

## Let's setup the environment by importing required libraries.

In [1]:

```
import pandas as pd # Library for data analysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # Library to handle JSON files

!conda install -c conda-forge geopy --yes # uncommen
from geopy.geocoders import Nominatim # convert an a

import requests # Library to handle requests from
pandas.io.json import json_normalize # tranform

# import k-means from clustering stage from
sklearn.cluster import KMeans

#!conda install -c conda-forge folium=0.5.0 --yes #
import folium # map rendering library

from bs4 import BeautifulSoup import
csv

!pip install geocoder
import geocoder # to get Longitude and Latitude

!conda install -c conda-forge folium=0.5.0 --yes
import folium # Map plotting Library import numpy
as np
from pandas.io.json import json_normalize # Tranform

# Matplotlib and associated plotting modules import
matplotlib.cm as cm
import matplotlib.colors as colors

# Import k-means from clustering stage from
sklearn.cluster import KMeans

print('Libraries imported.')
```

```
usage: conda-script.py [-h] [-V] command ...
conda-script.py: error: unrecognized arguments: # un
comment this line if you haven't completed the Fours
quare API lab
Requirement already satisfied: geocoder in c:\users
\shaba\anaconda3\lib\site-packages (1.38.1) Requirement
already satisfied: ratelim in c:\users\s
haba\anaconda3\lib\site-packages (from geocoder) (0.
1.6)
Requirement already satisfied: future in c:\users\sha
ba\anaconda3\lib\site-packages (from geocoder) (0.1
7.1)
Requirement already satisfied: click in c:\users\sha
ba\anaconda3\lib\site-packages (from geocoder) (7.0)
Requirement already satisfied: requests in c:\users
\shaba\anaconda3\lib\site-packages (from geocoder)
(2.22.0)
```

```
Requirement already satisfied: six in c:\users\shaba\anaconda3\lib\site-packages (from geocoder) (1.12.0)
Requirement already satisfied: decorator in c:\users\shaba\anaconda3\lib\site-packages (from ratelim->geocoder) (4.4.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\shaba\anaconda3\lib\site-packages (from requests->geocoder) (2019.11.28)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\shaba\anaconda3\lib\site-packages (from requests->geocoder) (2.8)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\shaba\anaconda3\lib\site-packages (from requests->geocoder) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in c:\users\shaba\anaconda3\lib\site-packages (from requests->geocoder) (1.24.2)
Collecting package metadata (current_repodata.json): ...working... done
Solving environment: ...working... done
```

```
# All requested packages already installed.
```

```
Libraries imported.
```

```
In [2]:
```

```
source = requests.get('https://en.wikipedia.org/wiki/Manchester')
soup = BeautifulSoup(source, 'lxml')
```

```
In [3]:
```

```
csv_file = open('manchester.csv', 'w')
csv_writer = csv.writer(csv_file)
csv_writer.writerow(['Neighbourhood'])
```

```
Out[3]: 15
```

```
In [4]:
```

```
mwcg = soup.find_all(class_ = "mw-category-group")

length = len(mwcg) # Gets the length of number of `m

for i in range(1, length): # Gets all the neighbour
    lists = mwcg[i].find_all('a')    for list in lists:
        nbd = list.get('title') # Gets the title of
    csv_writer.writerow([nbd]) # Writes the name
```

```
In [5]:
```

```
csv_file.close()
```

```
In [6]:
```

```
manchester_raw = pd.read_csv('manchester.csv')
```

```
manchester_raw.shape
```

```
In [7]:
```

(75 1)  
(75, 1)  
Out[7]:

In  
[8]:

manchester\_raw

	Neighbourhood
Out[8]:	
0	Baguley
1	Barlow Moor
2	Belle Vue, Manchester
3	Benchill
4	Beswick, Manchester
5	Blackley
6	Bradford-with-Beswick
7	Bradford, Manchester
8	Brooklands (Manchester ward)
9	Burnage
10	Burnage (ward)
11	Castlefield
12	Cheetham, Manchester
13	Chorlton (ward)
14	Chorlton Park (ward)
15	Chorlton-cum-Hardy
16	Chorlton-on-Medlock
17	Chorltonville
18	Clayton, Manchester
19	Collyhurst
20	Crumpsall
21	Denton North East (ward)
22	Denton South (ward)
23	Denton West (ward)
24	Didsbury
25	Didsbury East (ward)
26	Didsbury West (ward)

