

# battle\_of\_neighborhoods\_part2

August 2, 2019



## 1 Capstone Project - The Battle of Neighborhoods (Week 2)

### 1.1 Introduction

New York City comprises 5 boroughs sitting where the Hudson River meets the Atlantic Ocean. At its core is Manhattan, a densely populated borough that's among the world's major commercial, financial and cultural centers. Its iconic sites include skyscrapers such as the Empire State Building and sprawling Central Park. Broadway theater is staged in neon-lit Times Square.

London, the capital of England and the United Kingdom, is a 21st-century city with history stretching back to Roman times. At its centre stand the imposing Houses of Parliament, the iconic 'Big Ben' clock tower and Westminster Abbey, site of British monarch coronations. Across the Thames River, the London Eye observation wheel provides panoramic views of the South Bank cultural complex, and the entire city.

### 1.2 Description of the problem

We will explore New York City and London and segmented and clustered their neighborhoods. Both cities are very diverse and are very similar. Both cities are a densely populated boroughs that's among the world's major commercial, financial and cultural centers. . We will compare the neighborhoods of the two cities and determine how similar or dissimilar they are. We will define that people like to do more in the cities, which places are often visited. Knowing this information we can think of how to use this. For example, open a new restaurant or supermarket, entertainment center or gift shop. As we can see in the next task that although there are Mexican restaurants in London, but they are not popular, entertainment is centrally located and almost none in areas farther from the center. We may also use this information for advertising purposes, etc

### 1.3 Description of Data.

This project will rely on public data from Wikipedia and Foursquare.

London is the capital of and largest city in England and the United Kingdom. It is administered by the City of London and 32 London boroughs.

We will get information about the areas of London [https://en.wikipedia.org/wiki/List\\_of\\_areas\\_of\\_London](https://en.wikipedia.org/wiki/List_of_areas_of_London)

I will use dataset [https://geo.nyu.edu/catalog/nyu\\_2451\\_34572](https://geo.nyu.edu/catalog/nyu_2451_34572) for information about boroughs of NYC

```
In [1]: # library for BeautifulSoup
        from bs4 import BeautifulSoup

        import numpy as np
        import pandas as pd
        pd.set_option('display.max_columns', None)
        pd.set_option('display.max_rows', None)

        # library to handle JSON files
        import json
        print('numpy, pandas, ..., imported...')

        !pip -q install geopy
        # conda install -c conda-forge geopy --yes # uncomment this line if you haven't comple
        print('geopy installed...')
        # convert an address into latitude and longitude values
        from geopy.geocoders import Nominatim
        print('Nominatim imported...')

        # library to handle requests
        import requests
        print('requests imported...')

        # tranform JSON file into a pandas dataframe
        from pandas.io.json import json_normalize
        print('json_normalize imported...')

        # Matplotlib and associated plotting modules
        import matplotlib.cm as cm
        import matplotlib.colors as colors
        print('matplotlib imported...')

        # import k-means from clustering stage
        from sklearn.cluster import KMeans
        print('Kmeans imported...')

        # install the Geocoder
        !pip -q install geocoder
        import geocoder
```

```

# import time
import time

# !conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you haven't
!pip -q install folium
print('folium installed...')
import folium # map rendering library
print('folium imported...')
print('...Done')

from PIL import Image # converting images into arrays

%matplotlib inline

import matplotlib as mpl
import matplotlib.pyplot as plt

mpl.style.use('ggplot') # optional: for ggplot-like style

# check for latest version of Matplotlib
print ('Matplotlib version: ', mpl.__version__) # >= 2.0.0

# install wordcloud
!conda install -c conda-forge wordcloud==1.4.1 --yes

# import package and its set of stopwords
from wordcloud import WordCloud, STOPWORDS

print ('Wordcloud is installed and imported!')

numpy, pandas, ..., imported...
geopy installed...
Nominatim imported...
requests imported...
json_normalize imported...
matplotlib imported...
Kmeans imported...
folium installed...
folium imported...
...Done
Matplotlib version: 2.2.3
Solving environment: done

==> WARNING: A newer version of conda exists. <==
current version: 4.5.12
latest version: 4.7.10

```

Please update conda by running

```
$ conda update -n base conda
```

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

```
Wordcloud is installed and imported!
```

### 1.3.1 London

```
In [2]: # download data and parse it:
```

```
r = requests.get('https://en.wikipedia.org/wiki/List_of_areas_of_London')
soup = BeautifulSoup(r.text, 'html.parser')
table=soup.find('table', attrs={'class':'wikitable sortable'})
```

```
In [3]: #get headers:
```

```
headers=table.findAll('th')
for i, head in enumerate(headers): headers[i]=str(headers[i]).replace("<th>", "").replace("</th>", "")
#headers
```

```
In [4]: #Find all items and skip first one:
```

```
rows=table.findAll('tr')
rows=rows[1:len(rows)]
#rows
```

```
In [5]: # skip all meta symbols and line feeds between rows:
```

```
for i, row in enumerate(rows): rows[i] = str(rows[i]).replace("\n</td></tr>", "").replace("<br>", "")
#rows
```

```
In [6]: # make dataframe, expand rows and drop the old one:
```

```
df=pd.DataFrame(rows)
df[headers] = df[0].str.split("</td>\n<td>", n = 7, expand = True)
df.drop(columns=[0],inplace=True)#

df.rename(columns={'Location': 'neighborhoods', 'London\xa0borough': 'borough', 'Post t...
df.drop(columns={'OS grid ref'},inplace=True)
df.head(3)
```

```
Out [6]:
```

```
neighborhoods \
0 <a href="/wiki/Abbey_Wood" title="Abbey Wood">...
1 <a href="/wiki/Acton,_London" title="Acton, Lo...
2 <a href="/wiki/Addington,_London" title="Addin...

borough posttown postcode \
0 Bexley, Greenwich <sup class="reference" id="... LONDON SE2
```

```

1 Ealing, Hammersmith and Fulham<sup class="refe... LONDON W3, W4
2 Croydon<sup class="reference" id="cite_ref-mil... CROYDON CRO

```

```

Dialäcode
0      020
1      020
2      020

```

```

In [7]: df.update(df.neighborhoods.loc[lambda x: x.str.contains('title')].str.extract('title=\\
# delete Toronto annotation from Neighbourhood:
df.update(df.neighborhoods.loc[lambda x: x.str.contains('London')].str.replace(", Lond

```

```

In [8]: for i in range(0, df.shape[0]-1):
        #print(df.borough.get_values()[i])
        c = df.borough.get_values()[i].split('<')[0]
        df.borough[i] = c

```

```

df = df.drop('borough', axis=1).join(df['borough'].str.split(',', expand=True).stack()
df = df.drop('posttown', axis=1).join(df['posttown'].str.split(',', expand=True).stack
df = df.drop('postcode', axis=1).join(df['postcode'].str.split(',', expand=True).stack

```

```

In [9]: df.head()

```

```

Out[9]: neighborhoods Dialäcode borough posttown postcode
0      Abbey Wood      020 Bexley LONDON SE2
0      Abbey Wood      020 Bexley LONDON SE2
0      Abbey Wood      020 Bexley LONDON SE2
0      Abbey Wood      020 Bexley LONDON SE2
0      Abbey Wood      020 Bexley LONDON SE2

```

```

In [10]: df.shape

```

```

Out[10]: (2856, 5)

```

```

In [11]: df.drop_duplicates(keep = False, inplace = True)

```

```

In [12]: df.head()

```

```

Out[12]: neighborhoods Dialäcode borough posttown postcode
2      Addington      020 Croydon CROYDON CRO
3      Addiscombe      020 Croydon CROYDON CRO
5 Aldborough Hatch      020 Redbridge ILFORD IG2
6      Aldgate      020 City LONDON EC3
7      Aldwych      020 Westminster LONDON WC2

```

```

In [13]: df.shape

```

```

Out[13]: (578, 5)

```

Now, only the Boroughs with London Post-town will be used for our search of location. Therefore, all the non-post-town are dropped.

```
In [14]: df_london = df
df_london = df_london[df_london['posttown'].str.contains('LONDON')]

df_london.drop_duplicates(keep = False, inplace = True)
```

/opt/conda/lib/python3.6/site-packages/ipykernel\_launcher.py:4: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html>  
after removing the cwd from sys.path.

```
In [15]: df_london.head()
```

```
Out[15]:
```

	neighborhoods	Dialcode	borough	posttown	postcode
6	Aldgate	020	City	LONDON	EC3
7	Aldwych	020	Westminster	LONDON	WC2
9	Anerley	020	Bromley	LONDON	SE20
10	Angel	020	Islington	LONDON	EC1
10	Angel	020	Islington	LONDON	N1

**Geocoder dont read my whole data, and i divide my dataset on smaller parts**

```
In [16]: # Geocoder starts here
import geocoder # import geocoder

# Defining a function to use --> get_latlng()'''
def get_latlng(arcgis_geocoder):

    # Initialize the Location (lat. and long.) to "None"
    lat_lng_coords = None

    # While loop helps to create a continuous run until all the location coordinates are found
    while(lat_lng_coords is None):
        g = geocoder.arcgis('{}, London, United Kingdom'.format(arcgis_geocoder))
        lat_lng_coords = g.latlng
    return lat_lng_coords
# Geocoder ends here

In [17]: # New dataframe for postcodes started with "W"
df_w = df_london[df_london['postcode'].str.startswith(('W'))].reset_index(drop=True)

In [18]: df_w.head()
```

```
Out[18]:
```

	neighborhoods	Dialcode	borough	posttown	postcode
0	Aldwych	020	Westminster	LONDON	WC2
1	Bayswater	020	Westminster	LONDON	W2
2	Bedford Park	020	Ealing	LONDON	W4
3	Bloomsbury	020	Camden	LONDON	WC1
4	Charing Cross	020	Westminster	LONDON	WC2

```
In [19]: postcode = df_w['postcode']
         postcode
         coordinates = [get_latlng(postcode) for postcode in postcode.tolist()]
```

```
In [20]: df_with_coordinates = df_w
```

```
# The obtained coordinates (latitude and longitude) are joined with the dataframe as .
df_with_coordinates = pd.DataFrame(coordinates, columns = ['Latitude', 'Longitude'])
df_w['Latitude'] = df_with_coordinates['Latitude']
df_w['Longitude'] = df_with_coordinates['Longitude']
```

```
In [21]: df_w.head()
```

```
Out[21]:
```

	neighborhoods	Dialcode	borough	posttown	postcode	Latitude	Longitude
0	Aldwych	020	Westminster	LONDON	WC2	51.51651	-0.11968
1	Bayswater	020	Westminster	LONDON	W2	51.51494	-0.18048
2	Bedford Park	020	Ealing	LONDON	W4	51.48944	-0.26194
3	Bloomsbury	020	Camden	LONDON	WC1	51.52450	-0.12273
4	Charing Cross	020	Westminster	LONDON	WC2	51.51651	-0.11968

```
In [22]: # # New dataframe for postcodes started with "S"
         df_s = df_london[df_london['postcode'].str.startswith(('S'))].reset_index(drop=True)
```

```
In [23]: df_s.head()
```

```
Out[23]:
```

	neighborhoods	Dialcode	borough	posttown	postcode
0	Anerley	020	Bromley	LONDON	SE20
1	Balham	020	Wandsworth	LONDON	SW12
2	Bankside	020	Southwark	LONDON	SE1
3	Barnes	020	Richmond upon Thames	LONDON	SW13
4	Battersea	020	Wandsworth	LONDON	SW11

```
In [24]: postcode = df_s['postcode']
         postcode
         coordinates = [get_latlng(postcode) for postcode in postcode.tolist()]
```

```
In [25]: df_with_coordinates_s = df_s
```

```
# The obtained coordinates (latitude and longitude) are joined with the dataframe as .
df_with_coordinates_s = pd.DataFrame(coordinates, columns = ['Latitude', 'Longitude'])
df_s['Latitude'] = df_with_coordinates_s['Latitude']
df_s['Longitude'] = df_with_coordinates_s['Longitude']
```

```
In [26]: df_s.head()
```

```
Out[26]:
```

