Applied Data Science Capstone Project

CHOOSING THE TOWN
TO LIVE IN SINGAPORE

BOB DJ

January 2021

IBM Data Science Professional Certificate

SECTION 1 : INTRODUCTION

" Life is about choices. Some we regret, some we're proud of. Some will haunt us forever "

Making a choice is always a difficult thing to do, because we don't know if we will be happy with our choice or regret it, days / weeks / months from when we made the choice. Let's face the truth, most of the times we regret our choices, saying I should have talked to more people before deciding on something or 'done more research'. The other side is always greener or the food on the next plate always looks more tastier etc. and we could go on with the quotes or proverbs that highlight the effect and perception of a choice that we make in life.

" Before you make a choice make sure you can live with it "

Especially if you have to choose your place of living, you have to literally live with your choice. Choosing your place of living is a big choice for almost everyone and we try to get as much details possible before we make that leap of faith.

Singapore (SG) the city state is an amazing place to live in, if only we knew where to make our choice of stay. It has a plethora of attractions, restaurants, amenities, venues, towns to choose from. That statement poses the first problem for anyone hunting for a place to stay. Fortunately that's where Data Science comes in, as we have abundant data and the proper knowledge, tools and means to sift through them to slice and dice it, and get meaningful results out of it in an easy to understand and visually attractive format.

PROBLEM TO SOLVE - TARGET AUDIENCE

A prospective homebuyer or a tenant wannabe decides to choose a place to live in the island of Singapore. The person has a few basic specifications or needs that cannot be compromised. Let's call the person with a gender neutral common name - Robin. Robin has a few specifications or conditions that are a must to be adhered to as a basic need. This Capstone project paper is an attempt to help Robin or any similar person to navigate that first essential step towards finalising the place to live. So anyone who wants to buy a property or find a place to live in Singapore is a target audience for this paper. Robin's basic requirements regarding the town/place of stay in Singapore is

1. There should not be too many people living around.

2. Some key amenities are required to be around the living area within a walking distance

3. There should be an open water area close by.

This paper aims to help solve the above requirements (addressed as problem to solve going forward) using the concepts learnt and practised in the Coursera IBM Data Science Professional Certificate course.

To elaborate a bit more on the background of the problem, Robin wants to live in a locality that has less number of people compared to surrounding areas and has amenities for doing exercises and workouts, do regular shopping and places to relax having a drink with friends and family. Robin has a preference towards particular food and drinks that need to be taken into account also. The final requirement is the presence of a waterbody in the vicinity to have a peaceful walk around or if there are sports options for a swim or canoeing etc.

The above 3 problems are objectively quantified in the below table

Problem statement	Interpretation	Variable	Quantification
1. There should not be too many people living around	The number of people living around should be less than average of people across the towns of Singapore	Population	Population < Average population across SG Towns
1. Some amenities are required to be around the living area in a walking distance	Requirement is to have exercising options like gym and Shopping Malls close by, Needs options for drinking with places serving whisky and wine in that order.	Venues	Venue choices to be within a 1 kilometre radius
1. There should be a open water area close by	Any kind of waterbody like a Pond, River, Stream or Sea to be in the vicinity.	Natural feature	Feature to be within a 1 kilometre radius

SECTION 2 DESCRIPTION OF DATA

To help solve the above described problem, 3 different types of data were required, first for identifying the various towns in Singapore and their population, second for getting their coordinates to plot in the map and to third to help finding venues, analysing details regarding the venues and natural features.

On researching the internet for the above sets of data, the following 3 sets of data satisfying the entire requirement was discovered.

1. Singapore towns and their population : The Singapore Government publishes data sets available for public view in **data.gov.sg** and this site has a specific and suitable data set for the project's requirement. The Housing and Development Board (HDB) of Singapore has a data set with the various SG Towns and their population for a 10 year period between 2008 and 2018.

