Coursera Capstone Project

Applied Data Science

Project Title: Analysing Melbourne Housing Market

By: Zin Myint Naung

1. Introduction

Melbourne topped the “Global Liveability Index” list for seven years from 2011 to 2017. Although it was beaten by Vienna in 2018, it still maintains in top three with a score of 98.4 in 2019. Moreover, it achieved scores of 100 in healthcare, education and infrastructure. Melbourne is often considered as Australia’s cultural capital with its thriving arts scene. With city population reaching to almost 5millions in 2019, the demand for owning a house in Melbourne has become a challenge to many new buyers. In this project, we will be analysing Melbourne’s housing market in order to highlight our findings so that it can help both potential buyers and real estate agencies while making their decision.

1.1. Business Problem

It is common that most of the liveable cities attract people from different background, such as investors, property builders, skilled migrants, etc. Therefore, Melbourne’s housing market becomes one of the most attractive sectors in Australia property market. However, there are some challenges such as knowledge of housing prices in Melbourne. In order to address such challenges, this project determines to provide necessary information regarding housing prices of Melbourne by analysing housing market data.

1.2 Target Audiences

With analysed housing market data, real estate agents can take full advantage while suggesting good buying options to their potential buyers. Housing developers can easily find hotspot location and able to estimate turnover per units. As for individual family, this data provide how much budget they should set to get their dream home in Melbourne. In order to fulfil Australians’ dream, this project utilise location data from Foursquare API to compare housing prices between different localities.

2. Data

In order to analyse, we need to define our data set for Melbourne's housing price. The data set needs to have following columns:

- Suburb (List of neighbourhoods)
- Price (Housing prices, to calculate average price for each neighbourhoods)
- Latitude (To visualize suburb location)
- Longitude (To visualize suburb location)
- Nearby Venues (To fetch using Foursquare API)

2.1 Suburb Data

Melbourne's housing price list can be downloaded from Kaggle's open datasets. This dataset save our time from scraping suburban data from other third party website such as Wikipedia. The dataset also contains latitude and longitude values for further analysis. The URL for the data is as shown below:

https://www.kaggle.com/anthonypino/melbourne-housing-market#Melbourne_housing_FULL.csv

The csv file includes total of 21 columns with sold price for each address in a suburb. There are total of 34,857 records in the data set.

In [9]:
```
#CSV from https://www.kaggle.com/anthonypino/melbourne-housing-market#Melbourne_housing_FULL.csv
df = pd.read_csv('Melbourne_housing_FULL.csv')
df.head()
```

Out[9]:

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	Bedroom2	Bathroom	Car	Landsize	BuildingArea	Yea
0	Abbotsford	68 Studley St	2	h	nan	SS	Jellis	3/09/2016	2.50	3067.00	2.00	1.00	1.00	126.00	nan	
1	Abbotsford	85 Turner St	2	h	1480000.00	S	Biggin	3/12/2016	2.50	3067.00	2.00	1.00	1.00	202.00	nan	
2	Abbotsford	25 Bloomburg St	2	h	1035000.00	S	Biggin	4/02/2016	2.50	3067.00	2.00	1.00	0.00	156.00	79.00	19
3	Abbotsford	18/659 Victoria St	3	u	nan	VB	Rounds	4/02/2016	2.50	3067.00	3.00	2.00	1.00	0.00	nan	
4	Abbotsford	5 Charles St	3	h	1465000.00	SP	Biggin	4/03/2017	2.50	3067.00	3.00	2.00	0.00	134.00	150.00	19

In [10]:
```
df.shape
```

Out[10]: (34857, 21)

2.2 Data Wrangling

There are null values in our price columns. In the case of less data set, we should fill those values by taking assumption such as getting average values, etc. However, since we have large dataset, we would simply drop the null values records. Also drop the columns that are not required for our purpose. Therefore, our data frame should look something like below.

