Applied Data Science Capstone

1. Introduction

The Socialist Republic of Vietnam is an S-shaped strip of land, located in the center of Southeast Asia, in the east of the Indochinese peninsula, on the north by China, on the west by Laos, Cambodia, and in the east. South overlooks the East Sea and Pacific Ocean. Vietnam coastline is 3 260 km long, land border is 4 510 km long. On the mainland, from the northernmost point to the southernmost point (according to the flight path), the length is 650km, from the easternmost point to the westernmost point where the widest place is 600km (North), 400 km (South), the narrowest place is 50km. (Quang Binh).

It is a country rich in cultural and economic traditions that are on the rise in recent years. Vietnam promises to be one of the tourism, shopping and entertainment centers of the region. With many landscapes and historical sites, each year tens of millions of people come to relax and visit here.

Realizing that when tourists come to Vietnam and especially Hanoi capital, there are still many difficulties in identifying and searching for places, entertainment services, eating and visiting here.

Just as it is difficult for investors and service developers to have an overview of the current status of services in each area in Hanoi city.

After completing the IBM Data Science course, I came up with an idea to build an application that provides tourists as well as investors with an overview, the status of services in the area to confirm. to decide where to visit, eat, and develop services here.

2. Data description

- Data on administrative geography of the city of Hanoi

https://diaocthinhvuong.vn/danh-sach-don-vi-hanh-chinh-thanh-pho-ha-noi/

(I use BeatifulSoup to extract data from the website)

- Once I have the administrative geographic data I use the Foursquare API to discover the types of services that are active in the area.

- Foursquare account info to make a request

https://developer.foursquare.com/

3. Methodology

In this section, I will guide how to get source data and perform data processing and analysis:

- Data retrieval, exploration and wrangling
- Performing K-means clustering algorithm to segment neighborhoods
- Visualizing population projections and neighborhood segments

Data retrieval, exploration and wrangling

- First use Beautifulsoup to get data from the website.

https://pypi.org/project/beautifulsoup4/

- The following results:

```
[6]: source = requests.get('https://diaocthinhvuong.vn/danh-sach-don-vi-hanh-chinh-thanh-pho-ha-noi/').text
soup = BeautifulSoup(source, 'html5lib')
print(soup.title)
from IPython.display import display_html
tab = str(soup.table)
display_html(tab, raw=True)
```

```
<title>Danh sách đơn vị hành chính trực thuộc thành phố Hà Nội</title>
```

Danh sách các đơn vị hành chính trực thuộc thành phố Hà Nội

STT	Quận Huyện	Mã QH	Phường Xã	Mã PX	Cấp
1	Quận Ba Đình	001	Phường Phúc Xá	00001	Phường
2	Quận Ba Đình	001	Phường Trúc Bạch	00004	Phường
3	Quận Ba Đình	001	Phường Vĩnh Phúc	00006	Phường
4	Quận Ba Đình	001	Phường Cống Vị	00007	Phường
5	Quận Ba Đình	001	Phường Liễu Giai	00008	Phường
6	Quận Ba Đình	001	Phường Nguyễn Trung Trực	00010	Phường
7	Quận Ba Đình	001	Phường Quán Thánh	00013	Phường
8	Quận Ba Đình	001	Phường Ngọc Hà	00016	Phường
9	Quận Ba Đình	001	Phường Điện Biên	00019	Phường
10	Quận Ba Đình	001	Phường Đội Cấn	00022	Phường

- After that, I select some necessary information for analysis such as postcode, name of district, commune and ward:

- The following results:

```
[30]: table_contents = []
table = soup.find('table')
for row in table.findAll('tr'):
    row_contents = {}
    row_contents['PostCodeDistrict'] = row.findChildren()[2].text
    row_contents['District'] = row.findChildren()[1].text
    row_contents['Ward'] = row.findChildren()[3].text
    row_contents['PostCodeWard'] = row.findChildren()[4].text
    table_contents.append(row_contents)

df_original = pd.DataFrame(table_contents)
df_original = df[1:]
df_original.head()
```

