Melbourne Dwelling and Café dataset.

Table of Contents

Use case:

In a 6-week project aimed at guiding potential investors in the Melbourne Cafe business, the focus is on understanding suburb growth and existing cafe landscapes. Data on dwelling growth and the number of cafes with seating capacities will be collected, cleaned, and integrated. During weeks 3-4, exploratory data analysis and visualizations will be conducted to highlight growth patterns, and provide a quick, informative snapshot for users. The following week will be dedicated to implementing and fine-tuning an ARIMA model to predict dwelling growth, considering time series characteristics. The goal is to provide investors with a tool that allows them to swiftly identify potential areas for cafe business growth in Melbourne.

Data from MOP

Café, restaurant, bistro seats:
https://data.melbourne.vic.gov.au/explore/dataset/cafes-and-restaurants-with-seating-capacity/information/

Development Activity Monitor:
https://data.melbourne.vic.gov.au/explore/dataset/development-activity-monitor/information/?disjunctive.status&disjunctive.clue_small_area&disjunctive.clue_block

User Story:

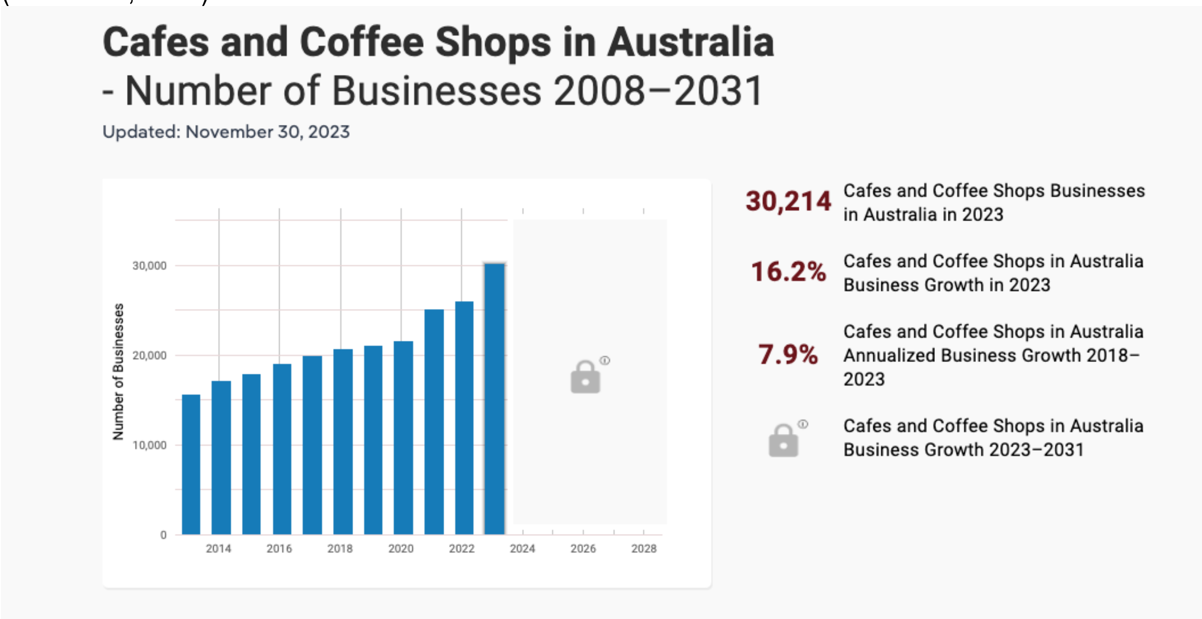
As an inspired Coffee shop owner living in the city of Melbourne, I want to find a suburb with a high potential where there is a lack of competition. I am looking for a product which will provide a quick insight in the dwelling development in Melbourne and how this relates to the number of currently available Cafés.
The product needs to provide a snapshot of each suburb's dwellings and number of Café seats available. Moreover, a ratio of number of seats per dwelling is required to pinpoint the areas of high growth and low competition.
A visualisation including the map of Melbourne including information such as suburb name and ratio of seats to dwelling is essential.

