The Best special wards of Tokyo to open a new restaurant

Tsukasa Sugiura

July 21, 2011

Introduction

Tokyo is the biggest city of Japan. As of 2021, Tokyo has an estimated population of 13,960,236. In addition, Tokyo is the political and economic center of Japan. In the coronavirus crisis, Many Japanese can't eat out. Also, quite a few restaurants have closed. But, after calming down this crisis, it is expected that people return to downtown to eat out.

Business Problem

Business Problem is to open a restaurant in Tokyo after after calming down the coronavirus crisis. The special wards(There are 23.) are the most crowded in Tokyo. Tokyo special wards are most likely to give a good business.

Data acquisition and cleaning

Data sources

To get data of special wards of Tokyo, I used the following Wikipedia page.

https://en.wikipedia.org/wiki/Special_wards_of_Tokyo

Next, I got Latitude and Longitude of each ward using Geopy library. Finally, I got venue data in Tokyo from Foursquare using Foursquare API.

```
In [63]: html = urlopen("https://en.wikipedia.org/wiki/Special_wards_of_Tokyo")
         html_parser = BeautifulSoup(html, "html.parser")
```

Data cleaning

I did web scraping to wikipedia page by Beautiful Soup. To get the wards table from the wikipedia page, I found the name of table class, and extracted the elements between “td' and “'th”.

```
In [65]: table = html_parser.findAll("table", {"class":"wikitable sortable"})[0]
         rows = table.findAll("tr")

         with open("tokyo.csv", "w", encoding='utf-8') as file:
             writer = csv.writer(file)
             for row in rows:
                 csvRow = []
                 for cell in row.findAll(['td', 'th']):
                     csvRow.append(cell.get_text())
                 writer.writerow(csvRow)
```

Feature selection

Next, I did data processing and tabulation by pandas Like below.

-Tabulation the elements of the table of wikipedia page
-Getting latitude and longitude of each ward by geocoder
-Plotting of each ward by Folium
-Getting venu information of each ward by Foursquare API
-making CSV file from the table.

```
In [70]: tokyo_ward = tokyo_data.drop(23)
         tokyo_ward = tokyo_ward.rename(columns={'Name¥n': 'Neighbourhood'})
         tokyo_ward
```

```
In [71]: lat = []
         lng = []
         lat_lng_coords = None

         neighbourhoods = tokyo_ward['Neighbourhood']

         for nh in neighbourhoods:
             g = geocoder.arcgis('{}, Tokyo, JP'.format(nh))
             lat_lng_coords = g.latlng
             lat.append(lat_lng_coords[0])
             lng.append(lat_lng_coords[1])
```

	Neighbourhood	Latitude	Longitude
0	Chiyoda	35.693930	139.753711
1	Chūō	35.670572	139.771988
2	Minato	35.658017	139.751546
3	Shinjuku	35.693798	139.703440
4	Bunkyō	35.707595	139.752210

In [87]:
```
tokyo_map = folium.Map(location=[latitude, longitude], zoom_start=11)

for lat, lng, label in zip(tokyo_geo['Latitude'], tokyo_geo['Longitude'], tokyo_geo['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(tokyo_map)

tokyo_map
```

In [86]:
```
from geopy.geocoders import Nominatim

address = 'Tokyo, JP'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Tokyo are {}, {}.'.format(latitude, longitude))
```

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

```
In [64]: os.chdir('/tmp')
         path = 'tokyo.csv'
         csv_file = open(path, 'w')
         csv_writer = csv.writer(csv_file)
```

Exploratory Data Analysis

Then, I did data analysis by pandas Like below.

-Making the Top 10 of the venue categories to each ward.

-Making tables of each ward include top 10 of the venue categories

At this point, I found that there must be some area with many restaurants.

	Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue Name	Venue Category	Venue Latitude	Venue Longitude
0	Chiyoda	35.69393	139.753711	Kanda Tendonya (神田天丼家)	Tempura Restaurant	35.695765	139.754682
1	Chiyoda	35.69393	139.753711	Bondy (欧風カレー ボンディ)	Japanese Curry Restaurant	35.695544	139.757356
2	Chiyoda	35.69393	139.753711	Jimbocho Kurosu (神保町 黒須)	Ramen Restaurant	35.695539	139.754851
3	Chiyoda	35.69393	139.753711	National Museum of Modern Art (東京国立近代美術館)	Art Museum	35.690541	139.754694
4	Chiyoda	35.69393	139.753711	Warayakiya (わらやき屋)	Sake Bar	35.696017	139.751388

	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	Adachi	Convenience Store	Supermarket	Discount Store	Drugstore	Grocery Store	BBQ Joint	Noodle House	Ramen Restaurant	Furniture / Home Store	Pizza Place
1	Arakawa	Ramen Restaurant	Park	BBQ Joint	Supermarket	Japanese Restaurant	Grocery Store	Drugstore	Sandwich Place	Discount Store	Deli / Bodega
2	Bunkyō	Hotel	Baseball Stadium	Martial Arts School	Supermarket	Café	Seafood Restaurant	Pastry Shop	Chinese Restaurant	History Museum	Ramen Restaurant
3	Chiyoda	Café	Ramen Restaurant	Japanese Curry Restaurant	BBQ Joint	Sushi Restaurant	Tea Room	Coffee Shop	Comedy Club	Historic Site	Sake Bar
4	Chūō	Ramen Restaurant	Soba Restaurant	Italian Restaurant	Tonkatsu Restaurant	Sushi Restaurant	Coffee Shop	Yoshoku Restaurant	Juice Bar	Steakhouse	Burger Joint

Predictive Modeling

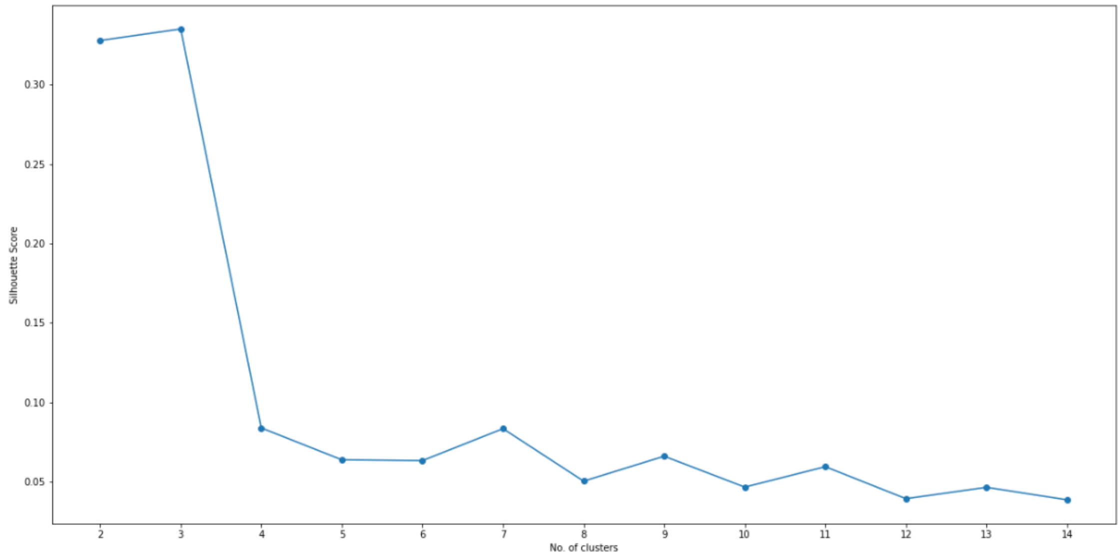
I selected “Clustering by K-Means” as “Unsupervised Learning Model”.

Result of the elbow method , I found that the suitable number of clusters is 4.

Then, I Vidualized of the clusters by Folium.

And I did One hot encoding by get dummies.