In [17]:
```
df = df[['Suburb', 'Price', 'Postcode', 'Lattitude', 'Longtitude']]
df['Neighbourhood'] = df[['Suburb']]
df.head()
```

Out[17]:

	Suburb	Price	Postcode	Lattitude	Longtitude	Neighbourhood
0	Abbotsford	nan	3067.00	-37.80	145.00	Abbotsford
1	Abbotsford	1480000.00	3067.00	-37.80	145.00	Abbotsford
2	Abbotsford	1035000.00	3067.00	-37.81	144.99	Abbotsford
3	Abbotsford	nan	3067.00	-37.81	145.01	Abbotsford
4	Abbotsford	1465000.00	3067.00	-37.81	144.99	Abbotsford

In [18]:
```
df = df.dropna()
df.shape
```

Out[18]: (20993, 6)

2.3 Calculating *mean (μ)* value

We will calculate the mean value for each suburb using *pandas groupby()* method. Then re-create our data frame using python list as shown below.

In [19]:
```
df.reset_index(drop=True)
df = pd.DataFrame(df.groupby(['Suburb'], sort = False).mean())
df.head()
```

Out[19]:

Suburb	Price	Postcode	Lattitude	Longtitude
Abbotsford	1096603.90	3067.00	-37.80	145.00
Airport West	780529.42	3042.00	-37.72	144.88
Albert Park	1983664.71	3206.00	-37.84	144.95
Alphington	1441155.56	3078.00	-37.78	145.03
Altona	872917.93	3018.00	-37.87	144.82

In [20]:
```
df.shape
```

Out[20]: (338, 4)

In [21]:
```
ls = []

for index, row in df.iterrows():
    ls.append((index, row['Postcode'], row['Price'], row['Lattitude'], row['Longtitude']))
data = pd.DataFrame(ls, columns = ('Suburb', 'Postcode', 'AvgPrice', 'Lattitude', 'Longtitude'))
data.head()
```

Out[21]:

	Suburb	Postcode	AvgPrice	Lattitude	Longtitude
0	Abbotsford	3067.00	1096603.90	-37.80	145.00
1	Airport West	3042.00	780529.42	-37.72	144.88
2	Albert Park	3206.00	1983664.71	-37.84	144.95
3	Alphington	3078.00	1441155.56	-37.78	145.03
4	Altona	3018.00	872917.93	-37.87	144.82

3. Methodology

As we have prepared our data in previous data wrangling stage, next step is to define our methodology. To solve our business problem, methodology can be broken down in three phases. Firstly, we are required to visualize our data, i.e. to plot housing price data on the map of Melbourne. This will give us better understanding on how spread the data is and how much it covers for the city of Melbourne. Secondly, we will fetch nearby venues data using Foursquare API so that we can group the most frequent venues. Finally, we will cluster our data set based on the most frequent venues and merge with our initial Melbourne housing data set. In this way, we can examine the average housing price for each cluster according to the nearby venues.

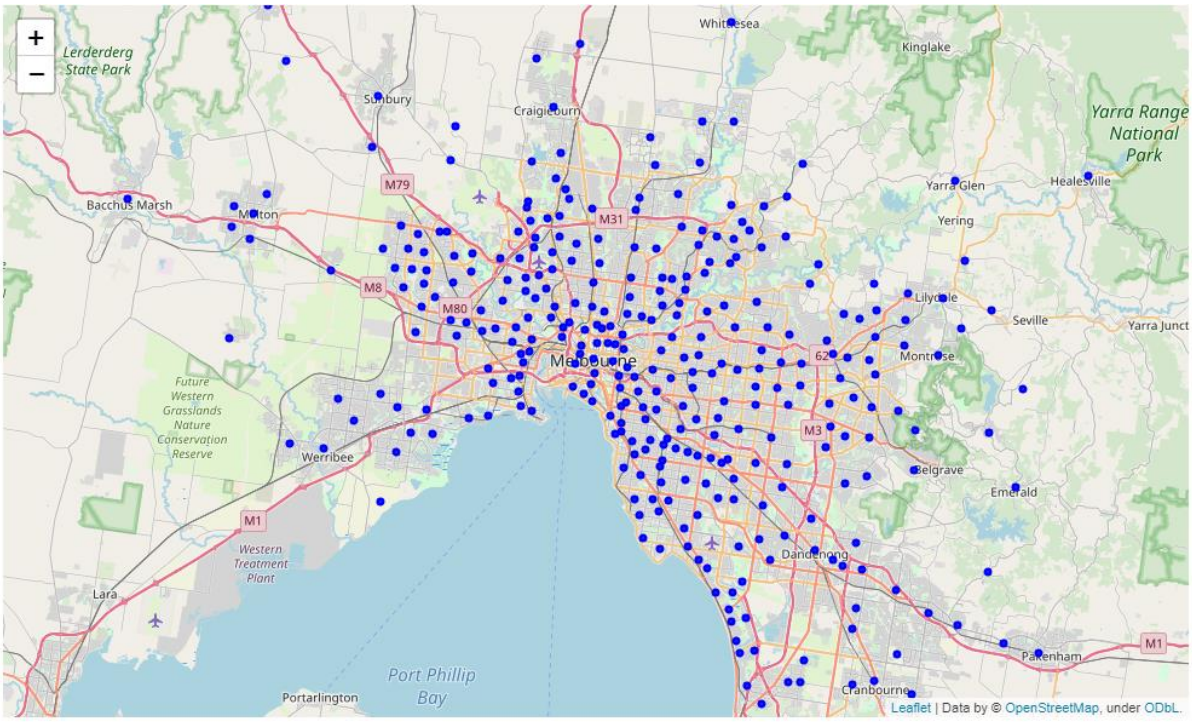
3.1 Visualizing Housing Price of Melbourne