Exploratory data analysis (EDA):

The EDA is aimed at understanding the dataset, which includes viewing the different features in each dataset and identify the features required for further analysis.
The second step is to check the selected features for duplicates and null values and rectify if necessary. After which the datasets need reshaping to provide necessary insights for analysis and for predictive modelling. Moreover, the two datasets need to be merged and insight from both dataset displayed both using visuals such as plots and in tables for quick analytics.

Step 1: Checking online for the potential of the project.

(IBISWorld, 2023)



City of Melbourne
https://news.melbourne.vic.gov.au › melbourne-coffee...

Melbourne baristas share what's next in coffee

26 Sept 2022 — There are almost 50 dedicated **coffee shops** amid 840 cafes in the City of **Melbourne**, according to data from last year's Census of Land Use and ...

People also ask

- What is the growth rate of Coffee Shops?
- Is the coffee industry growing in Australia?
- Is Melbourne the coffee capital of Australia?
- Why is coffee so popular in Melbourne?

Feedback

IBISWorld
https://www.ibisworld.com › number-of-businesses › c...

Cafes and Coffee Shops in Australia - Number of Businesses

29 Nov 2023 — The number of businesses in the Cafes and **Coffee Shops** industry in Australia has grown 7.9% per year on average over the five years between 2018 ...

Kona Coffee Farmers Association
https://konacoffeefarmers.org › 2015/05 › Cafe...

Cafes-and-Coffee-Shops-in-Australia-Industry- ...

The Cafes and **Coffee Shops** industry has grown strongly over the past five years, owing much of its success to Australia's love for quality coffee and ...
31 pages

The pages above provide informative insights in the growth of Coffee shops in Melbourne and Australia.

Step 2: Identify the dataset from Melbourne Open Data, access the data through APIs.

- Appling APIs:

API function

In [3]:

```python
def fetch_data(base_url, dataset, api_key, num_records, max_offset, offset=0):
    '''
        The Function is used to return a dataset from API

    '''
    all_records =[]
    #Maximum number of requests

    while True:
        # Maximum limit check
        if offset > max_offset:
            break

        # Create API request URL
        filters = f'{dataset}/records?limit={num_records}&offset={offset}&refine=census_
        url = f'{base_url}{filters}&api_key={api_key}'

        # Start request
        try:
            result = requests.get(url, timeout=10)
            result.raise_for_status()
            records = result.json().get('results')
        except request.exceptions.RequestException as e:
            raise Exception(f'API request failed: {e}')

        if records is None:
            break

        all_records.extend(records)
        if len(records) < num_records:
            break

        # next cycle offset
        offset += num_records

    # Dataframe all data
    df = pd.DataFrame(all_records)
    return df

```

API for the café dataset

In [4]:

```python
# API deconstructed below:
API_KEY = '501c0c6bc1c0b59eb726ecacb4075dc40a606494551bd44bf024087c'
BASE_URL = "https://data.melbourne.vic.gov.au/api/explore/v2.1/catalog/datasets/"
DATASET = 'cafes-and-restaurants-with-seating-capacity'
NUM_RECORD = 20
MAX_OFFSET = 3031
```

API for Dwelling dataset

- Display the dataset

In [3]:

```python
# API decontructed below:
API_KEY = 'd503386bd7565cee5ef152d9dec187036f46a47236cb9ffff66a05b6'
BASE_URL = "https://data.melbourne.vic.gov.au/api/explore/v2.1/catalog/datasets/"
DATASET = 'development-activity-monitor'
NUM_RECORD = 20
MAX_OFFSET = 1406
```

Step 3: Understanding the dataset features and select features.

- Café dataset:

In [10]:
```
# Describe
df.describe()
```

Out[10]:

	block_id	number_of_seats	longitude	latitude
count	3031.000000	3031.000000	3031.000000	3031.000000
mean	346.588255	57.149786	144.960496	-37.812743
std	437.984272	139.872363	0.011168	0.009101
min	1.000000	2.000000	144.904228	-37.849719
25%	52.500000	16.000000	144.955504	-37.817309
50%	95.000000	33.000000	144.962183	-37.813296
75%	644.000000	68.000000	144.967240	-37.808975
max	2546.000000	4920.000000	144.990561	-37.777494

In [8]:
```
df.head(2)
```

Out[8]:

	census_year	block_id	property_id	base_property_id	building_address	clue_small_area	trading_name	business_address	i
0	2022	1110	620301	620301	120 Pearl River Road DOCKLANDS VIC 3008	Docklands	Yassas	Shop 14A, Ground 120 Pearl River Road DOCKLAND...	
1	2022	1112	103980	103980	Flinders Wharf Apartments 40-66 Siddeley Stree...	Docklands	Them Authentic Vietnamese Cuisine & Rolls	Part Unit 13, Ground 60 Siddeley Street DOCKLA...	

