Main Objective of the Analysis:

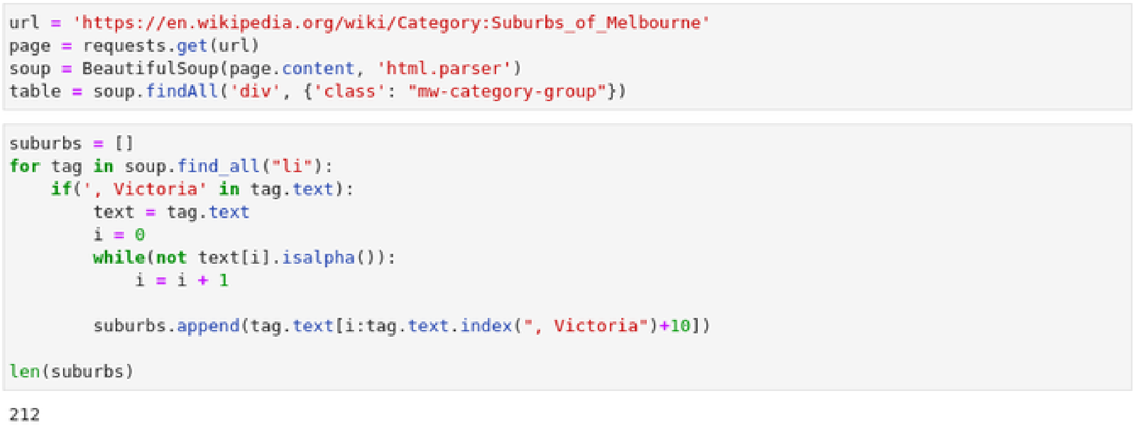
The main goal of this project is to collect and analyze data in order to select a location in Melbourne to open a Cafeteria. We want to help a business owner planning to open up a Cafe in a location by exploring better facilities around the Suburb.

This is an unsupervised machine learning problem where we need to group together suburbs having similar facilities. We will use K Means Clustering and Hierarchical Clustering to solve this problem.

Data Description:

- List of Suburbs in Melbourne, Australia which I have extracted from: https://en.wikipedia.org/wiki/Category:Suburbs_of_Melbourne

- Latitude & Longitude of all the suburbs using Geocoder- venues in each suburb from foursquare API https://foursquare.com/



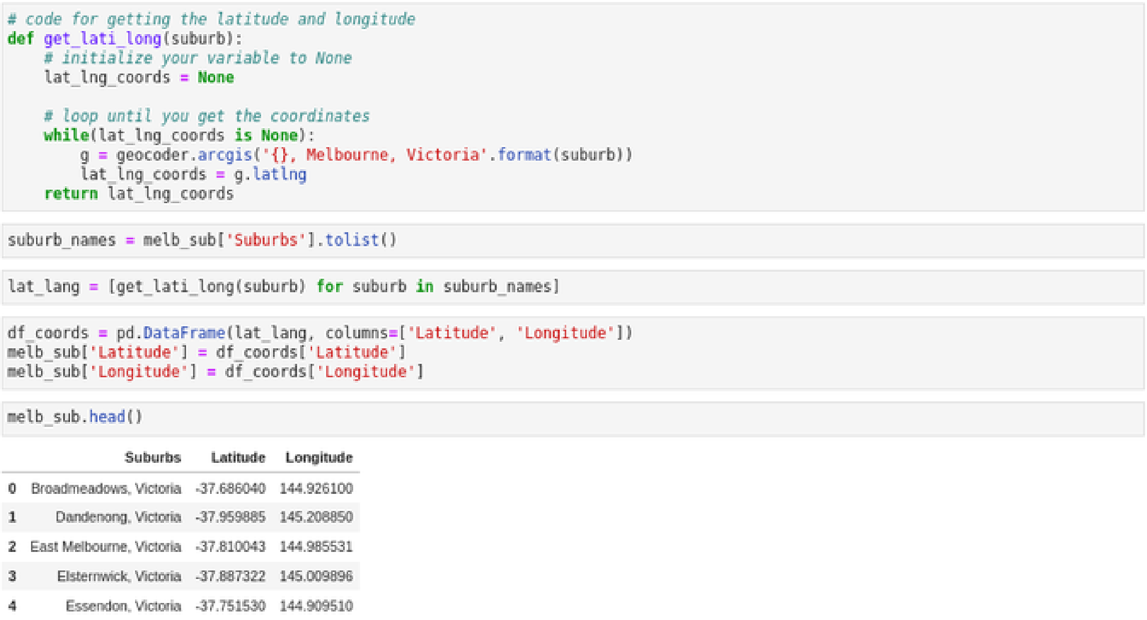
```
url = 'https://en.wikipedia.org/wiki/Category:Suburbs_of_Melbourne'
page = requests.get(url)
soup = BeautifulSoup(page.content, 'html.parser')
table = soup.findAll('div', {'class': "mw-category-group"})
```

```
suburbs = []
for tag in soup.find_all("li"):
    if(', Victoria' in tag.text):
        text = tag.text
        i = 0
        while(not text[i].isalpha()):
            i = i + 1

        suburbs.append(tag.text[i:tag.text.index(", Victoria")+10])

len(suburbs)
```

```
212
```



```
# code for getting the latitude and longitude
def get_lati_long(suburb):
    # initialize your variable to None
    lat_lng_coords = None

    # loop until you get the coordinates
    while(lat_lng_coords is None):
        g = geocoder.arcgis('{}, Melbourne, Victoria'.format(suburb))
        lat_lng_coords = g.latlng
    return lat_lng_coords
```

```
suburb_names = melb_sub['Suburbs'].tolist()
```

```
lat_lang = [get_lati_long(suburb) for suburb in suburb_names]
```

```
df_coords = pd.DataFrame(lat_lang, columns=['Latitude', 'Longitude'])
melb_sub['Latitude'] = df_coords['Latitude']
melb_sub['Longitude'] = df_coords['Longitude']
```

```
melb_sub.head()
```

