BATTLE OF THE SUBURBS

Exploring the Coffee Scene in Melbourne's Top Suburbs

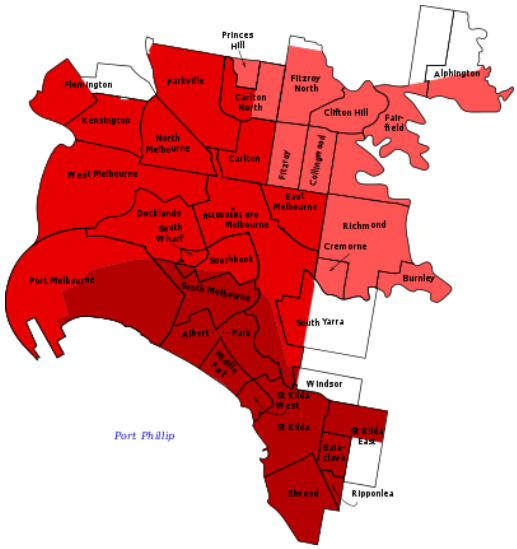
Author: Ting Xuan Ng

1. Introduction

Over the years, Melbourne has become the epicentre of world coffee culture, with Melbourne Cafes becoming a genre in the food and beverages scene all over the world. With over a century of history behind us, Melbournians not only have the taste, but also a knowledge and literacy in coffee flavours. The competition that arises from this literacy therefore makes opening a café here a rather exciting yet risky adventure for any coffee loving entrepreneur.

My client plans to open a Café in Melbourne that prides itself in the quality of their coffee, while having the most reasonable prices than their competitors. It is therefore important that before he chooses a suitable location around Melbourne, specifically a suburb that offers good business opportunities without providing too much competition to their business. Using the data visualization techniques and machine learning algorithms learned throughout the course, this project aims to perform a comparative analysis between suburbs in Melbourne.

In Australia, a suburb is a named and bounded locality of a city, with an urban nature. Each suburb has a 4 digit post code. In this project, we will be exploring the suburbs within the Inner City Municipalities (Red region shown below).



Map of Suburbs in the Inner Melbourne Councils

Target Audience:

- Entrepreneurs or business owners interested in opening a café or expanding their business across Melbourne
- Any aspiring data scientists looking to play around with Python and implement some data analysis and visualization techniques.

2. Data

For this project, three main sources of data were used: a Wikipedia page of a list of suburbs in Melbourne, an excel sheet of Melbourne's population forecast obtained from the Australian Bureau of Statistics website, and Foursquare for the most common venues within a specified radius of each suburb.

Due to difficulties in scraping the Wikipedia page and time restrictions, an initial dataframe of each of the Melbourne Suburbs, their postcodes as well as their coordinates were extracted manually and saved into a .csv file. The excel file of Melbourne's population forecast was easily downloaded and imported from the website for further cleaning and preparation. Lastly, Foursquare data on venues around the suburbs were obtained via API calls.

[4]:

	Suburb	2021 Population Forecast
7	Melbourne (CBD)	61190.0
13	Southbank	29047.0
3	Carlton	26239.0
9	North Melbourne	19772.0
4	Docklands	19709.0
6	Kensington	11905.0
10	Parkville	9194.0
5	East Melbourne	6152.0
12	South Yarra	4846.0
11	Port Melbourne	13.0

[3]:

	Suburb	2021 Population Forecast	Avg. Annual % Change
7	Melbourne (CBD)	61190.0	4.10175
13	Southbank	29047.0	3.93994
3	Carlton	26239.0	2.47083
9	North Melbourne	19772.0	4.01197
4	Docklands	19709.0	5.27668
6	Kensington	11905.0	2.69354
10	Parkville	9194.0	1.38997
5	East Melbourne	6152.0	0.939546
12	South Yarra	4846.0	0.35626
11	Port Melbourne	13.0	34.5063

