Final Report | Capstone Project – The Battle of Neighborhoods Finding A Place To Visit in Penang

1. Introduction:

Penang, the city the author lives in, is a Malaysian state located on the northwest coast of Peninsular Malaysia, by the Malacca Strait, attracts a large number of tourists every year. For tourists, finding the right place to eat and hang around can be a challenge, though. Some crave for delicious food while other may prefer sightseeing. Thus, with this exercise, I wanted to give a simple reference and recommendation to tourists in Penang. Which district in Penang should food hunter visit? Which areas are best for coffee lovers? If a tourist like sunbath a lot, which hotel should he/she stays?

2. Data Section

I will use the overview of districts/city parts of Northeast Penang from Wikipedia: https://en.wikipedia.org/wiki/Northeast_Penang_Island_District

Foursquare API:

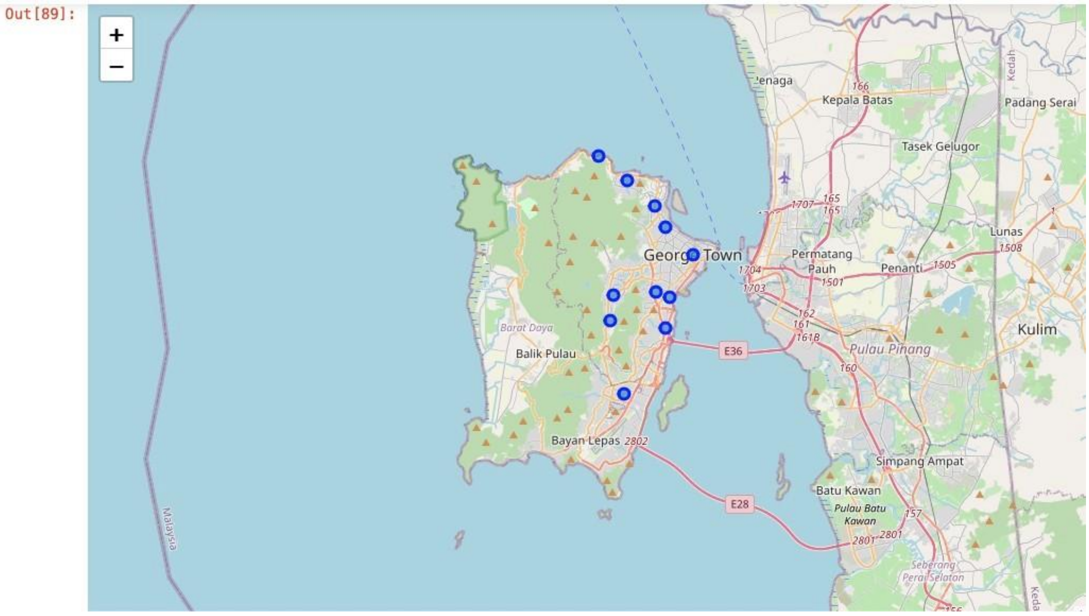
As requested by the assignment task, this project would use Four-square API as its prime data gathering source about Penang Northeast Region as it has a database of millions of places, especially their places API which provides the ability to perform location search, location sharing and details about a business.
I will use foursquare data such as the venue name, ID, location and category(vegetarian, Italian etc.).

After finding the list of neighborhoods, then connect to the Foursquare API to gather information about venues inside each and every neighborhood. For each district, we have chosen the radius to be 500 meter.