	Suburbs	Latitude	Longitude
0	Broadmeadows, Victoria	-37.686040	144.926100
1	Dandenong, Victoria	-37.959885	145.208850
2	East Melbourne, Victoria	-37.810043	144.985531
3	Elsternwick, Victoria	-37.887322	145.009896
4	Essendon, Victoria	-37.751530	144.909510

Data Understanding

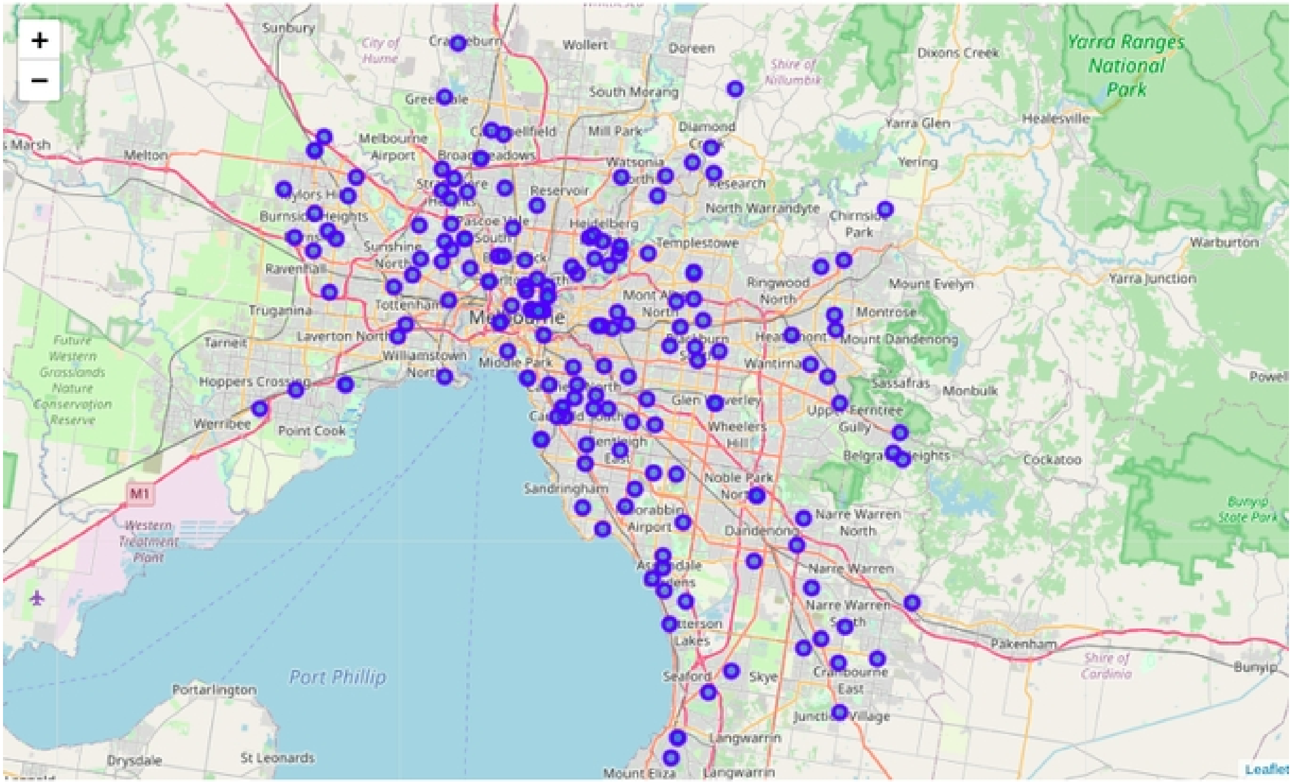
- The Wikipedia page contains a list of suburbs in Melbourne. There are 212 suburbs in Melbourne which I extracted using a web scraping technique with the help of Python BeautifulSoup and Request packages.

