TRAINING FOR THAI

Training cluster models to determine the best location to open a Thai restaurant in Sydney, Australia

Data Science
Capstone Project

Coursera Data Science Capstone Project

By **Zai Hassan**

26th April 2020

The accompanying Jupyter notebook for this report can be found on GitHub:

Nippet > Data Science Capstone Project IPYNB

Contents

Coursera Data Science Capstone Project 1
Introduction 2
Background 2
Problem Statement 2
Scope 2
Data 3
Train Station Entrance Data 3
Foursquare Data 3
Statistical Area 1 Socio-Economic Indexes for Australia (SEIFA) 4
Statistical Area 1 Shapefiles 4
Methodology Walk-through 6
Summary 6
Data Retrieval, Cleaning and Wrangling 6
Density-Based Spatial Clustering of Applications with Noise 11
DBSCAN Evaluation 12
K-Means Clustering Approach 13
Results 13
Data Validation 17
Discussion and Recommendations 18
Conclusion 18
Future Improvements 19

Introduction

Background

Sydney has a very competitive food and restaurant scene, locals are spoilt for choice when deciding to eat out. One of the key drivers fuelling this industry in Sydney is the easy access to fresh and affordable produce. This results in healthy competition and of course a great selection of dining venues for the consumer. To many residents, tourists or visitors it is instantly observable that Thai cuisine is extremely popular in Sydney. This report will look at the restaurant venues around Sydney, and in particular those of Thai cuisine.

Problem Statement

If someone were to consider opening a new Thai restaurant, where in Sydney would be the most ideal location? Where is the best location that has access to decent public transport, high population density, but perhaps is not already over-saturated with Thai restaurants. This is a question restaurateurs, entrepreneurs or like-minded business people might ask, and thus forms the target audience of this exercise.

Scope

To better refine the scope, this project will look only at areas that have a train station entrance in proximity. The justification being that there is still a large section of the community that do not drive (or have access to a car), taxis are expensive in Sydney and considering accessibility to trains means increasing the catchment area for potential patrons.

The data sourcing will mainly come via the Foursquare API, and will look at these key attributes:

- Competing Thai restaurants in the immediate vicinity
- Clusters of restaurants and food services in these areas

Enriching this with population density data from the Australian Bureau of Statistics will enable us to determine the numbers of restaurants per capita, in the hope of finding the answer to the problem statement.

Data

Train Station Entrance Data

This exercise will use train station entrance data as the fundamental location for retrieving venue data via Foursquare. This will mean we are only targeting locations that have an immediate access to the train network in Sydney.

The latitude and longitude of train station entrances is publicly available on the Transport New South Wales open data website. Transport NSW is the Government Agency that runs the train network in Sydney. Sydney is in the State of New South Wales in Australia. A free account is required to download this data, and upon downloading the dataset the following features are available:

	Train_Station	Street_Name	Street_Type	Entrance_Type	LAT	LONG	Exit_Number
0	Aberdeen	Macqueen	St	Ramp	-32.166886	150.891957	NaN
1	Aberdeen	Macqueen	St	Stairs	-32.166900	150.891975	NaN
2	Adamstown	Park	Ave	Path	-32.933706	151.720452	NaN
3	Adamstown	Park	Ave	Path	-32.933827	151.720236	NaN
4	Adamstown	St James	Rd	Stairs	-32.933414	151.720363	NaN

From the above we can see that the latitude and longitude data is available for each train station. As part of the shapefile processing we will need to filter this complete dataset for Sydney train stations only. The complete dataset is for all stations in the State of New South Wales which is an exceptionally large area. We will also look to retrieve data for only one entrance per train station (there are usually at least 2).

Foursquare Data

As per the capstone project requirements, we will be using the Foursquare API to retrieve venue data around the vicinity of train stations. We will look at a radius of 650 metres and we will particularly focus on Thai restaurants already in the area.

Running a function to call the Foursquare API using the co-ordinate information in our train station, the following features are available:

	Station	Station Latitude	Station Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Aberdeen	-32.166886	150.891957	Aberdeen Station	-32.166837	150.891829	Train Station
1	Aberdeen	-32.166886	150.891957	Croft's Pies	-32.167318	150.891716	Bakery
2	Aberdeen	-32.166900	150.891975	Aberdeen Station	-32.166837	150.891829	Train Station
3	Aberdeen	-32.166900	150.891975	Croft's Pies	-32.167318	150.891716	Bakery
4	Adamstown	-32.933706	151.720452	Adamstown Train Gates	-32.933192	151.720401	Train Station
5	Adamstown	-32.933706	151.720452	The Gates Hotel	-32.933370	151.721400	Beer Garden
6	Adamstown	-32.933706	151.720452	Adamstown Station	-32.933759	151.720079	Train Station
7	Adamstown	-32.933706	151.720452	The Nags Head Hotel	-32.935290	151.725370	Pub
8	Adamstown	-32.933827	151.720236	The Gates Hotel	-32.933370	151.721400	Beer Garden
9	Adamstown	-32.933827	151.720236	Adamstown Train Gates	-32.933192	151.720401	Train Station

Statistical Area 1 Socio-Economic Indexes for Australia (SEIFA)

The Australian Bureau of Statistics (ABS) is a Federal Government Agency in Australia that collects all sorts of population statistics through a 5-yearly census. This data is de-identified, but it is very useful to statisticians and data scientists as data on attributes like population density, remoteness and socio-economic indexes are collected.

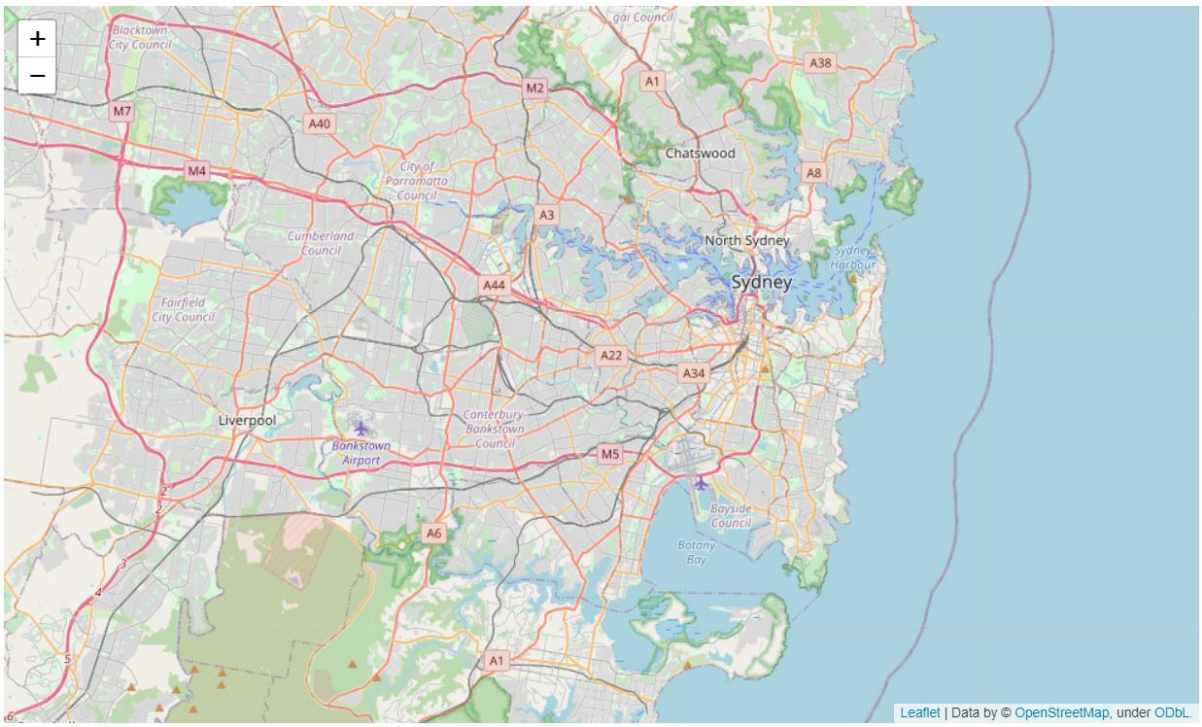
For this exercise we are interested in the population density, and also the socio-economic index of a statistical area. Statistical areas are an official region defined by the ABS. Statistical area 1 (referred to as SA1) is the smallest unit for release of census data. More information on SA1 can be found on the ABS website. Population density is normally collected as the 'usual resident population' of an area. The Socio-Economic Index for Areas (SEIFA) is a product that ranks areas in Australia according to relative socio-economic advantage and disadvantage. The measure looked at in this report is the Index of Relative Socio-Economic Advantage and Disadvantage (IRSAD).