Left Table: Dataframe of Melbourne Suburbs, Post Codes and Coordinates (First 5 lines)
Right Table: Dataframe of Population Forecast in different Melbourne suburbs

[4]:

	Suburb	2021 Population Forecast
7	Melbourne (CBD)	61190.0
13	Southbank	29047.0
3	Carlton	26239.0
9	North Melbourne	19772.0
4	Docklands	19709.0
6	Kensington	11905.0
10	Parkville	9194.0
5	East Melbourne	6152.0
12	South Yarra	4846.0
11	Port Melbourne	13.0

Merged Dataframe containing the top 10 Suburbs in Melbourne by population

3. Methodology & Results

The general workflow of this project is as follows:

i. Obtaining data either by scraping a Wikipedia page or importing the relevant files, followed by the necessary cleaning and preparation.
ii. Pass the geographical coordinates of each suburb to the Foursquare API, which returns a list of venues in the suburb within a user-specified radius and call limit. These exploratory data will be used to explore the coffee culture in Melbourne.
iii. Using the Elbow method to determine the optimum K value in order to perform k-means clustering, an unsupervised Machine Learning technique.
iv. Perform k-means clustering on the top venues data to cluster the suburbs according to certain features. In this project, the algorithm is expected to cluster the suburbs based on their most popular venues.
v. Using information provided by these clusters to identify suitable locations to open his/her café.

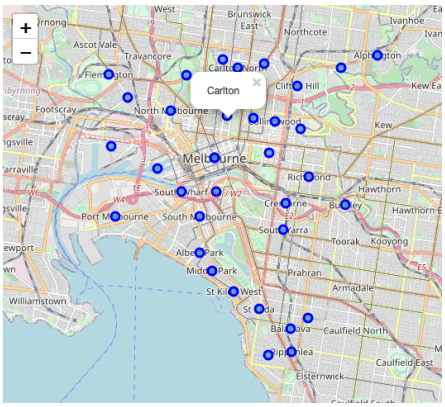
3.1 Visualizing and Exploring the initial Dataset (All Melbourne Suburbs)

Using the Folium library, I created a leaflet map of Melbourne City with all 35 suburbs superimposed onto the map. Using the code shown below:

```
[*]: # Converting Melbourne address to coordinates:
address = 'Melbourne, AU'

geolocator = Nominatim(user_agent="Melbourne_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Melbourne are {}, {}.'.format(latitude, longitude))
```

```
[*]: # Plotting map:
map_melb = folium.Map(location=[latitude, longitude], zoom_start=12)

# add markers to map
for lat, lng, label in zip(melb_suburb['Latitude'], melb_suburb['Longitude'], melb_suburb['Suburb']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_melb)

map_melb
```

As a result, the leaflet map looks like this:



Leaflet map of Melbourne City and its suburbs

Next, by passing the geographical coordinates of each suburb in the Foursquare API call, Foursquare API returns a list of venues in each suburb within a specified radius and call limit. In my case, to ensure not to exceed the http request limitations, the call was set to return 100 outputs and the radius parameter to 500 (500m radius), as shown in the code snippet below.

```
# Getting top 100 venues within Melbourne CBD???
# type your answer here
LIMIT = 100
radius = 500
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(CLIENT_ID, CLIENT_SECRET, VERSI
print(url)
results = requests.get(url).json()
#results
```