[30]:

	PostCodeDistrict	District	Ward	PostCodeWard
2	001	Quận Ba Đình	Phường Trúc Bạch	00004
3	001	Quận Ba Đình	Phường Vĩnh Phúc	00006
4	001	Quận Ba Đình	Phường Cống Vị	00007
5	001	Quận Ba Đình	Phường Liễu Giai	00008
6	001	Quận Ba Đình	Phường Nguyễn Trung Trực	00010

- Continue to filter data by district:

- The following results

```
for postCodeDistrict in df_original['PostCodeDistrict'].unique():
    new_row = {}
    new_row['PostCode'] = postCodeDistrict
    new_row['District'] = df_original[df_original['PostCodeDistrict'] == postCodeDistrict].iloc[0, 1]
    new_row['Wards'] = ', '.join((df_original[df_original['PostCodeDistrict'] == postCodeDistrict].iloc[:, 2]).to_numpy())
    info_districts.append(new_row)

df_info_districts = pd.DataFrame(info_districts)
df_info_districts.head()
```

[73]:

	PostCode	District	Wards
0	001	Quận Ba Đình	Phường Trúc Bạch, Phường Vĩnh Phúc, Phường Cốn...
1	002	Quận Hoàn Kiếm	Phường Phúc Tân, Phường Đồng Xuân, Phường Hàng...
2	003	Quận Tây Hồ	Phường Phú Thượng, Phường Nhật Tân, Phường Tứ ...
3	004	Quận Long Biên	Phường Thượng Thanh, Phường Ngọc Thụy, Phường ...
4	005	Quận Cầu Giấy	Phường Nghĩa Đô, Phường Nghĩa Tân, Phường Mai ...

- After I have administrative geographic information, I use geocoders to get information about the geographical coordinates of each area.

- The following results:

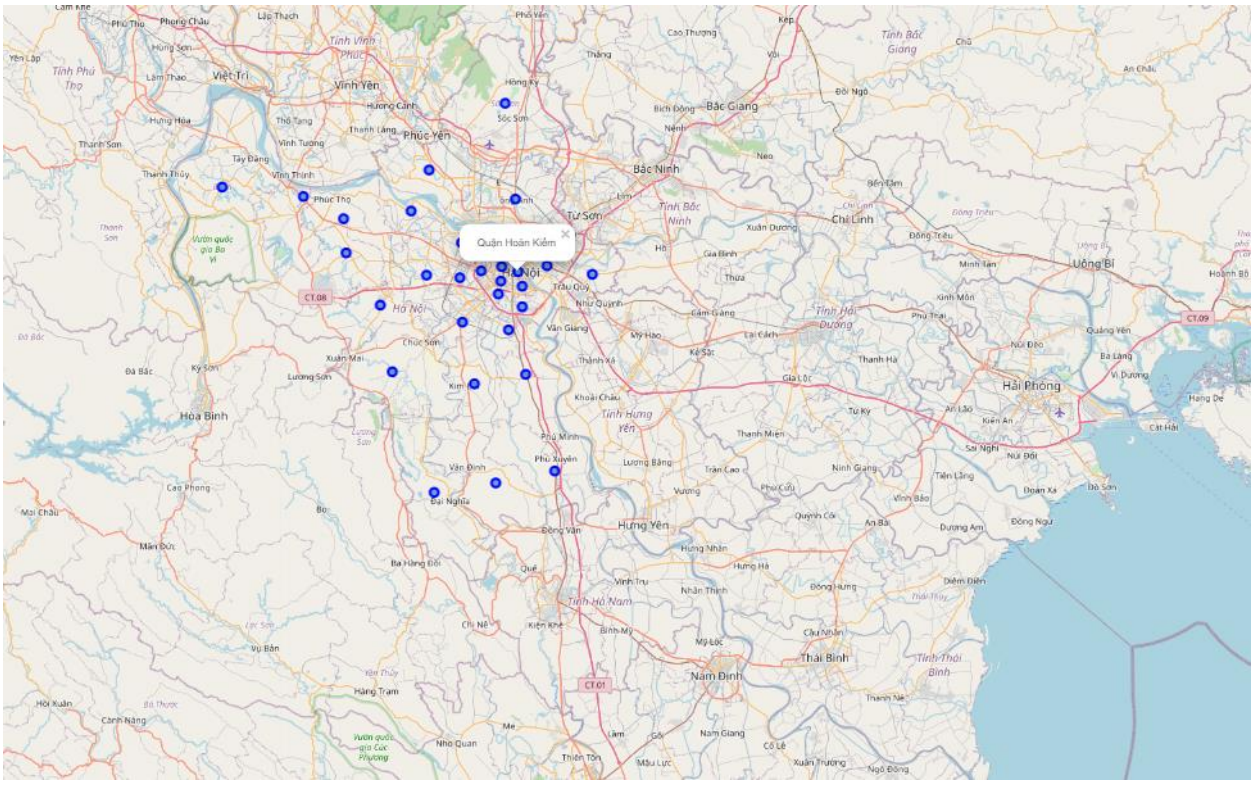
```
from geopy.geocoders import Nominatim # module to convert an address into latitude and longitude values
geolocator = Nominatim(user_agent="HaNoi")

major_Dist_Coord = df_info_districts['District'].apply(geolocator.geocode).apply(lambda x: (x.latitude, x.longitude))
df_info_districts[['Latitude', 'Longitude']] = major_Dist_Coord.apply(pd.Series)
df_info_districts.head()
```

