Battle of Neighborhood

July 29, 2020

0.1 Introduction/Business Problem

As Tokyo is the most populated city in Japan, there are a bunch of people who is willing to try different kinds of restaurants in this city. The goal of this project is to find restaurants that people like mostly. We aim to provide effective information based on our investigation on multiple neighborhoods and to assist in the selection of most appropriate areas that will promote restaurant industry.

0.2 Data

The datasets we need are displayed below: 1. Wikipedia: Special wards of Tokyo 2. Foursquare API: information on restaurants in the neighborhoods of Tokyo 3. Geopy: geological location in Tokyo

0.3 Methodology

First, use Special wards of Tokyo from Wikipedia to create a dataframe that includes names, districts, areas of Tokyo.

[47]:
```
import numpy as np
import pandas as pd
from geopy.geocoders import Nominatim
```

[23]:
```
df=pd.read_html('https://en.wikipedia.org/wiki/
→Special_wards_of_Tokyo#List_of_special_wards')[3]
df.head()
```

[23]:
```
  No. Flag      Name Kanji  Population(as of October 2016  Density(/km2)  \
0  01  NaN   Chiyoda                                59441           5100
1  02  NaN      Chūō                               147620          14460
2  03  NaN    Minato                               248071          12180
3  04  NaN  Shinjuku                               339211          18620
4  05  NaN    Bunkyō                               223389          19790

   Area(km2)                                   Major districts
0      11.66  Nagatachō, Kasumigaseki, Ōtemachi, Marunouchi,…
1      10.21  Nihonbashi, Kayabachō, Ginza, Tsukiji, Hatchōb…
2      20.37  Odaiba, Shinbashi, Hamamatsuchō, Mita, Roppong…
3      18.22  Shinjuku, Takadanobaba, Ōkubo, Kagurazaka, Ich…
```

```
4      11.29                              Hongō, Yayoi, Hakusan
```

Then we do data-processing step.

[24]:
```
df=df.drop(columns=['Flag', 'Major districts'] )
df.head()
```

[24]:
```
  No.       Name Kanji  Population(as of October 2016  Density(/km2)  Area(km2)
0  01    Chiyoda                                59441           5100      11.66
1  02       Chūō                               147620          14460      10.21
2  03     Minato                               248071          12180      20.37
3  04   Shinjuku                               339211          18620      18.22
4  05     Bunkyō                               223389          19790      11.29
```

[44]:
```
df=df.rename(columns={df.columns[3]:'Population', 'Density(/km2)':'Density',␣
 ↪'Area(km2)':'Area'})
df=df.drop([23])
df
```

[44]:
```
   No.         Name Kanji  Population  Density   Area
0   01      Chiyoda             59441     5100  11.66
1   02         Chūō            147620    14460  10.21
2   03       Minato            248071    12180  20.37
3   04     Shinjuku            339211    18620  18.22
4   05       Bunkyō            223389    19790  11.29
5   06        Taitō            200486    19830  10.11
6   07       Sumida            260358    18910  13.77
7   08         Kōtō            502579    12510  40.16
8   09    Shinagawa            392492    17180  22.84
9   10       Meguro            280283    19110  14.67
10  11          Ōta            722608    11910  60.66
11  12     Setagaya            910868    15690  58.05
12  13      Shibuya            227850    15080  15.11
13  14       Nakano            332902    21350  15.59
14  15     Suginami            570483    16750  34.06
15  16      Toshima            294673    22650  13.01
16  17         Kita            345063    16740  20.61
17  18      Arakawa            213648    21030  10.16
18  19     Itabashi            569225    17670  32.22
19  20       Nerima            726748    15120  48.08
20  21       Adachi            674067    12660  53.25
21  22   Katsushika            447140    12850  34.80
22  23      Edogawa            685899    13750  49.90
```

[45]:
```
!conda install -c conda-forge geopy --yes
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

