Battle Of Neighbourhoods

Prateek Shamprasad Pawar

Email: prateekspawar@gmail.com

Introduction

- Tom wants to relocate from ‘Parkwoods , Toronto’ to New York.
- Problem is to find similar neighbourhood.
- Factors Identified:

1) Plenty of Shops and Services making daily life easy.

2) A great number of food venues available.

3) Good transportation services.

4) Less Nightlife spots in the neighbourhood.

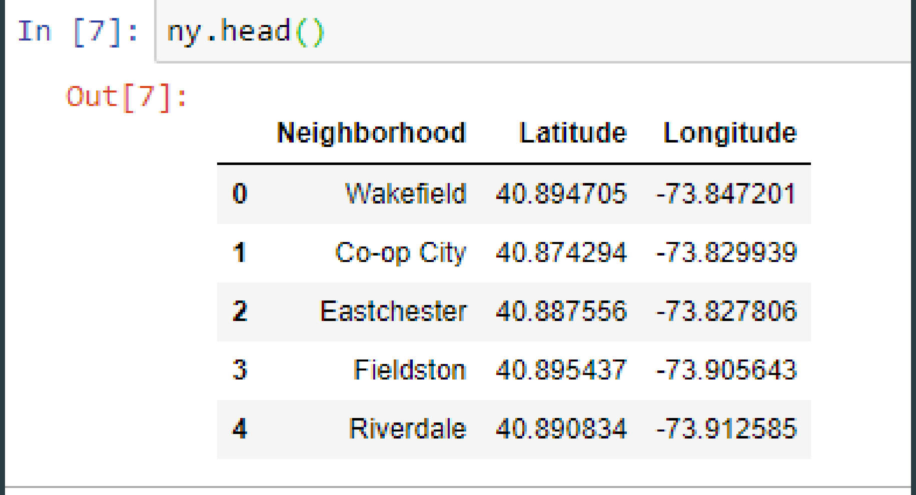
Data (New York)

1)New York Neighbourhoods geospatial data (.json):

https://geo.nyu.edu/catalog/nyu_2451_34572 .

2)Extract data using json.load() .

3) Append Data in ny DataFrame.



```
In [7]: ny.head()
```

```
Out[7]:
```

	Neighborhood	Latitude	Longitude
0	Wakefield	40.894705	-73.847201
1	Co-op City	40.874294	-73.829939
2	Eastchester	40.887556	-73.827806
3	Fieldston	40.895437	-73.905643
4	Riverdale	40.890834	-73.912585

Data (Toronto):

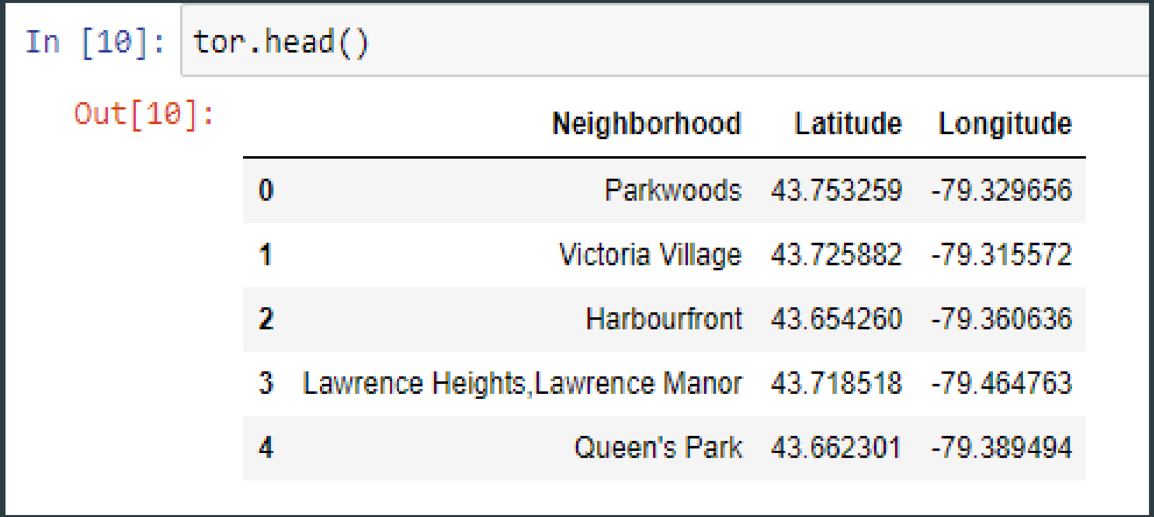
1)Toronto postal code data for neighbourhoods:

https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M .

