Capstone Project - Battle of Neighborhoods in Dong Da District, Ha Noi

Applied Data Science Capstone by IBM/Coursera

Table of contents

1. **Introduction: Business Problem** 1
2. **Data** 1
3. **Methodology** 6
4. **Analysis** 7
5. **Results and Discussion** 8
6. **Conclusion** 9
7. **Final Notes** 9

1. Introduction: Business Problem

My friend wanted to open a restaurant or a cafe in Dong Da district, Ha Noi, but he didn't know where to open with little competition. This data analysis article will clarify and may help him with some useful information for his decision

In this project we will try to find an optimal location for a restaurant or cafe. Specifically, this report will be targeted to stakeholders interested in opening an **Restaurant or Cafe** in **Dong Da District, Ha Noi**, Viet nam.

We will use our data science powers to generate a few most promissing neighborhoods based on this criteria. Advantages of each area will then be clearly expressed so that best possible final location can be chosen by stakeholders.

2. Data

Based on definition of our problem, factors that will influence our decission are:

- Detail information of neighborhoods in Dong Da District, list of districts, wards of Dong Da district, Ha Noi from the following URL https://www.gso.gov.vn/dmhc2015/Default.aspx or file data xls from the following https://github.com/TC1894/Coursera_Capstone/blob/master/DONGDA_DISTRICT.xls

- Number of existing restaurants in the neighborhood (any type of restaurant)

Google map API

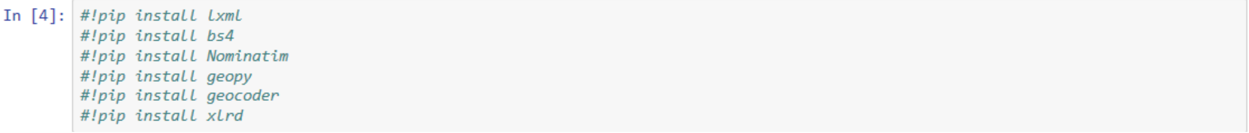
This project would use Google Map API Geocoder to get the Latitude and Longitude of each area

Foursquare API

This project would use Four-square API as its prime data gathering source. This API provides the ability to perform location search, location sharing and details about a business.

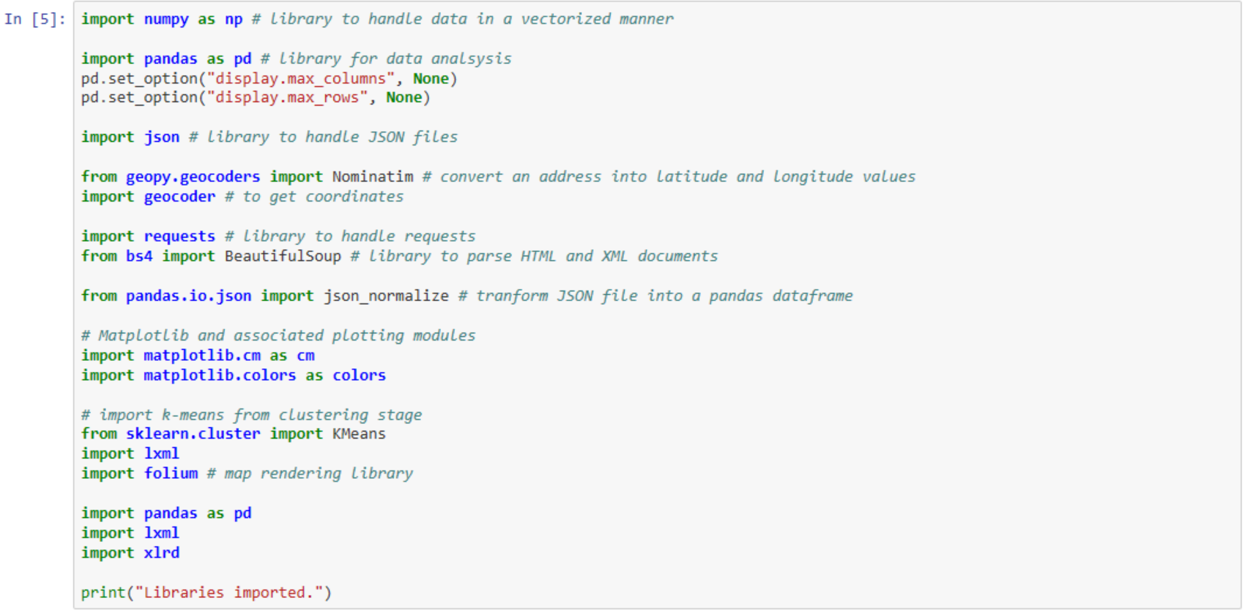
Step by step following

install packages



```
In [4]: #!pip install lxml
        #!pip install bs4
        #!pip install Nominatim
        #!pip install geopy
        #!pip install geocoder
        #!pip install xlrd
```