In [33]:
```
df.shape
```

Out[33]: (3031, 15)

  - The dataset contains most the information required for the project. No change in dataset features yet as more interrogation is required such as duplicates.

- Dwelling dataset:

In [8]:
```
# Checking the Shape of the DataSet
df.shape
```

Out[8]: (1407, 42)

In [9]:
```
# Describe
df.describe()
```

Out[9]:

	clue_block	floors_above	resi_dwellings	studio_dwe	one_bdrm_dwe	two_bdrm_dwe	three_bdrm_dwe	student_apar
count	1407.000000	1407.000000	1407.000000	1407.000000	1407.000000	1407.000000	1407.000000	1407.(
mean	551.705046	13.852168	78.786780	1.314854	23.115849	29.083866	4.879886	12.(
std	507.439410	15.688088	156.933143	10.708723	63.554807	75.944719	15.724167	72.(
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.(
25%	114.500000	3.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.(
50%	430.000000	8.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.(
75%	790.000000	19.000000	82.500000	0.000000	3.000000	9.000000	0.000000	0.(
max	2547.000000	100.000000	1139.000000	181.000000	537.000000	627.000000	185.000000	783.(

  - A reduction in features was required as field such as 'property_id' was not required.

2 rows × 42 columns

In [13]:
```
# Selecting columns as per index
data = df.iloc[:,[0,2,3,4,13,14,15,16,17,18]]
```

Step 4: Data interrogation

- Café dataset:
  - Checking for Null values – none present

In [9]:
```
# Checking for Null values
df.isna().sum()
```

Out[9]:
```
census_year                    0
block_id                       0
property_id                    0
base_property_id               0
building_address               0
clue_small_area                0
trading_name                   0
business_address               0
industry_anzsic4_code          0
industry_anzsic4_description   0
seating_type                   0
number_of_seats                0
longitude                      0
latitude                       0
location                       0
dtype: int64
```

  - Checked over for duplicates: Initially found a few duplicates. Checked over the data and duplicates occurred due to various reasons. Some businesses would have two locations close to each other such as subway and some locations were big enough to have two or more addresses.

- The step is to check duplicates in regards to seating_type

In [16]:
```
check_seating = df[bool_series].loc[:,['trading_name','seating_type']].sort_values(by=['
```

In [17]:
```
check_seating.head(5)
```

Out[17]:

trading_name	seating_type
11 Inch Pizza	2
127 Cafe Go	2
14 Days Of Cheese	2
18 Pence Lane Coffee & Food	2
1932 Cafe & Restaurant	2

In [19]:
```
# Checking whether there are seating_type greater than 2
check_seating.loc[check_seating['seating_type']>2]
```

Out[19]:

trading_name	seating_type
A Treat of France	4
Assembly Store	4
Benny's Bakery Cafe	4
Bluebag	4
Cafenatics	6
Degani Bakery Cafe	6
Earl Canteen	4
Goz City	4
Grill'd	4
In A Rush Espresso	4

In [20]:
```python
# Checking 'Universal Restaurant'
df.loc[df['trading_name']=="Universal Restaurant"]
```

Out[20]:

	census_year	block_id	property_id	base_property_id	building_address	clue_small_area	trading_name	business_addres
516	2022	254	106090	106090	135-137 Lygon Street CARLTON VIC 3053	Carlton	Universal Restaurant	135-137 Lygo Street CARLTO VIC 305
517	2022	254	106090	106090	135-137 Lygon Street CARLTON VIC 3053	Carlton	Universal Restaurant	135-137 Lygo Street CARLTO VIC 305
1857	2022	254	106091	106091	139-141 Lygon Street CARLTON VIC 3053	Carlton	Universal Restaurant	139-141 Lygo Street CARLTO VIC 305
2085	2022	254	106091	106091	139-141 Lygon Street CARLTON VIC 3053	Carlton	Universal Restaurant	139-141 Lygo Street CARLTO VIC 305