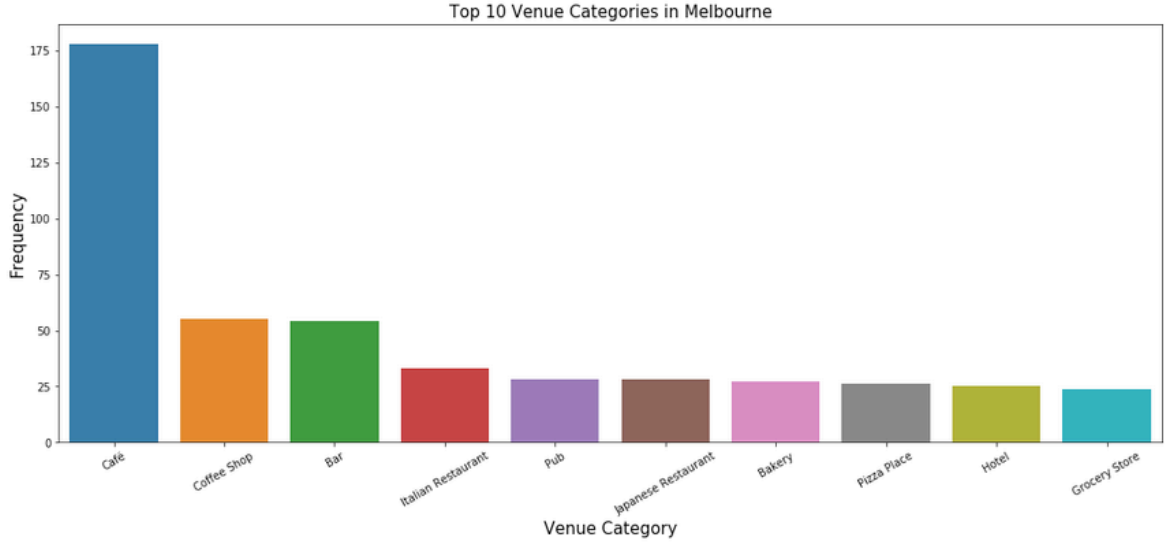
To obtain a list of nearby venues in each of the 35 suburbs, a function *getNearbyVenues* was written. Calling the function returns a list with a total of 200 unique venue categories. The table below is the resulting dataframe named *melb_venues:*

[12]:

	Suburb	Suburb Latitude	Suburb Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Carlton	-37.800423	144.968434	Carlton Wine Room	-37.798584	144.968610	Wine Bar
1	Carlton	-37.800423	144.968434	D.O.C. Pizza & Mozzarella Bar	-37.798954	144.968490	Pizza Place
2	Carlton	-37.800423	144.968434	Yo-Chi	-37.798659	144.967849	Frozen Yogurt Shop
3	Carlton	-37.800423	144.968434	Gewürzhaus	-37.799050	144.967480	Gourmet Shop
4	Carlton	-37.800423	144.968434	Baker D. Chirico	-37.798788	144.968499	Bakery

First 5 rows of the melb_venues Dataframe containing 100 most popular venues for each of the 35 suburbs

Note that the popular spots returned by Foursquare API depends on the foot traffic at the time the API call is made. As a result, we may get slightly different popular venues at different times of the day.