2.1. Load necessary library



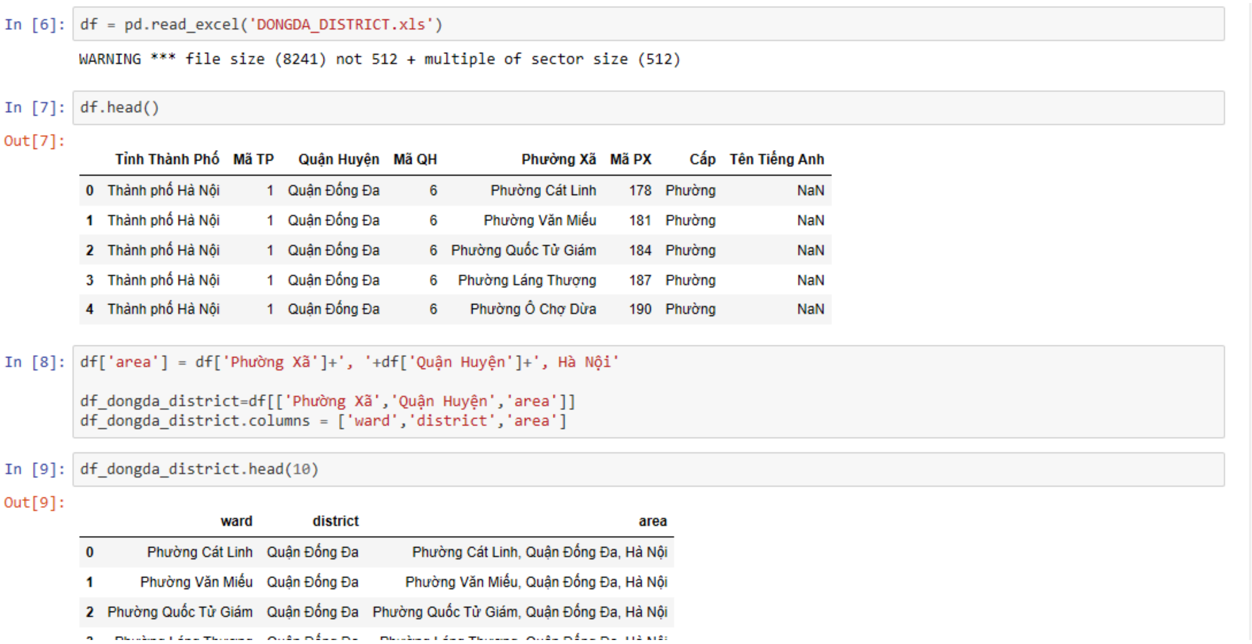
```
In [5]: import numpy as np # library to handle data in a vectorized manner

        import pandas as pd # library for data analsysis
        pd.set_option("display.max_columns", None)
        pd.set_option("display.max_rows", None)

        import json # library to handle JSON files

        from geopy.geocoders import Nominatim # convert an address into latitude and longitude values
        import geocoder # to get coordinates

        import requests # library to handle requests
        from bs4 import BeautifulSoup # library to parse HTML and XML documents

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

        # Matplotlib and associated plotting modules
        import matplotlib.cm as cm
        import matplotlib.colors as colors

        # import k-means from clustering stage
        from sklearn.cluster import KMeans
        import lxml
        import folium # map rendering library

        import pandas as pd
        import lxml
        import xlrd

        print("Libraries imported.")
```

2.2. Get Data Dong Da districts

https://www.gso.gov.vn/dmhc2015/Default.aspx

https://github.com/TC1894/Coursera_Capstone/blob/master/DONGDA_DISTRICT.xls

2.3. Load file excel districts, wards of VietNam



```
In [6]: df = pd.read_excel('DONGDA_DISTRICT.xls')
```

```
WARNING *** file size (8241) not 512 + multiple of sector size (512)
```

```
In [7]: df.head()
```

Out[7]:

	Tỉnh Thành Phố	Mã TP	Quận Huyện	Mã QH	Phường Xã	Mã PX	Cấp	Tên Tiếng Anh
0	Thành phố Hà Nội	1	Quận Đống Đa	6	Phường Cát Linh	178	Phường	NaN
1	Thành phố Hà Nội	1	Quận Đống Đa	6	Phường Văn Miếu	181	Phường	NaN
2	Thành phố Hà Nội	1	Quận Đống Đa	6	Phường Quốc Tử Giám	184	Phường	NaN
3	Thành phố Hà Nội	1	Quận Đống Đa	6	Phường Láng Thượng	187	Phường	NaN
4	Thành phố Hà Nội	1	Quận Đống Đa	6	Phường Ô Chợ Dừa	190	Phường	NaN

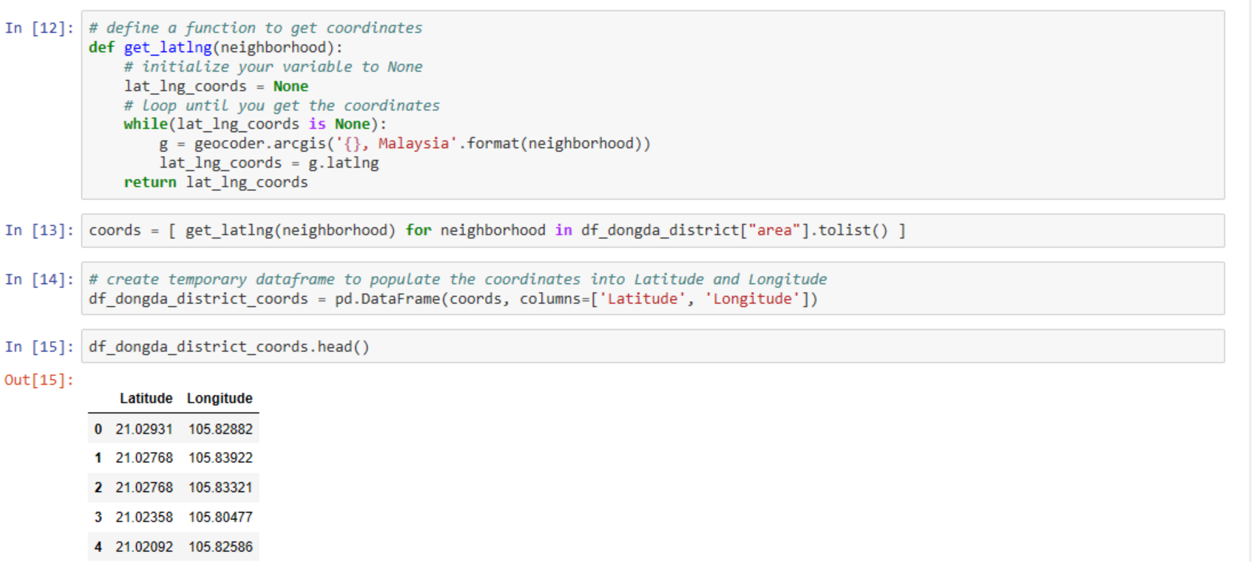
```
In [8]: df['area'] = df['Phường Xã']+', '+df['Quận Huyện']+', Hà Nội'

        df_dongda_district=df[['Phường Xã','Quận Huyện','area']]
        df_dongda_district.columns = ['ward','district','area']
```

```
In [9]: df_dongda_district.head(10)
```