In [22]:
```python
# Checking 'Subway'
df.loc[df['trading_name']=="Subway"].head(5)
```

Out[22]:

	census_year	block_id	property_id	base_property_id	building_address	clue_small_area	trading_name	business_address
101	2022	51	105301	105301	175-177 King Street MELBOURNE VIC 3000	Melbourne (CBD)	Subway	Part Ground 175-177 King Street MELBOURNE VIC ...
123	2022	54	103180	103180	187-193 Elizabeth Street MELBOURNE VIC 3000	Melbourne (CBD)	Subway	Part Ground 187-193 Elizabeth Street MELBOURNE...
407	2022	114	562692	562692	465 Elizabeth Street MELBOURNE VIC 3000	Melbourne (CBD)	Subway	Ground 465 Elizabeth Street MELBOURNE VIC 3000

- Dwelling dataset:
  - During the interrogation of the dataset, multiple 'year_completed' feature was missing. This was due to uncompleted projects. The null value were dropped as they do not bring value to this project.

In [7]:
```python
# Checking for Null values
df.isna().sum()
```

Out[7]:
```
data_format                   0
development_key               0
status                        0
year_completed              377
clue_small_area               0
clue_block                    0
street_address                0
property_id                   0
property_id_2              1239
property_id_3              1367
property_id_4              1393
property_id_5              1404
floors_above                  0
resi_dwellings                0
studio_dwe                    0
one_bdrm_dwe                  0
two_bdrm_dwe                  0
three_bdrm_dwe                0
student_apartments            0
student_beds                  0
student_accommodation_units   0
institutional_accom_beds      0
hotel_rooms                   0
serviced_apartments           0
hotels_serviced_apartments    0
hostel_beds                   0
childcare_places              0
office_flr                    0
retail_flr                    0
industrial_flr                0
storage_flr                   0
education_flr                 0
hospital_flr                  0
recreation_flr                0
```

Step 5: Understanding the data and insights.

- **Café dataset:** The café dataset provided insight about the number of seats per café along with the name, and longitude and latitude. The dataset provided both nominal and ordinal data in the form of suburb name, industry, and number of seats respectively.
  From the dataset, the number of seats were grouped by suburb and industry type providing the total number of seats per industry in each suburb.

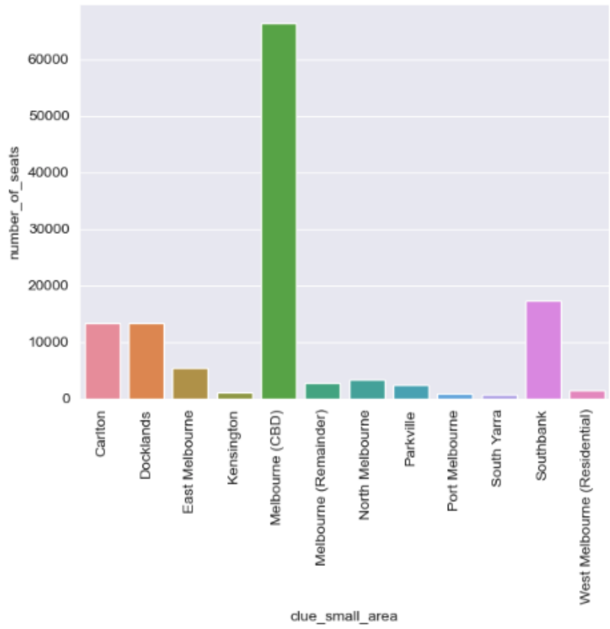
In [4]:
```
final = (data.loc[data['industry_anzsic4_description']=='Cafes and Restaurants']
         .groupby(['clue_small_area','industry_anzsic4_description'])
         .agg({'number_of_seats':'sum','longitude':'mean','latitude':'mean'})
         .reset_index()
        )

