The Battle of Neighborhoods – Osaka

1. Problem description

Osaka is well-known as the kitchen of Japan, which says a lot, given the internationally highly praised food culture of Japan. Moreover, with its increasing popularity as a tourist destination, it may be of interest to have an overview of the city's neighborhoods for hungry tourists looking for a good spot to replenish their energy throughout the day, or enjoy a nice dinner after an active day of exploring.

Thus, we formulate our business problem as follows: can we provide a segmentation of Osaka's neighborhoods in terms of restaurants that is useful for tourists visiting the city, given the difficulty of choice due to the large number of possibilities to choose from?

In an attempt to solve this problem, we will leverage Foursquare location data on restaurants in Osaka, while referring to the different neighborhoods of the city so as to provide useful information regardless of which part of the city a hungry tourist may find themselves in.

2. Data

We shall require the following data in order to solve the above problem:

- **Neighborhoods & coordinates:** Similarly to previous labs, we will scrape the Wikipedia page on the list of Osaka's wards, after which we will obtain the coordinates of the wards using the geocoder class of the Geopy client. This data consists of features such as the name of the ward, its population, its area, and most importantly its coordinates.
- **Restaurants:** we will source the data on restaurants in Osaka's neighborhoods from Foursquare API. This data might consist of features such as venue category, type of restaurant.

Further details on the collection of data are given in the next section.

3. Methodology

3.1 Data Collection

3.1.1 Wikipedia Data

In an initial attempt to scrape the Wikipedia page for Osaka's wards, we noticed that the table was not parsed properly. Hence, we saved the table in a CSV file, and read it from that file using pandas. This gave us the following dataframe:

In [12]:

```
wiki_data_osaka = pd.read_csv("Osaka_Wards_Wiki.csv")
wiki_data_osaka
```

Out[12]:

	No.	Name	Kanji	Population	Land area in km2	Pop. Density per km2
0	1	Abeno-ku	阿倍野区	107,000	5.99	18,440
1	2	Asahi-ku	旭区	90,854	6.32	14,376
2	3	Chūō-ku	中央区	100,998	8.87	11,386
3	4	Fukushima-ku	福島区	78,348	4.67	16,777
4	5	Higashinari-ku	東成区	83,684	4.54	18,433
5	6	Higashisumiyoshi-ku	東住吉区	126,704	9.75	12,995
6	7	Higashiyodogawa-ku	東淀川区	176,943	13.27	13,334
7	8	Hirano-ku	平野区	193,282	15.28	12,649
8	9	Ikuno-ku	生野区	129,641	8.37	15,489
9	10	Jōtō-ku	城東区	167,925	8.38	20,039
10	11	Kita-ku (administrative center)	北区	136,602	10.34	13,211
11	12	Konohana-ku	此花区	65,086	19.25	3,381
12	13	Minato-ku	港区	80,759	7.86	10,275
13	14	Miyakojima-ku	都島区	107,555	6.08	17,690
14	15	Naniwa-ku	浪速区	74,992	4.39	17,082
15	16	Nishi-ku	西区	103,089	5.21	19,787
16	17	Nishinari-ku	西成区	108,654	7.37	14,743
17	18	Nishiyodogawa-ku	西淀川区	95,960	14.22	6,748
18	19	Suminoe-ku	住之江区	120,629	20.61	5,853

