## 1. Introduction: Business Problem

My sister – in - law wanted to open a restaurant or a cafe in District 2, Ho Chi Minh, but she didn't know where to open with little competition. This data analysis article will clarify and may help him with some useful information for her 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 **District 2, Ha Noi**, **Viet nam**.

We will use our data science powers to generate a few most promising 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 District 2, list of districts, wards of District 2, Ho Chi Minh from the following URL: http://www.pso.hochiminhcity.gov.vn/web/guest/danhmucthongke-danhmuctinhthanhpho http://www.pso.hochiminhcity.gov.vn/web/guest/danhmucthongke-danhmucphuongxa or file data xls from the following: https://github.com/chaudb39/Capstone_Cousera/blob/e4b872054271da617fcb10566faa3ea8966df29a/HCM_DISTRICT2.xlsx

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

**Google map API**

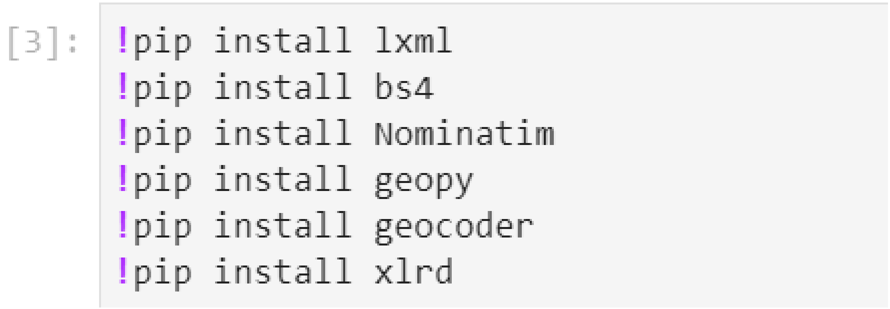
This project would use Google Map API Geocoder to get the Latitude and Longitude of each area

**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.

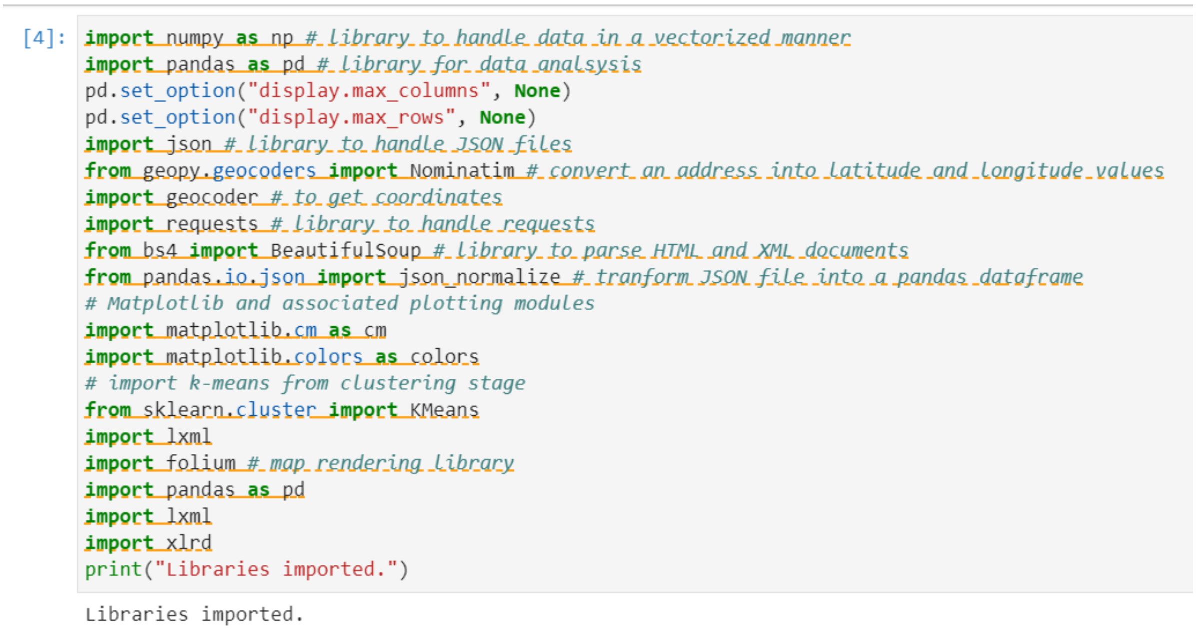
**Step by step following**

**Install packages**



```
!pip install lxml
!pip install bs4
!pip install Nominatim
!pip install geopy
!pip install geocoder
!pip install xlrd
```

**2.1. Load necessary library**



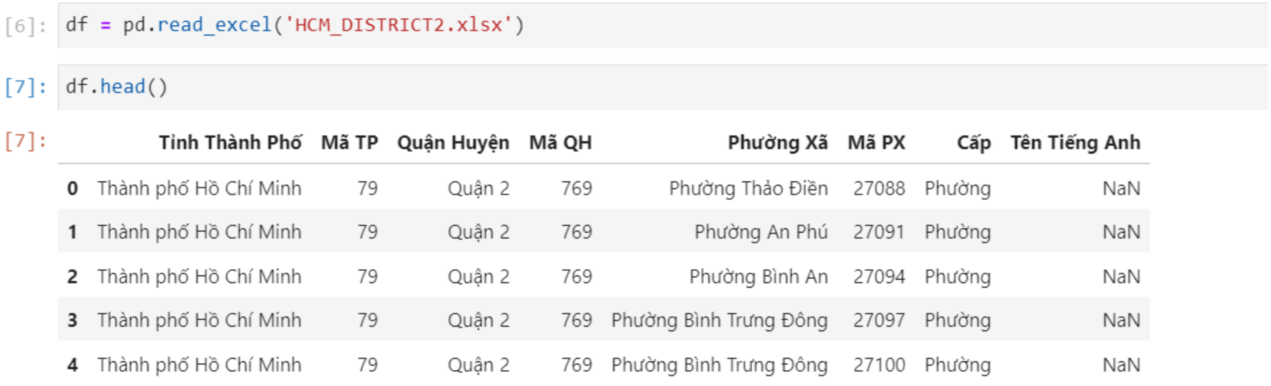
```
[4]: import numpy as np # library to handle data in a vectorized manner
     import pandas as pd # library for data analsysis
     pd.set_option("display.max_columns", None)
     pd.set_option("display.max_rows", None)
     import json # library to handle JSON files
     from geopy.geocoders import 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.")

     Libraries imported.
```

