Backpacker to Tokyo

Where to Stay When You Travel to Tokyo?

Miftachul Umam

December 5, 2020

1. Introduction

1.1 Background

In this project, I will try to find an optimal location to stay on vacation. Specifically, this report will be targeted to low budget travelers or backpackers are interested in a vacation to Tokyo Japan. Multinational consulting firm Mercer's 2020 Cost of Living Survey has determined that Tokyo comes in third for the most expensive city in the world. We need advice before going on vacation to Tokyo.

According to www.travelandleisure.com Tokyo is one of the most well-connected cities on the planet in terms of transportation. The best time to visit Tokyo is in the fall and spring when temperatures are temperate and the scenery is stunning. Both cherry blossom season and autumn foliage season are excellent times to visit. Many festivals take place during July when Mount Fuji is also open for climbing.

Tokyo is a dizzying whirl of activity: you can practically feel its pulse, with the neon signs, the crush of people, and the perfectly punctual, high-speed trains whizzing by. Then there's a pause, a bit of green and calm on the city's temple grounds or classical gardens.

Navigating the world's largest metropolis—home to more than 13 million people—can be a daunting prospect for visitors. Tokyo's maze of neighborhoods seems to offer up every imaginable sight and sound—some of them cacophonous and modern (speeding bullet trains; herds of hurrying, be-suited businessmen; bizarrely futuristic toilets), and some of the ancient (Buddhist shrines and temples; the waddling combat of sumo wrestlers). The trick here is to explore one enclave at a time; for instance, starting in Ginza or Shibuya for shopping, then heading to Shinjuku or Roppongi for nightlife.

Since there are lots of Hotel in Tokyo we will try to detect **the best price of hotels** based on http://insideairbnb.com/

We will use our data science powers to generate a few most promising neighborhoods based on these criteria. The advantages of each area will then be clearly expressed so that the best possible final location can be chosen by the backpacker.

1.2 Problem

Using data science methodology and machine learning techniques like clustering, this project aims to provide solutions to answer the question: In the city of Tokyo, if a traveler is looking to stay on vacation, where would you recommend that they stay?

2. Data

2.1 Data Source

Based on the definition of our problem, the factors that will influence our decision are:

1. The list data of neighborhoods.
2. Latitude and longitude coordinates of those neighborhoods. This is required to plot the map and also to get the venue data.
3. Dataset of rating and price of existing hotels in the neighborhoods. We will use this data to perform clustering on the neighborhoods.
4. Number of venues including tourist attractions, restaurants, and shopping malls in the neighborhood.

2.2 Data Extract

Following data sources will be needed to extract/generate the required information:

I will get the list of neighborhoods in Tokyo from Wikipedia. I will use web scraping techniques to extract the data from the Wikipedia page, with the help of Python requests and pandas packages. Then we can get the latitude and longitude coordinates of the neighborhoods using the Python Geocoder package.

Airbnb Inside will provide many categories of rating and price of an existing hotel in the neighborhoods and Foursquare API will provide many categories of the venue data, and we are particularly interested in the tourist attractions category to help us clustering the neighborhoods. This project will make us use many of data science skills, from web scraping (Wikipedia), working with API (Foursquare), data cleaning, data wrangling, to machine learning (K-means clustering) and map visualization (Folium).

3. Methodology

3.1 Perform scraping using Python requests and beautifulsoup packages to extract the list of neighborhood data.

3.2 Download data inside from Airbnb for getting the data of accommodations, Then, we will randomly choose 1000 accommodations. In this case, I got other data of neighborhoods name by Airbnb that different from the Wikipedia, then I consider using Airbnb neighborhood data because I have struggled to merging based on haversine formula to merging the Wikipedia's data and Airbnb's data. I don't clean the data of the neighborhood based on Wikipedia data for ideation the next project.

3.3 I will use Foursquare API to obtain information on venues nearby to our top 1000 accommodations. In our final step, we will cluster our accommodations with k-means clustering and provide recommendations to travelers.

3.4 Using The Foursquare API allows application developers to interact with the Foursquare platform. The API itself is a RESTful set of addresses to which you can send requests, so there's nothing to download onto your server.

4. Exploritory Data Analysis

4.1 Let's scraping the neighborhood data from Wikipedia

```
# Send the GET request
data = requests.get("https://en.wikipedia.org/wiki/Category:Neig
hborhoods_of_Tokyo").text
# Parse data from the html into a beautifulsoup object
soup = BeautifulSoup(data, 'html.parser')
# Create a list to store neighbourhood data
neighborhoodList = []
# Append the data into the list
for row in soup.find_all("div", id="mw-pages")[0].findAll("li"):
  neighborhoodList.append(row.text)
# Create a new DataFrame from the list
Tokyo_df = pd.DataFrame({"Neighborhood": neighborhoodList})
Tokyo_df
```

	Neighborhood
0	Agariyashiki
1	Akihabara
2	Aoyama, Tokyo
3	Arai, Tokyo
4	Asagaya
...	...
97	Yotsuya
98	Yoyogi
99	Yoyogikamizonochō
100	Yūrei zaka
101	Zōshigaya

102 rows × 1 columns

4.2 Let's get the geographical coordinates of the neighborhoods using the Geocoder package

In [4]:

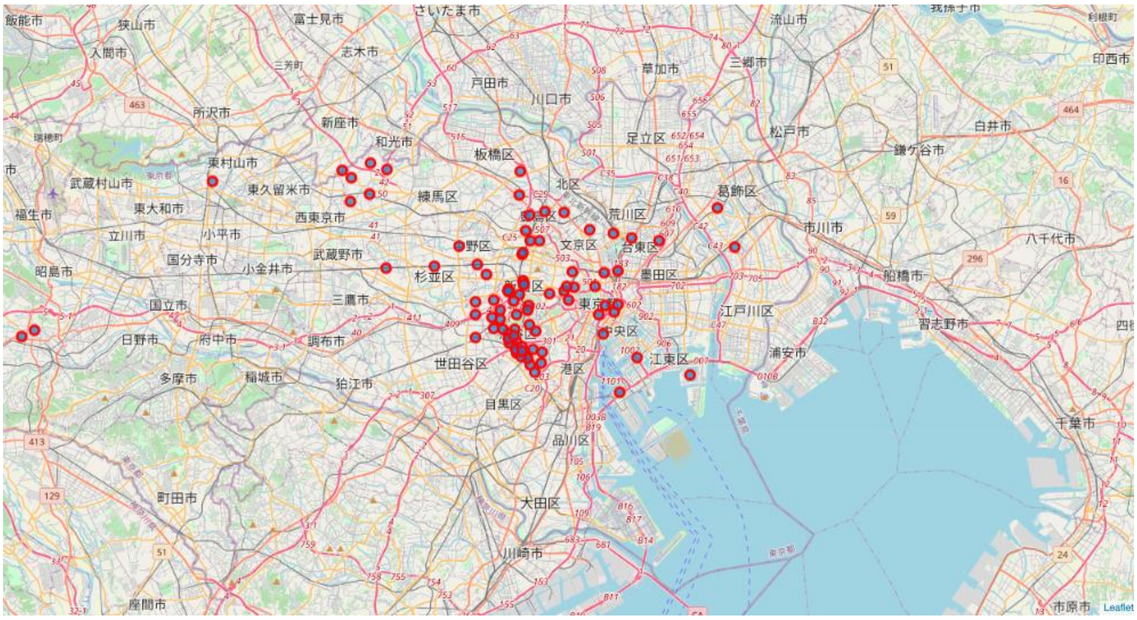
```
# Create temporary dataframe to populate the coordinates into La
titude and Longitude
df_coords = pd.DataFrame(coords, columns=['Latitude', 'Longitud
e'])
# Merge the coordinates into the original dataframe
Tokyo_df['Latitude'] = df_coords['Latitude']
Tokyo_df['Longitude'] = df_coords['Longitude']
print(Tokyo_df.shape)
Tokyo_df.head()
```

```
(102, 3)
```

Out[4]:

	Neighborhood	Latitude	Longitude
0	Agariyashiki	35.726462	139.705156
1	Akihabara	35.702171	139.774409
2	Aoyama, Tokyo	35.689456	139.691716
3	Arai, Tokyo	35.689456	139.691716
4	Asagaya	35.704890	139.636260

4.3 Let's Visualisation the Map of Tokyo



4.4 We can get the Airbnb Inside Data from http://insideairbnb.com/

we can get the Airbnb Inside Data from http://insideairbnb.com/

In [7]:
```
df_hotel=open('listings.csv')
df_hotel = pd.read_csv(df_hotel)
df_hotel.head()
```

Out[7]:

	id	name	host_id	host_name	neighbourhood_group	neighbourh
0	197677	Oshiage Holiday Apartment	964081	Yoshimi & Marek	NaN	Sumida Ku
1	776070	Kero-kero house room 1	801494	Kotaro Kei And Litte Miya	NaN	Kita Ku
2	899003	Classy room @Shinjuku, Takadanoba	4799233	Yu	NaN	Shinjuku Ku
3	905944	4F - Near Shinjuku & Shibuya w/Free WiFi	4847803	Best Stay In Tokyo!	NaN	Shibuya Ku
4	1016831	WOMAN ONLY LICENSED ! Cosy & Cat behnd Shibuya	5596383	Wakana	NaN	Setagaya Ku

4.5 Descriptive Statistic of Airbnb Data

In [9]:
```
Tokyo_Hotel_Dec=Tokyo_Hotel.describe()
Tokyo_Hotel_Dec
```

Out[9]:

	latitude	longitude	price
count	11715.000000	11715.000000	1.171500e+04
mean	35.698565	139.737132	1.090098e+04
std	0.042028	0.071649	3.035951e+04
min	35.540520	139.118472	7.370000e+02
25%	35.686730	139.700865	3.671000e+03
50%	35.703790	139.732300	6.000000e+03
75%	35.723790	139.786280	1.136400e+04
max	35.832220	139.911430	1.035714e+06

4.6 I Also Get Data of Neighborhood Name by Airbnb Data

I have to consider using Airbnb data or Wikipedia? I tried using the haversine formula to merge the data but fail, and I choose using Airbnb neighborhood data for the next analysis.

