# Exploring the Neighborhoods in Singapore: Data Science in Real Life

As a part of the final IBM Capstone Project, we get a to know of what data scientists go through in real life. Objectives of the final assignments were to define a business problem, look for data in the web and, use Foursquare location data to compare different districts within wards (municipalities) of Singapore (choice of city depends on the students) to figure out which neighborhood is suitable for starting a restaurant business('idea' also depends on individual students).
Before we get the data and start exploring it, let's download all the dependencies that we will need.

## 1. Discussion and Background of the Business Problem:

### Problem Statement: Prospects of a Lunch Restaurant, Singapore.

Singapore's economic freedom score is 89.4, making its economy the 2nd freest in the 2019 Index. Its overall score has increased by 0.6 point, with increases in scores for trade freedom and government integrity outpacing modest declines in labor freedom and property rights. Singapore is ranked 2nd among 43 countries in the Asia–Pacific region, and its overall score is well above the regional and world averages.

The aim of this project is to explore the areas of Singapore and find the best place to open a breakfast cum lunch restaurant

### Target Audience

What type of clients or a group of people would be interested in this project?

1. Business personnel who want to invest or open a restaurant. This analysis will be a comprehensive guide to start or expand restaurants targeting the large pool of office workers in Singapore during lunch hours.
2. Freelancer who loves to have their own restaurant as a side business. This analysis will give an idea, how beneficial it is to open a restaurant and what are the pros and cons of this business.
3. New graduates, to find reasonable lunch/breakfast place close to office.
4. Budding Data Scientists, who want to implement some of the most used Exploratory Data Analysis techniques to obtain necessary data, analyze it, and, finally be able to tell a story out of it.

## 2. Data Preparation:

### 2.1. Get The Names of Wards, Major Districts and Population from Wikipedia

I first make use of page Planning_Areas_of_Singapore from Wiki to scrap the table to create a data-frame. For this, I've used requests and Beautifulsoup4 library to create a data-frame containing name of the 55 areas of Singapore, Region, population and density. We start as below —

```
from bs4 import BeautifulSoup

response_obj = requests.get('https://en.wikipedia.org/wiki/Planning_Areas_of_Singapore').text
print (type (response_obj))
```

```
<class 'str'>
```

```
soup = BeautifulSoup(response_obj,'html.parser')
print (soup.prettify())
```

After little manipulation, the data-frame is obtained as below —

```
Singapore_data
```

| | Name | Region | Area_SqKm | Population | Density_Per_SqKm |
|---|---|---|---|---|---|
| 0 | Ang Mo Kio | North-East | 13.94 | 165,710 | 12,000 |
| 1 | Bedok | East | 21.69 | 281,300 | 13,000 |
| 2 | Bishan | Central | 7.62 | 88,490 | 12,000 |
| 3 | Boon Lay | West | 8.23 | 30 | 3.6 |
| 4 | Bukit Batok | West | 11.13 | 144,410 | 13,000 |
| 5 | Bukit Merah | Central | 14.34 | 151,870 | 11,000 |
| 6 | Bukit Panjang | West | 8.99 | 140,820 | 16,000 |
| 7 | Bukit Timah | Central | 17.53 | 77,280 | 4,400 |
| 8 | Central Water Catchment | North | 37.15 | * | * |
| 9 | Changi | East | 40.61 | 2,080 | 62.3 |
| 10 | Changi Bay | East | 1.7 | * | * |
| 11 | Choa Chu Kang | West | 6.11 | 187,510 | 31,000 |
| 12 | Clementi | West | 9.49 | 93,000 | 9,800 |

## 2.2. Getting Coordinates of Areas : Geopy Client

Next objective is to get the coordinates of these 55 areas using geocoder class of Geopy client.