The data file we are interested in is publicly available here as a zip file called Statistical Area Level 1, Indexes, SEIFA 2016. The Table 1 summary contains the appropriate dataset:

	Unnamed: 0	Unnamed: 1	Index of Relative Socio-economic Advantage and Disadvantage	Unnamed: 3	Index of Education and Occupation	Unnamed: 5	Unnamed: 6
0	2016 Statistical Area Level 1 (SA1) 7-Digit Code	2016 Statistical Area Level 1 (SA1) 11-Digit ...	IRSAD Score	IRSAD Decile	Occ Score	Occ Decile	Usual Resident Population
1	1100701	10102100701	972	4	984	5	256
2	1100702	10102100702	1044	7	1055	7	381
3	1100703	10102100703	962	4	1010	6	428
4	1100704	10102100704	970	4	1026	6	446
5	1100705	10102100705	936	3	979	5	402
6	1100706	10102100706	943	3	942	3	249
7	1100707	10102100707	1016	6	1047	7	359
8	1100708	10102100708	1046	7	1111	9	458
9	1100709	10102100709	1063	8	1048	7	380

Statistical Area 1 Shapefiles

Shapefiles for statistical area 1 (SA1) constructs are freely available on the ABS website. The particular file we will be using in this exercise is called Statistical Area Level 1 (SA1) ASGS Ed 2016 Digital Boundaries in ESRI Shapefile Format. The latest release is from 2016, since the ABS only performs the census every 5 years the next official release will be in 2021. As such these are the latest files, an example of the area we are interested in is shown in the Folium generated map below:



Using the shapefile downloaded from the ABS website, an external GIS tool called QGIS has been used to create a GeoJSON file of just Sydney SA1 regions. This tool was used because the shapefile is exceedingly large, and QGIS handles the visual manipulation of large vector layers better through a proper application GUI to ensure the right features are selected.

The shapefile contains data on statistical area 1 through to statistical area 4 (SA4). Australian Statistical Geography Standards are explained on the ABS website, essentially SA1s are contained within SA2s in a natural hierarchy and so on. Simple filtering was done on statistical area 4 (SA4) to search for the term 'Sydney'. This means we are only looking at Sydney SA1s, an example of what was finally stored in the GeoJSON is shown below (please note the SA4_NAME16 field):


```
{'features': {0: {'type': 'Feature',
   'properties': {'SA1_MAIN16': '11501129001',
    'SA1_7DIG16': '1129001',
    'SA2_MAIN16': '115011290',
    'SA2_5DIG16': '11290',
    'SA2_NAME16': 'Baulkham Hills (East)',
    'SA4_NAME16': 'Sydney - Baulkham Hills and Hawkesbury',
    'STE_NAME16': 'New South Wales',
    'AREASQKM16': 0.274},
   'geometry': {'type': 'MultiPolygon',
    'coordinates': [[[[150.99443028000007, -33.76461951599998],
       [150.9944598080001, -33.76449549199998],
       [150.99385609900003, -33.763934793999965],
       [150.9948060470001, -33.76350977499993],
       [150.99598561800008, -33.76466163999993],
       [150.99783850400001, -33.76710745199995],
       [150.99876465500006, -33.76805489199995],
       [150.99214238700006, -33.76975099099997],
       [150.99123623800006, -33.767203706999965],
       [150.990663626, -33.76716317299997],
       [150.9905512790001, -33.76607121099994],
       [150.99109848700004, -33.76603097399993],
       [150.99443028000007, -33.76461951599998]]]]}}}}
```

The vector geometry was also simplified slightly to reduce the file size and make it more manageable. Later on we will create buffers around each station to select all SA1's in the immediate area, but since these SA1 areas themselves are quite small there should be minimal impact in simplification process. Also, the Foursquare venue data is not dependent on the shapefiles themselves which again justifies the use of the simplification process.

Methodology Walk-through

Summary

At a high level the following steps were taken to retrieve, clean and wrangle the data, and finally apply the clustering algorithms:

1. Review and clean train station entrance dataset
2. Filter the dataset to only contain one entrance per train station
3. Overlaid the SA1 shapefile onto the train station dataset for review
4. Created a buffer zone around each train station and selected the neighbouring SA1 areas
5. Consumed the SA1 SEIFA information and generated mean values on key features
6. Retrieved the Foursquare venue data
7. Transformed the venue data to look at restaurants and provide counts for each train station
8. Merged the train station, SEIFA and Foursquare data to create a single dataset
9. Fit the data to a DBSCAN model and visualised results
10. Fit the data to a K-Means model and visualised results
11. Identified the best cluster and filtered further to produce a final recommendation

Data Retrieval, Cleaning and Wrangling

The train station data set is first read in, cleaned and then transformed to produce a unique ID for each train station entrance:

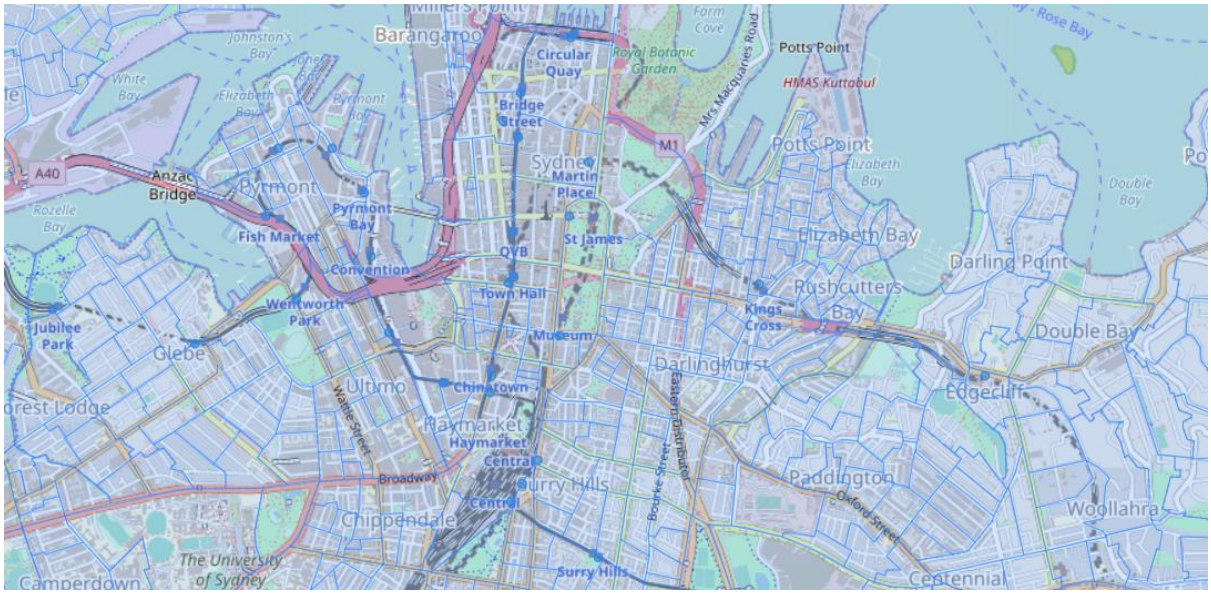
	Train_Station_ID	Train_Station	LAT	LONG	Row Num
1	Aberdeen 1	Aberdeen	-32.166900	150.891975	1
0	Aberdeen 2	Aberdeen	-32.166886	150.891957	2
3	Adamstown 1	Adamstown	-32.933827	151.720236	1
2	Adamstown 2	Adamstown	-32.933706	151.720452	2
4	Adamstown 3	Adamstown	-32.933414	151.720363	3
5	Adamstown 4	Adamstown	-32.933402	151.720347	4
7	Albion Park 1	Albion Park	-34.563507	150.799189	1
6	Albion Park 2	Albion Park	-34.563409	150.799390	2

Using Folium the plots were marked so that the they could be represented in a geographical space. Since most stations in Sydney have more than one entrance it was necessary to understand whether the dataset could be trimmed to only look at one entrance per station:

As was evident in the visualisation, there were clusters of entrances that were not providing additional value - they were at times less than 100 metres of each other. The following filtering was applied to keep only one entrance per station:

```
df_stations_clean = df_stations_clean[df_stations_clean['Row Num'] == 1]
```

For additional validation we overlay the SA1 shapefile onto the filtered train station entrance data set to confirm the above process is acceptable:

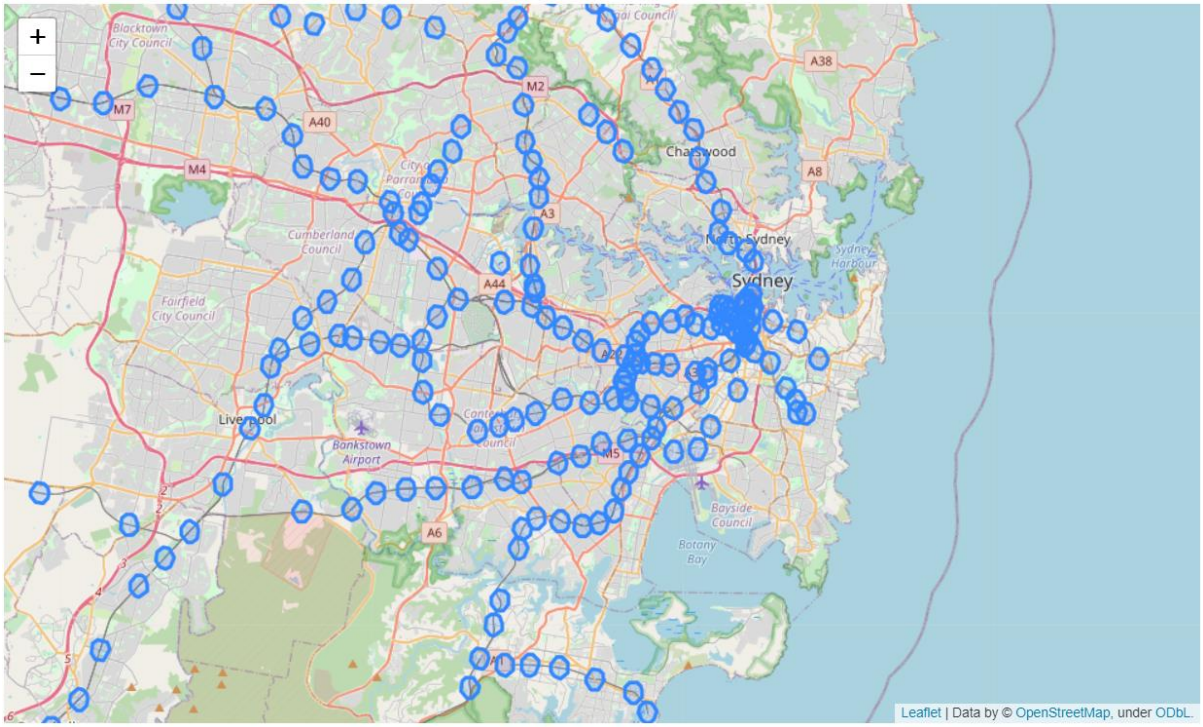


In reviewing the above map, I formed the conclusion that one entrance per station is acceptable however there are many exits on the edge of polygons. To compensate for this a buffer will be created around each exit to grab all the SA1 areas within an appropriate vicinity.

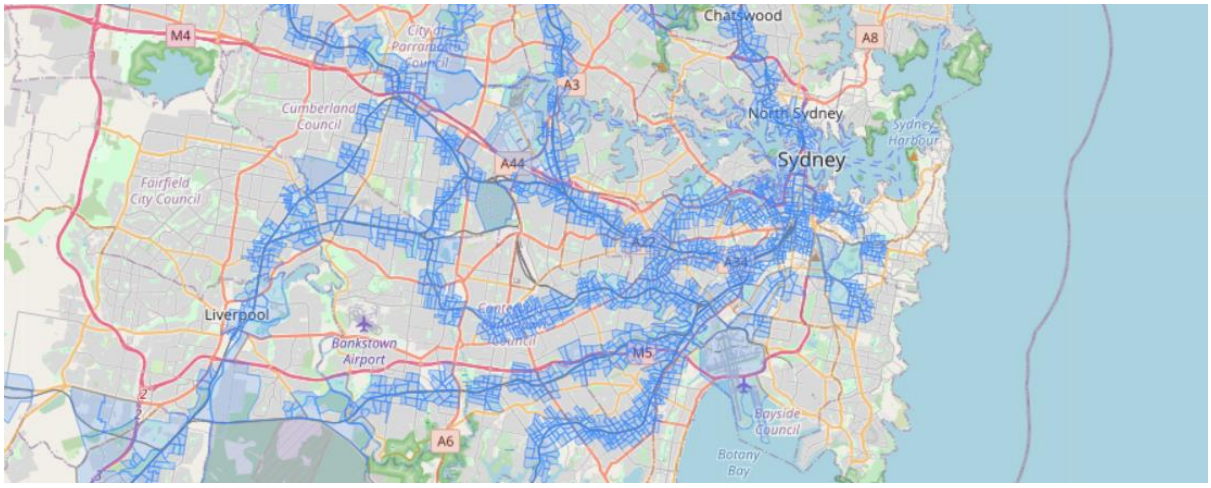
The following code was used to implement this using geopandas and a GeoDataFrame:

```
# We will now create the buffer around each station

geo_stations['Buffer'] = geo_stations.geometry.buffer(0.005, resolution=16)
geo_stations.set_geometry('Buffer', inplace=True)
```