To visualize our data set on the map of Melbourne, we will use *Nominatim*, a search engine for *OpenStreetMap* data to fetch the latitude and longitude data of Melbourne city. Then we will use *Folium* library to draw the map and add our data as markers on the map. The final code for this step should look like as shown below:-

In [7]:
```
address = 'Melbourne, Victoria'

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))
```

```
The geograpical coordinate of Melbourne are -37.8142176, 144.9631608.
```

In [8]:
```
# create map of Melbourne using latitude and longitude values
map_melbourne = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, suburb in zip(data['Lattitude'], data['Longtitude'], data['Suburb']):
    label = '{}, {}'.format(data, suburb)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=2,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_melbourne)

map_melbourne
```



3.2 Fetching Nearby Venues

Housing prices are strongly influenced by its locations. Therefore, it is crucial to examine pricing based on where the property is located. To serve this purpose, we will use Foursquare API to fetch nearby venues of all the suburbs in our data set. Below display our venues data set after consuming Foursquare API:-

Out[19]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Abbotsford	-37.80	145.00	Three Bags Full	-37.81	145.00	Café
1	Abbotsford	-37.80	145.00	The Kitchen at Weylandts	-37.81	145.00	Café
2	Abbotsford	-37.80	145.00	Retreat Hotel	-37.80	145.00	Pub
3	Abbotsford	-37.80	145.00	Laird Hotel	-37.81	144.99	Gay Bar
4	Abbotsford	-37.80	145.00	Salvos Store	-37.81	145.00	Thrift / Vintage Store

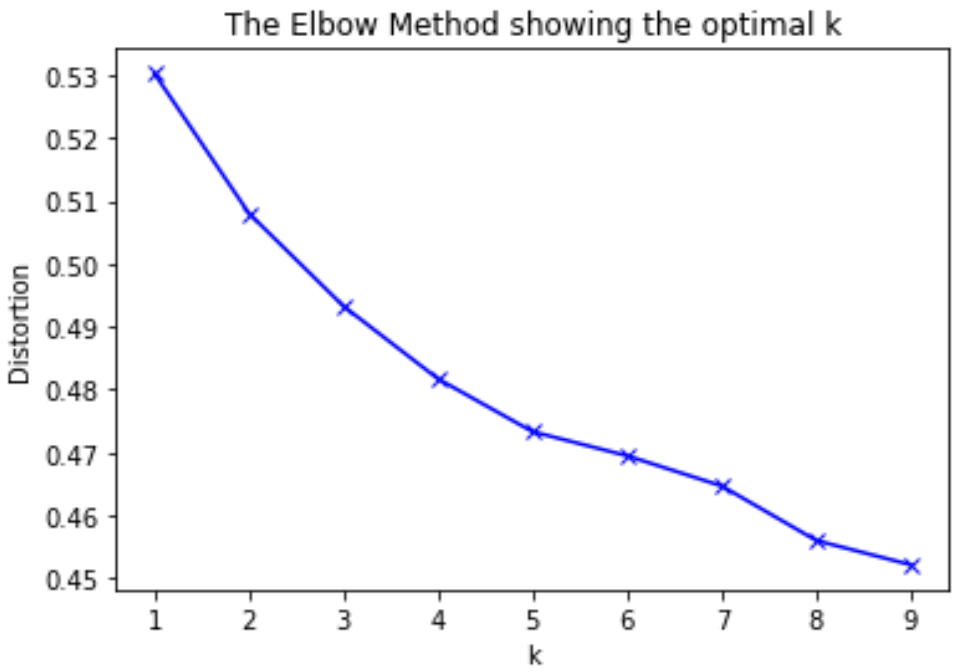
In [20]:
```
melbourne_venues.groupby('Neighborhood').count()
```