## 2.2. Get Data District 2

http://www.pso.hochiminhcity.gov.vn/web/guest/danhmucthongke-danhmuctinhthanhpho

http://www.pso.hochiminhcity.gov.vn/web/guest/danhmucthongke-danhmucphuongxa or file data xls from the following: https://github.com/chaudb39/Capstone_Cousera/blob/e4b872054271da617fcb10566faa3ea8966df29a/HCM_DISTRICT2.xlsx

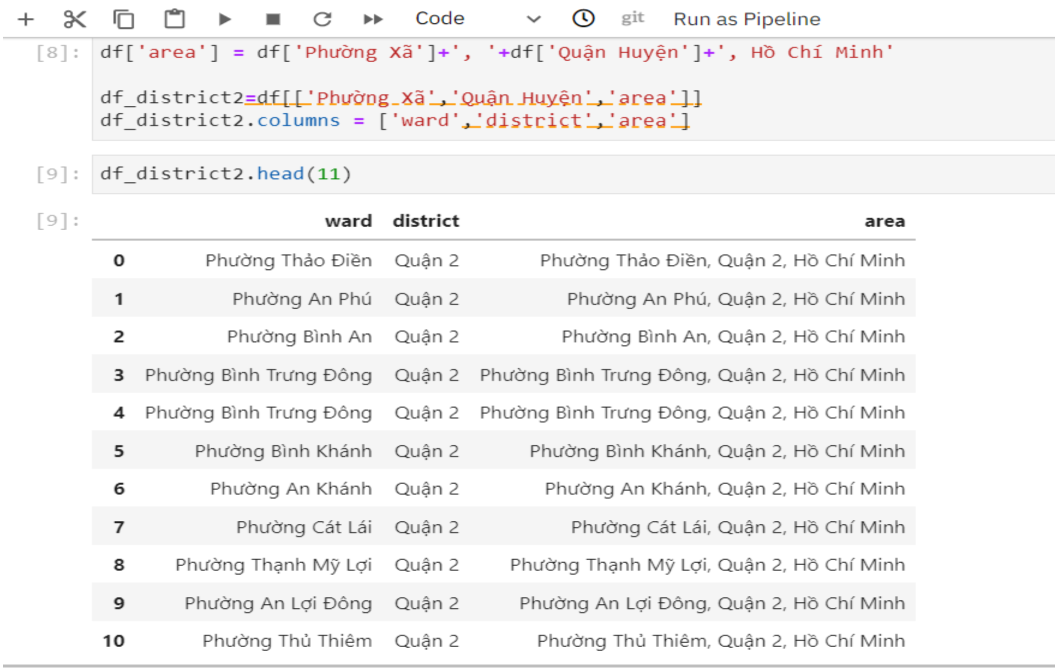
## 2.3. Load file excel districts, wards of VietNam



```
[6]: df = pd.read_excel('HCM_DISTRICT2.xlsx')

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

[7]:

| | Tỉnh Thành Phố | Mã TP | Quận Huyện | Mã QH | Phường Xã | Mã PX | Cấp | Tên Tiếng Anh |
|---|---|---|---|---|---|---|---|---|
| 0 | Thành phố Hồ Chí Minh | 79 | Quận 2 | 769 | Phường Thảo Điền | 27088 | Phường | NaN |
| 1 | Thành phố Hồ Chí Minh | 79 | Quận 2 | 769 | Phường An Phú | 27091 | Phường | NaN |
| 2 | Thành phố Hồ Chí Minh | 79 | Quận 2 | 769 | Phường Bình An | 27094 | Phường | NaN |
| 3 | Thành phố Hồ Chí Minh | 79 | Quận 2 | 769 | Phường Bình Trưng Đông | 27097 | Phường | NaN |
| 4 | Thành phố Hồ Chí Minh | 79 | Quận 2 | 769 | Phường Bình Trưng Đông | 27100 | Phường | NaN |



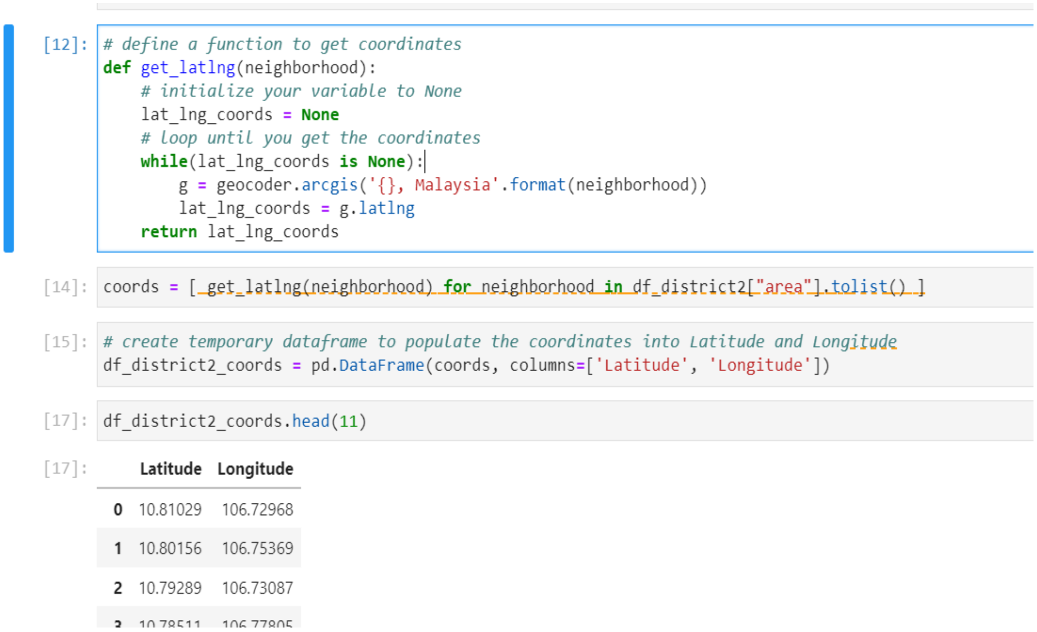
```
[8]: df['area'] = df['Phường Xã']+', '+df['Quận Huyện']+', Hồ Chí Minh'

     df_district2=df[['Phường Xã','Quận Huyện','area']]
     df_district2.columns = ['ward','district','area']
```