```
# All requested packages already installed.
```

Now we use geopy to get geospatial data.

[49]:
```
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=df.drop(['Major_Dist_Coord'], axis=1)
df
```

[49]:

	No.	Name	Kanji	Population	Density	Area	Latitude	Longitude
0	01	Chiyoda		59441	5100	11.66	35.693810	139.753216
1	02	Chūō		147620	14460	10.21	35.666255	139.775565
2	03	Minato		248071	12180	20.37	35.643227	139.740055
3	04	Shinjuku		339211	18620	18.22	35.693763	139.703632
4	05	Bunkyō		223389	19790	11.29	35.718810	139.744732
5	06	Taitō		200486	19830	10.11	35.717450	139.790859
6	07	Sumida		260358	18910	13.77	35.700429	139.805017
7	08	Kōtō		502579	12510	40.16	35.649154	139.812790
8	09	Shinagawa		392492	17180	22.84	35.599252	139.738910
9	10	Meguro		280283	19110	14.67	35.621250	139.688014
10	11	Ōta		722608	11910	60.66	35.561206	139.715843
11	12	Setagaya		910868	15690	58.05	35.646096	139.656270
12	13	Shibuya		227850	15080	15.11	35.664596	139.698711
13	14	Nakano		332902	21350	15.59	35.718123	139.664468
14	15	Suginami		570483	16750	34.06	35.699493	139.636288
15	16	Toshima		294673	22650	13.01	35.736156	139.714222
16	17	Kita		345063	16740	20.61	35.755838	139.736687
17	18	Arakawa		213648	21030	10.16	35.737529	139.781310
18	19	Itabashi		569225	17670	32.22	35.774143	139.681209
19	20	Nerima		726748	15120	48.08	35.748360	139.638735
20	21	Adachi		674067	12660	53.25	35.783703	139.795319
21	22	Katsushika		447140	12850	34.80	35.751733	139.863816
22	23	Edogawa		685899	13750	49.90	35.678278	139.871091

Then using python folium library to visualize geographic details of Tokyo and its 23 major districts.

[50]:
```
import json
import requests # library to handle requests
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

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you␣
↪haven't completed the Foursquare API lab
import folium
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

[51]:
```
address = 'Tokyo'
geolocator = Nominatim(user_agent="Tokyo_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Tokyo are {}, {}.'.format(latitude,␣
↪longitude))
```

```
The geograpical coordinate of Tokyo are 35.6828387, 139.7594549.
```

[52]:
```
# create map of Tokyo using latitude and longitude values
map_tokyo = folium.Map(location=[latitude, longitude], zoom_start=11)

for lat, lng, label in zip(df['Latitude'], df['Longitude'], df['Name']):
    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_tokyo)

map_tokyo
```

[52]:
```
<folium.folium.Map at 0x7f98997ce550>
```

Now using Foursquare API to explore the neighborhoods.

```
[53]: CLIENT_ID = 'MQ0XIFC4T5X5MCUWN1M3VZ011RFZQPP5JZ1A002PIIWJZPN0' # your␣
 ↪Foursquare ID
CLIENT_SECRET = 'D03SU02BKY0AAVW0WC3302I5QZNZLI1ULXDJSWE4A0CHR5S5' # your␣
 ↪Foursquare Secret
VERSION = '20180604'
LIMIT = 30
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: MQ0XIFC4T5X5MCUWN1M3VZ011RFZQPP5JZ1A002PIIWJZPN0
CLIENT_SECRET:D03SU02BKY0AAVW0WC3302I5QZNZLI1ULXDJSWE4A0CHR5S5
```

Now using Chiyoda as an example.

```
[56]: neighborhood_latitude = df.loc[0, 'Latitude'] # neighborhood latitude value
neighborhood_longitude = df.loc[0, 'Longitude'] # neighborhood longitude value

neighborhood_name = df.loc[0, 'Name'] # neighborhood name

print('Latitude and longitude values of {} are {}, {}.'.
 ↪format(neighborhood_name,
                                                                         ␣
 ↪neighborhood_latitude,
                                                                         ␣
 ↪neighborhood_longitude))
```

```
Latitude and longitude values of Chiyoda are 35.6938097, 139.7532163.
```

```
[57]: LIMIT = 50
radius = 500

url = 'https://api.foursquare.com/v2/venues/explore?
 ↪&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    neighborhood_latitude,
    neighborhood_longitude,
    radius,
    LIMIT)
```

```
[58]: results = requests.get(url).json()
```

```
[59]: def get_category_type(row):
    try:
        categories_list = row['categories']
```

```python
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

[60]:

```python
venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat',␣
 ↪'venue.location.lng']
nearby_venues =nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type,␣
 ↪axis=1)