2. Singapore towns' Co-ordinates : For getting the SG Town's coordinates of latitude and longitude we will use the Python Geocoder ArcGIS package

3. Venue Details of Singapore towns : Foursquare API will be used to get the venue details of the SG Towns. The Foursquare API provides comprehensive location data that can be used to get details of nearby venues, and details of these venues in a given area of SG Towns. Interestingly, it also has natural features like river & Waterfront that will be useful in our specific project requirement. **https://developer.foursquare.com/developer/**

SECTION 3 : METHODOLOGY

The sequence of steps to be followed in solving the problem for this Project will be as follows

A. First install the required packages of Python and import relevant Python libraries that are required for data loading, wrangling, analysis, graphs and maps rendering

B. Next get the available data on towns of Singapore from the SG Government's official data website and load that data into the Jupiter notebook

C. Then analyse and clean the SG towns' data and shortlist the data based on our requirement.

D. Then use the Geocoder package to get the geographical coordinates (Latitude & Longitude) for the SG towns

E. Followed by using the Foursquare API to receive the nearby venues and their details for all the towns that are shortlisted.

F. Finally analyse the venue data based on our initial 3 requirements, and finalise the most appropriate SG Town where Robin can choose to stay

G. As an additional step, display the SG map with the shortlisted SG Towns and the selected township will be hardcoded and displayed with a marker.

A. INSTALLATION & IMPORT OF PACKAGES

First the required packages of Python and import relevant Python libraries that are required for data loading, wrangling, analysis, graphs and maps rendering were installed.

```
! pip install folium==0.5.0
print('Installed Folium for the map')
!pip install geopy
print('Installed Geopy')
!pip install geocoder
print('Installed Geocoder')
```

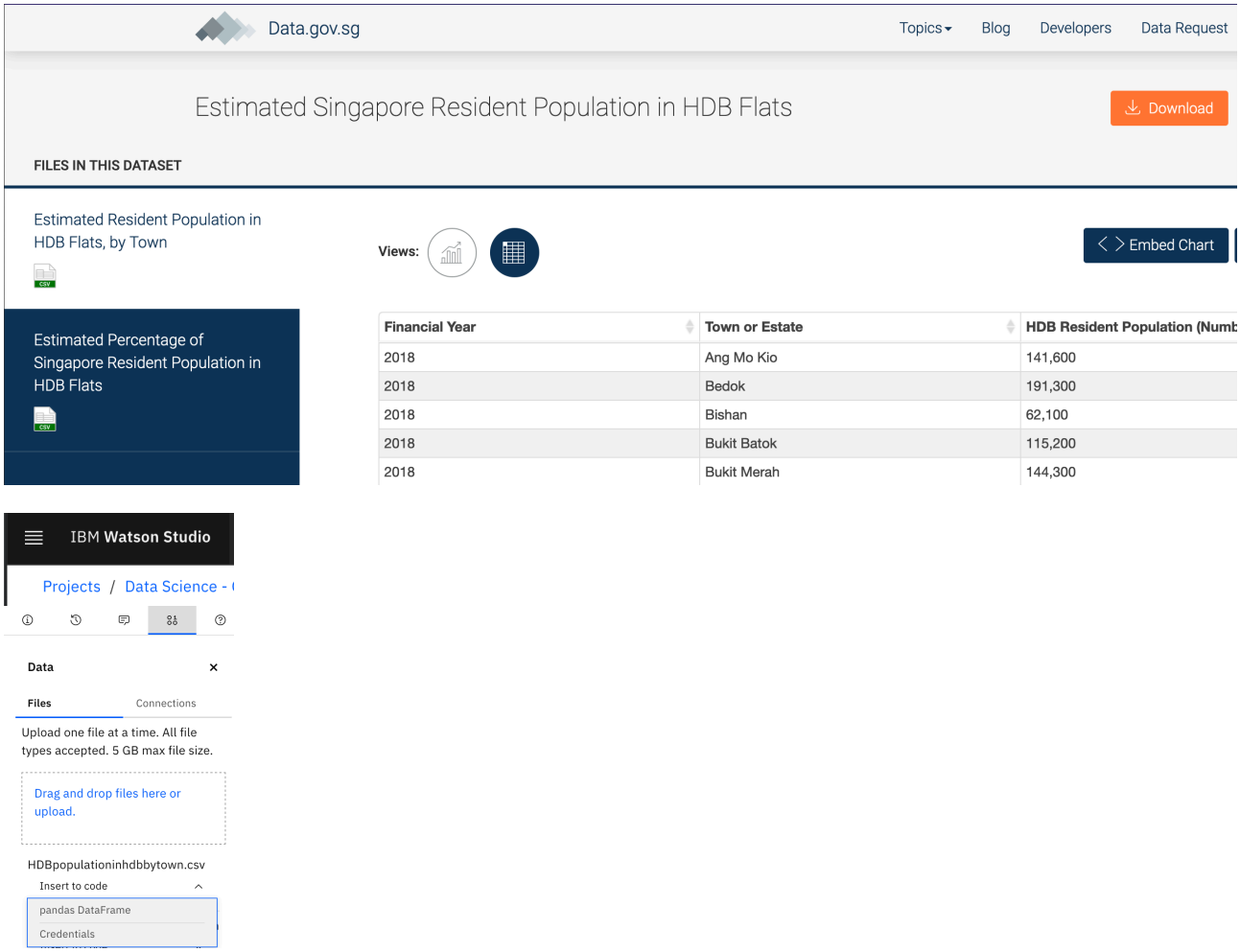
```
import pandas as pd # This library is for data analsysis
import numpy as np  # This library is for handling data
import requests # This library is for handling requests
import folium   # This library is for creating the maps
import json     # This library is for handling json files
import geocoder # This library is for getting coordinates
from geopy.geocoders import Nominatim     # This library is to convert into latitude and longitude values
from IPython.display import Image         # THis library is for rendering the map images
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe
import matplotlib.pyplot as plt           # This library is to plot bar graphs etc
import matplotlib.cm as cm
import matplotlib.colors as colors

print('All the necessary libraries have been imported')
```

```
All the necessary libraries have been imported
```

