Capstone Final Project

Clustering of Melbourne Suburbs

Yolanda G.

Background

- Melbourne - most liveable city.
- Convenient suburbs with cafes, restaurants, gyms, banks, post offices, supermarkets around.
- Which suburbs are similar to mine?

Business Problem

- Given the input of my interested suburbs (a pre-defined list), I need to cluster them based on the popular facility types of each suburb.
- And the cluster with my current suburb in would be my top interest list.

Out[8]:

	venue_id	lat	lng	category	suburb	details
0	548e129b498e3012587f3e5c	-37.810986	144.964059	Little Rogue Coffee	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d1e0931735', 'name': 'C...
1	552db1fd498ef6abfdb29b4c	-37.811320	144.966155	Boilermaker House	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d122941735', 'name': 'W...
2	4b058748f964a520cf8822e3	-37.813445	144.962137	Brother Baba Budan	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d1e0931735', 'name': 'C...
3	54631f6e498ed0dde017e53c	-37.813527	144.961978	Tipo 00	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d110941735', 'name': 'I...
4	54b629b2498eb25a34354f3c	-37.811798	144.966687	Union Electric	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d11e941735', 'name': 'C...
...	...	...	...	...	...	...
95	4b56a492f964a520751728e3	-37.824396	144.977303	Tan Track	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d106941735', 'name': 'T...
96	4b14e1f7f964a52039a723e3	-37.799097	144.954388	Lanna Thai	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d149941735', 'name': 'T...
97	4ed710f9a17c6e17b5d3645a	-37.802773	144.947505	Twenty & Six Espresso	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...
98	4b078997f964a5208bfe22e3	-37.829438	144.960265	Dead Man Espresso	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...
99	4b058754f964a5209d8c22e3	-37.832242	144.956573	South Melbourne Market	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d1fa941735', 'name': 'F...

100 rows × 6 columns

Now i want to explore my interested field:

In [9]:
```
dataframe['category'][99]
```
Out[9]: 'South Melbourne Market'

In [10]:
```
dataframe['details'][99][0]['shortName']
```
Out[10]: "Farmer's Market"

Raw Dataframe

- use Foursquare API to explore venues for each suburb
- transform into dataframe
- sort the details into a category list we used to categorise venue

Sorted Dataframe

Out[11]:

	venue_id	lat	lng	category	suburb	details
0	548e129b498e3012587f3e5c	-37.810986	144.964059	Coffee Shop	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d1e0931735', 'name': 'C...
1	552db1fd498ef6abfdb29b4c	-37.811320	144.966155	Whisky Bar	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d122941735', 'name': 'W...
2	4b058748f964a520cf8822e3	-37.813445	144.962137	Coffee Shop	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d1e0931735', 'name': 'C...
3	54631f6e498ed0dde017e53c	-37.813527	144.961978	Italian	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d110941735', 'name': 'I...
4	54b629b2498eb25a34354f3c	-37.811798	144.966687	Cocktail	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d11e941735', 'name': 'C...
...	...	...	...	...	...	...
95	4b56a492f964a520751728e3	-37.824396	144.977303	Track	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d106941735', 'name': 'T...
96	4b14e1f7f964a52039a723e3	-37.799097	144.954388	Thai	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d149941735', 'name': 'T...
97	4ed710f9a17c6e17b5d3645a	-37.802773	144.947505	Café	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...
98	4b078997f964a5208bfe22e3	-37.829438	144.960265	Café	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...
99	4b058754f964a5209d8c22e3	-37.832242	144.956573	Farmer's Market	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d1fa941735', 'name': 'F...

100 rows × 6 columns

Loop Apply to All Suburbs

Out[14]:

	venue_id	lat	lng	category	suburb	details
0	548e129b498e3012587f3e5c	-37.810986	144.964059	Coffee Shop	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d1e0931735', 'name': 'C...
1	552db1fd498ef6abfdb29b4c	-37.811320	144.966155	Whisky Bar	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d122941735', 'name': 'W...
2	4b058748f964a520cf8822e3	-37.813445	144.962137	Coffee Shop	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d1e0931735', 'name': 'C...
3	54631f6e498ed0dde017e53c	-37.813527	144.961978	Italian	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d110941735', 'name': 'I...
4	54b629b2498eb25a34354f3c	-37.811798	144.966687	Cocktail	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d11e941735', 'name': 'C...
...	...	...	...	...	...	...
2471	57aed307cd10e52b3471623c	-37.885230	145.001841	Café	Carnegie, VIC 3163	[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...
2472	4b636fd8f964a520c0792ae3	-37.889264	145.040981	Café	Carnegie, VIC 3163	[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...
2473	4c645d703590d13afbece2bc	-37.877182	145.082746	Gym	Carnegie, VIC 3163	[{'id': '4bf58dd8d48988d176941735', 'name': 'G...
2474	4b760f6af964a520ea392ee3	-37.861732	145.028334	Café	Carnegie, VIC 3163	[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...
2475	4d73ea67170ab1f7976fed94	-37.884406	145.001035	Café	Carnegie, VIC 3163	[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...