Wikipedia Data	Airbnb Data

Wikipedia Data

In [13]:
```
# Recall the Wikipedia data
Tokyo_df.head(10)
```

Out[13]:

	Neighborhood	Latitude	Longitude
0	Agariyashiki	35.726462	139.705156
1	Akihabara	35.702171	139.774409
2	Aoyama, Tokyo	35.689456	139.691716
3	Arai, Tokyo	35.689456	139.691716
4	Asagaya	35.704890	139.636260
5	Banchō	35.691973	139.741446
6	Chūō, Nakano, Tokyo	35.699599	139.675060
7	Daikanyamachō, Shibuya	35.650765	139.704683
8	Dōgenzaka (district)	35.658362	139.697913
9	Ebisu, Shibuya	35.645326	139.716505

Airbnb Data

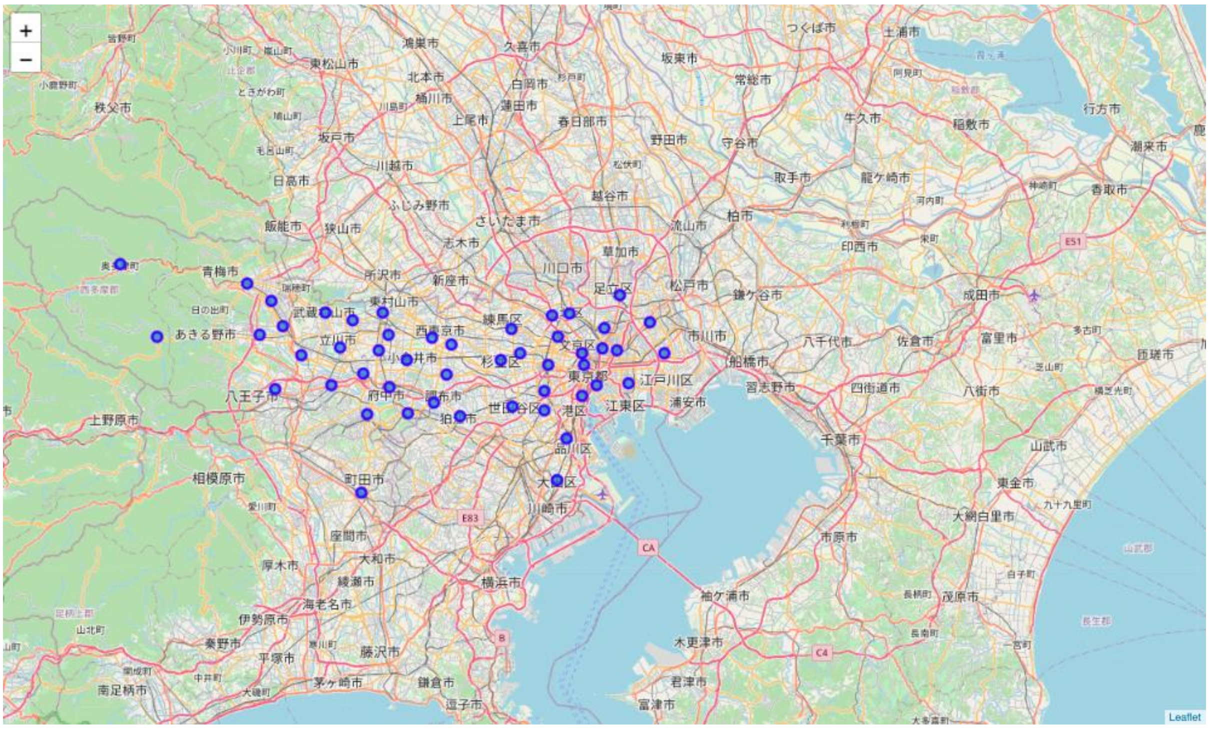
In [12]:
```
Neighbourhood_airbnb.sort_values('neighbourhood').head(10)
```

Out[12]:

	neighbourhood
5	Adachi Ku
31	Akiruno Shi
13	Akishima Shi
18	Arakawa Ku
14	Bunkyo Ku
32	Chiyoda Ku
38	Chofu Shi
21	Chuo Ku
17	Edogawa Ku
34	Fuchu Shi

Based on the comparison the Wikipedia and Airbnb, the list names neighborhood is **different**. I have tried using the haversine formula to merging the two data, but I have struggled then I consider choosing neighborhood **data by Airbnb** for a simple way. Generally speaking, people can search for this location in google maps when they want to know the location.

4.7 Neighborhood Visualisation (Airbnb Data)



4.8 The 1000 Cheapest Hotel in Tokyo

Cheapest 1000 Hotels

In [17]:
```
Hotel_cheapest=Tokyo_Hotel.sort_values('price').head(1000)
Hotel_cheapest
```

Out[17]:

	name	neighbourhood	latitude	longitude	price
1969	Poket Wi-Fi 509 GK	Sumida Ku	35.69405	139.81376	737
1958	Poket Wi-Fi 402 GK	Sumida Ku	35.69506	139.81293	810
1966	Poket Wi-Fi 506 GK	Sumida Ku	35.69485	139.81236	840
1961	Poket Wi-Fi 501 GK	Sumida Ku	35.69388	139.81217	880
1963	Poket Wi-Fi 503 GK	Sumida Ku	35.69540	139.81297	880
...	...	...	...	...	...
8685	上野附近温馨独立公寓101室 Ueno Nearby Cozy Apartment 101	Taito Ku	35.72538	139.78261	2600
8103	Koenji Station walk 3	Suginami Ku	35.70220	139.64919	2600
8101	3 minutes on foot from Koenji	Suginami Ku	35.70395	139.64907	2600
2627	[C3]near to Shinjuku,Shibuya,Ueno etc,24H checkIn	Kita Ku	35.73874	139.74878	2600
8108	Koenji Station walk 3	Suginami Ku	35.70353	139.65078	2600