A print message is at the end of all the installations and import to signal that it's completed.

B. DATA LOADING

Next the available data on towns of Singapore & relevant population from the SG Government's official data website was identified. The data was loaded into the Jupyter notebook in IBM Watson studio with the Insert to code option utilising the pandas DataFrame option and utilising the auto generated code and reading into a DataFrame.



As the above screenshots show, the SG Government data is in a tabular format with Financial Year, Town or Estate and HDB Resident population columns. This data in .csv format was downloaded to the local computer and using the data upload facility in IBM Watson studio. From there the tool offers a simple GUI based option to read the csv file Ito the jpny. Using the option of Insert into code —> pandas dataFrame, the csv file was read into the Jupiter notebook into our Capstone Project .

A basic analysis of the data shows that the imported csv file had 296 rows and 3 columns of data related to the year, town and population.

Once the data was loaded into the code of Capstone project in the Jupiter notebook, the next step of analysing the data was initiated.

C. DATA ANALYSIS / WRANGLING

The data read into the pandas DataFrame was then analysed, for presence of any empty values in the rows or columns and any corrupted data. The column headers of the data was renamed into Year, SGTown and Pop for ease of handling coding.

The 296 rows and 3 columns of data read into the data frame df_HDB was clean data that did not require much of data cleaning. The data was limited to the year 2018 and the rest were dropped. Once the 2018 data was selected the column Year was not required anymore and was then dropped to make the data set relevant. The data was also further reduced later to the expected population as per requirement, with towns selected having a population less than average across the towns of Singapore.