27	Fallowfield
28	Glenbrook, Greater Manchester
29	Gorton
30	Great Heaton
31	Green Quarter
32	Greenheys, Manchester
33	Harpurhey
34	Highfield Country Park
35	Hulme
36	Hyde Godley (ward)
37	Hyde Newton (ward)
38	Hyde Werneth (ward)
39	Ladybarn
40	Levenshulme
41	Longdendale (ward)
42	Longsight
43	Manchester city centre
44	Merseybank
45	Miles Platting
46	Moss Side
47	Mossley (ward)
48	Moston, Manchester
49	New Islington
50	New Moston
51	Newall Green
52	Newton Heath
53	Northenden
54	Northern Moor
55	Northern Quarter (Manchester)
56	Old Moat (ward)
57	Openshaw
58	Parrs Wood
59	Peel Hall, Wythenshawe
60	Ringway, Manchester

61	Sharston
62	Smedley, Manchester
63	Spinningfields
64	St John's, Manchester
65	St. Peters (ward) 66 Stalybridge North (ward)
67	Stalybridge South (ward)
68	Strangeways, Manchester
69	Victoria Park, Manchester
70	West Gorton
71	Whalley Range, Manchester
72	Withington
73	Woodhouse Park
74	Wythenshawe

In [9]:

```
# Initialize variables
lat = [] lng = []
lat_lng_coords = None

# Get postcodes from neighbourhoods table
neighbourhoods = manchester_raw['Neighbourhood']

# Store Latitude and Longitude values in lat and lng
for nh in neighbourhoods:
    g = geocoder.arcgis('{}, Manchester, UK'.format(
lat_lng_coords = g.latlng
lat.append(lat_lng_coords[0])
lng.append(lat_lng_coords[1])
```

In [10]:

```
man_data = manchester_raw
man_data['Latitude'] = lat
man_data['Longitude'] = lng
```

In [11]:

```
man_data.head()
```

	Neighbourhood	Latitude	Longitude
0	Baguley	53.399090	-2.285610
1	Barlow Moor	53.422164	-2.245970
2	Belle Vue, Manchester	53.429585	-2.145901

3	Benchill53.381730	- 2.261250
4	Beswick, Manchester53.478390	- 2.200320

## Let's setup

```
CLIENT_ID = 'JW1LOUKCHOBJJGPYYZFROQZINFPN0AU4E44WNXM
CLIENT_SECRET = 'QEEXIE4ODGP1UZVI35RD00QL2XS2WQWE3Z N
VERSION = '20180605' # Foursquare API version
```

## FourSquare credentials.

In [12]:

```
print('Your credentials:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET: ' + CLIENT_SECRET)
```

Your credentials:

```
CLIENT_ID: JW1LOUKCHOBJJGPYYZFROQZINFPN0AU4E44WNXM51
NPILF3W
CLIENT_SECRET: QEEXIE4ODGP1UZVI35RD00QL2XS2WQWE3ZNAQ3
T4CMSOMZNJ
```

## Now we will get Latitude and Longitude of Manchester.

In [16]:

```
from geopy.geocoders import Nominatim

address = 'Manchester, UK'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address) latitude =
location.latitude longitude = location.longitude
print('The georapical coordinate of Manchester are
```

The georapical coordinate of Manchester are 53.4794  
892, -2.2451148.