2)Toronto coordinates of postal codes:

https://cocl.us/Geospatial_data/Geospatial_Coordinates.csv .

3) Combine to get tor DataFrame.

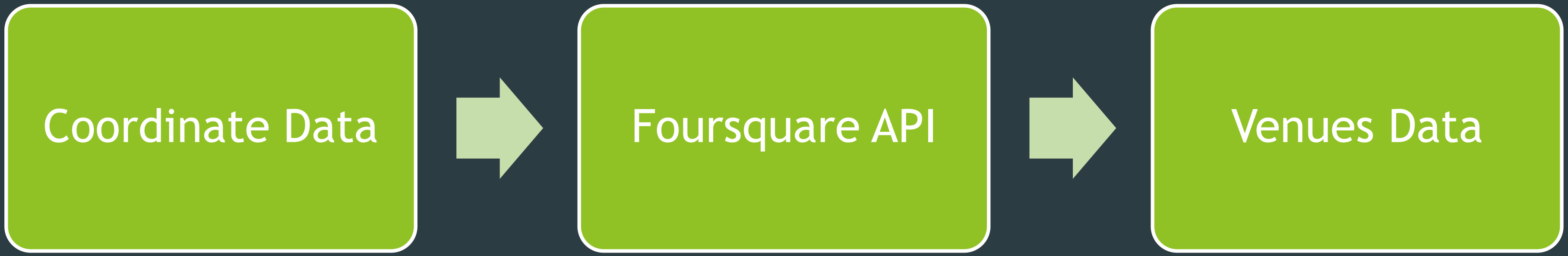


```
In [10]: tor.head()
```

Out[10]:

	Neighborhood	Latitude	Longitude
0	Parkwoods	43.753259	-79.329656
1	Victoria Village	43.725882	-79.315572
2	Harbourfront	43.654260	-79.360636
3	Lawrence Heights,Lawrence Manor	43.718518	-79.464763
4	Queen's Park	43.662301	-79.389494

1. Acquiring Venues Data



Coordinate Data

Foursquare API

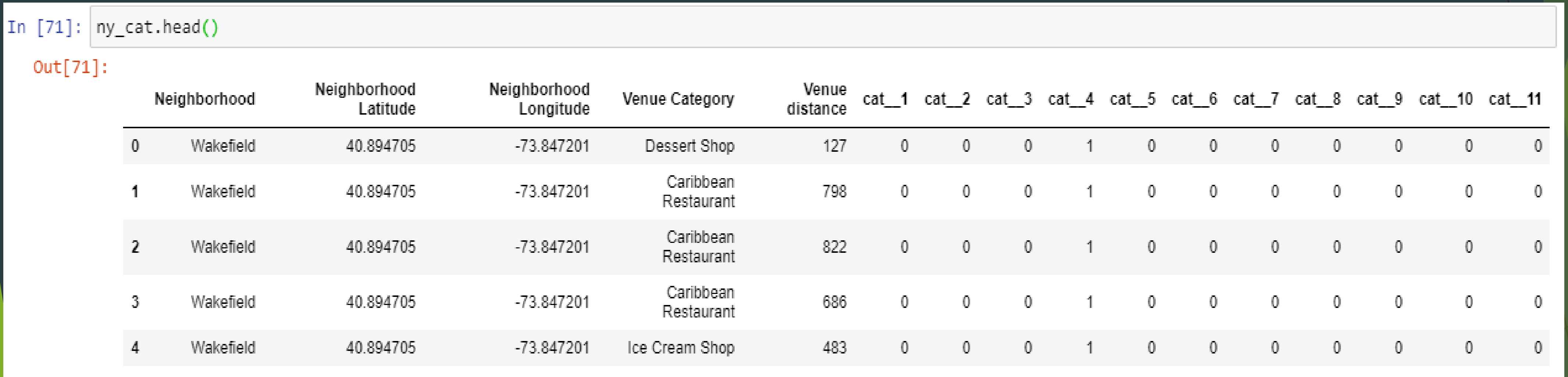
Venues Data

2. Defining Categories:

- 1.Arts & Entertainment
- 2.College & university
- 3.Event
- 4.Food
- 5.Nightlife Spot
- 6.Outdoors & Recreation
- 7.Professional & Other Places
- 8.school
- 9.Residence
- 10.Shop & Service
- 11.Travel & Transport

3. One-Hot Encoding:

- Adding Dummy columns of each category and converting category data to binary.



```
In [71]: ny_cat.head()
```