In [4]:
```
df_HDB = pd.read_csv(body)
df_HDB.rename(columns = {'financial_year':'YEAR','town_or_estate':'SGTOWN','population':'POP'}, inplace = True)
df_HDB
```

Out[4]:

	YEAR	SGTOWN	POP
0	2008	Ang Mo Kio	148200
1	2008	Bedok	197900
2	2008	Bishan	66500
3	2008	Bukit Batok	109100
4	2008	Bukit Merah	140600
...	...	...	...
290	2018	Serangoon	68000
291	2018	Tampines	231800
292	2018	Toa Payoh	104200
293	2018	Woodlands	242800
294	2018	Yishun	197300

295 rows × 3 columns

In [5]:
```
# Choosing only the latest year data and then dropping the year column and resetting the index
df_HDB1 = df_HDB[df_HDB['YEAR']==2018]
df_HDB1 = df_HDB1.drop(columns=['YEAR'])
df_HDB1.reset_index(drop=True, inplace=True)
df_HDB1
```

Out[5]:

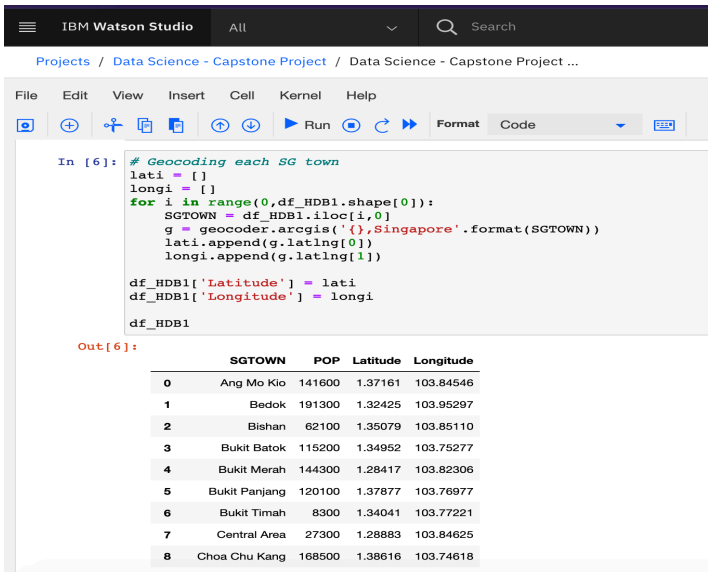
	SGTOWN	POP
0	Ang Mo Kio	141600
1	Bedok	191300
2	Bishan	62100
3	Bukit Batok	115200
4	Bukit Merah	144300
5	Bukit Panjang	120100
6	Bukit Timah	8300
7	Central Area	27300
8	Choa Chu Kang	168500
9	Clementi	71900

At the end of the data analysis and wrangling exercise, the dataset had 26 rows and 2 columns now with the 26 towns of Singapore and their population in the year 2018. It was then time to get the coordinates.

D. GETTING COORDINATES

The Geocoder package was used to get the geographical coordinates (Latitude & Longitude) for the SG towns. So the resulting DataFame now had 4 columns - towns, population, latitude and the longitude. The number of records remained the same at 26 rows for the 26 towns of Singapore.