The data retrieved from Foursquare contained information of venues within a specified distance of the longitude and latitude of the postcodes. The information obtained per venue as follows:

1. Neighborhood
2. Neighborhood Latitude
3. Neighborhood Longitude
4. Venue
5. Name of the venue e.g. the name of a store or restaurant
6. Venue Latitude
7. Venue Longitude
8. Venue Category

Map of Northeast Penang



3. Methodology Section

Work Flow:

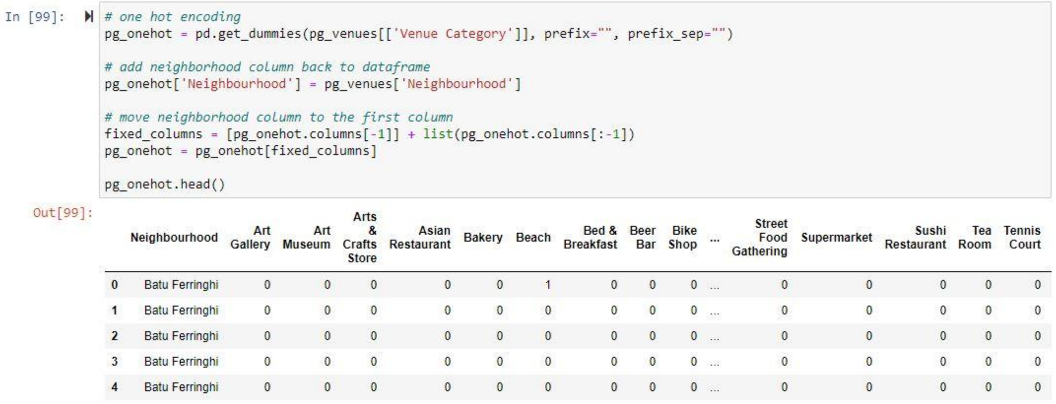
Using credentials of Foursquare API features of near-by places of the neighborhoods would be mined. Due to http request limitations the number of places per neighborhood parameter would reasonably be set to 100 and the radius parameter would be set to 500.

Clustering Approach:

To compare the similarities of the districts, i will explore the neighborhoods, segment them, and group them into clusters to find similar neighborhoods in each district. To be able to do that, we need to cluster data which is a form of unsupervised machine learning: k-means clustering algorithm.

Using One-hot encoding:

To find clusters of restaurant types in the different city districts, I first transformed the data frame with the venues information, associated to city districts, by one-hot encoding (0/1), as seen in the picture below



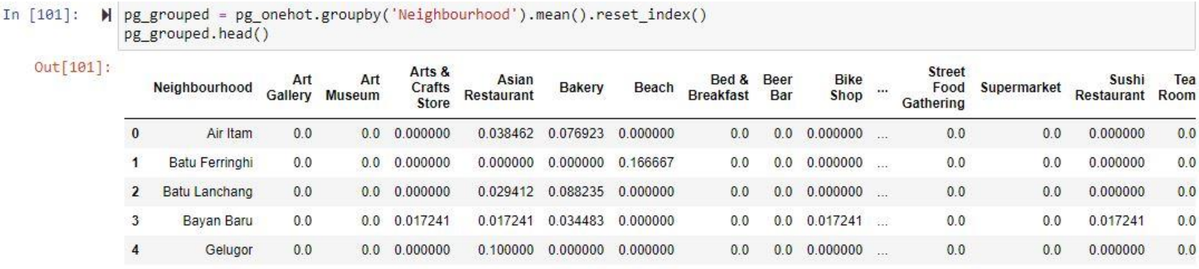
In [99]:

```
# one hot encoding
pg_onehot = pd.get_dummies(pg_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
pg_onehot['Neighbourhood'] = pg_venues['Neighbourhood']

# move neighborhood column to the first column
fixed_columns = [pg_onehot.columns[-1]] + list(pg_onehot.columns[:-1])
pg_onehot = pg_onehot[fixed_columns]

pg_onehot.head()
```

Out[99]:

	Neighbourhood	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	Bakery	Beach	Bed & Breakfast	Beer Bar	Bike Shop	...	Street Food Gathering	Supermarket	Sushi Restaurant	Tea Room	Tennis Court
0	Batu Ferringhi	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0
1	Batu Ferringhi	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
2	Batu Ferringhi	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
3	Batu Ferringhi	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
4	Batu Ferringhi	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0

Next, I used grouping to show the frequency of each category of venues in each city district.