In [107]:
```python
max_range = 15

from sklearn.metrics import silhouette_samples, silhouette_score

indices = []
scores = []

for tokyo_clusters in range(2, max_range) :

    tokyo_gc = tokyo_grouped_clustering
    kmeans = KMeans(n_clusters = tokyo_clusters, init = 'k-means++', random_state = 0).fit_predict(tokyo_gc)

    score = silhouette_score(tokyo_gc, kmeans)

    indices.append(tokyo_clusters)
    scores.append(score)
```



	Neighbourhood	African Restaurant	American Restaurant	Arcade	Art Museum	Asian Restaurant	Athletics & Sports	BBQ Joint	Bakery	Bar	Baseball Stadium	Bath House	Bed & Breakfast	Beer Bar	Beer Garden	Bistro	Boarding House	Bookstore
0	Chiyoda	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	(
1	Chiyoda	0	0	0	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	0	0	0	(
3	Chiyoda	0	0	0	1	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	0	0	0	(

Discussion

Closely examining the contents(Top 10 categories of the restaurant business) of each cluster, it is clear between clusters that a big difference exists.
Then I found that Cluster 4 is thought to be the most appropriate place to open the restaurant business because there are many restaurant categories in the ward.
Especially, Chuo and Shinjuku, These wards are populous areas in Cluster 4, look like good locations for open a new restaurant.

	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
1	Chūō	Ramen Restaurant	Soba Restaurant	Italian Restaurant	Tonkatsu Restaurant	Sushi Restaurant	Coffee Shop	Yoshoku Restaurant	Juice Bar	Steakhouse	Burger Joint
2	Minato	Japanese Restaurant	Ramen Restaurant	Historic Site	BBQ Joint	Tonkatsu Restaurant	Liquor Store	Buddhist Temple	Scenic Lookout	Soba Restaurant	Kaiseki Restaurant
6	Sumida	Japanese Restaurant	Café	Ramen Restaurant	Sukiyaki Restaurant	Soba Restaurant	Bakery	Unagi Restaurant	Park	Deli / Bodega	Buddhist Temple
7	Kōtō	Ramen Restaurant	Café	French Restaurant	BBQ Joint	Convenience Store	Park	Climbing Gym	Discount Store	Tonkatsu Restaurant	Hotel
11	Setagaya	Ramen Restaurant	Soba Restaurant	Sake Bar	Japanese Restaurant	Café	Candy Store	Indian Restaurant	Supermarket	Cupcake Shop	Szechuan Restaurant
14	Suginami	Ramen Restaurant	Sake Bar	BBQ Joint	Thai Restaurant	Wagashi Place	Café	Italian Restaurant	Imported Food Shop	Music Venue	Indian Restaurant
16	Kita	Ramen Restaurant	Café	Park	Sake Bar	Museum	Theater	Drugstore	Fried Chicken Joint	Garden	Convenience Store
17	Arakawa	Ramen Restaurant	Park	BBQ Joint	Supermarket	Japanese Restaurant	Grocery Store	Drugstore	Sandwich Place	Discount Store	Deli / Bodega
18	Itabashi	Ramen Restaurant	Sake Bar	Yoshoku Restaurant	Café	Steakhouse	Udon Restaurant	French Restaurant	Deli / Bodega	Sushi Restaurant	Coffee Shop

Conclusion

Web Scraping by Beautiful Soup is very helpful to gather data for data analysis. But there seems to be lots of websites that it is hard to do webscraping.
Web API such as Foursquare is very valuable for data scientist. It is very easy and effective to extract data that they have.
Also, data analysys and machine learning by python can be very helpful in determining solutions of certain business problems, Python's inbuilt libraries such as Pandas, Geopy, Folium make it very simple for data scientist to develop programs. Abobe all, sklearn.cluster is very helpful for me to develop statistics programs.
I had a hard time to solve errors by the version difference of libraries. I felt the need of performing enough preparations including the learning about the necessary library before programming. Do not program it immediately, We should take time to prepare.