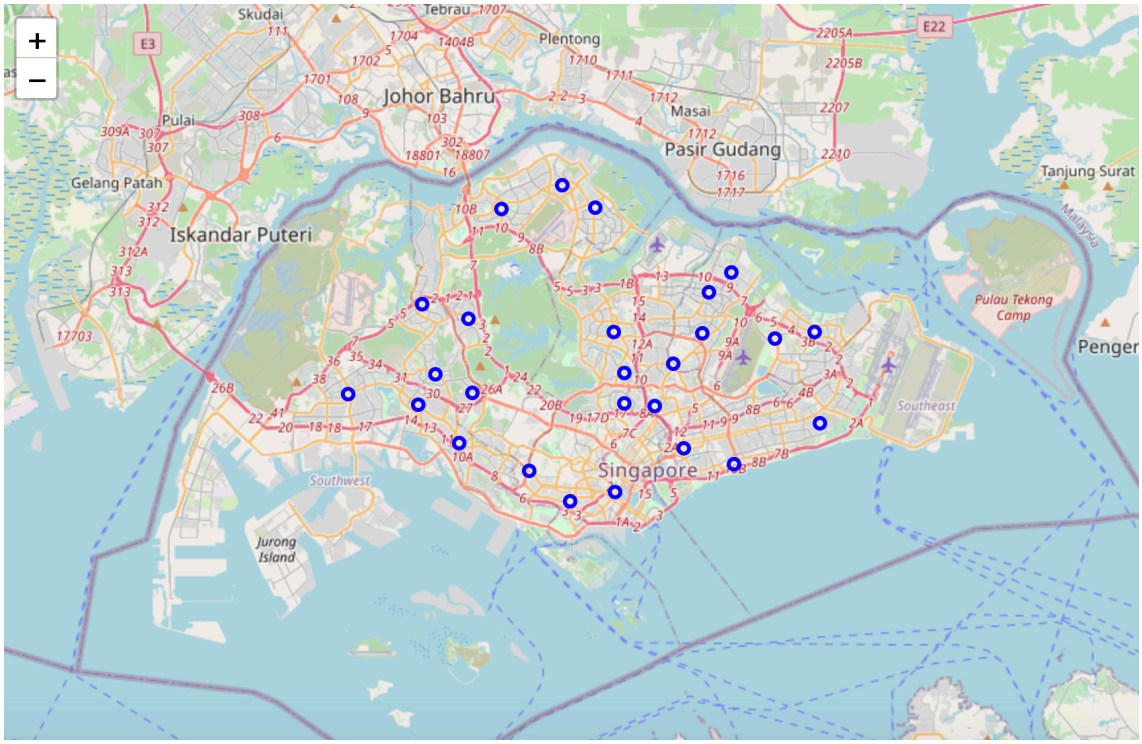
And to validate the veracity of the coordinates data, a simple map was displayed, using folium. The resultant map showed all the 26 towns of Singapore using the coordinates read into the DataFrame df_HDB1.




```
In [6]: # Geocoding each SG town
lati = []
longi = []
for i in range(0,df_HDB1.shape[0]):
    SGTOWN = df_HDB1.iloc[i,0]
    g = geocoder.arcgis('{},Singapore'.format(SGTOWN))
    lati.append(g.latlng[0])
    longi.append(g.latlng[1])

df_HDB1['Latitude'] = lati
df_HDB1['Longitude'] = longi

df_HDB1
```

Out[6]:

	SGTOWN	POP	Latitude	Longitude
0	Ang Mo Kio	141600	1.37161	103.84546
1	Bedok	191300	1.32425	103.95297
2	Bishan	62100	1.35079	103.85110
3	Bukit Batok	115200	1.34952	103.75277
4	Bukit Merah	144300	1.28417	103.82306
5	Bukit Panjang	120100	1.37877	103.76977
6	Bukit Timah	8300	1.34041	103.77221
7	Central Area	27300	1.28883	103.84625
8	Choa Chu Kang	168500	1.38616	103.74618



Now that the the basic population and coordinate dataset was ready, the venue information could now be sourced as the next step.

E. FOURSQUARE API TO GET VENUES

Foursquare API is an excellent source of information which is relatively current with latest available data. Using the Foursquare API and my credentials, the nearby venues and their details for all the towns of Singapore was obtained and read into the dataFrame df_SGVENUE

For the purpose of our project radius was limited to 1 Kilometre and venues to 100.

The below screenshots show the first 10 records read into the data frame. There are 180 unique venues across the island, and a snapshot of the unique type of venues is as below. While analysing this data, it was realised that the feature information required for the project is also part of the data available in Foursquare venues - River, Waterfront.

```
df_SGVENUE.head(10)
```

Out[62]:

	SGTOWN	NAME	TYPE
0	Bishan	Tori-Q	Japanese Restaurant
1	Bishan	Starbucks	Coffee Shop
2	Bishan	Dian Xiao Er 店小二	Chinese Restaurant
3	Bishan	Gymm Boxx XL	Gym
4	Bishan	Bishan Cafeteria (Eating House)	Food Court
5	Bishan	食香阁 Shi Xiang Ge La Mian . Shaved Noodle	Shaanxi Restaurant
6	Bishan	Din Tai Fung 鼎泰豐	Dumpling Restaurant
7	Bishan	Popular Bookstore	Bookstore
8	Bishan	Bishan Sports Hall	Stadium
9	Bishan	Pet Lovers Centre	Pet Store

```
In [16]: df_SGVENUE['TYPE'].unique()
```

```
Out[16]: array(['Japanese Restaurant', 'Coffee Shop', 'Chinese Restaurant', 'Gym',
                'Food Court', 'Shaanxi Restaurant', 'Dumpling Restaurant',
                'Bookstore', 'Stadium', 'Pet Store', 'Pool', 'Electronics Store',
                'Ice Cream Shop', 'Supermarket', 'Shopping Mall',
                'Asian Restaurant', 'Cosmetics Shop', 'Pharmacy',
                'Thai Restaurant', 'Seafood Restaurant', 'Bubble Tea Shop', 'Park',
                'Fried Chicken Joint', 'Café', 'Multiplex', 'Gastropub',
                'Italian Restaurant', 'Department Store',
                'Eastern European Restaurant', 'Bus Station', 'Basketball Court',
                'Trail', 'Noodle House', 'Steakhouse', 'Bakery', 'Bus Line',
                'Korean Restaurant', 'Music Venue', 'Bistro', 'Indian Restaurant',
                'BBQ Joint', 'Nature Preserve', 'Escape Room', 'Dessert Shop',
                'Beer Store', 'Dim Sum Restaurant', 'Diner', 'Sandwich Place',
                'Bar', 'Hainan Restaurant', 'Gas Station', 'Grocery Store',
                'Massage Studio', 'Fast Food Restaurant', 'Australian Restaurant',
                'Spanish Restaurant', 'Gaming Cafe', 'Pizza Place', 'Soup Place',
                'College Cafeteria', 'Convenience Store', 'Resort', 'Cafeteria',
                'Gym / Fitness Center', 'Historic Site', 'Snack Place',
                'Bike Trail', 'Waterfront', 'Cocktail Bar', 'Shopping Plaza',
                'Vegetarian / Vegan Restaurant', 'Hotel', 'River', 'Nightclub',
                'Wine Shop', 'Hotel Bar', 'Brewery', 'Hotpot Restaurant',
                'Mexican Restaurant', 'Yoga Studio', 'Lounge', 'Art Gallery',
                'Wine Bar', 'Buffet', 'Sake Bar', 'Hostel', 'Whisky Bar',
                'Pedestrian Plaza', 'Miscellaneous Shop', 'Canal',
                'French Restaurant', 'Persian Restaurant', 'Restaurant',
                'Monument / Landmark', 'History Museum', 'Comfort Food Restaurant',
                'Beer Garden', 'Salad Place', 'Bridge', 'Spa',
                'Martial Arts School', 'Event Space', 'Concert Hall',
                'English Restaurant', 'Buddhist Temple', 'Video Game Store',
                'Arts & Crafts Store', 'Chinese Breakfast Place', 'Garden', 'Pub',
```

F. ANALYSIS OF VENUES TO SHORTLIST

The venue data was then analysed based on our initial 3 requirements, to finalise the most appropriate SG Town where Robin can choose to stay. The data was shortlisted to include only the specific venues - Gym, Waterfront, Shopping Mall, Wine & Whisky bar.

In [19]:
```
SGPIV = pd.pivot_table(df_SGVENUE,index='SGTOWN',columns='TYPE',aggfunc=np.size,fill_value=0)
SGPIV.columns = SGPIV.columns.droplevel(0)
SGPIV
```

Out[19]:

TYPE	ATM	Accessories Store	American Restaurant	Art Gallery	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Australian Restaurant	Auto Garage	BBQ Joint	...	Trail	Train Station
SGTOWN													
Bishan	0	0	0	0	0	3	0	0	0	0	...	1	0
Bukit Timah	0	0	0	0	0	2	0	1	0	1	...	1	0
Central Area	0	0	0	2	0	0	0	0	0	1	...	0	0
Clementi	0	0	0	0	1	4	0	0	0	1	...	1	0
Geylang	0	0	0	0	0	4	1	0	0	4	...	1	0
Jurong East	0	2	1	0	0	0	0	0	0	0	...	1	0

A custom weightage was applied to the different types of venues under consideration to arrive at a consolidated score across the towns shortlisted. The resultant data was then sorted by descending order of the score to get the Town with highest score displayed at the top of the data set.

