Battle of the Neighborhoods

Devang Rai

Table of Contents

Introduction and Business Understanding

Although we are currently facing low tourism levels due to the COVID-19 pandemic, it is certain that once restrictions are lifted, tourism will flourish once again. Our business problem is simple: **how do we provide support for tourists who flock to Tokyo to help them visualize different districts and neighborhoods to scout out culinary and food venues that satisfy their needs?**

With nearly 50 million tourists annually, Tokyo is expected to see a greater number of tourists due to the current quarantine conditions.

Tokyo is one of many culinary capitals of the world, branding its own takes on popular foods, such as ramen and tendura. However, it is not easy for tourists to locate ideal restaurants. A quick Google search will show many varying reviews for a variety of different locations. This makes it hard for tourists to identify neighborhoods with the best restaurants and cuisine.

In this project, we will be utilizing Foursquare and machine learning to group different neighborhoods by their restaurant venues information.

Data Requirements

For this project, we will need:

- Tokyo district data with coordinates
  - The data source is: https://en.wikipedia.org/wiki/Special_wards_of_Tokyo#list
  - This data will be utilized to obtain the coordinates of the neighborhoods by using the Geocoder class of Geopy
- Restaurants in each district in Tokyo
  - Data source: Foursquare API
  - We will use the Foursquare API to identify venues and then filter it down to restaurants

Methodology

Preparation of Data: Scrapping Data from Wikipedia

First, we will use the pandas dataframe to take data from the Wikipedia list. We will edit the dataframe to change the names of some columns and drop irrelevant columns.

[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

Preparation of Data: Getting coordinates

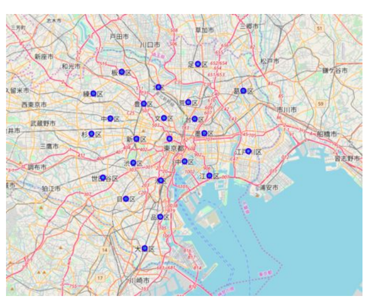
We will now try to use the Geopy client to obtain the coordinates (latitude and longitude) of the districts.

```
[14]: 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

We can use a the folium library in python to get a better visual of Tokyo.

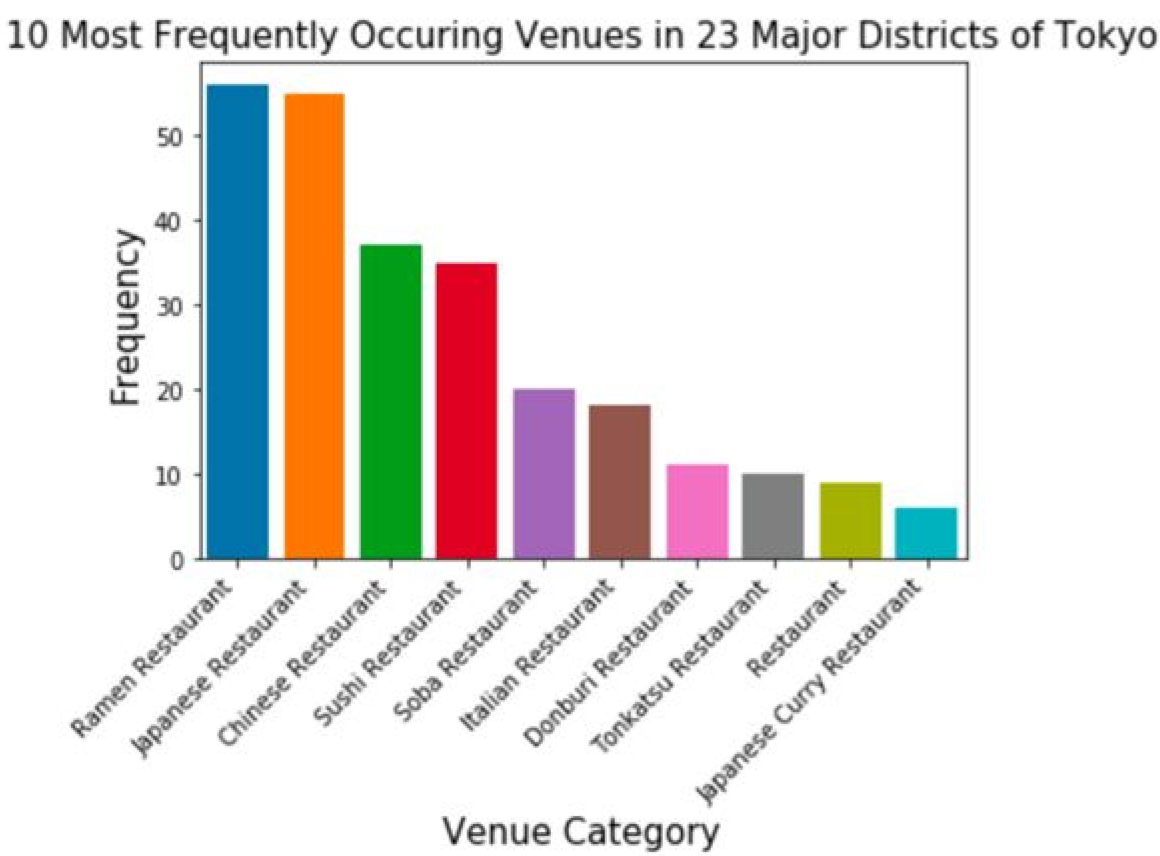


Exploratory Data Analysis: Using Foursquare

Now, we can use Foursquare API to get the top 100 venues in Chiyoda within a 500 meter radius.