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues.head()
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:3:
FutureWarning: pandas.io.json.json_normalize is deprecated, use
pandas.json_normalize instead
  This is separate from the ipykernel package so we can avoid doing imports
until
```

[60]:

	name	categories	lat	lng
0	Jimbocho Kurosu ( )	Ramen Restaurant	35.695539	139.754851
1	Kanda Tendonya ( )	Tempura Restaurant	35.695765	139.754682
2	Kitanomaru Park ( )	Park	35.691653	139.751201
3	Nippon Budokan ( )	Stadium	35.693356	139.749865
4	Mori no Butchers ( )	Gastropub	35.694770	139.755980

[61]:

```python
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={}'.format(
            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 = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']


    return(nearby_venues)
```

[62]:

```
Tokyo_venues = getNearbyVenues(names=df['Name'],
                                   latitudes=df['Latitude'],
                                   longitudes=df['Longitude']
                                  )
```

```
Chiyoda
Chūō
Minato
Shinjuku
Bunkyō
Taitō
```

```
Sumida
Kōtō
Shinagawa
Meguro
Ōta
Setagaya
Shibuya
Nakano
Suginami
Toshima
Kita
Arakawa
Itabashi
Nerima
Adachi
Katsushika
Edogawa
```

[65]:

```
Tokyo_restaurant = Tokyo_venues[Tokyo_venues['Venue Category'].str.
 ↪contains('Restaurant')].reset_index(drop=True)
Tokyo_restaurant.index = np.arange(1, len(Tokyo_Venues_only_restaurant )+1)
```

[67]:

```
print (Tokyo_restaurant['Venue Category'].value_counts())
```

```
Ramen Restaurant             41
Japanese Restaurant          33
Chinese Restaurant           31
Sushi Restaurant             26
Italian Restaurant           16
Soba Restaurant              13
Donburi Restaurant            9
Tonkatsu Restaurant           9
Unagi Restaurant              7
Indian Restaurant             6
Restaurant                    5
Yakitori Restaurant           5
Seafood Restaurant            4
Yoshoku Restaurant            4
Thai Restaurant               3
Japanese Curry Restaurant     3
Korean Restaurant             3
Dumpling Restaurant           3
Tempura Restaurant            3
Szechuan Restaurant           2
Sukiyaki Restaurant           2
Asian Restaurant              2
Nabe Restaurant               2
Kaiseki Restaurant            1
```

```
Fast Food Restaurant          1
Kushikatsu Restaurant         1
Hotpot Restaurant             1
Middle Eastern Restaurant     1
Vietnamese Restaurant         1
French Restaurant             1
Kosher Restaurant             1
Spanish Restaurant            1
Mexican Restaurant            1
Shabu-Shabu Restaurant        1
Monjayaki Restaurant          1
Brazilian Restaurant          1
Udon Restaurant               1
German Restaurant             1
Kebab Restaurant              1
Dongbei Restaurant            1
South Indian Restaurant       1
African Restaurant            1
Name: Venue Category, dtype: int64
```

[76]:
```
# create a dataframe of top 10 categories
Tokyo_Top10 = Tokyo_restaurant['Venue Category'].value_counts()[0:10].
 ↪to_frame(name='frequency')
Tokyo_Top10=Tokyo_Top10.reset_index()
Tokyo_Top10

Tokyo_Top10.rename(index=str, columns={"index": "Venue_Category", "frequency":␣
 ↪"Frequency"}, inplace=True)
Tokyo_Top10
```

[76]:

	Venue_Category	Frequency
0	Ramen Restaurant	41
1	Japanese Restaurant	33
2	Chinese Restaurant	31
3	Sushi Restaurant	26
4	Italian Restaurant	16
5	Soba Restaurant	13
6	Donburi Restaurant	9
7	Tonkatsu Restaurant	9
8	Unagi Restaurant	7
9	Indian Restaurant	6

[79]:
```
Tokyo_Venues_restaurant = Tokyo_restaurant.groupby(['Neighborhood'])['Venue␣
 ↪Category'].apply(lambda x: x[x.str.contains('Restaurant')].count())
```

[80]:
```
Tokyo_Venues_restaurant
```

[80]:

```
Neighborhood
Adachi          2
Arakawa         9
Bunkyō          3
Chiyoda        20
Chūō           39
Edogawa         2
Itabashi        3
Katsushika      4
Kita           14
Kōtō            3
Meguro          6
Minato         11
Nakano          8
Nerima          1
Setagaya        7
Shibuya        13
Shinagawa       7
Shinjuku       15
Suginami       12
Sumida          8
Taitō          16
Toshima        20
Ōta            28
Name: Venue Category, dtype: int64
```

[83]:

```python
Tokyo_Venues_restaurant_df = Tokyo_Venues_restaurant.to_frame().reset_index()
Tokyo_Venues_restaurant_df.columns = ['Neighborhood', 'Number of Restaurant']
Tokyo_Venues_restaurant_df.index = np.arange(1,␣
↪len(Tokyo_Venues_restaurant_df)+1)
Tokyo_Venues_restaurant_df['Number of Restaurant'].to_list()
Tokyo_Venues_restaurant_df['Neighborhood'].to_list()
Tokyo_Venues_restaurant_df
```

[83]:

	Neighborhood	Number of Restaurant
1	Adachi	2
2	Arakawa	9
3	Bunkyō	3
4	Chiyoda	20
5	Chūō	39
6	Edogawa	2
7	Itabashi	3
8	Katsushika	4
9	Kita	14
10	Kōtō	3
11	Meguro	6
12	Minato	11

```
13        Nakano                       8
14        Nerima                       1
15      Setagaya                       7
16       Shibuya                      13
17     Shinagawa                       7
18      Shinjuku                      15
19      Suginami                      12
20        Sumida                       8
21         Taitō                      16
22       Toshima                      20
23           Ōta                      28
```

[97]:

```
Tokyo_onehot = pd.get_dummies(Tokyo_restaurant[['Venue Category']], prefix="",␣
 ↪prefix_sep="")

# add neighborhood column back to dataframe
Tokyo_onehot['Neighborhood'] = Tokyo_restaurant['Neighborhood']
Tokyo_onehot.head()
Tokyo_onehot.shape
```

[97]: (251, 43)

[93]:

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

[93]:

	Neighborhood	African Restaurant	Asian Restaurant	Brazilian Restaurant	\
0	Adachi	0.000000	0.000000	0.000000	
1	Arakawa	0.000000	0.000000	0.000000	
2	Bunkyō	0.000000	0.000000	0.000000	
3	Chiyoda	0.000000	0.000000	0.000000	
4	Chūō	0.000000	0.000000	0.000000	
5	Edogawa	0.000000	0.000000	0.000000	
6	Itabashi	0.000000	0.000000	0.000000	
7	Katsushika	0.000000	0.000000	0.000000	
8	Kita	0.000000	0.000000	0.000000	
9	Kōtō	0.000000	0.000000	0.000000	
10	Meguro	0.000000	0.000000	0.000000	
11	Minato	0.000000	0.000000	0.000000	
12	Nakano	0.000000	0.000000	0.000000	
13	Nerima	0.000000	0.000000	0.000000	
14	Setagaya	0.000000	0.000000	0.000000	
15	Shibuya	0.076923	0.076923	0.076923	
16	Shinagawa	0.000000	0.000000	0.000000	
17	Shinjuku	0.000000	0.000000	0.000000	
18	Suginami	0.000000	0.083333	0.000000	
19	Sumida	0.000000	0.000000	0.000000	
20	Taitō	0.000000	0.000000	0.000000	

```
21      Toshima           0.000000          0.000000              0.000000
22          Ōta           0.000000          0.000000              0.000000

    Chinese Restaurant  Donburi Restaurant  Dongbei Restaurant  \
0             0.000000            0.000000                0.00
1             0.222222            0.111111                0.00
2             0.333333            0.000000                0.00
3             0.200000            0.000000                0.00
4             0.000000            0.025641                0.00
5             0.000000            0.000000                0.00
6             0.333333            0.000000                0.00
7             0.000000            0.500000                0.00
8             0.071429            0.071429                0.00
9             0.333333            0.000000                0.00
10            0.333333            0.000000                0.00
11            0.090909            0.000000                0.00
12            0.250000            0.125000                0.00
13            1.000000            0.000000                0.00
14            0.000000            0.000000                0.00
15            0.153846            0.076923                0.00
16            0.000000            0.142857                0.00
17            0.066667            0.000000                0.00
18            0.083333            0.000000                0.00
19            0.250000            0.000000                0.00
20            0.125000            0.000000                0.00
21            0.100000            0.050000                0.05
22            0.178571            0.000000                0.00

