BATTLE OF NEIGHBORHOOD CAPSTONE REPORT

BY

MANISH

1. INTRODUCTION

   Singapore is an economic hub even though it is small in size but it has been Southeast Asia most modern city for over a century. The city hosts people from multiple cultures & religions but not limited to Malay, Chinese, Indian and Arab etc. Due to efficient and determined government, Singapore has become a flourishing country for trade and tourism. This has created a lot of job opportunities in the country. Immigration to Singapore is historically the main source of population growth in the country since the founding of modern Singapore in the early 19th century. Immigration & immigration workers have been closely associated with Singapore's economic development.

2. BUSINESS PROBLEM

   For the current capstone project "Battle of Neighborhood", I am doing analysis for Singapore neighborhood where find the right accommodation is really tough depending upon the location & places to eat. Some places like living near to work place but with limited availability of resources. This Capstone project can help in discussing some of the below problems:

- Finding the right accommodation for person who is moving to new location.
- Finding the right location where there are their required facilities available
- Finding the neighborhood near to particular attractions such as parks, galleries, specific restaurants.

3. DATA

   For this project we need the following data:

   Singapore neighborhood data which contains list of Building Names, their latitudes and longitudes.

   Data source: open repository: https://github.com/xkjyeah/singapore-postal-codes.

   Description: This data set contains the required information. And we will use this data set to explore various neighborhood of each locality.

For getting all venues within 500 meters and their geographical coordinates and venues categories.

Data source: Foursquare API: "https://developer.foursquare.com/"

Description: By using this API we will get all the venues in each neighborhood.

4. DATA CLEANING & PROCESSING

   For the purpose of data fetching, collection, cleaning & processing, jupyter notebook is used. Following steps and process is being followed.

   - Import of required libraries.

[1]:
```
import numpy as np # library to handle data in a vectorized manner

import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

!conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

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

#!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.9.1
  latest version: 4.9.2

Please update conda by running

    $ conda update -n base -c defaults conda



# All requested packages already installed.

Libraries imported.
```

   - Importing Building data from json file

[2]:
```
with open('buildings.json') as building_data:
    data = json.load(building_data)
```

