APPLIED DATA SCIENCE CAPSTONE PROJECT

Geospatial Analysis of Sydney and Melbourne

By: Nikko Dote

INTRODUCTION

For a long time, there has always been a heated debate about which Australian city is better, Sydney or Melbourne. So much so that the capital of this continent country is **not** either of the two, but is located between them (see the map) as a compromise and probably to avoid civil unrest. There has been no attempt to conduct a suburb segmentation of these 2 cities.

Business Problem

Although, I'm not brave enough to settle this dispute, but this project aims to give the reader an idea of the Sydney and Melbourne's landscape. As domestic migration is common in Australia, moving between states means taking a chance on what benefits you will be gaining or compromise you will be taking in terms of your lifestyle, work, necessities. Maybe this time, an informed choice is what these big-dreamers need.

Target Audience

This analysis is useful to people who are at the fork of making a decision to move to the biggest cities in Australia depending on their current need and interest. So which city is for you?

DESCRIPTION OF DATA

The data to get the suburbs are scraped but various webpages (wikipedia and namecensus). The suburbs are then grouped according to postal code to make it simpler and give a general feel of the dataset.

To get the Postal Code and Suburb Names we use these webpages:

Melbourne : https://en.m.wikipedia.org/wiki/List_of_Melbourne_suburbs

Sydney : https://namecensus.com/igapo/australia/postcodes/sydney-numeric.html

Once the Post Code and coordinates are gathered and the data is cleaned, the coordinates can be procured using geocoder module. Using the module, we can get the latitude and longitude of the specific postal code which will be used in the using the **Foursquare API.** Then the nearby venues are derived using the Foursquare API through explore method.The data

obtained using Explore method of Foursquare will indicate what kinds of amenities are around the suburb which can range from 'African Restaurants' to 'Zoo'. For example, a lot of suburbs in the Inner City likely have a lot of shopping malls, bars and cafes while remote suburbs may have more parks or gym than shopping malls. It would be interesting to see if some suburbs are center for certain facilities like which suburbs have more sports facilities than others, which ones are the business powerhouses and center for food diversity. In saying that, the data is limited to the foot traffic recorded by the company. Also, the venue category can be too specific like 'Beer Garden', 'Beer Bar', 'Pub' which are places for entertainment.

The number of these venues per suburb will be the feature of each subrub and are then counted using one-hot coding so each venue category represent a number (0 or 1 for yes or no). Then the average frequency is calculated for each postal code e.g. if there are only 5 venue categories for Suburb Sunbury then each category will get 0.2. The suburbs are then clustered and examined using K-means clustering according to the frequency of venue categories. It is a machine learning algorithm that will segment the observations into the pre-set number of clusters.

Clustering the suburbs using these data will tell us how similar the clusters are to each other according the frequency of the venues and how different each cluster are to each other. It can be used to plan which suburb to move to or how different the suburbs are in Sydney and Melbourne according to their vicinity. One might consider going to where there are a lot of parks for their growing family in the Western Suburbs of Sydney or move close to restaurants in Inner City of Melbourne to find a job. The possibility is endless.

METHODOLOGY

Using the data of suburbs and postal code, we will need to get the coordinates of the suburbs using **geocoder arcgis** to search for the postal code and the corresponding city i.e. Sydney or Melbourne.

```
[14] address_MEL = "Melbourne, VIC"
     location_mel = geocoder.arcgis(address_MEL)
     latitude_mel = location_mel.latlng[0]
     longitude_mel = location_mel.latlng[1]
     print('Melbourne\'s coordinates are : Latitude ( {} ) , Longitude ( {} )'.format(latitude_mel, longitude_mel))

     Melbourne's coordinates are : Latitude ( -37.81738999999993 ) , Longitude ( 144.96751000000006 )
```

We can use these data to **visualize** the suburbs in the map.

```
[15] map_melbourne = folium.Map(location = [latitude_mel, longitude_mel], zoom_start= 10)

     for lat, lng, post, sub in zip(melb_merged['Latitude'],melb_merged['Longitude'], melb_merged['Postal Code'],melb_merged['Suburb'] ):
       label = '{}{}'.format(sub,post)
       label = folium.Popup(label, parse_html=True)
       folium.CircleMarker(
           location = [lat,lng],
           radius = 5,
           popup = label,
           color = 'blue',
           fill = True,
           fill_color = '#14B6B0',
           fill_opacity = 0.6,
           parse_html = False).add_to(map_melbourne)

     map_melbourne
```