2476 rows × 6 columns

```
In [17]: # group rows by suburb and by taking the mean of the frequency of occurrence of each category
df = df.groupby('suburb').mean().reset_index()

df
```

Out[17]:

	suburb	Afghan	African	Apparel	Arcade	Argentinian	Art Gallery	Arts & Crafts	Asian	Athletics & Sports	...	Vietnamese	Warehouse Store	Whisky Bar	Wine Bar	Wine Shop	Xinjia
0	Ascot Vale, VIC 3032	0.000000	0.01	0.01	0.00	0.00	0.01	0.00	0.010000	0.00	...	0.000000	0.00	0.00	0.02	0.00	0.00
1	Ashwood, VIC 3147	0.000000	0.00	0.02	0.00	0.00	0.00	0.01	0.010000	0.00	...	0.000000	0.00	0.00	0.00	0.00	0.00
2	Boronia, VIC 3155	0.000000	0.00	0.00	0.00	0.00	0.00	0.01	0.000000	0.00	...	0.010000	0.00	0.00	0.00	0.00	0.00
3	Brighton, VIC 3186	0.000000	0.00	0.00	0.00	0.00	0.00	0.00	0.000000	0.00	...	0.010309	0.00	0.00	0.00	0.00	0.00
4	Burwood, VIC 3125	0.000000	0.00	0.01	0.00	0.00	0.00	0.01	0.010000	0.00	...	0.010000	0.00	0.00	0.00	0.01	0.00
5	Camberwell, VIC 3124	0.000000	0.00	0.00	0.00	0.00	0.00	0.00	0.010000	0.00	...	0.010000	0.00	0.00	0.00	0.01	0.00
6	Carnegie, VIC 3163	0.000000	0.00	0.02	0.00	0.00	0.00	0.00	0.010000	0.00	...	0.000000	0.00	0.00	0.00	0.00	0.00
7	Caulfield, VIC 3162	0.000000	0.00	0.00	0.00	0.00	0.00	0.00	0.000000	0.00	...	0.010000	0.00	0.00	0.01	0.00	0.00
8	Docklands, VIC 3008	0.000000	0.01	0.00	0.00	0.01	0.01	0.00	0.000000	0.00	...	0.000000	0.00	0.02	0.00	0.00	0.00

Get Dummies and Group Suburbs

dummy variable: unique venue.category

K-means to Cluster

```
In [22]: # set number of clusters
         kclusters = 5

         df = df.drop('suburb', 1)

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

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

Out[22]: array([4, 2, 1, 0, 2, 0, 0, 0, 3, 0], dtype=int32)

In [23]: # add clustering labels
         newdf.insert(0, 'Cluster Labels', kmeans.labels_)

         mergedf = dataframe

         mergedf = mergedf.join(newdf.set_index('suburb'), on='suburb')

         mergedf
```

Out[23]:

	venue_id	lat	lng	category	suburb	details	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Mo Comm Venue
0	548e129b498e3012587f3e5c	-37.810986	144.964059	Coffee Shop	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d1e093 735', 'name': 'C...	3	Café	Coffee Shop	Park	Cockta
1	552db1fd498ef6abfdb29b4c	-37.811320	144.966155	Whisky Bar	Melbourne City, VIC 3000	[{'id': '4bf58dd8d48988d12294 735', 'name': 'W...	3	Café	Coffee Shop	Park	Cockta

Result

In [25]:
```
final0 = final.loc[final['Cluster Labels'] == 0, 'suburb']
final0 = final0.unique()
print(final0)
```

```
['Kew, VIC 3101' 'Toorak, VIC 3142' 'St Kilda, VIC 3182'
 'Camberwell, VIC 3124' 'Malvern, VIC 3144' 'Caulfield, VIC 3162'
 'Brighton, VIC 3186' 'Glen Iris, VIC 3146' 'Carnegie, VIC 3163']
```

In [26]:
```
final1 = final.loc[final['Cluster Labels'] == 1, 'suburb']
final1 = final1.unique()
print(final1)
```

```
['Boronia, VIC 3155' 'Keysborough, VIC 3173' 'Springvale, VIC 3171'
 'Oakleigh, VIC 3166']
```

In [27]:
```
final2 = final.loc[final['Cluster Labels'] == 2, 'suburb']
final2 = final2.unique()
print(final2)
```

```
['Mount Waverley, VIC 3149' 'Ashwood, VIC 3147' 'Burwood, VIC 3125']
```

In [28]:
```
final3 = final.loc[final['Cluster Labels'] == 3, 'suburb']
final3 = final3.unique()
print(final3)
```

```
['Melbourne City, VIC 3000' 'Port Melbourne, VIC 3207'
 'Docklands, VIC 3008' 'Parkville, VIC 3050' 'North Melbourne, VIC 3051'
 'Richmond, VIC 3121' 'South Yarra, VIC 3141']
```

In [29]:
```
final4 = final.loc[final['Cluster Labels'] == 4, 'suburb']
final4 = final4.unique()
print(final4)
```

```
['Moonee Ponds, VIC 3039' 'Ascot Vale, VIC 3032']
```

Discussion & Conclusion

- My current suburb is in 'final0' group, and the other suburbs in the 'final0' group are indeed similar to my current suburb. So it confirms my pre-judgement and now I can make decision with more confidence level.
- Further to this, I think this method can be applied to other cities which I totally have no knowledge about.
- This simple model can cluster different suburbs, so that user can get some brief idea about the similarity of different suburbs.
- This model can be improved later by picking up venues in a more detailed level (dining, school, hospital, etc.), and make it a separate input parameter to the model.