	PostCode	District	Wards	Latitude	Longitude
0	001	Quận Ba Đình	Phường Trúc Bạch, Phường Vĩnh Phúc, Phường Cốn...	21.035552	105.824835
1	002	Quận Hoàn Kiếm	Phường Phúc Tân, Phường Đồng Xuân, Phường Hàng...	21.028524	105.850716
2	003	Quận Tây Hồ	Phường Phú Thượng, Phường Nhật Tân, Phường Tứ ...	21.070976	105.823682
3	004	Quận Long Biên	Phường Thượng Thanh, Phường Ngọc Thụy, Phường ...	21.036685	105.897756
4	005	Quận Cầu Giấy	Phường Nghĩa Đô, Phường Nghĩa Tân, Phường Mai ...	21.029727	105.791333

- Perform data visualization to know the geographical distribution of each area:

- The following results:



- Use the Foursquare API to do a search for all types of related services in each area

Foursquare API is used to explore types of venues in each area. Foursquare identifies 10 top level categories. There are multiple sub categories which will not be used it for the time

- Arts & Entertainment
- College & University
- Event
- Food
- Nightlife Spot
- Outdoors & Recreation
- Professional & Other Places
- Residence
- Shop & Service
- Travel & Transport
- The following results

	District	District Latitude	District Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Quận Ba Đình	21.035552	105.824835	Bánh Xèo - Nem Lụi Đà Nẵng	21.034937	105.825510	Vietnamese Restaurant
1	Quận Ba Đình	21.035552	105.824835	Cơm Chay Hà Thành	21.032318	105.825379	Vegetarian / Vegan Restaurant
2	Quận Ba Đình	21.035552	105.824835	Ashima	21.034310	105.827236	Hotpot Restaurant
3	Quận Ba Đình	21.035552	105.824835	Eté Resto Bar	21.034335	105.827238	Bar
4	Quận Ba Đình	21.035552	105.824835	Hồ Hữu Tiệp (B52 Lake) (Hồ Hữu Tiệp)	21.034556	105.826089	Lake

- Standardize data on each type of service to prepare for the training process

- The following results:

```
.25]: hanoi_onehot = pd.get_dummies(hanoi_venues[['Venue Category']], prefix="", prefix_sep="")
hanoi_onehot['District'] = hanoi_venues['District']

fixed_columns = [hanoi_onehot.columns[-1]] + list(hanoi_onehot.columns[:-1])
hanoi_onehot = hanoi_onehot[fixed_columns]
hanoi_onehot.head()
```