	neighborhoods	Dialcode	borough	posttown	postcode	Latitude	Longitude
0	Anerley	020	Bromley	LONDON	SE20	51.41009	-0.05683
1	Balham	020	Wandsworth	LONDON	SW12	51.44822	-0.14839
2	Bankside	020	Southwark	LONDON	SE1	51.49960	-0.09613
3	Barnes	020	Richmond upon Thames	LONDON	SW13	51.47457	-0.24212
4	Battersea	020	Wandsworth	LONDON	SW11	51.46760	-0.16290

```
In [27]: # df_london_allpart = df_s and df_w
df_london_allpart = df_s.append(df_w, ignore_index=True)
```

```
In [28]: df_london_allpart.shape
```

```
Out[28]: (129, 7)
```

```
In [29]: # New dataframe for postcodes started with "E"
df_e = df_london[df_london['postcode'].str.startswith(('E'))].reset_index(drop=True)
```

```
In [30]: postcode = df_e['postcode']
postcode
coordinates = [get_latlng(postcode) for postcode in postcode.tolist()]
```

```
In [31]: f_with_coordinates_e = df_e
```

```
# The obtained coordinates (latitude and longitude) are joined with the dataframe as
df_with_coordinates_e = pd.DataFrame(coordinates, columns = ['Latitude', 'Longitude'])
df_e['Latitude'] = df_with_coordinates_e['Latitude']
df_e['Longitude'] = df_with_coordinates_e['Longitude']
```

```
In [32]: df_london_allpart = df_london_allpart.append(df_e, ignore_index=True)
```

```
In [33]: # New dataframe for postcodes started with "N"
df_n = df_london[df_london['postcode'].str.startswith(('N'))].reset_index(drop=True)
```

```
In [34]: postcode = df_n['postcode']
postcode
coordinates = [get_latlng(postcode) for postcode in postcode.tolist()]
```

```
In [35]: df_with_coordinates_s = df_n
```

```
# The obtained coordinates (latitude and longitude) are joined with the dataframe as
df_with_coordinates_n = pd.DataFrame(coordinates, columns = ['Latitude', 'Longitude'])
df_n['Latitude'] = df_with_coordinates_n['Latitude']
df_n['Longitude'] = df_with_coordinates_n['Longitude']
```



```
In [36]: df_london_allpart = df_london_allpart.append(df_n, ignore_index=True)
```

```
In [37]: df_london_allpart.head(20)
```

```
Out[37]:
```

	neighborhoods	Dialcode	borough	posttown	postcode \
0	Anerley	020	Bromley	LONDON	SE20
1	Balham	020	Wandsworth	LONDON	SW12
2	Bankside	020	Southwark	LONDON	SE1
3	Barnes	020	Richmond upon Thames	LONDON	SW13
4	Battersea	020	Wandsworth	LONDON	SW11
5	Belgravia	020	Westminster	LONDON	SW1
6	Bellingham	020	Lewisham	LONDON	SE6
7	Bermondsey	020	Southwark	LONDON	SE1
8	Blackheath	020	Lewisham	LONDON	SE3
9	Blackheath	020	Greenwich	LONDON	SE3
10	Brixton	020	Lambeth	LONDON	SW2
11	Brockley	020	Lewisham	LONDON	SE4
12	Brompton	020	Kensington and Chelsea	LONDON	SW3
13	Camberwell	020	Southwark	LONDON	SE5
14	Castelnau	020	Richmond upon Thames	LONDON	SW13
15	Catford	020	Lewisham	LONDON	SE6
16	Charlton	020	Greenwich	LONDON	SE7
17	Chelsea	020	Kensington and Chelsea	LONDON	SW3
18	Chinbrook	020	Lewisham	LONDON	SE12
19	Colliers Wood	020	Merton	LONDON	SW19

	Latitude	Longitude
0	51.41009	-0.05683
1	51.44822	-0.14839
2	51.49960	-0.09613
3	51.47457	-0.24212
4	51.46760	-0.16290
5	51.49713	-0.13829
6	51.43722	-0.01868
7	51.49960	-0.09613
8	51.47138	0.02338
9	51.47138	0.02338
10	51.45295	-0.12083
11	51.46268	-0.03558
12	51.49014	-0.16248
13	51.47480	-0.09313
14	51.47457	-0.24212
15	51.43722	-0.01868
16	51.48759	0.02912
17	51.49014	-0.16248
18	51.44759	0.01350
19	51.42170	-0.20796

```
In [38]: # New dataframe for postcodes started with "d"
```

```

df_d = df_london[df_london['postcode'].str.startswith(('D'))].reset_index(drop=True)

In [39]: postcode = df_d['postcode']
        postcode
        coordinates = [get_latlng(postcode) for postcode in postcode.tolist()]

In [40]: df_with_coordinates_s = df_d

        # The obtained coordinates (latitude and longitude) are joined with the dataframe as
df_with_coordinates_d = pd.DataFrame(coordinates, columns = ['Latitude', 'Longitude'])
df_d['Latitude'] = df_with_coordinates_d['Latitude']
df_d['Longitude'] = df_with_coordinates_d['Longitude']

In [41]: df_london_allpart = df_london_allpart.append(df_d, ignore_index=True)

In [42]: # New dataframe for postcodes started with "I" / same =E18
        df_i = df_london[df_london['postcode'].str.startswith(('I'))].reset_index(drop=True)

In [43]: postcode = df_i['postcode']
        postcode
        coordinates = [get_latlng(postcode) for postcode in postcode.tolist()]

In [44]: df_with_coordinates_s = df_i

        # The obtained coordinates (latitude and longitude) are joined with the dataframe as
df_with_coordinates_i = pd.DataFrame(coordinates, columns = ['Latitude', 'Longitude'])
df_i['Latitude'] = df_with_coordinates_i['Latitude']
df_i['Longitude'] = df_with_coordinates_i['Longitude']

In [45]: df_london_allpart = df_london_allpart.append(df_i, ignore_index=True)

In [46]: df_london_allpart.shape

Out[46]: (285, 7)

In [47]: df_london_allpart['borough'].unique()

Out[47]: array(['Bromley', 'Wandsworth', 'Southwark', 'Richmond upon Thames',
                'Westminster', 'Lewisham', 'Greenwich', 'Lambeth',
                'Kensington and Chelsea', 'Merton', 'Bexley',
                'Hammersmith and Fulham', 'Kingston upon Thames', 'Croydon',
                'Ealing', 'Camden', 'Hounslow', 'Camden and Islington', 'City',
                'Islington', 'Tower Hamlets', 'Waltham Forest', 'Newham', 'Hackney',
                'Islington & City', 'Redbridge', 'Enfield', 'Haringey',
                'Barnet', 'Brent', 'Haringey and Barnet', 'Dartford'], dtype=object)

In [49]: print('The dataframe has {} boroughs and {} neighborhoods.'.format(
        len(df_london_allpart['borough'].unique()),
        df_london_allpart.shape[0]
    )
)

```

The dataframe has 32 boroughs and 285 neighborhoods.

Use geopy library to get the latitude and longitude values of London. In order to define an instance of the geocoder, we need to define a user\_agent. We will name our agent ny\_explorer, as shown below.

```
In [50]: address = 'London, uk'

         geolocator = Nominatim(user_agent="ny_explorer")
         location = geolocator.geocode(address)
         latitude = location.latitude
         longitude = location.longitude
         print('The geographical coordinate of London, uk {}, {}'.format(latitude, longitude))
```

The geographical coordinate of London, uk 51.4893335, -0.144055084527687.

Create a map of London with borough superimposed on top.

```
In [51]: # create map of London using latitude and longitude values
         map_london = folium.Map(location=[latitude, longitude], zoom_start=10)

         # add markers to map
         for lat, lng, label in zip(df_london_allpart['Latitude'], df_london_allpart['Longitude']):
             label = '{}.format(borough)'.format(borough)
             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(map_london)

         map_london
```

```
Out[51]: <folium.folium.Map at 0x7fe50fca8240>
```

### 1.3.2 New York

Download and Explore Dataset Neighborhood has a total of 5 boroughs and 306 neighborhoods. In order to segment the neighborhoods and explore them, we will essentially need a dataset that contains the 5 boroughs and the neighborhoods that exist in each borough as well as the latitude and longitude coordinates of each neighborhood.

The link to the dataset: [https://geo.nyu.edu/catalog/nyu\\_2451\\_34572](https://geo.nyu.edu/catalog/nyu_2451_34572)

```
In [52]: # load the data
!wget -q -O 'newyork_data.json' https://cocl.us/new_york_dataset
print('Data downloaded!')
```

Data downloaded!

```
In [53]: with open('newyork_data.json') as json_data:
    newyork_data = json.load(json_data)
```

```
In [54]: neighborhoods_data = newyork_data['features']
```

```
In [55]: # define the dataframe columns
column_names = ['Borough', 'Neighborhood', 'Latitude', 'Longitude']

# instantiate the dataframe
neighborhoods = pd.DataFrame(columns=column_names)
```

```
In [56]: neighborhoods
```

```
Out[56]: Empty DataFrame
Columns: [Borough, Neighborhood, Latitude, Longitude]
Index: []
```

```
In [57]: # let's loop through the data and fill the dataframe one row at a time.
for data in neighborhoods_data:
    borough = neighborhood_name = data['properties']['borough']
    neighborhood_name = data['properties']['name']

    neighborhood_latlon = data['geometry']['coordinates']
    neighborhood_lat = neighborhood_latlon[1]
    neighborhood_lon = neighborhood_latlon[0]

    neighborhoods = neighborhoods.append({'Borough': borough,
                                          'Neighborhood': neighborhood_name,
                                          'Latitude': neighborhood_lat,
                                          'Longitude': neighborhood_lon}, ignore_index=True)
```

```
In [58]: neighborhoods.head()
```

```
Out[58]:
```

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

```
In [59]: print('The dataframe has {} boroughs and {} neighborhoods.'.format(
    len(neighborhoods['Borough'].unique()),
    neighborhoods.shape[0]
))
```

The dataframe has 5 boroughs and 306 neighborhoods.

Use geopy library to get the latitude and longitude values of New York City. In order to define an instance of the geocoder, we need to define a user\_agent. We will name our agent ny\_explorer, as shown below.

```
In [60]: address = 'New York City, NY'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of New York City are {}, {}'.format(latitude, longitude))
```

The geograpical coordinate of New York City are 40.7127281, -74.0060152.

Create a map of New York with neighborhoods superimposed on top.

```
In [61]: # create map of New York using latitude and longitude values
map_newyork = folium.Map(location=[latitude, longitude], zoom_start=10)

# add markers to map
for lat, lng, borough, neighborhood in zip(neighborhoods['Latitude'], neighborhoods['Longitude'],
neighborhoods['Borough'], neighborhoods['Neighborhood']):
    label = '{} {}'.format(neighborhood, borough)
    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(map_newyork)

map_newyork
```

```
Out[61]: <folium.folium.Map at 0x7fe50dec63c8>
```

## 1.4 Methodology

### 1.4.1 Data Exploration

Create a new dataframe of the borough Kensington and Chelsea.

```
In [63]: new_data = df_london_allpart[df_london_allpart['borough'] == 'Kensington and Chelsea']
new_data = new_data.drop_duplicates()
new_data.head(10)
```

```

Out [63]:      neighborhoods Dialcode      borough posttown postcode \
0      Brompton      020 Kensington and Chelsea LONDON      SW3
1      Chelsea      020 Kensington and Chelsea LONDON      SW3
2      Earls Court      020 Kensington and Chelsea LONDON      SW5
3      Kensington      020 Kensington and Chelsea LONDON      SW7
4      South Kensington      020 Kensington and Chelsea LONDON      SW7
5      West Brompton      020 Kensington and Chelsea LONDON      SW10
6      Holland Park      020 Kensington and Chelsea LONDON      W8
7      North Kensington      020 Kensington and Chelsea LONDON      W10
8      Notting Hill      020 Kensington and Chelsea LONDON      W11

      Latitude Longitude
0  51.49014  -0.16248
1  51.49014  -0.16248
2  51.49004  -0.18971
3  51.49807  -0.17404
4  51.49807  -0.17404
5  51.48563  -0.18144
6  51.50162  -0.19173
7  51.52346  -0.21353
8  51.51244  -0.20639

```

```

In [160]: address = 'Kensington and Chelsea, uk'
          #address = 'City of London, uk'
          #address = 'Islington, uk'

          geolocator = Nominatim(user_agent="uk_explorer")
          location = geolocator.geocode(address)
          latitude = location.latitude
          longitude = location.longitude
          print('The geograpical coordinate of London, uk {}, {}'.format(latitude, longitude))

```

The geograpical coordinate of London, uk 51.4989948, -0.1991229.

```

In [162]: new_data.head()

```

```

Out [162]:      neighborhoods Dialcode      borough posttown postcode \
0      Brompton      020 Kensington and Chelsea LONDON      SW3
1      Chelsea      020 Kensington and Chelsea LONDON      SW3
2      Earls Court      020 Kensington and Chelsea LONDON      SW5
3      Kensington      020 Kensington and Chelsea LONDON      SW7
4      South Kensington      020 Kensington and Chelsea LONDON      SW7

      Latitude Longitude Cluster Labels
0  51.49014  -0.16248           2
1  51.49014  -0.16248           2
2  51.49004  -0.18971           3