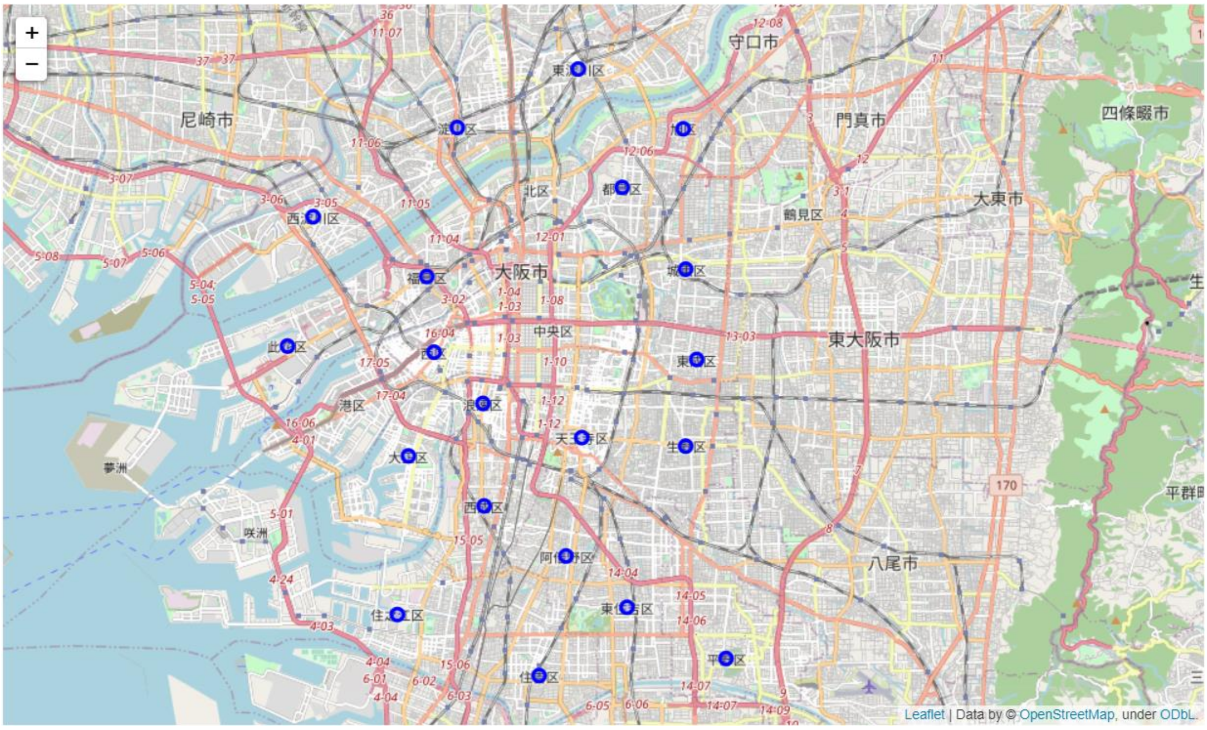
3.1.2 Geopy Data (coordinates)

Next, we obtain the coordinates of each of the wards of Osaka by means of the geocoder class of Geopy client. We obtain the following extended dataframe:

```
geolocator = Nominatim(user_agent="Osaka_explorer")

wiki_data_osaka['Ward Coord'] = wiki_data_osaka['Kanji'].apply(geolocator.geocode).apply(lambda x: (x.latitude, x.longitude)
wiki_data_osaka[['Latitude', 'Longitude']] = wiki_data_osaka['Ward Coord'].apply(pd.Series)

wiki_data_osaka.drop(['Ward Coord'], axis=1, inplace=True)

# rename the column "Name" to "Ward"
wiki_data_osaka.rename(columns = {'Name':'Ward'}, inplace = True)

wiki_data_osaka
```

	No.	Ward	Kanji	Population	Land area in km2	Pop. Density per km2	Latitude	Longitude
0	1	Abeno-ku	阿倍野区	107,000	5.99	18,440	34.627501	135.514095
1	2	Asahi-ku	旭区	90,854	6.32	14,376	34.726483	135.546952

Then, after obtaining the coordinates of the city itself, we use the folium library to produce a plot of Osaka with markers on the centers of its wards:



3.1.3 Foursquare Location Data

After initializing our Foursquare API credentials, we create a function for obtaining nearby venue categories as follows:

```python
def get_nearby_venues(names, latitudes, longitudes, radius = 500):
    list_of_venues = []
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)
        # create the URL for the API request
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}'.format(
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        lat,
        lng,
        radius
        )
        # the get-request
        results = requests.get(url).json()["response"]['groups'][0]['items']
        # return only the relevant information for every nearby venue
        list_of_venues.append([(
        name,
        lat,
        lng,
        v['venue']['name'],
        v['venue']['categories'][0]['name']) for v in results])
    nearby_venues = pd.DataFrame([item for list_of_venues in list_of_venues for item in list_of_venues])
    nearby_venues.columns = ['Neighborhood',
                             'Neighborhood Latitude',
                             'Neighborhood Longitude',
                             'Venue',
                             'Venue Category']
    return(nearby_venues)
```

Using this function, we collect the venues (and their categories) for each neighborhood in Osaka.