```
[9]: df_district2.head(11)
```

[9]:

| | ward | district | area |
|---|---|---|---|
| 0 | Phường Thảo Điền | Quận 2 | Phường Thảo Điền, Quận 2, Hồ Chí Minh |
| 1 | Phường An Phú | Quận 2 | Phường An Phú, Quận 2, Hồ Chí Minh |
| 2 | Phường Bình An | Quận 2 | Phường Bình An, Quận 2, Hồ Chí Minh |
| 3 | Phường Bình Trưng Đông | Quận 2 | Phường Bình Trưng Đông, Quận 2, Hồ Chí Minh |
| 4 | Phường Bình Trưng Đông | Quận 2 | Phường Bình Trưng Đông, Quận 2, Hồ Chí Minh |
| 5 | Phường Bình Khánh | Quận 2 | Phường Bình Khánh, Quận 2, Hồ Chí Minh |
| 6 | Phường An Khánh | Quận 2 | Phường An Khánh, Quận 2, Hồ Chí Minh |
| 7 | Phường Cát Lái | Quận 2 | Phường Cát Lái, Quận 2, Hồ Chí Minh |
| 8 | Phường Thạnh Mỹ Lợi | Quận 2 | Phường Thạnh Mỹ Lợi, Quận 2, Hồ Chí Minh |
| 9 | Phường An Lợi Đông | Quận 2 | Phường An Lợi Đông, Quận 2, Hồ Chí Minh |
| 10 | Phường Thủ Thiêm | Quận 2 | Phường Thủ Thiêm, Quận 2, Hồ Chí Minh |

## 2.4. Add latitude, longitude by call Google Geocode API



```
[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
```

```
[14]: coords = [ get_latlng(neighborhood) for neighborhood in df_district2["area"].tolist() ]
```

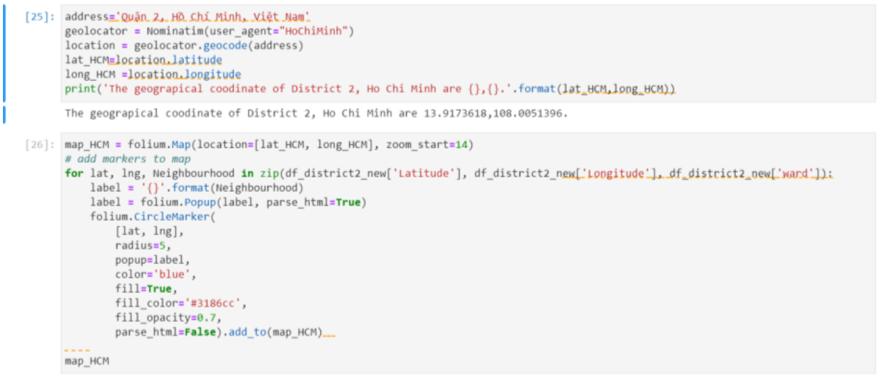
```
[15]: # create temporary dataframe to populate the coordinates into Latitude and Longitude
      df_district2_coords = pd.DataFrame(coords, columns=['Latitude', 'Longitude'])
```

```
[17]: df_district2_coords.head(11)
```

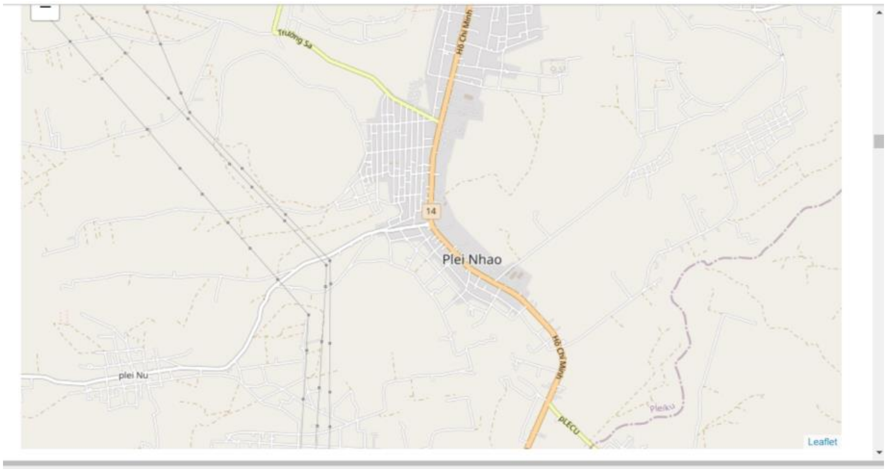
[17]:

| | Latitude | Longitude |
|---|---|---|
| 0 | 10.81029 | 106.72968 |
| 1 | 10.80156 | 106.75369 |
| 2 | 10.79289 | 106.73087 |
| 3 | 10.78511 | 106.77805 |

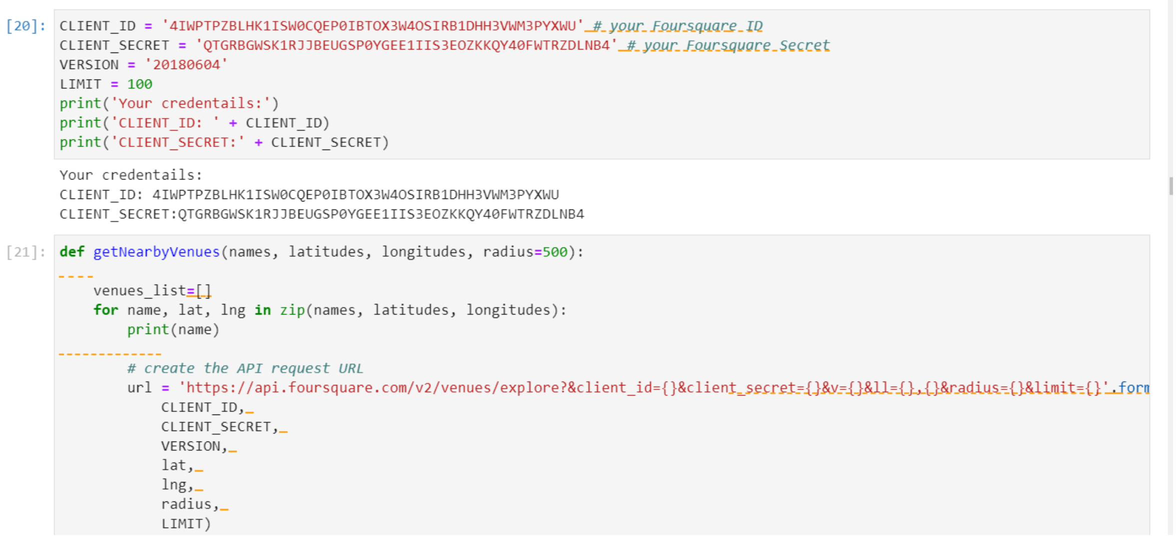
## 2.5. Create a map of  District 2 – Ho Chi Minh City with neighborhoods superimposed on top



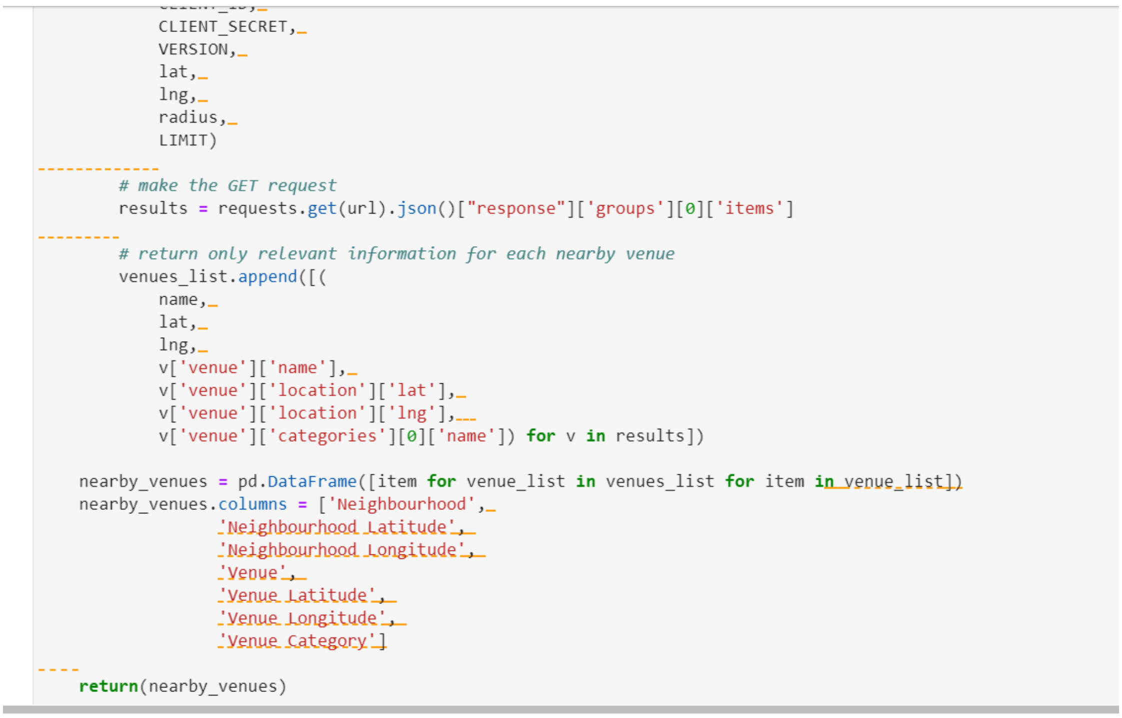
```
[25]: address='Quận 2, Hồ Chí Minh, Việt Nam'
      geolocator = Nominatim(user_agent="HoChiMinh")
      location = geolocator.geocode(address)
      lat_HCM=location.latitude
      long_HCM =location.longitude
      print('The geograpical coodinate of District 2, Ho Chi Minh are {},{}.'.format(lat_HCM,long_HCM))