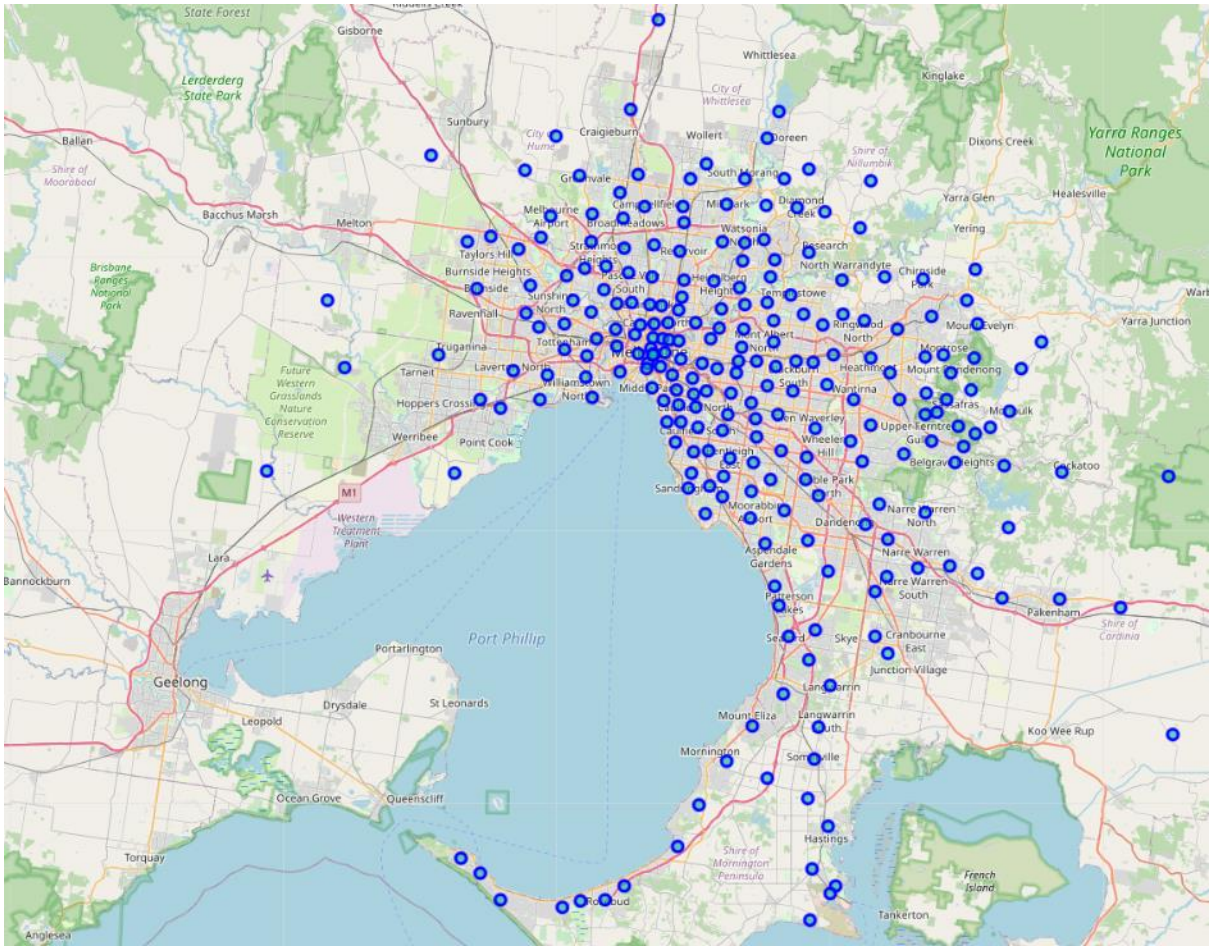

Figure 1. Map of Melbourne with circle markers to represent suburbs.

Then, we will use the coordinates (latitude and longitude) it to **explore the area around each suburb using the Foursquare API.** We can get the nearby venues for each suburb from the information that we were given by the foursquare.

