Introduction to Business Problem

With nearly 10 million inhabitants and one of the highest metropolitan GDP in the world, an estimated 22 million overseas visitors are expected to flock to the capital and surrounding regions to explore the culture, food and historical shrine.

Bangkok is well-known as the restaurant capital of the world, with over hundreds of thousands of places to choose from around it's 50 districts.

I believe it's difficult for a traveller, especially restaurant-goers, to make a choice from among many options since there is also too much information on the web because everybody's got their own take of where to go and it's all so fragmented that you have to assemble it yourself especially if you're interested in non-touristy recommendations.

The objective of this report is to help restaurant-goers to make a choice from many options since the amount of information on the internet is abundant, making a decision to pick a restaurant can be time consuming. it will also aid tourist to pick a place with non-tourist recommendations.

Data

Bangkok data that contains list districts along with their latitude and longitude.

Data source: https://en.wikipedia.org/wiki/List_of_districts_of_Bangkok

Geopy - For getting the co-ordinated of different locations.
We will Bangkok districts Table from Wikipedia and get the coordinates of these 50 districts using geocoder class of Geopy client.

Data source: Foursquare APIs

Description: By using this API we will get all the venues in each neighbourhood. We can filter these venues to get only restaurants.

Methodology

Data Preparation

I first make use of district of Bangkok page from Wiki to scrap the table to create a data-frame. For this, I've used pandas to transform the data in the table on the Wikipedia page into a data frame containing name of the 50 districts of Bangkok, post-code, population, latitude and longitude. We start as below:

```
In [92]: df = pd.read_html('https://en.wikipedia.org/wiki/List_of_districts_of_Bangkok')[0]
         df.head(25)
```

Out[92]:

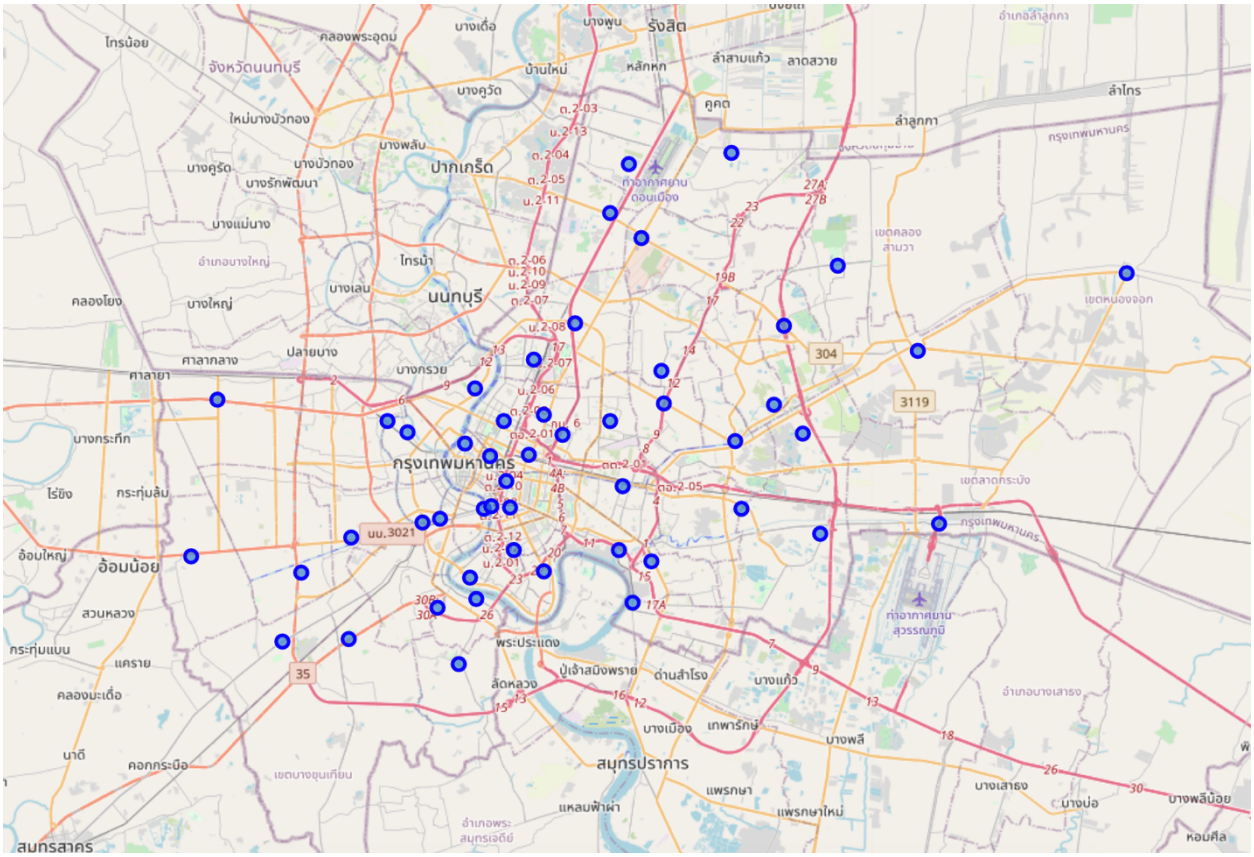
	District(Khet)	MapNr	Post-code	Thai	Popu-lation	No. ofSubdis-trictsKhwaeng	Latitude	Longitude
0	Bang Bon	50	10150	บางบอน	105161	4	13.659200	100.399100
1	Bang Kapi	6	10240	บางกะปิ	148465	2	13.765833	100.647778
2	Bang Khae	40	10160	บางแค	191781	4	13.696111	100.409444
3	Bang Khen	5	10220	บางเขน	189539	2	13.873889	100.596389
4	Bang Kho Laem	31	10120	บางคอแหลม	94956	3	13.693333	100.502500
5	Bang Khun Thian	21	10150	บางขุนเทียน	165491	2	13.660833	100.435833
6	Bang Na	47	10260	บางนา	95912	2	13.680081	100.591800
7	Bang Phlat	25	10700	บางพลัด	99273	4	13.793889	100.505000
8	Bang Rak	4	10500	บางรัก	45875	5	13.730833	100.524167
9	Bang Sue	29	10800	บางซื่อ	132234	2	13.809722	100.537222
10	Bangkok Noi	20	10700	บางกอกน้อย	117793	5	13.770867	100.467933
11	Bangkok Yai	16	10600	บางกอกใหญ่	72321	2	13.722778	100.476389
12	Bueng Kum	27	10240	บึงกุ่ม	145830	3	13.785278	100.669167
13	Chatuchak	30	10900	จตุจักร	160906	5	13.828611	100.559722
14	Chom Thong	35	10150	จอมทอง	158005	4	13.677222	100.484722
15	Din Daeng	26	10400	ดินแดง	130220	2	13.769722	100.552778
16	Don Mueang	36	10210	ดอนเมือง	166261	3	13.913611	100.589722
17	Dusit	2	10300	ดุสิต	107655	5	13.776944	100.520556
18	Huai Khwang	17	10310	ห้วยขวาง	78175	3	13.776667	100.579444
19	Khan Na Yao	43	10230	คันนายาว	88678	2	13.827100	100.674300
20	Khlong Sam Wa	46	10510	คลองสามวา	169489	5	13.859722	100.704167
21	Khlong San	18	10600	คลองสาน	76446	4	13.730278	100.509722
22	Khlong Toei	33	10110	คลองเตย	109041	3	13.708056	100.583889
23	Lak Si	41	10210	หลักสี่	109770	2	13.887500	100.578889

Getting Coordinates of Major Districts

After a little manipulation and processing the data. The objective is to get the coordinates of these 50 districts using geocoder class of Geopy.