The resulting visualisation aptly demonstrates the intent of this process:



To wrap this up, a geospatial join is done between the buffer zones and the SA1 shapefile to obtain all the intersections. As there are instances of the same SA1 selected more than once due to the proximity of stations, only the first instance of an SA1 is selected for the purpose of visualising what will be modelled on a map:



**Note the image above is not representative of all the SA1s being analysed in the Sydney region. It has been cropped for documentation purposes.*

The next part of the data wrangling process is to combine the SEIFA data into the train station/SA1 dataset. The SEIFA data is read in and the dual header row issue is resolved, then we clean up the dataframe so the data can be joined easily:

	SA1_MAIN16	IRSAD Score	IRSAD Decile	Occ Score	Occ Decile	Usual Resident Population
0	10102100701	972	4	984	5	256
1	10102100702	1044	7	1055	7	381
2	10102100703	962	4	1010	6	428
3	10102100704	970	4	1026	6	446
4	10102100705	936	3	979	5	402

For the purposes of this exercise we have used the decile columns, rather than the raw scores. The 'Occ' columns are the Indexes for Education and Occupation which have been left in for information purposes. Once the above data is merged with the GeoDataFrame we obtain a dataset similar to below (noting the image has been cropped due to the number of columns):

I16	SA2_5DIG16	SA2_NAME16	SA4_NAME16	STE_NAME16	AREASQKM16	geometry	index_right	Train_Station_ID	Train_Station	Row Num	IRSAD Score	IRSAD Decile
291	11291	Baulkham Hills (West) - Bella Vista	Sydney - Baulkham Hills and Hawkesbury	New South Wales	0.1455	MULTIPOLYGON (((150.96253 -33.73971, 150.96244...	675	Norwest 1	Norwest	1	1145	10
291	11291	Baulkham Hills (West) - Bella Vista	Sydney - Baulkham Hills and Hawkesbury	New South Wales	0.1099	MULTIPOLYGON (((150.97105 -33.73365, 150.97243...	675	Norwest 1	Norwest	1	1123	9
291	11291	Baulkham Hills (West) - Bella Vista	Sydney - Baulkham Hills and Hawkesbury	New South Wales	0.0993	MULTIPOLYGON (((150.96453 -33.73557, 150.96683...	675	Norwest 1	Norwest	1	1081	8
291	11291	Baulkham Hills (West) - Bella Vista	Sydney - Baulkham Hills and Hawkesbury	New South Wales	1.9525	MULTIPOLYGON (((150.95836 -33.73092, 150.96034...	675	Norwest 1	Norwest	1	1122	9
291	11291	Baulkham Hills (West) - Bella Vista	Sydney - Baulkham Hills and Hawkesbury	New South Wales	0.0572	MULTIPOLYGON (((150.95837 -33.73786, 150.95838	675	Norwest 1	Norwest	1	1098	9

We end up with multiple rows for the same train station because we have selected the immediate SA1 areas within proximity. Further data validation is done to check for data types and unique values, as well as cleaning up '-' values present in the original dataset:

```
# Replace the '-' with NaN values in place

gp_SA1_SEIFA['IRSAD Decile'].replace(to_replace='-', value=np.nan, inplace=True)
```

At this step in the process is we want to derive the mean SEIFA values for each train station:

```
# Now we create a dataframe looking at the mean values by Train station
# Columns are re-ordered to make it easier to read
# These columns are mean values for all the SA1 areas around each station

df_SA1_SEIFA_mean = pd.DataFrame(gp_SA1_SEIFA.groupby('Train_Station').mean())

cols = df_SA1_SEIFA_mean.columns.tolist()
cols = cols[4:]
df_SA1_SEIFA_mean = df_SA1_SEIFA_mean[cols]
df_SA1_SEIFA_mean = df_SA1_SEIFA_mean.round(1)
df_SA1_SEIFA_mean.reset_index(inplace=True)

df_SA1_SEIFA_mean.head(10)
```

Output:

	Train_Station	IRSAD Decile	Occ Decile	Usual Resident Population
0	Allawah	5.7	6.6	520.5
1	Arlington LR	8.3	8.9	423.2
2	Arncliffe	5.6	6.0	579.2
3	Artarmon	9.2	9.4	441.1
4	Ashfield	6.2	7.8	451.8
5	Asquith	8.7	8.7	477.5
6	Auburn	2.2	2.6	540.2
7	Banksia	5.5	5.1	390.1
8	Bankstown	2.4	4.4	458.4
9	Bardwell Park	7.6	7.5	409.1

The next step in the process is to run the Foursquare API function call to retrieve all the venues against the train station dataset. Once the data is retrieved we perform some filtering to create dataframes for Thai restaurants and also for all restaurants around each train station. This enables us to generate counts of restaurants for each station:

```
# We now create dataframes based on whether the venue category contains restaurant or Thai.
# We can then use this to get counts of all restaurants and counts of Thai restaurants around each Train Station

df_restaurant = df_venues[df_venues['Venue Category'].str.contains('Restaurant')]
df_thai = df_venues[df_venues['Venue Category'].str.contains('Thai')]

# Generate counts of Thai restaurants around each Train Station

df_thai_count = df_thai[['Station', 'Venue Category']].groupby(['Station']).agg('count')
df_thai_count.reset_index(inplace=True)
df_thai_count.rename(columns={'Station':'Train_Station', 'Venue Category':'Thai Count'}, inplace=True)
df_thai_count.head(10)
```

	Train_Station	Thai Count
0	Arncliffe	1
1	Artarmon	2
2	Ashfield	1
3	Auburn	1
4	Bella Vista	2
5	Berala	1
6	Bexley North	1
7	Blackheath	1
8	Blacktown	1
9	Blaxland	1

	Train_Station	Total Restaurants
0	Arlington LR	4
1	Arncliffe	3
2	Artarmon	7
3	Ashfield	17
4	Asquith	1
5	Auburn	11
6	Banksia	2
7	Bankstown	19
8	Beecroft	3
9	Bella Vista	4

Finally all this information is merged, and a new feature is calculated based on the number of restaurants and the mean usual resident population:

	Train_Station	LAT	LONG	IRSAD Decile	Occ Decile	Usual Resident Population	Thai Count	Total Restaurants	Restaurants Per Capita
0	Allawah	-33.969969	151.114945	5.7	6.6	520.5	0.0	0.0	0.000000
1	Arlington LR	-33.902050	151.138043	8.3	8.9	423.2	0.0	4.0	0.009452
2	Arncliffe	-33.936963	151.147423	5.6	6.0	579.2	1.0	3.0	0.005180
3	Artarmon	-33.808759	151.184720	9.2	9.4	441.1	2.0	7.0	0.015869
4	Ashfield	-33.887892	151.125797	6.2	7.8	451.8	1.0	17.0	0.037627

We end up with a dataset with which to perform the clustering algorithms over. In the dataset there are 248 stations in Sydney captured.

