Capstone Project - The Battle of Neighborhoods

(Week 1 & 2)

1. Introduction/Business Understanding

1.1 Description of the problem

The business problem we are currently posing is: Ahead of the Tokyo 2020 Summer Olympic and Paralympics Games, how could we provide support to different visitors to list and visualize Tokyo districts that fit their needs in term of culinary/ food venues.

1.2 Discussion of the background

With nearly 44 million inhabitants and the highest metropolitan GDP in the world, an estimated 600,000 overseas visitors are expected to flock to the Japanese capital and surrounding regions to attend this upcoming event.

Tokyo is well-known as the restaurant capital of the world, with over 160,000 places to choose from around it's 23 districts.
I believe it's difficult for a travelers, especially restaurant-goers, to make a choice from among many options since there is also too much information on the web because everybody's got their own take of where to go and it's all so fragmented that you have to assemble it yourself especially if you're wanting non-touristy recommendations.

So, how could we leverage Foursquare location data and machine learning to help us make decision and find appropriate neighborhoods? This is the problem I would like to address in this capstone project taking Tokyo as an example. In this project, I am going to use Foursquare location data and clustering methods to group the districts to different group by their restaurant venues information.

Figure 1 Tokyo 2020 Olympics and Paralympics mascots - Image by @Tokyo 2020

2. Data Requirements

For this project we need following data:

- **Tokyo data that contains list districts (Wards) along with their latitude and longitude.**
  Datasource : https://en.wikipedia.org/wiki/Special_wards_of_Tokyo#List_of_special_wards
  Description: We will Scrap Tokyo districts (Wards) Table from Wikipedia and get the coordinates of these 23 major districts using geocoder class of Geopy client.

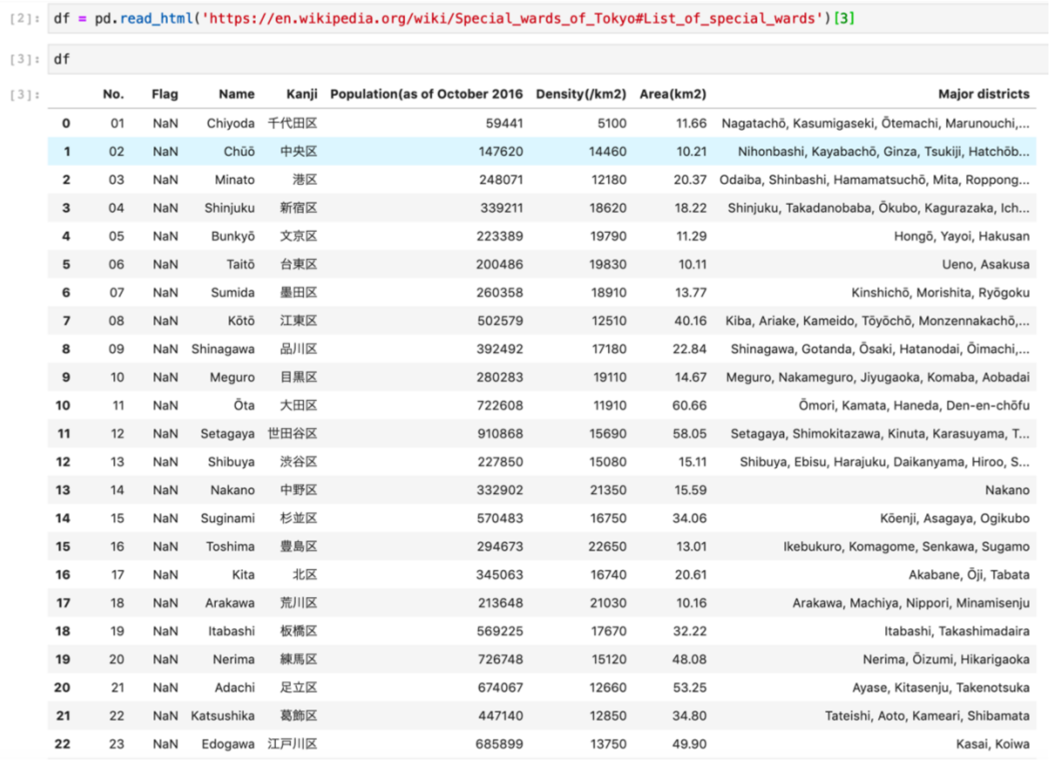
- **Restaurants in each neighborhood of Tokyo:**
  Data source: Foursquare APIs
  Description : By using this API we will get all the venues in each neighborhood. We can filter these venues to get only restaurants.