      The geograpical coodinate of District 2, Ho Chi Minh are 13.9173618,108.0051396.

[26]: map_HCM = folium.Map(location=[lat_HCM, long_HCM], zoom_start=14)
      # add markers to map
      for lat, lng, Neighbourhood in zip(df_district2_new['Latitude'], df_district2_new['Longitude'], df_district2_new['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_HCM)

      map_HCM
```



## 2.6. Use the Foursquare API to explore the neighborhoods

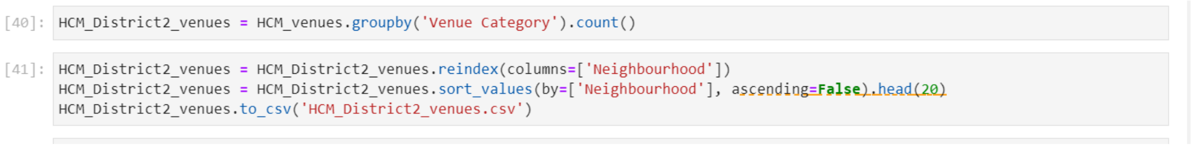


```
[20]: CLIENT_ID = '4IWPTPZBLHK1ISW0CQEP0IBTOX3W4OSIRB1DHH3VWM3PYXWU' # your Foursquare ID
      CLIENT_SECRET = 'QTGRBGWSK1RJJBEUGSP0YGEE1IIS3EOZKKQY40FWTRZDLNB4' # your Foursquare Secret
      VERSION = '20180604'
      LIMIT = 100
      print('Your credentails:')
      print('CLIENT_ID: ' + CLIENT_ID)
      print('CLIENT_SECRET:' + CLIENT_SECRET)

      Your credentails:
      CLIENT_ID: 4IWPTPZBLHK1ISW0CQEP0IBTOX3W4OSIRB1DHH3VWM3PYXWU
      CLIENT_SECRET:QTGRBGWSK1RJJBEUGSP0YGEE1IIS3EOZKKQY40FWTRZDLNB4