Out[9]:

	ward	district	area
0	Phường Cát Linh	Quận Đống Đa	Phường Cát Linh, Quận Đống Đa, Hà Nội
1	Phường Văn Miếu	Quận Đống Đa	Phường Văn Miếu, Quận Đống Đa, Hà Nội
2	Phường Quốc Tử Giám	Quận Đống Đa	Phường Quốc Tử Giám, Quận Đống Đa, Hà Nội
3	Phường Láng Thượng	Quận Đống Đa	Phường Láng Thượng, Quận Đống Đa, Hà Nội

2.4. Add latitude, longitude by call Google Geocode API



```
In [12]: # define a function to get coordinates
         def get_latlng(neighborhood):
             # initialize your variable to None
             lat_lng_coords = None
             # loop until you get the coordinates
             while(lat_lng_coords is None):
                 g = geocoder.arcgis('{}, Malaysia'.format(neighborhood))
                 lat_lng_coords = g.latlng
             return lat_lng_coords
```

```
In [13]: coords = [ get_latlng(neighborhood) for neighborhood in df_dongda_district["area"].tolist() ]
```

```
In [14]: # create temporary dataframe to populate the coordinates into Latitude and Longitude
         df_dongda_district_coords = pd.DataFrame(coords, columns=['Latitude', 'Longitude'])
```

```
In [15]: df_dongda_district_coords.head()
```

Out[15]:

	Latitude	Longitude
0	21.02931	105.82882
1	21.02768	105.83922
2	21.02768	105.83321
3	21.02358	105.80477
4	21.02092	105.82586

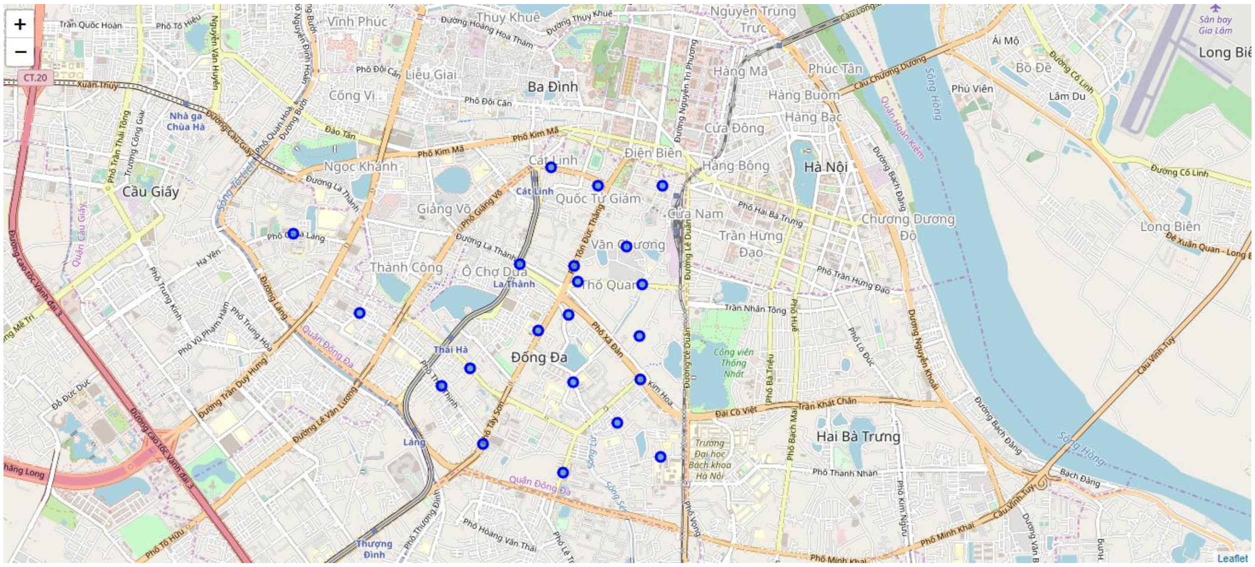
2.5. Create a map of Dong da district's Ha Noi with neighborhoods superimposed on top

```
In [18]: address='Đống Đa, Hà Nội, Việt Nam'
         geolocator = Nominatim(user_agent="HaNoi")

         location = geolocator.geocode(address)
         lat_HN=location.latitude
         long_HN =location.longitude
         print('The geograpical coodinate of Dong Da District, HaNoi are {},{}.'.format(lat_HN,long_HN))
```

```
The geograpical coodinate of Dong Da District, HaNoi are 21.0128913,105.8277098.
```

```
In [52]: map_HN = folium.Map(location=[lat_HN, long_HN], zoom_start=13)

         # add markers to map
         for lat, lng, Neighbourhood in zip(df_dongda_district_new['Latitude'], df_dongda_district_new['Longitude'], df_dongda_district_n
         ew['ward']):
             label = '{}'.format(Neighbourhood)
             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,
                 parse_html=False).add_to(map_HN)

         map_HN
```