```

```

3  51.49807  -0.17404          3
4  51.49807  -0.17404          1

```

```

In [163]: #create map of North York using latitude and longitude values #new_data['neighborh
map_london_borough = folium.Map(location=[latitude, longitude], zoom_start=12)

# add markers to map
for lat, lng, label in zip(new_data['Latitude'], new_data['Longitude'], new_data['bor
    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(map_london_borough)

map_london_borough

```

```

Out[163]: <folium.folium.Map at 0x7fe50ca2f668>

```

Use geopy library to get the latitude and longitude values borough Manhattan.

```

In [164]: address = 'Manhattan, usa'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Manhattan are {}, {}'.format(latitude, longitude)

```

The geograpical coordinate of Manhattan are 40.7900869, -73.9598295.

```

In [165]: ny_data = neighborhoods[neighborhoods['Borough'] == 'Manhattan'].reset_index(drop=True)
ny_data.head()

```

```

Out[165]:
   Borough      Neighborhood  Latitude  Longitude
0  Manhattan      Marble Hill   40.876551  -73.910660
1  Manhattan      Chinatown    40.715618  -73.994279
2  Manhattan  Washington Heights  40.851903  -73.936900
3  Manhattan           Inwood   40.867684  -73.921210
4  Manhattan  Hamilton Heights  40.823604  -73.949688

```

Let's visualizat Manhattan

```

In [166]: # create map of Manhattan using latitude and longitude values
map_nyc_m = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(ny_data['Latitude'], ny_data['Longitude'], ny_data['Neighborhood']):
    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(map_nyc_m)

map_nyc_m

```

```

Out[166]: <folium.folium.Map at 0x7fe50c9df5c0>

```

Next, we are going to start utilizing the Foursquare API to explore the neighborhoods and segment them.

```

In [70]: CLIENT_ID = 'B3D1FREXU3FMFKGOXFFFWLZH1UBNQKQGVTG4XWBI3N32354V' # your Foursquare ID
CLIENT_SECRET = 'UAFKLDYGA1SQEBZY04P5DYUAS4DBRF5QA53DURWY03FTRQP3' # your Foursquare secret
VERSION = '20180604'
LIMIT = 30
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET: ' + CLIENT_SECRET)

```

Your credentails:

```

CLIENT_ID: B3D1FREXU3FMFKGOXFFFWLZH1UBNQKQGVTG4XWBI3N32354V
CLIENT_SECRET:UAFKLDYGA1SQEBZY04P5DYUAS4DBRF5QA53DURWY03FTRQP3

```

```

In [71]: address = 'Manhattan, usa'

geolocator = Nominatim(user_agent="ny_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Manhattan, usa are {}, {}'.format(latitude, longitude))

```

The geograpical coordinate of Manhattan, usa are 40.7900869, -73.9598295.

Get the neighborhood's latitude and longitude values.

Now, let's get the top 200 venues that are in Manhattan within a radius of 1000 meters. First, let's create the GET request URL. Name your URL url.



```
In [72]: LIMIT = 200 # limit of number of venues returned by Foursquare API
        radius = 1000 # define radius
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v=
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            latitude,
            longitude,
            radius,
            LIMIT)
        url # display URL
```

```
Out[72]: 'https://api.foursquare.com/v2/venues/explore?&client_id=B3D1FREXU3FMFKGOXFFFWLZH1UBN'
```

Send the GET request and examine the results

```
In [167]: results = requests.get(url).json()
        #results
```

```
In [74]: # function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

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

Now we are ready to clean the json and structure it into a pandas dataframe.

```
In [75]: venues = results['response']['groups'][0]['items']

        nearby_venues = json_normalize(venues) # flatten JSON

        # filter columns
        filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.lo
        nearby_venues =nearby_venues.loc[:, filtered_columns]

        # filter the category for each row
        nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)

        # clean columns
        nearby_venues.columns = [col.split(".")[1] for col in nearby_venues.columns]

        nearby_venues.head(10)
```

```

Out [75]:
          name      categories      lat \
0      North Meadow      Park  40.792027
1      Central Park Tennis Center  Tennis Court  40.789313
2      East Meadow      Field  40.790160
3      Central Park - Gate Of All Saints      Park  40.791591
4      The Jewish Museum      Museum  40.785276
5      Da Capo      Café  40.787679
6      Central Park - Wild West Playground      Playground  40.789715
7      K&D Wines & Spirits      Wine Shop  40.787096
8      Russ & Daughters  Kosher Restaurant  40.785332
9      Central Park - Woodman's Gate      Park  40.787786

          lng
0 -73.959853
1 -73.961862
2 -73.955498
3 -73.964795
4 -73.957411
5 -73.953899
6 -73.965471
7 -73.954261
8 -73.957481
9 -73.955924

```

And how many venues were returned by Foursquare?

```

In [76]: print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))

100 venues were returned by Foursquare.

```

Let's create a function to repeat the same process to all the neighborhoods.

```

In [77]: def getNearbyVenues(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&version={}&lat={}&lng={}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

```

```

# make the GET request
results = requests.get(url).json()["response"]["groups"][0]['items']

# return only relevant information for each nearby venue
venues_list.append([
    name,
    lat,
    lng,
    v['venue']['name'],
    v['venue']['location']['lat'],
    v['venue']['location']['lng'],
    v['venue']['categories'][0]['name']) for v in results])

nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
nearby_venues.columns = ['Neighborhood',
                        'Neighborhood Latitude',
                        'Neighborhood Longitude',
                        'Venue',
                        'Venue Latitude',
                        'Venue Longitude',
                        'Venue Category']

return(nearby_venues)

```

Create a new dataframe called ny\_venues.

```

In [78]: ny_venues = getNearbyVenues(names=ny_data['Neighborhood'],
                                     latitudes=ny_data['Latitude'],
                                     longitudes=ny_data['Longitude']
                                     )

```

```

Marble Hill
Chinatown
Washington Heights
Inwood
Hamilton Heights
Manhattanville
Central Harlem
East Harlem
Upper East Side
Yorkville
Lenox Hill
Roosevelt Island
Upper West Side
Lincoln Square
Clinton
Midtown
Murray Hill

```

Chelsea  
 Greenwich Village  
 East Village  
 Lower East Side  
 Tribeca  
 Little Italy  
 Soho  
 West Village  
 Manhattan Valley  
 Morningside Heights  
 Gramercy  
 Battery Park City  
 Financial District  
 Carnegie Hill  
 Noho  
 Civic Center  
 Midtown South  
 Sutton Place  
 Turtle Bay  
 Tudor City  
 Stuyvesant Town  
 Flatiron  
 Hudson Yards

```
In [79]: print(ny_venues.shape)
```

```
(3315, 7)
```

```
In [168]: ny_venues.head(5)
```

```
Out[168]:
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue \
0	Marble Hill	40.876551	-73.91066	Arturo's
1	Marble Hill	40.876551	-73.91066	Bikram Yoga
2	Marble Hill	40.876551	-73.91066	Tibbett Diner
3	Marble Hill	40.876551	-73.91066	Starbucks
4	Marble Hill	40.876551	-73.91066	Dunkin'

  

	Venue Latitude	Venue Longitude	Venue Category
0	40.874412	-73.910271	Pizza Place
1	40.876844	-73.906204	Yoga Studio
2	40.880404	-73.908937	Diner
3	40.877531	-73.905582	Coffee Shop
4	40.877136	-73.906666	Donut Shop

Let's check how many venues were returned for each neighborhood

```
In [81]: ny_venues.groupby('Neighborhood').count()
```

Out[81]:

Neighborhood	Latitude	Longitude	Venue	\
Battery Park City	99	99	99	
Carnegie Hill	100	100	100	
Central Harlem	42	42	42	
Chelsea	100	100	100	
Chinatown	100	100	100	
Civic Center	100	100	100	
Clinton	100	100	100	
East Harlem	43	43	43	
East Village	100	100	100	
Financial District	100	100	100	
Flatiron	100	100	100	
Gramercy	100	100	100	
Greenwich Village	100	100	100	
Hamilton Heights	61	61	61	
Hudson Yards	75	75	75	
Inwood	57	57	57	
Lenox Hill	100	100	100	
Lincoln Square	100	100	100	
Little Italy	100	100	100	
Lower East Side	60	60	60	
Manhattan Valley	56	56	56	
Manhattanville	40	40	40	
Marble Hill	26	26	26	
Midtown	100	100	100	
Midtown South	100	100	100	
Morningside Heights	40	40	40	
Murray Hill	100	100	100	
Noho	100	100	100	
Roosevelt Island	29	29	29	
Soho	100	100	100	
Stuyvesant Town	17	17	17	
Sutton Place	100	100	100	
Tribeca	100	100	100	
Tudor City	83	83	83	
Turtle Bay	100	100	100	
Upper East Side	100	100	100	
Upper West Side	100	100	100	
Washington Heights	87	87	87	
West Village	100	100	100	
Yorkville	100	100	100	

Neighborhood	Venue Latitude	Venue Longitude	Venue Category
Battery Park City	99	99	99
Carnegie Hill	100	100	100
Central Harlem	42	42	42

Chelsea	100	100	100
Chinatown	100	100	100
Civic Center	100	100	100
Clinton	100	100	100
East Harlem	43	43	43
East Village	100	100	100
Financial District	100	100	100
Flatiron	100	100	100
Gramercy	100	100	100
Greenwich Village	100	100	100
Hamilton Heights	61	61	61
Hudson Yards	75	75	75
Inwood	57	57	57
Lenox Hill	100	100	100
Lincoln Square	100	100	100
Little Italy	100	100	100
Lower East Side	60	60	60
Manhattan Valley	56	56	56
Manhattanville	40	40	40
Marble Hill	26	26	26
Midtown	100	100	100
Midtown South	100	100	100
Morningside Heights	40	40	40
Murray Hill	100	100	100
Noho	100	100	100
Roosevelt Island	29	29	29
Soho	100	100	100
Stuyvesant Town	17	17	17
Sutton Place	100	100	100
Tribeca	100	100	100
Tudor City	83	83	83
Turtle Bay	100	100	100
Upper East Side	100	100	100
Upper West Side	100	100	100
Washington Heights	87	87	87
West Village	100	100	100
Yorkville	100	100	100

Let's find out how many unique categories can be curated from all the returned venues

```
In [82]: print('There are {} uniques categories.'.format(len(ny_venues['Venue Category'].unique())))
```

There are 342 uniques categories.

### 1.4.2 Analyze Each Neighborhood

```
In [83]: # one hot encoding
ny_onehot = pd.get_dummies(ny_venues[['Venue Category']], prefix="", prefix_sep="")
```

```

# add neighborhood column back to dataframe
ny_onehot['Neighborhood'] = ny_venues['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [ny_onehot.columns[-1]] + list(ny_onehot.columns[:-1])
ny_onehot = ny_onehot[fixed_columns]

ny_onehot.head()

```

```

Out[83]:  Neighborhood  Accessories Store  Adult Boutique  Afghan Restaurant  \
0  Marble Hill          0                0                0
1  Marble Hill          0                0                0
2  Marble Hill          0                0                0
3  Marble Hill          0                0                0
4  Marble Hill          0                0                0

    African Restaurant  American Restaurant  Antique Shop  Arcade  \
0                0                0                0        0
1                0                0                0        0
2                0                0                0        0
3                0                0                0        0
4                0                0                0        0

    Arepa Restaurant  Argentinian Restaurant  Art Gallery  Art Museum  \
0                0                0                0        0
1                0                0                0        0
2                0                0                0        0
3                0                0                0        0
4                0                0                0        0

    Arts & Crafts Store  Asian Restaurant  Athletics & Sports  Auditorium  \
0                0                0                0        0
1                0                0                0        0
2                0                0                0        0
3                0                0                0        0
4                0                0                0        0

    Australian Restaurant  Austrian Restaurant  Auto Workshop  BBQ Joint  \
0                0                0                0        0
1                0                0                0        0
2                0                0                0        0
3                0                0                0        0
4                0                0                0        0

    Baby Store  Bagel Shop  Bakery  Bank  Bar  Baseball Field  \
0                0        0        0    0    0                0
1                0        0        0    0    0                0

```

2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0

	Basketball Court	Bed & Breakfast	Beer Bar	Beer Garden	Beer Store \
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

	Big Box Store	Bike Rental / Bike Share	Bike Shop	Bike Trail	Bistro \
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