[21]: 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={}'.form
                  CLIENT_ID,
                  CLIENT_SECRET,
                  VERSION,
                  lat,
                  lng,
                  radius,
                  LIMIT)
```



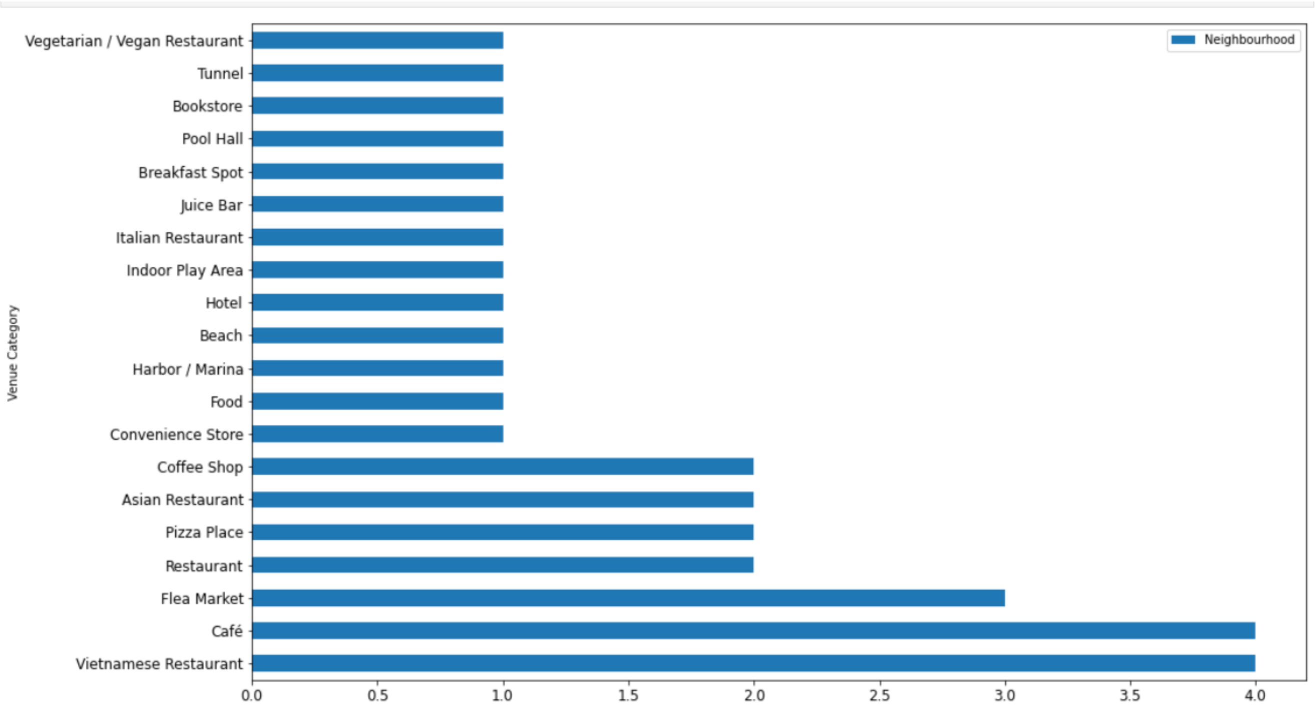
```
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # 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

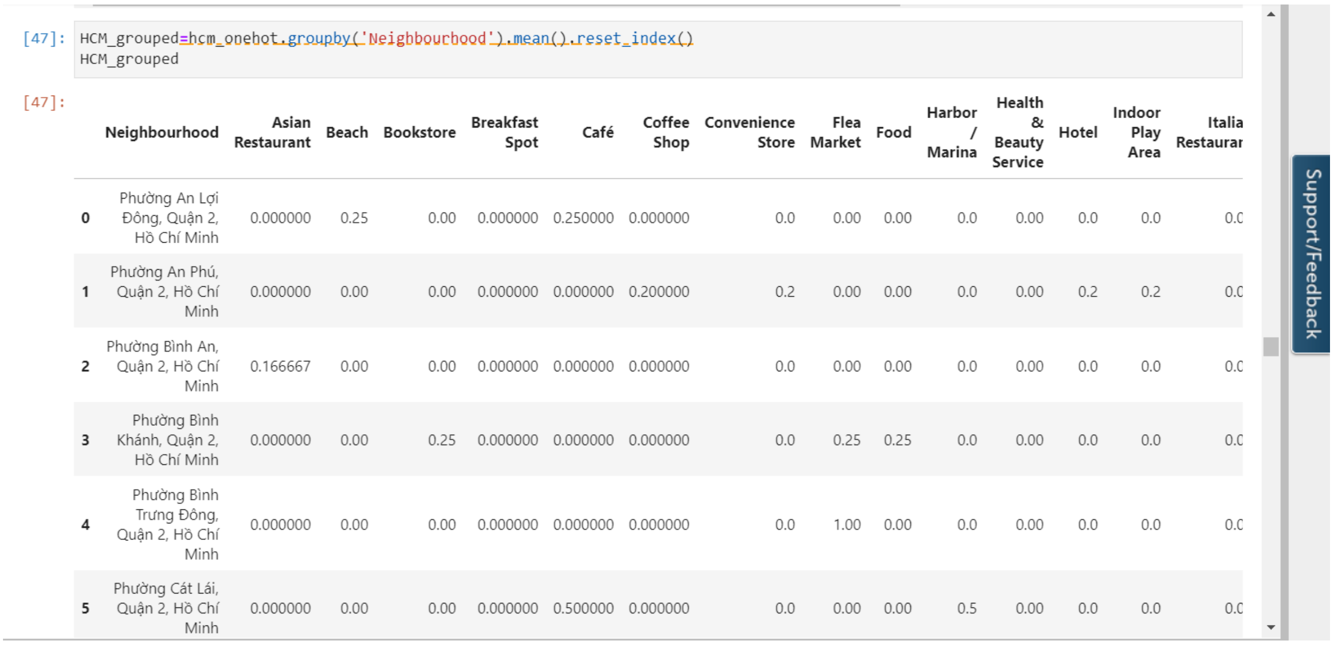


```
[40]: HCM_District2_venues = HCM_venues.groupby('Venue Category').count()

[41]: HCM_District2_venues = HCM_District2_venues.reindex(columns=['Neighbourhood'])
      HCM_District2_venues = HCM_District2_venues.sort_values(by=['Neighbourhood'], ascending=False).head(20)
      HCM_District2_venues.to_csv('HCM_District2_venues.csv')
```