3. Methodology

3.1 Data Preparation

3.1.1 Scraping Tokyo Wards Table from Wikipedia

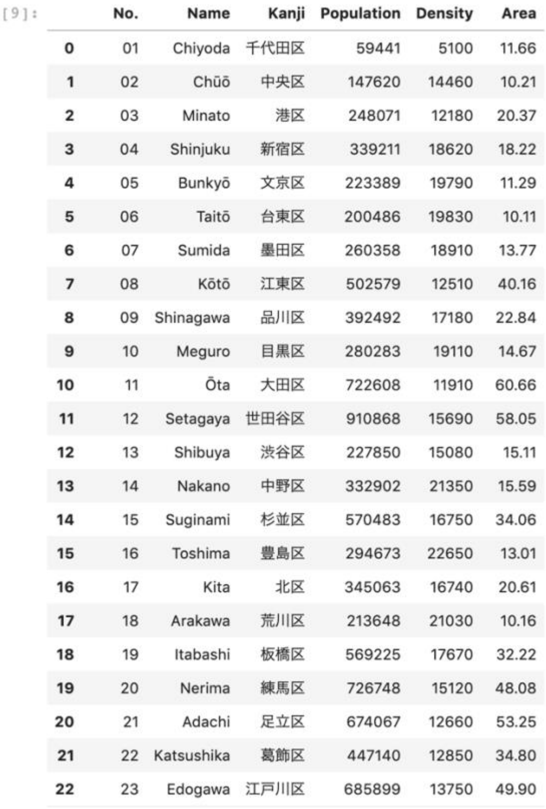
I first make use of Special Wards of Tokyo page from Wiki to scrap the table to create a data-frame. For this, I’ve used pandas to transform the data in the table on the Wikipedia page into a dataframe containing name of the 23 wards of Tokyo, Area, population. We start as below —



```
[2]: df = pd.read_html('https://en.wikipedia.org/wiki/Special_wards_of_Tokyo#List_of_special_wards')[3]

[3]: df

[3]:
```

	No.	Flag	Name	Kanji	Population(as of October 2016	Density(/km2)	Area(km2)	Major districts
0	01	NaN	Chiyoda	千代田区	59441	5100	11.66	Nagatachō, Kasumigaseki, Ōtemachi, Marunouchi,...
1	02	NaN	Chūō	中央区	147620	14460	10.21	Nihonbashi, Kayabachō, Ginza, Tsukiji, Hatchōb...
2	03	NaN	Minato	港区	248071	12180	20.37	Odaiba, Shinbashi, Hamamatsuchō, Mita, Roppong...
3	04	NaN	Shinjuku	新宿区	339211	18620	18.22	Shinjuku, Takadanobaba, Ōkubo, Kagurazaka, Ich...
4	05	NaN	Bunkyō	文京区	223389	19790	11.29	Hongō, Yayoi, Hakusan
5	06	NaN	Taitō	台東区	200486	19830	10.11	Ueno, Asakusa
6	07	NaN	Sumida	墨田区	260358	18910	13.77	Kinshichō, Morishita, Ryōgoku
7	08	NaN	Kōtō	江東区	502579	12510	40.16	Kiba, Ariake, Kameido, Tōyōchō, Monzennakachō,...
8	09	NaN	Shinagawa	品川区	392492	17180	22.84	Shinagawa, Gotanda, Ōsaki, Hatanodai, Ōimachi,...
9	10	NaN	Meguro	目黒区	280283	19110	14.67	Meguro, Nakameguro, Jiyugaoka, Komaba, Aobadai
10	11	NaN	Ōta	大田区	722608	11910	60.66	Ōmori, Kamata, Haneda, Den-en-chōfu
11	12	NaN	Setagaya	世田谷区	910868	15690	58.05	Setagaya, Shimokitazawa, Kinuta, Karasuyama, T...
12	13	NaN	Shibuya	渋谷区	227850	15080	15.11	Shibuya, Ebisu, Harajuku, Daikanyama, Hiroo, S...
13	14	NaN	Nakano	中野区	332902	21350	15.59	Nakano
14	15	NaN	Suginami	杉並区	570483	16750	34.06	Kōenji, Asagaya, Ogikubo
15	16	NaN	Toshima	豊島区	294673	22650	13.01	Ikebukuro, Komagome, Senkawa, Sugamo
16	17	NaN	Kita	北区	345063	16740	20.61	Akabane, Ōji, Tabata
17	18	NaN	Arakawa	荒川区	213648	21030	10.16	Arakawa, Machiya, Nippori, Minamisenju
18	19	NaN	Itabashi	板橋区	569225	17670	32.22	Itabashi, Takashimadaira
19	20	NaN	Nerima	練馬区	726748	15120	48.08	Nerima, Ōizumi, Hikarigaoka
20	21	NaN	Adachi	足立区	674067	12660	53.25	Ayase, Kitasenju, Takenotsuka
21	22	NaN	Katsushika	葛飾区	447140	12850	34.80	Tateishi, Aoto, Kameari, Shibamata
22	23	NaN	Edogawa	江戸川区	685899	13750	49.90	Kasai, Koiwa