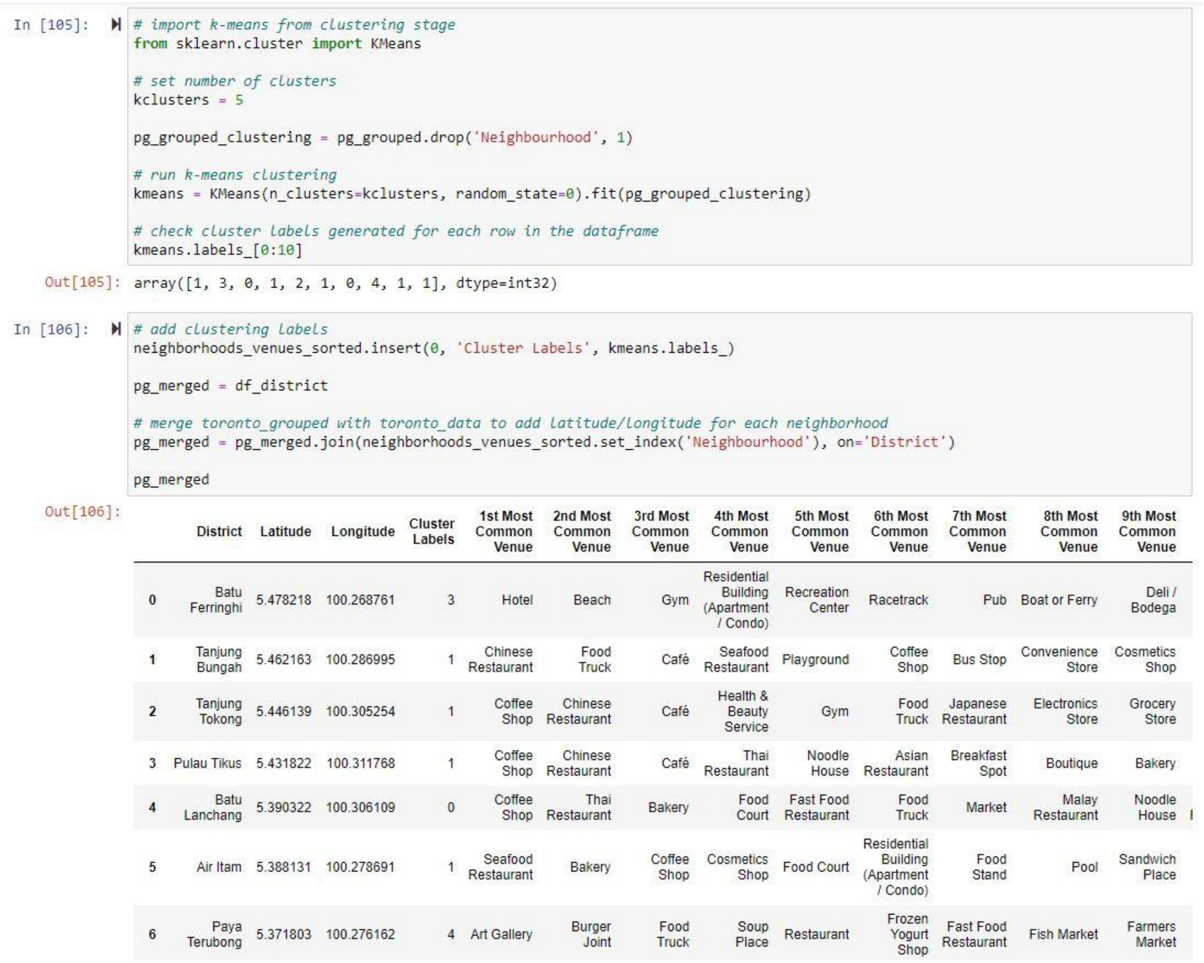
```
In [101]:  pg_grouped = pg_onehot.groupby('Neighbourhood').mean().reset_index()
           pg_grouped.head()
```