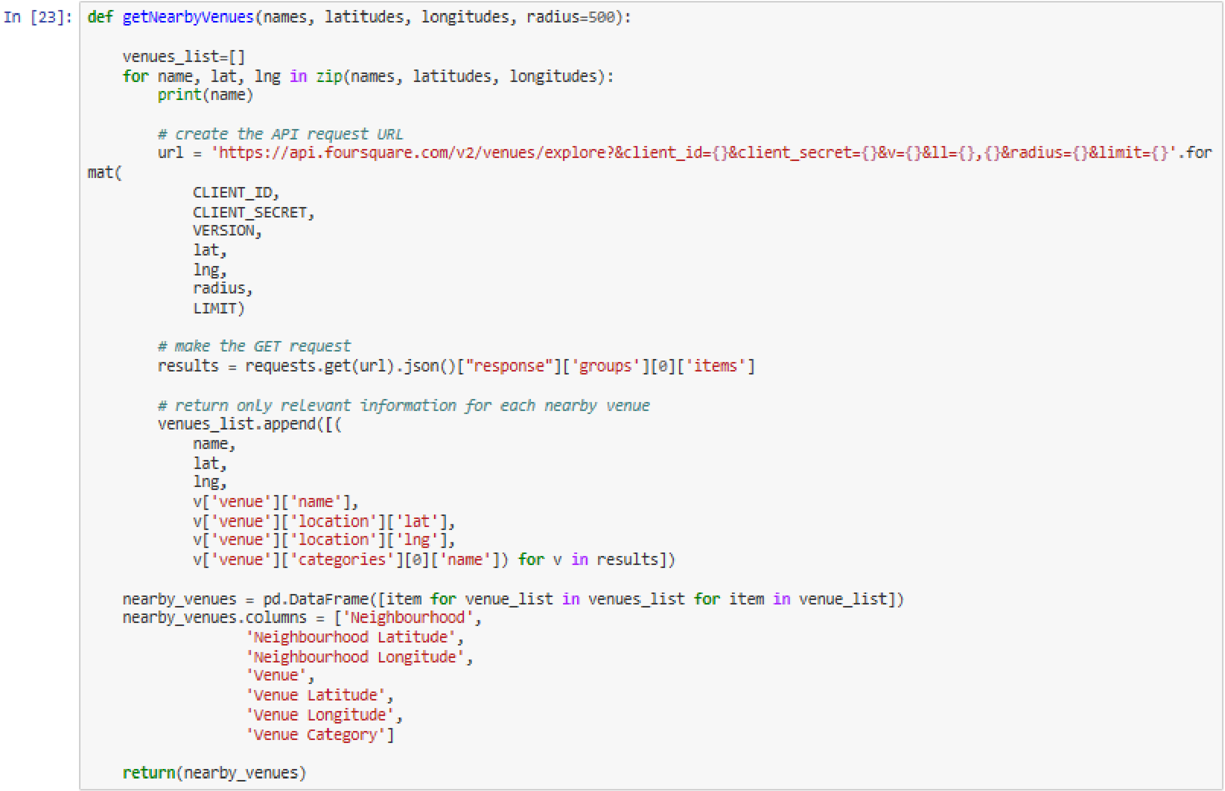
Out[52]:



2.6. Use the Foursquare API to explore the neighborhoods

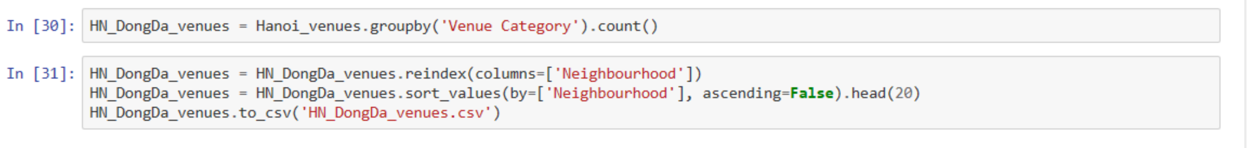
```
In [21]: # define Foursquare Credentials and Version
         CLIENT_ID='1QOE1NIUN3XHN0WH2PUFTX02E4OVH2WJTDZ1HLX01JUZKXD4'
         CLIENT_SECRET='12OUF5GTP5NCYOGLEBLIVLDQBQISD3XCE2EBKH5TWGY4E520'
         VERSION=20180605
```

```
In [22]: # defining radius and limit of venues to get
         radius=500
         LIMIT=100
```



```
In [23]: def getNearbyVenues(names, latitudes, longitudes, radius=500):

             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={}'.for
         mat(
                     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']['location']['lat'],
                     v['venue']['location']['lng'],
                     v['venue']['categories'][0]['name']) for v in results])

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

             return(nearby_venues)
```