	Board Shop	Boat or Ferry	Bookstore	Boutique	Boxing Gym \
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

	Brazilian Restaurant	Breakfast Spot	Bridal Shop	Bridge	Bubble Tea Shop \
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

	Building	Burger Joint	Burrito Place	Bus Line	Bus Station	Bus Stop \
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0

	Business Service	Butcher	Cafeteria	Café	Cambodian Restaurant \
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

	Camera Store	Candy Store	Cantonese Restaurant	Caribbean Restaurant \
0	0	0	0	0



1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

	Caucasian Restaurant	Cheese Shop	Chinese Restaurant	Chocolate Shop \
0		0		0
1		0		0
2		0		0
3		0		0
4		0		0

	Circus	Climbing Gym	Clothing Store	Club House	Cocktail Bar \
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

	Coffee Shop	College Academic Building	College Bookstore \
0	0		0
1	0		0
2	0		0
3	1		0
4	0		0

	College Cafeteria	College Theater	Comedy Club	Community Center \
0		0	0	0
1		0	0	0
2		0	0	0
3		0	0	0
4		0	0	0

	Concert Hall	Convenience Store	Cosmetics Shop	Coworking Space	Creperie \
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

	Cuban Restaurant	Cultural Center	Cupcake Shop	Cycle Studio \
0		0	0	0
1		0	0	0
2		0	0	0
3		0	0	0
4		0	0	0

	Czech Restaurant	Dance Studio	Daycare	Deli / Bodega	Department Store \
--	------------------	--------------	---------	---------------	--------------------

0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

	Design Studio	Dessert Shop	Dim Sum Restaurant	Diner	Discount Store	\
0	0	0		0	0	0
1	0	0		0	0	0
2	0	0		0	1	0
3	0	0		0	0	0
4	0	0		0	0	0

	Dive Bar	Doctor's Office	Dog Run	Donut Shop	Drugstore	Dry Cleaner	\
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0

	Dumpling Restaurant	Duty-free Shop	Eastern European Restaurant	\
0	0	0		0
1	0	0		0
2	0	0		0
3	0	0		0
4	0	0		0

	Electronics Store	Empanada Restaurant	English Restaurant	\
0	0		0	0
1	0		0	0
2	0		0	0
3	0		0	0
4	0		0	0

	Ethiopian Restaurant	Event Space	Exhibit	Falafel Restaurant	\
0	0	0	0		0
1	0	0	0		0
2	0	0	0		0
3	0	0	0		0
4	0	0	0		0

	Farmers Market	Fast Food Restaurant	Field	Filipino Restaurant	\
0	0		0	0	0
1	0		0	0	0
2	0		0	0	0
3	0		0	0	0
4	0		0	0	0

	Financial or Legal Service	Fish Market	Flea Market	Flower Shop	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Food & Drink Shop	Food Court	Food Stand	Food Truck	Fountain	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	French Restaurant	Fried Chicken Joint	Frozen Yogurt Shop	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	Furniture / Home Store	Gaming Cafe	Garden	Garden Center	Gas Station	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	Gastropub	Gay Bar	General College & University	General Entertainment	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	German Restaurant	Gift Shop	Golf Course	Gourmet Shop	Greek Restaurant	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	Grocery Store	Gym	Gym / Fitness Center	Gym Pool	Gymnastics Gym	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	Harbor / Marina	Hardware Store	Hawaiian Restaurant	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	Health & Beauty Service	Health Food Store	Heliport	Herbs & Spices Store	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	High School	Himalayan Restaurant	Historic Site	History Museum	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Hobby Shop	Hookah Bar	Hostel	Hot Dog Joint	Hotel	Hotel Bar	\
0	0	0	0	0	0	0	
1	0	0	0	0	0	0	
2	0	0	0	0	0	0	
3	0	0	0	0	0	0	
4	0	0	0	0	0	0	

	Hotpot Restaurant	Ice Cream Shop	Indian Restaurant	Indie Movie Theater	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Indie Theater	Indoor Play Area	Irish Pub	Israeli Restaurant	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Italian Restaurant	Japanese Curry Restaurant	Japanese Restaurant	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	

4	0	0	0
---	---	---	---

	Jazz Club	Jewelry Store	Jewish Restaurant	Juice Bar	Karaoke Bar	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	Kebab Restaurant	Kids Store	Korean Restaurant	Kosher Restaurant	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Latin American Restaurant	Laundromat	Laundry Service	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	Leather Goods Store	Lebanese Restaurant	Library	Lingerie Store	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Liquor Store	Lounge	Malay Restaurant	Market	Martial Arts Dojo	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	Massage Studio	Medical Center	Mediterranean Restaurant	Memorial Site	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Men's Store	Metro Station	Mexican Restaurant	Middle Eastern Restaurant	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	

3	0	0	0	0
4	0	0	0	0

	Mini Golf	Miscellaneous Shop	Mobile Phone Shop	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	Modern European Restaurant	Molecular Gastronomy Restaurant	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	

	Monument / Landmark	Moroccan Restaurant	Movie Theater	Museum	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Music School	Music Venue	Nail Salon	New American Restaurant	Newsstand	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	Nightclub	Non-Profit	Noodle House	North Indian Restaurant	Office	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	Opera House	Optical Shop	Organic Grocery	Other Nightlife	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Outdoor Sculpture	Outdoors & Recreation	Paella Restaurant	\
0	0	0	0	
1	0	0	0	

2	0	0	0
3	0	0	0
4	0	0	0

	Pakistani Restaurant	Paper / Office Supplies Store	Park	Pastry Shop \
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

	Performing Arts Venue	Persian Restaurant	Peruvian Restaurant	Pet Café \
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

	Pet Service	Pet Store	Pharmacy	Photography Studio	Piano Bar	Pie Shop \
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0

	Pilates Studio	Pizza Place	Playground	Plaza	Poke Place	Pool	Pub \
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0

	Public Art	Ramen Restaurant	Record Shop	Recreation Center \
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

	Rental Car Location	Residential Building (Apartment / Condo)	Resort \
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

	Rest Area	Restaurant	River	Rock Club	Roof Deck	Russian Restaurant \
0	0	0	0	0	0	0

1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0

	Sake Bar	Salad Place	Salon / Barbershop	Sandwich Place	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Scandinavian Restaurant	Scenic Lookout	School	Sculpture Garden	\
0		0	0	0	
1		0	0	0	
2		0	0	0	
3		0	0	0	
4		0	0	0	

	Seafood Restaurant	Shanghai Restaurant	Shipping Store	Shoe Repair	\
0		0	0	0	
1		0	0	0	
2		0	0	0	
3		0	0	0	
4		0	0	0	

	Shoe Store	Shopping Mall	Skate Park	Ski Shop	Smoke Shop	Smoothie Shop	\
0	0	0	0	0	0	0	
1	0	0	0	0	0	0	
2	0	0	0	0	0	0	
3	0	0	0	0	0	0	
4	0	0	0	0	0	0	

	Snack Place	Soba Restaurant	Soccer Field	Social Club	Soup Place	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	South American Restaurant	South Indian Restaurant	\
0		0	
1		0	
2		0	
3		0	
4		0	

	Southern / Soul Food Restaurant	Spa	Spanish Restaurant	Speakeasy	\
--	---------------------------------	-----	--------------------	-----------	---



0				0	0		0	0
1				0	0		0	0
2				0	0		0	0
3				0	0		0	0
4				0	0		0	0
	Spiritual Center	Sporting Goods Shop	Sports Bar	Sports Club	Stables	\		
0	0	0	0	0	0	0		
1	0	0	0	0	0	0		
2	0	0	0	0	0	0		
3	0	0	0	0	0	0		
4	0	0	0	0	0	0		
	Steakhouse	Street Art	Strip Club	Supermarket	Supplement Shop	\		
0	0	0	0	0	0	0		
1	0	0	0	0	0	0		
2	0	0	0	0	0	0		
3	0	0	0	0	0	0		
4	0	0	0	0	0	0		
	Sushi Restaurant	Swiss Restaurant	Szechuan Restaurant	Taco Place	\			
0	0	0	0	0	0	0		
1	0	0	0	0	0	0		
2	0	0	0	0	0	0		
3	0	0	0	0	0	0		
4	0	0	0	0	0	0		
	Tailor Shop	Taiwanese Restaurant	Tapas Restaurant	Tattoo Parlor	\			
0	0	0	0	0	0	0		
1	0	0	0	0	0	0		
2	0	0	0	0	0	0		
3	0	0	0	0	0	0		
4	0	0	0	0	0	0		
	Tea Room	Tech Startup	Tennis Court	Tennis Stadium	Thai Restaurant	\		
0	0	0	0	0	0	0		
1	0	0	0	0	0	0		
2	0	0	0	0	0	0		
3	0	0	0	0	0	0		
4	0	0	0	0	0	0		
	Theater	Theme Park Ride / Attraction	Thrift / Vintage Store	Tiki Bar	\			
0	0	0	0	0	0	0		
1	0	0	0	0	0	0		
2	0	0	0	0	0	0		
3	0	0	0	0	0	0		
4	0	0	0	0	0	0		

	Tourist Information Center	Toy / Game Store	Trail	Tree	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Turkish Restaurant	Udon Restaurant	Used Bookstore	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	Vegetarian / Vegan Restaurant	Venezuelan Restaurant	Veterinarian	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	Video Game Store	Video Store	Vietnamese Restaurant	Volleyball Court	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Watch Shop	Waterfront	Weight Loss Center	Whisky Bar	Wine Bar	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	Wine Shop	Wings Joint	Women's Store	Yoga Studio
0	0	0	0	0
1	0	0	0	1
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

In [84]: ny\_onehot.shape

Out[84]: (3315, 343)

Next, let's group rows by neighborhood and by taking the mean of the frequency of occurrence of each category

```
In [85]: ny_grouped = ny_onehot.groupby('Neighborhood').mean().reset_index()
ny_grouped.head(10)
```

```
Out[85]:
```

	Neighborhood	Accessories Store	Adult Boutique	Afghan Restaurant	\
0	Battery Park City	0.00	0.0	0.0	
1	Carnegie Hill	0.00	0.0	0.0	
2	Central Harlem	0.00	0.0	0.0	
3	Chelsea	0.00	0.0	0.0	
4	Chinatown	0.00	0.0	0.0	
5	Civic Center	0.00	0.0	0.0	
6	Clinton	0.00	0.0	0.0	
7	East Harlem	0.00	0.0	0.0	
8	East Village	0.00	0.0	0.0	
9	Financial District	0.01	0.0	0.0	

	African Restaurant	American Restaurant	Antique Shop	Arcade	\
0	0.000000	0.010101	0.00	0.0	
1	0.000000	0.010000	0.00	0.0	
2	0.071429	0.047619	0.00	0.0	
3	0.000000	0.030000	0.01	0.0	
4	0.000000	0.040000	0.00	0.0	
5	0.000000	0.030000	0.01	0.0	
6	0.000000	0.040000	0.00	0.0	
7	0.000000	0.000000	0.00	0.0	
8	0.000000	0.020000	0.01	0.0	
9	0.000000	0.030000	0.00	0.0	

	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	\
0	0.00	0.00	0.00000	0.00	
1	0.00	0.01	0.00000	0.01	
2	0.00	0.00	0.02381	0.00	
3	0.00	0.00	0.03000	0.00	
4	0.00	0.00	0.00000	0.00	
5	0.00	0.00	0.02000	0.00	
6	0.00	0.00	0.01000	0.00	
7	0.00	0.00	0.00000	0.00	
8	0.02	0.01	0.01000	0.00	
9	0.00	0.00	0.00000	0.00	

	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Auditorium	\
0	0.00	0.00	0.010101	0.010101	
1	0.00	0.00	0.000000	0.000000	
2	0.00	0.00	0.000000	0.000000	
3	0.00	0.01	0.000000	0.000000	
4	0.00	0.02	0.000000	0.000000	
5	0.00	0.01	0.000000	0.000000	
6	0.00	0.00	0.000000	0.000000	
7	0.00	0.00	0.000000	0.000000	

8	0.01	0.00	0.000000	0.000000
9	0.00	0.00	0.000000	0.000000

	Australian Restaurant	Austrian Restaurant	Auto Workshop	BBQ Joint	\
0	0.00	0.00	0.0	0.020202	
1	0.00	0.00	0.0	0.000000	
2	0.00	0.00	0.0	0.023810	
3	0.00	0.00	0.0	0.000000	
4	0.00	0.01	0.0	0.000000	
5	0.01	0.00	0.0	0.000000	
6	0.00	0.00	0.0	0.000000	
7	0.00	0.00	0.0	0.000000	
8	0.00	0.00	0.0	0.000000	
9	0.00	0.00	0.0	0.000000	

