         0.0  0.0          0.0
2        0.0         0.0  0.0          0.0
3        0.0         0.0  0.0          0.0
4        0.0         0.0  0.0          0.0

[5 rows x 197 columns]
```

[42]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

[45]:
```
num_top_venues = 10 # top 10 venues

indicators = ['st', 'nd', 'rd']

# creating  columns according to number of top venues
columns = ['city']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# creating a new dataframe having city names and its top 10 venues
v_venues = pd.DataFrame(columns=columns)
v_venues['city'] = vn_group['city']

for ind in np.arange(vn_group.shape[0]):
    v_venues.iloc[ind, 1:] = return_most_common_venues(vn_group.iloc[ind, :],␣
 ↪num_top_venues)

v_venues.head()  # top 10 venues dataframe
```

[45]:
```
            city  1st Most Common Venue 2nd Most Common Venue  \
0       Biên Hòa  Vietnamese Restaurant             Multiplex
1  Buôn Ma Thuột                  Café                 Hotel
2      Bình Long  Vietnamese Restaurant           Zoo Exhibit
3       Bạc Liêu  Vietnamese Restaurant            Restaurant
4      Bắc Giang      Afghan Restaurant        Cosmetics Shop

  3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
0         Shopping Mall                 Hotel          Noodle House
1               Airport        Airport Lounge         History Museum
2      Department Store           Film Studio  Fast Food Restaurant
3                 Café      Asian Restaurant           Zoo Exhibit
4          Soccer Field          Food Service                   Bar

  6th Most Common Venue 7th Most Common Venue  8th Most Common Venue  \
0   Japanese Restaurant                 Café       Asian Restaurant
1      Asian Restaurant          Dessert Shop            Film Studio
2                  Farm               Factory      Electronics Store
3      Department Store  Fast Food Restaurant                   Farm
4                 Trail     Mobile Phone Shop  Vietnamese Restaurant

         9th Most Common Venue 10th Most Common Venue
0               Train Station            Pizza Place
1        Fast Food Restaurant                   Farm
2  Eastern European Restaurant       Doner Restaurant
3                     Factory      Electronics Store
4                       Hotel            Coffee Shop
```

Apply Kmeans Clustering with k=4:

[46]:
```
# set number of clusters
k = 4

# droping city column to get only top 10 venues columns
vn_clust = vn_group.drop('city', 1)

# run k-means clustering
# fit kmean model with fin_clust dataframe
kmean = KMeans(n_clusters=k, random_state=0).fit(vn_clust)

# check cluster labels generated for each row in the dataframe
kmean.labels_[0:10]
```

[46]:
```
array([0, 3, 2, 3, 0, 0, 0, 3, 0, 1])
```

[47]:
```
# add clustering labels
v_venues.insert(0, 'Cluster Labels', kmean.labels_)
```

```
# copying original city dataframe to new dataframe
vndf = df

# merge city dataframe with city venues to add latitude/longitude for each city
vndf = vndf.join(v_venues.set_index('city'), on='city')

vndf.head() # check new datframe and new cluster label column
```

[47]:

```
              city  Latitude  Longitude  country iso2       Region  capital  \
0  Ho Chi Minh City   10.8167   106.6333  Vietnam   VN  Hồ Chí Minh    admin
1             Hanoi   21.0245   105.8412  Vietnam   VN       Hà Nội  primary
2          Haiphong   20.8000   106.6667  Vietnam   VN    Hải Phòng    admin
3           Cần Thơ   10.0333   105.7833  Vietnam   VN      Cần Thơ    admin
4          Biên Hòa   10.9575   106.8426  Vietnam   VN     Đồng Nai    admin

   population  population_proper  Cluster Labels  1st Most Common Venue  \
0  13312000.0          7431000.0               0  Vietnamese Restaurant
1   7785000.0          7785000.0               0            Coffee Shop
2   2103500.0          2103500.0               0                  Hotel
3   1237300.0          1237300.0               0                  Hotel
4   1104000.0          1104000.0               0  Vietnamese Restaurant

   2nd Most Common Venue  3rd Most Common Venue 4th Most Common Venue  \
0                  Hotel            Coffee Shop                 Café
1                  Hotel  Vietnamese Restaurant          Noodle House
2                  Café            Supermarket                Market
3  Vietnamese Restaurant            Coffee Shop                Resort
4              Multiplex          Shopping Mall                 Hotel