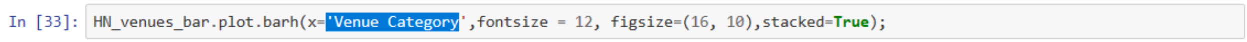
Check how many venues were returned for each neighborhood



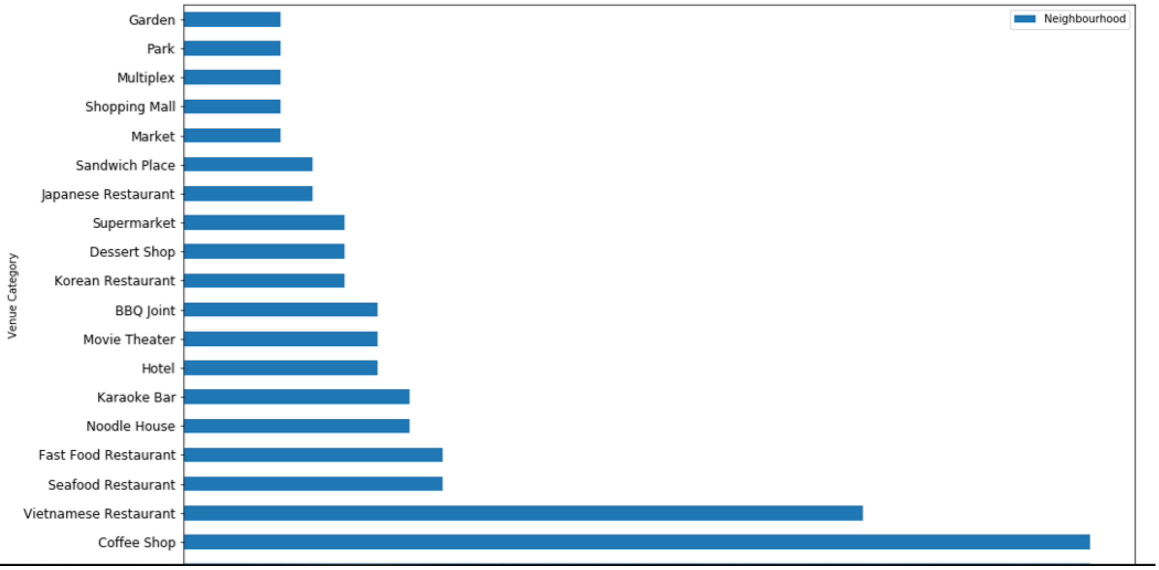
```
In [30]: HN_DongDa_venues = Hanoi_venues.groupby('Venue Category').count()
```

```
In [31]: HN_DongDa_venues = HN_DongDa_venues.reindex(columns=['Neighbourhood'])
         HN_DongDa_venues = HN_DongDa_venues.sort_values(by=['Neighbourhood'], ascending=False).head(20)
         HN_DongDa_venues.to_csv('HN_DongDa_venues.csv')
```

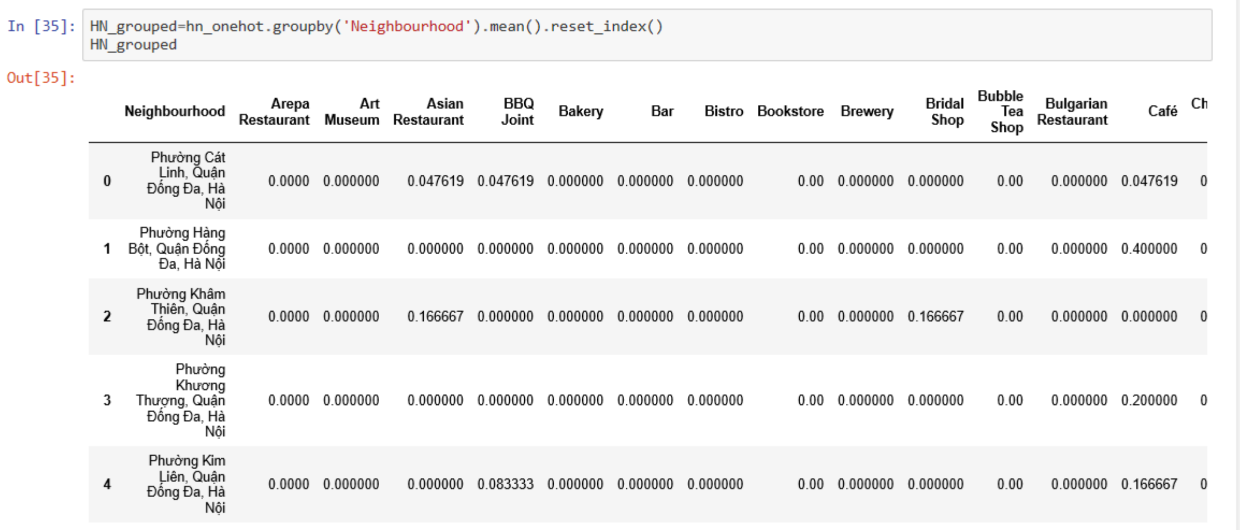
Draw char top Venue Category common



```
In [33]: HN_venues_bar.plot.barh(x='Venue Category',fontsize = 12, figsize=(16, 10),stacked=True);
```



Next, let's group rows by neighborhood and by taking the mean of the frequency of occurrence of each category