	Baby Store	Bagel Shop	Bakery	Bank	Bar	Baseball Field	\
0	0.00	0.000000	0.010101	0.0	0.000000	0.0	
1	0.00	0.01000	0.030000	0.0	0.040000	0.0	
2	0.00	0.02381	0.000000	0.0	0.047619	0.0	
3	0.00	0.000000	0.040000	0.0	0.010000	0.0	
4	0.00	0.000000	0.030000	0.0	0.010000	0.0	
5	0.01	0.01000	0.030000	0.0	0.010000	0.0	
6	0.00	0.000000	0.010000	0.0	0.020000	0.0	
7	0.00	0.000000	0.093023	0.0	0.000000	0.0	
8	0.00	0.02000	0.010000	0.0	0.060000	0.0	
9	0.00	0.01000	0.010000	0.0	0.030000	0.0	

	Basketball Court	Bed & Breakfast	Beer Bar	Beer Garden	Beer Store	\
0	0.0	0.0	0.000000	0.010101	0.00	
1	0.0	0.0	0.000000	0.000000	0.00	
2	0.0	0.0	0.023810	0.000000	0.00	
3	0.0	0.0	0.010000	0.000000	0.00	
4	0.0	0.0	0.000000	0.000000	0.00	
5	0.0	0.0	0.000000	0.000000	0.00	
6	0.0	0.0	0.000000	0.000000	0.00	
7	0.0	0.0	0.023256	0.000000	0.00	
8	0.0	0.0	0.000000	0.000000	0.01	
9	0.0	0.0	0.000000	0.000000	0.00	

	Big Box Store	Bike Rental / Bike Share	Bike Shop	Bike Trail	Bistro	\
0	0.0	0.0	0.00	0.0	0.010101	
1	0.0	0.0	0.00	0.0	0.000000	
2	0.0	0.0	0.00	0.0	0.000000	
3	0.0	0.0	0.00	0.0	0.000000	
4	0.0	0.0	0.01	0.0	0.000000	
5	0.0	0.0	0.00	0.0	0.000000	
6	0.0	0.0	0.00	0.0	0.000000	
7	0.0	0.0	0.00	0.0	0.000000	

8	0.0	0.0	0.00	0.0	0.000000
9	0.0	0.0	0.00	0.0	0.000000

	Board Shop	Boat or Ferry	Bookstore	Boutique	Boxing Gym \
0	0.0	0.030303	0.00000	0.00000	0.00
1	0.0	0.000000	0.03000	0.00000	0.00
2	0.0	0.000000	0.02381	0.02381	0.00
3	0.0	0.000000	0.02000	0.00000	0.00
4	0.0	0.000000	0.01000	0.00000	0.00
5	0.0	0.000000	0.01000	0.00000	0.01
6	0.0	0.000000	0.00000	0.00000	0.00
7	0.0	0.000000	0.00000	0.00000	0.00
8	0.0	0.000000	0.00000	0.00000	0.00
9	0.0	0.000000	0.00000	0.00000	0.00

	Brazilian Restaurant	Breakfast Spot	Bridal Shop	Bridge	Bubble Tea Shop \
0		0.0	0.0	0.0	0.00
1		0.0	0.0	0.0	0.00
2		0.0	0.0	0.0	0.00
3		0.0	0.0	0.0	0.00
4		0.0	0.0	0.0	0.03
5		0.0	0.0	0.0	0.01
6		0.0	0.0	0.0	0.00
7		0.0	0.0	0.0	0.00
8		0.0	0.0	0.0	0.00
9		0.0	0.0	0.0	0.00

	Building	Burger Joint	Burrito Place	Bus Line	Bus Station	Bus Stop \
0	0.010101	0.020202	0.010101	0.0	0.0	0.0
1	0.000000	0.010000	0.000000	0.0	0.0	0.0
2	0.000000	0.000000	0.000000	0.0	0.0	0.0
3	0.000000	0.000000	0.000000	0.0	0.0	0.0
4	0.000000	0.000000	0.000000	0.0	0.0	0.0
5	0.010000	0.000000	0.010000	0.0	0.0	0.0
6	0.010000	0.010000	0.000000	0.0	0.0	0.0
7	0.000000	0.000000	0.000000	0.0	0.0	0.0
8	0.000000	0.010000	0.000000	0.0	0.0	0.0
9	0.000000	0.010000	0.000000	0.0	0.0	0.0

	Business Service	Butcher	Cafeteria	Café	Cambodian Restaurant \
0		0.0	0.00000	0.000000	0.0
1		0.0	0.00000	0.040000	0.0
2		0.0	0.02381	0.023810	0.0
3		0.0	0.00000	0.010000	0.0
4		0.0	0.00000	0.010000	0.0
5		0.0	0.00000	0.020000	0.0
6		0.0	0.00000	0.010000	0.0
7		0.0	0.00000	0.023256	0.0

8	0.0	0.00	0.00000	0.000000	0.0
9	0.0	0.00	0.00000	0.020000	0.0

	Camera Store	Candy Store	Cantonese Restaurant	Caribbean Restaurant	\
0	0.0	0.0	0.00	0.00000	
1	0.0	0.0	0.00	0.00000	
2	0.0	0.0	0.00	0.02381	
3	0.0	0.0	0.00	0.00000	
4	0.0	0.0	0.01	0.00000	
5	0.0	0.0	0.00	0.00000	
6	0.0	0.0	0.00	0.00000	
7	0.0	0.0	0.00	0.00000	
8	0.0	0.0	0.00	0.01000	
9	0.0	0.0	0.00	0.00000	

	Caucasian Restaurant	Cheese Shop	Chinese Restaurant	Chocolate Shop	\
0	0.00	0.00	0.010101	0.0	
1	0.00	0.00	0.000000	0.0	
2	0.00	0.00	0.047619	0.0	
3	0.00	0.00	0.010000	0.0	
4	0.00	0.00	0.090000	0.0	
5	0.00	0.00	0.000000	0.0	
6	0.01	0.00	0.010000	0.0	
7	0.00	0.00	0.023256	0.0	
8	0.00	0.01	0.040000	0.0	
9	0.00	0.00	0.010000	0.0	

	Circus	Climbing Gym	Clothing Store	Club House	Cocktail Bar	\
0	0.0	0.0	0.020202	0.0	0.000000	
1	0.0	0.0	0.010000	0.0	0.010000	
2	0.0	0.0	0.000000	0.0	0.000000	
3	0.0	0.0	0.000000	0.0	0.010000	
4	0.0	0.0	0.000000	0.0	0.050000	
5	0.0	0.0	0.010000	0.0	0.030000	
6	0.0	0.0	0.000000	0.0	0.020000	
7	0.0	0.0	0.023256	0.0	0.023256	
8	0.0	0.0	0.000000	0.0	0.030000	
9	0.0	0.0	0.000000	0.0	0.030000	

	Coffee Shop	College Academic Building	College Bookstore	\
0	0.060606	0.0	0.0	
1	0.070000	0.0	0.0	
2	0.000000	0.0	0.0	
3	0.060000	0.0	0.0	
4	0.010000	0.0	0.0	
5	0.040000	0.0	0.0	
6	0.030000	0.0	0.0	
7	0.023256	0.0	0.0	

8	0.030000	0.0	0.0
9	0.080000	0.0	0.0

	College Cafeteria	College Theater	Comedy Club	Community Center	\
0	0.0	0.00	0.00	0.00	
1	0.0	0.00	0.00	0.01	
2	0.0	0.00	0.00	0.00	
3	0.0	0.01	0.00	0.00	
4	0.0	0.00	0.00	0.00	
5	0.0	0.00	0.00	0.00	
6	0.0	0.00	0.01	0.00	
7	0.0	0.00	0.00	0.00	
8	0.0	0.00	0.00	0.00	
9	0.0	0.00	0.00	0.00	

	Concert Hall	Convenience Store	Cosmetics Shop	Coworking Space	Creperie	\
0	0.00	0.000000	0.010101	0.00	0.00	
1	0.01	0.000000	0.030000	0.00	0.00	
2	0.00	0.000000	0.023810	0.00	0.00	
3	0.00	0.000000	0.010000	0.00	0.01	
4	0.00	0.000000	0.010000	0.00	0.00	
5	0.00	0.000000	0.010000	0.00	0.00	
6	0.01	0.000000	0.000000	0.00	0.00	
7	0.00	0.023256	0.000000	0.00	0.00	
8	0.00	0.000000	0.000000	0.00	0.00	
9	0.00	0.000000	0.000000	0.01	0.00	

	Cuban Restaurant	Cultural Center	Cupcake Shop	Cycle Studio	\
0	0.000000	0.0	0.020202	0.000000	
1	0.000000	0.0	0.000000	0.000000	
2	0.000000	0.0	0.000000	0.02381	
3	0.000000	0.0	0.020000	0.02000	
4	0.000000	0.0	0.000000	0.000000	
5	0.010000	0.0	0.000000	0.000000	
6	0.000000	0.0	0.000000	0.000000	
7	0.023256	0.0	0.000000	0.000000	
8	0.000000	0.0	0.010000	0.000000	
9	0.010000	0.0	0.000000	0.01000	

	Czech Restaurant	Dance Studio	Daycare	Deli / Bodega	Department Store	\
0	0.0	0.000000	0.0	0.000000	0.020202	
1	0.0	0.010000	0.0	0.000000	0.000000	
2	0.0	0.000000	0.0	0.000000	0.000000	
3	0.0	0.000000	0.0	0.000000	0.000000	
4	0.0	0.000000	0.0	0.000000	0.000000	
5	0.0	0.020000	0.0	0.010000	0.000000	
6	0.0	0.000000	0.0	0.000000	0.000000	
7	0.0	0.023256	0.0	0.069767	0.000000	

8	0.0	0.000000	0.0	0.010000	0.000000
9	0.0	0.000000	0.0	0.010000	0.000000

	Design Studio	Dessert Shop	Dim Sum Restaurant	Diner	Discount Store	\
0	0.0	0.00000	0.00	0.00	0.0	
1	0.0	0.00000	0.00	0.00	0.0	
2	0.0	0.02381	0.00	0.00	0.0	
3	0.0	0.01000	0.00	0.00	0.0	
4	0.0	0.00000	0.02	0.01	0.0	
5	0.0	0.00000	0.00	0.01	0.0	
6	0.0	0.00000	0.01	0.01	0.0	
7	0.0	0.00000	0.00	0.00	0.0	
8	0.0	0.02000	0.00	0.00	0.0	
9	0.0	0.00000	0.00	0.00	0.0	

	Dive Bar	Doctor's Office	Dog Run	Donut Shop	Drugstore	Dry Cleaner	\
0	0.00	0.00	0.00	0.000000	0.0	0.0	
1	0.00	0.01	0.00	0.000000	0.0	0.0	
2	0.00	0.00	0.00	0.000000	0.0	0.0	
3	0.00	0.00	0.00	0.000000	0.0	0.0	
4	0.00	0.00	0.00	0.000000	0.0	0.0	
5	0.00	0.00	0.00	0.000000	0.0	0.0	
6	0.01	0.00	0.01	0.000000	0.0	0.0	
7	0.00	0.00	0.00	0.023256	0.0	0.0	
8	0.00	0.00	0.01	0.000000	0.0	0.0	
9	0.00	0.01	0.00	0.000000	0.0	0.0	

	Dumpling Restaurant	Duty-free Shop	Eastern European Restaurant	\
0	0.00	0.0	0.0	
1	0.00	0.0	0.0	
2	0.00	0.0	0.0	
3	0.00	0.0	0.0	
4	0.03	0.0	0.0	
5	0.00	0.0	0.0	
6	0.00	0.0	0.0	
7	0.00	0.0	0.0	
8	0.00	0.0	0.0	
9	0.00	0.0	0.0	

	Electronics Store	Empanada Restaurant	English Restaurant	\
0	0.010101	0.0	0.00	
1	0.000000	0.0	0.00	
2	0.000000	0.0	0.00	
3	0.010000	0.0	0.00	
4	0.000000	0.0	0.01	
5	0.000000	0.0	0.00	
6	0.000000	0.0	0.00	
7	0.000000	0.0	0.00	



8	0.000000	0.0	0.00
9	0.000000	0.0	0.00

	Ethiopian Restaurant	Event Space	Exhibit	Falafel Restaurant \
0	0.00000	0.00000	0.00	0.00
1	0.00000	0.00000	0.01	0.00
2	0.02381	0.02381	0.00	0.00
3	0.00000	0.00000	0.00	0.00
4	0.00000	0.00000	0.00	0.00
5	0.00000	0.00000	0.00	0.02
6	0.00000	0.00000	0.00	0.00
7	0.00000	0.00000	0.00	0.00
8	0.00000	0.00000	0.00	0.00
9	0.00000	0.03000	0.00	0.02