Out[20]:

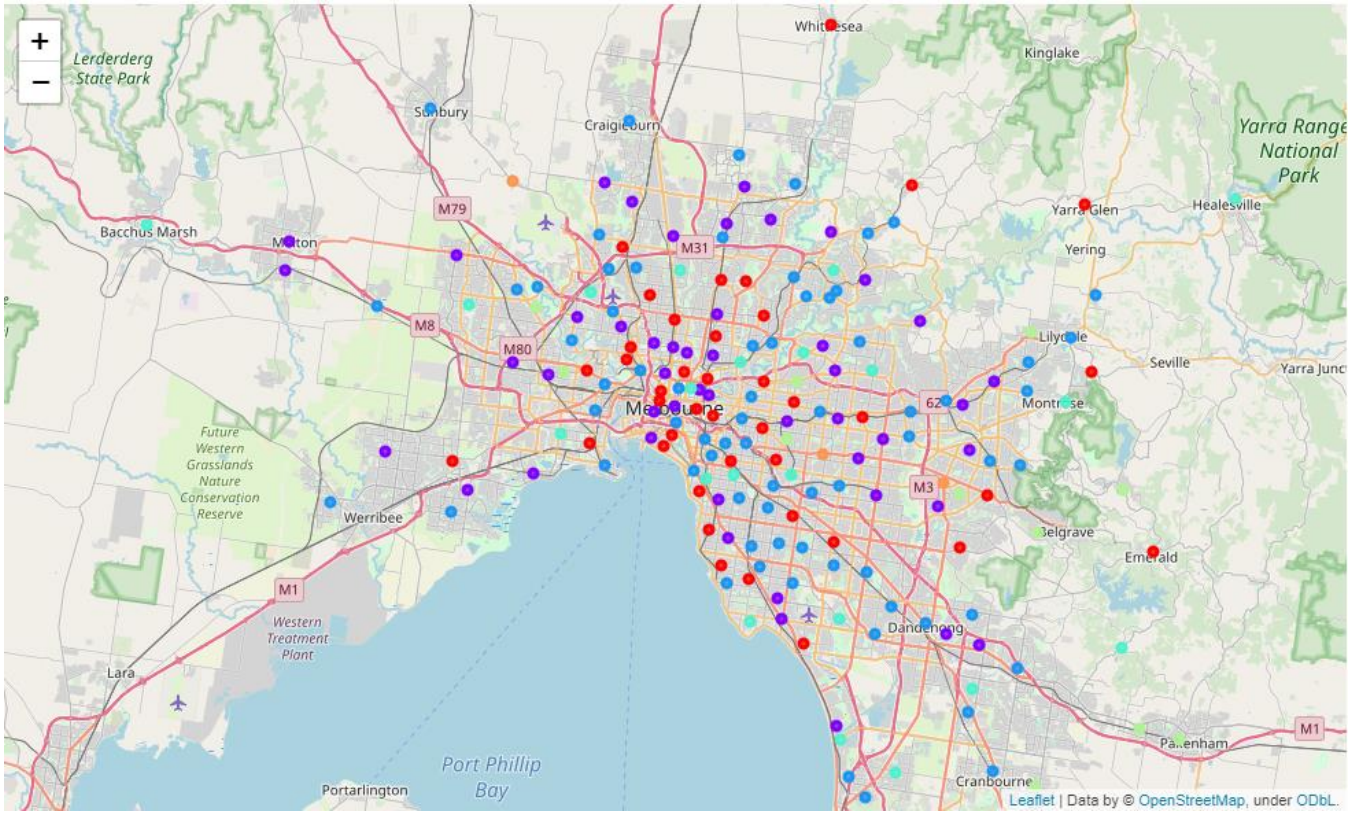
Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Abbotsford	18	18	18	18	18	18
Aberfeldie	2	2	2	2	2	2
Airport West	3	3	3	3	3	3
Albanvale	1	1	1	1	1	1
Albert Park	30	30	30	30	30	30
Albion	1	1	1	1	1	1
Alphington	8	8	8	8	8	8
Altona	1	1	1	1	1	1
Altona North	7	7	7	7	7	7
Ardeer	1	1	1	1	1	1