```

In [5]:
```
final.head(10)
```

Out[5]:

	clue_small_area	industry_anzsic4_description	number_of_seats	longitude	latitude
0	Carlton	Cafes and Restaurants	13294	144.966085	-37.800992
1	Docklands	Cafes and Restaurants	13407	144.945594	-37.818142
2	East Melbourne	Cafes and Restaurants	5429	144.981429	-37.812974
3	Kensington	Cafes and Restaurants	1041	144.927706	-37.791884
4	Melbourne (CBD)	Cafes and Restaurants	66452	144.963550	-37.813519
5	Melbourne (Remainder)	Cafes and Restaurants	2765	144.979236	-37.839969
6	North Melbourne	Cafes and Restaurants	3247	144.951079	-37.802489
7	Parkville	Cafes and Restaurants	2340	144.954590	-37.791788
8	Port Melbourne	Cafes and Restaurants	935	144.916663	-37.825455
9	South Yarra	Cafes and Restaurants	682	144.981562	-37.834691

  - **Visualisation:** The visualisation provided the total number of Café seats per suburb with additional information about the percentage of Café seats per suburb.




```
The Percentage of cafe in each area:

               clue_small_area  Percentage %
9                  South Yarra          0.53
8               Port Melbourne          0.73
3                   Kensington          0.81
11 West Melbourne (Residential)         1.06
7                    Parkville          1.83
5        Melbourne (Remainder)          2.16
6              North Melbourne          2.53
2               East Melbourne          4.24
0                      Carlton         10.37
1                    Docklands         10.46
10                   Southbank         13.42
4              Melbourne (CBD)         51.85
```

- **The Dwelling dataset:** The Dwelling dataset provided insights about the number of different dwellings in a particular area per year. The nominal data were the suburb area and dwelling type. The ordinal values were the amount of dwelling, year of each dwelling. This provided an opportunity to group the dataset per suburb and year resulting in the total of dwelling per year.

In [10]:
```
dwelling= (dataD.loc[dataD['year'].dt.year== 2022]).reset_index()
```

In [11]:
```
dwelling.head(10)
```

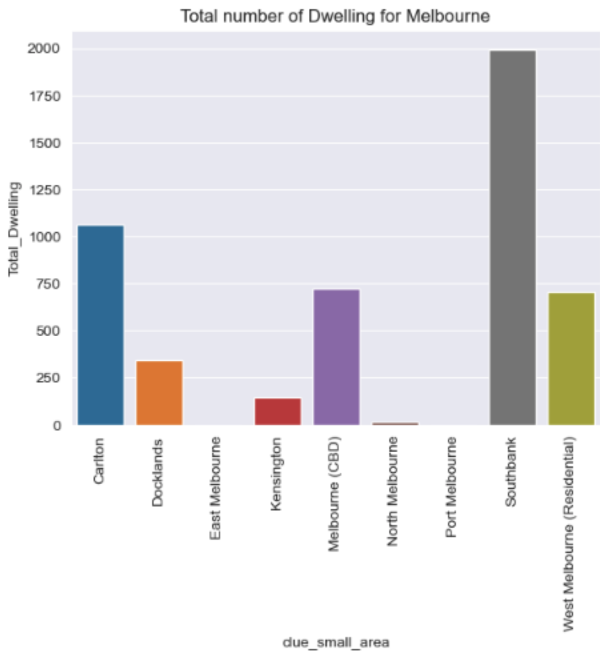
Out[11]:

	index	clue_small_area	year	Total_Dwelling
0	20	Carlton	2022-01-01	1065
1	41	Docklands	2022-01-01	344
2	55	East Melbourne	2022-01-01	0
3	74	Kensington	2022-01-01	146
4	96	Melbourne (CBD)	2022-01-01	724
5	127	North Melbourne	2022-01-01	16
6	164	Port Melbourne	2022-01-01	0
7	190	Southbank	2022-01-01	1994
8	218	West Melbourne (Residential)	2022-01-01	704

-

  - **Visualisation:** The visual presented the new dwellings for each suburb in Melbourne along with the percentage of new dwelling per suburb for 2022.