```
[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
```

We find out unique venue categories as we chart the data.



Now we will analyze each neighborhood for the top 5 venues. We first create a dataframe with pandas one hot encoding for the venue categories.

```
# 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()
```

	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	Japa 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	

We will now calculate the mean of the frequency of occurrence of each venue category.

```
Tokyo_grouped = Tokyo_onehot.groupby('Neighborhood').mean().reset_index()
Tokyo_grouped
```

	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

Now we can output each neighborhood 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
```

Now, we can use K-Means clustering.

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)
```

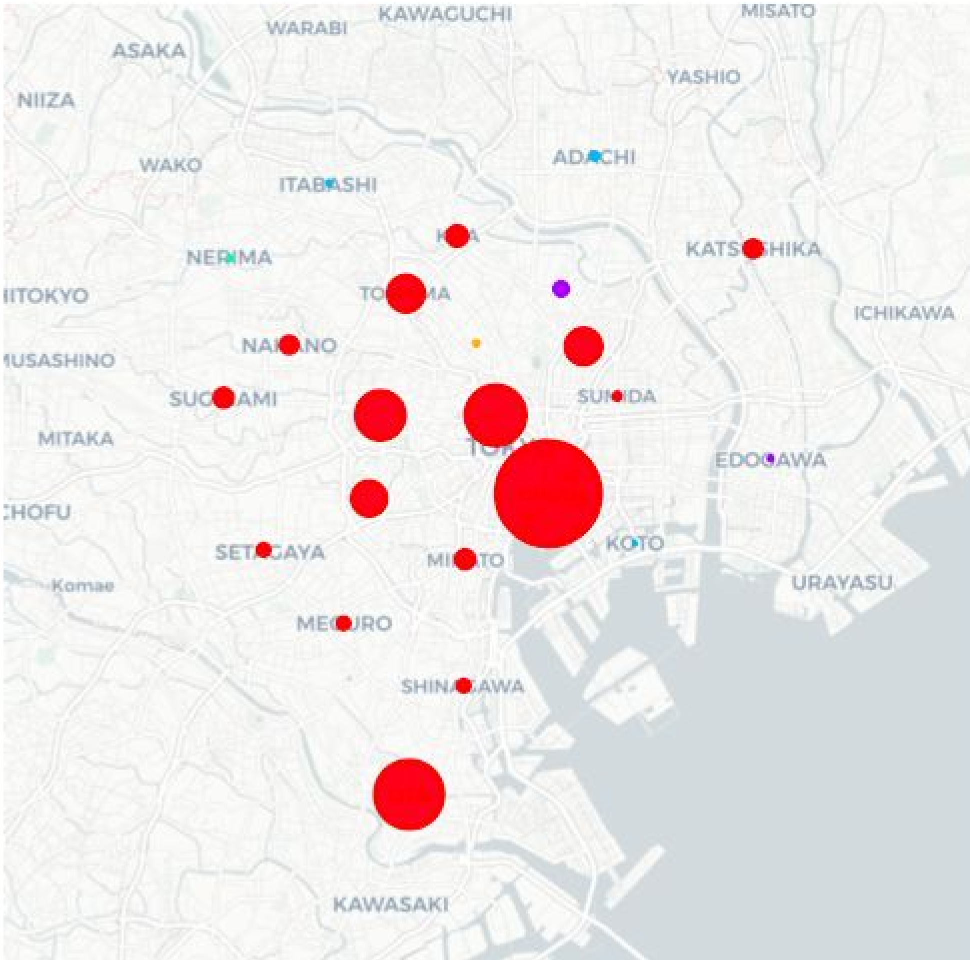
Let's create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.

[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

Using folium, we can visually represent these 5 clusters.


WARABI
KAWAGUCHI
MISATO
ASAKA
YASHIO
NIIZA
WAKO
ICHIKAWA
MITAKA
Komae
URAYASU
KAWASAKI

Results & Discussion

First, let’s summarize the findings. Firstly, we learned that ramen restaurants top the most common venue charts in 23 districts. Out of all of the districts, Chuo ward and Chiyoda have the most restaurants, while Koto, Edogawa, Adachi, Itabashi, Nerima, and Sumida have the lowest number of restaurants. Because the clustering was solely based on the category of restaurants, the five central districts fall under the same cluster, which indicates that they have similar cuisine options available to tourists. Our clustering may wildly vary if we used a density clustering method such as DBSCAN.

Conclusion

In a fast-paced world, machine learning is essential to find solutions to problems ranging in difficulty. In this problem, we used data of neighborhoods in Tokyo to determine the most common food venues per district. This simple algorithm can assist tourists to understand which districts they want to visit.

To create this algorithm, we first scrapped data from Wikipedia. Then, we used Foursquare API to explore the districts of Tokyo, and used Folium to properly visualize it. We then used unsupervised learning to create a cluster model.

By using machine learning algorithms, we can perform real world analysis in a deeper scope.

Acknowledgements

Sources

"Special Wards of Tokyo." *Wikipedia*, Wikimedia Foundation, 17 June 2020, en.wikipedia.org/wiki/Special_wards_of_Tokyo.