- the geographical coordinates such as latitude and longitude of each suburb were collected using Python's Geocoder package.

- Then, Foursquare API was used to extract details about the various venues present in each suburb.

- Once, the location data was extracted by using Geocoder, I used the Folium package to visualize the data on a map. This ensured us that the data we retrieved was correct.

- Foursquare API was used to obtain the top 100 venues within a radius of 2000 meters.



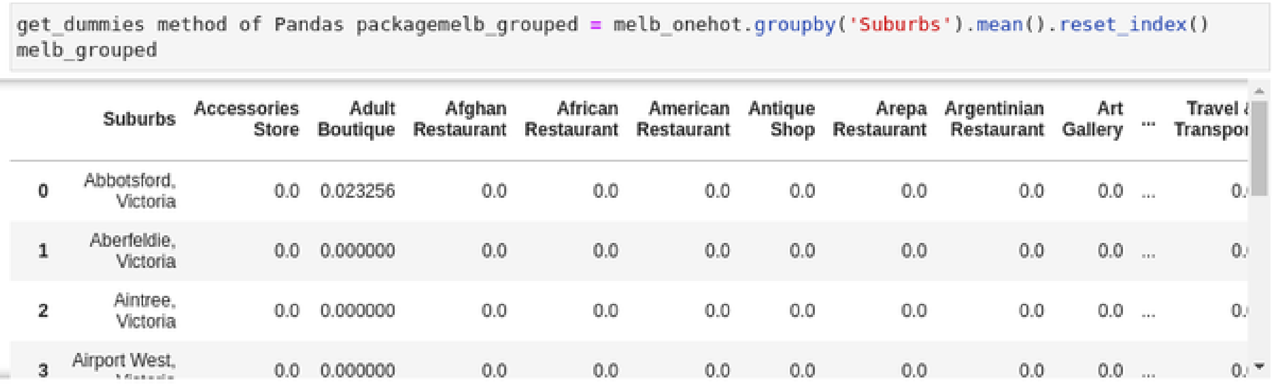
Data Cleaning and Feature Engineering

- Converted the data into dummy variables using get_dummies method of Pandas package that will be essential for performing clustering algorithm

- Grouped the data by Suburb & also taking the mean of the frequency of occurrence of each category.