  5th Most Common Venue 6th Most Common Venue 7th Most Common Venue  \
0          Noodle House   Japanese Restaurant           Pizza Place
1             BBQ Joint             Multiplex           Pizza Place
2             Multiplex     Electronics Store      Asian Restaurant
3                  Farm                 Café             Multiplex
4          Noodle House   Japanese Restaurant                  Café

          8th Most Common Venue 9th Most Common Venue 10th Most Common Venue
0  Vegetarian / Vegan Restaurant                  Park       Department Store
1                           Park            Steakhouse            Golf Course
2                         Lounge           Opera House   Ukrainian Restaurant
3                Floating Market                Museum                 Market
4                Asian Restaurant         Train Station            Pizza Place
```

Plotting cities of Vietnam on the map of Vietnam in clusters:

[48]:

```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=6)
```

```
# set color scheme for the clusters
x = np.arange(k)
ys = [i + x + (i*x)**2 for i in range(k)]
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(vndf['Latitude'], vndf['Longitude'],␣
 ↪vndf['city'], vndf['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)

map_clusters
```

[48]:
```
<folium.folium.Map at 0x1f43ce43670>
```

Results of Cluster:

1.3.1 Final results

Show the city for each cluster:

[49]:
```
vndf.loc[vndf['Cluster Labels'] == 0,vndf.columns[[0] + list(range(5, vndf.
 ↪shape[1]))]]
```

[49]:

	city	Region	capital	population	\
0	Ho Chi Minh City	Hồ Chí Minh	admin	13312000.0	
1	Hanoi	Hà Nội	primary	7785000.0	
2	Haiphong	Hải Phòng	admin	2103500.0	
3	Cần Thơ	Cần Thơ	admin	1237300.0	
4	Biên Hòa	Đồng Nai	admin	1104000.0	
5	Quảng Hà	Quảng Nam	NaN	1000000.0	
6	Bắc Ninh	Bắc Ninh	admin	520000.0	
7	Hải Dương	Hải Dương	admin	507469.0	
8	Vinh	Nghệ An	admin	490000.0	
9	Huế	Thừa Thiên-Huế	admin	455230.0	
11	Nha Trang	Khánh Hòa	admin	392279.0	
12	Nam Định	Nam Định	admin	352108.0	
15	Vũng Tàu	Bà Rịa-Vũng Tàu	admin	327000.0	
16	Cà Mau	Cà Mau	admin	315270.0	

17	Quy Nhơn	Bình Định	admin	311000.0
18	Sóc Trăng	Sóc Trăng	admin	300000.0
20	Việt Trì	Phú Thọ	admin	277539.0
24	Rạch Giá	Kiến Giang	admin	250660.0
26	Tuy Hòa	Phú Yên	admin	242840.0
28	Sa Đéc	Đồng Tháp	minor	213610.0
29	Phan Thiết	Bình Thuận	admin	205333.0
30	Sơn Tây	Hà Nội	minor	189547.0
31	Phan Rang-Tháp Chàm	Ninh Thuận	admin	179773.0
32	Hạ Long	Quảng Ninh	admin	172915.0
33	Hà Tĩnh	Hà Tĩnh	admin	165396.0
34	Đồng Hới	Quảng Bình	admin	160325.0
35	Châu Đốc	An Giang	NaN	157298.0
38	Lạng Sơn	Lạng Sơn	admin	148000.0
39	Cam Ranh	Khánh Hòa	NaN	146771.0
41	Tân An	Long An	admin	137498.0
42	Trà Vinh	Trà Vinh	admin	131360.0
43	Ninh Bình	Ninh Bình	admin	130517.0
44	Tây Ninh	Tây Ninh	admin	126370.0
45	Cam Ranh	Khánh Hòa	minor	125311.0
46	Mỹ Tho	Tiền Giang	admin	124143.0
47	Hội An	Quảng Nam	NaN	121716.0
48	Hòa Bình	Hòa Bình	admin	121309.0
49	Vĩnh Long	Vĩnh Long	admin	103314.0
50	Vị Thanh	Hậu Giang	admin	97200.0
52	Quảng Trị	Quảng Trị	NaN	72722.0
53	Phú Quốc	Kiến Giang	minor	70000.0
54	Lào Cai	Lào Cai	admin	67206.0
56	Bắc Giang	Bắc Giang	admin	53728.0
59	Hà Giang	Hà Giang	admin	38362.0
61	Bắc Kạn	Bắc Kạn	admin	29227.0

	population_proper	Cluster Labels	1st Most Common Venue	\
0	7431000.0	0	Vietnamese Restaurant	
1	7785000.0	0	Coffee Shop	
2	2103500.0	0	Hotel	
3	1237300.0	0	Hotel	
4	1104000.0	0	Vietnamese Restaurant	
5	887069.0	0	Hotel	
6	520000.0	0	Restaurant	
7	507469.0	0	Hotel	
8	490000.0	0	Hotel	
9	455230.0	0	Hotel	
11	392279.0	0	Resort	
12	352108.0	0	Shopping Mall	
15	327000.0	0	Vietnamese Restaurant	
16	315270.0	0	Boat or Ferry	

17	311000.0	0	Hotel
18	173922.0	0	Athletics & Sports
20	277539.0	0	Hotel
24	250660.0	0	Seafood Restaurant
26	242840.0	0	Seafood Restaurant
28	213610.0	0	Hotel
29	205333.0	0	Resort
30	40636.0	0	Historic Site
31	91520.0	0	Seafood Restaurant
32	148066.0	0	Hotel
33	165396.0	0	Hotel
34	160325.0	0	Hotel
35	157298.0	0	Hotel
38	148000.0	0	Buddhist Temple
39	23883.0	0	Vietnamese Restaurant
41	64801.0	0	Jewelry Store
42	131360.0	0	Hotel
43	130517.0	0	Hotel
44	126370.0	0	Vietnamese Restaurant
45	125311.0	0	Vietnamese Restaurant
46	122310.0	0	Vietnamese Restaurant
47	121716.0	0	Vietnamese Restaurant
48	121309.0	0	Vietnamese Restaurant
49	103314.0	0	River
50	97200.0	0	Market
52	72722.0	0	Vietnamese Restaurant
53	70000.0	0	Resort
54	36502.0	0	Hotel
56	53728.0	0	Afghan Restaurant
59	32690.0	0	Hostel
61	29227.0	0	Vietnamese Restaurant

	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	\
0	Hotel	Coffee Shop	Café	
1	Hotel	Vietnamese Restaurant	Noodle House	
2	Café	Supermarket	Market	