    Dumpling Restaurant  Fast Food Restaurant  French Restaurant  …  \
0              0.000000              0.000000           0.000000  …
1              0.000000              0.000000           0.000000  …
2              0.000000              0.000000           0.000000  …
3              0.000000              0.000000           0.000000  …
4              0.000000              0.000000           0.000000  …
5              0.000000              0.000000           0.000000  …
6              0.000000              0.000000           0.000000  …
7              0.250000              0.000000           0.000000  …
8              0.071429              0.000000           0.000000  …
9              0.000000              0.000000           0.000000  …
10             0.000000              0.000000           0.000000  …
11             0.000000              0.000000           0.090909  …
12             0.000000              0.000000           0.000000  …
13             0.000000              0.000000           0.000000  …
14             0.000000              0.142857           0.000000  …
15             0.000000              0.000000           0.000000  …
16             0.000000              0.000000           0.000000  …
17             0.000000              0.000000           0.000000  …
```

18	0.083333	0.000000	0.000000	…
19	0.000000	0.000000	0.000000	…
20	0.000000	0.000000	0.000000	…
21	0.000000	0.000000	0.000000	…
22	0.000000	0.000000	0.000000	…

	Sushi Restaurant	Szechuan Restaurant	Tempura Restaurant	\
0	0.000000	0.000000	0.000000	
1	0.000000	0.000000	0.000000	
2	0.000000	0.333333	0.000000	
3	0.100000	0.000000	0.050000	
4	0.435897	0.000000	0.025641	
5	0.000000	0.000000	0.000000	
6	0.000000	0.000000	0.000000	
7	0.000000	0.000000	0.000000	
8	0.000000	0.000000	0.000000	
9	0.000000	0.000000	0.000000	
10	0.166667	0.000000	0.000000	
11	0.000000	0.000000	0.000000	
12	0.000000	0.000000	0.000000	
13	0.000000	0.000000	0.000000	
14	0.000000	0.142857	0.000000	
15	0.000000	0.000000	0.000000	
16	0.142857	0.000000	0.000000	
17	0.066667	0.000000	0.000000	
18	0.000000	0.000000	0.000000	
19	0.125000	0.000000	0.000000	
20	0.125000	0.000000	0.000000	
21	0.000000	0.000000	0.000000	
22	0.035714	0.000000	0.035714	

	Thai Restaurant	Tonkatsu Restaurant	Udon Restaurant	Unagi Restaurant	\
0	0.000000	0.000000	0.00	0.000000	
1	0.000000	0.000000	0.00	0.000000	
2	0.000000	0.000000	0.00	0.000000	
3	0.000000	0.000000	0.00	0.000000	
4	0.000000	0.025641	0.00	0.051282	
5	0.000000	0.000000	0.00	0.000000	
6	0.000000	0.000000	0.00	0.000000	
7	0.000000	0.000000	0.00	0.000000	
8	0.000000	0.000000	0.00	0.000000	
9	0.000000	0.000000	0.00	0.000000	
10	0.000000	0.000000	0.00	0.000000	
11	0.000000	0.000000	0.00	0.000000	
12	0.000000	0.125000	0.00	0.000000	
13	0.000000	0.000000	0.00	0.000000	
14	0.000000	0.000000	0.00	0.142857	

```
15          0.000000          0.000000          0.00          0.000000
16          0.000000          0.000000          0.00          0.000000
17          0.133333          0.066667          0.00          0.066667
18          0.000000          0.083333          0.00          0.000000
19          0.000000          0.125000          0.00          0.125000
20          0.000000          0.000000          0.00          0.062500
21          0.050000          0.050000          0.05          0.000000
22          0.000000          0.107143          0.00          0.035714

    Vietnamese Restaurant  Yakitori Restaurant  Yoshoku Restaurant
0                0.000000             0.000000            0.000000
1                0.000000             0.000000            0.000000
2                0.000000             0.000000            0.000000
3                0.000000             0.000000            0.050000
4                0.000000             0.025641            0.000000
5                0.000000             0.000000            0.000000
6                0.000000             0.000000            0.000000
7                0.000000             0.000000            0.000000
8                0.000000             0.000000            0.000000
9                0.000000             0.000000            0.000000
10               0.000000             0.000000            0.000000
11               0.000000             0.090909            0.000000
12               0.000000             0.000000            0.000000
13               0.000000             0.000000            0.000000
14               0.000000             0.000000            0.000000
15               0.000000             0.000000            0.000000
16               0.000000             0.000000            0.000000
17               0.000000             0.133333            0.066667
18               0.000000             0.000000            0.000000
19               0.000000             0.000000            0.000000
20               0.000000             0.000000            0.000000
21               0.000000             0.050000            0.050000
22               0.035714             0.000000            0.035714