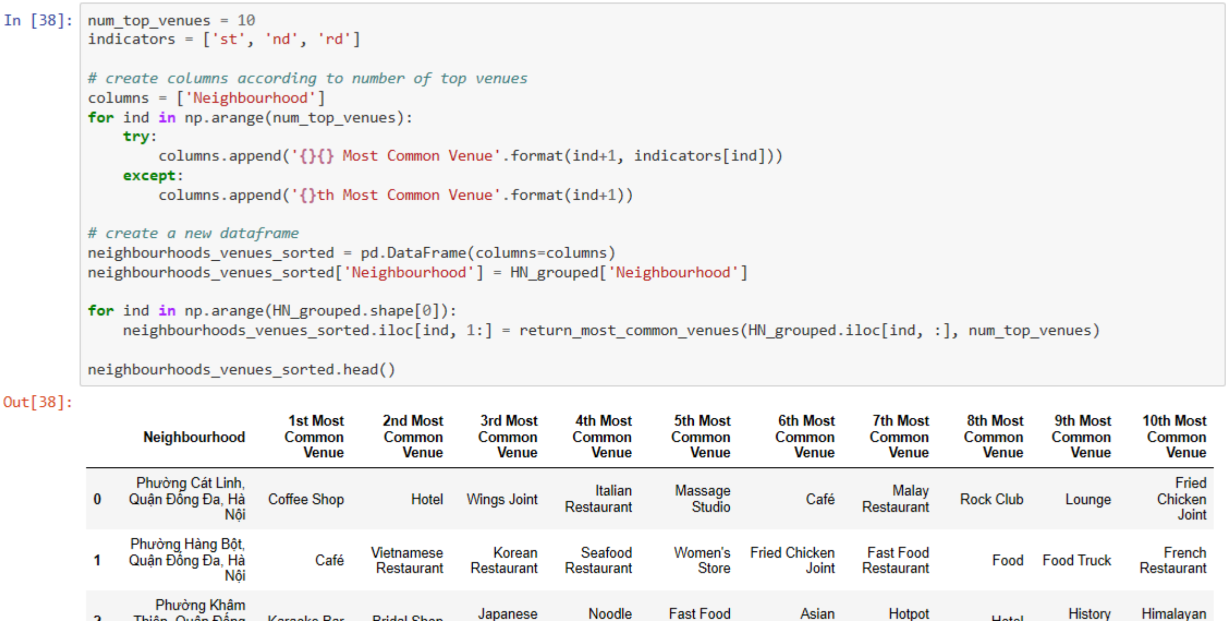


```
In [35]: HN_grouped=hn_onehot.groupby('Neighbourhood').mean().reset_index()
         HN_grouped
```

Out[35]:

	Neighbourhood	Arepa Restaurant	Art Museum	Asian Restaurant	BBQ Joint	Bakery	Bar	Bistro	Bookstore	Brewery	Bridal Shop	Bubble Tea Shop	Bulgarian Restaurant	Café	Ch
0	Phường Cát Linh, Quận Đống Đa, Hà Nội	0.0000	0.000000	0.047619	0.047619	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.047619	0
1	Phường Hàng Bột, Quận Đống Đa, Hà Nội	0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.400000	0
2	Phường Khâm Thiên, Quận Đống Đa, Hà Nội	0.0000	0.000000	0.166667	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.166667	0.00	0.000000	0.000000	0
3	Phường Khương Thượng, Quận Đống Đa, Hà Nội	0.0000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.200000	0
4	Phường Kim Liên, Quận Đống Đa, Hà Nội	0.0000	0.000000	0.000000	0.083333	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.00	0.000000	0.166667	0

Create the new dataframe and display the top 10 venues for each neighborhood



```
In [38]: num_top_venues = 10
         indicators = ['st', 'nd', 'rd']

         # create columns according to number of top venues
         columns = ['Neighbourhood']
         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))

         # create a new dataframe
         neighbourhoods_venues_sorted = pd.DataFrame(columns=columns)
         neighbourhoods_venues_sorted['Neighbourhood'] = HN_grouped['Neighbourhood']

         for ind in np.arange(HN_grouped.shape[0]):
             neighbourhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(HN_grouped.iloc[ind, :], num_top_venues)

         neighbourhoods_venues_sorted.head()
```

Out[38]:

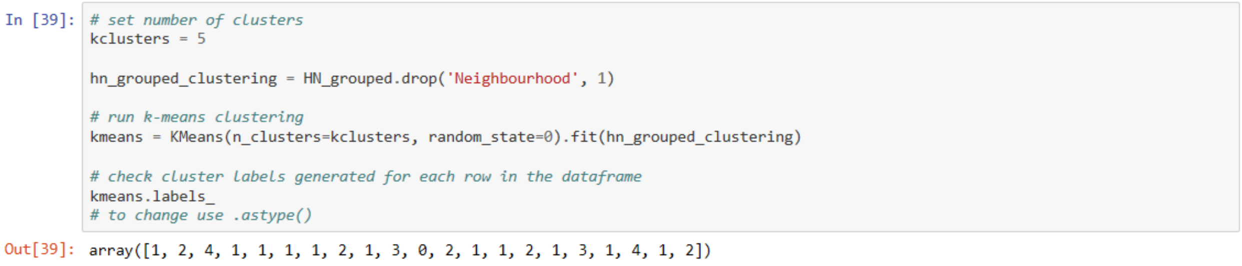
	Neighbourhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Phường Cát Linh, Quận Đống Đa, Hà Nội	Coffee Shop	Hotel	Wings Joint	Italian Restaurant	Massage Studio	Café	Malay Restaurant	Rock Club	Lounge	Fried Chicken Joint
1	Phường Hàng Bột, Quận Đống Đa, Hà Nội	Café	Vietnamese Restaurant	Korean Restaurant	Seafood Restaurant	Women's Store	Fried Chicken Joint	Fast Food Restaurant	Food	Food Truck	French Restaurant
2	Phường Khâm Thiên, Quận Đống	Karaoke Bar	Bridal Shop	Japanese	Noodle	Fast Food	Asian	Hotpot	Hotel	History	Himalayan

3. Methodology

After data acquisition and cleaning, this project applies **K-mean clustering unsupervised machine learning algorithm** to cluster the venues based on a list of locations for different types of food and beverage service points such as bars, cafes, Chinese restaurants, Vietnamese restaurants, Seafood restaurants, etc. This would give a better understanding of the similarities and dissimilarities between the chosen neighborhoods to retrieve more insights.