.25]:

	District	Arts & Crafts Store	Asian Restaurant	BBQ Joint	Bakery	Bar	Beer Bar	Brewery	Café	Camera Store	...	Spa	Stadium	Steakhouse	Sushi Restaurant	Train Station	Ukrainian Restaurant	Vegetarian / Vegan Restaurant	Vietnamese Restaurant	W
0	Quận Ba Đình	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	
1	Quận Ba Đình	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	
2	Quận Ba Đình	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
3	Quận Ba Đình	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	
4	Quận Ba Đình	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	

5 rows × 59 columns

```
hanoi_grouped = hanoi_onehot.groupby('District').mean().reset_index()
hanoi_grouped
```

	District	Arts & Crafts Store	Asian Restaurant	BBQ Joint	Bakery	Bar	Beer Bar	Brewery	Café	Camera Store	...	Spa	Stadium	Steakhouse	Sushi Restaurant	Train Station	Ukrainian Restaurant	Vegetarian / Vegan Restaurant	Vietnamese Restaurant	Wedding Hall	Wine Bar
0	Huyện Thường Tín	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.250000	0.25	...	0.000000	0.0	0.000000	0.000000	0.25	0.000000	0.000000	0.250000	0.0	0.000000
1	Huyện Đông Anh	0.000000	0.2	0.000000	0.000000	0.000000	0.000000	0.000000	0.600000	0.00	...	0.000000	0.0	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.0	0.000000
2	Quận Ba Đình	0.000000	0.0	0.000000	0.000000	0.100000	0.000000	0.000000	0.100000	0.00	...	0.000000	0.0	0.000000	0.000000	0.00	0.000000	0.200000	0.100000	0.0	0.000000
3	Quận Bắc Từ Liêm	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	...	0.000000	0.0	0.000000	0.000000	0.00	0.000000	0.000000	1.000000	0.0	0.000000

- Display the top services in each area for an overview:

- The following results:

```
num_top_venues = 5
for hood in hanoi_grouped['District']:
    print('----' + hood + '----')
    temp = hanoi_grouped[hanoi_grouped["District"] == hood].T.reset_index()
    temp.columns = ['venue', 'freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

```
----Huyện Thường Tín----
                venue  freq
0        Camera Store  0.25
1  Vietnamese Restaurant  0.25
2        Train Station  0.25
3                Café  0.25
4        Movie Theater  0.00


----Huyện Đông Anh----
              venue  freq
0             Café   0.6
1  Asian Restaurant   0.2
2     Shopping Mall   0.2
3              Pub   0.0
4             Lake   0.0


----Quận Ba Đình----
                        venue  freq
0  Vegetarian / Vegan Restaurant   0.2
1                          Lake   0.1
2             Hotpot Restaurant   0.1
3          Vietnamese Restaurant   0.1
4                           Bar   0.1
```

- Get 10 services with high frequency in each area for training

- The following results:

```
district_venues_sorted.head()
```

	District	1st Most common venue	2nd Most common venue	3rd Most common venue	4th Most common venue	5th Most common venue	6th Most common venue	7th Most common venue	8th Most common venue	9th Most common venue	10th Most common venue
0	Huyện Thường Tín	Café	Vietnamese Restaurant	Train Station	Camera Store	Wine Bar	Fast Food Restaurant	Fish & Chips Shop	Food Truck	Garden	Historic Site
1	Huyện Đông Anh	Café	Asian Restaurant	Shopping Mall	Wine Bar	Electronics Store	Hotpot Restaurant	Hotel	Hostel	History Museum	Historic Site
2	Quận Ba Đình	Vegetarian / Vegan Restaurant	Hotpot Restaurant	Vietnamese Restaurant	Lake	Bar	History Museum	Cosmetics Shop	Café	Coffee Shop	Garden

Performing K-means clustering algorithm to segment neighborhoods

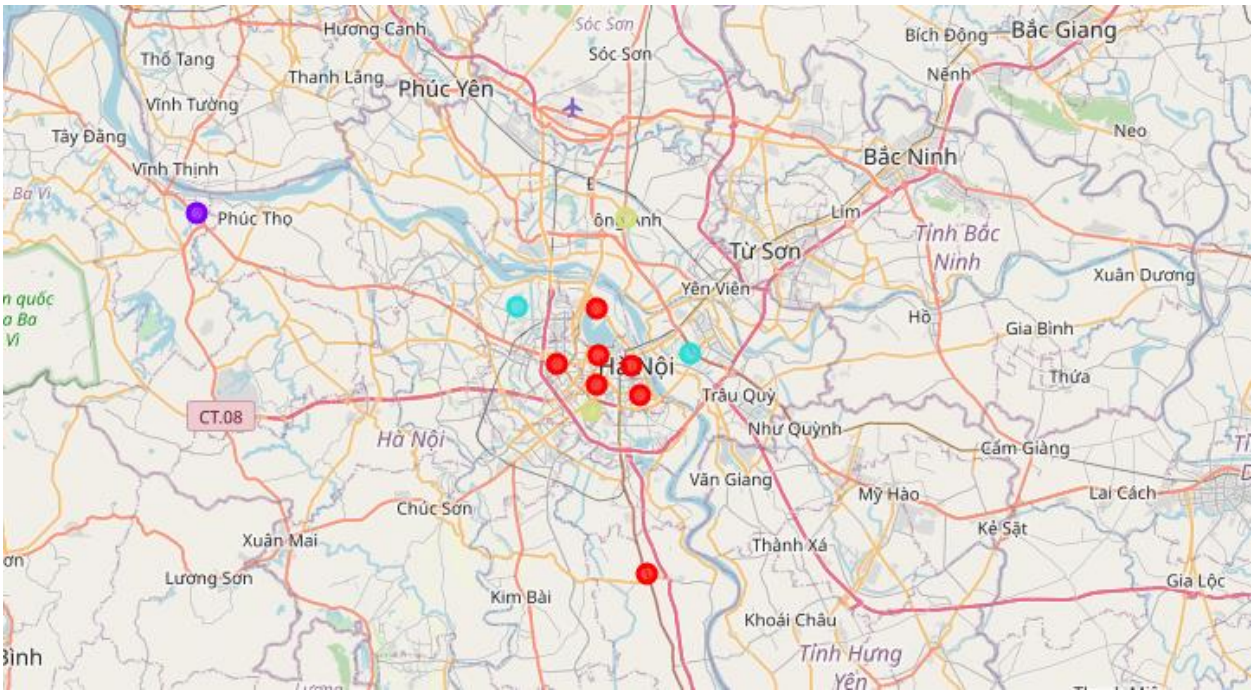
```
kclusters = 4
hanoi_grouped_clustering = hanoi_grouped.drop('District', 1)
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(hanoi_grouped_clustering)
kmeans.labels_
```

```
(30, 5)
```

```
district_venues_sorted = district_venues_sorted.drop('Cluster Labels', axis=1)
district_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
district_venues_sorted
hanoi_merged = df_info_districts
hanoi_merged = hanoi_merged.join(district_venues_sorted.set_index('District'), on='District')

hanoi_merged.head()
```

- Perform model training and display results



4. Result

Categorization of development areas with typical services attached

The downtown area is popular with restaurant, hotel services and sightseeing monuments

- The following result:

The downtown area is popular with restaurant, hotel services and sightseeing monuments

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

	District	Cluster Labels	1st Most common venue	2nd Most common venue	3rd Most common venue	4th Most common venue	5th Most common venue	6th Most common venue	7th Most common venue	8th Most common venue	9th Most common venue	10th Most common venue
0	Quận Ba Đình	0	Vegetarian / Vegan Restaurant	Hotpot Restaurant	Vietnamese Restaurant	Lake	Bar	History Museum	Cosmetics Shop	Café	Coffee Shop	Garden
1	Quận Hoàn Kiếm	0	Café	Hotel	Coffee Shop	Noodle House	Vietnamese Restaurant	Hostel	Fast Food Restaurant	Spa	BBQ Joint	Park
2	Quận Tây Hồ	0	Café	Vietnamese Restaurant	Noodle House	Wine Bar	Beer Bar	Lounge	Pastry Shop	Pub	Rock Club	Modern European Restaurant
4	Quận Cầu Giấy	0	Japanese Restaurant	Vietnamese Restaurant	Park	BBQ Joint	Korean Restaurant	Fish & Chips Shop	Seafood Restaurant	Garden	Fast Food Restaurant	Food Truck

Suburban area with a number of eateries, tourist sites

- The following result:

Suburban area with a number of eateries, tourist sites