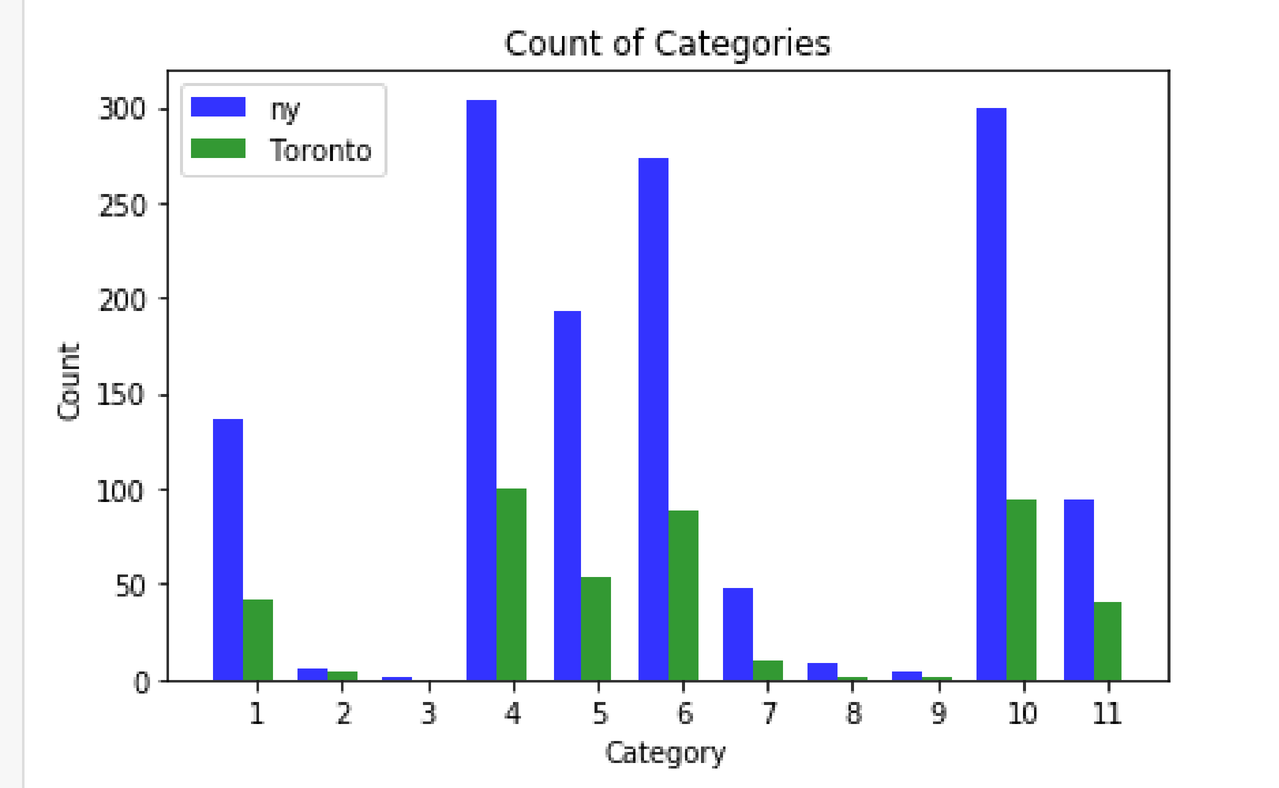
Out[71]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue Category	Venue distance	cat__1	cat__2	cat__3	cat__4	cat__5	cat__6	cat__7	cat__8	cat__9	cat__10	cat__11
0	Wakefield	40.894705	-73.847201	Dessert Shop	127	0	0	0	1	0	0	0	0	0	0	0
1	Wakefield	40.894705	-73.847201	Caribbean Restaurant	798	0	0	0	1	0	0	0	0	0	0	0
2	Wakefield	40.894705	-73.847201	Caribbean Restaurant	822	0	0	0	1	0	0	0	0	0	0	0
3	Wakefield	40.894705	-73.847201	Caribbean Restaurant	686	0	0	0	1	0	0	0	0	0	0	0
4	Wakefield	40.894705	-73.847201	Ice Cream Shop	483	0	0	0	1	0	0	0	0	0	0	0

4,5. Data conditioning and Grouping

- Condition:
- If the distance is less than 500 m the influence remains 1.
- If the distance is more than 500m the influence becomes 0.5.
- Grouping:
- Using Neighbourhoods to group dataframes.
- The category data for each neighbourhood gets added.