	Name	Latitude	Longitude
0	Bang Bon	13.659200	100.399100
1	Bang Kapi	13.765833	100.647778
2	Bang Khae	13.696111	100.409444
3	Bang Khen	13.873889	100.596389
4	Bang Kho Laem	13.693333	100.502500
5	Bang Khun Thian	13.660833	100.435833
6	Bang Na	13.680081	100.591800
7	Bang Phlat	13.793889	100.505000
8	Bang Rak	13.730833	100.524167
9	Bang Sue	13.809722	100.537222
10	Bangkok Noi	13.770867	100.467933
11	Bangkok Yai	13.722778	100.476389
12	Bueng Kum	13.785278	100.669167
13	Chatuchak	13.828611	100.559722
14	Chom Thong	13.677222	100.484722
15	Din Daeng	13.769722	100.552778
16	Don Mueang	13.913611	100.589722
17	Dusit	13.776944	100.520556
18	Huai Khwang	13.776667	100.579444
19	Khan Na Yao	13.827100	100.674300
20	Khlong Sam Wa	13.859722	100.704167
21	Khlong San	13.730278	100.509722
22	Khlong Toei	13.708056	100.583889
23	Lak Si	13.887500	100.578889
24	Lat Krabang	13.722317	100.759669

I used python **folium** library to visualize geographic details of Bangkok 50 district and I created a map of Bangkok with boroughs superimposed on top. I used latitude and longitude values to get the visual as below:



Exploratory Data Analysis:

Firstly, I will use *exploratory data analysis(EDA)* to uncover hidden properties of data and provide useful insights to the reader, both future traveler and investor.

Using Foursquare Location Data

Finally, let's make use of Foursquare API and get the list of venue category in each district and convert it into a data frame.

```
In [31]: #Convert the venue list into dataframe
venues_df = pd.DataFrame(venues)
venues_df.columns = ['Name', 'Latitude', 'Longitude', 'Venue name', 'Venue Lat', 'Venue Lng', 'Venue Category', 'Venue ID']
venues_df.head()
```

Out[31]:

	Name	Latitude	Longitude	Venue name	Venue Lat	Venue Lng	Venue Category	Venue ID
0	Bang Bon	13.6592	100.3991	ขาหมูบางหว้า	13.657136	100.395230	Thai Restaurant	4e880a81f790e992e01d7284
1	Bang Bon	13.6592	100.3991	ร้านต้นไม้ ริมถนนกาญจนาภิเษก	13.654098	100.405054	Garden Center	4bf8e392508c0f4796f13e31
2	Bang Bon	13.6592	100.3991	TPD Bowling	13.663977	100.408965	Bowling Alley	52a0891811d20bdfe3d3ed2d
3	Bang Bon	13.6592	100.3991	เจ๊ไน้ ก๋วยเตี๋ยวเป็ดตุ๋น	13.654137	100.405323	Noodle House	4d69adfa342b8cfa9bbccc2c
4	Bang Bon	13.6592	100.3991	Irashaimase Japanese Restaurant	13.658358	100.401403	Japanese Restaurant	5218ae3411d247f3bb76befc

Then getting the list of all the categories of all the restaurant present in venues_df dataframe.