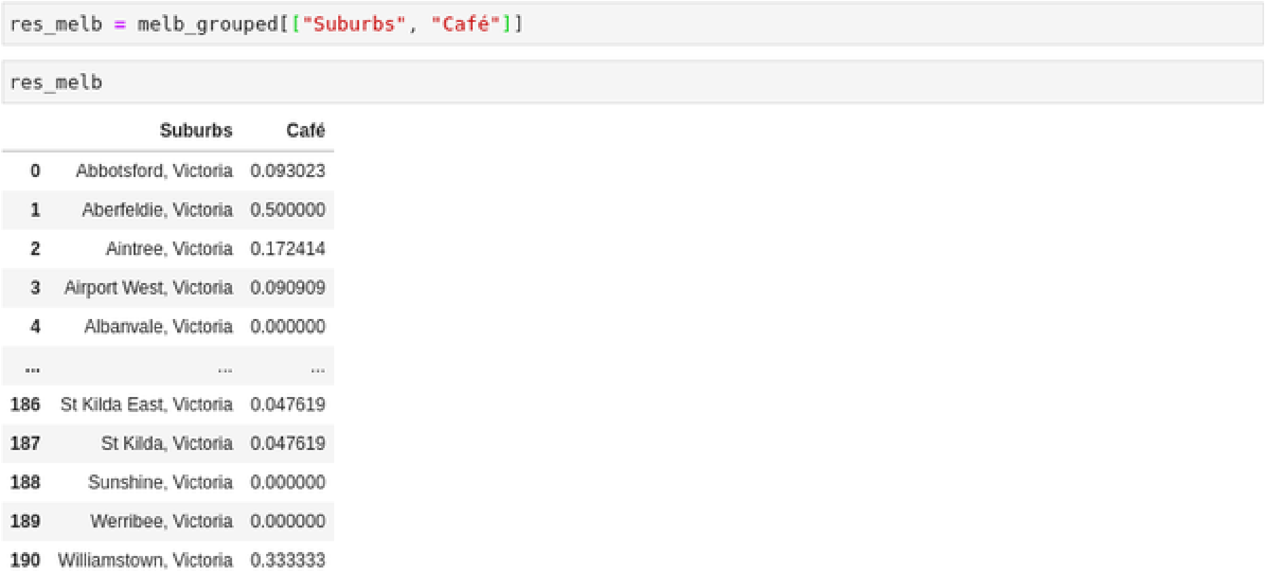
- I extracted the data of the Cafeteria only

- Our final data frame had two variables: suburb name and the mean of the frequency of occurrence of cafes



```
get_dummies method of Pandas packagemelb_grouped = melb_onehot.groupby('Suburbs').mean().reset_index()
melb_grouped
```

	Suburbs	Accessories Store	Adult Boutique	Afghan Restaurant	African Restaurant	American Restaurant	Antique Shop	Arepa Restaurant	Argentinian Restaurant	Art Gallery	...	Travel & Transpo
0	Abbotsford, Victoria	0.0	0.023256	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.
1	Aberfeldie, Victoria	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.
2	Aintree, Victoria	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.
3	Airport West,	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.



```
res_melb = melb_grouped[["Suburbs", "Café"]]
```

```
res_melb
```

	Suburbs	Café
0	Abbotsford, Victoria	0.093023
1	Aberfeldie, Victoria	0.500000
2	Aintree, Victoria	0.172414
3	Airport West, Victoria	0.090909
4	Albanvale, Victoria	0.000000
...	...	...
186	St Kilda East, Victoria	0.047619
187	St Kilda, Victoria	0.047619
188	Sunshine, Victoria	0.000000
189	Werribee, Victoria	0.000000
190	Williamstown, Victoria	0.333333

Modeling

- Performed clustering on the data using K-means clustering and Hierarchical Clustering.

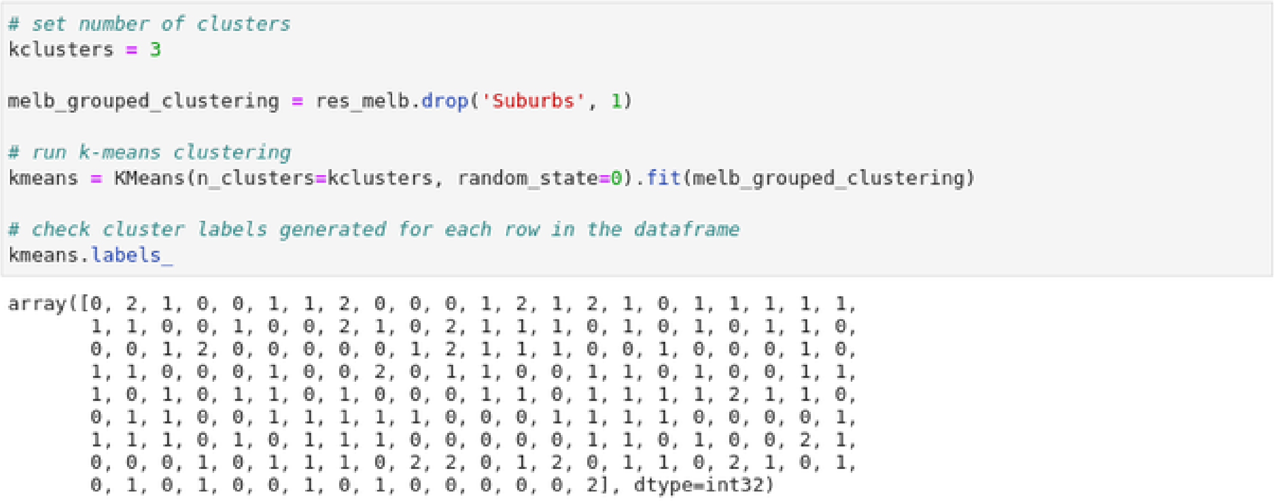
- For K means Clustering I used k = 3, 4, 5 clusters based on the frequency of occurrence of Cafes in each suburb.