3.2 Exploratory Data Analysis

We start by checking the shape of our dataframe via the pandas .shape function, and find that there are 445 venues (rows), with features given in 5 columns as specified in our above shown function get_nearby_venues.

An example of the first 5 rows looks as follows:

```
osaka_venues.head()
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Category
0	Abeno-ku	34.627501	135.514095	Usagi to Boku (うさぎとぼく)	Coffee Shop
1	Abeno-ku	34.627501	135.514095	ライフ セントラルスクエア 北畠店	Supermarket
2	Abeno-ku	34.627501	135.514095	7-Eleven (セブンイレブン 大阪阪南町3丁目店)	Convenience Store
3	Abeno-ku	34.627501	135.514095	FamilyMart (ファミリーマート 阿倍野昭和町店)	Convenience Store
4	Abeno-ku	34.627501	135.514095	モスバーガー 昭和町店	Fast Food Restaurant

We immediately see that there are more than just restaurants in our list of venues, so in the next step we filter for restaurants only (remembering that the goal of our exercise is to provide an overview of restaurant types in Osaka's wards).

```
osaka_restaurants = osaka_venues[osaka_venues['Venue Category'].str.contains("Restaurant|restaurant")]
osaka_restaurants
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Category
4	Abeno-ku	34.627501	135.514095	モスバーガー 昭和町店	Fast Food Restaurant
13	Abeno-ku	34.627501	135.514095	Sea茶	Indian Restaurant
17	Asahi-ku	34.726483	135.546952	MOS Burger (モスバーガー)	Fast Food Restaurant
28	Asahi-ku	34.726483	135.546952	なか卯 千林大宮店	Donburi Restaurant
33	Asahi-ku	34.726483	135.546952	中国料理 風来坊	Chinese Restaurant
...	...	...	...	...	...
423	Tennōji-ku	34.655043	135.518370	ENTERTAIN麺T style JUNK STORY M.I Label	Ramen Restaurant
424	Tennōji-ku	34.655043	135.518370	其蘭	Chinese Restaurant
425	Tennōji-ku	34.655043	135.518370	四天王寺すし割鮮 天山	Sushi Restaurant
433	Yodogawa-ku	34.726613	135.483397	来来亭 十三店	Ramen Restaurant
441	Yodogawa-ku	34.726613	135.483397	藤や	Udon Restaurant

131 rows × 5 columns

We see that there are 131 venues left, all of which are now restaurants.