Using the code snippet as below —

```
from geopy.geocoders import Nominatim
geolocator = Nominatim()
Singapore_data_Final['Area_Name_Coord']= Singapore_data_Final['Name'].apply(geolocator.geocode).apply(lamb
da x: (x.latitude, x.longitude))

Singapore_data_Final
```

| | Name | Region | Area_SqKm | Area_Name_Coord |
|---|---|---|---|---|
| 1 | Ang Mo Kio | North-East | 13.94 | (1.369842, 103.8466086) |
| 2 | Bedok | East | 21.69 | (1.3239765, 103.930216) |
| 3 | Bishan | Central | 7.62 | (1.3514521, 103.8482496) |
| 4 | Boon Lay | West | 8.23 | (1.3456401, 103.7118018) |
| 5 | Bukit Batok | West | 11.13 | (1.3490572, 103.7495906) |
| 6 | Bukit Merah | Central | 14.34 | (4.5592879, 101.0255816) |
| 7 | Bukit Panjang | West | 8.99 | (1.377921, 103.7718658) |
| 8 | Bukit Timah | Central | 17.53 | (1.3546901, 103.7763724) |
| 9 | Central Water Catchment | North | 37.15 | (-33.55936435, 118.150468671534) |

## 2.3 Cleaning The Data

But here we see problem with coordinates for some places like Bukit Merah, Central Water Catchment, Changi, DowntownCore, Mandai, Museum, Newton, North-Eastern Islands, Orchard, Outram, Pioneer, Queenstown, RiverValley, Simpang, Tengah, Western Islands, Woodlands. So we need to replace them manually

**Final Data-Frame with Coordinates of the Major District**

```
Singapore_data_Final
```

| | Name | Region | Area_SqKm | Latitude | Longitude |
|---|---|---|---|---|---|
| 1 | Ang Mo Kio | North-East | 13.94 | 1.369842 | 103.846609 |
| 2 | Bedok | East | 21.69 | 1.323976 | 103.930216 |
| 3 | Bishan | Central | 7.62 | 1.351452 | 103.848250 |
| 4 | Boon Lay | West | 8.23 | 1.345640 | 103.711802 |
| 5 | Bukit Batok | West | 11.13 | 1.349057 | 103.749591 |
| 6 | Bukit Merah | Central | 14.34 | 1.281905 | 103.821711 |
| 7 | Bukit Panjang | West | 8.99 | 1.377921 | 103.771866 |
| 8 | Bukit Timah | Central | 17.53 | 1.354690 | 103.776372 |
| 9 | Central Water Catchment | North | 37.15 | 1.355205 | 103.795011 |
| 10 | Changi | East | 40.61 | 1.345005 | 103.981011 |
| 11 | Changi Bay | East | 1.7 | 1.316850 | 104.020649 |
| 12 | Choa Chu Kang | West | 6.11 | 1.389260 | 103.743728 |
| 13 | Clementi | West | 9.49 | 1.314026 | 103.762410 |
| 14 | Downtown Core | Central | 4.34 | 1.286705 | 103.851311 |
| 15 | Geylang | Central | 9.64 | 1.318186 | 103.887056 |

## 2.4. Using Foursquare Location Data:

Foursquare data is very comprehensive and it powers location data for Apple, Uber etc. For this business problem I have used, as a part of the assignment, the Foursquare API to retrieve information about the popular spots around all the Areas of Singapore. The popular spots returned depend on the highest foot traffic and thus it depends on the time when the call is made. So we may get different popular venues depending upon different time of the day. The call returns a JSON file and we need to turn that into a data-frame. Here I've chosen 100 popular spots for each Area within a radius of 1 km. Below is the data-frame obtained from the JSON file that was returned by Foursquare —

```
print(singapore_Venues.shape)
singapore_Venues.head()
```

```
(3240, 7)
```

| | Neighborhood | Neighborhood_Latitude | Neighborhood_Longitude | Venue | Venue_Latitude | Venue_Longitude | Venue_Category |
|---|---|---|---|---|---|---|---|
| 0 | Ang Mo Kio | 1.369842 | 103.846609 | Old Chang Kee | 1.369094 | 103.848389 | Snack Place |
| 1 | Ang Mo Kio | 1.369842 | 103.846609 | Bun Master | 1.369242 | 103.849031 | Bakery |
| 2 | Ang Mo Kio | 1.369842 | 103.846609 | Subway | 1.369136 | 103.847612 | Sandwich Place |
| 3 | Ang Mo Kio | 1.369842 | 103.846609 | MOS Burger | 1.369170 | 103.847831 | Burger Joint |
| 4 | Ang Mo Kio | 1.369842 | 103.846609 | PLAYe | 1.369109 | 103.848225 | Hobby Shop |