Bar chart of the Top 10 Most Popular venues across Melbourne

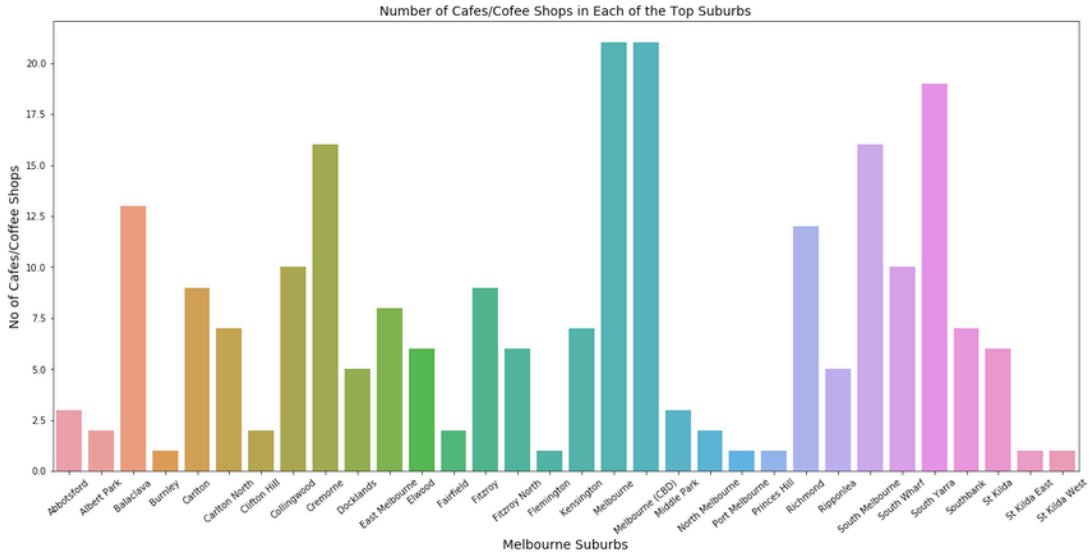
As expected, the top venues in Melbourne are Cafes and Coffee shops. One assumption to note is that since the client is interested in opening a cafe that emphasizes on their coffee in terms of sales and quality, Cafés and Coffee Shops were considered to be in the same category in this project.

In fact, using the code snippet shown below, we can then dive deeper into the Melbourne coffee scene to find out which of the 35 suburbs have the highest number of cafes/coffee shops.

[16]:

```
# creating a dataframe of all cafes around Melbourne
top_melb_cafe = melb_venues[melb_venues['Venue Category'].str.contains('Coffee Shop|Café|Cafe')].reset_index

print(top_melb_cafe.shape)
#top_melb_cafe

compare = top_melb_cafe.groupby(['Suburb'])['Venue Category'].apply(lambda x: x[x.str.contains('Coffee Shop|
compare_df = compare.to_frame().reset_index()
compare_df.columns = ['Suburb', 'No of Cafes/Coffee Shops']
compare_df.index = np.arange(1, len(compare_df)+1)
#compare_df.head()
```

Plotting the compare_df dataframe would thus result in the following:



Bar chart

3.2 Exploring the Cafes/Coffee Shops in Melbourne

For the following part of the project, I decided to focus on the top 10 suburbs in Melbourne, namely: Carlton, Docklands, East Melbourne, Kensington, Melbourne (CBD), North Melbourne, Parkville, Port Melbourne, South Yarra and Southbank.

Now calling the function get NearbyVenues again, but this time using the dataset that contains just the top 10 suburbs.

(Left) Leaflet map of all the cafes/coffee shops in the Top 10 Suburbs
(Right) Bar Chart of the Number of Cafes in the Top Suburbs

Again, note that some suburbs may not appear on the bar chart. This is because the popular spots returned by Foursquare API depends on the foot traffic at the time the API call is made. Therefore we may get slightly different popular venues at different times of the day.

3.3 Elbow Method to Determine Optimum K-Value

A crucial step for any unsupervised algorithm is to determine the number of clusters into which the data may be clustered. One of the ways to do this would be the Elbow Method:

```
melb_grouped_clustering = melb_grouped.drop('Suburb', 1)

distortions = []
inertias = []
mapping1 = {}
mapping2 = {}
K = range(1,10)
for k in K:
    kmeanModel = KMeans(n_clusters=k, random_state=0).fit(melb_grouped_clustering)
    kmeanModel.fit(melb_grouped_clustering)

    distortions.append(sum(np.min(cdist(melb_grouped_clustering, kmeanModel.cluster_centers_, 'euclidean'), axis=1))/ melb_grouped_clustering.shape[0])
    inertias.append(kmeanModel.inertia_)

    mapping1[k] = sum(np.min(cdist(melb_grouped_clustering, kmeanModel.cluster_centers_, 'euclidean'), axis=1)) / melb_grouped_clustering.shape[0]
    mapping2[k] = kmeanModel.inertia_
```

Two important definitions to note are Distortion and Inertia.