Density-Based Spatial Clustering of Applications with Noise

Since there is a geospatial element to the clustering of stations, my first guess was to try a DBSCAN model approach. This model is really unique in that it is able to identify noise and outliers. The model was created and the data fit using the following features:

- Latitude
- Longitude
- IRSAD Decile
- Restaurants Per Capita

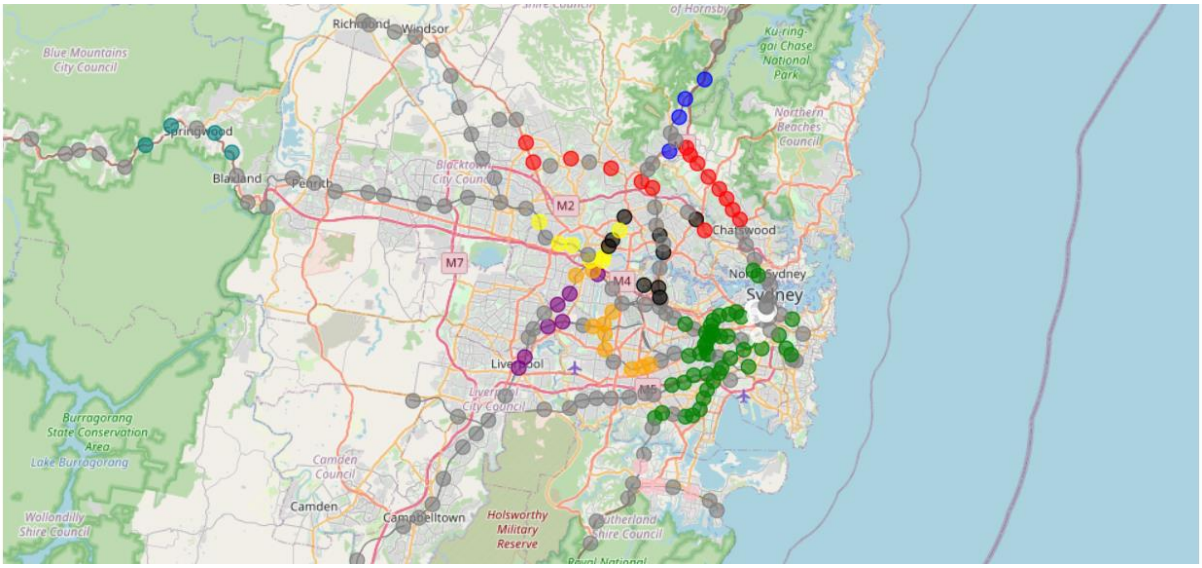
The number of clusters was then checked and mapped onto a folium chart for better analysis. The following epsilon values were tested along with minimum samples parameter:

Epsilon	0.20	0.25	0.30	0.4
Minimum values	6	6	5	4

DBSCAN Evaluation

The output from the last run is shown below, utilising an epsilon of 0.4 and a minimum samples parameter of just 4:

	Train_Station	LAT	LONG	IRSAD Decile	Occ Decile	Usual Resident Population	Thai Count	Total Restaurants	Restaurants Per Capita	Clus_Db
0	Allawah	-33.969969	151.114945	5.7	6.6	520.5	0.0	0.0	0.000000	0
1	Arlington LR	-33.902050	151.138043	8.3	8.9	423.2	0.0	4.0	0.009452	0
2	Arncliffe	-33.936963	151.147423	5.6	6.0	579.2	1.0	3.0	0.005180	0
3	Artarmon	-33.808759	151.184720	9.2	9.4	441.1	2.0	7.0	0.015869	-1
4	Ashfield	-33.887892	151.125797	6.2	7.8	451.8	1.0	17.0	0.037627	-1
5	Asquith	-33.688691	151.108266	8.7	8.7	477.5	0.0	1.0	0.002094	1
6	Auburn	-33.849663	151.032919	2.2	2.6	540.2	1.0	11.0	0.020363	-1
7	Banksia	-33.944900	151.140849	5.5	5.1	390.1	0.0	2.0	0.005127	0
8	Bankstown	-33.918052	151.034110	2.4	4.4	458.4	0.0	19.0	0.041449	-1
9	Bardwell Park	-33.931378	151.125379	7.6	7.5	409.1	0.0	0.0	0.000000	0
10	Bargo	-34.291045	150.580008	4.7	3.7	443.3	0.0	0.0	0.000000	-1




```
# We group the dataset by cluster and aggregate based on mean to get an understanding of the dataset

df_dbscan_master.groupby('Clus_Db').mean().drop(columns=['LAT','LONG','Occ Decile'])
```

Clus_Db	IRSAD Decile	Usual Resident Population	Thai Count	Total Restaurants	Restaurants Per Capita
-1	6.144186	473.234109	0.930233	8.906977	0.018899
0	8.008889	472.624444	0.488889	2.400000	0.005056
1	8.725000	426.025000	0.000000	1.000000	0.002361
2	9.766667	493.973333	0.333333	1.866667	0.003738
3	2.790000	484.010000	0.200000	3.200000	0.006546
4	5.900000	590.771429	0.285714	2.714286	0.004634
5	1.657143	517.285714	0.000000	1.000000	0.001898
6	7.914286	542.285714	2.214286	14.214286	0.027349
7	8.188889	918.988889	0.555556	3.111111	0.004341
8	7.150000	443.050000	0.750000	3.250000	0.007304
9	7.950000	387.325000	0.000000	0.000000	0.000000

To interpret this data we want to identify clusters with relatively high IRSAD scores (areas with better socio-economic prospects and thus disposable income), with reasonably high restaurants per capita (signifying food/restaurant hubs or districts), but with low numbers of competing Thai restaurants. Clusters 6 (the heart of Sydney CBD), cluster 7 (around inner West and North Western Sydney) and cluster 8 (the Sutherland Shire) are probably the best candidates here, with cluster 7 having less Thai restaurant competition and greater population density on average. It is important to note alot of stations have been identified as outliers (cluster -1 and coloured grey).

Prior to commencing a process to determine the best epsilon, I made the realisation that the DBSCAN method is probably not sufficient for our purposes. Taking a step back and reviewing this clustering method, whilst it is great from a geospatial point of view, it is however considering lots of stations as outliers or noise. This is influenced in part due to the geospatial structure of the Sydney train network - ie its outward branching nature, and unfortunately does not fully support the goal of this exercise. As an alternative we will look at the K-means clustering method instead.

