Applied Data Science Capstone Project

The Battle of Tokyo Neighborhoods Restaurants

Introduction/Business Problem

The objective of this capstone project is to help the travelers of Tokyo city to choose the best restaurant that fits their needs since Tokyo is well-known as the restaurant capital of the world with over 160,000 places , using data science methodology and machine learning techniques especially clustering, this project aims to provide solutions for this problem

Foursquare's slogan is "Foursquare helps you find the places you'll love, anywhere in the world". Since the launch of the Foursquare mobile app in 2009, Foursquare has helped 60 million users discover new exciting places worldwide [1]. The app provides personalized recommendations of places to visit in the vicinity of a user's current location based on "previous browsing history, purchases, or check-in history". [2] As a result, the Foursquare app has gained popularity for helping users to discover brand new places that match their interests. Using Foursquare API will help us collect the data that we need to resolve our Business Problem

Data section

For this project we need following data:
Tokyo data that contains list districts (Wards) along with their latitude and longitude.
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:
Datasource : https://en.wikipedia.org/wiki/Special_wards_of_Tokyo#List_of_special_wards
Description : By using this API we will get all the venues in each neighborhood. We can filter these venues to get only restaurants.

Data Preparation

First , I'm gonna Scrap Data from Wikipedia Special Wards of Tokyo page to create my initial Dataframe using Pandas

```
Entrée [32]: df_initial = pd.read_html('https://en.wikipedia.org/wiki/Special_wards_of_Tokyo#List_of_special_wards')[3]
             df_initial.head()
```

Out[32]:

	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

Next , I'm gonna leave only the two columns that I'll need for the rest of the project

```
Entrée [3]: df = df_initial[["Kanji","Name"]]
            df.head()
```

Out[3]:

	Kanji	Name
0	千代田区	Chiyoda
1	中央区	Chūō
2	港区	Minato
3	新宿区	Shinjuku
4	文京区	Bunkyō

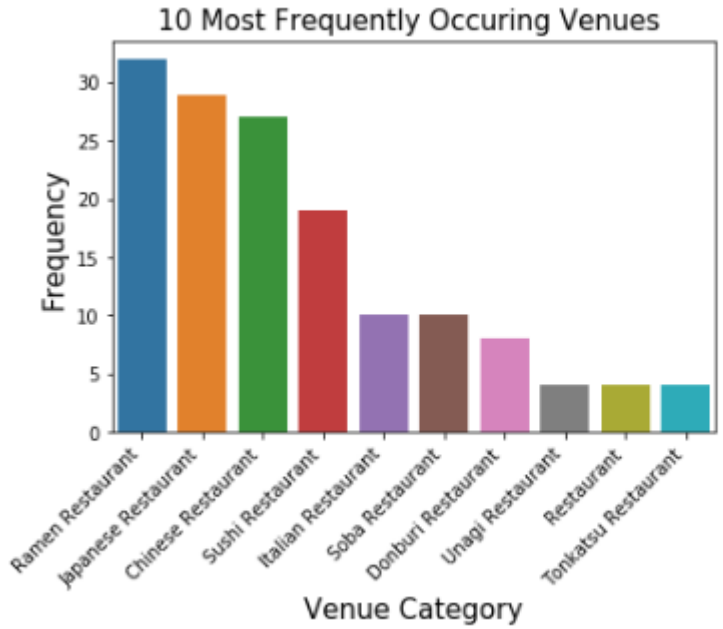
Objective now is to get the coordinates of the 23 Tokyo major districts that we have in the dataset using geocoder class of Geopy client

```
: 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.drop(['Kanji'], axis=1, inplace=True)
  df.head()
```

	Name	Latitude	Longitude
0	Chiyoda	35.693810	139.753216
1	Chūō	35.666255	139.775565
2	Minato	35.643227	139.740055
3	Shinjuku	35.693763	139.703632
4	Bunkyō	35.718810	139.744732

EDA

I will concentrate in Restaurant Category only and explore all the 23 districts , let's now see the top 10 types of restaurant that our districts has as follow



So as we can see the most commun type of restaurants in Tokyo is the 'Ramen Restaurant' , so Ramen Restaurants would be your first choice if you visit Tokyo 😉 .