```
hanoi_merged.loc[hanoi_merged['Cluster Labels'] == 1, hanoi_merged.columns[[1] + list(range(5, hanoi_merged.shape[1]))]]
```

	District	Cluster Labels	1st Most common venue	2nd Most common venue	3rd Most common venue	4th Most common venue	5th Most common venue	6th Most common venue	7th Most common venue	8th Most common venue	9th Most common venue	10th Most common venue
17	Thị xã Sơn Tây	1	Juice Bar	Historic Site	Electronics Store	Ice Cream Shop	Hotpot Restaurant	Hotel	Hostel	History Museum	Golf Course	Garden

The area is famous for traditional Vietnamese dishes

- The following result:

The area is famous for traditional Vietnamese dishes

```
hanoi_merged.loc[hanoi_merged['Cluster Labels'] == 2, hanoi_merged.columns[[1] + list(range(5, hanoi_merged.shape[1]))]]
```

	District	Cluster Labels	1st Most common venue	2nd Most common venue	3rd Most common venue	4th Most common venue	5th Most common venue	6th Most common venue	7th Most common venue	8th Most common venue	9th Most common venue	10th Most common venue
3	Quận Long Biên	2	Vietnamese Restaurant	Golf Course	Wine Bar	Dim Sum Restaurant	Hotpot Restaurant	Hotel	Hostel	History Museum	Historic Site	Garden
14	Quận Bắc Từ Liêm	2	Vietnamese Restaurant	Wine Bar	Dim Sum Restaurant	Hotpot Restaurant	Hotel	Hostel	History Museum	Historic Site	Golf Course	Garden

Suburban area cafes, restaurants and some entertainment services, shopping

- The following result:

Suburban area cafes, restaurants and some entertainment services, shopping

```
hanoi_merged.loc[hanoi_merged['Cluster Labels'] == 3, hanoi_merged.columns[[1] + list(range(5, hanoi_merged.shape[1]))]]
```

	District	Cluster Labels	1st Most common venue	2nd Most common venue	3rd Most common venue	4th Most common venue	5th Most common venue	6th Most common venue	7th Most common venue	8th Most common venue	9th Most common venue	10th Most common venue
8	Quận Thanh Xuân	3	Café	Korean Restaurant	Soccer Field	Performing Arts Venue	Wine Bar	Electronics Store	Hotel	Hostel	History Museum	Historic Site
10	Huyện Đông Anh	3	Café	Asian Restaurant	Shopping Mall	Wine Bar	Electronics Store	Hotpot Restaurant	Hotel	Hostel	History Museum	Historic Site

5. Discuss

The application is for academic purposes only to support all knowledge gained from the course.

The results of the application are not really impressive because the available data is still very limited. The problem applied to Hanoi city is still very specific because most services are concentrated in the city center

6. Conclude

I have accomplished the project's goal of giving visitors and developers an overview of the current state of service development in different regions.

7. Acknowledgments and sources

I sincerely thank the IBM Data Science Professional Certificate course for having a scientific study path, providing me with a lot of knowledge about Data Science.

The course helps me to have an overview and detail of a Data Science project in practice with many specific and vivid lessons.

https://www.coursera.org/professional-certificates/ibm-machine-learning

Thank you for review my final project !