## Draw char top Venue Category common



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

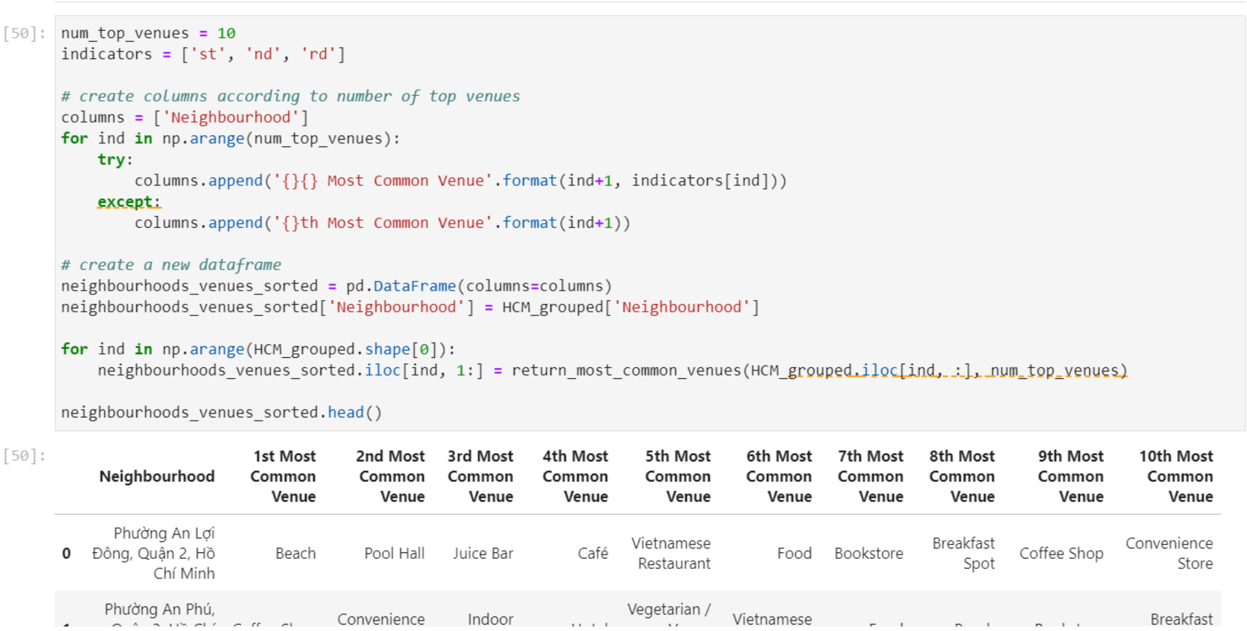


```
[47]: HCM_grouped=hcm_onehot.groupby('Neighbourhood').mean().reset_index()
      HCM_grouped
```

[47]:

| | Neighbourhood | Asian Restaurant | Beach | Bookstore | Breakfast Spot | Café | Coffee Shop | Convenience Store | Flea Market | Food | Harbor / Marina | Health & Beauty Service | Hotel | Indoor Play Area | Italia Restaurar |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Phường An Lợi Đông, Quận 2, Hồ Chí Minh | 0.000000 | 0.25 | 0.00 | 0.000000 | 0.250000 | 0.000000 | 0.0 | 0.00 | 0.00 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 |
| **1** | Phường An Phú, Quận 2, Hồ Chí Minh | 0.000000 | 0.00 | 0.00 | 0.000000 | 0.000000 | 0.200000 | 0.2 | 0.00 | 0.00 | 0.0 | 0.00 | 0.2 | 0.2 | 0.0 |
| **2** | Phường Bình An, Quận 2, Hồ Chí Minh | 0.166667 | 0.00 | 0.00 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.00 | 0.00 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 |
| **3** | Phường Bình Khánh, Quận 2, Hồ Chí Minh | 0.000000 | 0.00 | 0.25 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.25 | 0.25 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 |
| **4** | Phường Bình Trưng Đông, Quận 2, Hồ Chí Minh | 0.000000 | 0.00 | 0.00 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 1.00 | 0.00 | 0.0 | 0.00 | 0.0 | 0.0 | 0.0 |
| **5** | Phường Cát Lái, Quận 2, Hồ Chí Minh | 0.000000 | 0.00 | 0.00 | 0.000000 | 0.500000 | 0.000000 | 0.0 | 0.00 | 0.00 | 0.5 | 0.00 | 0.0 | 0.0 | 0.0 |

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



```
[50]: 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'] = HCM_grouped['Neighbourhood']

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

      neighbourhoods_venues_sorted.head()
```

[50]:

| | 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 An Lợi Đông, Quận 2, Hồ Chí Minh | Beach | Pool Hall | Juice Bar | Café | Vietnamese Restaurant | Food | Bookstore | Breakfast Spot | Coffee Shop | Convenience Store |
| 1 | Phường An Phú, | | Convenience | Indoor | | Vegetarian / | Vietnamese | | | | Breakfast |

## 3. Methodology

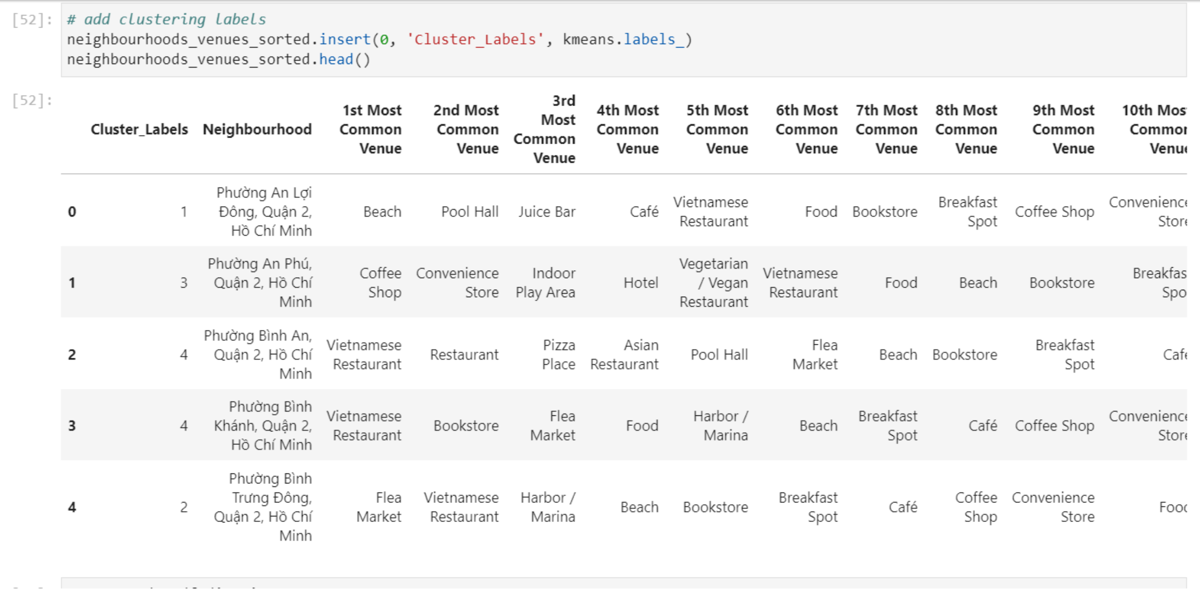
After data acquisition and cleaning, this project applies **K-mean clustering unsupervised machine learning algorithm** to cluster the venues based on a list of locations for different types of food and beverage service points such as bars, cafes, Chinese restaurants, Vietnamese restaurants, Seafood 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