After little manipulation, the data-frame is obtained as below

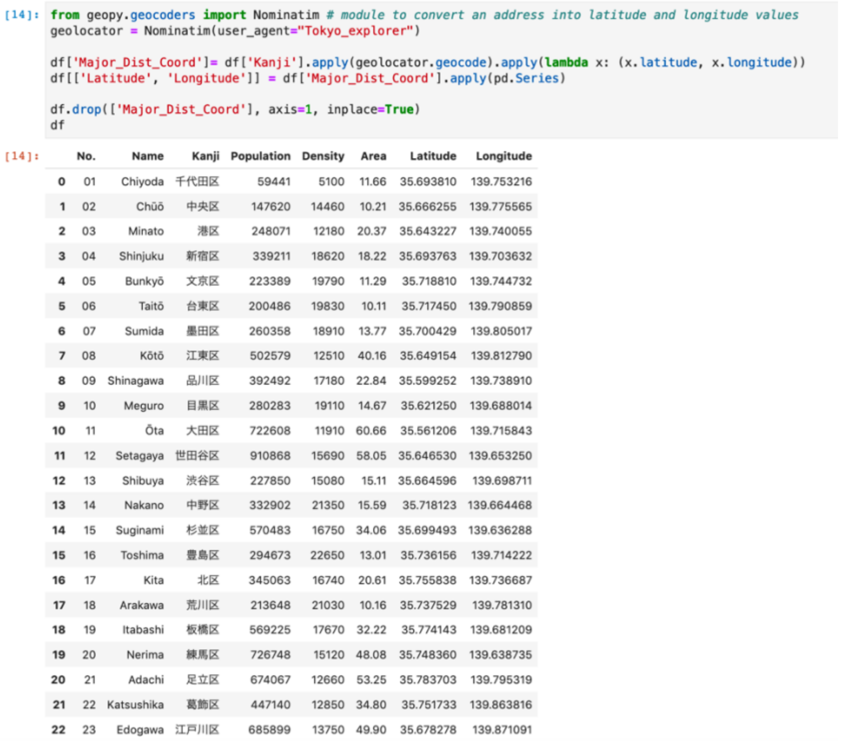


[9]:

	No.	Name	Kanji	Population	Density	Area
0	01	Chiyoda	千代田区	59441	5100	11.66
1	02	Chūō	中央区	147620	14460	10.21
2	03	Minato	港区	248071	12180	20.37
3	04	Shinjuku	新宿区	339211	18620	18.22
4	05	Bunkyō	文京区	223389	19790	11.29
5	06	Taitō	台東区	200486	19830	10.11
6	07	Sumida	墨田区	260358	18910	13.77
7	08	Kōtō	江東区	502579	12510	40.16
8	09	Shinagawa	品川区	392492	17180	22.84
9	10	Meguro	目黒区	280283	19110	14.67
10	11	Ōta	大田区	722608	11910	60.66
11	12	Setagaya	世田谷区	910868	15690	58.05
12	13	Shibuya	渋谷区	227850	15080	15.11
13	14	Nakano	中野区	332902	21350	15.59
14	15	Suginami	杉並区	570483	16750	34.06
15	16	Toshima	豊島区	294673	22650	13.01
16	17	Kita	北区	345063	16740	20.61
17	18	Arakawa	荒川区	213648	21030	10.16
18	19	Itabashi	板橋区	569225	17670	32.22
19	20	Nerima	練馬区	726748	15120	48.08
20	21	Adachi	足立区	674067	12660	53.25
21	22	Katsushika	葛飾区	447140	12850	34.80
22	23	Edogawa	江戸川区	685899	13750	49.90

3.1.2 Getting Coordinates of Major Districts : Geopy Client

Next objective is to get the coordinates of these 23 major districts using geocoder class of Geopy client as follow:



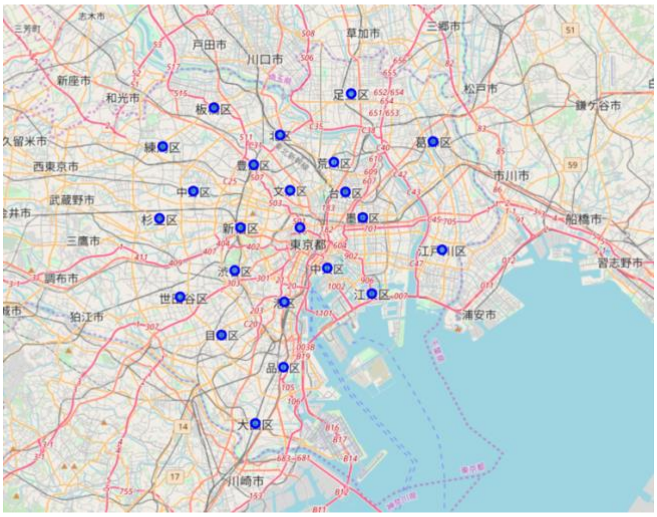
[14]:

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

df['Major_Dist_Coord']= df['Kanji'].apply(geolocator.geocode).apply(lambda x: (x.latitude, x.longitude))
df[['Latitude', 'Longitude']] = df['Major_Dist_Coord'].apply(pd.Series)

df.drop(['Major_Dist_Coord'], axis=1, inplace=True)
df
```

[14]:

	No.	Name	Kanji	Population	Density	Area	Latitude	Longitude
0	01	Chiyoda	千代田区	59441	5100	11.66	35.693810	139.753216
1	02	Chūō	中央区	147620	14460	10.21	35.666255	139.775565
2	03	Minato	港区	248071	12180	20.37	35.643227	139.740055
3	04	Shinjuku	新宿区	339211	18620	18.22	35.693763	139.703632
4	05	Bunkyō	文京区	223389	19790	11.29	35.718810	139.744732
5	06	Taitō	台東区	200486	19830	10.11	35.717450	139.790859
6	07	Sumida	墨田区	260358	18910	13.77	35.700429	139.805017
7	08	Kōtō	江東区	502579	12510	40.16	35.649154	139.812790
8	09	Shinagawa	品川区	392492	17180	22.84	35.599252	139.738910
9	10	Meguro	目黒区	280283	19110	14.67	35.621250	139.688014
10	11	Ōta	大田区	722608	11910	60.66	35.561206	139.715843
11	12	Setagaya	世田谷区	910868	15690	58.05	35.646530	139.653250
12	13	Shibuya	渋谷区	227850	15080	15.11	35.664596	139.698711
13	14	Nakano	中野区	332902	21350	15.59	35.718123	139.664468
14	15	Suginami	杉並区	570483	16750	34.06	35.699493	139.636288
15	16	Toshima	豊島区	294673	22650	13.01	35.736156	139.714222
16	17	Kita	北区	345063	16740	20.61	35.755838	139.736687
17	18	Arakawa	荒川区	213648	21030	10.16	35.737529	139.781310
18	19	Itabashi	板橋区	569225	17670	32.22	35.774143	139.681209
19	20	Nerima	練馬区	726748	15120	48.08	35.748360	139.638735
20	21	Adachi	足立区	674067	12660	53.25	35.783703	139.795319
21	22	Katsushika	葛飾区	447140	12850	34.80	35.751733	139.863816
22	23	Edogawa	江戸川区	685899	13750	49.90	35.678278	139.871091

I used python **folium** library to visualize geographic details of Tokyo and its 23 major districts and I created a map of Tokyo with boroughs superimposed on top. I used latitude and longitude values to get the visual as below:



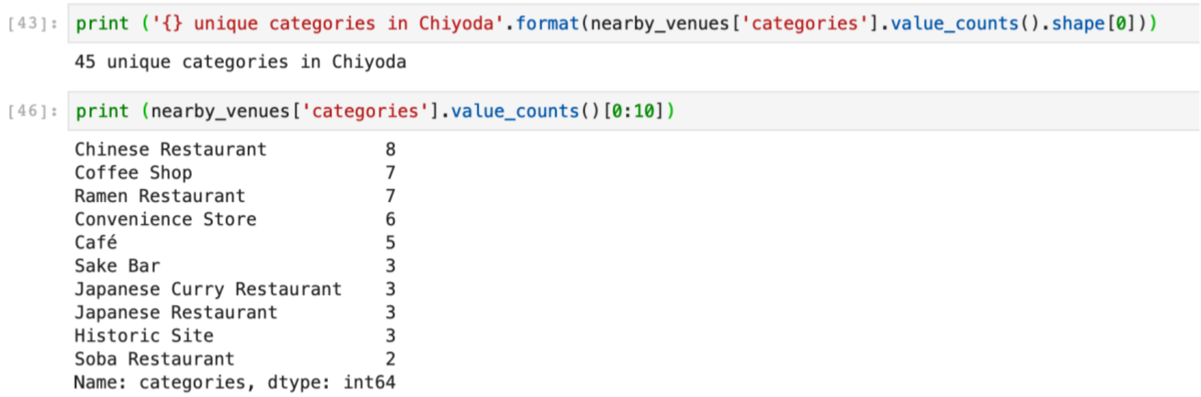
3.2. Exploratory Data Analysis:

Firstly, I will use *exploratory data analysis(EDA)* to uncover hidden properties of data and provide useful insights to the reader, both future traveler and investor.

3.1.3 Using Foursquare Location Data

Finally, let's make use of Foursquare API and get the top 100 venues that are in Chiyoda within a radius of 500 meters.

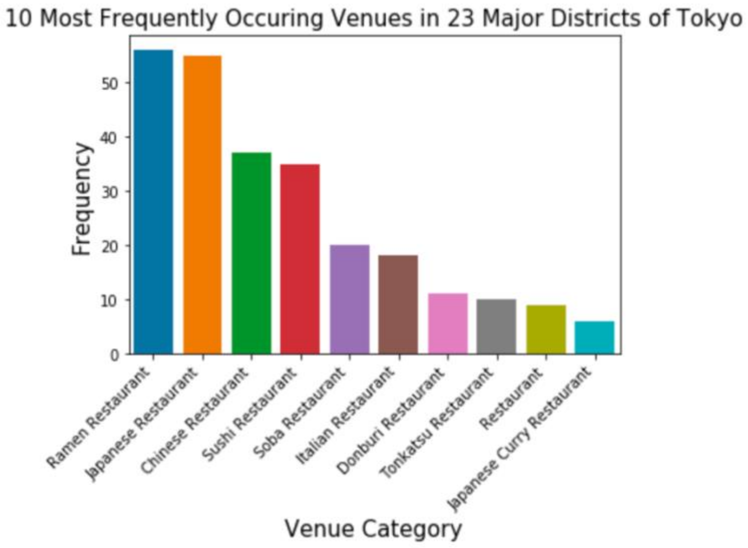
We notice that 45 unique venue categories were returned by Foursquare and Chinese Restaurants in the top of the list as we can see below.