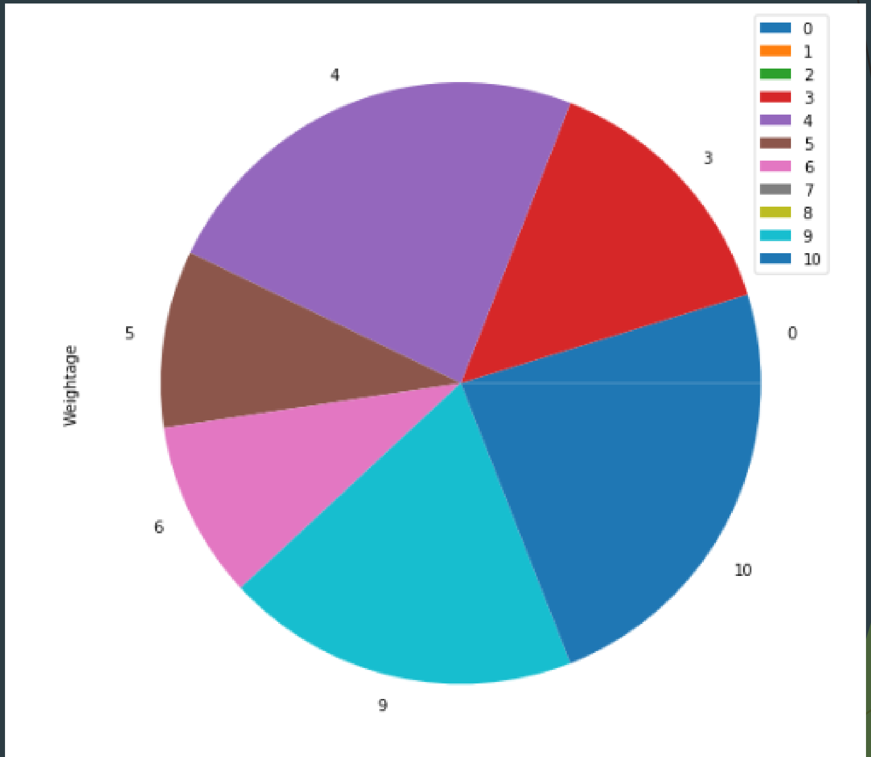
6. Category Selection:



The categories 2,3,8,9 are neglected.

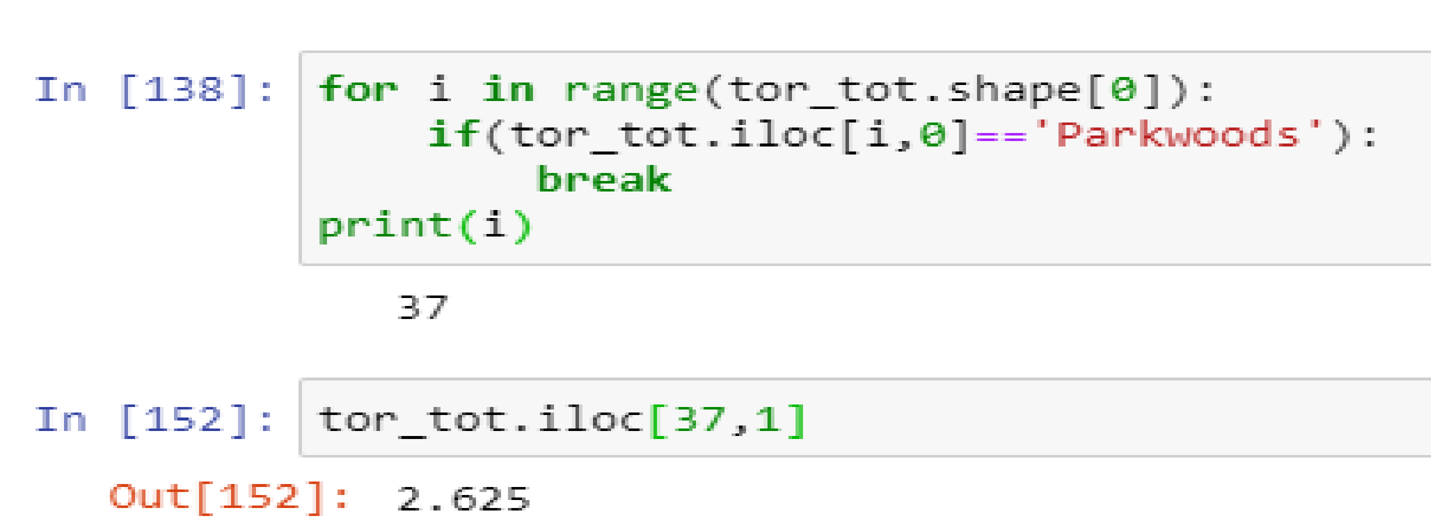
7. Category Weightage:

No	Category	Weight
1	Arts & Entertainment	0.05
2	College & university	0
3	Event	0
4	Food	0.15
5	Nightlife Spot	-0.20
6	Outdoors & Recreation	0.10
7	Professional & Other Places	0.10
8	school	0.0
9	Residence	0.0
10	Shop & Service	0.20
11	Travel & Transport	0.20



8. Total Score:

- Total score is weighted sum of all the categories in the neighbourhood.
- The dataframes are arranged in descending order of Total score.
- The total score is parameter to define compatibility of Tom’s family in given neighbourhood.