[3]:
```
sg_data = pd.json_normalize(data)
```

[4]:
```
sg_data.shape
```

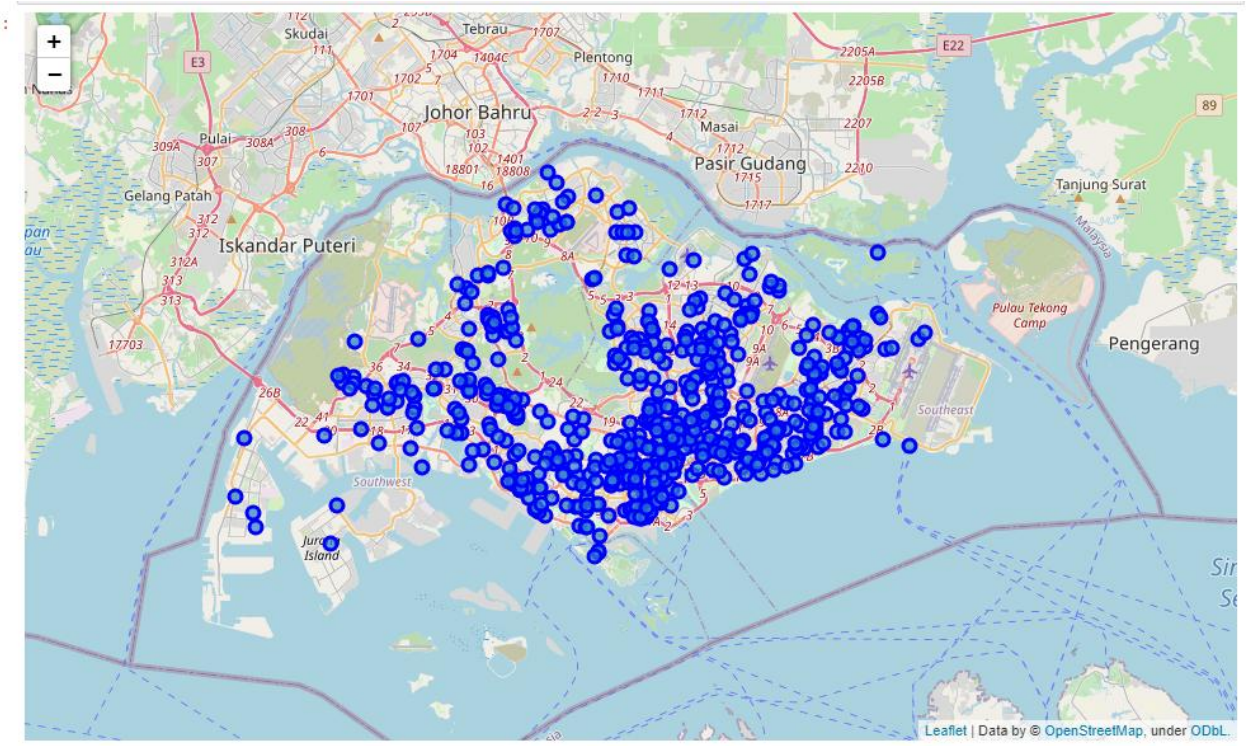
[4]:
```
(141726, 11)
```

There are 141726 samples available in the dataset and there are 11 columns. This data needs to be filtered and duplicates need to be removed. Initially, only required columns are kept and others are deleted. Columns which are kept are BUILDING name, postal code, latitude and longitude.

Later all the values where there is no value for BUILDING name are ignored as they will not add any advantage to us.
Further it is checked that if there are any multiple entries with same building/neighborhood and only 1 entry corresponding to the neighborhood is being kept.
Next, all entries related to conservation area is also being removed.
After cleaning, the final dataset shape is (16150, 4)

5. ANALYSIS

For this assignment and analysis purpose, we will take only random 600 samples from the whole dataset and analyze this data.
First using folium library, we will print the map for the Singapore neighborhood and its location on the map.



We will start with picking 1 one of the neighborhood
The first neighborhood is TAMPINES GREENWOOD and using foursquare credentials, all the venues under radius of 500m is being fetched. Total 7 number of unique venues are being fetched using foursquare API credentials.

Out[41]:

	name	categories	lat	lng
0	Tampines Central Park	Park	1.354111	103.936393
1	NTUC Fairprice	Supermarket	1.355541	103.934758
2	Playground @ Tampines Blk 869	Playground	1.354625	103.933893
3	Madison's	Sandwich Place	1.354242	103.933217
4	Bus Stop 75139 (Blk 863)	Bus Station	1.355861	103.936358

[42]:
```
print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))
```
```
7 venues were returned by Foursquare.
```

Now, all the neighborhoods present in the dataset are analyzed. Venues for each neighborhood are being fetched using Foursquare API and it is found that there are 373 unique venues in the entire neighborhood.

Further, each neighborhood is being analyzed and top 5 venues for each neighborhood is being calculated. During analysis, it was found that there are couple of venues which are being named as neighborhood which interferes with our original neighborhood when the venue data is being merged with original dataset of 600 samples. For this purpose, those venues are being ignored and removed from the venue list.

In [54]:
```
num_top_venues = 5

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

```
----57 @ KOVAN----
                          venue  freq
0              Indian Restaurant  0.11
1               Asian Restaurant  0.11
2                    Food Court  0.11
3                   Noodle House  0.11
4  Vegetarian / Vegan Restaurant  0.11


----ACACIA WELFARE HOME----
            venue  freq
0     Bus Station   0.2
1  Harbor / Marina   0.1
2  Thai Restaurant   0.1
3       Baby Store   0.1
4  Automotive Shop   0.1
```

Now, top 10 venues for each neighborhood is calculated as below:

In [56]:
```
num_top_venues = 10

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'] = sg_grouped['Neighborhood']

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

neighborhoods_venues_sorted.head()
```

Out[56]:

	Neighborhood	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	57 @ KOVAN	Vegetarian / Vegan Restaurant	Food Court	Asian Restaurant	Noodle House	Indian Restaurant	Supermarket	Bar	Sandwich Place	Park	Furniture / Home Store
1	ACACIA WELFARE HOME	Bus Station	Harbor / Marina	Thai Restaurant	Baby Store	Café	Automotive Shop	Kids Store	Bus Stop	Asian Restaurant	Zoo Exhibit
2	AL - ISTIQAMAH MOSQUE KINDERGARTEN	Asian Restaurant	Pet Store	Coffee Shop	Supermarket	Playground	Noodle House	Chinese Restaurant	Breakfast Spot	Food Court	Electronics Store
3	ALJUNIED COMMUNITY CENTRE	Coffee Shop	Noodle House	Bakery	Food Court	Café	Cafeteria	Breakfast Spot	Gym	Bus Stop	Bus Station
4	ALKAFF MANSION	Bakery	Bus Station	Scenic Lookout	Trail	Bus Line	Karaoke Bar	Gym	Miscellaneous Shop	Clothing Store	Club House

In [57]:
```
neighborhoods_venues_sorted.shape
```

Out[57]: (596, 11)

6. MODELLING

Now clustering of the data is being done. So we use K-Means algorithm to create clusters. There are 5 number of cluster being selected.

4. Cluster Neighborhoods