```
[43]: print ('{} unique categories in Chiyoda'.format(nearby_venues['categories'].value_counts().shape[0]))

      45 unique categories in Chiyoda

[46]: print (nearby_venues['categories'].value_counts()[0:10])

      Chinese Restaurant            8
      Coffee Shop                   7
      Ramen Restaurant              7
      Convenience Store             6
      Café                          5
      Sake Bar                      3
      Japanese Curry Restaurant     3
      Japanese Restaurant           3
      Historic Site                 3
      Soba Restaurant               2
      Name: categories, dtype: int64
```

Later on, I will concentrate in Restaurant Category only and explore all the 23 districts.

We find out 46 unique venue categories and Ramen Restaurants top the charts as we can see in the plot below:

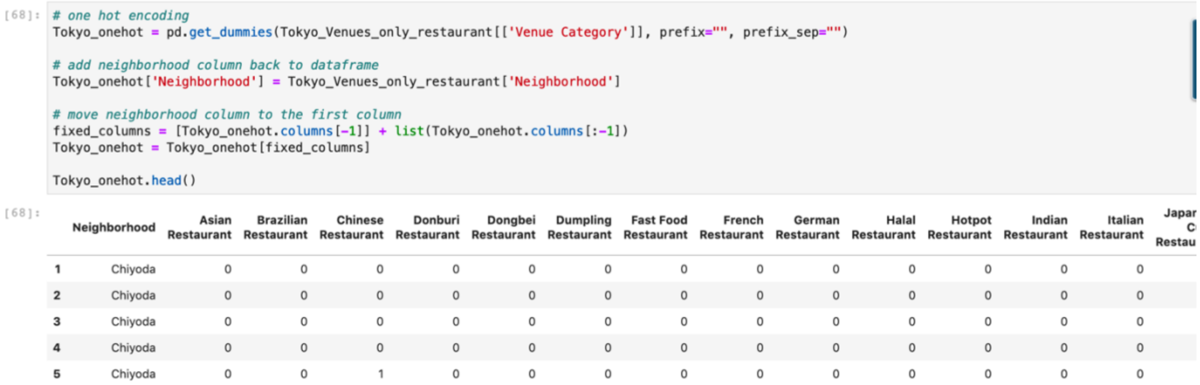


So, definitely you need to try the delicious Ramen if you visit Tokyo, but in which district Ramen restaurant are more common? Let's get back to exploring the data a little more.

Let's analyze each neighborhood to know about the top 5 venues of each one.

So, we proceed as follows:

1) Create a data-frame with pandas one hot encoding for the venue categories.

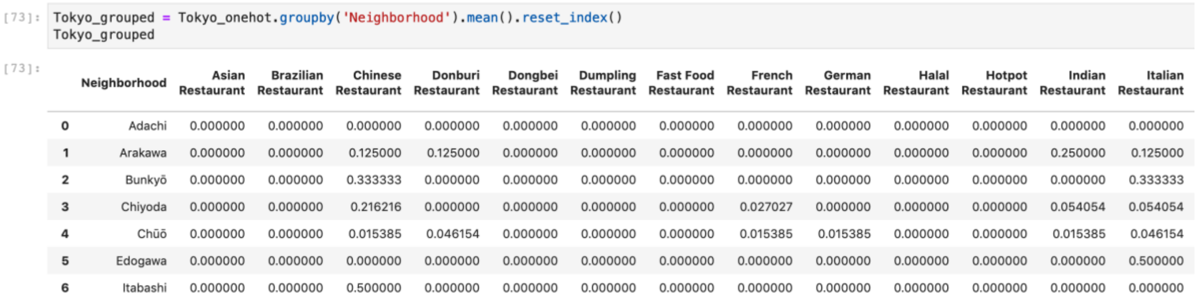