3.3 Clustering

In this section, we will use K-means algorithm to cluster nearby venues data set. Therefore, we will use elbow method to find the best value for k.



As shown above, the best value for k can be determined as 6. Below map displays how our cluster spread on the map of Melbourne city.

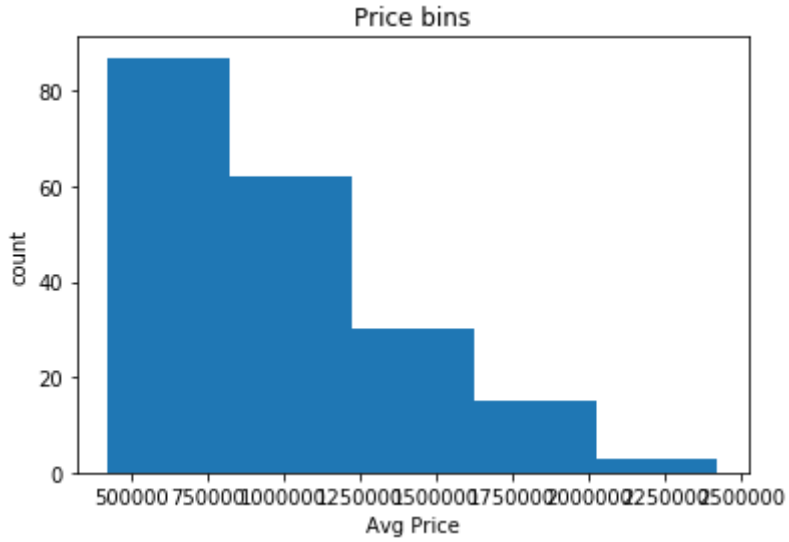


3.4 Binning for Average Price

In this section, we will set average housing price with five bins. We will use *matplotlib* histogram to visualize each bin. Approximately, these five bins will be classified as follows:-

1. Low Level : < 800,000
2. Below Average : 800,000 – 1,200,000
3. Average : 1,200,000 – 1,600,000
4. Above Average : 1,600,000 – 2,000,000
5. High Level : > 2,000,000

```
In [78]: %matplotlib inline
import matplotlib as plt
from matplotlib import pyplot
plt.pyplot.hist(melbourne_merged["AvgPrice"],bins=5)

# set x/y labels and plot title
plt.pyplot.xlabel("Avg Price")
plt.pyplot.ylabel("count")
plt.pyplot.title("Price bins")
```