	Suburb	Suburb Latitude	Suburb Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
319	Pascoe Vale ,Pascoe Vale South	-37.733552	144.93583	Pascoe Vale Hot Bread Kitchen	-37.731619	144.938204	Bakery
320	Pascoe Vale ,Pascoe Vale South	-37.733552	144.93583	Pascoe Vale RSL	-37.731593	144.938608	Australian Restaurant
321	Pascoe Vale ,Pascoe Vale South	-37.733552	144.93583	BWS	-37.731451	144.938532	Liquor Store
322	Pascoe Vale ,Pascoe Vale South	-37.733552	144.93583	Ferguson Plarre Bakehouses	-37.731710	144.938890	Bakery
323	Pascoe Vale ,Pascoe Vale South	-37.733552	144.93583	Coles	-37.731844	144.939293	Supermarket

Now, we acquired the information about the venues in each postal code that we supply the foursquare API.

We then can **use one hot coding so we can get what's the frequency** of having a specific venue within the vicinity of the suburb. We convert these categories to meaningful numbers so we can use them for our data analysis.

```
[23] #onehot coding
     melb_onehot = pd.get_dummies(melb_venues['Venue Category'], prefix = "", prefix_sep="")
     melb_onehot.shape
```

```
(2681, 273)
```

```
[24] melb_onehot.head(2)
```

	African Restaurant	American Restaurant	Antique Shop	Arcade	Art Gallery	Art Museum	Arts & Crafts Store	Arts & Entertainment	Asian Restaurant	Athletics & Sports	Australian Restaurant	Austrian Restaurant	Automc
0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	0	0	

2 rows × 273 columns

```
[25] #add suburb column back to onehot dataframe
     melb_onehot['Suburb'] = melb_venues['Suburb']
     neighb_col = melb_onehot.pop('Suburb')
     melb_onehot.insert(0, 'Suburb', neighb_col)
     print(len(melb_onehot['Suburb'].unique()))
     print(len(melb_venues['Suburb'].unique()))
```

```
241
241
```

```
[26] #group by suburb then get mean of frequency

     melb_grouped = melb_onehot.groupby('Suburb').mean().reset_index()
```

We can then **apply exploratory data analysis techniques.** For simplicity, we'll tally the top 5 venues on each suburb.

```
[27] num_top_venues = 5
     for sub in melb_grouped['Suburb']:
       #print suburb on top
       print('**' + sub + '**')

       #select the corresponding suburb then transpose data to categories line up vertically
       temp = melb_grouped[melb_grouped['Suburb'] == sub].T.reset_index()

       temp.columns = ['Venues', 'Frequency']

       #Neighborhood is the row, the succeeding ones are Venues and Frequency
       temp = temp.iloc[1:]
       temp['Frequency'] = temp['Frequency'].astype(float)
       temp = temp.round({'Frequency': 4})


       #sort the values of the transposed data by frequency in descendin order then print the top venues
       print(temp.sort_values('Frequency', ascending=False).reset_index(drop = True).head(num_top_venues))
       print ('\n')
```

```
**Albanvale ,Kealba ,Kings Park ,St Albans **
                    Venues  Frequency
0     Fast Food Restaurant       0.25
1  Health & Beauty Service       0.25
2                    Hotel       0.25
3              Pizza Place       0.25
4       African Restaurant       0.00


**Albert Park ,Middle Park **
                  Venues  Frequency
0                  Café     0.1905
1           Tram Station     0.1429
2           Soccer Field     0.0952
3     Athletics & Sports     0.0952
4  Australian Restaurant     0.0476


**Albion ,Sunshine ,Sunshine North ,Sunshine West **
                    Venues  Frequency
0                    Café        0.2
1  Fruit & Vegetable Store        0.2
2                   Trail        0.2
3               BBQ Joint        0.2
4                    Food        0.2


**Alphington ,Fairfield ,Alphington ,Fairfield **
        Venues  Frequency
0         Café        0.4
1  Sports Club        0.2
2  Gas Station        0.2
3          Bar        0.2
4        Plaza        0.0
```

Finally, we can cluster the suburbs using **Kmeans according to the frequency** of having certain venues. K-means clustering is a machine learning algorithm where there is pre-determined number of means, k, and the data points are clustered to where the nearest mean is. The means are called cluster centroid. The data points of the same clusters are then deemed similar, and each cluster is different from each other.