```
[68]: # one hot encoding
      Tokyo_onehot = pd.get_dummies(Tokyo_Venues_only_restaurant[['Venue Category']], prefix="", prefix_sep="")

      # add neighborhood column back to dataframe
      Tokyo_onehot['Neighborhood'] = Tokyo_Venues_only_restaurant['Neighborhood']

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

      Tokyo_onehot.head()
```

[68]:

	Neighborhood	Asian Restaurant	Brazilian Restaurant	Chinese Restaurant	Donburi Restaurant	Dongbei Restaurant	Dumpling Restaurant	Fast Food Restaurant	French Restaurant	German Restaurant	Halal Restaurant	Hotpot Restaurant	Indian Restaurant	Italian Restaurant	Japar C Restau
1	Chiyoda	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	Chiyoda	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	Chiyoda	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	Chiyoda	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	Chiyoda	0	0	1	0	0	0	0	0	0	0	0	0	0	

2) Use pandas groupby on neighborhood column and calculate the mean of the frequency of occurrence of each venue category.

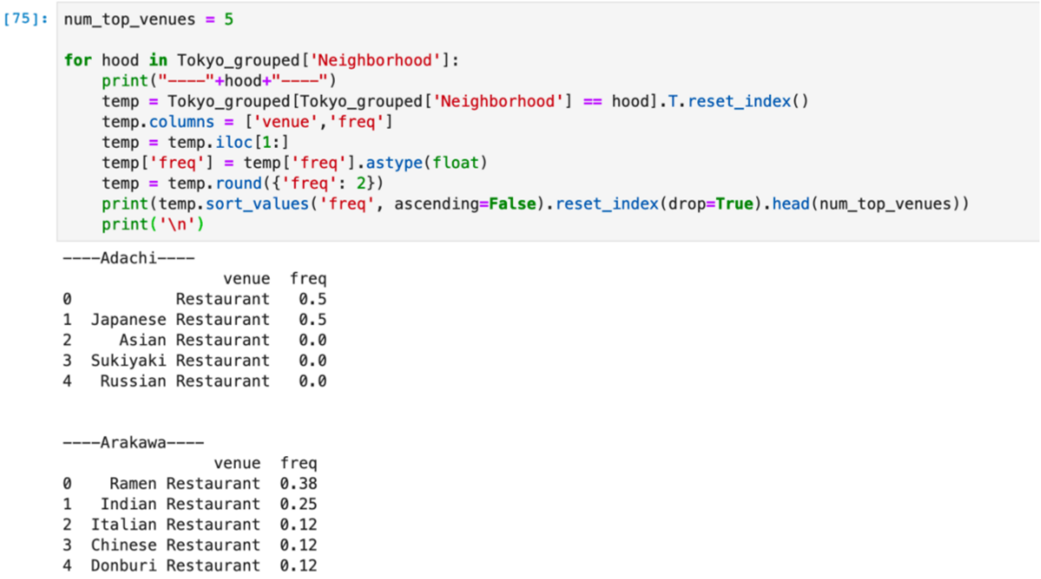