	Farmers Market	Fast Food Restaurant	Field	Filipino Restaurant \
0	0.00		0.0	0.00
1	0.00		0.0	0.00
2	0.00		0.0	0.00
3	0.00		0.0	0.00
4	0.00		0.0	0.00
5	0.00		0.0	0.00
6	0.00		0.0	0.00
7	0.00		0.0	0.00
8	0.01		0.0	0.01
9	0.01		0.0	0.00

	Financial or Legal Service	Fish Market	Flea Market	Flower Shop \
0		0.0	0.00	0.00
1		0.0	0.00	0.00
2		0.0	0.00	0.00
3		0.0	0.01	0.00
4		0.0	0.00	0.00
5		0.0	0.00	0.00
6		0.0	0.00	0.00
7		0.0	0.00	0.00
8		0.0	0.00	0.01
9		0.0	0.00	0.00

	Food & Drink Shop	Food Court	Food Stand	Food Truck	Fountain \
0	0.0	0.020202	0.0	0.010101	0.0
1	0.0	0.000000	0.0	0.010000	0.0
2	0.0	0.000000	0.0	0.000000	0.0
3	0.0	0.000000	0.0	0.000000	0.0
4	0.0	0.000000	0.0	0.000000	0.0
5	0.0	0.000000	0.0	0.000000	0.0
6	0.0	0.020000	0.0	0.000000	0.0
7	0.0	0.000000	0.0	0.000000	0.0

8	0.0	0.000000	0.0	0.000000	0.0
9	0.0	0.010000	0.0	0.020000	0.0

	French Restaurant	Fried Chicken Joint	Frozen Yogurt Shop \
0	0.000000	0.00000	0.0
1	0.020000	0.00000	0.0
2	0.047619	0.02381	0.0
3	0.020000	0.00000	0.0
4	0.000000	0.00000	0.0
5	0.040000	0.00000	0.0
6	0.010000	0.01000	0.0
7	0.023256	0.00000	0.0
8	0.010000	0.00000	0.0
9	0.010000	0.01000	0.0

	Furniture / Home Store	Gaming Cafe	Garden	Garden Center	Gas Station \
0	0.00	0.0	0.010101	0.00	0.000000
1	0.00	0.0	0.000000	0.00	0.000000
2	0.00	0.0	0.000000	0.00	0.000000
3	0.00	0.0	0.000000	0.00	0.000000
4	0.01	0.0	0.000000	0.01	0.000000
5	0.01	0.0	0.000000	0.00	0.000000
6	0.00	0.0	0.000000	0.00	0.000000
7	0.00	0.0	0.000000	0.00	0.023256
8	0.00	0.0	0.000000	0.01	0.000000
9	0.00	0.0	0.000000	0.00	0.000000

	Gastropub	Gay Bar	General College & University	General Entertainment \
0	0.010101	0.0	0.0	0.00
1	0.000000	0.0	0.0	0.00
2	0.000000	0.0	0.0	0.00
3	0.000000	0.0	0.0	0.00
4	0.000000	0.0	0.0	0.01
5	0.000000	0.0	0.0	0.01
6	0.000000	0.0	0.0	0.00
7	0.000000	0.0	0.0	0.00
8	0.000000	0.0	0.0	0.00
9	0.000000	0.0	0.0	0.00

	German Restaurant	Gift Shop	Golf Course	Gourmet Shop	Greek Restaurant \
0	0.0	0.00	0.0	0.010101	0.00
1	0.0	0.01	0.0	0.010000	0.00
2	0.0	0.00	0.0	0.000000	0.00
3	0.0	0.01	0.0	0.000000	0.00
4	0.0	0.01	0.0	0.000000	0.01
5	0.0	0.00	0.0	0.010000	0.00
6	0.0	0.01	0.0	0.000000	0.00
7	0.0	0.00	0.0	0.000000	0.00

8	0.0	0.01	0.0	0.010000	0.01
9	0.0	0.00	0.0	0.000000	0.01

	Grocery Store	Gym	Gym / Fitness Center	Gym Pool	Gymnastics Gym \
0	0.020202	0.040404	0.00000	0.0	0.0
1	0.030000	0.030000	0.02000	0.0	0.0
2	0.000000	0.023810	0.02381	0.0	0.0
3	0.010000	0.010000	0.01000	0.0	0.0
4	0.000000	0.010000	0.00000	0.0	0.0
5	0.000000	0.020000	0.05000	0.0	0.0
6	0.010000	0.020000	0.05000	0.0	0.0
7	0.023256	0.023256	0.00000	0.0	0.0
8	0.000000	0.000000	0.00000	0.0	0.0
9	0.000000	0.040000	0.03000	0.0	0.0

	Harbor / Marina	Hardware Store	Hawaiian Restaurant \
0	0.0	0.0	0.0
1	0.0	0.0	0.0
2	0.0	0.0	0.0
3	0.0	0.0	0.0
4	0.0	0.0	0.0
5	0.0	0.0	0.0
6	0.0	0.0	0.0
7	0.0	0.0	0.0
8	0.0	0.0	0.0
9	0.0	0.0	0.0

	Health & Beauty Service	Health Food Store	Heliport	Herbs & Spices Store \
0	0.00	0.00	0.0	0.0
1	0.00	0.00	0.0	0.0
2	0.00	0.00	0.0	0.0
3	0.02	0.00	0.0	0.0
4	0.00	0.00	0.0	0.0
5	0.01	0.00	0.0	0.0
6	0.00	0.01	0.0	0.0
7	0.00	0.00	0.0	0.0
8	0.00	0.00	0.0	0.0
9	0.00	0.00	0.0	0.0

	High School	Himalayan Restaurant	Historic Site	History Museum \
0	0.0	0.0	0.010101	0.0
1	0.0	0.0	0.000000	0.0
2	0.0	0.0	0.000000	0.0
3	0.0	0.0	0.000000	0.0
4	0.0	0.0	0.010000	0.0
5	0.0	0.0	0.000000	0.0
6	0.0	0.0	0.000000	0.0
7	0.0	0.0	0.023256	0.0

8	0.0	0.0	0.000000	0.0
9	0.0	0.0	0.010000	0.0

	Hobby Shop	Hookah Bar	Hostel	Hot Dog Joint	Hotel	Hotel Bar	\
0	0.00	0.0	0.0	0.00	0.050505	0.00	
1	0.00	0.0	0.0	0.01	0.010000	0.00	
2	0.00	0.0	0.0	0.00	0.000000	0.00	
3	0.00	0.0	0.0	0.00	0.030000	0.00	
4	0.00	0.0	0.0	0.00	0.010000	0.00	
5	0.00	0.0	0.0	0.00	0.040000	0.02	
6	0.00	0.0	0.0	0.00	0.040000	0.01	
7	0.00	0.0	0.0	0.00	0.000000	0.00	
8	0.01	0.0	0.0	0.01	0.000000	0.00	
9	0.00	0.0	0.0	0.00	0.040000	0.00	

	Hotpot Restaurant	Ice Cream Shop	Indian Restaurant	Indie Movie Theater	\
0	0.00	0.020202	0.00	0.00	
1	0.00	0.010000	0.02	0.00	
2	0.00	0.000000	0.00	0.00	
3	0.00	0.050000	0.01	0.00	
4	0.02	0.030000	0.00	0.01	
5	0.00	0.000000	0.01	0.00	
6	0.00	0.010000	0.00	0.00	
7	0.00	0.000000	0.00	0.00	
8	0.00	0.040000	0.00	0.00	
9	0.00	0.010000	0.01	0.00	

	Indie Theater	Indoor Play Area	Irish Pub	Israeli Restaurant	\
0	0.00	0.0	0.0	0.00	
1	0.00	0.0	0.0	0.00	
2	0.00	0.0	0.0	0.00	
3	0.01	0.0	0.0	0.01	
4	0.00	0.0	0.0	0.00	
5	0.00	0.0	0.0	0.00	
6	0.02	0.0	0.0	0.00	
7	0.00	0.0	0.0	0.00	
8	0.00	0.0	0.0	0.00	
9	0.00	0.0	0.0	0.00	

	Italian Restaurant	Japanese Curry Restaurant	Japanese Restaurant	\
0	0.030303	0.00	0.00	
1	0.020000	0.00	0.03	
2	0.000000	0.00	0.00	
3	0.050000	0.00	0.01	
4	0.010000	0.00	0.00	
5	0.040000	0.00	0.00	
6	0.040000	0.00	0.00	
7	0.000000	0.00	0.00	

8	0.010000	0.00	0.02
9	0.020000	0.01	0.02

	Jazz Club	Jewelry Store	Jewish Restaurant	Juice Bar	Karaoke Bar	\
0	0.00000	0.00	0.0	0.010101	0.0	
1	0.00000	0.00	0.0	0.000000	0.0	
2	0.02381	0.00	0.0	0.023810	0.0	
3	0.00000	0.00	0.0	0.000000	0.0	
4	0.00000	0.01	0.0	0.000000	0.0	
5	0.00000	0.00	0.0	0.000000	0.0	
6	0.00000	0.00	0.0	0.000000	0.0	
7	0.00000	0.00	0.0	0.000000	0.0	
8	0.00000	0.00	0.0	0.010000	0.0	
9	0.00000	0.02	0.0	0.020000	0.0	

	Kebab Restaurant	Kids Store	Korean Restaurant	Kosher Restaurant	\
0	0.0	0.00	0.00	0.00	0.00
1	0.0	0.00	0.00	0.00	0.01
2	0.0	0.00	0.00	0.00	0.00
3	0.0	0.00	0.00	0.00	0.00
4	0.0	0.00	0.02	0.00	0.00
5	0.0	0.01	0.00	0.00	0.00
6	0.0	0.00	0.00	0.00	0.00
7	0.0	0.00	0.00	0.00	0.00
8	0.0	0.00	0.02	0.00	0.00
9	0.0	0.00	0.00	0.00	0.00

	Latin American Restaurant	Laundromat	Laundry Service	\
0	0.000000	0.0	0.00	
1	0.000000	0.0	0.00	
2	0.000000	0.0	0.00	
3	0.000000	0.0	0.00	
4	0.000000	0.0	0.00	
5	0.000000	0.0	0.01	
6	0.000000	0.0	0.00	
7	0.046512	0.0	0.00	
8	0.000000	0.0	0.00	
9	0.000000	0.0	0.00	

	Leather Goods Store	Lebanese Restaurant	Library	Lingerie Store	\
0	0.0	0.0	0.00000	0.010101	
1	0.0	0.0	0.00000	0.000000	
2	0.0	0.0	0.02381	0.000000	
3	0.0	0.0	0.00000	0.000000	
4	0.0	0.0	0.00000	0.000000	
5	0.0	0.0	0.00000	0.000000	
6	0.0	0.0	0.00000	0.000000	
7	0.0	0.0	0.00000	0.000000	

8	0.0	0.0	0.00000	0.000000
9	0.0	0.0	0.00000	0.000000

	Liquor Store	Lounge	Malay Restaurant	Market	Martial Arts Dojo \
0	0.000000	0.00	0.00	0.010101	0.00
1	0.000000	0.00	0.00	0.000000	0.01
2	0.000000	0.00	0.00	0.000000	0.00
3	0.000000	0.00	0.00	0.020000	0.00
4	0.000000	0.00	0.02	0.000000	0.00
5	0.000000	0.00	0.00	0.000000	0.01
6	0.000000	0.02	0.00	0.000000	0.00
7	0.023256	0.00	0.00	0.000000	0.00
8	0.000000	0.00	0.00	0.000000	0.00
9	0.000000	0.00	0.00	0.010000	0.00

	Massage Studio	Medical Center	Mediterranean Restaurant	Memorial Site \
0	0.00	0.00	0.010101	0.040404
1	0.00	0.00	0.010000	0.000000
2	0.00	0.00	0.000000	0.000000
3	0.00	0.00	0.000000	0.000000
4	0.01	0.00	0.000000	0.000000
5	0.00	0.01	0.000000	0.000000
6	0.00	0.00	0.020000	0.000000
7	0.00	0.00	0.000000	0.000000
8	0.00	0.00	0.000000	0.000000
9	0.00	0.00	0.020000	0.000000