Run Kmeans clustering

```
[32] #Initialize Kmeans object then Fit data into the Kmeans
     kmeans_melb = KMeans(n_clusters=kclusters,init='k-means++',random_state = 0).fit(melb_grouped_clustering)

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

```
array([0, 0, 0, 0, 0, 0, 0, 0, 2, 0], dtype=int32)
```

```
[33] #add cluster labels
     suburb_venues_sorted.insert(0, 'Cluster Labels', kmeans_melb.labels_)
```

```
[34] #Consolidate all the relevant data into one
     melb_all = melb_merged
     melb_all = melb_all.join(suburb_venues_sorted.set_index('Suburb'), on = 'Suburb')
     melb_all = melb_all.dropna(axis=0, how='any')
```

```
[35] melb_all.head(2)
```

	Postal Code	Suburb	Latitude	Longitude	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd
0	3000	Melbourne	-37.810993	144.964485	0.0	Bar	Cocktail Bar	
1	3002	East Melbourne	-37.815425	144.982591	0.0	Café	Cricket Ground	

Then we can get cluster labels for each suburb which we can use to examine the data and visualize it in the map.

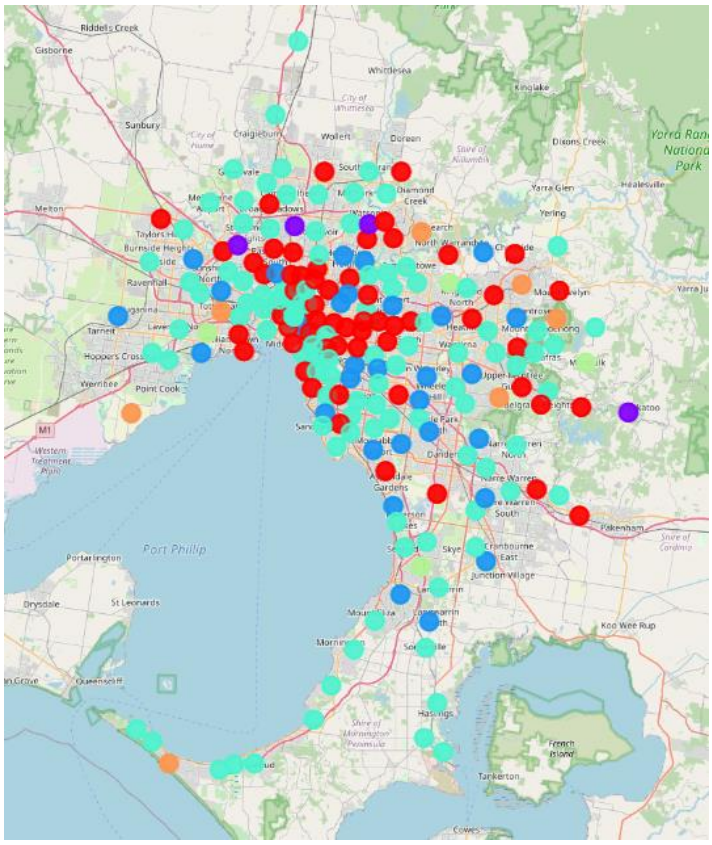

Figure 2. Map of Melbourne color-coded according to each cluster

We can then examine the data, one on isolation for each city and next, maybe we can uncover something that the both capital cities have in common. For example:

EXAMINE CLUSTERS

Now, we can examine each cluster and determine the discriminating venue categories that distinguish each cluste
assign a name to each cluster.

```
[ ]
```

Cluster 1 - Cafe

```
[ ]  cluster1 = melb_all.loc[melb_all['Cluster Labels'] == 0, melb_all.columns[[0]+ [1] + li
     cluster1
```

	Postal Code	Suburb	1st Most Common Venue	2nd
1	3002	East Melbourne	Café	C
5	3008	Docklands	Café	
9	3015	Newport ,Spotswood ,South Kingsville	Café	
10	3016	Williamstown ,Williamstown North	Pub	Conv
26	3033	Keilor East ,Keilor East	Home Service	
29	3037	Calder Park ,Delahey ,Hillside ,Sydenham ,Hill...	Fast Food Restaurant	
31	3039	Moonee Ponds	Café	
32	3040	Aberfeldie ,Essendon ,Essendon West	Food & Drink Shop	
36	3044	Pascoe Vale ,Pascoe Vale South	Bakery	
39	3047	Broadmeadows ,Dallas ,Jacana	Grocery Store	W
45	3054	Carlton North ,Carlton North ,Princes Hill	Café	
47	3056	Brunswick	Café	
48	3057	Brunswick East	Café	
49	3058	Coburg ,Coburg North ,Coburg ,Coburg North	Café	Gym / F
58	3067	Abbotsford	Café	
60	3070	Northcote	Café	
61	3071	Thornbury	Café	Food

Cluster 2 - Grocery Store