K-Means Clustering Approach

Since the DBSCAN approach did not seem to be adequate, the next machine learning algorithm worth trying is K-Means clustering. Having spent time running the model with 7 and 8 clusters, I found that 6 was more manageable to unpack and interpret. 6 clusters over 12 iterations gives us a decent spectrum for the features included as part of the below code snippet:

```
# Create a dataframe for use with the K-Means model
# Standardise the data so it can be fit to the model

df_kmeans_master = df_master.copy()
x = df_kmeans_master[['IRSAD Decile', 'Restaurants Per Capita', 'Usual Resident Population', 'Thai Count']]
clusterset = StandardScaler().fit_transform(x)
```

```
clusterNum = 6
k_means = KMeans(init = "k-means++", n_clusters = clusterNum, n_init = 12)
k_means.fit(clusterset)
labels = k_means.labels_
print(labels)
```

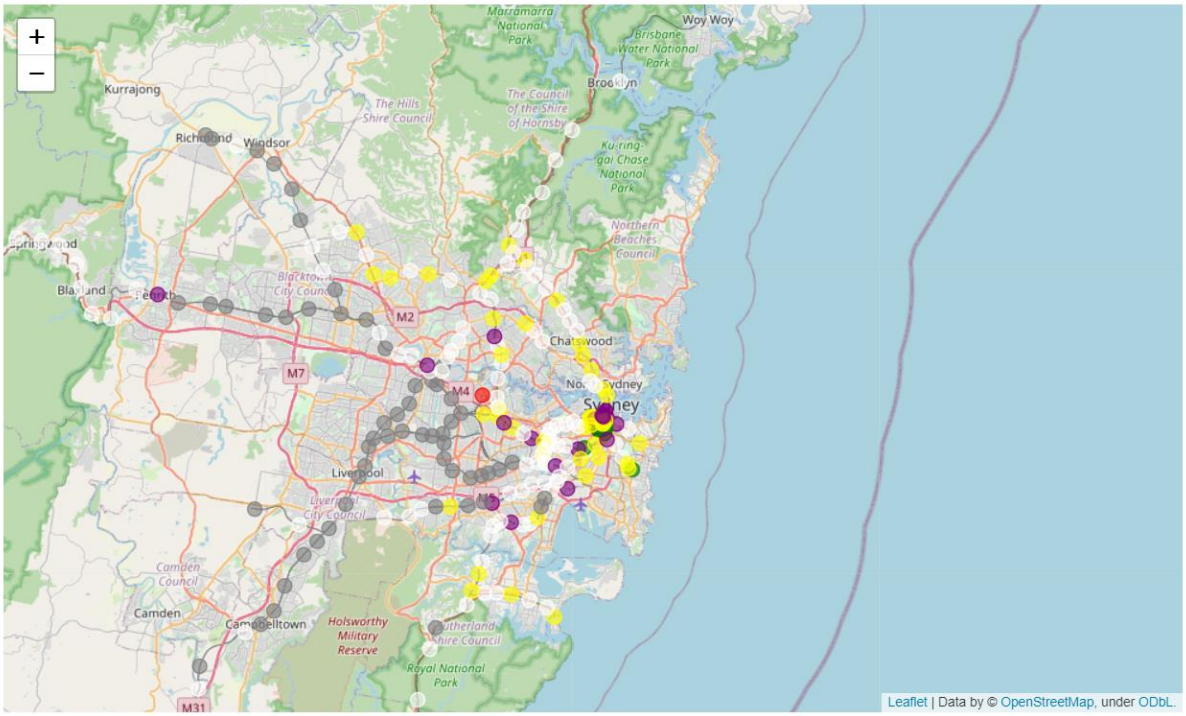
Results

To get an understanding of each cluster/classification produced by the above model we view the mean values for each of the core features:

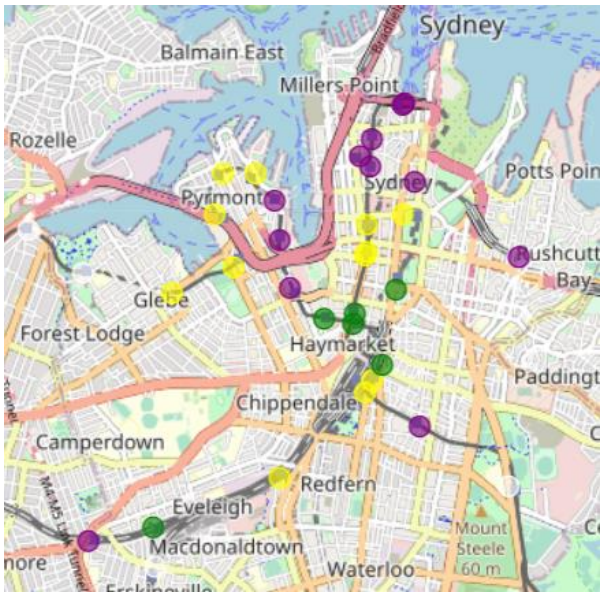
Class	Unnamed: 0	IRSAD Decile	Usual Resident Population	Thai Count	Restaurants Per Capita
0	116.952381	7.671429	419.509524	1.238095	0.054579
1	116.257576	3.250000	463.121212	0.151515	0.007370
2	112.142857	7.571429	635.800000	8.428571	0.049489
3	161.000000	9.000000	4263.000000	2.000000	0.002346
4	131.479167	8.120833	515.100000	1.562500	0.019954
5	126.114286	7.930476	479.960000	0.171429	0.003373

We then visualise the classifications utilising the Folium library and the following marker bins:

```
# We then visualise the classifications using folium along with the station plots

kmeans_map = folium.Map(location=[-33.893831, 151.125733], zoom_start=12)

df_km['marker_color'] = pd.cut(df_km['Class'], bins=6,
                               labels=['purple', 'grey', 'green', 'red', 'yellow', 'white'])

for index, row in df_km.iterrows():
    folium.CircleMarker([row['LAT'], row['LONG']],
                radius=6,
                weight=1,
                fill=True,
                fill_color=row['marker_color'],
                fill_opacity=0.6,
                popup=row['Train_Station'] + ' - cluster ' + str(row['Class']),
                color=row['marker_color']).add_to(kmeans_map)

kmeans_map
```