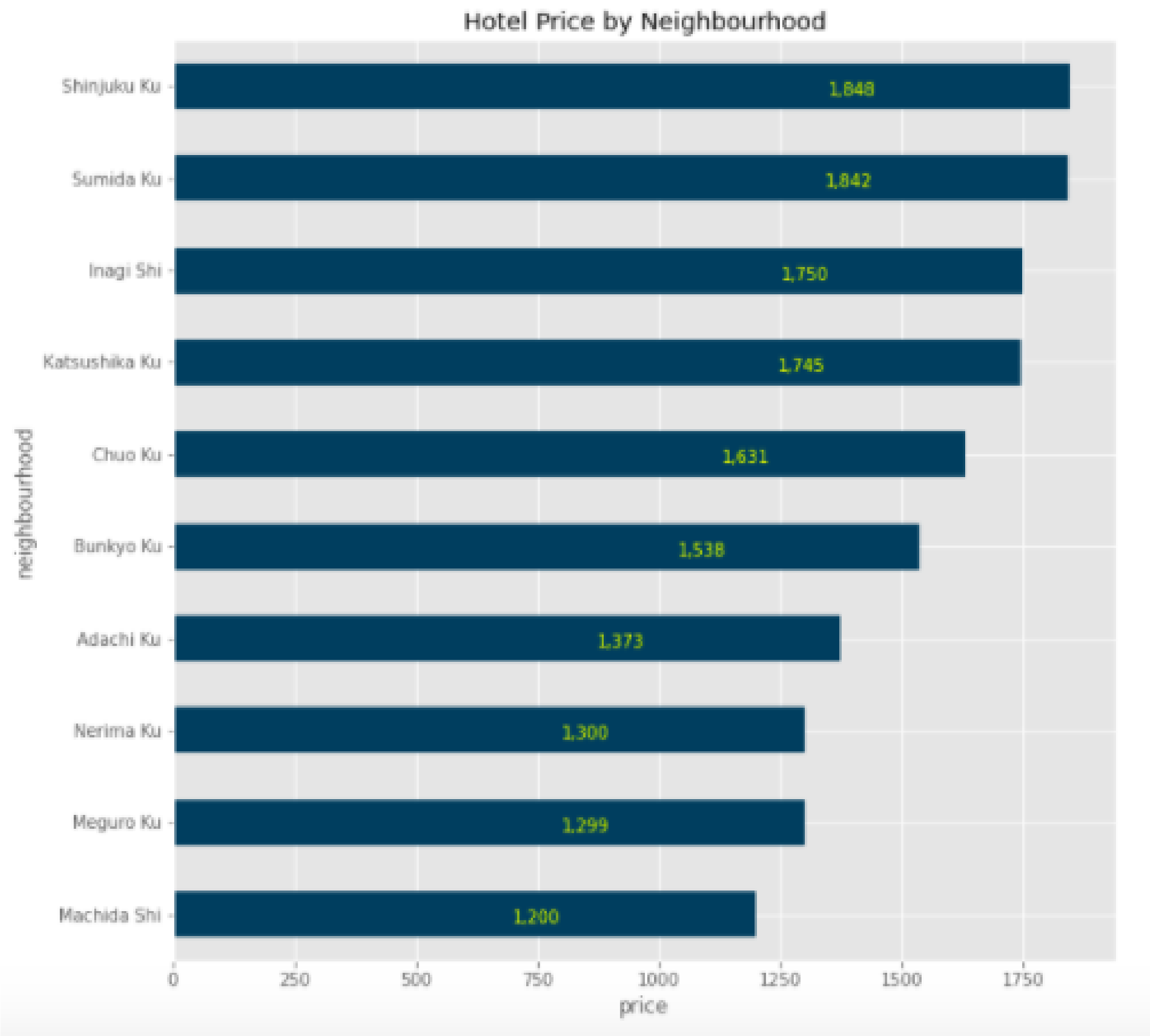
1000 rows × 5 columns

In [18]:
```
Hotel_cheapest.describe()
```

Out[18]:

	latitude	longitude	price
count	1000.000000	1000.000000	1000.000000
mean	35.710002	139.750787	1949.230000
std	0.039965	0.076328	501.826234
min	35.542580	139.372870	737.000000
25%	35.695422	139.703520	1500.000000
50%	35.707160	139.770305	2018.500000
75%	35.735172	139.802380	2418.000000
max	35.810410	139.904810	2600.000000

4.9 Bar chart of Cheapest Hotel by Neighborhood



5. Analysis

5.1 Hotel Clustering Based On Price

There are some types of models, I used clustering that can be used to location segmentation . **Cluster analysis** or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (clusters). This project use K-means Algorithm for analysis.