3	Vietnamese Restaurant	Coffee Shop	Resort	
4	Multiplex	Shopping Mall	Hotel	
5	Resort	Coffee Shop	Vietnamese Restaurant	
6	Hotel	Cosmetics Shop	Zoo Exhibit	
7	Vietnamese Restaurant	Rest Area	Supermarket	
8	Park	Airport	Supermarket	
9	Vietnamese Restaurant	Café	Historic Site	
11	Hotel	Café	Beach	
12	Hotel	Train Station	Zoo Exhibit	
15	Asian Restaurant	Hotel	Seafood Restaurant	
16	Karaoke Bar	Fish & Chips Shop	Café	

17	Vietnamese Restaurant	Historic Site	Seafood Restaurant
18	Factory	Vietnamese Restaurant	Bed & Breakfast
20	Temple	Train Station	Shopping Mall
24	Airport	Resort	Vietnamese Restaurant
26	Airport	Resort	Beach
28	Rest Area	Market	Antique Shop
29	Hotel	Café	Seafood Restaurant
30	Hotel	Resort	Campground
31	Asian Restaurant	Resort	Fried Chicken Joint
32	Seafood Restaurant	Vietnamese Restaurant	Beach
33	Fast Food Restaurant	Vietnamese Restaurant	Udon Restaurant
34	Café	American Restaurant	Noodle House
35	Historic Site	Market	Vietnamese Restaurant
38	Pharmacy	Market	Vietnamese Restaurant
39	Department Store	Café	Resort
41	Vietnamese Restaurant	Restaurant	Insurance Office
42	Market	Vietnamese Restaurant	Park
43	Vietnamese Restaurant	Restaurant	Cave
44	Hotel	Coffee Shop	Park
45	Department Store	Café	Resort
46	Asian Restaurant	Café	Rest Area
47	Hotel	Resort	Coffee Shop
48	Hotel	Cafeteria	Golf Course
49	Miscellaneous Shop	Resort	Chinese Restaurant
50	Hotel	IT Services	Zoo Exhibit
52	River	Historic Site	Candy Store
53	Vietnamese Restaurant	Hotel	Beach
54	Vietnamese Restaurant	Bus Station	Pizza Place
56	Cosmetics Shop	Soccer Field	Food Service
59	Hotel	Resort	Vietnamese Restaurant
61	Hotel	Sake Bar	Diner

	5th Most Common Venue	6th Most Common Venue	\
0	Noodle House	Japanese Restaurant	
1	BBQ Joint	Multiplex	
2	Multiplex	Electronics Store	
3	Farm	Café	
4	Noodle House	Japanese Restaurant	
5	Restaurant	Spa	
6	Department Store	Fast Food Restaurant	
7	Zoo Exhibit	Department Store	
8	Airport Lounge	Airport Service	
9	Bed & Breakfast	Vegetarian / Vegan Restaurant	
11	Vietnamese Restaurant	Asian Restaurant	
12	Department Store	Fast Food Restaurant	
15	Café	Resort	
16	Flea Market	Food Truck	

17	Noodle House	Snack Place
18	Dessert Shop	Film Studio
20	Zoo Exhibit	Dessert Shop
24	Zoo Exhibit	Deli / Bodega
26	Department Store	Fast Food Restaurant
28	Zoo Exhibit	Dessert Shop
29	Vietnamese Restaurant	Spa
30	Golf Course	Zoo Exhibit
31	Shopping Mall	Beach
32	Resort	Asian Restaurant
33	Zoo Exhibit	Department Store
34	Resort	Market
35	Resort	Restaurant
38	Soccer Field	Zoo Exhibit
39	Beach	Bus Station
41	Department Store	Zoo Exhibit
42	Multiplex	Seafood Restaurant
43	Resort	Palace
44	Café	Zoo Exhibit
45	Beach	Bus Station
46	Travel & Transport	Halal Restaurant
47	Tailor Shop	Spa
48	Zoo Exhibit	Department Store
49	Department Store	Fast Food Restaurant
50	Dessert Shop	Film Studio
52	Asian Restaurant	Zoo Exhibit
53	Asian Restaurant	Bar
54	Market	Restaurant
56	Bar	Trail
59	Motel	Department Store
61	Noodle House	Tibetan Restaurant

	7th Most Common Venue	8th Most Common Venue	\
0	Pizza Place	Vegetarian / Vegan Restaurant	
1	Pizza Place	Park	
2	Asian Restaurant	Lounge	
3	Multiplex	Floating Market	
4	Café	Asian Restaurant	
5	Beach	Café	
6	Farm	Factory	
7	Fast Food Restaurant	Farm	
8	Temple	Korean Restaurant	
9	Noodle House	Hotel Pool	
11	Spa	Seafood Restaurant	
12	Farm	Factory	
15	Pool	Beach	
16	Shopping Mall	Bus Station	

17	Train Station	Beach Bar
18	Fast Food Restaurant	Farm
20	Fast Food Restaurant	Farm
24	Farm	Factory
26	Farm	Factory
28	Fast Food Restaurant	Farm
29	Surf Spot	Clothing Store
30	Department Store	Farm
31	Vietnamese Restaurant	Arts & Entertainment
32	Bay	Cave
33	Farm	Factory
34	Pub	Vietnamese Restaurant
35	Mountain	Café
38	Farm	Factory
39	Island	Asian Restaurant
41	Fast Food Restaurant	Farm
42	Zoo Exhibit	Dessert Shop
43	Café	Fast Food Restaurant
44	Dessert Shop	Fast Food Restaurant
45	Island	Asian Restaurant
46	Boat or Ferry	Noodle House
47	Café	Bed & Breakfast
48	Fast Food Restaurant	Farm
49	Farm	Factory
50	Fast Food Restaurant	Farm
52	Department Store	Fast Food Restaurant
53	Spa	Seafood Restaurant
54	Café	Deli / Bodega
56	Mobile Phone Shop	Vietnamese Restaurant
59	Farm	Factory
61	Zoo Exhibit	Deli / Bodega

	9th Most Common Venue	10th Most Common Venue
0	Park	Department Store
1	Steakhouse	Golf Course
2	Opera House	Ukrainian Restaurant
3	Museum	Market
4	Train Station	Pizza Place
5	Hotel Bar	BBQ Joint
6	Electronics Store	Eastern European Restaurant
7	Factory	Electronics Store
8	Electronics Store	Doner Restaurant
9	Breakfast Spot	Monument / Landmark
11	Indian Restaurant	Restaurant
12	Electronics Store	Eastern European Restaurant
15	Coffee Shop	Restaurant
16	Mobile Phone Shop	Hotel