```
[ ] cluster2 =melb_all.loc[melb_all['Cluster Labels'] == 1, melb_all.columns[[0] + [1] + li
    cluster2
```

	Postal Code	Suburb	1st Most Common Venue	2nd Most C
34	3042	Keilor Park ,Airport West ,Niddrie	Grocery Store	Zoo E
51	3060	Fawkner	Grocery Store	Shopping
74	3087	Watsonia ,Watsonia North	Grocery Store	Zoo E
211	3781	Cockatoo ,Mount Burnett ,Nangana	Grocery Store	Memori

```
[ ] print('{} post codes belong to cluster 2'.format(cluster2.shape[0]))

    3 post codes belong to cluster 2
```

Cluster 3 Parks

```
[ ] cluster3 = melb_all.loc[melb_all['Cluster Labels'] == 2, melb_all.columns[[0] + [1] + l
    cluster3
```

	Postal Code	Suburb	1st Most Common Venue	2nd
3	3004	Melbourne ,Melbourne	Park	B
11	3018	Altona ,Seaholme	Pizza Place	
12	3019	Braybrook	Park	Cor
14	3021	Albanvale ,Kealba ,Kings Park ,St Albans	Tennis Court	Fast F
22	3029	Truganina ,Hoppers Crossing ,Tarneit ,Truganina	Soccer Field	Fast F
46	3055	Brunswick West	Park	Ita
69	3081	Bellfield ,Heidelberg Heights ,Heidelberg West	Park	
72	3084	Eaglemont ,Heidelberg ,Rosanna ,Viewbank	Park	
85	3101	Kew	Park	

RESULTS AND DISCUSSION

After analysing the data, we can see that there are 3 predominating clusters in Melbourne with Cluster 4 (restaurants and entertainment) taking the lion share of 136 suburbs. Cluster 1 with café as there most common denominator, and cluster 3 surrounded by parks. As observed from the map, suburbs with a lot of cafes and restaurants are more concentrated at the centre showing how the city tries to accommodate the needs of the population not only for daily needs but also for commercial purposes. Interestingly, the parks are noticeably well spaced around the area showing how the urban planning prioritise quality of life for its citizens. The Morning Peninsula is studded with suburbs with numerous restaurants which is not surprising because the area is known for its quality restaurants and a holiday destination. Cluster 5 coincides with the location of Melbourne zoos and as clusters 2 (Grocery stores) and 6 (Home Services) are noticeably on the outskirts of this capital city because these areas are less urbanised than the Central Business District.

With Sydney, there are 2 predominant clusters in Sydney namely cluster 6 (shopping needs, bars and coffee shops) with 107 post codes and cluster 1 with a lot of cafes. As expected, the Central Business District of Sydney is crowded by food and retail commercial businesses to cater the demand for them not only by everyday workers but also for the tourists, and citizens. Cafes don't only for provide the employees morning coffee, but they are also a place for meetings, for being productive, or to grab a quick snack for a diverse group of people. The area around the Sydney Opera house is lined by a lot of restaurant, even the Sydney Opera

House itself offer restaurant services. The city has a lot of different restaurants, but cluster 3 looks to be predominated by fast food restaurants to be specific. Interestingly, cluster 4 are have a lot of nearby parks. The suburbs around Ryde have a lot of greeneries while still being close to the city which may be good for people who are looking for jobs close to the city and live near parks to spend time with their family and pets.

The data is limited to foursquare's data, but there were a lot of insights that can out of it that can't be ignore. As we can observe from both analysis, both Sydney and Melbourne are sprinkled with cafes and restaurants which is consistent with the their culture. These cities are known for these industries. In a way, that's where these two are similar, top-notch hospitality services and world-renowned coffee culture. But there are differences as well like the placement of parks and shopping centres. Some suburbs are very distinct which reflects the differences of these lands' geography, history and more importantly, it's people.

If I may recommend, it would also be interesting to overlay the crime rates in suburbs to rank the safest ones or use some more data to find the best area to build a specific business.

Conclusion

In conclusion, the data may be limited but it has provided us important insights on the commercial, social, and geographical facets of 2 of the world's biggest cities. It's valuable to people who are interested in looking into moving to these cities for whatever purposes. Most importantly, the future direction of this project promising.