```
# set number of clusters
kclusters = 5

sg_grouped_clustering = sg_grouped.drop('Neighborhood', 1)

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

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

```
58]: array([2, 1, 2, 3, 3, 2, 2, 2, 3, 2])
```

```
len(kmeans.labels_)
```

```
59]: 596
```

```
neighborhoods_venues_sorted.shape
```

```
60]: (596, 11)
```

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

Next, cluster labels are merged with neighborhood data with top 10 venues.

```
In [75]: # merge manhattan_grouped with manhattan_data to add latitude/longitude for each neighborhood
sg_merged = sg_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

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

Out[75]:

	Neighborhood	LATITUDE	LONGITUDE	POSTAL	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 Mc Comm Ven
0	TAMPINES GREENWOOD	1.355674	103.933008	524868	1	Park	Track	Wine Bar	Playground	Bus Station	Supermarket	Sandwich Place	Z Exh
1	CHENG HONG SIANG TNG KEW HUANG KENG COMBINED ...	1.325754	103.892159	409966	3	Food Court	Coffee Shop	Café	Metro Station	BBQ Joint	Diner	Auto Garage	B Stati
2	GRACE CHILD DEVELOPMENT CENTRE	1.389292	103.900231	544644	3	Breakfast Spot	Shopping Mall	Basketball Court	Park	Indonesian Restaurant	Supermarket	Gym / Fitness Center	Fast Fo Restaur
3	CHEERY HUES EDUCATION PTE. LTD.	1.331255	103.943053	488531	2	Asian Restaurant	Food Court	Chinese Restaurant	Seafood Restaurant	Dessert Shop	Shopping Mall	Hong Kong Restaurant	Fish Ch Sh
4	HARBOURFRONT SINGAPORE	1.265343	103.821069	098867	3	Chinese Restaurant	Japanese Restaurant	Clothing Store	Toy / Game Store	Fast Food Restaurant	Bakery	Multiplex	Coff Sh

Further, map is created for different clusters using folium library.

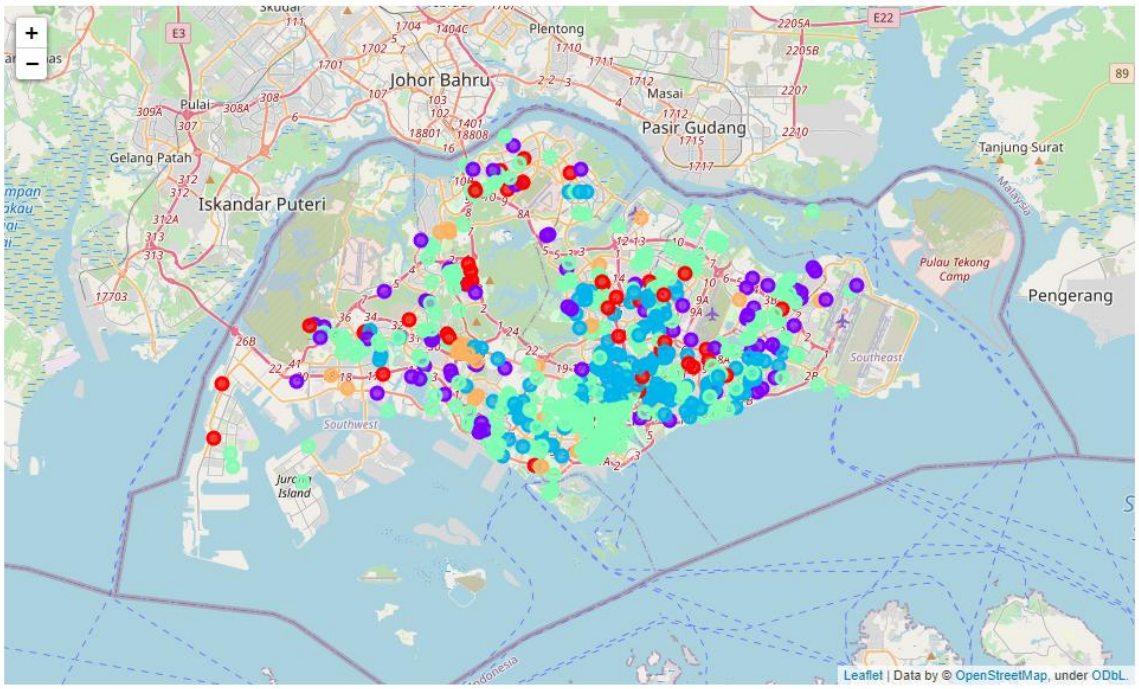
```
# create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i + x + (i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(sg_merged['LATITUDE'], sg_merged['LONGITUDE'], sg_merged['Neighborhood'], sg_merged['Cluste
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

Out[77]:



7. RESULTS

Case 1: If immigrant is moving to Singapore, which would be best place to move in?