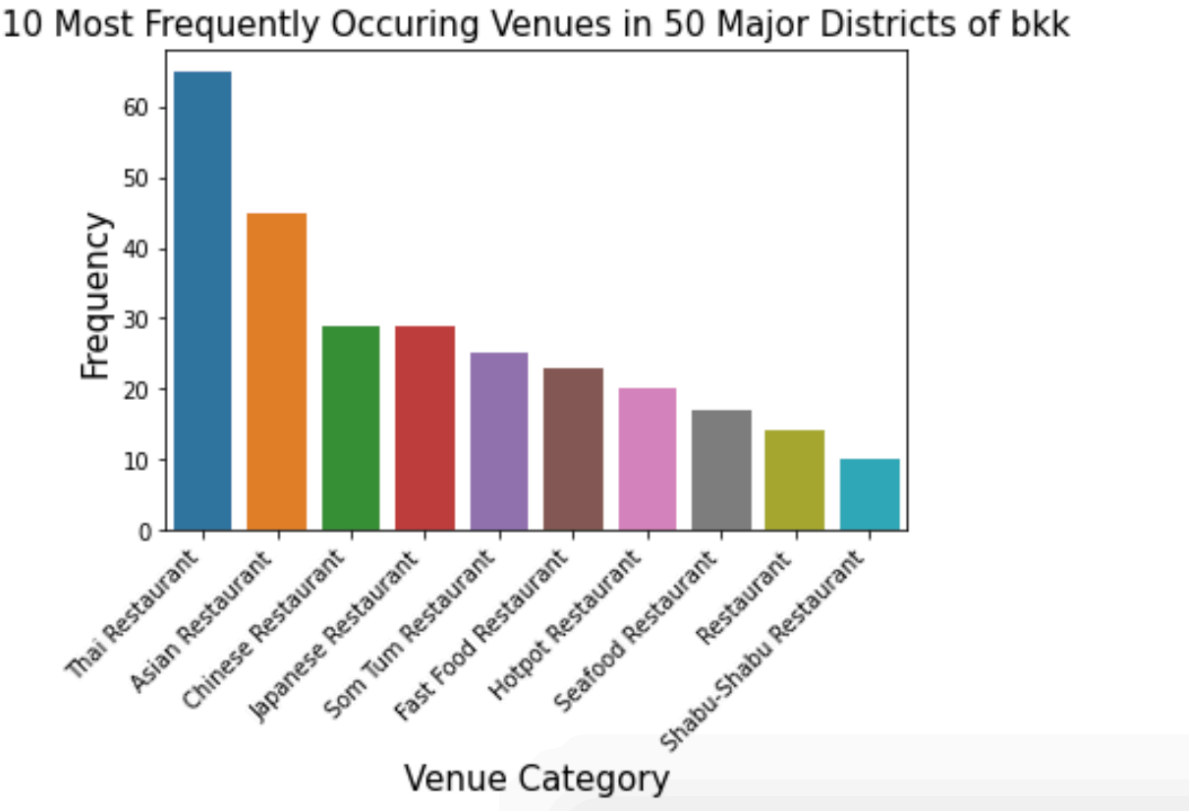
```
In [98]: demo1_df = pd.DataFrame({'Venue Category':res_df.index[:50]})
category_strength=[]
for i in range(50):
    category_strength.append(res_df['Count'][i])
demo2_df = pd.DataFrame(category_strength, columns=['Count'])
demo_df = pd.DataFrame({'Venue Category': demo1_df['Venue Category'], 'Count': demo2_df['Count']})
demo_df.head(20)
```

Out[98]:

	Venue Category	Count
0	Noodle House	376
1	Thai Restaurant	357
2	Coffee Shop	332
3	Convenience Store	259
4	Café	166
5	Hotel	131
6	Japanese Restaurant	125
7	Asian Restaurant	120
8	Chinese Restaurant	97
9	Som Tum Restaurant	96
10	Bar	88
11	Dessert Shop	87
12	Seafood Restaurant	79
13	Hotpot Restaurant	77
14	Ice Cream Shop	70
15	Fast Food Restaurant	66
16	Restaurant	66
17	Bakery	65
18	BBQ Joint	64
19	Shopping Mall	62

Later on, I will concentrate in Restaurant Category only and explore all the 50 district.

We find out 31 unique venue categories and Noodle Restaurants top the charts as we can see in the plot below:



So, definitely you need to try the delicious noodles when in Bangkok, but in which district noodle restaurant are more common? Let's get back to exploring the data a little more.

Let’s analyze each neighborhood to know about the top 5 venues of each one.