- Distortion: the average of the squared distances from the cluster centers of the respective clusters. Typically uses Euclidean distance metric.
- Inertia: the sum of the square distances of samples to their closest cluster center.

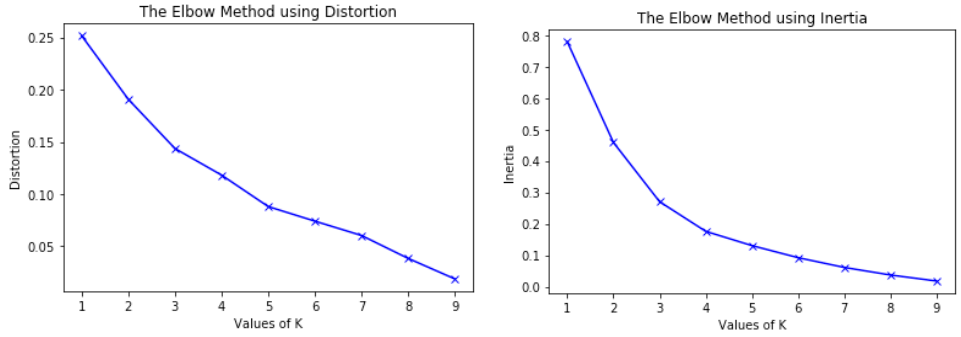


Figure 1: Bar chart

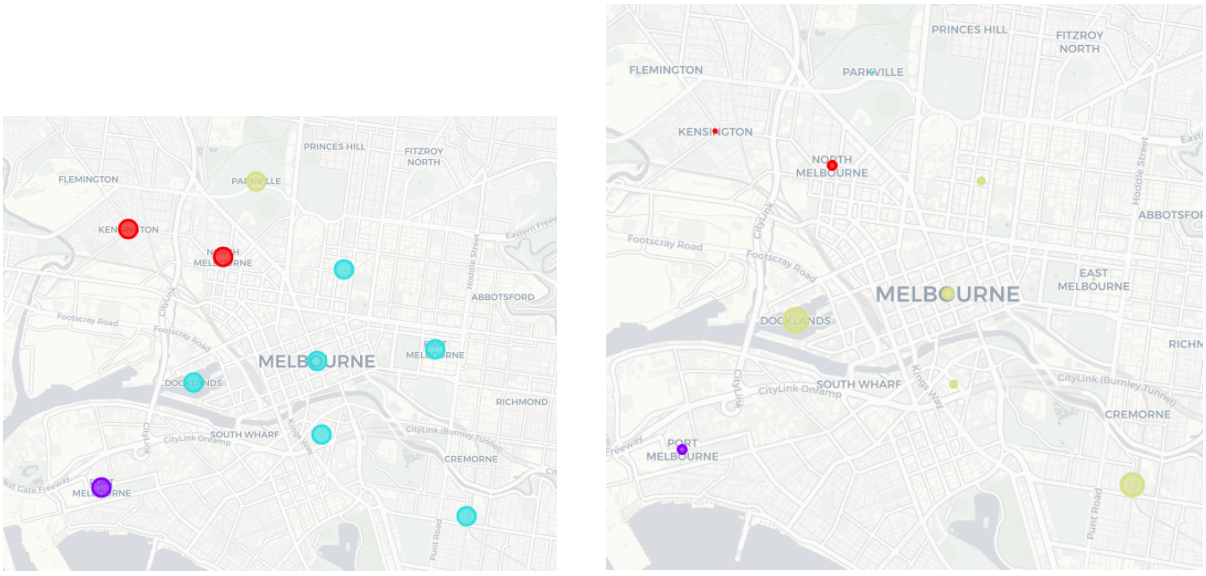
Based on the elbow method using Inertia, it is clear that the elbow is located at K=4. We shall thus perform K-Means Cluster with 4 Clusters

3.3 K-Means Clustering

K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. Clustering the data may allow us to discover patterns or valuable information that can help determine the best location for a new café.