```
[73]: Tokyo_grouped = Tokyo_onehot.groupby('Neighborhood').mean().reset_index()
      Tokyo_grouped
```

[73]:

	Neighborhood	Asian Restaurant	Brazilian Restaurant	Chinese Restaurant	Donburi Restaurant	Dongbei Restaurant	Dumpling Restaurant	Fast Food Restaurant	French Restaurant	German Restaurant	Halal Restaurant	Hotpot Restaurant	Indian Restaurant	Italian Restaurant
0	Adachi	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	Arakawa	0.000000	0.000000	0.125000	0.125000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.250000	0.125000
2	Bunkyō	0.000000	0.000000	0.333333	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.333333
3	Chiyoda	0.000000	0.000000	0.216216	0.000000	0.000000	0.000000	0.000000	0.027027	0.000000	0.000000	0.000000	0.054054	0.054054
4	Chūō	0.000000	0.000000	0.015385	0.046154	0.000000	0.000000	0.000000	0.015385	0.015385	0.000000	0.000000	0.015385	0.046154
5	Edogawa	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.500000
6	Itabashi	0.000000	0.000000	0.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

3) Output each neighborhood along with the top 5 most common venues:

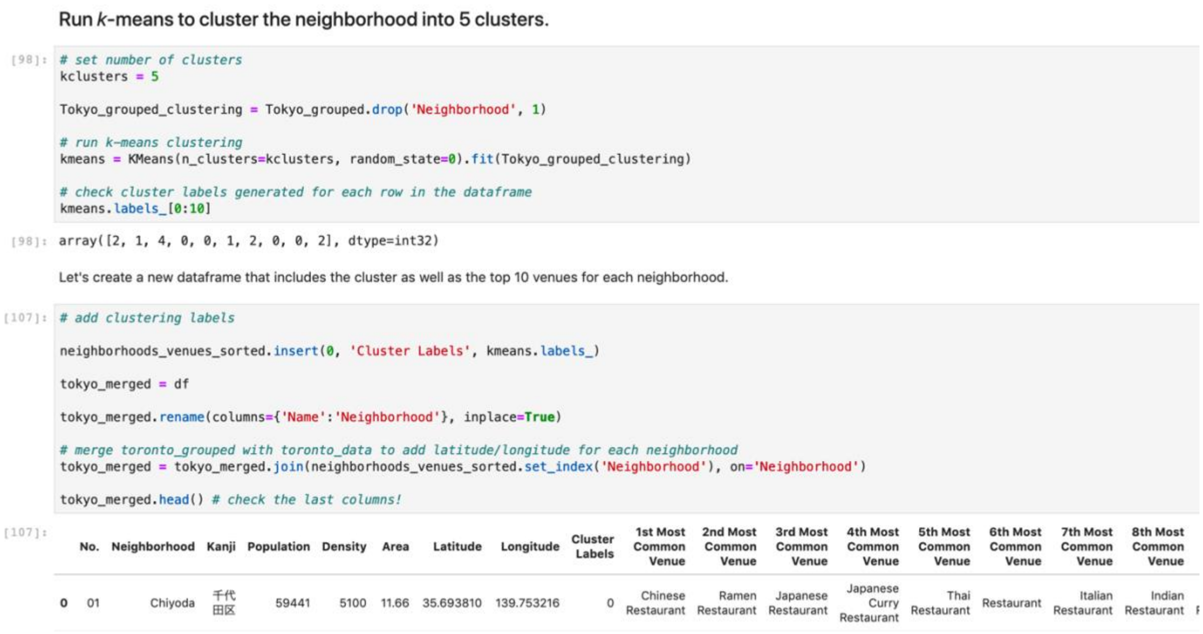


```
[75]: num_top_venues = 5

      for hood in Tokyo_grouped['Neighborhood']:
          print("----"+hood+"----")
          temp = Tokyo_grouped[Tokyo_grouped['Neighborhood'] == 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')
```

```
----Adachi----
                venue  freq
0          Restaurant   0.5
1  Japanese Restaurant   0.5
2     Asian Restaurant   0.0
3  Sukiyaki Restaurant   0.0
4   Russian Restaurant   0.0