Analyze Each Neighborhood, group rows by neighborhood and by taking the mean of the frequency of occurrence of each category. Next, create the new data frame and display the top 10 venues for each neighborhood.

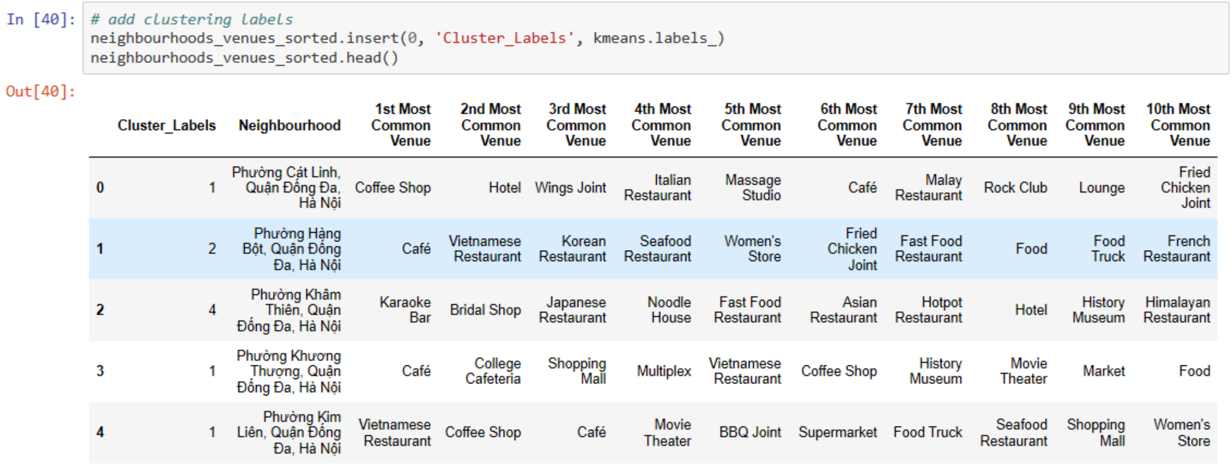
Then use the Kmean algorithm from the sklearn library to divide it into 5 groups with similar properties. Next, assign labels from Kmean result to each neighborhood using the Pandas merge function



```
In [39]: # set number of clusters
         kclusters = 5

         hn_grouped_clustering = HN_grouped.drop('Neighbourhood', 1)

         # run k-means clustering
         kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(hn_grouped_clustering)

         # check cluster labels generated for each row in the dataframe
         kmeans.labels_
         # to change use .astype()

Out[39]: array([1, 2, 4, 1, 1, 1, 1, 2, 1, 3, 0, 2, 1, 1, 2, 1, 3, 1, 4, 1, 2])
```

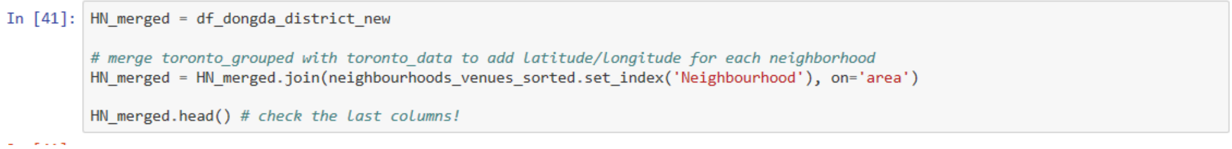
4. Analysis

Let's create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.



```
In [40]: # add clustering labels
         neighbourhoods_venues_sorted.insert(0, 'Cluster_Labels', kmeans.labels_)
         neighbourhoods_venues_sorted.head()

Out[40]:
```

	Cluster_Labels	Neighbourhood	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	1	Phường Cát Linh, Quận Đống Đa, Hà Nội	Coffee Shop	Hotel	Wings Joint	Italian Restaurant	Massage Studio	Café	Malay Restaurant	Rock Club	Lounge	Fried Chicken Joint
1	2	Phường Hàng Bột, Quận Đống Đa, Hà Nội	Café	Vietnamese Restaurant	Korean Restaurant	Seafood Restaurant	Women's Store	Fried Chicken Joint	Fast Food Restaurant	Food	Food Truck	French Restaurant
2	4	Phường Khâm Thiên, Quận Đống Đa, Hà Nội	Karaoke Bar	Bridal Shop	Japanese Restaurant	Noodle House	Fast Food Restaurant	Asian Restaurant	Hotpot Restaurant	Hotel	History Museum	Himalayan Restaurant
3	1	Phường Khương Thượng, Quận Đống Đa, Hà Nội	Café	College Cafeteria	Shopping Mall	Multiplex	Vietnamese Restaurant	Coffee Shop	History Museum	Movie Theater	Market	Food
4	1	Phường Kim Liên, Quận Đống Đa, Hà Nội	Vietnamese Restaurant	Coffee Shop	Café	Movie Theater	BBQ Joint	Supermarket	Food Truck	Seafood Restaurant	Shopping Mall	Women's Store