In [20]:
```
# Selecting only the choice of venues for selection criteria & applying weightage
SGPIV1 = SGPIV.loc[:,['Gym','Waterfront','Shopping Mall','Wine Bar','Whisky Bar']]

SGPIV1['Score'] = SGPIV1['Gym']*0.2+ \
                  SGPIV1['Waterfront']*0.2+ \
                  SGPIV1['Shopping Mall']*0.1+ \
                  SGPIV1['Wine Bar']*0.2+ \
                  SGPIV1['Whisky Bar']*0.1

# Get the dataframe with Score in descending order
SGPIV1 = SGPIV1.sort_values(by='Score',ascending=False)
SGPIV1
```

Out[20]:

TYPE	Gym	Waterfront	Shopping Mall	Wine Bar	Whisky Bar	Score
SGTOWN						
Central Area	2	1	1	3	1	1.4
Jurong East	1	0	4	0	0	0.6
Kallang/Whampoa	3	0	0	0	0	0.6
Clementi	2	0	1	0	0	0.5
Marine Parade	0	1	1	1	0	0.5
Serangoon	1	0	1	1	0	0.5

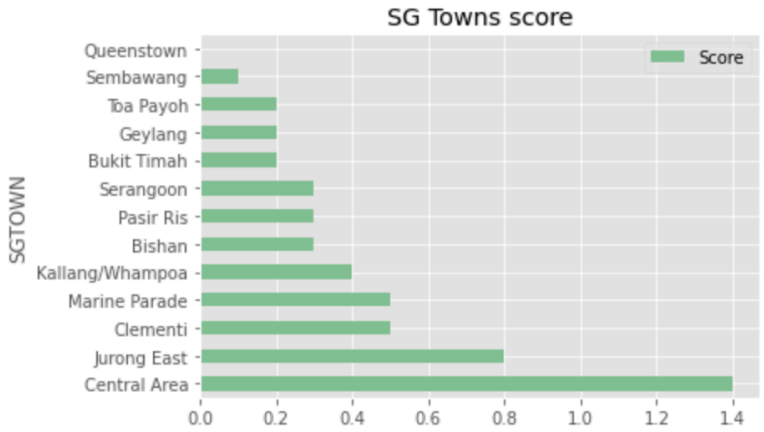
SECTION 4 : RESULTS

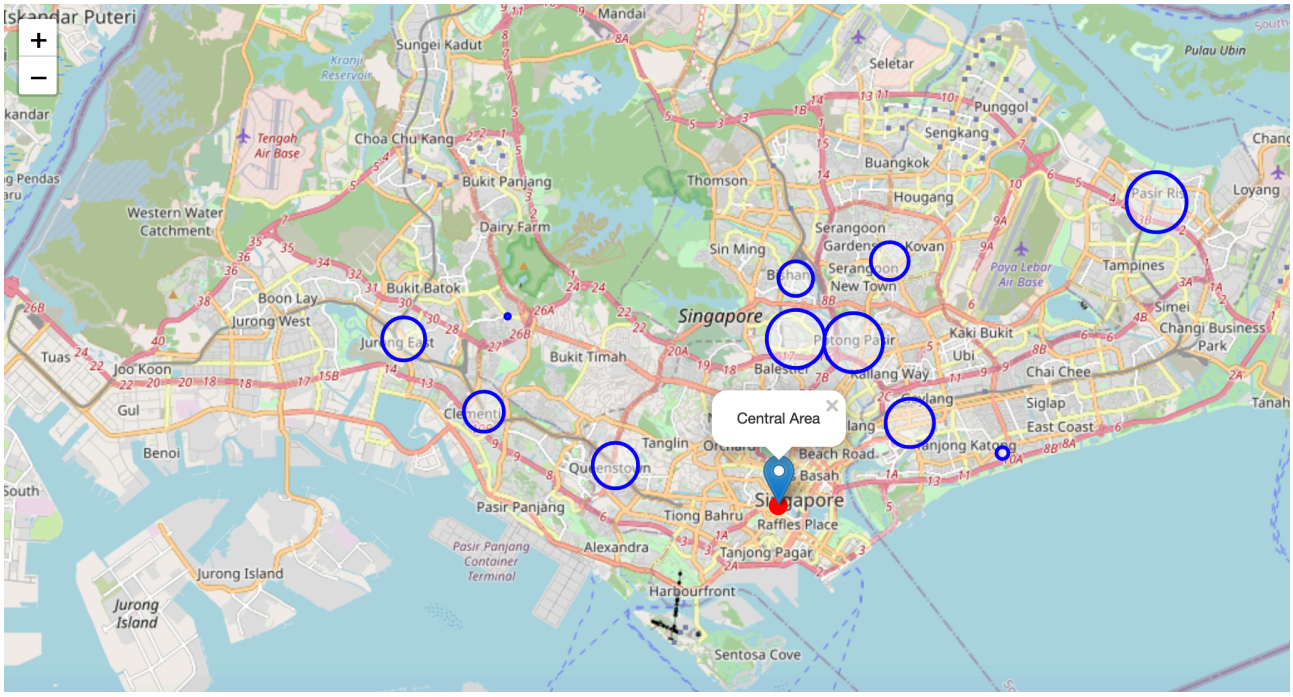
G. DISPLAY RESULTS IN GRAPH & MAP

The scores were calculated and results sorted by descending order to find the highest scoring town of Singapore based on Robin's requirements.

It was graphically displayed as below to have a better visual representation for ease of viewing and understanding. It was seen that the Central area of Singapore had the highest score and also the expected population falling within Robin's criteria as stated at the beginning of this project

A map of Singapore was rendered with the shortlisted SG Towns indicated by their population size and the selected township with a popup. As the current analysis pointed to Central area town for Robin to live in, it is displayed in red and a popup with the town name, on clicking the marker.





SECTION 5 : DISCUSSION

- The available Singapore Government data was very relevant for the problem with the available population across the Singapore town areas. But being a fast growing city, this data needs to be refreshed to have the latest population which could affect the shortlisted town areas, and potentially change the highest scoring town as well.