```
17                Beach                          Farm
18    Electronics Store  Eastern European Restaurant
20              Factory            Electronics Store
24    Electronics Store  Eastern European Restaurant
26    Electronics Store  Eastern European Restaurant
28              Factory            Electronics Store
29               Market                   Restaurant
30              Factory            Electronics Store
31                Diner                  Zoo Exhibit
32          Night Market                       Museum
33    Electronics Store  Eastern European Restaurant
34          Coffee Shop                      Airport
35     Department Store                      Factory
38    Electronics Store  Eastern European Restaurant
39        Shopping Mall                        Diner
41              Factory            Electronics Store
42                 Farm                      Factory
43          Film Studio              Buddhist Temple
44                 Farm                      Factory
45        Shopping Mall                        Diner
46          Candy Store                 Music Venue
47            Beach Bar                        Beach
48              Factory            Electronics Store
49    Electronics Store  Eastern European Restaurant
50              Factory            Electronics Store
52                 Farm                      Factory
53           Lighthouse                 Cocktail Bar
54              Factory            Electronics Store
56                Hotel                  Coffee Shop
59    Electronics Store  Eastern European Restaurant
61                 Farm                      Factory
```

[50]:
```
vndf.loc[vndf['Cluster Labels'] == 1,vndf.columns[[0] + list(range(5, vndf.
↪shape[1]))]]
```

[50]:
```
           city       Region capital  population  population_proper  \
21    Thái Bình    Thái Bình   admin    268167.0           268167.0
40       Pleiku      Gia Lai   admin    142900.0           114225.0
57    Cao Bằng     Cao Bằng   admin     41112.0            41112.0
60  Tuyên Quang  Tuyên Quang   admin     36430.0            36430.0