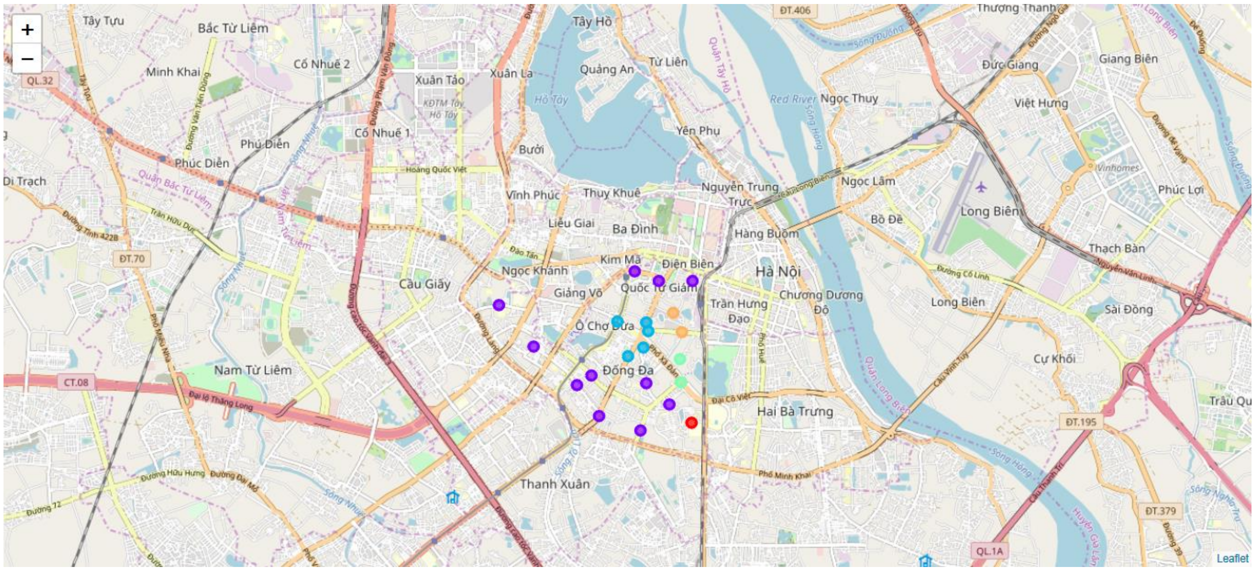
```
In [41]: HN_merged = df_dongda_district_new

         # merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
         HN_merged = HN_merged.join(neighbourhoods_venues_sorted.set_index('Neighbourhood'), on='area')

         HN_merged.head() # check the last columns!
```

Create map cluster

In [44]:

```
# create map
map_clusters = folium.Map(location=[lat_HN, long_HN], zoom_start=13)

# set color scheme for the clusters
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(HN_merged['Latitude'], HN_merged['Longitude'], HN_merged['area'], HN_merged['Cluster_Label
s']):
    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
```

Out[44]:



5. Results and Discussion

Cluster 1

In [45]:

```
HN_merged.loc[HN_merged['Cluster_Labels'] == 0, HN_merged.columns[[0] + list(range(5, HN_merged.shape[1]))]]
```

Out[45]:

	ward	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
18	Phường Phương Mai	0	Karaoke Bar	Ice Cream Shop	BBQ Joint	Coffee Shop	Garden	Fast Food Restaurant	Food	Food Truck	French Restaurant	Fried Chicken Joint

Cluster 2

In [46]:

```
HN_merged.loc[HN_merged['Cluster_Labels'] == 1, HN_merged.columns[[0] + list(range(5, HN_merged.shape[1]))]]
```

Out[46]:

	ward	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	Phường Cát Linh	1	Coffee Shop	Hotel	Wings Joint	Italian Restaurant	Massage Studio	Café	Malay Restaurant	Rock Club	Lounge	Fried Chicken Joint
1	Phường Văn	1	Vietnamese Restaurant	Coffee Shop	Hotel	Café	Sandwich Place	Malay Restaurant	Food	Dessert Shop	Confucian Temple	Park

After reviewing the data of each cluster, I have some discussions:

- At Cluster 1 most common venue is Karaoke Bar. Cafe shop and Restaurant is only ranked 4 to 10, so it is possible to open a cafe in Cluster 1
- At Cluster 2, 3 ,4 focus mainly on Vietnamese restaurants, Cafe, so need to be careful when you intend to open a Vietnamese restaurant or cafe
- Cluster 5, there is no coffee shop, so you can rest assured that you can open a coffee shop without much competition.

6. Conclusion

Finally, I have got a small glimpse of how real-life data-science projects look like. I used various types of APIs to collect data, used the Pandas library to eliminate redundant data, used it, and used Python libraries to draw graphs, using unsupervised machine learning algorithms to group data into similar characteristics. From that it is possible to discover the information that is hidden in it, making it easier to make decisions such as where to open a restaurant or a cafe is appropriate and less competitive

7. Final Notes

This is my assignment: a part of the IBM Data Science Course on Coursera.

The full project Jupiter Notebook from data scraping to preprocessing to results here: https://github.com/TC1894/Coursera_Capstone/blob/master/Battle-of-Neighborhoods-in-DongDa-District-HaNoi.ipynb