Out[101]:

	Neighbourhood	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	Bakery	Beach	Bed & Breakfast	Beer Bar	Bike Shop	...	Street Food Gathering	Supermarket	Sushi Restaurant	Tea Room
0	Air Itam	0.0	0.0	0.000000	0.038462	0.076923	0.000000	0.0	0.0	0.000000	...	0.0	0.0	0.000000	0.0
1	Batu Ferringhi	0.0	0.0	0.000000	0.000000	0.000000	0.166667	0.0	0.0	0.000000	...	0.0	0.0	0.000000	0.0
2	Batu Lanchang	0.0	0.0	0.000000	0.029412	0.088235	0.000000	0.0	0.0	0.000000	...	0.0	0.0	0.000000	0.0
3	Bayan Baru	0.0	0.0	0.017241	0.017241	0.034483	0.000000	0.0	0.0	0.017241	...	0.0	0.0	0.017241	0.0
4	Gelugor	0.0	0.0	0.000000	0.100000	0.000000	0.000000	0.0	0.0	0.000000	...	0.0	0.0	0.000000	0.0

Using K-Means Clustering Approach:

With the data, k-means clustering algorithm from the scikit-learn package is used. One could use the elbow method to systematically define the k value, but I simply chose k to be 5, having been inspired by one of the coursera courses to do so.



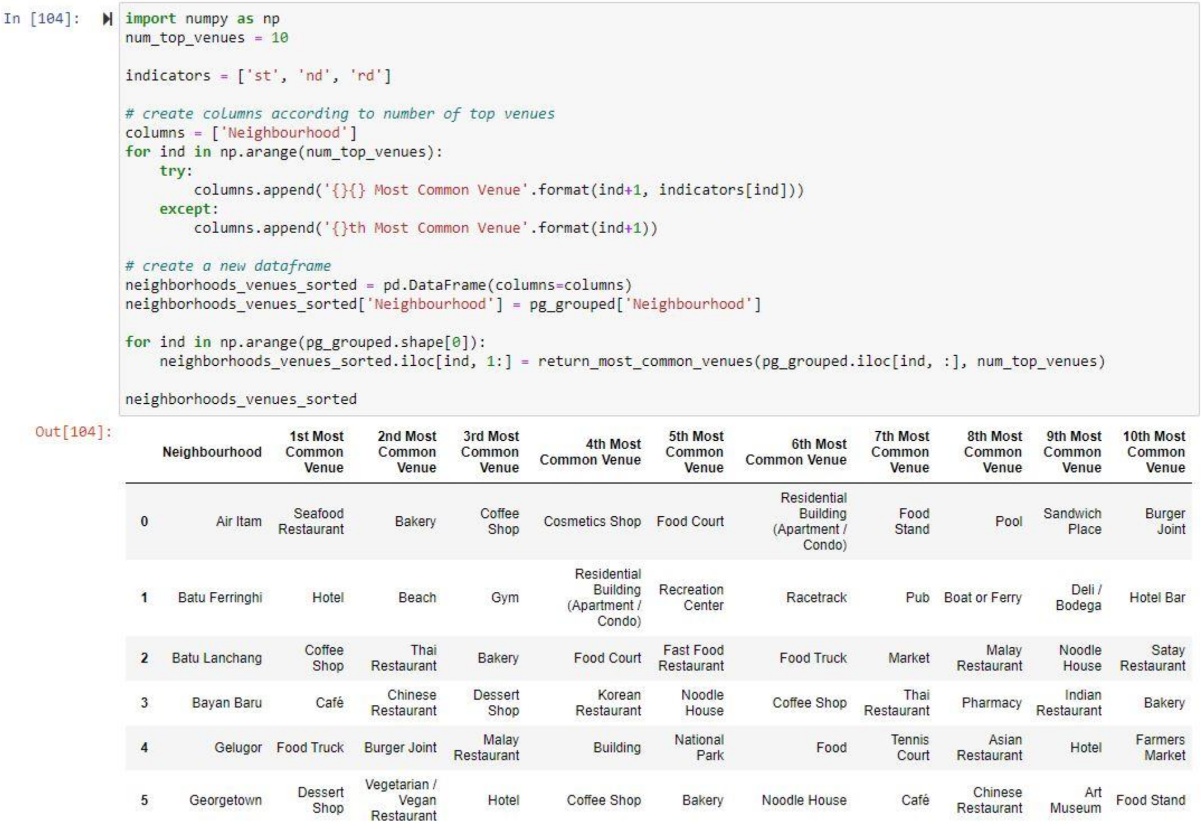
```
In [105]:  # import k-means from clustering stage
           from sklearn.cluster import KMeans

           # set number of clusters
           kclusters = 5

           pg_grouped_clustering = pg_grouped.drop('Neighbourhood', 1)

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

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

Out[105]:  array([1, 3, 0, 1, 2, 1, 0, 4, 1, 1], dtype=int32)
```

```
In [106]:  # add clustering labels
           neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

           pg_merged = df_district

           # merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
           pg_merged = pg_merged.join(neighborhoods_venues_sorted.set_index('Neighbourhood'), on='District')

           pg_merged
```