## 4. Analysis

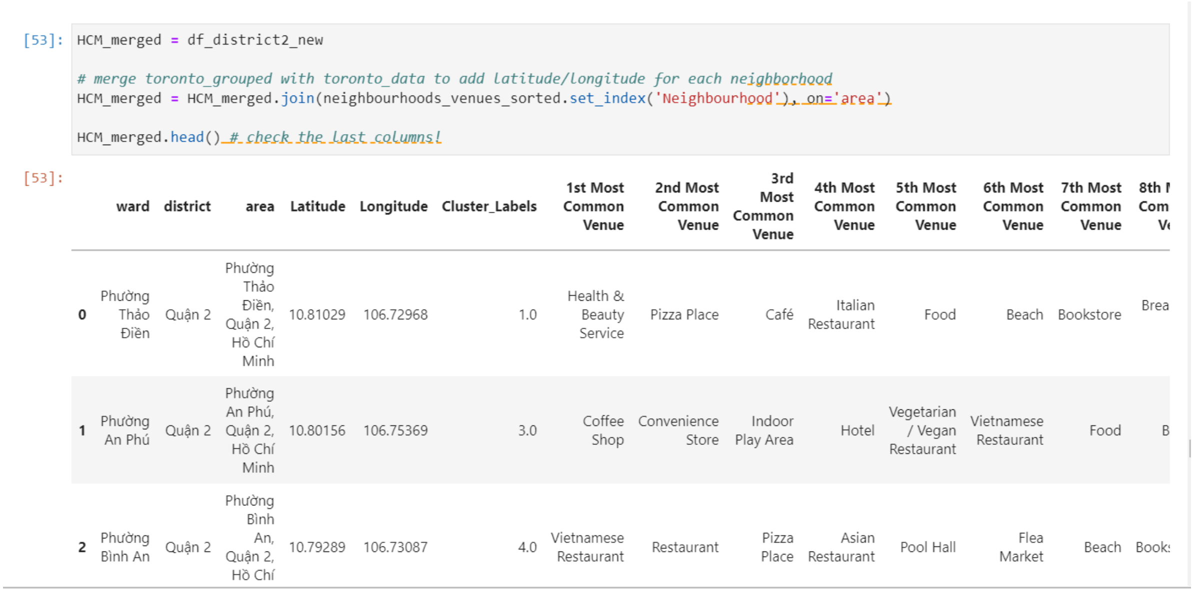
## Let’s create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.



[52]:
```
# add clustering labels
neighbourhoods_venues_sorted.insert(0, 'Cluster_Labels', kmeans.labels_)
neighbourhoods_venues_sorted.head()
```

[52]:

| | 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 Mos Commo Venu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | Phường An Lợi Đông, Quận 2, Hồ Chí Minh | Beach | Pool Hall | Juice Bar | Café | Vietnamese Restaurant | Food | Bookstore | Breakfast Spot | Coffee Shop | Convenienc Stor |
| 1 | 3 | Phường An Phú, Quận 2, Hồ Chí Minh | Coffee Shop | Convenience Store | Indoor Play Area | Hotel | Vegetarian / Vegan Restaurant | Vietnamese Restaurant | Food | Beach | Bookstore | Breakfas Spo |
| 2 | 4 | Phường Bình An, Quận 2, Hồ Chí Minh | Vietnamese Restaurant | Restaurant | Pizza Place | Asian Restaurant | Pool Hall | Flea Market | Beach | Bookstore | Breakfast Spot | Caf |
| 3 | 4 | Phường Bình Khánh, Quận 2, Hồ Chí Minh | Vietnamese Restaurant | Bookstore | Flea Market | Food | Harbor / Marina | Beach | Breakfast Spot | Café | Coffee Shop | Convenienc Stor |
| 4 | 2 | Phường Bình Trưng Đông, Quận 2, Hồ Chí Minh | Flea Market | Vietnamese Restaurant | Harbor / Marina | Beach | Bookstore | Breakfast Spot | Café | Coffee Shop | Convenience Store | Foo |