[23 rows x 43 columns]
```

[102]:
```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

[105]:
```
num_top_venues = 10
indicators = ['st', 'nd', 'rd']
columns = ['Neighborhood']

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))
```

[110]:
```
neighborhoods_venues = pd.DataFrame(columns=columns)
neighborhoods_venues['Neighborhood'] = onehot_df['Neighborhood']

for ind in np.arange(onehot_df.shape[0]):
    neighborhoods_venues.iloc[ind, 1:] = return_most_common_venues(onehot_df.
 ↪iloc[ind, :], num_top_venues)

neighborhoods_venues.head(23)
```

[110]:
```
   Neighborhood 1st Most Common Venue 2nd Most Common Venue  \
0        Adachi            Restaurant   Japanese Restaurant
1       Arakawa      Ramen Restaurant    Chinese Restaurant
2        Bunkyō    Chinese Restaurant   Japanese Restaurant
3       Chiyoda      Ramen Restaurant    Chinese Restaurant
4          Chūō      Sushi Restaurant   Japanese Restaurant
5       Edogawa      Ramen Restaurant    Italian Restaurant
6      Itabashi    Chinese Restaurant            Restaurant
7    Katsushika    Donburi Restaurant      Ramen Restaurant
8          Kita      Ramen Restaurant   Japanese Restaurant
9          Kōtō    Chinese Restaurant            Restaurant
10       Meguro    Chinese Restaurant   Japanese Restaurant
11       Minato       Soba Restaurant   Yakitori Restaurant
12       Nakano      Ramen Restaurant    Chinese Restaurant
13       Nerima    Chinese Restaurant    Yoshoku Restaurant
14     Setagaya      Ramen Restaurant      Unagi Restaurant
15      Shibuya    Chinese Restaurant   Japanese Restaurant
16    Shinagawa   Japanese Restaurant    Donburi Restaurant
17     Shinjuku   Japanese Restaurant       Thai Restaurant
18     Suginami      Ramen Restaurant    Italian Restaurant
19       Sumida    Chinese Restaurant   Japanese Restaurant
20        Taitō   Japanese Restaurant    Chinese Restaurant
21      Toshima      Ramen Restaurant   Japanese Restaurant
22          Ōta      Ramen Restaurant    Chinese Restaurant

        3rd Most Common Venue   4th Most Common Venue  \
0          Yoshoku Restaurant       Hotpot Restaurant
1           Indian Restaurant      Donburi Restaurant
2         Szechuan Restaurant      Yoshoku Restaurant
3    Japanese Curry Restaurant       Sushi Restaurant
4          Italian Restaurant         Soba Restaurant
5          Yoshoku Restaurant       Hotpot Restaurant
6          Italian Restaurant      Yoshoku Restaurant
```

7	Dumpling Restaurant	Yoshoku Restaurant
8	Italian Restaurant	Chinese Restaurant
9	Indian Restaurant	Yoshoku Restaurant
10	Italian Restaurant	Sushi Restaurant
11	Indian Restaurant	Kosher Restaurant
12	Tonkatsu Restaurant	Donburi Restaurant
13	Hotpot Restaurant	Kosher Restaurant
14	Japanese Restaurant	Szechuan Restaurant
15	Ramen Restaurant	Mexican Restaurant
16	Restaurant	Italian Restaurant
17	Yakitori Restaurant	Yoshoku Restaurant
18	Soba Restaurant	Shabu-Shabu Restaurant
19	Unagi Restaurant	Tonkatsu Restaurant
20	Nabe Restaurant	Sushi Restaurant
21	Chinese Restaurant	Yoshoku Restaurant
22	Japanese Restaurant	Tonkatsu Restaurant