Out[106]:

	District	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	Batu Ferringhi	5.478218	100.268761	3	Hotel	Beach	Gym	Residential Building (Apartment / Condo)	Recreation Center	Racetrack	Pub	Boat or Ferry	Deli / Bodega
1	Tanjung Bungah	5.462163	100.286995	1	Chinese Restaurant	Food Truck	Café	Seafood Restaurant	Playground	Coffee Shop	Bus Stop	Convenience Store	Cosmetics Shop
2	Tanjung Tokong	5.446139	100.305254	1	Coffee Shop	Chinese Restaurant	Café	Health & Beauty Service	Gym	Food Truck	Japanese Restaurant	Electronics Store	Grocery Store
3	Pulau Tikus	5.431822	100.311768	1	Coffee Shop	Chinese Restaurant	Café	Thai Restaurant	Noodle House	Asian Restaurant	Breakfast Spot	Boutique	Bakery
4	Batu Lanchang	5.390322	100.306109	0	Coffee Shop	Thai Restaurant	Bakery	Food Court	Fast Food Restaurant	Food Truck	Market	Malay Restaurant	Noodle House
5	Air Itam	5.388131	100.278691	1	Seafood Restaurant	Bakery	Coffee Shop	Cosmetics Shop	Food Court	Residential Building (Apartment / Condo)	Food Stand	Pool	Sandwich Place
6	Paya Terubong	5.371803	100.276162	4	Art Gallery	Burger Joint	Food Truck	Soup Place	Restaurant	Frozen Yogurt Shop	Fast Food Restaurant	Fish Market	Farmers Market

Most Common venues near Neighborhood:



In [104]:

```
import numpy as np
num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighbourhood']
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['Neighbourhood'] = pg_grouped['Neighbourhood']

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

neighborhoods_venues_sorted
```

Out[104]:

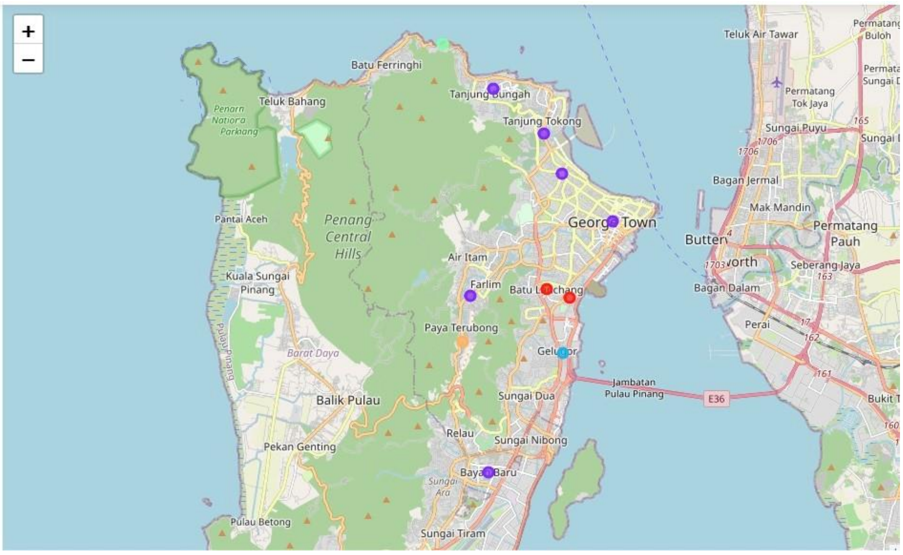
	Neighbourhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Air Itam	Seafood Restaurant	Bakery	Coffee Shop	Cosmetics Shop	Food Court	Residential Building (Apartment / Condo)	Food Stand	Pool	Sandwich Place	Burger Joint
1	Batu Ferringhi	Hotel	Beach	Gym	Residential Building (Apartment / Condo)	Recreation Center	Racetrack	Pub	Boat or Ferry	Deli / Bodega	Hotel Bar
2	Batu Lanchang	Coffee Shop	Thai Restaurant	Bakery	Food Court	Fast Food Restaurant	Food Truck	Market	Malay Restaurant	Noodle House	Satay Restaurant
3	Bayan Baru	Café	Chinese Restaurant	Dessert Shop	Korean Restaurant	Noodle House	Coffee Shop	Thai Restaurant	Pharmacy	Indian Restaurant	Bakery
4	Gelugor	Food Truck	Burger Joint	Malay Restaurant	Building	National Park	Food	Tennis Court	Asian Restaurant	Hotel	Farmers Market
5	Georgetown	Dessert Shop	Vegetarian / Vegan Restaurant	Hotel	Coffee Shop	Bakery	Noodle House	Café	Chinese Restaurant	Art Museum	Food Stand

4. Results Section

What we see in the map are the city districts and their most common venues, and they now have been assigned five different cluster labels from 0 to 4.We can now use the cluster labels to show the city districts marked with a cluster-specific color on a map, again using folium:

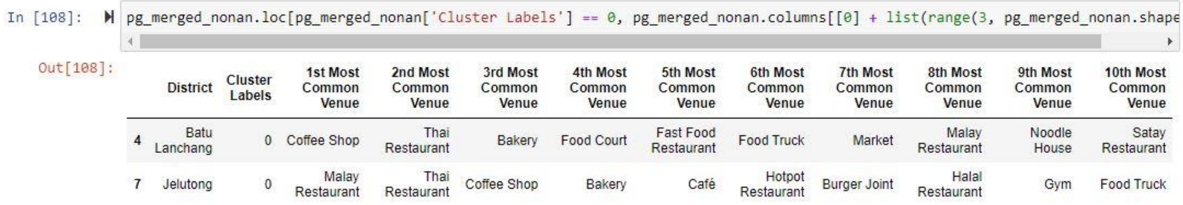
Map of Clusters in Northeast Penang

Out[107]:



You will see 5 colors for the city districts, for the five different clusters, which I named according to the venues concentration the data shows. There is 1 light green dot on top of the Island and 1 light orange in the middle if you can't see them.