In [14]:
```
graph_dwelling(dwelling)
```




```
The Percentage of growth per suburb is displaying below:

              clue_small_area  Percentage %
7                   Southbank         39.94
0                     Carlton         21.33
4             Melbourne (CBD)         14.50
8  West Melbourne (Residential)       14.10
1                   Docklands          6.89
3                  Kensington          2.92
5             North Melbourne          0.32
2              East Melbourne          0.00
6              Port Melbourne          0.00
```

- **New dataset from merging both datasets:**
  - Merging the two datasets provided the ration of café seats available per new dwelling. This provides an opportunity to identify the area of potential which produces a small ration. The new information is then used on the map to identify the area of growth.

```
In [23]: final.sort_values('seat/Dwelling',ascending = True).head(10)
```

Out[23]:

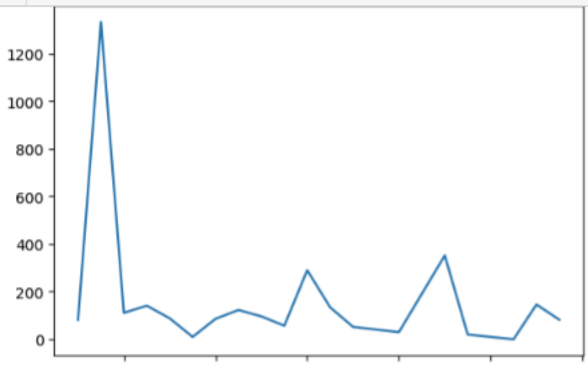
	clue_small_area	industry_anzsic4_description	number_of_seats	longitude	latitude	Percentage %	seat/Dwelling
8	West Melbourne (Residential)	Cafes and Restaurants	1360	144.950661	-37.807716	1.06	1.93
3	Kensington	Cafes and Restaurants	1041	144.927706	-37.791884	0.81	7.13
7	Southbank	Cafes and Restaurants	17198	144.961741	-37.823605	13.42	8.62
0	Carlton	Cafes and Restaurants	13294	144.966085	-37.800992	10.37	12.48
1	Docklands	Cafes and Restaurants	13407	144.945594	-37.818142	10.46	38.97
4	Melbourne (CBD)	Cafes and Restaurants	66452	144.963550	-37.813519	51.85	91.78
5	North Melbourne	Cafes and Restaurants	3247	144.951079	-37.802489	2.53	202.94
2	East Melbourne	Cafes and Restaurants	5429	144.981429	-37.812974	4.24	inf
6	Port Melbourne	Cafes and Restaurants	935	144.916663	-37.825455	0.73	inf

Step 6: Predictive modelling for growth.

- **ARIMA:** Arima is used from predictive timeseries modelling. The model takes in the data from the Dwelling dataset and predict the growth of the selected suburb.

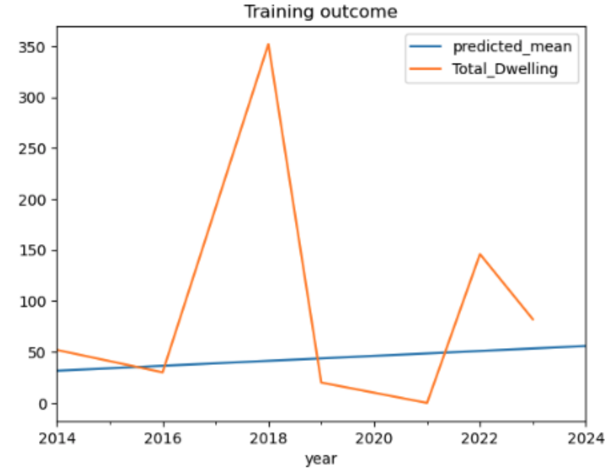
  - **Kensington:**
    - Snap shot of the suburb of Kensington

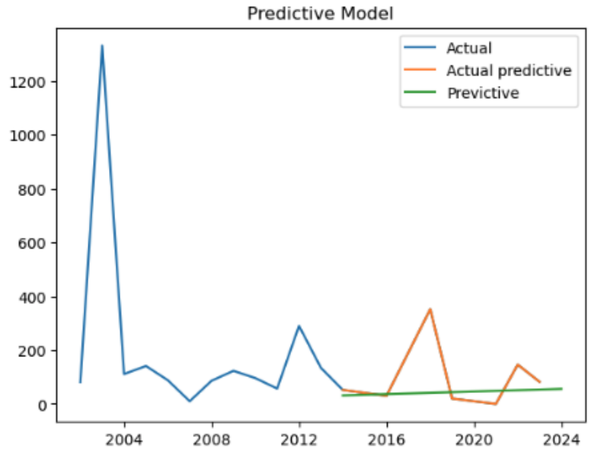
```
In [164]: dataK = setting_data(data, 'Kensington')
```