	Men's Store	Metro Station	Mexican Restaurant	Middle Eastern Restaurant \
0	0.010101	0.00000	0.010101	0.00
1	0.000000	0.00000	0.010000	0.00
2	0.000000	0.02381	0.000000	0.00
3	0.010000	0.00000	0.010000	0.00
4	0.010000	0.00000	0.010000	0.00
5	0.000000	0.00000	0.000000	0.00
6	0.000000	0.00000	0.000000	0.00
7	0.000000	0.00000	0.116279	0.00
8	0.000000	0.00000	0.040000	0.01
9	0.000000	0.00000	0.010000	0.00

	Mini Golf	Miscellaneous Shop	Mobile Phone Shop \
0	0.0	0.0	0.00
1	0.0	0.0	0.01
2	0.0	0.0	0.00
3	0.0	0.0	0.00
4	0.0	0.0	0.00
5	0.0	0.0	0.00
6	0.0	0.0	0.00
7	0.0	0.0	0.00

8	0.0	0.0	0.00
9	0.0	0.0	0.00

	Modern European Restaurant	Molecular Gastronomy Restaurant	\
0	0.0	0.00	
1	0.0	0.00	
2	0.0	0.00	
3	0.0	0.00	
4	0.0	0.00	
5	0.0	0.01	
6	0.0	0.00	
7	0.0	0.00	
8	0.0	0.00	
9	0.0	0.00	

	Monument / Landmark	Moroccan Restaurant	Movie Theater	Museum	\
0	0.010101	0.00	0.010101	0.00	
1	0.000000	0.00	0.000000	0.01	
2	0.000000	0.00	0.000000	0.00	
3	0.000000	0.00	0.000000	0.00	
4	0.000000	0.00	0.000000	0.01	
5	0.010000	0.00	0.000000	0.01	
6	0.000000	0.00	0.010000	0.00	
7	0.000000	0.00	0.000000	0.00	
8	0.000000	0.01	0.000000	0.00	
9	0.020000	0.00	0.000000	0.01	

	Music School	Music Venue	Nail Salon	New American Restaurant	Newsstand	\
0	0.00	0.00000	0.00		0.00	0.0
1	0.00	0.00000	0.00		0.01	0.0
2	0.00	0.02381	0.00		0.00	0.0
3	0.00	0.00000	0.00		0.00	0.0
4	0.00	0.01000	0.00		0.01	0.0
5	0.00	0.00000	0.01		0.01	0.0
6	0.01	0.00000	0.00		0.02	0.0
7	0.00	0.00000	0.00		0.00	0.0
8	0.00	0.00000	0.00		0.00	0.0
9	0.00	0.00000	0.00		0.01	0.0

	Nightclub	Non-Profit	Noodle House	North Indian Restaurant	Office	\
0	0.00	0.0	0.00		0.0	0.00
1	0.00	0.0	0.00		0.0	0.00
2	0.00	0.0	0.00		0.0	0.00
3	0.04	0.0	0.01		0.0	0.01
4	0.00	0.0	0.02		0.0	0.00
5	0.00	0.0	0.00		0.0	0.00
6	0.00	0.0	0.00		0.0	0.00
7	0.00	0.0	0.00		0.0	0.00

8	0.00	0.0	0.00	0.0	0.00
9	0.00	0.0	0.00	0.0	0.00

	Opera House	Optical Shop	Organic Grocery	Other Nightlife	\
0	0.0	0.00	0.00	0.0	
1	0.0	0.00	0.00	0.0	
2	0.0	0.00	0.00	0.0	
3	0.0	0.00	0.00	0.0	
4	0.0	0.01	0.01	0.0	
5	0.0	0.00	0.00	0.0	
6	0.0	0.00	0.01	0.0	
7	0.0	0.00	0.00	0.0	
8	0.0	0.00	0.01	0.0	
9	0.0	0.00	0.00	0.0	

	Outdoor Sculpture	Outdoors & Recreation	Paella Restaurant	\
0	0.0	0.0	0.00	
1	0.0	0.0	0.00	
2	0.0	0.0	0.00	
3	0.0	0.0	0.01	
4	0.0	0.0	0.00	
5	0.0	0.0	0.00	
6	0.0	0.0	0.00	
7	0.0	0.0	0.00	
8	0.0	0.0	0.00	
9	0.0	0.0	0.00	

	Pakistani Restaurant	Paper / Office Supplies Store	Park	Pastry Shop	\
0	0.0	0.00	0.070707	0.0	
1	0.0	0.00	0.000000	0.0	
2	0.0	0.00	0.023810	0.0	
3	0.0	0.00	0.010000	0.0	
4	0.0	0.01	0.000000	0.0	
5	0.0	0.00	0.030000	0.0	
6	0.0	0.00	0.010000	0.0	
7	0.0	0.00	0.023256	0.0	
8	0.0	0.00	0.010000	0.0	
9	0.0	0.00	0.020000	0.0	

	Performing Arts Venue	Persian Restaurant	Peruvian Restaurant	Pet Café	\
0	0.010101	0.0	0.00	0.00	
1	0.000000	0.0	0.00	0.00	
2	0.000000	0.0	0.00	0.00	
3	0.000000	0.0	0.00	0.00	
4	0.000000	0.0	0.00	0.00	
5	0.000000	0.0	0.00	0.00	
6	0.000000	0.0	0.01	0.00	
7	0.023256	0.0	0.00	0.00	



8	0.000000	0.0	0.00	0.01
9	0.010000	0.0	0.00	0.00

	Pet Service	Pet Store	Pharmacy	Photography Studio	Piano Bar	Pie Shop \
0	0.0	0.010101	0.000000	0.00	0.0	0.00
1	0.0	0.010000	0.000000	0.00	0.0	0.00
2	0.0	0.000000	0.000000	0.00	0.0	0.00
3	0.0	0.010000	0.000000	0.01	0.0	0.00
4	0.0	0.000000	0.010000	0.00	0.0	0.00
5	0.0	0.000000	0.000000	0.00	0.0	0.00
6	0.0	0.000000	0.000000	0.00	0.0	0.01
7	0.0	0.023256	0.023256	0.00	0.0	0.00
8	0.0	0.000000	0.000000	0.00	0.0	0.00
9	0.0	0.010000	0.000000	0.00	0.0	0.00

	Pilates Studio	Pizza Place	Playground	Plaza	Poke Place	Pool \
0	0.0	0.020202	0.020202	0.020202	0.00	0.00
1	0.0	0.060000	0.010000	0.000000	0.00	0.00
2	0.0	0.023810	0.000000	0.000000	0.00	0.00
3	0.0	0.010000	0.000000	0.000000	0.01	0.01
4	0.0	0.010000	0.000000	0.000000	0.00	0.00
5	0.0	0.000000	0.010000	0.010000	0.00	0.00
6	0.0	0.010000	0.000000	0.000000	0.01	0.00
7	0.0	0.023256	0.000000	0.000000	0.00	0.00
8	0.0	0.040000	0.000000	0.000000	0.00	0.00
9	0.0	0.030000	0.000000	0.010000	0.00	0.00

	Pub	Public Art	Ramen Restaurant	Record Shop	Recreation Center \
0	0.010101	0.000000	0.00	0.00	0.0
1	0.020000	0.000000	0.01	0.00	0.0
2	0.000000	0.047619	0.00	0.00	0.0
3	0.000000	0.000000	0.00	0.00	0.0
4	0.000000	0.000000	0.01	0.01	0.0
5	0.000000	0.000000	0.00	0.00	0.0
6	0.010000	0.000000	0.01	0.00	0.0
7	0.000000	0.000000	0.00	0.00	0.0
8	0.000000	0.000000	0.03	0.02	0.0
9	0.010000	0.000000	0.00	0.00	0.0

	Rental Car Location	Residential Building (Apartment / Condo)	Resort \
0	0.00000		0.00 0.0
1	0.00000		0.00 0.0
2	0.02381		0.00 0.0
3	0.00000		0.00 0.0
4	0.00000		0.00 0.0
5	0.00000		0.00 0.0
6	0.00000		0.01 0.0
7	0.00000		0.00 0.0

8	0.00000	0.00	0.0
9	0.00000	0.00	0.0

	Rest Area	Restaurant	River	Rock Club	Roof Deck	Russian Restaurant	\
0	0.0	0.000000	0.0	0.0	0.00		0.0
1	0.0	0.010000	0.0	0.0	0.00		0.0
2	0.0	0.000000	0.0	0.0	0.00		0.0
3	0.0	0.000000	0.0	0.0	0.00		0.0
4	0.0	0.000000	0.0	0.0	0.01		0.0
5	0.0	0.000000	0.0	0.0	0.00		0.0
6	0.0	0.010000	0.0	0.0	0.01		0.0
7	0.0	0.023256	0.0	0.0	0.00		0.0
8	0.0	0.000000	0.0	0.0	0.00		0.0
9	0.0	0.010000	0.0	0.0	0.01		0.0

	Sake Bar	Salad Place	Salon / Barbershop	Sandwich Place	\
0	0.00	0.010101		0.020202	
1	0.00	0.010000		0.000000	
2	0.00	0.000000		0.000000	
3	0.00	0.000000		0.010000	
4	0.01	0.000000		0.020000	
5	0.00	0.010000		0.040000	
6	0.00	0.000000		0.030000	
7	0.00	0.000000		0.023256	
8	0.00	0.000000		0.000000	
9	0.00	0.010000		0.010000	

	Scandinavian Restaurant	Scenic Lookout	School	Sculpture Garden	\
0		0.00	0.010101	0.0	0.010101
1		0.00	0.000000	0.0	0.000000
2		0.00	0.000000	0.0	0.000000
3		0.00	0.010000	0.0	0.000000
4		0.00	0.000000	0.0	0.000000
5		0.00	0.000000	0.0	0.000000
6		0.00	0.000000	0.0	0.000000
7		0.00	0.000000	0.0	0.000000
8		0.01	0.000000	0.0	0.000000
9		0.00	0.000000	0.0	0.000000

	Seafood Restaurant	Shanghai Restaurant	Shipping Store	Shoe Repair	\
0	0.010101		0.0	0.00	0.00
1	0.000000		0.0	0.01	0.00
2	0.047619		0.0	0.00	0.00
3	0.030000		0.0	0.00	0.01
4	0.000000		0.0	0.00	0.00
5	0.000000		0.0	0.00	0.00
6	0.010000		0.0	0.00	0.00
7	0.000000		0.0	0.00	0.00

8	0.020000	0.0	0.00	0.00
9	0.010000	0.0	0.00	0.00

	Shoe Store	Shopping Mall	Skate Park	Ski Shop	Smoke Shop	Smoothie Shop \
0	0.00	0.020202	0.0	0.0	0.010101	0.00
1	0.01	0.000000	0.0	0.0	0.000000	0.00
2	0.00	0.000000	0.0	0.0	0.000000	0.00
3	0.00	0.000000	0.0	0.0	0.000000	0.01
4	0.01	0.000000	0.0	0.0	0.000000	0.00
5	0.00	0.000000	0.0	0.0	0.000000	0.00
6	0.00	0.000000	0.0	0.0	0.000000	0.00
7	0.00	0.000000	0.0	0.0	0.000000	0.00
8	0.00	0.000000	0.0	0.0	0.000000	0.00
9	0.01	0.000000	0.0	0.0	0.010000	0.00

	Snack Place	Soba Restaurant	Soccer Field	Social Club	Soup Place \
0	0.00	0.0	0.0	0.0	0.00
1	0.00	0.0	0.0	0.0	0.00
2	0.00	0.0	0.0	0.0	0.00
3	0.00	0.0	0.0	0.0	0.00
4	0.01	0.0	0.0	0.0	0.00
5	0.00	0.0	0.0	0.0	0.00
6	0.00	0.0	0.0	0.0	0.00
7	0.00	0.0	0.0	0.0	0.00
8	0.00	0.0	0.0	0.0	0.01
9	0.00	0.0	0.0	0.0	0.00

	South American Restaurant	South Indian Restaurant \
0	0.0	0.0
1	0.0	0.0
2	0.0	0.0
3	0.0	0.0
4	0.0	0.0
5	0.0	0.0
6	0.0	0.0
7	0.0	0.0
8	0.0	0.0
9	0.0	0.0

	Southern / Soul Food Restaurant	Spa	Spanish Restaurant	Speakeasy \
0	0.00000	0.000000	0.000000	0.00
1	0.00000	0.030000	0.000000	0.00
2	0.02381	0.023810	0.000000	0.00
3	0.00000	0.000000	0.000000	0.01
4	0.00000	0.030000	0.010000	0.00
5	0.00000	0.030000	0.000000	0.00
6	0.00000	0.030000	0.000000	0.00
7	0.00000	0.023256	0.023256	0.00