	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	\
0	Korean Restaurant	Kebab Restaurant	Kaiseki Restaurant	
1	Japanese Restaurant	Italian Restaurant	Yoshoku Restaurant	
2	Hotpot Restaurant	Korean Restaurant	Kebab Restaurant	
3	Italian Restaurant	Indian Restaurant	Yoshoku Restaurant	
4	Unagi Restaurant	German Restaurant	Tonkatsu Restaurant	
5	Korean Restaurant	Kebab Restaurant	Kaiseki Restaurant	
6	Hotpot Restaurant	Korean Restaurant	Kebab Restaurant	
7	Indian Restaurant	Korean Restaurant	Kebab Restaurant	
8	Donburi Restaurant	Dumpling Restaurant	Kushikatsu Restaurant	
9	Hotpot Restaurant	Korean Restaurant	Kebab Restaurant	
10	Yoshoku Restaurant	Hotpot Restaurant	Korean Restaurant	
11	Korean Restaurant	Chinese Restaurant	Kebab Restaurant	
12	Italian Restaurant	Yoshoku Restaurant	Indian Restaurant	
13	Korean Restaurant	Kebab Restaurant	Kaiseki Restaurant	
14	Restaurant	Fast Food Restaurant	Yoshoku Restaurant	
15	South Indian Restaurant	Asian Restaurant	Brazilian Restaurant	
16	Sushi Restaurant	Indian Restaurant	Soba Restaurant	
17	Unagi Restaurant	Tonkatsu Restaurant	Chinese Restaurant	
18	Asian Restaurant	Chinese Restaurant	Tonkatsu Restaurant	
19	Sushi Restaurant	Ramen Restaurant	Yoshoku Restaurant	
20	Sukiyaki Restaurant	Soba Restaurant	Unagi Restaurant	
21	Dongbei Restaurant	Yakitori Restaurant	Middle Eastern Restaurant	
22	Yoshoku Restaurant	Sushi Restaurant	Japanese Curry Restaurant	

	8th Most Common Venue	9th Most Common Venue	\
0	Japanese Curry Restaurant	Italian Restaurant	
1	Korean Restaurant	Kebab Restaurant	
2	Kaiseki Restaurant	Japanese Curry Restaurant	
3	Tempura Restaurant	Hotpot Restaurant	

4	Yakitori Restaurant	Tempura Restaurant
5	Japanese Restaurant	Japanese Curry Restaurant
6	Kaiseki Restaurant	Japanese Restaurant
7	Kaiseki Restaurant	Japanese Restaurant
8	Yoshoku Restaurant	Korean Restaurant
9	Kaiseki Restaurant	Japanese Restaurant
10	Kebab Restaurant	Kaiseki Restaurant
11	Kaiseki Restaurant	Japanese Restaurant
12	Korean Restaurant	Kebab Restaurant
13	Japanese Restaurant	Japanese Curry Restaurant
14	Hotpot Restaurant	Kebab Restaurant
15	Donburi Restaurant	Seafood Restaurant
16	Yoshoku Restaurant	Hotpot Restaurant
17	Sushi Restaurant	Hotpot Restaurant
18	Dumpling Restaurant	Yoshoku Restaurant
19	Hotpot Restaurant	Kebab Restaurant
20	Monjayaki Restaurant	Italian Restaurant
21	Seafood Restaurant	Soba Restaurant
22	Italian Restaurant	Soba Restaurant

	10th Most Common Venue
0	Indian Restaurant
1	Kaiseki Restaurant
2	Italian Restaurant
3	Kebab Restaurant
4	Donburi Restaurant
5	Indian Restaurant
6	Japanese Curry Restaurant
7	Japanese Curry Restaurant
8	Kebab Restaurant
9	Japanese Curry Restaurant
10	Japanese Curry Restaurant
11	French Restaurant
12	Kaiseki Restaurant
13	Italian Restaurant
14	Kaiseki Restaurant
15	African Restaurant
16	Kebab Restaurant
17	Soba Restaurant
18	Kebab Restaurant
19	Kaiseki Restaurant
20	Yoshoku Restaurant
21	Korean Restaurant
22	Korean Restaurant

0.3.1 Clustering

[111]:
```
kclusters = 5

onehot_df_clustering = onehot_df.drop('Neighborhood', 1)

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

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

[111]:
```
array([2, 1, 4, 1, 4, 1, 0, 1, 1, 0], dtype=int32)
```

[112]:
```
neighborhoods_venues.insert(0, 'Cluster Labels', kmeans.labels_)