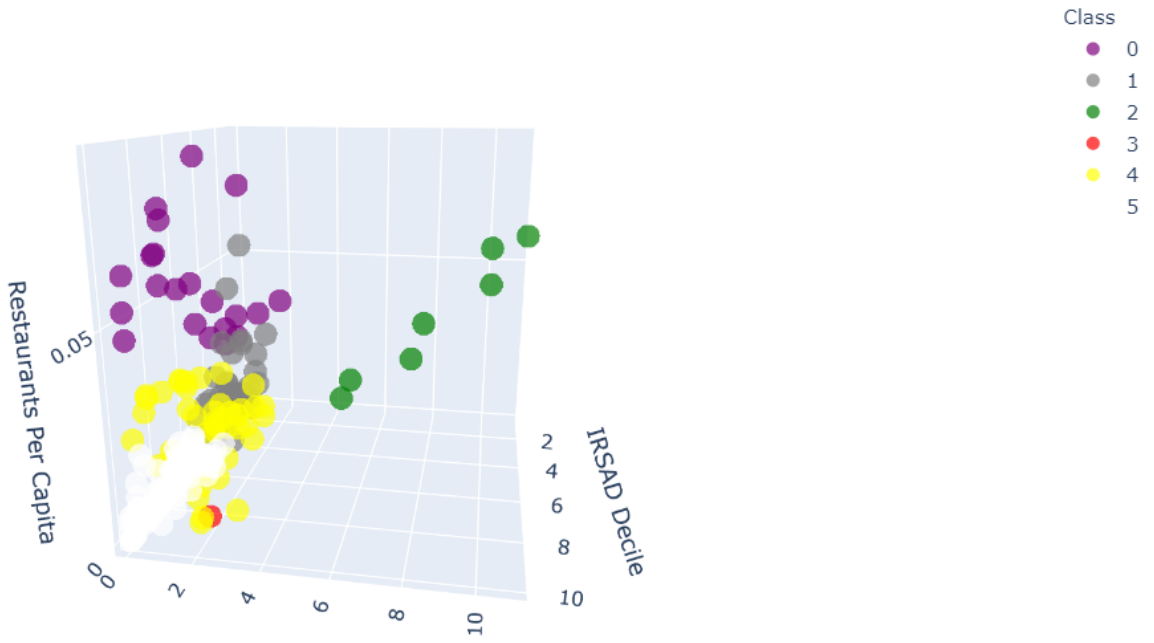


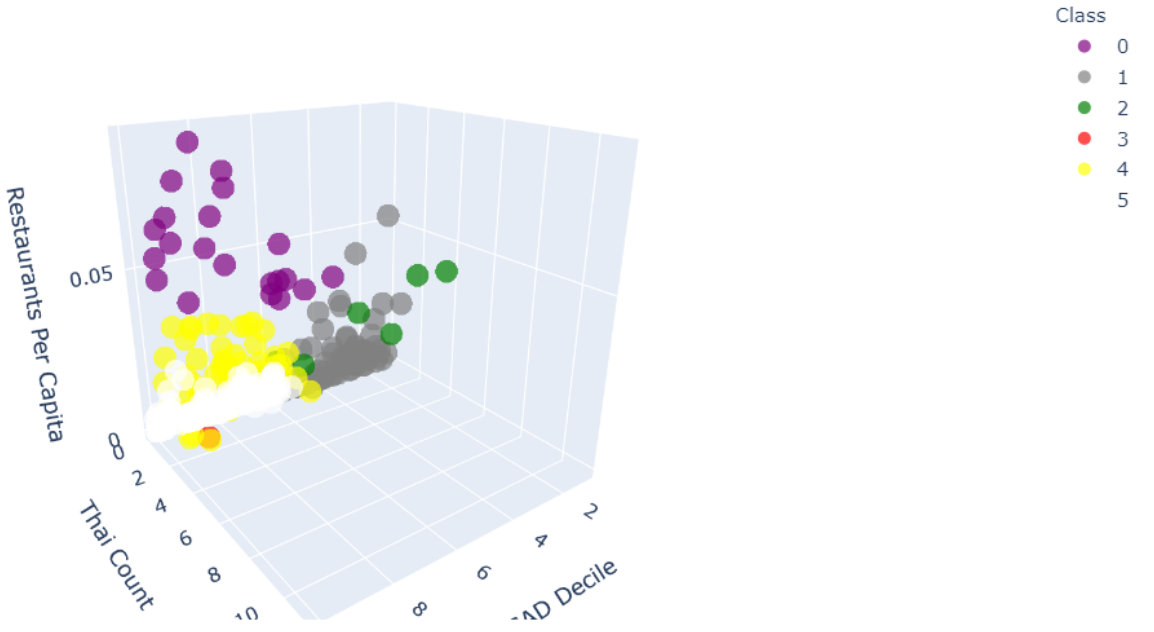
Zoomed in on CBD:



Cluster 0 (purple) and cluster 2 (green) really stand out here. Cluster 0 has the most restaurants per capita, and when we look at the map it strongly aligns with local intelligence. Places like Parramatta, Strathfield, Eastwood, Marrickville, Surry Hills are well known restaurant districts and this is reflected in the map. Cluster 2 is more densely populated, and it should be because it is right in the heart of Sydney's CBD, however on the downside there is a lot more Thai restaurant competition. Note that the grey cluster are not outliers as this is a K-means approach.

For even more comprehensive analysis, we utilise the plotly express library to produce an interactive 3d scatter plot of the dataframe:





Analysing the interactive scatter plot is extremely easy due to its interactive nature. We know we can discard cluster 1 straight away as it representative of the lower end of the IRSAD measure. Our justification is that we want to find areas with relative decent socio-economic indicators as these would have higher household budgets and propensity to eat out. Cluster 5 doesn't appear to have a lot of restaurants in their immediate area. Cluster 3 only comprises of one station which is concerning, so we will validate that in the data validation section. Cluster 4 appears to be the middle ground on the feature selection, but it just doesn't prove convincing against cluster 0 and cluster 2.

We will focus on cluster 0 because it is representative of lower Thai restaurant competition. We further refine cluster 0 by selecting those train stations where the IRSAD decile is greater than 6 and also sort to highlight the Thai restaurant count and the restaurants per capita:

	Train_Station	LAT	LONG	IRSAD Decile	Occ Decile	Usual Resident Population	Thai Count	Total Restaurants	Restaurants Per Capita	Class	marker_color
73	Exhibition Centre LR	-33.876788	151.199250	7.2	9.0	520.5	0	36	0.069164	0	purple
66	Eastwood	-33.790411	151.082636	7.7	8.8	470.2	0	29	0.061676	0	purple
244	Wynyard	-33.865834	151.206282	9.4	9.8	330.7	0	20	0.060478	0	purple
245	Wynyard LR	-33.866709	151.207262	9.5	9.8	399.3	0	21	0.052592	0	purple
134	Martin Place	-33.867962	151.211624	9.5	9.8	345.5	0	16	0.046310	0	purple
48	Circular Quay LR	-33.861570	151.210517	9.1	9.8	308.0	1	23	0.074675	0	purple
24	Bridge Street LR	-33.864464	151.207433	9.4	9.8	320.4	1	21	0.065543	0	purple
47	Circular Quay	-33.861466	151.210712	9.3	9.7	376.3	1	22	0.058464	0	purple
174	Pyrmont Bay LR	-33.869535	151.197708	8.3	9.2	451.6	1	25	0.055359	0	purple
4	Ashfield	-33.887892	151.125797	6.2	7.8	451.8	1	17	0.037627	0	purple
204	Surry Hills LR	-33.888329	151.212211	8.9	9.3	421.5	2	36	0.085409	0	purple

Reviewing the above we can see that we have isolated stations around the central business district of Sydney, as well other areas known for their food offerings. In general terms, rents in the heart of Sydney will be significantly higher than in areas further away. Since we don't have a dataset on commercial rents, for this particular exercise we will see if we can find suitable areas not within the immediate CBD area of Sydney. The above dataset is merged again to the SA1 dataset that still contains SA2 information. This allows us to filter out Sydney CBD stations:

```
# Finally we filter the dataset to remove SA2 areas that are within the CBD area

df_purple_SA2[(df_purple_SA2['SA2_NAME16'] != 'Sydney - Haymarket - The Rocks') &
              (df_purple_SA2['SA2_NAME16'] != 'Pyrmont - Ultimo') &
              (df_purple_SA2['SA2_NAME16'] != 'Surry Hills') &
              (df_purple_SA2['SA2_NAME16'] != 'Potts Point - Woolloomooloo')]
```

	Train_Station	LAT	LONG	IRSAD Decile	Occ Decile	Usual Resident Population	Thai Count	Total Restaurants	Restaurants Per Capita	Class	marker_color	SA2_NAME16	geometry
1	Eastwood	-33.790411	151.082636	7.7	8.8	470.2	0	29	0.061676	0	purple	Eastwood - Denistone	MULTIPOLYGON (((151.08304 -33.78698, 151.08307...
9	Ashfield	-33.887892	151.125797	6.2	7.8	451.8	1	17	0.037627	0	purple	NaN	None
12	Strathfield	-33.872208	151.094284	6.4	8.0	563.7	2	25	0.044350	0	purple	Strathfield	MULTIPOLYGON (((151.09310 -33.87154, 151.09220...
13	Marrickville	-33.913976	151.153316	6.4	7.5	410.1	2	16	0.039015	0	purple	Marrickville	MULTIPOLYGON (((151.15301 -33.91389, 151.15311...
15	Parramatta	-33.818238	151.005929	7.1	8.4	613.3	3	29	0.047285	0	purple	Parramatta - Rosehill	MULTIPOLYGON (((151.00370 -33.81802, 151.00339...
16	Newtown	-33.897948	151.178892	9.6	9.8	444.2	3	21	0.047276	0	purple	Newtown - Camperdown - Darlington	MULTIPOLYGON (((151.17830 -33.89828, 151.17871...