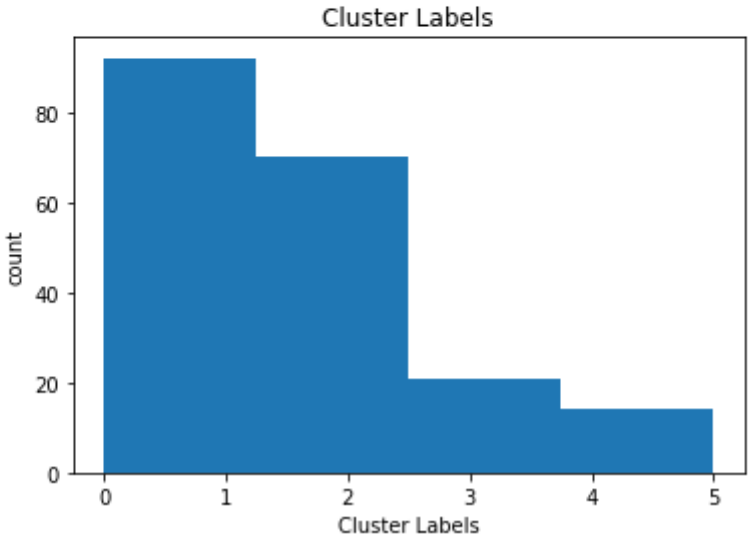
3.5 Binning for Cluster Labels

For nearby venues, we will create four bins and label them as follows:-

1. Café
2. Mixed Social Venues
3. Bars, Store and Restaurant
4. Shops
5. Parks and places

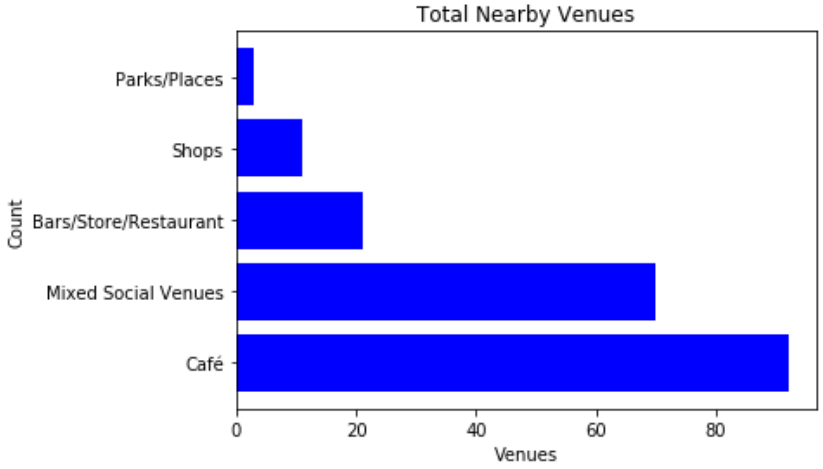
In [82]:
```
%matplotlib inline
import matplotlib as plt
from matplotlib import pyplot
plt.pyplot.hist(melbourne_merged["Cluster Labels"],bins=4)

# set x/y labels and plot title
plt.pyplot.xlabel("Cluster Labels")
plt.pyplot.ylabel("count")
plt.pyplot.title("Cluster Labels")
```

Out[82]: Text(0.5,1,'Cluster Labels')



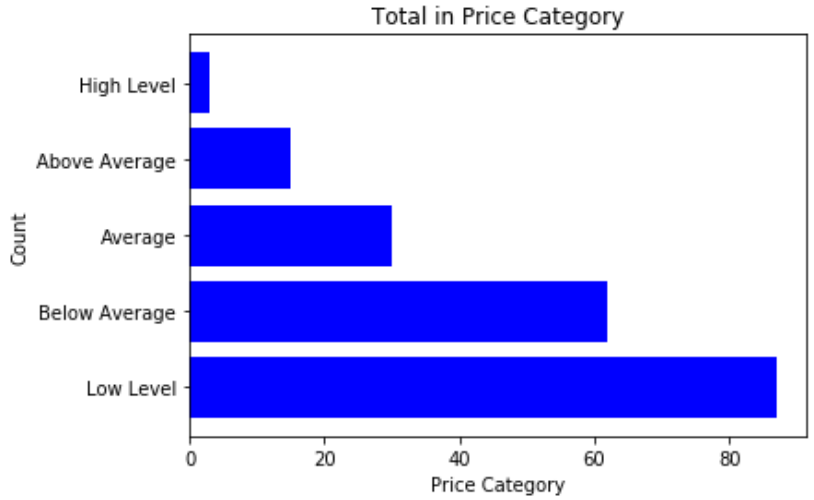
4. Result & Findings

In the previous chapter, we have discussed how our methodology section utilized our data set to solve our business problem. In this chapter, we will examine our result data set with visualization. First of all, let's explore the distribution of nearby venues.