Kmeans algorithm is an iterative algorithm that tries to partition the dataset into *K*pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

The way kmeans algorithm works is as follows:

1. Specify number of clusters *K*.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn’t changing.

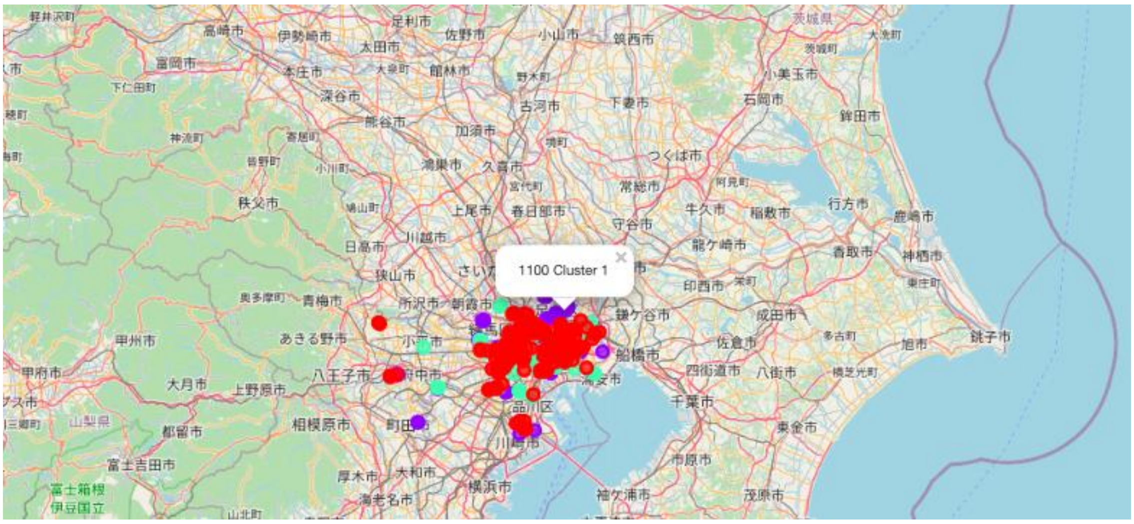
Here the result of K-means clustering data analysis :

1. I set three number of clusters K

In [27]:
```
Hotel_cheapest["Labels"] = labels
Hotel_cheapest.head(10)
```

Out[27]:

	name	neighbourhood	latitude	longitude	price	Labels
1969	Poket Wi-Fi 509 GK	Sumida Ku	35.69405	139.81376	737	1
1958	Poket Wi-Fi 402 GK	Sumida Ku	35.69506	139.81293	810	1
1966	Poket Wi-Fi 506 GK	Sumida Ku	35.69485	139.81236	840	1
1961	Poket Wi-Fi 501 GK	Sumida Ku	35.69388	139.81217	880	1
1963	Poket Wi-Fi 503 GK	Sumida Ku	35.69540	139.81297	880	1
1962	Poket Wi-Fi 502 GK	Sumida Ku	35.69383	139.81346	910	1
1989	Poket Wi-Fi 401 GK	Sumida Ku	35.69493	139.81418	980	1
10865	Enjoy your private time – Nest Inn Tabata A -	Arakawa Ku	35.73542	139.76562	1071	1
10351	Enjoy your private time - Nest Inn Tabata B -	Kita Ku	35.73620	139.76439	1082	1
1765	8 mins to Station/private room with TV★S3-3	Adachi Ku	35.80921	139.76721	1100	1



You can see a map of the hotel cluster, the purple color is the cheap hotel, the green is medium and the red is the most expensive.

Based on the map, all areas have cheap accommodations. But there are two areas that strongly purple which are namely Machida and Adhaci.

5.2 Hotel Clustering Based On Price

I have used Foursquare API to obtain information on venues nearby to our top 1000 hotels.

ANALYSIS OF NEIGHBORHOOD BASED ON FOURSQUARE API

In [29]:
```
CLIENT_ID = 'LGBK1J3QAKLBF04KM2OUGWUVXE2420DCLLUWPF2SMY3QE4AS' # your Foursquare ID
CLIENT_SECRET = 'VV3QEH2WFLTRM2EXHANYDI0OAB0LOLGBAFDX2ONGMNYRCYDX' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version
LIMIT = 100 # A default Foursquare API limit value

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: LGBK1J3QAKLBF04KM2OUGWUVXE2420DCLLUWPF2SMY3QE4AS
CLIENT_SECRET:VV3QEH2WFLTRM2EXHANYDI0OAB0LOLGBAFDX2ONGMNYRCYDX
```

In our final step, we will cluster our accommodations with k-means clustering and provide recommendations to travellers.

```
In [32]: print(Tokyo_venues.shape)
         Tokyo_venues.head()
```

(2269, 7)

Out[32]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Sumida Ku	35.710707	139.80154	Suke6 Diner / Manufacture	35.712273	139.799319	Café
1	Sumida Ku	35.710707	139.80154	塩パン屋 パン・メゾン	35.708709	139.802399	Bakery
2	Sumida Ku	35.710707	139.80154	MONTEE 13 (モンティ 13)	35.708663	139.801741	Thai Restaurant
3	Sumida Ku	35.710707	139.80154	ポポンデッタ with 東武鉄道ギャラリー	35.711852	139.798468	Hobby Shop
4	Sumida Ku	35.710707	139.80154	Authentique (オーセンティック)	35.711249	139.797825	Vietnamese Restaurant

Analize Venues in Neighbourhood

```
In [35]: # one hot encoding
         Tokyo_onehot = pd.get_dummies(Tokyo_venues[['Venue Category']], prefix="", prefix_sep="")

         # add neighborhood column back to dataframe
         Tokyo_onehot['Neighborhood'] = Tokyo_venues['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()
```

Out[35]:

	Neighborhood	ATM	Accessories Store	African Restaurant	American Restaurant	Aquarium	Arcade	Art Gallery	Art Museum	Arts & Crafts Store	...	Vietnamese Restaurant	Wagashi Place	Whisky Bar	Wine Bar	Wine Shop	Wi Joi
0	Sumida Ku	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
1	Sumida Ku	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
2	Sumida Ku	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
3	Sumida Ku	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
4	Sumida Ku	0	0	0	0	0	0	0	0	0	...	1	0	0	0	0	0

5 rows × 246 columns

```
In [36]: Tokyo_grouped = Tokyo_onehot.groupby('Neighborhood').mean().reset_index()
         Tokyo_grouped
```

I have used one hot encoding and k-means clustering segment the top 1000 hotels inTokyo. I have segmented my data into six clusters. After this, I have examined each cluster and my result of the investigation is shown in the results and discussion section.Below a map plot to visualise the distribution of hotel across Tokyo by clusters.