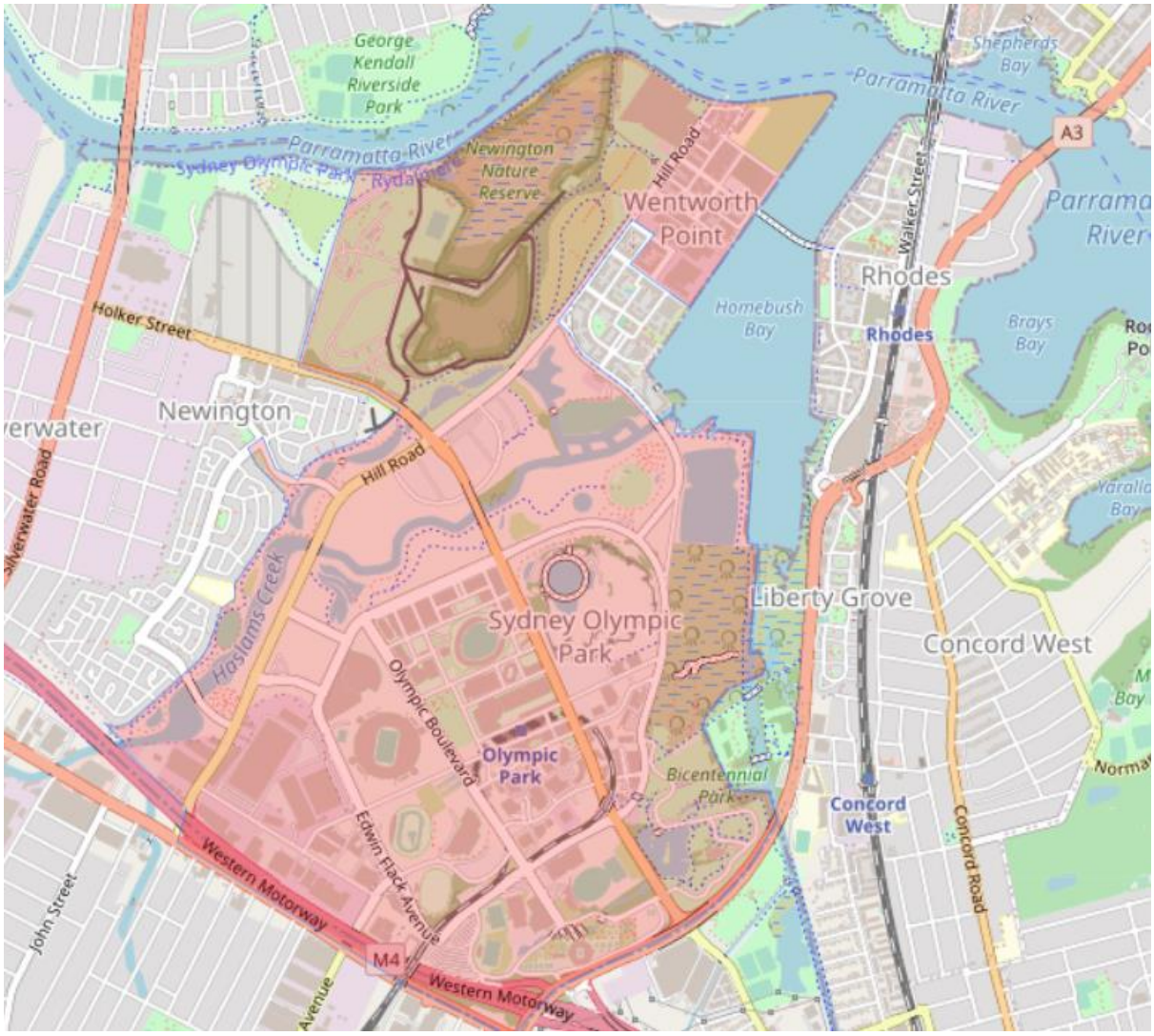
Data Validation

When looking at the scatter plot it is immediately apparent that cluster 3 (red) only has one station. Below we have a look at why this might be the case and provide a possible explanation.

```
# Which station is cluster 3

df_km[df_km['Class'] == 3]
```

	Unnamed: 0	Train_Station	LAT	LONG	IRSAD Decile	Occ Decile	Usual Resident Population	Thai Count	Total Restaurants	Restaurants Per Capita	Class	marker_color
161	161	Olympic Park	-33.846984	151.068272	9.0	10.0	4263.0	2	10	0.002346	3	red

We obtain the SA1 data and plot the polygon on Folium to visualise what is being captured in the area:



Olympic Park encompasses large area with hotels but also includes Wentworth Point which has some very high density housing (high-rise apartment blocks). On census night, if you are staying in a hotel then this is where you would record your location according to the census rules. Wentworth point in combination with the hotels around the Olympic park venue are probably the reason for this remarkably high usual resident population figure.

Discussion and Recommendations

Based on the final filtering we have Eastwood, Ashfield, Strathfield, Marrickville and Parramatta as prime candidates to answer the problem statement. These areas exhibit adequate socio-economic indicators and have high restaurants per capita which in theory attracts potential patrons to the area. They also don't have many competing Thai restaurants, which presents a viable opportunity to provide a new cuisine choice.

It is difficult to beat **Eastwood** as there are no competing Thai restaurants, yet the area is well known for its Asian food offerings. It has decent population density and the socio-economic indicators are in the upper range. Its key features are as follows:

- Mean IRSAD Decile for SA1s in the immediate area = 7.7
- Mean Index of Education and Occupation = 8.8
- Mean usual resident population = 470.2
- Thai restaurants within 650m of the station = 0
- Total restaurants = 29
- Restaurants Per Capita = 0.062

Based on the results of the clustering methods we have applied along with the integration of SEIFA data; Eastwood would be our recommendation candidate.

Conclusion

We have successfully used Foursquare venue data integrated with SEIFA information from the Australian Bureau of Statistics to create a solid dataset around Sydney train stations. After utilising two clustering approaches we have been able to identify **Eastwood** as the best candidate for opening up a new Thai restaurant based on features including the average IRSAD score for the area, its proximity to other restaurants and the lack of local Thai restaurant competition.

Caveats:

This is limited to the accuracy and currency of the Foursquare dataset – a key question is how current and complete is the venue data? It has been several years since the last census, the next one is due in 2021 so it could be worthwhile to re-run this exercise once the new SEIFA data is available.

Future Improvements

This is by no means a perfect exercise, areas where this exercise could be improved:

1. Consuming all restaurant venue data in Sydney (if possible), not just those in the 650 metre radius of train stations
2. Running DBSCAN over the complete restaurant dataset from point 1 would better identify restaurant hubs that would provide an attractive location
3. Integrating data on average commercial rents for restaurant sized locations
4. Rather than creating a buffer zone around each station, use a minimum distance between polygons formula