    Cluster Labels 1st Most Common Venue 2nd Most Common Venue  \
21               1                Hotel           Bus Station
40               1                Hotel               Airport
57               1                Hotel           Pizza Place
60               1                Hotel      Asian Restaurant
```

```
    3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
21           Pizza Place      Department Store  Fast Food Restaurant
40  Vietnamese Restaurant           Bus Station           Zoo Exhibit
57                  Café           Zoo Exhibit      Department Store
60           Bus Station          Cocktail Bar           Coffee Shop

   6th Most Common Venue 7th Most Common Venue 8th Most Common Venue  \
21                  Farm               Factory     Electronics Store
40          Dessert Shop           Film Studio  Fast Food Restaurant
57  Fast Food Restaurant                  Farm               Factory
60           Film Studio  Fast Food Restaurant                  Farm

         9th Most Common Venue       10th Most Common Venue
21  Eastern European Restaurant             Doner Restaurant
40                         Farm                      Factory
57           Electronics Store  Eastern European Restaurant
60                     Factory            Electronics Store
```

[51]:
```
vndf.loc[vndf['Cluster Labels'] == 2,vndf.columns[[0] + list(range(5, vndf.
↪shape[1]))]]
```

[51]:
```
         city      Region capital  population  population_proper  \
51    Yên Bái     Yên Bái   admin     96540.0            96540.0
58  Bình Long  Bình Phước     NaN     40279.0            40279.0

    Cluster Labels  1st Most Common Venue 2nd Most Common Venue  \
51               2  Vietnamese Restaurant     Hotpot Restaurant
58               2  Vietnamese Restaurant           Zoo Exhibit

   3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
51          Coffee Shop           Zoo Exhibit          Dessert Shop
58     Department Store           Film Studio  Fast Food Restaurant

   6th Most Common Venue 7th Most Common Venue 8th Most Common Venue  \
51          Film Studio  Fast Food Restaurant                  Farm
58                 Farm               Factory     Electronics Store