- A relatively shorter subset of the venue was taken with 5 unique types However there are a lot of relevant types that can be added to this mix and a much different ordering could result if the entire list is analysed and chosen.

- The weightage assigned to different types is another key criteria in tweaking the results of the paper, that can vary with different weightage applied.

- The resultant data set could be modified to include the venue details as a table and also as markers or popups in the folium map to have a holistic view of the requirement being met.

- This approach can be applied to any city / country in the world to choose a place where people want to live based on their criteria. It can be a starting point to which various other criteria can be easily added like the cost of homes, HDB housing Vs Condominiums, etc.

- The folium map rendered has markers whose radius depicts the population size across the shortlisted towns and the final chosen Town by a popup. This can further be refined by using a choropleth map to portray population across for a different visualisation approach.

SECTION 6 : CONCLUSION

In this Capstone project we have established an approach for a prospective homebuyer or a tenant who can decide to choose a place to live in the island of Singapore using Data Science tools and techniques.

This approach caters to their specifications or conditions utilising the data available from Government and public agencies. This would be an useful tool for anyone who is trying to shortlist localities or suburbs to live, based on their wants and desires and can be easily upgraded with additional criteria and tweaked to the requirement.

This Capstone project paper would therefore help Robin or any similar person to navigate that first essential step towards finalising the place to live.