So, we proceed as follows:

1. Create a data-frame with pandas one hot coding for the venue categories.

In [49]:
```
# one hot encoding
bkk_onehot = pd.get_dummies(bkk_Venues_only_restaurant[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
bkk_onehot['Neighborhood'] = bkk_Venues_only_restaurant['Neighborhood']
```

In [50]:
```
# move neighborhood column to the first column
fixed_columns = [bkk_onehot.columns[-1]] + bkk_5_Dist_Venues_restaurant_df = bkk_5_Dist_Venues_restaurant.to_frame().reset_index()
bkk_5_Dist_Venues_restaurant_df.columns = ['District', 'Number of Restaurant'](bkk_onehot.columns[:-1])
bkk_onehot = bkk_onehot[fixed_columns]

bkk_onehot.head()
```

2. Use pandas groupby on neighborhood column and calculate the mean of the frequency of occurrence of each venue category.

Out[53]:

	Neighborhood	American Restaurant	Asian Restaurant	Cantonese Restaurant	Chinese Restaurant	Comfort Food Restaurant	Dim Sum Restaurant	Donburi Restaurant	Dumpling Restaurant	Fast Food Restaurant	...	Restaurant	Seafood Restaurant	Shabu-Shabu Restaurant	Som Tum Restaurant	Sushi Restaurant	Thai Restaurant	Tonkatsu Restaurant	Udon Restaurant	Vegetarian / Vegan Restaurant	Vi R
0	Bang Bon	0.000000	0.250000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.500000	0.000000	0.000000	0.000000	
1	Bang Kapi	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.333333	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
2	Bang Khae	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
3	Bang Khen	0.000000	0.500000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.250000	0.000000	0.000000	0.000000	0.000000	0.000000	
4	Bang Kho Laem	0.000000	0.000000	0.0	0.285714	0.000000	0.000000	0.000000	0.000000	0.142857	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.285714	0.000000	0.000000	0.000000	
5	Bang Khun Thian	0.000000	0.000000	0.0	0.076923	0.000000	0.000000	0.000000	0.000000	0.076923	...	0.153846	0.000000	0.000000	0.000000	0.000000	0.230769	0.000000	0.000000	0.000000	
6	Bang Na	0.000000	0.400000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.200000	0.000000	0.000000	0.000000	0.400000	0.000000	0.000000	0.000000	
7	Bang Phlat	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.500000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.500000	0.000000	0.000000	0.000000	
8	Bang Rak	0.000000	0.000000	0.0	0.375000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.125000	0.125000	0.125000	0.000000	0.250000	0.000000	0.000000	0.000000	
9	Bang Sue	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.200000	0.000000	0.000000	0.000000	0.600000	0.000000	0.000000	0.000000	
10	Bangkok Noi	0.000000	0.142857	0.0	0.000000	0.000000	0.142857	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.142857	0.571429	0.000000	0.000000	0.000000	0.000000	0.000000	
11	Bangkok Yai	0.000000	0.333333	0.0	0.166667	0.000000	0.000000	0.000000	0.000000	0.166667	...	0.000000	0.166667	0.000000	0.000000	0.000000	0.166667	0.000000	0.000000	0.000000	
12	Bueng Kum	0.000000	0.500000	0.0	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	0.000000	0.000000	
13	Chatuchak	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.250000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.750000	0.000000	0.000000	0.000000	
14	Chom Thong	0.333333	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.333333	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.333333	0.000000	0.000000	0.000000	
15	Don Mueang	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
16	Dusit	0.000000	0.666667	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.333333	0.000000	0.000000	0.000000	0.000000	0.000000	
17	Huai Khwang	0.000000	0.166667	0.0	0.166667	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.166667	0.000000	0.000000	0.333333	0.000000	0.000000	0.000000	0.000000	0.000000	
18	Khan Na Yao	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.117647	...	0.058824	0.000000	0.117647	0.058824	0.058824	0.117647	0.058824	0.058824	0.000000	
19	Khlong Sam Wa	0.000000	0.000000	0.0	0.200000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.200000	0.000000	0.200000	0.000000	0.000000	0.200000	0.000000	0.000000	0.000000	
20	Khlong San	0.000000	0.000000	0.1	0.100000	0.000000	0.100000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.400000	0.000000	0.000000	0.000000	
21	Khlong Toei	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	
22	Lak Si	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.285714	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.285714	0.000000	0.000000	0.000000	
23	Lat Krabang	0.000000	0.250000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.250000	0.000000	0.500000	0.000000	0.000000	0.000000	
24	Lat Phrao	0.000000	0.400000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.400000	0.000000	0.000000	0.000000	0.000000	0.000000	
25	Min Buri	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	