- Found out the suburb which had the highest concentration of Cafes and also the lowest concentration

Results

I decided to use 3 clusters for this problem as this gives the best result. Categorized the data into 3 categories using K-means clustering based on the frequency of occurrence for ‘Cafe’.

- Cluster 0: Suburbs with a low number of Cafes.
- Cluster 1: Suburbs with a moderate number of cafes.
- Cluster 2: Suburbs with a high concentration of Cafe.



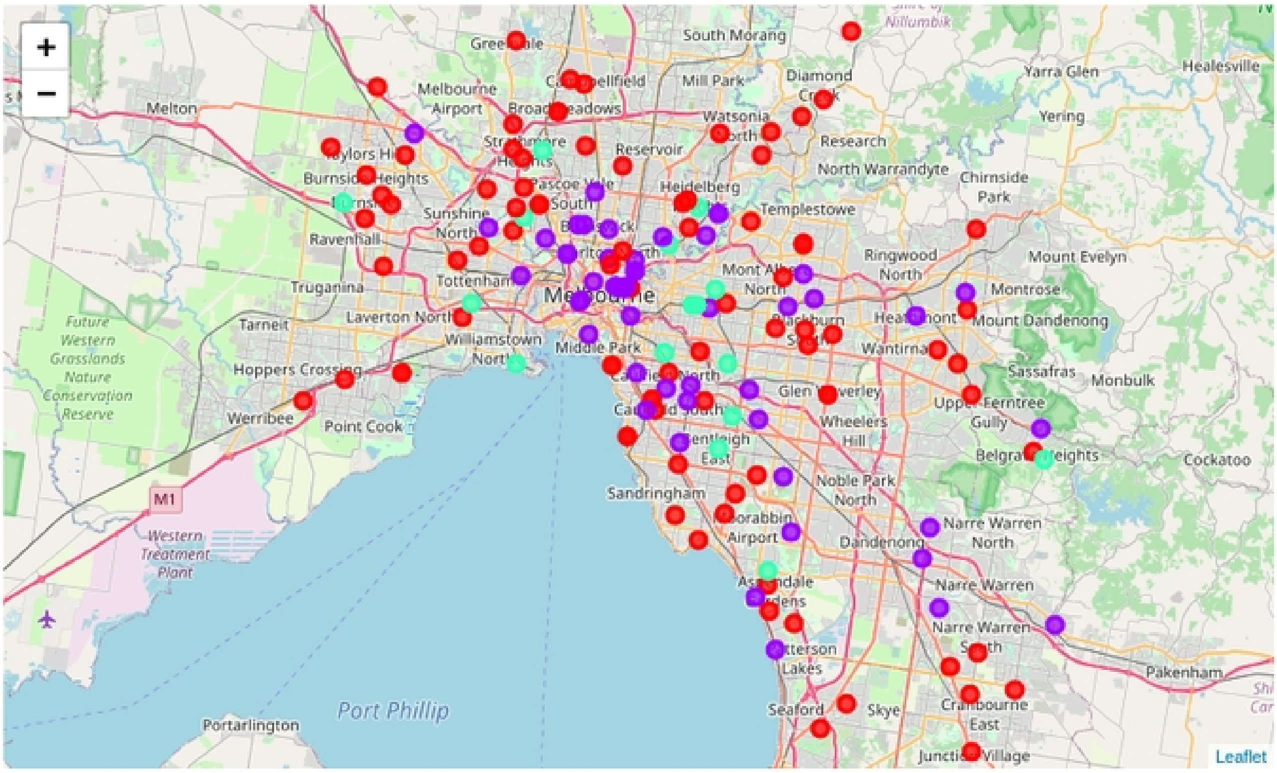
```
# set number of clusters
kclusters = 3

melb_grouped_clustering = res_melb.drop('Suburbs', 1)

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

# check cluster labels generated for each row in the dataframe
kmeans.labels_
```