With the optimal k value found, we can now perform K-Means Clustering from the Sci-kit learn library, splitting the dataset into 4 clusters as shown below. To visualize the results, two plots were made: the left figure shows the distribution of aech of the clusters, whereas the figure on the right is a similar plot with the radius of the clusters representing the number of Cafes/ Coffee Shops in each suburb.

```
kclusters = 4
melb_grouped_clustering = melb_grouped.drop('Suburb', 1)

kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(melb_grouped_clustering)
print ("Check Cluster labels :", kmeans.labels_[0:10]) # checking cluster labels for each row of the dataframe


# add clustering labels
top_sub_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

# merge Sorted most visited venues in each suburb dataframe with melb_merged to add latitude/longitude for each suburb
melb_sub_coord_cluster = melb_merged

melb_sub_coord_cluster = melb_sub_coord_cluster.join(top_sub_venues_sorted.set_index('Suburb'), on='Suburb')
melb_sub_coord_cluster
```



(Left) Distribution of all the clusters in different colours
(Right) Distribution of the clusters, with the radius of each cluster representing the number of cafes/coffee shops in the suburb

The results of the exploratory data analysis and k-means clustering can be summarized as below:

- Cafes and coffee shops are the two most popular venue categories in Melbourne
- Exploratory analysis shows that Melbourne CBD, South Yarra, South Melbourne Cremorne and Richmond have the most coffee shops/cafes out of all suburbs.
- The Elbow Method using Inertia indicates that splitting to 4 clusters yield the best results.
- Since the clustering was based mainly on the most popular venues of each suburb, Carlton, Docklands, Melbourne CBD etc were grouped into the same cluster dince their popular venues are dominated by cafes, restaurants and bars.
- North Melbourne and Kensington in Cluster 0 both have a relatively good balance of venue categories, with Cafes being the most common followed by outdoor parks, gyms and grocery stores.
- Cluster 1 and 3 seem to stand out, with no popular cafes or coffee shops around both areas.

4. Discussion

According to this analysis, Parkville and Port Melbourne will provide least competition for a new café, with venues involving outdoor activities being the most common venue in the area.

However, it is worth noting that Port Melbourne used to be an industrial port, with the lowest population among the other suburbs, thus other factors such as foot traffic and demographics should also be considered. On the other hand, with a good balance of food and activity related venues, North Melbourne and Kensington from Cluster 0 can be considered as residential areas. This could also be a good location for families who want to avoid the hustle and bustle of the metropolitan areas.

Some of the challenges in this project include the lack of complete/consistent data source related to Melbourne City. One example would be the discrepancies regarding the number of Melbourne suburbs in the Population Forecast data and the list of suburbs in Melbourne on Wikipedia. Due to time restrictions, a list of Melbourne suburbs was extracted manually together with their postcodes and then imported into a .csv file, and more detailed analysis was done only on suburbs that contain population forecast data.

Furthermore, many other factors such as land prices or rental costs, demographics, crime rates, general accessibility etc should also be taken into consideration when choosing which of the suburbs to analyse. However, since of the 10 suburbs were relatively evenly distributed around Melbourne CBD, the most populated part of Melbourne, it can thus be argued that the study could still produce fairly accurate and unbiased results.

5. Conclusion

In this project, I was given the opportunity to apply what I've learned such as using python libraries for web scraping, utilizing the Foursquare location data to explore the coffee scene in Melbourne, as well as applying Machine Learning algorithms to cluster and segment data. As predicted, the clustering was done based on the most popular venues in each suburb. Limitations and challenges were also discussed along with suggestions for improvement. All in all, the results of this project have proven successful in providing an interesting perspective on the coffee culture around Melbourne, along with some promising opportunities for a new café.

6. References

https://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/3218.02018-19?OpenDocument

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