## 3. Visualization and Data Exploration:

### 3.1. Folium Library and Leaflet Map:

Folium is a python library that can create interactive leaflet map using coordinate data. Since I am interested in restaurants as popular spots first I create a data-frame where the 'Venue_Category' column in previous data-frame contains the word 'Restaurant'. I used the following snippet of code

```
# Create a Data-Frame out of it to Concentrate Only on Restaurants

Singapore_Venues_only_restaurant = singapore_Venues[singapore_Venues['Venue_Category']\
                                                    .str.contains('Restaurant')].reset_index(drop=True)
Singapore_Venues_only_restaurant.index = np.arange(1, len(Singapore_Venues_only_restaurant)+1)
print ("Shape of the Data-Frame with Venue Category only Restaurant: ", Singapore_Venues_only_restaurant.shape)
Singapore_Venues_only_restaurant.head(3)
```

```
Shape of the Data-Frame with Venue Category only Restaurant:  (964, 7)
```

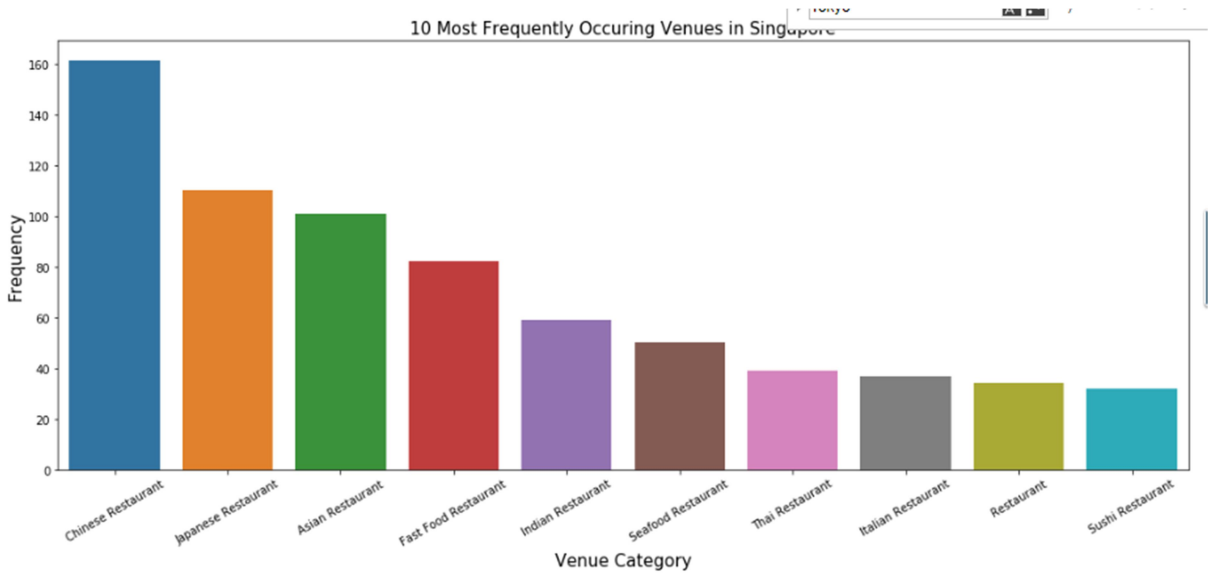
| | Neighborhood | Neighborhood_Latitude | Neighborhood_Longitude | Venue | Venue_Latitude | Venue_Longitude | Venue_Category |
|---|---|---|---|---|---|---|---|
| 1 | Ang Mo Kio | 1.369842 | 103.846609 | Kam Jia Zhuang Restaurant | 1.368167 | 103.844118 | Asian Restaurant |
| 2 | Ang Mo Kio | 1.369842 | 103.846609 | Collin's Grille . Bento | 1.371713 | 103.847526 | Modern European Restaurant |
| 3 | Ang Mo Kio | 1.369842 | 103.846609 | Xi Xiang Feng Yong Tau Foo 喜相逢酿豆腐 | 1.371975 | 103.846408 | Chinese Restaurant |