Cluster 0 - The Thai Food Cluster

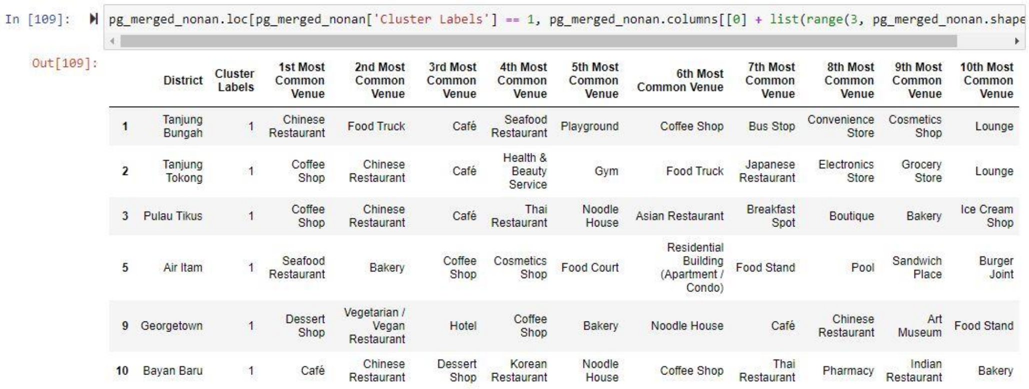


```
In [108]:  pg_merged_nonan.loc[pg_merged_nonan['Cluster Labels'] == 0, pg_merged_nonan.columns[[0] + list(range(3, pg_merged_nonan.shape
```

Out[108]:

	District	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	10th Most Common Venue
4	Batu Lanchang	0	Coffee Shop	Thai Restaurant	Bakery	Food Court	Fast Food Restaurant	Food Truck	Market	Malay Restaurant	Noodle House	Satay Restaurant
7	Jelutong	0	Malay Restaurant	Thai Restaurant	Coffee Shop	Bakery	Café	Hotpot Restaurant	Burger Joint	Halal Restaurant	Gym	Food Truck

Cluster 1 - The Chinese Food and Coffee shop Cluster

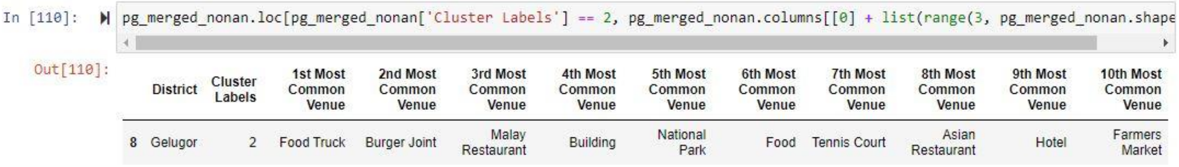


```
In [109]:  pg_merged_nonan.loc[pg_merged_nonan['Cluster Labels'] == 1, pg_merged_nonan.columns[[0] + list(range(3, pg_merged_nonan.shape
```

Out[109]:

	District	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	10th Most Common Venue
1	Tanjung Bungah	1	Chinese Restaurant	Food Truck	Café	Seafood Restaurant	Playground	Coffee Shop	Bus Stop	Convenience Store	Cosmetics Shop	Lounge
2	Tanjung Tokong	1	Coffee Shop	Chinese Restaurant	Café	Health & Beauty Service	Gym	Food Truck	Japanese Restaurant	Electronics Store	Grocery Store	Lounge
3	Pulau Tikus	1	Coffee Shop	Chinese Restaurant	Café	Thai Restaurant	Noodle House	Asian Restaurant	Breakfast Spot	Boutique	Bakery	Ice Cream Shop
5	Air Itam	1	Seafood Restaurant	Bakery	Coffee Shop	Cosmetics Shop	Food Court	Residential Building (Apartment / Condo)	Food Stand	Pool	Sandwich Place	Burger Joint
9	Georgetown	1	Dessert Shop	Vegetarian / Vegan Restaurant	Hotel	Coffee Shop	Bakery	Noodle House	Café	Chinese Restaurant	Art Museum	Food Stand
10	Bayan Baru	1	Café	Chinese Restaurant	Dessert Shop	Korean Restaurant	Noodle House	Coffee Shop	Thai Restaurant	Pharmacy	Indian Restaurant	Bakery