- Predictive modelling outcome using ARIMA:

```
In [167]:  1 arima_model(dataK,(0,0,0),4,80,'t',-7)

2014-01-01    31.563059
2015-01-01    33.990987
2016-01-01    36.418915
2017-01-01    38.846842
2018-01-01    41.274770
2019-01-01    43.702698
2020-01-01    46.130625
2021-01-01    48.558553
2022-01-01    50.986481
2023-01-01    53.414408
2024-01-01    55.842336
Freq: AS-JAN, Name: predicted_mean, dtype: float64
```

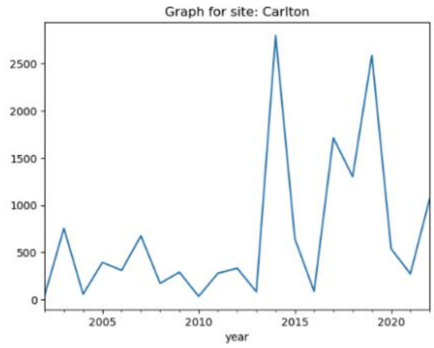




- **Carlton:**
  - Snapshot of the suburb of Carlton

Modelling Carlton

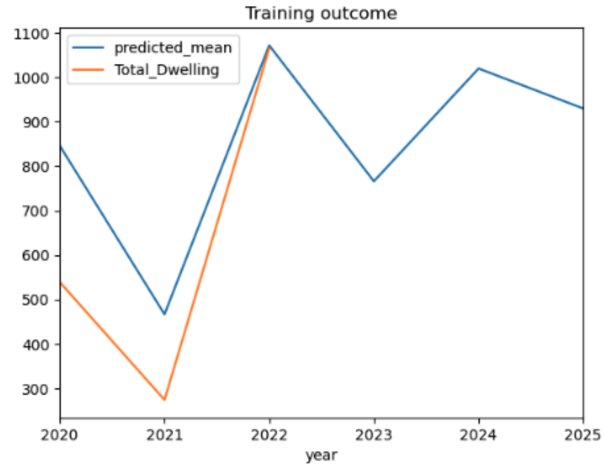
```
]:  1  dataW = setting_data(data, 'Carlton')
```

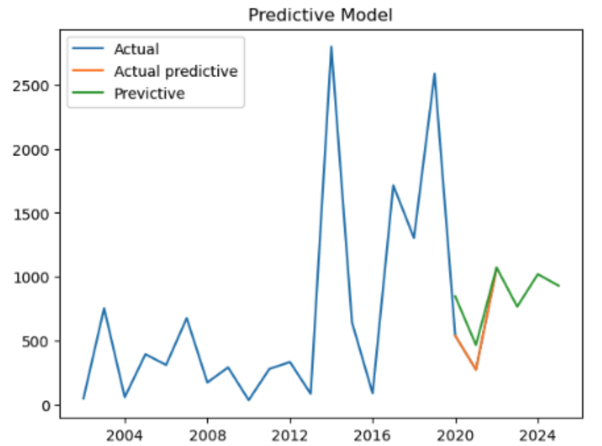


  - Predictive modelling outcome using ARIMA:

```
In [229]:  1  arima_model(dataW,(1,0,2),3,10,'t',-3)
```

```
2020-01-01     845.847841
2021-01-01     466.617584
2022-01-01    1071.827213
2023-01-01     765.957341
2024-01-01    1019.720530
2025-01-01     929.727590
Freq: AS-JAN, Name: predicted_mean, dtype: float64
```