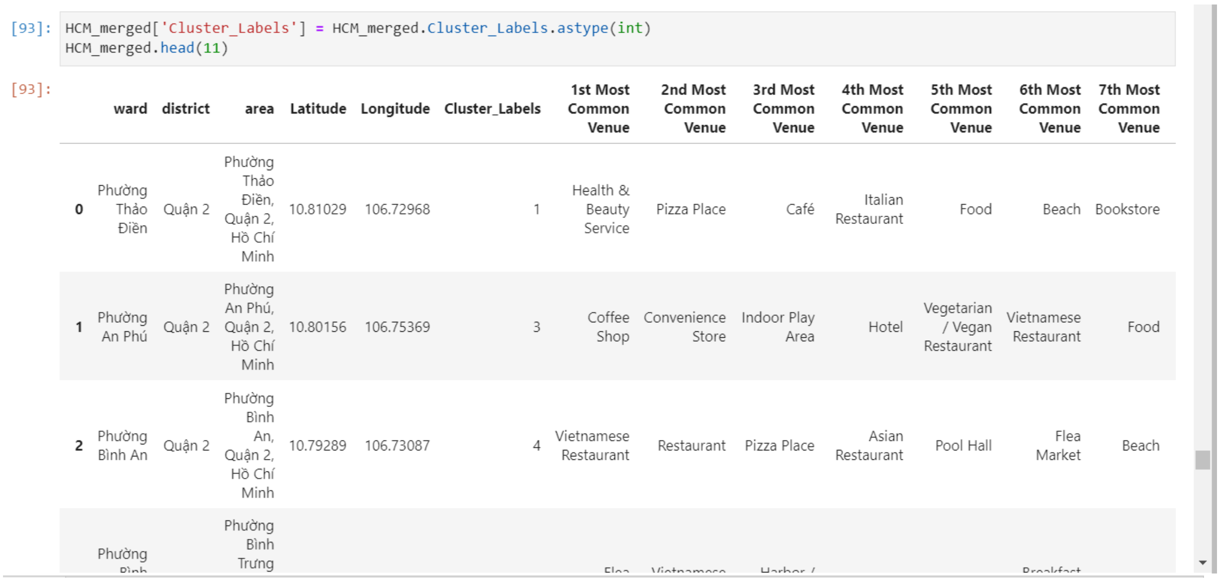


[53]:
```
HCM_merged = df_district2_new

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

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

[53]:

| | ward | district | area | Latitude | Longitude | 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 I Com Ve |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Phường Thảo Điền | Quận 2 | Phường Thảo Điền, Quận 2, Hồ Chí Minh | 10.81029 | 106.72968 | 1.0 | Health & Beauty Service | Pizza Place | Café | Italian Restaurant | Food | Beach | Bookstore | Brea |
| 1 | Phường An Phú | Quận 2 | Phường An Phú, Quận 2, Hồ Chí Minh | 10.80156 | 106.75369 | 3.0 | Coffee Shop | Convenience Store | Indoor Play Area | Hotel | Vegetarian / Vegan Restaurant | Vietnamese Restaurant | Food | B |
| 2 | Phường Bình An | Quận 2 | Phường Bình An, Quận 2, Hồ Chí | 10.79289 | 106.73087 | 4.0 | Vietnamese Restaurant | Restaurant | Pizza Place | Asian Restaurant | Pool Hall | Flea Market | Beach | Book: |

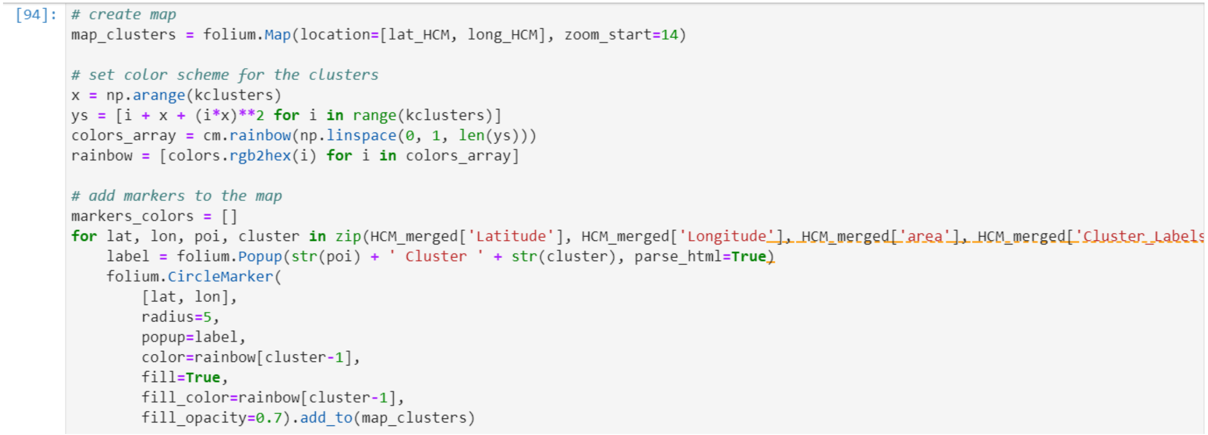
## Create map cluster



```
[93]: HCM_merged['Cluster_Labels'] = HCM_merged.Cluster_Labels.astype(int)
      HCM_merged.head(11)
```

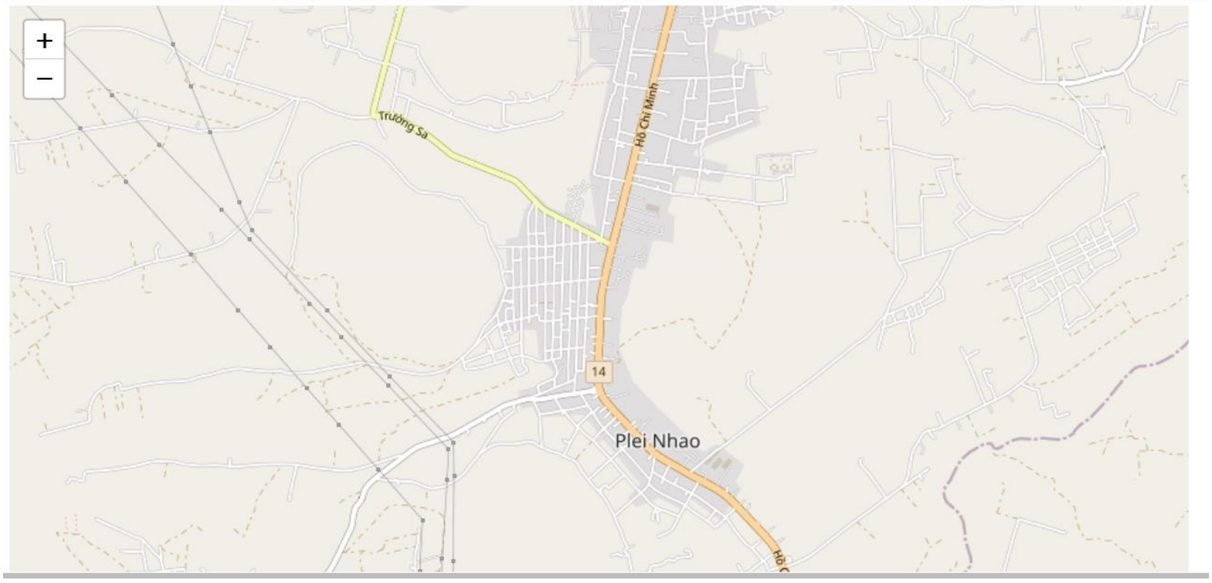
[93]:

| | ward | district | area | Latitude | Longitude | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Phường Thảo Điền | Quận 2 | Phường Thảo Điền, Quận 2, Hồ Chí Minh | 10.81029 | 106.72968 | 1 | Health & Beauty Service | Pizza Place | Café | Italian Restaurant | Food | Beach | Bookstore |
| **1** | Phường An Phú | Quận 2 | Phường An Phú, Quận 2, Hồ Chí Minh | 10.80156 | 106.75369 | 3 | Coffee Shop | Convenience Store | Indoor Play Area | Hotel | Vegetarian / Vegan Restaurant | Vietnamese Restaurant | Food |
| **2** | Phường Bình An | Quận 2 | Phường Bình An, Quận 2, Hồ Chí Minh | 10.79289 | 106.73087 | 4 | Vietnamese Restaurant | Restaurant | Pizza Place | Asian Restaurant | Pool Hall | Flea Market | Beach |
| | Phường Bình | | Phường Bình Trưng | | | | Flea | Vietnamese | Harbor / | | | Breakfast | |



```
[94]: # create map
      map_clusters = folium.Map(location=[lat_HCM, long_HCM], zoom_start=14)

      # 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(HCM_merged['Latitude'], HCM_merged['Longitude'], HCM_merged['area'], HCM_merged['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
```



## 5. 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

## 6. 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/chaudb39/Capstone_Cousera/blob/e4b872054271da617fcb10566faa3ea8966df29a/CourseraCapstone(Week2).ipynb

Dinh Bao Chau