Next step is to use this data-frame to create a leaflet map with Folium to see the distribution of the most visited restaurants in the Areas.

```
## Show in Map the Top Rated Restaruants

map_restaurants = folium.Map(location=[latitude, longitude], zoom_start=11, tiles="openstreetmap",
                             attr="<a href=https://github.com/python-visualization/folium/>Folium</a>")

# set color scheme for the Venues based on the Major Districts
Area_Name = Singapore_data_Final["Name"]

x = np.arange(len(Area_Name))

#rainbow = ['#00ff00', '#ff00ff','#0000ff','#ffa500' ,'#ff0000']



# add markers to the map
# markers_colors = []
for lat, lon, poi, distr in zip(Singapore_Venues_only_restaurant['Venue_Latitude'],
                                Singapore_Venues_only_restaurant['Venue_Longitude'],
                                Singapore_Venues_only_restaurant['Venue_Category'],
                                Singapore_Venues_only_restaurant['Neighborhood']):
    label = folium.Popup(str(poi) + ' ' + str(distr), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=7,
        popup=label,
 #       color=rainbow[Area_Name.index(distr)-1],
        fill=True,
  #      fill_color=rainbow[Area_Name.index(distr)-1],
        fill_opacity=0.3).add_to(map_restaurants)

map_restaurants
```

## 3.2. Exploratory Data Analysis:

There are 59 unique venue categories and Chinese Restaurants top the charts as we can see in the plot below —




To know about the top 5 venues of each Area we proceed as follows

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

- Use pandas groupby on the District column and obtain the mean of the one-hot encoded venue categories.

- Transpose the data-frame at step 2 and arrange in descending order.

Let's see the code snippet below —

```
# one hot encoding
singapore_onehot = pd.get_dummies(Singapore_Venues_only_restaurant[['Venue_Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
singapore_onehot['Neighborhood'] = Singapore_Venues_only_restaurant['Neighborhood']

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

singapore_onehot.head()
```

```
singapore_grouped = singapore_onehot.groupby('Neighborhood').mean().reset_index()
singapore_grouped
```

| | Neighborhood | American Restaurant | Asian Restaurant | Australian Restaurant | Belgian Restaurant | Cantonese Restaurant | Chinese Restaurant | Comfort Food Restaurant | Dim Sum Restaurant | Dumpling Restaurant | English Restaurant | Falafel Restaurant | Fast Food Restaurant | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Ang Mo Kio | 0.000000 | 0.047619 | 0.000000 | 0.000000 | 0.000000 | 0.190476 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.190476 | |
| 1 | Bedok | 0.029412 | 0.088235 | 0.000000 | 0.000000 | 0.000000 | 0.176471 | 0.000000 | 0.000000 | 0.029412 | 0.000000 | 0.000000 | 0.088235 | |
| 2 | Bishan | 0.000000 | 0.125000 | 0.000000 | 0.000000 | 0.000000 | 0.187500 | 0.000000 | 0.000000 | 0.062500 | 0.000000 | 0.000000 | 0.000000 | |
| 3 | Boon Lay | 0.000000 | 0.222222 | 0.000000 | 0.000000 | 0.000000 | 0.111111 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.148148 | |
| 4 | Bukit Batok | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.363636 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.363636 | |
| 5 | Bukit Merah | 0.041667 | 0.041667 | 0.000000 | 0.000000 | 0.000000 | 0.375000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.083333 | |
| 6 | Bukit Panjang | 0.181818 | 0.272727 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.272727 | |
| 7 | Bukit Timah | 0.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | |
| 8 | Changi | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.200000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.400000 | |
| 9 | Choa Chu Kang | 0.000000 | 0.083333 | 0.000000 | 0.000000 | 0.000000 | 0.083333 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.333333 | |
| 10 | Clementi | 0.000000 | 0.148148 | 0.000000 | 0.000000 | 0.000000 | 0.259259 | 0.000000 | 0.037037 | 0.000000 | 0.000000 | 0.000000 | 0.111111 | |
| 11 | Downtown Core | 0.000000 | 0.060606 | 0.000000 | 0.000000 | 0.000000 | 0.121212 | 0.030303 | 0.000000 | 0.030303 | 0.000000 | 0.000000 | 0.000000 | |
| 12 | Geylang | 0.000000 | 0.155556 | 0.000000 | 0.000000 | 0.022222 | 0.200000 | 0.000000 | 0.044444 | 0.000000 | 0.000000 | 0.000000 | 0.088889 | |

```
num_top_venues = 5

for hood in singapore_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = singapore_grouped[singapore_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')
```

```
----Ang Mo Kio----
                  venue  freq
0  Fast Food Restaurant  0.19
1    Chinese Restaurant  0.19
2   Japanese Restaurant  0.14
3      Malay Restaurant  0.05
4      Asian Restaurant  0.05


----Bedok----
                        venue  freq
0            Chinese Restaurant  0.18
1              Asian Restaurant  0.09
2           Japanese Restaurant  0.09
3          Fast Food Restaurant  0.09
4  Vegetarian / Vegan Restaurant  0.06


----Bishan----
```