Tokyo_merged = df

Tokyo_merged.rename(columns={'Name':'Neighborhood'}, inplace=True)
Tokyo_merged = Tokyo_merged.join(neighborhoods_venues.
 ↪set_index('Neighborhood'), on='Neighborhood')

Tokyo_merged.head()
```

[112]:
```
  No. Neighborhood Kanji  Population  Density   Area   Latitude   Longitude  \
0  01      Chiyoda            59441     5100  11.66  35.693810  139.753216
1  02         Chūō           147620    14460  10.21  35.666255  139.775565
2  03       Minato           248071    12180  20.37  35.643227  139.740055
3  04     Shinjuku           339211    18620  18.22  35.693763  139.703632
4  05       Bunkyō           223389    19790  11.29  35.718810  139.744732

   Cluster Labels 1st Most Common Venue 2nd Most Common Venue  \
0               1      Ramen Restaurant    Chinese Restaurant
1               4      Sushi Restaurant   Japanese Restaurant
2               4       Soba Restaurant   Yakitori Restaurant
3               4   Japanese Restaurant       Thai Restaurant
4               4    Chinese Restaurant   Japanese Restaurant

       3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
0  Japanese Curry Restaurant      Sushi Restaurant    Italian Restaurant
1         Italian Restaurant       Soba Restaurant      Unagi Restaurant
2          Indian Restaurant     Kosher Restaurant     Korean Restaurant
3        Yakitori Restaurant    Yoshoku Restaurant      Unagi Restaurant
4        Szechuan Restaurant    Yoshoku Restaurant     Hotpot Restaurant

  6th Most Common Venue 7th Most Common Venue 8th Most Common Venue  \
0     Indian Restaurant    Yoshoku Restaurant    Tempura Restaurant
```

```
1     German Restaurant   Tonkatsu Restaurant   Yakitori Restaurant
2    Chinese Restaurant      Kebab Restaurant    Kaiseki Restaurant
3   Tonkatsu Restaurant    Chinese Restaurant      Sushi Restaurant
4     Korean Restaurant      Kebab Restaurant    Kaiseki Restaurant

       9th Most Common Venue 10th Most Common Venue
0          Hotpot Restaurant       Kebab Restaurant
1         Tempura Restaurant     Donburi Restaurant
2        Japanese Restaurant      French Restaurant
3          Hotpot Restaurant        Soba Restaurant
4  Japanese Curry Restaurant     Italian Restaurant
```

[114]:

```
# create map
map_restaurants = folium.Map(location=[latitude,longitude],␣
 ↪tiles='cartodbpositron',
                              attr="<a href=https://github.com/
 ↪python-visualization/folium/>Folium</a>")

# 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]
#rainbow = ['#00ff00', '#ff00ff','#0000ff','#ffa500' ,'#ff0000']
#Districts = ['Nagatacho', 'Nihonbashi', 'Shinjuku', 'Shinagawa', 'Shibuya']

# add markers to the map
for lat, lon, poi, cluster in zip(Tokyo_merged['Latitude'],
                                  Tokyo_merged['Longitude'],
                                  Tokyo_merged['Neighborhood'],
                                  Tokyo_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=list_rest_no[list_dist.index(poi)]*0.5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_restaurants)

map_restaurants
```

[114]: <folium.folium.Map at 0x7f9899c73790>

0.4 Results

From the analysis above, we conclude the following results: 1. Chuo ward and Ota ward are two wards that have most restaurants 2. Most restaurants in Tokyo are divided into 2 clusters. 3. Ramen restaurants are the most common venue in Tokyo.

0.5 Discussion

Based on our observations above, most restaurants are in cluster 1 and cluster 4, which indicates that people are most likely to go to restaurants in these areas. However, there still are some shortcomings in our analysis since we only take restaurants categories such as ramen restaurants, Chinese restaurants into consideration. Other factors such as restaurants' prices, people's salaries are also important for us to do the analysis.

0.6 Conclusion

In this example, we use geospatial data of Tokyo to cluster neighborhoods based on the most common restaurants. The final results help people choose restaurants more easily. As we can see, many real-life cases could be solved by using data analysis. In this example, however, besides the frequency of restaurants categories we chose, many other factors should be taken into consideration in order to conclude a more comprehensive result.