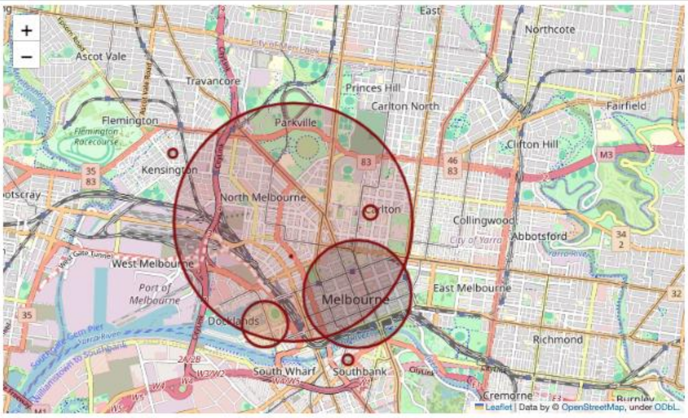


Visualisations:

Three different visualisation is provided:

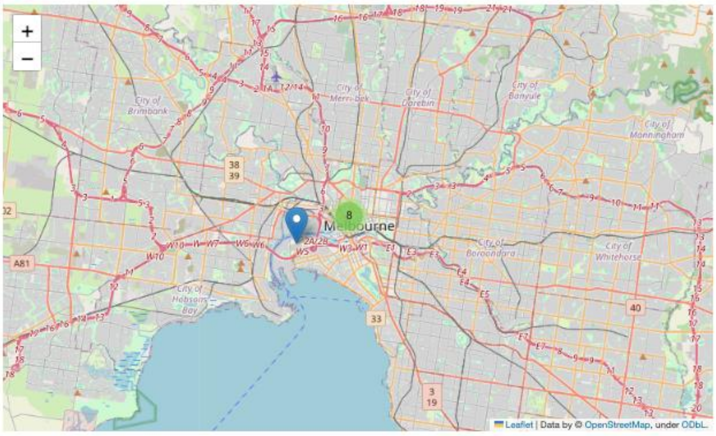
- A quick overview of the map with circles representing the amount of Café seats per Dwelling.

Out[24]:



- A collapsible map with the pin locations showing the amount of Café seats per Dwelling.

Out[25]:

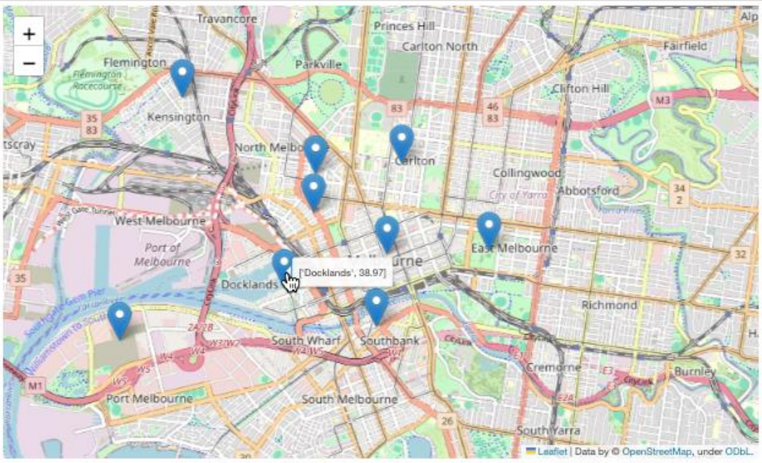


- Final visualisation uses markers on the map with the information of the Café seats per Dwelling.

In [26]:

```python
f =folium.Map([-37.8105, 144.9624], zoom_start=13)
for index, row in final.iterrows():
    folium.Marker([row['latitude'], row['longitude']],
                  popup=row['clue_small_area'],
                  tooltip=[row['clue_small_area'], row['seat/Dwelling']]
             ).add_to(f)


f
```

Out[26]:



Bibliography

IBISWorld. (2023). *Cafes and Coffee Shops in Australia.* IBISWorld.

Code references:

Folium:

https://www.linkedin.com/pulse/mapping-australian-geograph-data-python-dilan-jayasekara/

https://python-graph-gallery.com/312-add-markers-on-folium-map/

Arima modelling:

https://www.youtube.com/watch?v=8FCDpFhd1zk&t=419s

https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

API code:

API Tutorial.pdf: from MS teams, shared by ; 'Harley NGO'