Cluster 2 - The Food Truck Cluster

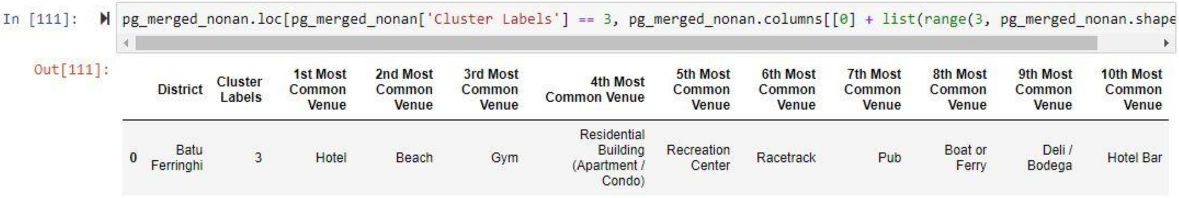


```
In [110]: pg_merged_nonan.loc[pg_merged_nonan['Cluster Labels'] == 2, pg_merged_nonan.columns[[0] + list(range(3, pg_merged_nonan.shape
```

Out[110]:

	District	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	10th Most Common Venue
8	Gelugor	2	Food Truck	Burger Joint	Malay Restaurant	Building	National Park	Food	Tennis Court	Asian Restaurant	Hotel	Farmers Market

Cluster 3 - The Hotel and Beach Cluster

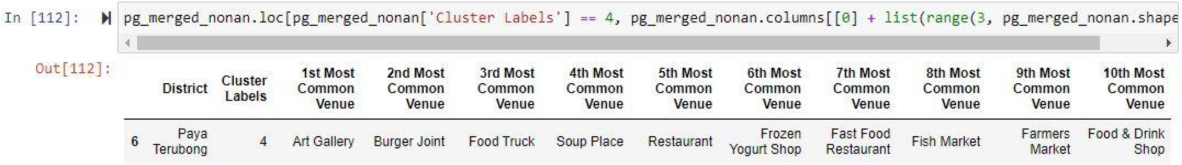


```
In [111]: pg_merged_nonan.loc[pg_merged_nonan['Cluster Labels'] == 3, pg_merged_nonan.columns[[0] + list(range(3, pg_merged_nonan.shape
```

Out[111]:

	District	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	10th Most Common Venue
0	Batu Ferringhi	3	Hotel	Beach	Gym	Residential Building (Apartment / Condo)	Recreation Center	Racetrack	Pub	Boat or Ferry	Deli / Bodega	Hotel Bar

Cluster 4 - The Art Cluster



```
In [112]: pg_merged_nonan.loc[pg_merged_nonan['Cluster Labels'] == 4, pg_merged_nonan.columns[[0] + list(range(3, pg_merged_nonan.shape
```

Out[112]:

	District	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	10th Most Common Venue
6	Paya Terubong	4	Art Gallery	Burger Joint	Food Truck	Soup Place	Restaurant	Frozen Yogurt Shop	Fast Food Restaurant	Fish Market	Farmers Market	Food & Drink Shop

5. Discussion Section

Problem Which Tried to Solve:

The major purpose of this project, is to suggest a place to visit in Penang for different tourist with specific interests. Tourists can visit the clusters based on their desires. Also, a simple laptop is enough to run the packages and provide comprehensive clustering for recommendation. What make this interesting is we could simply change the district name, from any location on earth, similar clustering can be done for other city.

6. Conclusion Section

Tourists can see in the results which city districts best match their desires. This is just one example of fantastic data science uses cases one can realize applying technology which is available for free today! I feel rewarded with the efforts and believe this course with all the topics covered is well worthy of appreciation. This project has shown me a practical application to resolve a real situation that has impacting personal and financial impact using Data Science tools. The mapping with Folium is a very powerful technique to consolidate information and make the analysis and decision better with confidence.