----Arakawa----
                venue  freq
0    Ramen Restaurant  0.38
1   Indian Restaurant  0.25
2  Italian Restaurant  0.12
3  Chinese Restaurant  0.12
4  Donburi Restaurant  0.12
```

I will use *prescriptive analytics* to help a traveler decide a location to go for a restaurant. I will use *clustering* (KMeans).

Finally, we try to cluster these 23 districts based on the venue categories and use K-Means clustering. So, our expectation would be based on the similarities of venue categories, these districts will be clustered. I have used the code below :



Run *k*-means to cluster the neighborhood into 5 clusters.

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

      Tokyo_grouped_clustering = Tokyo_grouped.drop('Neighborhood', 1)

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

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

[98]: array([2, 1, 4, 0, 0, 1, 2, 0, 0, 2], dtype=int32)
```

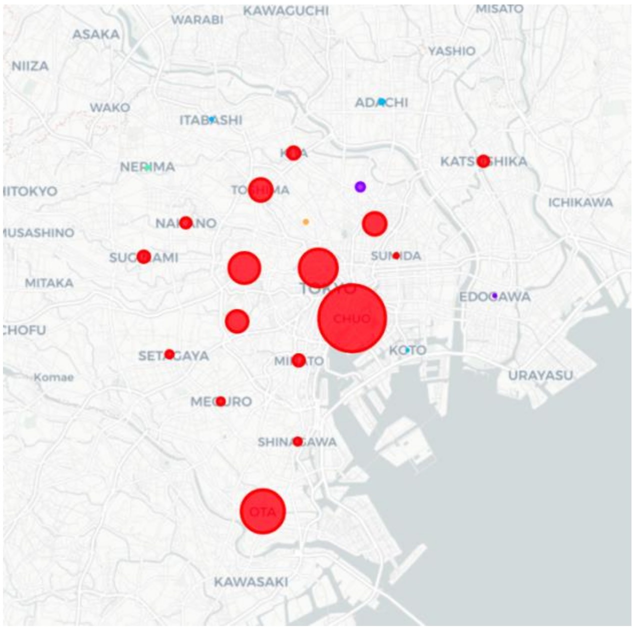
Let's create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.

```
[107]: # add clustering labels

       neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

       tokyo_merged = df

       tokyo_merged.rename(columns={'Name':'Neighborhood'}, inplace=True)

       # merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
       tokyo_merged = tokyo_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

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

[107]:

	No.	Neighborhood	Kanji	Population	Density	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 Most Common Venue	
0	01	Chiyoda	千代田区	59441	5100	11.66	35.693810	139.753216	0	Chinese Restaurant	Ramen Restaurant	Japanese Restaurant	Japanese Curry Restaurant	Thai Restaurant	Restaurant	Italian Restaurant	Indian Restaurant	F

We can represent these 5 clusters in a leaflet map using Folium library as below



4. Results & discussion

We got a glimpse of the Restaurants in Tokyo and were able to find out some interesting insights which might be useful to travelers as well as people with business interests. Let's summarize our findings:

- Ramen restaurants top the charts of most common venues in the 23 districts.
- Chuo ward and Chiyoda ward has maximum number of restaurants.
- Since the clustering was based only on the category of restaurants on each district, Tokyo's central 5 wards (C5W) all fall in the same cluster, which indicate that each of those districts presents a similar experience to the traveler in terms of category of food.

- It’s also important to note that C5W all fall within the Yamanote Line which make them accessible and easy to move between them.
- Koto, Edogawa, Adachi, Itabashi, Nerima, Sumida has the least number of restaurants.

The clustering is completely based on the most common venues obtained from Foursquare data. However, in our analysis, we have ignored other factors like distance of the venues from closest stations, range of prices of restaurants, Michelin Restaurants and so on, since we don’t have such data and it would be difficult to farm it for a small exploratory study like ours. Hence, our analysis only helps travelers to get an overview of Restaurants distribution by categories in the 23 major districts of Tokyo.

Furthermore, this results also could potentially vary if we use some other clustering techniques like DBSCAN.

6. Conclusion

In a fast-moving world, there are many real-life problems or scenarios where data can be used to find solutions to those problems. Like seen in the example above, data was used to cluster neighborhoods in Tokyo based on the most common food venues (Restaurants) in its 23 major districts. The results can help a traveler to decide about the district that fit the most his needs.

I have made use of some frequently used python libraries to scrap web-data, use Foursquare API to explore the major districts of Tokyo and saw the results of segmentation of districts using Folium leaflet map.

Similarly, data can also be used to solve other problems, which most people face in metropolitan cities. Potential for this kind of analysis in a real-life problem is discussed in great detail. Also, some of the drawbacks and chance for improvements to represent even more realistic pictures are mentioned.