Now lets move to our Clustering and to do that we need to do some Data preparation .

Since our venue category (wich include our restaurants types) is categorical we need to transform it to a numerical variable using the One Hot Encoding method .

Entrée [47]:
```
# one hot encoding
Tokyo_onehot = pd.get_dummies(Tokyo_Venues_only_restaurant_top10[['Venue Category']], prefix="", prefix_sep="")
# add neighborhood column back to dataframe
Tokyo_onehot['Neighborhood'] = Tokyo_Venues_only_restaurant_top10['Neighborhood']
Tokyo_onehot
```

Out[47]:

	Chinese Restaurant	Donburi Restaurant	Italian Restaurant	Japanese Restaurant	Ramen Restaurant	Restaurant	Soba Restaurant	Sushi Restaurant	Tonkatsu Restaurant	Unagi Restaurant	Neighborhood
1	0	0	0	0	1	0	0	0	0	0	Chiyoda
5	0	0	0	0	1	0	0	0	0	0	Chiyoda
7	0	0	0	0	1	0	0	0	0	0	Chiyoda
11	0	0	0	0	1	0	0	0	0	0	Chiyoda
66	0	0	0	0	1	0	0	0	0	0	Sumida
90	0	0	0	0	1	0	0	0	0	0	Ōta
93	0	0	0	0	1	0	0	0	0	0	Ōta
94	0	0	0	0	1	0	0	0	0	0	Ōta
96	0	0	0	0	1	0	0	0	0	0	Ōta
99	0	0	0	0	1	0	0	0	0	0	Ōta
103	0	0	0	0	1	0	0	0	0	0	Setagaya

Next i need calculate the mean of the frequency of occurrence of each restaurant categories.

Entrée [16]:
```
Tokyo_grouped = Tokyo_onehot.groupby('Neighborhood').mean().reset_index()
Tokyo_grouped.head()
```

Out[16]:

	Neighborhood	Chinese Restaurant	Donburi Restaurant	Italian Restaurant	Japanese Restaurant	Ramen Restaurant	Restaurant	Soba Restaurant	Sushi Restaurant	Tonkatsu Restaurant	Unagi Restaurant
0	Adachi	0.000000	0.000000	0.250000	0.250000	0.000000	0.5	0.000000	0.000000	0.0	0.0
1	Arakawa	0.285714	0.142857	0.142857	0.142857	0.285714	0.0	0.000000	0.000000	0.0	0.0
2	Bunkyō	0.500000	0.000000	0.000000	0.500000	0.000000	0.0	0.000000	0.000000	0.0	0.0
3	Chiyoda	0.454545	0.000000	0.090909	0.000000	0.363636	0.0	0.000000	0.090909	0.0	0.0
4	Chūō	0.000000	0.000000	0.105263	0.157895	0.000000	0.0	0.105263	0.631579	0.0	0.0

Finally , I will use *clustering* (KMeans) to help a traveler decide a location to go for a restaurant , and in order to do that we try to cluster our23 districts based on the restaurant categories and our expectation would be based on the similarities of venue categories, these districts will be clustered.