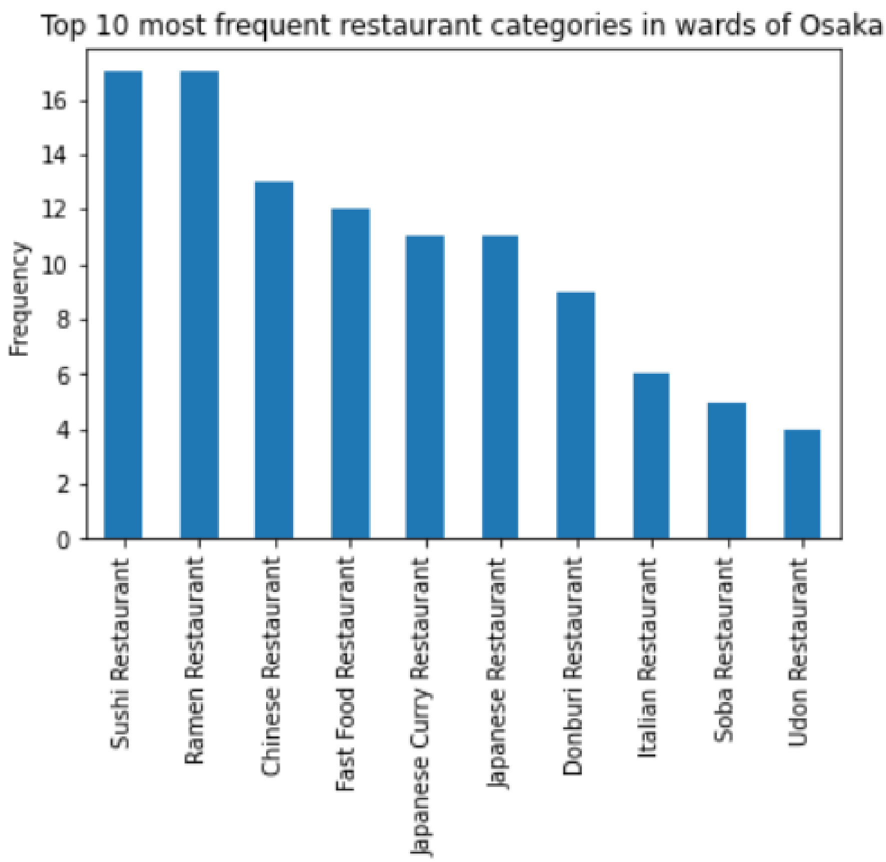
Grouping the restaurants per ward (neighborhood), we obtain the following dataframe:

```
osaka_restaurants.groupby('Neighborhood').head()
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Category
4	Abeno-ku	34.627501	135.514095	モスバーガー 昭和町店	Fast Food Restaurant
13	Abeno-ku	34.627501	135.514095	Sea茶	Indian Restaurant
17	Asahi-ku	34.726483	135.546952	MOS Burger (モスバーガー)	Fast Food Restaurant
28	Asahi-ku	34.726483	135.546952	なか卯 千林大宮店	Donburi Restaurant
33	Asahi-ku	34.726483	135.546952	中国料理 風来坊	Chinese Restaurant
...	...	...	...	...	...
420	Tennōji-ku	34.655043	135.518370	Royal Host (ロイヤルホスト 上本町店)	Restaurant
423	Tennōji-ku	34.655043	135.518370	ENTERTAIN麺T style JUNK STORY M.I Label	Ramen Restaurant
424	Tennōji-ku	34.655043	135.518370	其蘭	Chinese Restaurant
433	Yodogawa-ku	34.726613	135.483397	来来亭 十三店	Ramen Restaurant
441	Yodogawa-ku	34.726613	135.483397	藤や	Udon Restaurant

85 rows × 5 columns

Now we are interested in finding out which restaurant categories are most common in Osaka. For this, we produce a bar plot:



We see that sushi and ramen restaurants are on top of the list!

In a next step, we want to have a look at each ward's top 5 most common restaurant types. In order to achieve this, we one-hot encode the restaurant categories via the pandas get_dummies function:

```python
osaka_restaurants_cat = pd.get_dummies(osaka_restaurants[['Venue Category']], prefix = "", prefix_sep = "")
osaka_restaurants_cat
```

	Chinese Restaurant	Donburi Restaurant	Fast Food Restaurant	French Restaurant	Halal Restaurant	Indian Restaurant	Italian Restaurant	Japanese Curry Restaurant	Japanese Restaurant	Kaiseki Restaurant	...	Seafood Restaurant	Soba Restaurant	Sri L Rest
4	0	0	1	0	0	0	0	0	0	0	...	0	0	
13	0	0	0	0	0	1	0	0	0	0	...	0	0	
17	0	0	1	0	0	0	0	0	0	0	...	0	0	
28	0	1	0	0	0	0	0	0	0	0	...	0	0	
33	1	0	0	0	0	0	0	0	0	0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
423	0	0	0	0	0	0	0	0	0	0	...	0	0	
424	1	0	0	0	0	0	0	0	0	0	...	0	0	
425	0	0	0	0	0	0	0	0	0	0	...	0	0	
433	0	0	0	0	0	0	0	0	0	0	...	0	0	
441	0	0	0	0	0	0	0	0	0	0	...	0	0	

Next, we add the neighborhood to this one-hot encoded dataframe, and move it to the first column. A snapshot of the first 5 rows looks as follows:

```python
osaka_restaurants_cat['Neighborhood'] = osaka_restaurants['Neighborhood']

# move the 'Neighborhood' column to the first column
first_col = osaka_restaurants_cat.pop('Neighborhood')
osaka_restaurants_cat.insert(0, 'Neighborhood', first_col)

osaka_restaurants_cat.head()
```

	Neighborhood	Chinese Restaurant	Donburi Restaurant	Fast Food Restaurant	French Restaurant	Halal Restaurant	Indian Restaurant	Italian Restaurant	Japanese Curry Restaurant	Japanese Restaurant	...	Seafood Restaurant	Soba Restaurant	Sr Re
4	Abeno-ku	0	0	1	0	0	0	0	0	0	...	0	0	
13	Abeno-ku	0	0	0	0	0	1	0	0	0	...	0	0	
17	Asahi-ku	0	0	1	0	0	0	0	0	0	...	0	0	
28	Asahi-ku	0	1	0	0	0	0	0	0	0	...	0	0	
33	Asahi-ku	1	0	0	0	0	0	0	0	0	...	0	0	

5 rows × 26 columns

Using pandas .groupby, we calculate the mean of the occurrence frequency of each restaurant category:

```python
osaka_grouped = osaka_restaurants_cat.groupby('Neighborhood').mean().reset_index()
osaka_grouped
```

	Neighborhood	Chinese Restaurant	Donburi Restaurant	Fast Food Restaurant	French Restaurant	Halal Restaurant	Indian Restaurant	Italian Restaurant	Japanese Curry Restaurant	Japanese Restaurant	...	Seafood Restaurant	Soba Restaurant
0	Abeno-ku	0.000000	0.000000	0.500000	0.0	0.00	0.5	0.000000	0.000000	0.000000	...	0.000000	0.000000
1	Asahi-ku	0.200000	0.200000	0.200000	0.0	0.00	0.0	0.000000	0.000000	0.000000	...	0.000000	0.000000
2	Chūō-ku	0.000000	0.000000	0.000000	0.0	0.00	0.0	0.086957	0.000000	0.086957	...	0.043478	0.086957
3	Fukushima-ku	0.166667	0.083333	0.250000	0.0	0.00	0.0	0.083333	0.000000	0.083333	...	0.000000	0.000000

By writing an additional function, we can represent the top five venues including the respective frequencies as follows (shown here for a single example):

```
Chūō-ku
              Category  Frequency
0      Sushi Restaurant       0.57
1    Italian Restaurant       0.09
2   Japanese Restaurant       0.09
3       Soba Restaurant       0.09
4   Yakitori Restaurant       0.04
```

Alternatively, we can create a new, sorted dataframe containing the top 5 most frequent venue categories for each ward, with a snapshot of the first 5 wards looking as follows:

	Neighborhood	1st most common venue	2nd most common venue	3rd most common venue	4th most common venue	5th most common venue
0	Abeno-ku	Fast Food Restaurant	Indian Restaurant	Yoshoku Restaurant	Kosher Restaurant	Donburi Restaurant
1	Asahi-ku	Chinese Restaurant	Udon Restaurant	Donburi Restaurant	Fast Food Restaurant	Ramen Restaurant
2	Chūō-ku	Sushi Restaurant	Soba Restaurant	Italian Restaurant	Japanese Restaurant	Kaiseki Restaurant
3	Fukushima-ku	Fast Food Restaurant	Ramen Restaurant	Chinese Restaurant	Udon Restaurant	Donburi Restaurant
4	Higashisumiyoshi-ku	Chinese Restaurant	Fast Food Restaurant	Sushi Restaurant	Japanese Curry Restaurant	Kosher Restaurant

3.3 Prescriptive Analytics

In a last step, we start working on the prescriptive part of our analysis.

We use k-means clustering in order to segment Osaka's wards based on restaurant categories. More precisely, wards with similar restaurant categories should be grouped together.

We start by choosing a preliminary number of clusters for our k-means method, after which we train a model:

```
# choose a preliminary number of clusters
k = 5

osaka_grouped_cluster = osaka_grouped.drop('Neighborhood', 1)

# build k-means clustering model
k_means = KMeans(n_clusters = k, random_state=0).fit(osaka_grouped_cluster)

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

```
array([4, 1, 1, 1, 1, 1, 3, 1, 1, 2, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 2])
```

We then add the clustering label column to the top 5 most common restaurant categories:

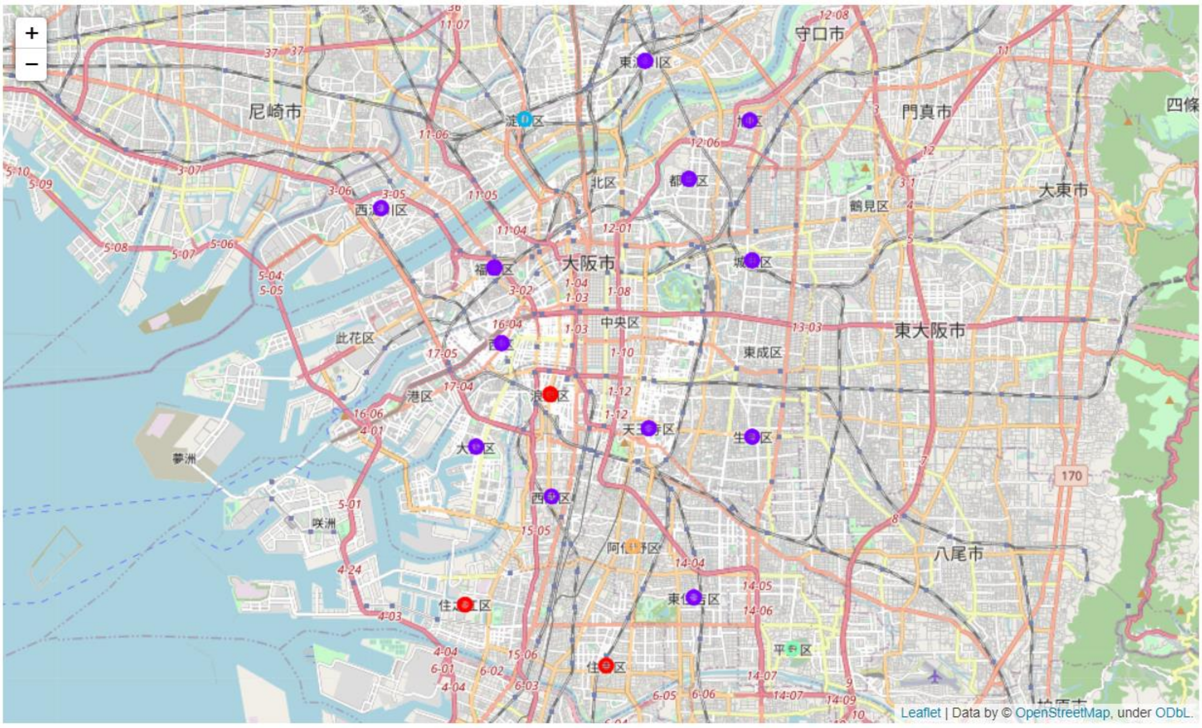
```
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', k_means.labels_)
```