3. Output each neighborhood along with the top 5 most common venues:

Top 5 most common venues

In [56]:
```
num_top_venues = 5

for hood in bkk_grouped['Neighborhood']:

    print("--",hood,"--")

    temp = bkk_grouped[bkk_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')
```

```
-- Khlong San --
                venue  freq
0       Thai Restaurant   0.4
1   Japanese Restaurant   0.1
2  Cantonese Restaurant   0.1
3    Chinese Restaurant   0.1
4    Dim Sum Restaurant   0.1


-- Khlong Toei --
                          venue  freq
0               Thai Restaurant   1.0
1            American Restaurant   0.0
2              Korean Restaurant   0.0
3  Vegetarian / Vegan Restaurant   0.0
4               Udon Restaurant   0.0
```

I will use *prescriptive analytics* to help a traveler decide a location to go for a restaurant. I will use *clustering* (KMeans).

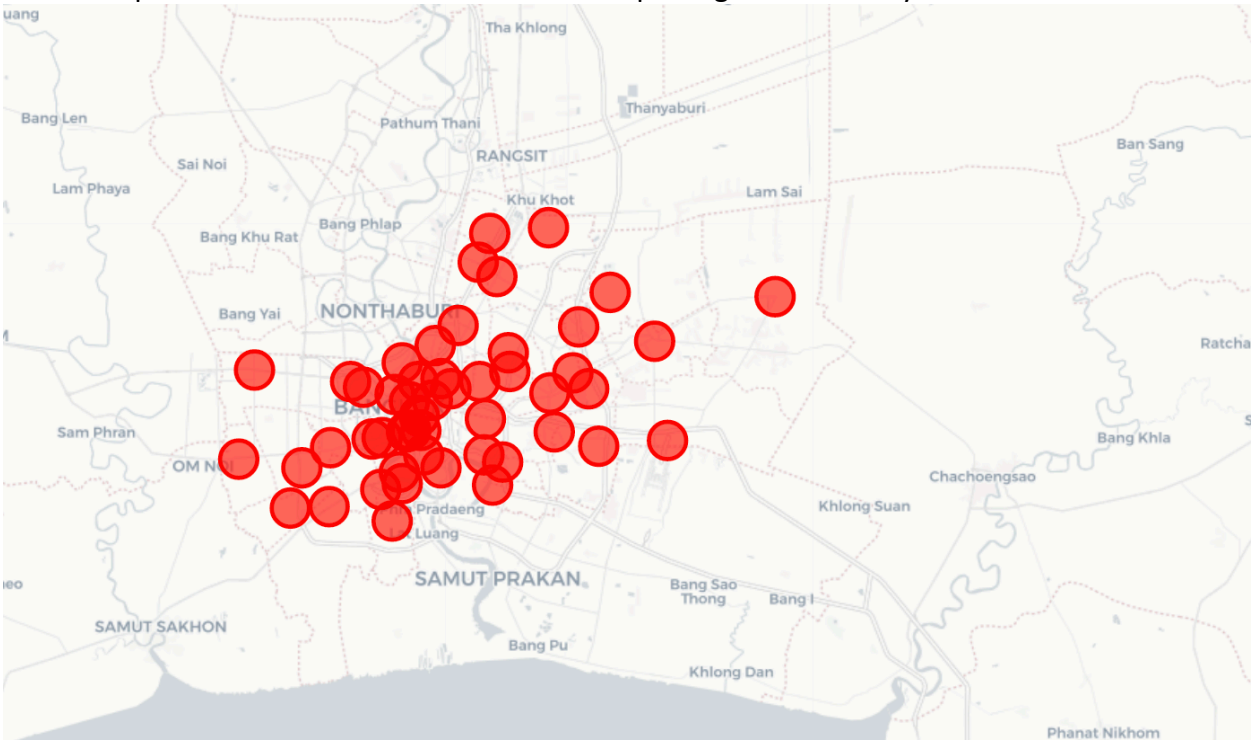
Finally, we try to cluster these 50 districts based on the venue categories and use K-Means clustering. So, our expectation would be based on the similarities of venue categories, these districts will be clustered. I have used the code below :