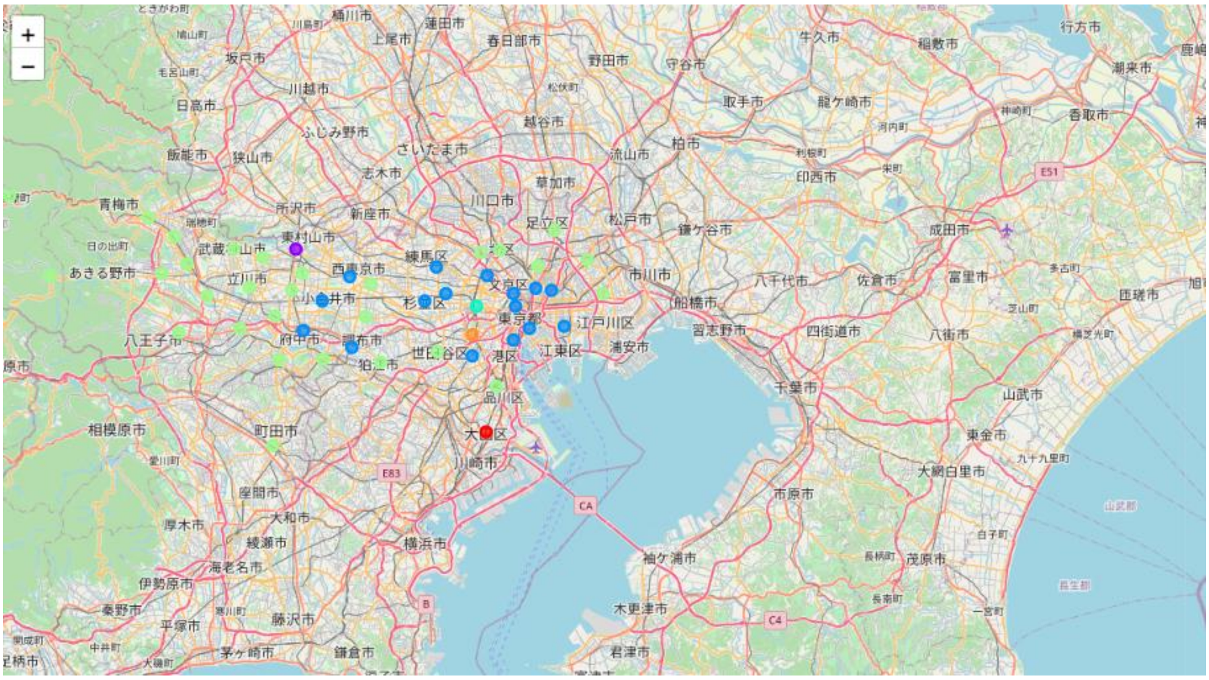
In [41]:
```python
# set number of clusters
kclusters = 6

# 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_[0:40]
```

Out[41]:
```
array([4, 4, 4, 4, 2, 2, 2, 2, 4, 2, 4, 4, 4, 1, 4, 4, 4, 4, 4, 4, 4, 4,
       2, 4, 4, 2, 4, 4, 2, 2, 4, 4, 4, 2, 2, 2, 4, 4, 0, 4], dtype=int32)
```