Next, we add the latitudes and longitudes for each neighborhood by joining appropriate dataframes on the Neighborhood column:

```
osaka_merged = osaka_restaurants
osaka_merged = osaka_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on = 'Neighborhood')
osaka_merged.head()
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Category	Cluster Labels	1st most common venue	2nd most common venue	3rd most common venue	4th most common venue	5th most common venue
4	Abeno-ku	34.627501	135.514095	モスバーガー 昭和町店	Fast Food Restaurant	4	Fast Food Restaurant	Indian Restaurant	Yoshoku Restaurant	Kosher Restaurant	Donburi Restaurant
13	Abeno-ku	34.627501	135.514095	Sea茶	Indian Restaurant	4	Fast Food Restaurant	Indian Restaurant	Yoshoku Restaurant	Kosher Restaurant	Donburi Restaurant
17	Asahi-ku	34.726483	135.546952	MOS Burger (モスバーガー)	Fast Food Restaurant	1	Chinese Restaurant	Udon Restaurant	Donburi Restaurant	Fast Food Restaurant	Ramen Restaurant
28	Asahi-ku	34.726483	135.546952	なか卯 千林大宮店	Donburi Restaurant	1	Chinese Restaurant	Udon Restaurant	Donburi Restaurant	Fast Food Restaurant	Ramen Restaurant
33	Asahi-ku	34.726483	135.546952	中国料理 風来坊	Chinese Restaurant	1	Chinese Restaurant	Udon Restaurant	Donburi Restaurant	Fast Food Restaurant	Ramen Restaurant

Finally, we drop any NaN values from the dataset, and we plot the clusters on a map using the folium library:



And there we have it: a visual representation of a clustering of Osaka by restaurant categories, based on k-means clustering!

4. Results & Discussion

The main results of our analysis are the following:

- Sushi and ramen restaurants lead in the list of most frequently occurring restaurant types in Osaka.

- In particular, we have found that the Chuo ward is the most prominent one in terms of number of sushi restaurants, while the Kita ward contains the most ramen restaurants in relative terms per ward. Japanese curry restaurants can be most frequently found in the Naniwa ward, while the Taisho ward most frequently contains donburi restaurants. The rest of the wards are fairly evenly split in terms of restaurant types.

- The clustering of wards by restaurant types leads to a main cluster around the central part of the city (with two exceptional central wards belonging to two other clusters), while the rest of the clusters are mainly spread out along the outer wards.

There are however several points that are important to note:

- Based on the number of restaurants per restaurant type, it is clear that we only obtained a limited number of data from Foursquare; this might be due to the fact that Osaka is still growing in terms of popularity as a destination, thus leading to a lower number of venues registered on Foursquare. In any case, our analysis of the most frequent restaurant types in Osaka might have to be taken with caution, due to the limited representativeness given by the low number of data found on Foursquare.

- Further on the topic of information obtained from Foursquare, we notice that there is a restaurant type called "Japanese restaurant". As it is not immediately clear what this entails, it could introduce a bias in our data in case some sushi/ramen/Japanese curry restaurants are in fact classified as "Japanese restaurants", instead of within the dedicated category of restaurant type.

- For the clustering, we ignored data features such as distance from public transportation stations, ranges of prices of the restaurants etc., as such data would be significantly more difficult to obtain.

- The results of our clustering exercise would potentially vary if choosing a different number of clusters k (this choice could also be made quantitatively, by looking at the accuracy of the model for different values of k), or by using a different clustering approach altogether, such as hierarchical clustering or density-based clustering methods.

5. Conclusion

The purpose of this project was to explore the different wards of Osaka, and segment them according to restaurant types. We explored the city, extrapolating common venues within each of the wards, and finally concluding with a clustering of similar wards.

We saw that each of the wards has a variety of experiences to offer, being unique in its own way. While this allows us to conclude that all parts of the city are interesting tourist destinations, a hungry tourist may still make use of our analysis to help them decide where to go for a meal of their preference.

Overall, such an analysis still contains many aspects that can be extended, but also applied to other kinds of business problems, depending on the domain of interest. For what our business problem is concerned, a visitor of Osaka may certainly also want to take into account various other aspects such as noise level, crowdedness, affordability and other characteristics when choosing where to dine.