	Total Nearby Venues
Café	92
Mixed Social Venues	70
Bars/Store/Restaurant	21
Shops	11
Parks/Places	3

It is clear that majority of venues fall under *Café, Mixed social venues* and *restaurants* category. Next, we will explore the distribution of housing prices.



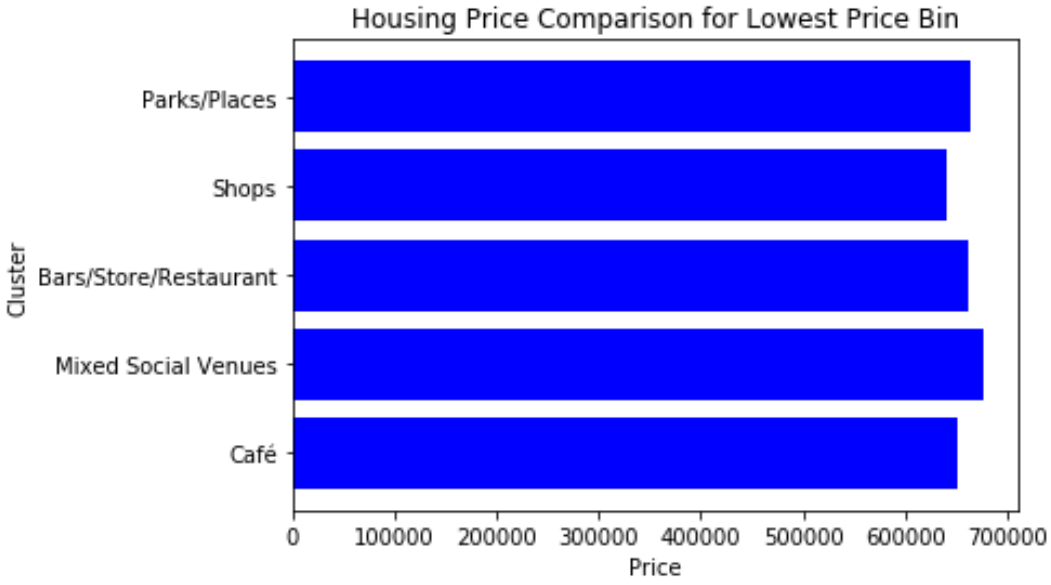
	Total in Price Category
Low Level	87
Below Average	62
Average	30
Above Average	15
High Level	3

With above histogram, it can be seen that majority of housing prices fall under *low level*, *below average* and *average* price category.

Finally, we would like to know average housing price for each price category for each individual venues.

	Café	Mixed Social Venues	Bars/Store/Restaurant	Shops	Parks/Places
Low Level	650443.98	676052.28	662906.31	640540.97	665000.00
Below Average	1017134.57	962482.67	984402.94	885100.00	1078272.75
Average	1393744.33	1390169.05	1369113.25	nan	1308933.03
Above Average	1801127.71	1765751.65	1747225.83	1858816.27	nan
High Level	2183091.25	2423333.33	nan	nan	nan

With above data table, we can easily suggest how much budget is needed to buy a house in Melbourne that is near to a Café or Bars or Shops. The cheapest housing prices are near either Café or Shops whereby most expensive housing are near Parks or Mixed Social Venues.



Above histogram displayed the first row of our data frame to compare which venues has the lowest average housing price inside “Lowest Price” bin. The answer is to look for a house near “Shops” venue. We can apply similar visualization to answer different questions regarding Melbourne housing price.

5. Conclusion

With the help of machine learning, namely unsupervised machine learning algorithm K-mean, we can cluster venues data and analyse the average housing price for each venue. For a real estate agent, he or she can make use of this data to suggest a buyer on how much money they may need to spend to buy a house. For individual family, this data allows further financial planning to fulfil Australians' dream. And even for property builders, they will know which spot is profitable to build a new lodging.

6. Recommendation

This notebook can be extended to further data science project such as predicting housing price in a Suburb where the house is located near a park, etc.

References

Melbourne Housing Price data set from Kaggle, https://www.kaggle.com/anthonypino/melbourne-housing-market#Melbourne_housing_FULL.csv