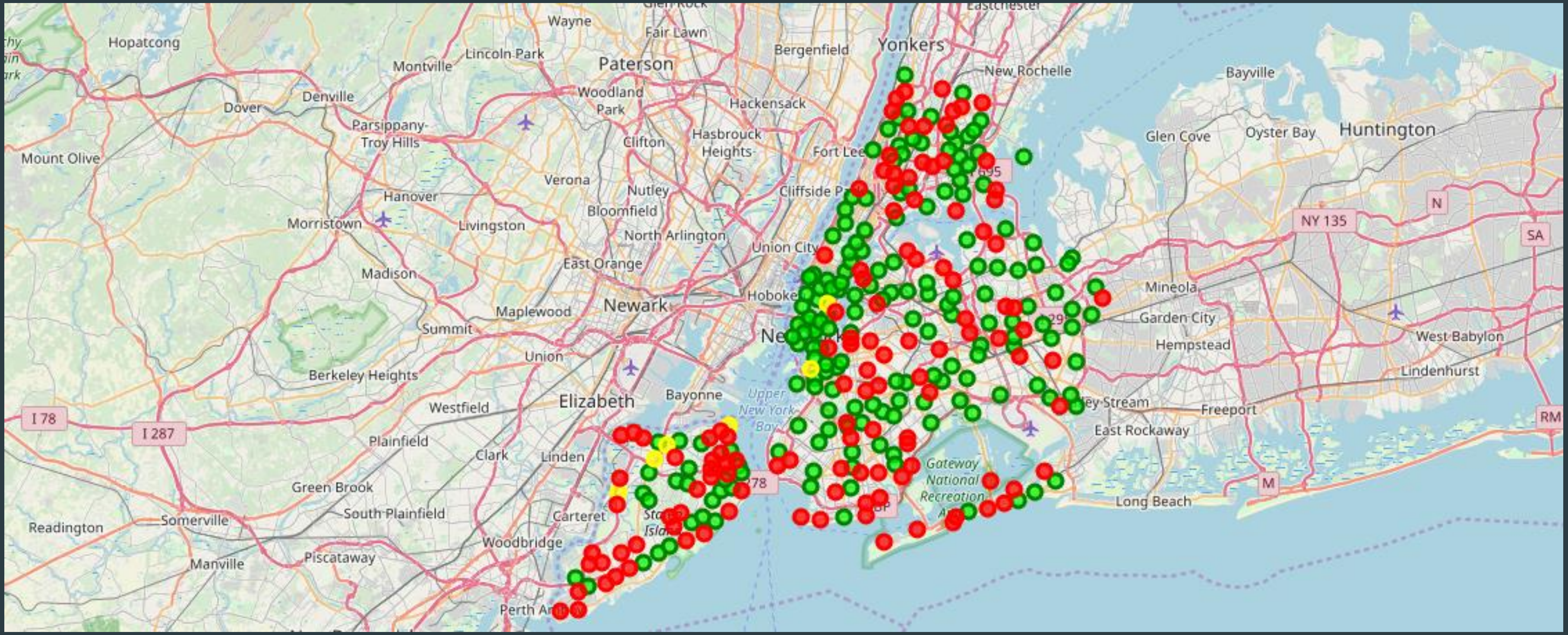
```
In [138]: for i in range(tor_tot.shape[0]):
              if(tor_tot.iloc[i,0]=='Parkwoods'):
                  break
          print(i)

          37

In [152]: tor_tot.iloc[37,1]

Out[152]: 2.625
```

9. Conditional Separating:

- Separating the New York neighbourhoods in three dataframes:

1) Neighbourhoods having more Total score than ‘Parkwoods’.(ny_up)
2) Neighbourhoods having same Total score as that of ‘Parkwoods’. (ny_same)
3) Neighbourhoods having less Total score than ‘Parkwoods’.(ny_down)

Result



ny_up	Green
ny_same	Yellow
ny_down	Red