## 4. Clustering the Districts

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

```
# import k-means from clustering stage
from sklearn.cluster import KMeans

# set number of clusters
kclusters = 5

singapore_grouped_clustering = singapore_grouped.drop('Neighborhood', 1)

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

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

Tokyo

```
array([1, 1, 0, 1, 0, 0, 1, 3, 1, 1], dtype=int32)
```

Let's create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.

```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'ClusterLabels', kmeans.labels_)

singapore_merged = Singapore_data_Final

# merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
singapore_merged = singapore_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Name')

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

| | Name | Region | Area_SqKm | Latitude | Longitude | ClusterLabels | 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 Mc Comm Ven |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Ang Mo Kio | North-East | 13.94 | 1.369842 | 103.846609 | 1.0 | Fast Food Restaurant | Chinese Restaurant | Japanese Restaurant | Shanghai Restaurant | Indian Restaurant | Modern European Restaurant | Halal Restaurant | Ramen Restaurant | Seafo Restaura |
| 2 | Bedok | East | 21.69 | 1.323976 | 103.930216 | 1.0 | Chinese Restaurant | Asian Restaurant | Fast Food Restaurant | Japanese Restaurant | Vegetarian / Vegan Restaurant | Thai Restaurant | French Restaurant | Sushi Restaurant | Indi Restaura |
| 3 | Bishan | Central | 7.62 | 1.351452 | 103.848250 | 0.0 | Thai Restaurant | Chinese Restaurant | Seafood Restaurant | Asian Restaurant | Japanese Restaurant | Italian Restaurant | Hakka Restaurant | Dumpling Restaurant | Ho Ko Restaura |
| 4 | Boon Lay | West | 8.23 | 1.345640 | 103.711802 | 1.0 | Asian Restaurant | Japanese Restaurant | Fast Food Restaurant | Indian Restaurant | Chinese Restaurant | Malay Restaurant | Japanese Curry Restaurant | Hong Kong Restaurant | Ha Restaura |
| 5 | Bukit Batok | West | 11.13 | 1.349057 | 103.749591 | 0.0 | Chinese Restaurant | Fast Food Restaurant | Malay Restaurant | Halal Restaurant | Greek Restaurant | Kebab Restaurant | Japanese Restaurant | Japanese Curry Restaurant | Itali Restaura |

## 5. Results and Discussion:

We reached at the end of the analysis, where we got a sneak peak of the different areas of Singapore and, as the business problem started with benefits and drawbacks of opening a lunch restaurant in one of the busiest areas, the data exploration was mostly concentrated on the restaurants. I have used data from web resources like Wikipedia, python libraries like Geopy, and Foursquare API, to set up a very realistic data-analysis scenario. We have found out that —

1. Chinese restaurants top the charts of most common venues in Singapore.
2. Geylang (Central region) and Novena (Central region) are dominated by restaurants as the most common venue.

According to this analysis, central region is highly populated with both commercial and residential area. Furthermore, this results also could potentially vary if we use some other clustering techniques like DBSCAN

## 6. Conclusion

Finally to conclude this project, we have got a small glimpse of how real life data-science projects look like. I've made use of some frequently used python libraries to scrap web-data, use Foursquare API to explore the areas of Singapore and saw the results of segmentation of districts using Folium leaflet map. Potential for this kind of analysis in a real life business problem is discussed in great detail. Also, some of the drawbacks and chance for improvements to represent even more realistic pictures are mentioned. Finally, since my analysis were mostly concentrated on the possibilities of opening restaurants targeting the huge pool of office workers. Hopefully, this kind of analysis will provide you initial guidance to take more real-life challenges using data-science.