On checking the number of neighborhoods in the merged data, it is found that cluster 0 and 4 are having least number of neighborhoods.

```
In [79]: sg_merged[sg_merged['Cluster Labels']==0].shape
Out[79]: (42, 15)

In [80]: sg_merged[sg_merged['Cluster Labels']==1].shape
Out[80]: (72, 15)

In [81]: sg_merged[sg_merged['Cluster Labels']==2].shape
Out[81]: (165, 15)

In [82]: sg_merged[sg_merged['Cluster Labels']==3].shape
Out[82]: (289, 15)

In [83]: sg_merged[sg_merged['Cluster Labels']==4].shape
Out[83]: (28, 15)
```

Cluster 3 is the one which has the most number of neighborhoods. This would be the best place for immigrant worker to settle down.

Case-2: A new migrant moved to place where it was expensive for him but he likes the neighborhood. Which would be best place for him considering if similar neighborhood can be available for him?

Case-2

A person staying in Pan Pacific Serviced Suites Beach Road wants to move to new location as it is expensive. Using this, it can be determined that which cluster would be most suitable

```
In [91]: sg_merged[sg_merged['Neighborhood']=='PAN PACIFIC SERVICED SUITES BEACH ROAD']['Cluster Labels'].iloc[0]
Out[91]: 3
```

```
In [92]: sg_merged[sg_merged['Cluster Labels']==3]
```

	CENTRE						Terminal	Restaurant			Restaurant
34	OCBC THE CENTRAL	1.288839	103.846558	059815	3	Japanese Restaurant	Nightclub	Bar	Hotel	Food Court	
37	BEDOK WATERWORKS	1.341781	103.917483	417901	3	Bus Line	Zoo Exhibit	Farm	Dumpling Restaurant	Duty-free Shop	
40	FERNVALE FLORA	1.393417	103.875340	794453	3	Coffee Shop	Asian Restaurant	Fast Food Restaurant	Food Court	Shoe Store	
45	THOMSON VIEW CONDOMINIUM	1.358909	103.829150	579635	3	Bakery	Creperie	Sporting Goods Shop	Shopping Mall	Asian Restaurant	
47	TOP TEN	1.304844	103.839705	229415	3	Hotel	Japanese Restaurant	Shopping Mall	Clothing Store	Steakhouse	
49	BUANGKOK GREEN MEDICAL PARK (BLOCK 2)	1.381734	103.883471	539747	3	Grocery Store	Supermarket	Playground	Fast Food Restaurant	Zoo Exhibit	
51	PCF SPARKLETOTS PRESCHOOL @ MARSILING	1.431004	103.781602	730339	3	Shopping Mall	Fast Food	Supermarket	Food Court	BBQ Joint	

The person can move to any other neighborhood in cluster -3. Something like FERNVALE FLORA

Case 3: All those neighborhoods which are near to Park

```
In [97]: sg_venues[sg_venues['Venue Category']=='Park']
```

Out[97]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	TAMPINES GREENWOOD	1.355674	103.933008	Tampines Central Park	1.354111	103.936393	Park
31	GRACE CHILD DEVELOPMENT CENTRE	1.389292	103.900231	STARLIGHT	1.393445	103.899074	Park
265	UOB UPPER ALJUNIED ROAD	1.332946	103.878886	Aljunied Park	1.329599	103.880724	Park
351	KLC INTERNATIONAL INSTITUTE	1.350887	103.873968	Serangoon Sunshine Park	1.347798	103.874320	Park
407	STANDARD CHARTERED BANK NEX SERANGOON	1.350783	103.872565	Serangoon Sunshine Park	1.347798	103.874320	Park
605	DBS IKEA TAMPINES	1.374073	103.932661	Brontosaurus Park	1.374578	103.936812	Park
684	RAINBOW CENTRE - MARGARET DRIVE SCHOOL	1.297513	103.809222	Alexandra Canal Linear Park	1.294646	103.811531	Park
714	STA JALAN BOON LAY INSPECTION CENTRE	1.338481	103.710699	Jurong Central Park	1.339250	103.708696	Park
732	DBS PENDING ROAD	1.376028	103.769749	Petir Park	1.375687	103.768567	Park