         9th Most Common Venue 10th Most Common Venue
51                     Factory      Electronics Store
58  Eastern European Restaurant       Doner Restaurant
```

[52]:
```
vndf.loc[vndf['Cluster Labels'] == 3,vndf.columns[[0] + list(range(5, vndf.
↪shape[1]))]]
```

[52]:
```
             city       Region capital  population  population_proper  \
10      Thanh Hóa    Thanh Hóa   admin    393294.0           393294.0
13  Buôn Ma Thuột      Đắk Lắk   admin    340000.0           340000.0
```

14	Thái Nguyên	Thái Nguyên	admin	330000.0	330000.0
19	Long Xuyên	An Giang	admin	278658.0	278658.0
22	Quảng Ngãi	Quảng Ngãi	admin	260252.0	260252.0
23	Ấp Đa Lợi	Lâm Đồng	NaN	256019.0	131377.0
25	Thủ Dầu Một	Bình Dương	admin	244277.0	244277.0
27	Bạc Liêu	Bạc Liêu	admin	225000.0	150000.0
36	Cẩm Phả	Quảng Ninh	minor	156000.0	156000.0
37	Cao Lãnh	Đồng Tháp	admin	149837.0	149837.0
55	Bến Tre	Bến Tre	admin	59442.0	59442.0
62	Sơn La	Sơn La	admin	19054.0	19054.0
63	Đông Hà	Quảng Trị	admin	17662.0	17662.0

	Cluster Labels	1st Most Common Venue	2nd Most Common Venue
10	3	Café	Hotel
13	3	Café	Hotel
14	3	Café	Hotel
19	3	Shopping Mall	Boat or Ferry
22	3	Café	Coffee Shop
23	3	Café	Vietnamese Restaurant
25	3	Café	Shopping Mall
27	3	Vietnamese Restaurant	Restaurant
36	3	Harbor / Marina	Park
37	3	Café	Department Store
55	3	Café	Candy Store
62	3	Shopping Mall	Historic Site
63	3	Café	Market

	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
10	Grocery Store	Bridge	Asian Restaurant
13	Airport	Airport Lounge	History Museum
14	Resort	Coffee Shop	Restaurant
19	Farm	Café	Bus Station
22	Vietnamese Restaurant	River	Bus Station
23	Coffee Shop	Hotel	Waterfall
25	Multiplex	Vietnamese Restaurant	Cantonese Restaurant
27	Café	Asian Restaurant	Zoo Exhibit
36	Café	Boat Rental	Korean Restaurant
37	Hotel	Flea Market	Vietnamese Restaurant
55	Street Food Gathering	Hotel	Vietnamese Restaurant
62	Coffee Shop	Café	Zoo Exhibit
63	Vietnamese Restaurant	Bus Station	Supermarket

	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue
10	Train Station	Zoo Exhibit	Dessert Shop
13	Asian Restaurant	Dessert Shop	Film Studio
14	Fried Chicken Joint	Nature Preserve	Zoo Exhibit
19	Fast Food Restaurant	Factory	Electronics Store

```
22          Zoo Exhibit          Dessert Shop  Fast Food Restaurant
23                 Lake           Snack Place            Restaurant
25   Japanese Restaurant           Theme Park    Italian Restaurant
27      Department Store  Fast Food Restaurant                  Farm
36          Zoo Exhibit                 Diner           Film Studio
37    Electronics Store                 River           Zoo Exhibit
55               Market          Food Service                  Lake
62                 Farm               Factory     Electronics Store
63          Zoo Exhibit          Dessert Shop  Fast Food Restaurant

         9th Most Common Venue 10th Most Common Venue
10        Fast Food Restaurant                   Farm
13        Fast Food Restaurant                   Farm
14            Department Store                Factory
19  Eastern European Restaurant       Doner Restaurant
22                        Farm                Factory
23                 Golf Course         Sandwich Place
25                   Cafeteria                  Motel
27                     Factory      Electronics Store
36        Fast Food Restaurant                   Farm
37                        Farm                Factory
55                       Diner   Fast Food Restaurant
62  Eastern European Restaurant       Doner Restaurant
63                        Farm                Factory
```

[54]:
```
#list of cities suitable for opening a hotels
suit_city = vndf.loc[vndf['Cluster Labels'] == 1,vndf.columns[[0] +␣
 ↪list(range(5, vndf.shape[1]))]]
suit_list = suit_city['city'].values.tolist()
suit_list
```

[54]:
```
['Thái Bình', 'Pleiku', 'Cao Bằng', 'Tuyên Quang']
```

1.4 4. Conclusion

From the cluster results, I can tell my colleage that he can think of opening his first hotel in one of 4 cities listed above: Thai Binh, Pleiku, Cao Bang, Tuyen Quang.

[ ]:
```

```