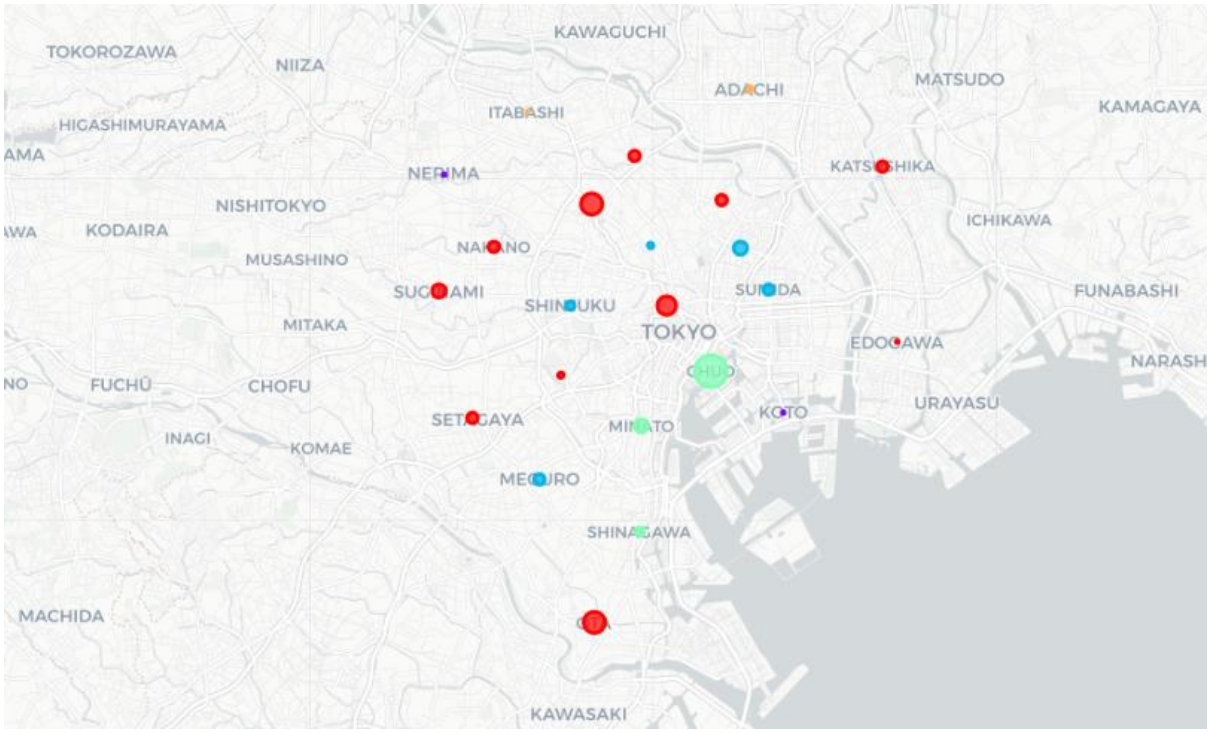
Entrée [53]:
```
# add clustering labels

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!
```

```
C:\Users\GOUTAIBABDERRAFII\anaconda3\lib\site-packages\pandas\core\frame.py:4133: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  errors=errors,
```

Out[53]:

	Neighborhood	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	9th Most Common Venue	10th I Com V
0	Chiyoda	35.693810	139.753216	0.0	Chinese Restaurant	Ramen Restaurant	Sushi Restaurant	Italian Restaurant	Unagi Restaurant	Tonkatsu Restaurant	Soba Restaurant	Restaurant	Japanese Restaurant	Do Resta
1	Chūō	35.666255	139.775565	3.0	Sushi Restaurant	Japanese Restaurant	Soba Restaurant	Italian Restaurant	Unagi Restaurant	Tonkatsu Restaurant	Restaurant	Ramen Restaurant	Donburi Restaurant	Chi Resta
2	Minato	35.643227	139.740055	3.0	Soba Restaurant	Japanese Restaurant	Chinese Restaurant	Unagi Restaurant	Tonkatsu Restaurant	Sushi Restaurant	Restaurant	Ramen Restaurant	Italian Restaurant	Do Resta
3	Shinjuku	35.693763	139.703632	2.0	Japanese Restaurant	Unagi Restaurant	Chinese Restaurant	Tonkatsu Restaurant	Sushi Restaurant	Soba Restaurant	Restaurant	Ramen Restaurant	Italian Restaurant	Do Resta
4	Bunkyō	35.718810	139.744732	2.0	Japanese Restaurant	Chinese Restaurant	Unagi Restaurant	Tonkatsu Restaurant	Sushi Restaurant	Soba Restaurant	Restaurant	Ramen Restaurant	Italian Restaurant	Do Resta

Lets make now our Clusters representation easier using Folium Map



Conclusion and Discussion

- ✓ Ramen restaurants top the charts of most common venues in the 23 districts.
- ✓ Chuo ward and Chiyoda ward has maximum number of restaurants.
- ✓ Koto, Edogawa, Adachi, Itabashi, Nerima, Sumida has the least number of restaurants.

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.