In [58]:
```
indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
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))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = bkk_grouped['Neighborhood']

for ind in np.arange(bkk_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(bkk_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.head()
```

Out[58]:

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Bang Bon	Thai Restaurant	Japanese Restaurant	Asian Restaurant	Japanese Curry Restaurant	Cantonese Restaurant
1	Bang Kapi	Shabu-Shabu Restaurant	Ramen Restaurant	Hotpot Restaurant	Vietnamese Restaurant	Italian Restaurant
2	Bang Khae	Fast Food Restaurant	Vietnamese Restaurant	Japanese Curry Restaurant	Asian Restaurant	Cantonese Restaurant
3	Bang Khen	Asian Restaurant	Vietnamese Restaurant	Som Tum Restaurant	Japanese Curry Restaurant	Cantonese Restaurant
4	Bang Kho Laem	Thai Restaurant	Chinese Restaurant	Vietnamese Restaurant	Fast Food Restaurant	Hotpot Restaurant

We can represent these 5 clusters in a leaflet map using Folium library as below:



Results & discussion

We got a glimpse of the Restaurants in Bangkok and were able to find out some interesting insights which might be useful to travelers as well as people with business interests. Let's summarize our findings:

- Noodle restaurants top the charts of most common venues in the 50 districts.
- Pathum wan and Phayathai has maximum number of restaurants.
- Since the clustering was based only on the category of restaurants on each district, all fall in the same cluster, which indicate that each of those districts presents a similar experience to the traveler in terms of category of food.
- It's also important to note that each district is connected to sky trains or bus stops which make them accessible and easy to move between them.
- The clustering is completely based on the most common venues obtained from Foursquare data.

However, in our analysis, we have ignored other factors like distance of the venues from closest stations, range of prices of restaurants, Michelin Restaurants and so on, since we don't have such data and it would be difficult to farm it for a small exploratory study like ours. Hence, our analysis only helps travelers to get an overview of Restaurants distribution by categories in the 50 districts of Bangkok.

Furthermore, this results also could potentially vary if we use some other clustering techniques like DBSCAN.

Conclusion

In a fast-moving world, there are many real-life problems or scenarios where data can be used to find solutions to those problems. Like seen in the example above, data was used to cluster neighborhoods in Bangkok based on the most common food venues (Restaurants) in its 50 major districts. The results can help a traveler to decide about the district that fit the most his needs.

I have made use of some frequently used python libraries to scrap web-data, use Foursquare API to explore the major districts of Bangkok and saw the results of segmentation of districts using Folium leaflet map.

Similarly, data can also be used to solve other problems, which most people face in metropolitan cities. Potential for this kind of analysis in a real-life problem is discussed in great detail. Also, some of the drawbacks and chance for improvements to represent even more realistic pictures are mentioned.