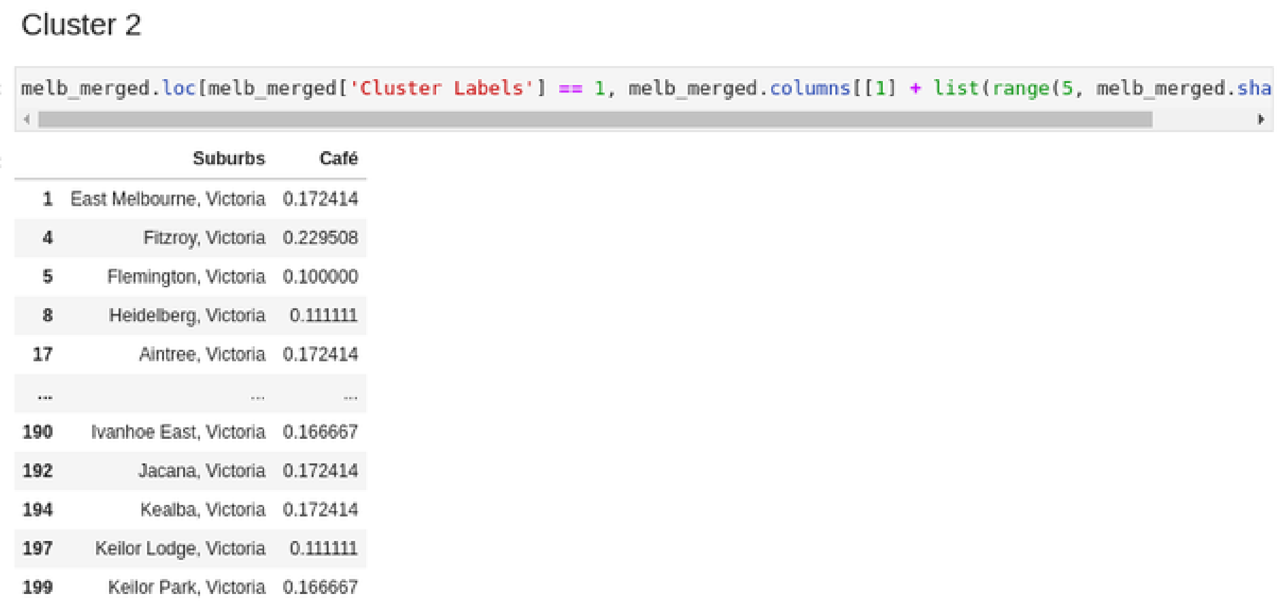
```
array([0, 2, 1, 0, 0, 1, 1, 2, 0, 0, 0, 1, 2, 1, 2, 1, 0, 1, 1, 1, 1, 1,
       1, 1, 0, 0, 1, 0, 0, 2, 1, 0, 2, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 2, 0, 0, 0, 0, 0, 1, 2, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0,
       1, 1, 0, 0, 0, 1, 0, 0, 2, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1,
       1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 2, 1, 1, 0,
       0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 2, 1,
       0, 0, 0, 1, 0, 1, 1, 1, 0, 2, 2, 0, 1, 2, 0, 1, 1, 0, 2, 1, 0, 1,
       0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 2], dtype=int32)
```



Evaluation

- Cluster 0 is displayed as the red color represents a greater opportunity and high potential but also suffers from the risk of having fewer customers as those areas are not busy areas.

- As a new business owner it wouldn’t be wise enough to choose cluster 2. Therefore, I would recommend that cluster 1 represented by blue color, should be chosen where there is medium competition but greater opportunity.



Cluster 2

```
melb_merged.loc[melb_merged['Cluster Labels'] == 1, melb_merged.columns[[1] + list(range(5, melb_merged.sha
```

	Suburbs	Café
1	East Melbourne, Victoria	0.172414
4	Fitzroy, Victoria	0.229508
5	Flemington, Victoria	0.100000
8	Heidelberg, Victoria	0.111111
17	Aintree, Victoria	0.172414
...	...	...
190	Ivanhoe East, Victoria	0.166667
192	Jacana, Victoria	0.172414
194	Kealba, Victoria	0.172414
197	Keilor Lodge, Victoria	0.111111
199	Keilor Park, Victoria	0.166667

Suggestions for Next Step:

I could get the population and average income of the suburbs, and then calculate the money to cafeterias ratio = population * income / number of cafes. The suburb with highest ratio would be the best opportunities, as they have a lot of population and money but less competition.