## Let's plot map of Manchester using latitude and longitude values.

In [17]:

```
man_map = folium.Map(location=[latitude, longitude],

# add markers to map
for lat, lng, label in zip(man_data['Latitude'], man
label = folium.Popup(label, parse_html=True)
folium.CircleMarker(
    [lat, lng],
    radius=5,
    popup=label,
    color='blue',
    fill=True,
        fill_color='#3186cc',
    fill_opacity=0.7,
    parse_html=False).add_to(man_map)
    man_map
```

Out[17]:

**In the following call we will create a function which will extract the category from the FourSquare vanues. Later we will apply FourSquare API in all the neighbourhoods on Manchester.**

In [18]:

```
# G t th      f th      t

In [18]:
# Gets
the name
of the
category

def
get_category_type(row):
    categories_list = row['Category']
    if
len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

```

In [34]: explore_df_list = []

for i, nhood_name in enumerate(man_data['Neighbourho

try :
    #Get neighbourhood data
    nhood_name = man_data.loc[i, 'Neighbourhood']
    nhood_lat = man_data.loc[i, 'Latitude']
    nhood_lng = man_data.loc[i, 'Longitude']

    radius = 1000
    LIMIT = 30

    url = 'https://api.foursquare.com/v2/venues/
    &client_secret={}&ll={},{ }&v={}&radius={}&li
    .format(CLIENT_ID, CLIENT_SECRET, nhood_lat,

    results = json.loads(requests.get(url).text)
    results = results['response']['groups'][0]['

    nearby = json_normalize(results) # Flatten J

    # Filter the columns
    filtered_columns = ['venue.name', 'venue.cat
    nearby = nearby.loc[:, filtered_columns]

    # Rename the columns
    columns = ['Name', 'Category', 'Latitude', '
    nearby.columns = columns

    # Get the categories
    nearby['Category'] = nearby.apply(get_catego

    # Get the required data
    for i, name
    in enumerate(nearby['Name']):
        s_list =
        f_list =
        nearby.loc[i, :].values.tolist(
        [nhood_name, nhood_lat, nhood_l
        explore_df_list.append(f_list)
    except Exception
    as e:
        pass

```

**Now we will create a dataframe for performing clustering.**

```

In [82]: explore_man = pd.DataFrame([item for item in explore
    ()
    explore_man.columns = ['Neighbourhood', 'Neighbourh o
    explore_man.head()
Out[82]:

```

	Neighbourhood	Neighbourhood	Venue
		Latitude	N Longitude
0	Baguley	53.39909	-2.28561McDon



1	Baguley	53.39909	-2.28561
2	Baguley	53.39909	-2.28561
3	Baguley	53.39909	-2.28561
4	Baguley	53.39909	-2.28561

## Now we will perform one-hot encoding to analyze neighbourhoods of Manchester.

In [83]:

```
man_1hot = pd.get_dummies(explore_man[['Venue Catego

# Add neighbourhood column back to dataframe
man_1hot['Neighbourhood'] = explore_man['Neighbourho

# Move neighbourhood column to the first column
fixed_columns = [man_1hot.columns[-1]] + man_1hot.co
man_1hot = man_1hot[fixed_columns]
man_1hot.head()
```

Out[83]:

	Neighbourhood	Airport	Airport Lounge	American Restaurant	Antique Shop	Ga
0	Baguley	0	0	0	0	
1	Baguley	0	0	0	0	
2	Baguley	0	0	0	0	
3	Baguley	0	0	0	0	
4	Baguley	0	0	0	0	

## In the following step we will agregate the venues by neighbourhoods and later we will create a function to return most common venues.

In [84]:

```
man_grouped = man_1hot.groupby('Neighbourhood').mean
```

```
man_grouped.head()
```

	Neighbourhood	Airport	American	Antique	
		Lounge	Restaurant	Shop	Ga

Out[84]:

0		Baguley	0.0	0.0	0.0	0.0
1		Barlow Moor	0.0	0.0	0.0	0.0
2	Belle Vue, Manchester	0.0 0.0	0.0	0.0		
3		Benchill	0.0	0.0	0.0	0.0
4	Beswick, Manchester	0.0 0.0	0.0	0.0		

In [85]:

```
def top_common_venues(row, top_venues):  
    row_categories = row.iloc[1:]  
    row_categories_sorted = row_categories.sort_valu  
    return row_categories_sorted.index.values[0:top_
```

**Now we will create a new dataframe to get the top 10 venues.**

In [86]:

```
top_venues = 10
indicators = ['st', 'nd', 'rd']

# Create columns according to number of top venues
columns = ['Neighbourhood'] for ind in
np.arange(top_venues):
    try:
        columns.append('{} {} Most Common Venue'.format(ind, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind))

# Create a new dataframe
neighbourhoods_venues_sorted = pd.DataFrame(columns=columns)
neighbourhoods_venues_sorted['Neighbourhood'] = man_grouped.index
for ind in np.arange(man_grouped.shape[0]):
    neighbourhoods_venues_sorted.iloc[ind, 1:] = top_venues[ind]

neighbourhoods_venues_sorted.head()
```

Out[86]:

	Neighbourhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th M Comm Ve
1	Supermarket Station	Tram Store	Hardware 0 Co	Baguley	
2	Belle Vue, Manchester	Donut Shop	Construction Landscaping	Barlow Moor Pub & Intersection	Bar Restau
3		Tram Station	Fast Food Restaurant	Discount Benchill Store	Sta
4	Beswick, Manchester	Track	Athletics & Stadium Sports	Sporting Goods	T Sta Shop

**Now we will apply K-means clustering on the dataframe.**

In [87]: man\_grouped\_clustering = man\_grouped.drop('Neighbour

In [88]:

```
import matplotlib.pyplot as plt
%matplotlib inline
def plot(x, y, xlabel, ylabel):
    plt.figure(figsize=(20,10))    plt.plot(np.arange(2,
x), y, 'o-')    plt.xlabel(xlabel)
plt.ylabel(ylabel)    plt.xticks(np.arange(2, x))
plt.show()
```

In [93]:

```
max_range = 15 #Max range 15 (number of clusters)

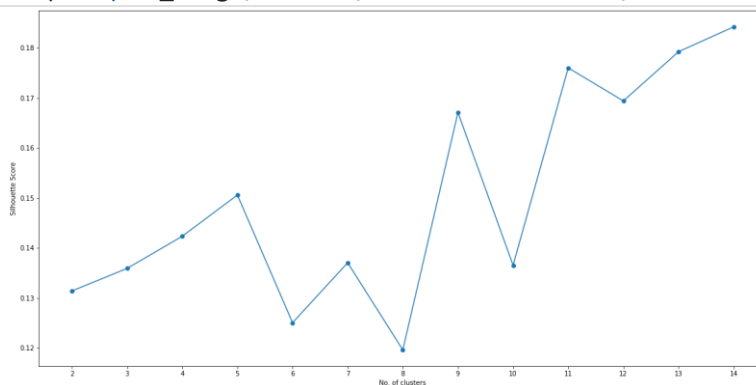
from sklearn.metrics import silhouette_samples, silh

indices = [] scores
= []
for man_clusters in range(2, max_range) :

    # Run k-means clustering
    man_gc = man_grouped_clustering
    kmeans = KMeans(n_clusters = man_clusters, init
    # Gets the score for the clustering operation pe
    score = silhouette_score(man_gc, kmeans)
    # Appending the index and score to the respectiv
    indices.append(man_clusters)    scores.append(score)
```

In [94]:

```
plot(max_range, scores, "No. of clusters", "Silhouet
```



**From the graph the optimal number is found to be 8 using the Elbow Method.**

```
opt_value = 8
```

In [95]:

**Now that we have calculated out optimum value of clusters, we can proceed with K-Means clustering.**

In [96]:

```
man_clusters = opt_value
# Run k-means clustering
man_gc = man_grouped_clustering
kmeans = KMeans(n_clusters = man_clusters, init = 'k
```

In [97]:

```
# Add clustering Labels
neighbourhoods_venues_sorted.insert(0, 'Cluster Labe
```

**Now we will crate a final version of dataframe containing all the necessary data.**

In [98]:

```
man_final = man_data man_final =
man_final.join(neighbourhoods_venues_sor
man_final.dropna(inplace = True) man_final['Cluster
Labels'] = man_final['Cluster Lab man_final.head()
```

Out[98]:

	Neighbourhood	Latitude	Longitude	Cluster	1st Most Common Labels Venue
0	Baguley	53.399090	-2.285610	2	Tram Station
1	Barlow Moor	53.422164	-2.245970	3	Pub
2	Belle Vue, Donut Manchester	42.955859	-71.459019	5	5 Shop
3	Benchill	53.381730	-2.261250	2	Tram Station
4	Beswick, Manchester	53.478390	-2.200320	4	Track Stadium

**In the next step we will visualize the clusters using Folium.**

In [99]:

```
map_clusters = folium.Map(location=[latitude, longit

# Setup color scheme for different clusters x
= np.arange(man_clusters)
ys = [i + x + (i*x)**2 for i in range(man_clusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

markers_colors = []
for lat, lon, poi, cluster in zip(man_final['Latitud
man_final['Cluster' label = folium.Popup(str(poi)
+ ' (Cluster ' + s map_clusters.add_child(
    folium.features.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7))
    map_clusters
```

Out[99]:

## Now lets have a look at all four clusters.

```
In [100... val = 1 man_final.loc[man_final['Cluster Labels'] ==
(val -
```

Out[100...

	Neighbourhood	1st Most Common Com Venue	2nd Most Common Venue	3rd Most Common Venue	4th
	Brooklands				
8	(Manchester ward)	Bar	Italian Restaurant	Asian Restaurant	Rest
11	Castlefield	Restaurant	Bar	Pub	
13	Chorlton (ward)	Bar	Pi a Pizza Place	Pub	
15	Chorlton-cum-Hardy	Bar	Pizza Pub Place		
	Chorlton-on-		Pizza		

16		Bar	Pub		
	Medlock		Place		
			Pizza		
17		Chorltonville	Bar	Pub	
		Place			
35	Hulme	Pub	Café	Bar	
	Manchester city				
43		Pub	Café	Bar	Tea
	centre				
			Sports	Miscellaneous	
47	Mossley (ward)	Bar			
			Club	Shop	
		Coffee			
49	New Islington		Bar	Beer Bar	
		Shop			
	Northern				
	Quarter	Coffee	Pizza	Arts & Crafts	R
55					
	(Manchester)	Shop	Place	Store	
	Ringway,	Coffee		Airport	San
60			Bar		
	Manchester	Shop		Lounge	
		Coffee			
63	Spinningfields		Bar	Pub	Rest
		Shop			
	Strangeways,		Italian	Asian	
68		Bar			Rest
	Manchester		Restaurant	Restaurant	

```
In [101... val = 2 man_final.loc[man_final['Cluster Labels'] ==
(val -
```

Out[101...

	Neighbourhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Mo Commo Venu
					Furnitu
				Sandwich	
5	Blackley	Hotel	Supermarket		/ Hom
				Place	
					Sto
					Gym

7	Bradford, Manchester	Hotel	Supermarket	Flea Market	Fitne Cent
28	Glenbrook, Greater Manchester	Hotel	Park	P Light Rail Station	
29	Gorton	Fitness Train Station	Hotel	Pa Center	
33	Harpurhey	Hotel	Supermarket	Flea Market	Gym Fitne Cent
48	Moston, Manchester	Hotel	Supermarket	Sandwich Place	FI Mark
70	West Gorton	Train Station	Gym / Fitness Center	Hotel	Pa
73	Woodhouse Park	Train Station	Gym / Fitness Center	Soccer Stadium	Sandwi Pla

```
In [102... val = 3 man_final.loc[man_final['Cluster Labels'] ==
(val -
```

Out[102...

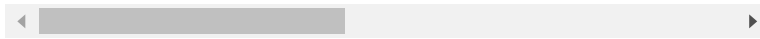
	Neighbourhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4t Co
0	Baguley	Tram Station	Supermarket	Hardware Store	
	Benchill Tram Station		Fast Food Discount	3	
			Restaurant	Store	
18	Clayton, Manchester	Supermarket	Tram Station	Discount Store	Fa Res
20	Crumpsall	Coffee Shop	Bakery	Pizza Place	Sa



New Moston Supermarket Coffee Shop

Restaurant Res

52 Newton Heath Supermarket Fast Food Tram B  
 Restaurant Station  
 Peel Hall, Fast Food Home D 59 Tram Station  
 Wythenshawe Restaurant Service



In [103...