In [42]:
```python
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

Tokyo_merged = Tokyo_neigh

# merge manhattan_grouped with manhattan_data to add latitude/longitude for each neighborhood
Tokyo_merged = Tokyo_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='neighbourhood')

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

Out[42]:

	neighbourhood	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
0	Sumida Ku	35.710707	139.801540	2	Japanese Restaurant	Convenience Store	Soba Restaurant	Wagashi Place	Café	Unagi Restaurant	Ramen Restaurant	Coffee Shop	Yoshoku Restaurant
1	Kita Ku	35.752839	139.733519	4	Convenience Store	Ramen Restaurant	Intersection	Park	Café	Theater	Japanese Restaurant	Pizza Place	Garden
2	Shinjuku Ku	35.693798	139.703440	3	Sake Bar	Ramen Restaurant	BBQ Joint	Bar	Rock Club	Pub	Japanese Restaurant	Bookstore	Noodle House
3	Shibuya Ku	35.663687	139.697791	5	Café	Record Shop	Nightclub	Rock Club	Chinese Restaurant	Ramen Restaurant	Sake Bar	Concert Hall	BBQ Joint
4	Setagaya Ku	35.646544	139.653222	4	Convenience Store	Café	Intersection	Bakery	Sake Bar	Ramen Restaurant	Soba Restaurant	Tea Room	Bistro



6. Result and Discussion

6.1 I have tried using the neighborhood data from Wikipedia, my assumption the Airbnb data will give me the same data of neighborhood then I could merge these data. Unfortunately, the neighborhood of Wikipedia and Airbnb is different, so I consider to use one of them. I prefer to use Airbnb data. In real life, data scientists have encountered confusing source data.

6.2 Based on the above analysis, the top 10 cheapest neighborhood to stay on vacation in Tokyo are:

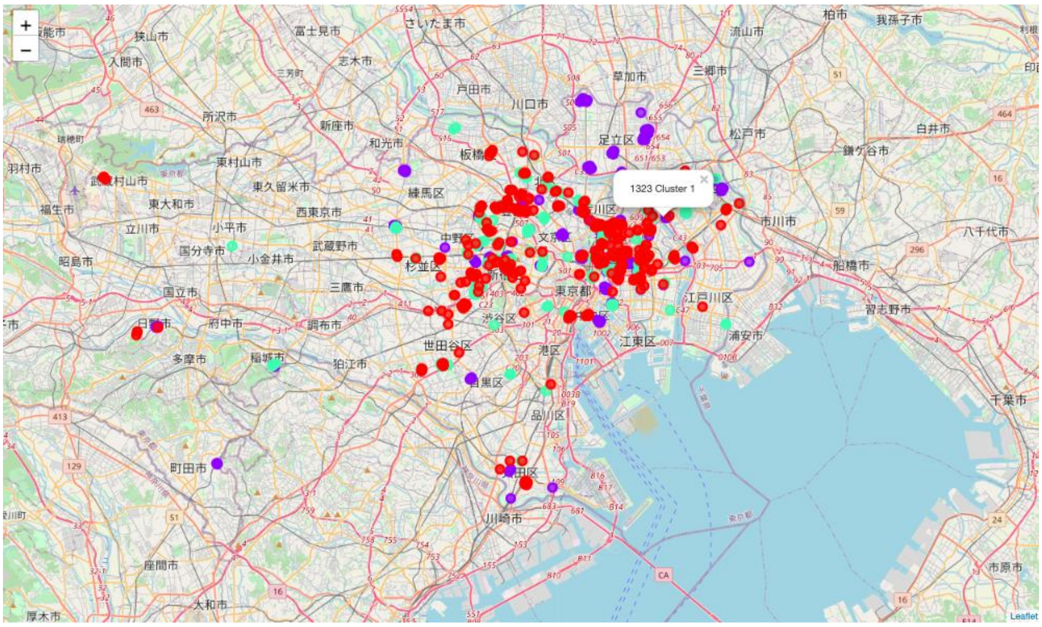
a. Machida Shi 1200
b. Meguro Ku 1299
c. Nerima Ku 1300
d. Adachi Ku 1373
e. Bunkyo Ku 1538
f. Chuo Ku 1631
g. Katsushika Ku 1745
h. Inagi Shi 1750
i. Sumida Ku 1842
j. Shinjuku Ku 1848

From these neighborhoods, Machida and Inagi are the farthest cities from the airport so it's reasonable for the cheap price. Others location is nearby the center of Tokyo, If you want to get easy transportation, Shinjuku is the most popular then others.

Based on Tokyo travel guidance, Shinjuku is a special ward in Tokyo, Japan. It is a major commercial and administrative center, housing the northern half of the busiest railway station in the world (Shinjuku Station) and the Tokyo Metropolitan Government Building, the administration center for the government of Tokyo.Sinjuku is famous for its nightlife.

After stay at Shinjuku, you can move to Sumida. Sumida is famous for top tourist locations such as Japan Skytree, Sensoji, Edo-Tokyo Museum and e.t.c.

6.3 You can see a map of the hotel cluster, the purple color is the cheap hotel, the green is medium and the red is the most expensive.



6.4 I have clustered neighborhoods based on the venues. We observed the venues of cluster 1 until cluster 6, and let us discuss the result: in my opinion:

6.4.1 cluster-1, Ota Ku has characteristic culinary traveling.

```
In [44]: Tokyo_merged.loc[Tokyo_merged['Cluster Labels'] == 0, Tokyo_merged.columns[[0] + list(range(4, Tokyo_merged.shape[1]))]]
```

Out[44]:

	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
30	Ota Ku	Ramen Restaurant	Chinese Restaurant	Japanese Restaurant	Sake Bar	Sushi Restaurant	Bed & Breakfast	Café	Tonkatsu Restaurant	Vietnamese Restaurant	Steakhouse

6.4.2 cluster-2, Higashimurayama Shi has characteristic natural tourism.

```
In [45]: Tokyo_merged.loc[Tokyo_merged['Cluster Labels'] == 1, Tokyo_merged.columns[[0] + list(range(4, Tokyo_merged.shape[1]))]]
```

Out[45]:

	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
20	Higashimurayama Shi	Convenience Store	Japanese Family Restaurant	Sake Bar	Park	Grocery Store	Thrift / Vintage Store	Fish Market	Tunnel	Ramen Restaurant	Supermarket

6.4.3 cluster-3, Sumida Ku, Meguro Ku, and e.t.c have characteristic complete tour.