```
val = 4
man_final.loc[man_final['Cluster Labels'] == (val -
```

Out[103...

		1st Most Neighbourhood Common	2nd Most Common Venue	3rd Most Common Venue	4th M Comm Ver
1	Barlow Moor	Pub	Italian Restaurant	Bar	Ind Restaur
9	B rnage	P b	Bar	Park	Cof
9	Burnage Pub	Bar	Park		Sh
					Cof
10	Burnage (ward)	Pub	Bar	Park	Sh
12	Cheetham, Manchester	Coffee Shop	Pub	Italian Restaurant	Brew
		Italian	Groc 24	Didsbury	Park
	Pub Restaurant St				
25	Didsbury East (ward)		HotelPub	Coffee C Shop	
26	Didsbury West (ward)	Italian Restaurant	Park Pub		Groc St
27			Gym / Fallowfield Fitness Center	Asian Pub Ch Restaurant	Fish Sh
30	Great Heaton	Hotel	Pub	Italian Restaurant	Car St

			Coffee	Ital	31	Green Quarter
	Pub	Bar				
			Shop			Restaur
						Vegetar
37	Hyde Newton			Deli /		
	(ward)	Pub	Bar		/ Veg	
				Bodega		Restaur
				Vegetarian		
56	Old Moat (ward)	Pub	Indian		/ Vegan	Restaur
			Restaurant			
				Restaurant		
			Italian			
58	Parrs Wood		Pub	Park		Ho
		Restaurant				
64	St John's,					Cock
	Manchester	Pub	Hotel	Plaza		
				Indian		Fast Fo
72	Withington	Pub	Hotel			
				Restaurant		Restaur

```
In [104... val = 5
man_final.loc[man_final['Cluster
Labels'] == (val -
```

Out[104...

	Neighbourhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	
4	Beswick, Manchester	Track Stadium	Athletics & Sports	Sporting Goods Shop	Tra
6	Bradford-with- Beswick	Track Stadium	Athletics & Sports	Sporting Goods Shop	Tra
14	Chorlton Park (ward)	Grocery Store	Park	Pizza Place	R
19	Collyhurst	Clothing	Fast Food	Coffee Shop	

19	Colyhurst	Store	Restaurant	Coffee Shop	
32	Greenheys, Manchester	Music Venue	Café	Chinese Restaurant	
34	Highfield Country Park	Gym	Fast Food Restaurant	Grocery Store	
36	Hyde Godley (ward)	Pizza Place	Hotel	Supermarket	
38	Hyde Werneth (ward)	Home Service	Pub	Supermarket	
39	Ladybarn	Fast Food Restaurant	Park	Grocery Store	Su
40	Levenshulme	Grocery Store	Indian Restaurant	Café	
42	Longsight	Supermarket	Park	Grocery Store	
44	Merseybank	Middle Eastern Restaurant	Indian Restaurant	Outdoor Supply Store	G
45	Miles Platting	Men's Store	Thai Restaurant	Tram Station	
46	Moss Side	Grocery Store	Park	Tea Room	Su
51	Newall Green	Tram Station	Bus Stop	Sandwich Place	H
53	Northenden	Golf Course	Sandwich Place	Middle Eastern Restaurant	V
54	Northern Moor	Tram Station	Grocery Store	Convenience Store	
57	Openshaw	Bus Stop	Hookah Bar	Clothing Store	
61	Sharston	Sandwich Place	Vietnamese Restaurant	Pub	
62	Smedley, Manchester	Clothing Store	Fast Food Restaurant	Tram Station	Cc