```
In [46]: Tokyo_merged.loc[Tokyo_merged['Cluster Labels'] == 2, Tokyo_merged.columns[[0] + list(range(4, Tokyo_merged.shape[1]))]]
```

Out[46]:

	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	Sumida Ku	Japanese Restaurant	Convenience Store	Soba Restaurant	Wagashi Place	Café	Unagi Restaurant	Ramen Restaurant	Coffee Shop	Yoshoku Restaurant	Park
7	Meguro Ku	Japanese Restaurant	Ramen Restaurant	BBQ Joint	Café	Italian Restaurant	Chinese Restaurant	Coffee Shop	Bakery	French Restaurant	Japanese Curry Restaurant
8	Toshima Ku	Ramen Restaurant	Café	Japanese Restaurant	BBQ Joint	Sake Bar	Coffee Shop	Steakhouse	Pet Café	Sushi Restaurant	Udon Restaurant
9	Koto Ku	Convenience Store	Sake Bar	Coffee Shop	Ramen Restaurant	Chinese Restaurant	Japanese Restaurant	Park	Café	Bed & Breakfast	Donburi Restaurant
11	Minato Ku	Japanese Restaurant	Chinese Restaurant	Ramen Restaurant	BBQ Joint	Tonkatsu Restaurant	Historic Site	Hotel	Convenience Store	Park	Thai Restaurant
12	Suginami Ku	Ramen Restaurant	Italian Restaurant	Dumpling Restaurant	Sake Bar	Café	Soba Restaurant	Bus Stop	Park	Gym / Fitness Center	Tonkatsu Restaurant
14	Bunkyo Ku	Baseball Stadium	Ramen Restaurant	Convenience Store	Coffee Shop	Theme Park Ride / Attraction	Japanese Restaurant	Italian Restaurant	BBQ Joint	Concert Hall	Bakery

6.4.4 cluster-4, Shinjuku has characteristic nightlife tour.

In [47]:
```
Tokyo_merged.loc[Tokyo_merged['Cluster Labels'] == 3, Tokyo_merged.columns[[0] + list(range(4, Tokyo_merged.shape[1]))]]
```

Out[47]:

	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
2	Shinjuku Ku	Sake Bar	Ramen Restaurant	BBQ Joint	Bar	Rock Club	Pub	Japanese Restaurant	Bookstore	Noodle House	Thai Restaurant

6.4.5 cluster-5, Kita-Ku, Adachi-Ku, and e.t.c is where residents live, not a tourist place

In [48]:
```
Tokyo_merged.loc[Tokyo_merged['Cluster Labels'] == 4, Tokyo_merged.columns[[0] + list(range(4, Tokyo_merged.shape[1]))]]
```

Out[48]:

	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	Kita Ku	Convenience Store	Ramen Restaurant	Intersection	Park	Café	Theater	Japanese Restaurant	Pizza Place	Garden	Takoyaki Place
4	Setagaya Ku	Convenience Store	Café	Intersection	Bakery	Sake Bar	Ramen Restaurant	Soba Restaurant	Tea Room	Bistro	Szechuan Restaurant
5	Adachi Ku	Convenience Store	Grocery Store	Kids Store	BBQ Joint	Deli / Bodega	Ramen Restaurant	Pharmacy	Bus Stop	Steakhouse	Furniture / Home Store
6	Katsushika Ku	Supermarket	Intersection	Bus Stop	Convenience Store	Sushi Restaurant	Discount Store	Japanese Restaurant	Drugstore	Automotive Shop	Clothing Store
10	Shinagawa Ku	Convenience Store	Japanese Restaurant	Donburi Restaurant	Ramen Restaurant	Steakhouse	Italian Restaurant	Coffee Shop	BBQ Joint	Theater	Park
13	Akishima Shi	Convenience Store	Intersection	Donburi Restaurant	Park	BBQ Joint	Motorcycle Shop	Ramen Restaurant	Zoo	Food & Drink Shop	Fishing Store
16	Itabashi Ku	Convenience Store	Ramen Restaurant	Chinese Restaurant	Shopping Mall	Grocery Store	Japanese Restaurant	Zoo	Bus Stop	Café	Dessert Shop
17	Edogawa Ku	Convenience Store	Grocery Store	Drugstore	Concert Hall	Donburi Restaurant	Bowling Alley	Rental Car Location	Electronics Store	Cultural Center	Clothing Store
18	Arakawa Ku	Convenience Store	Bus Stop	Park	Tram Station	Chinese Restaurant	Concert Hall	Café	Asian Restaurant	Intersection	Italian Restaurant

6.4.6 cluster-6, Shibuya Ku has characteristic nightlife tour.

In [49]:
```
Tokyo_merged.loc[Tokyo_merged['Cluster Labels'] == 5, Tokyo_merged.columns[[0] + list(range(4, Tokyo_merged.shape[1]))]]
```

Out[49]:

	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
3	Shibuya Ku	Café	Record Shop	Nightclub	Rock Club	Chinese Restaurant	Ramen Restaurant	Sake Bar	Concert Hall	BBQ Joint	Seafood Restaurant

Do you agree with my opinion based on the K-means clustering result from above?

6.5 Japan has the Japanese language, it is not easy to know the neighborhood name because we have difficulty reading Japanese. Maybe it's a factor that concludes the different name of the neighborhood between Wikipedia and Airbnb. The next time we could merge the data with the haversine formula to get more insight into Tokyo.

7. Conclusion

If you are a young person and backpacker, you would love to travel the world. One of them is Tokyo which is very famous. So we have to manage the budget properly. One of the cost savings that we can do is choosing an inexpensive place to stay.

As the result, people are visiting Tokyo for traveling. For this reason, people can be saving the cost of their vacation through their access to the platforms where such information is provided.

Not only for travelers but also major can manage the city more regularly of hotel pricing by using similar data analysis types or platforms.