8	0.00000	0.010000	0.000000	0.02
9	0.00000	0.020000	0.000000	0.00

	Spiritual Center	Sporting Goods Shop	Sports Bar	Sports Club	Stables \
0	0.0	0.00	0.00	0.00	0.0
1	0.0	0.00	0.00	0.00	0.0
2	0.0	0.00	0.00	0.00	0.0
3	0.0	0.01	0.00	0.00	0.0
4	0.0	0.00	0.00	0.01	0.0
5	0.0	0.03	0.00	0.00	0.0
6	0.0	0.01	0.01	0.00	0.0
7	0.0	0.00	0.00	0.00	0.0
8	0.0	0.00	0.00	0.00	0.0
9	0.0	0.00	0.00	0.00	0.0

	Steakhouse	Street Art	Strip Club	Supermarket	Supplement Shop \
0	0.010101	0.0	0.00	0.00	0.0
1	0.000000	0.0	0.00	0.01	0.0
2	0.000000	0.0	0.00	0.00	0.0
3	0.000000	0.0	0.00	0.00	0.0
4	0.000000	0.0	0.00	0.02	0.0
5	0.000000	0.0	0.01	0.00	0.0
6	0.010000	0.0	0.00	0.01	0.0
7	0.023256	0.0	0.00	0.00	0.0
8	0.000000	0.0	0.00	0.00	0.0
9	0.040000	0.0	0.00	0.00	0.0

	Sushi Restaurant	Swiss Restaurant	Szechuan Restaurant	Taco Place \
0	0.020202	0.00	0.0	0.000000
1	0.010000	0.00	0.0	0.010000
2	0.000000	0.00	0.0	0.000000
3	0.010000	0.00	0.0	0.010000
4	0.000000	0.00	0.0	0.000000
5	0.020000	0.00	0.0	0.000000
6	0.000000	0.00	0.0	0.010000
7	0.000000	0.00	0.0	0.023256
8	0.000000	0.01	0.0	0.000000
9	0.000000	0.00	0.0	0.000000

	Tailor Shop	Taiwanese Restaurant	Tapas Restaurant	Tattoo Parlor \
0	0.00	0.00	0.00	0.0
1	0.00	0.00	0.00	0.0
2	0.00	0.00	0.00	0.0
3	0.00	0.00	0.02	0.0
4	0.00	0.00	0.00	0.0
5	0.00	0.00	0.00	0.0
6	0.00	0.00	0.00	0.0
7	0.00	0.00	0.00	0.0

8	0.01	0.01	0.01	0.0
9	0.00	0.00	0.00	0.0

	Tea Room	Tech Startup	Tennis Court	Tennis Stadium	Thai Restaurant \
0	0.00	0.0	0.0	0.0	0.000000
1	0.00	0.0	0.0	0.0	0.000000
2	0.00	0.0	0.0	0.0	0.000000
3	0.00	0.0	0.0	0.0	0.000000
4	0.01	0.0	0.0	0.0	0.010000
5	0.00	0.0	0.0	0.0	0.000000
6	0.00	0.0	0.0	0.0	0.010000
7	0.00	0.0	0.0	0.0	0.046512
8	0.00	0.0	0.0	0.0	0.000000
9	0.00	0.0	0.0	0.0	0.000000

	Theater	Theme Park Ride / Attraction	Thrift / Vintage Store	Tiki Bar \
0	0.00	0.0	0.00	0.0
1	0.00	0.0	0.00	0.0
2	0.00	0.0	0.00	0.0
3	0.03	0.0	0.00	0.0
4	0.00	0.0	0.00	0.0
5	0.01	0.0	0.00	0.0
6	0.12	0.0	0.00	0.0
7	0.00	0.0	0.00	0.0
8	0.00	0.0	0.02	0.0
9	0.00	0.0	0.00	0.0

	Tourist Information Center	Toy / Game Store	Trail	Tree \
0	0.0	0.00	0.00	0.010101
1	0.0	0.00	0.00	0.000000
2	0.0	0.00	0.00	0.000000
3	0.0	0.00	0.00	0.000000
4	0.0	0.00	0.00	0.000000
5	0.0	0.01	0.00	0.000000
6	0.0	0.00	0.00	0.000000
7	0.0	0.00	0.00	0.000000
8	0.0	0.00	0.00	0.000000
9	0.0	0.00	0.01	0.000000

	Turkish Restaurant	Udon Restaurant	Used Bookstore \
0	0.0	0.0	0.00
1	0.0	0.0	0.00
2	0.0	0.0	0.00
3	0.0	0.0	0.00
4	0.0	0.0	0.00
5	0.0	0.0	0.00
6	0.0	0.0	0.00
7	0.0	0.0	0.00

8	0.0	0.0	0.01
9	0.0	0.0	0.00

	Vegetarian / Vegan Restaurant	Venezuelan Restaurant	Veterinarian \
0	0.00	0.0	0.0
1	0.01	0.0	0.0
2	0.00	0.0	0.0
3	0.01	0.0	0.0
4	0.01	0.0	0.0
5	0.00	0.0	0.0
6	0.01	0.0	0.0
7	0.00	0.0	0.0
8	0.03	0.0	0.0
9	0.00	0.0	0.0

	Video Game Store	Video Store	Vietnamese Restaurant	Volleyball Court \
0	0.0	0.0	0.00	0.0
1	0.0	0.0	0.02	0.0
2	0.0	0.0	0.00	0.0
3	0.0	0.0	0.00	0.0
4	0.0	0.0	0.04	0.0
5	0.0	0.0	0.00	0.0
6	0.0	0.0	0.00	0.0
7	0.0	0.0	0.00	0.0
8	0.0	0.0	0.02	0.0
9	0.0	0.0	0.00	0.0

	Watch Shop	Waterfront	Weight Loss Center	Whisky Bar	Wine Bar \
0	0.0	0.0	0.0	0.0	0.010101
1	0.0	0.0	0.0	0.0	0.010000
2	0.0	0.0	0.0	0.0	0.000000
3	0.0	0.0	0.0	0.0	0.000000
4	0.0	0.0	0.0	0.0	0.000000
5	0.0	0.0	0.0	0.0	0.010000
6	0.0	0.0	0.0	0.0	0.020000
7	0.0	0.0	0.0	0.0	0.000000
8	0.0	0.0	0.0	0.0	0.050000
9	0.0	0.0	0.0	0.0	0.010000

	Wine Shop	Wings Joint	Women's Store	Yoga Studio
0	0.030303	0.00	0.010101	0.00
1	0.030000	0.00	0.010000	0.03
2	0.000000	0.00	0.000000	0.00
3	0.020000	0.00	0.010000	0.00
4	0.000000	0.00	0.000000	0.00
5	0.020000	0.01	0.000000	0.03
6	0.030000	0.00	0.000000	0.00
7	0.000000	0.00	0.000000	0.00

8	0.020000	0.00	0.000000	0.00
9	0.030000	0.00	0.010000	0.00

Let's confirm the new size

```
In [86]: ny_grouped.shape
```

```
Out[86]: (40, 343)
```

Let's print each neighborhood along with the top 5 most common venues

```
In [87]: num_top_venues = 5
```

```
for hood in ny_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = ny_grouped[ny_grouped['Neighborhood'] == hood].T.reset_index()
    temp.columns = ['venue', 'freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

```
----Battery Park City----
```

	venue	freq
0	Park	0.07
1	Coffee Shop	0.06
2	Hotel	0.05
3	Memorial Site	0.04
4	Gym	0.04

```
----Carnegie Hill----
```

	venue	freq
0	Coffee Shop	0.07
1	Pizza Place	0.06
2	Bar	0.04
3	Café	0.04
4	Bakery	0.03

```
----Central Harlem----
```

	venue	freq
0	African Restaurant	0.07
1	American Restaurant	0.05
2	Bar	0.05
3	Public Art	0.05
4	Seafood Restaurant	0.05

----Chelsea----

	venue	freq
0	Coffee Shop	0.06
1	Italian Restaurant	0.05
2	Ice Cream Shop	0.05
3	Nightclub	0.04
4	Bakery	0.04

----Chinatown----

	venue	freq
0	Chinese Restaurant	0.09
1	Cocktail Bar	0.05
2	Vietnamese Restaurant	0.04
3	American Restaurant	0.04
4	Salon / Barbershop	0.04

----Civic Center----

	venue	freq
0	Gym / Fitness Center	0.05
1	Italian Restaurant	0.04
2	Sandwich Place	0.04
3	Hotel	0.04
4	French Restaurant	0.04

----Clinton----

	venue	freq
0	Theater	0.12
1	Gym / Fitness Center	0.05
2	Italian Restaurant	0.04
3	American Restaurant	0.04
4	Hotel	0.04

----East Harlem----

	venue	freq
0	Mexican Restaurant	0.12
1	Bakery	0.09
2	Deli / Bodega	0.07
3	Latin American Restaurant	0.05
4	Thai Restaurant	0.05

----East Village----

	venue	freq
--	-------	------



0	Bar	0.06
1	Wine Bar	0.05
2	Chinese Restaurant	0.04
3	Pizza Place	0.04
4	Ice Cream Shop	0.04

----Financial District----

	venue	freq
0	Coffee Shop	0.08
1	Steakhouse	0.04
2	Gym	0.04
3	Hotel	0.04
4	Pizza Place	0.03

----Flatiron----

	venue	freq
0	Gym	0.05
1	Yoga Studio	0.04
2	American Restaurant	0.04
3	Japanese Restaurant	0.04
4	Gym / Fitness Center	0.04

----Gramercy----

	venue	freq
0	Bar	0.07
1	Italian Restaurant	0.04
2	American Restaurant	0.04
3	Pizza Place	0.04
4	Bagel Shop	0.04

----Greenwich Village----

	venue	freq
0	Italian Restaurant	0.13
1	Clothing Store	0.04
2	Sushi Restaurant	0.04
3	Café	0.03
4	Chinese Restaurant	0.03

----Hamilton Heights----

	venue	freq
0	Café	0.07
1	Mexican Restaurant	0.07
2	Coffee Shop	0.07

3	Pizza Place	0.07
4	Caribbean Restaurant	0.03

----Hudson Yards----

	venue	freq
0	American Restaurant	0.07
1	Italian Restaurant	0.05
2	Café	0.05
3	Gym / Fitness Center	0.05
4	Hotel	0.04

----Inwood----

	venue	freq
0	Lounge	0.07
1	Mexican Restaurant	0.07
2	Café	0.07
3	Deli / Bodega	0.05
4	Pizza Place	0.05

----Lenox Hill----

	venue	freq
0	Coffee Shop	0.07
1	Italian Restaurant	0.05
2	Pizza Place	0.05
3	Sushi Restaurant	0.05
4	Burger Joint	0.03

----Lincoln Square----

	venue	freq
0	Gym / Fitness Center	0.06
1	Theater	0.06
2	Concert Hall	0.05
3	Plaza	0.05
4	Café	0.05

----Little Italy----

	venue	freq
0	Bakery	0.05
1	Café	0.04
2	Sandwich Place	0.03
3	Clothing Store	0.03
4	Bubble Tea Shop	0.03

----Lower East Side----

	venue	freq
0	Coffee Shop	0.07
1	Café	0.05
2	Chinese Restaurant	0.05
3	Art Gallery	0.05
4	Ramen Restaurant	0.05

----Manhattan Valley----

	venue	freq
0	Pizza Place	0.05
1	Coffee Shop	0.05
2	Indian Restaurant	0.05
3	Yoga Studio	0.04
4	Café	0.04

----Manhattanville----

	venue	freq
0	Italian Restaurant	0.05
1	Deli / Bodega	0.05
2	Seafood Restaurant	0.05
3	Coffee Shop	0.05
4	Park	0.05

----Marble Hill----

	venue	freq
0	Sandwich Place	0.12
1	Coffee Shop	0.08
2	Discount Store	0.08
3	Yoga Studio	0.04
4	Supplement Shop	0.04

----Midtown----

	venue	freq
0	Hotel	0.07
1	Clothing Store	0.04
2	Cocktail Bar	0.04
3	Coffee Shop	0.04
4	Theater	0.04

----Midtown South----

	venue	freq
--	-------	------

0	Korean Restaurant	0.14
1	Hotel	0.07
2	Japanese Restaurant	0.04
3	Hotel Bar	0.04
4	Cosmetics Shop	0.04

----Morningside Heights----

	venue	freq
0	Bookstore	0.08
1	American Restaurant	0.08
2	Coffee Shop	0.08
3	Park	0.08
4	Sandwich Place	0.05

----Murray Hill----

	venue	freq
0	Coffee Shop	0.05
1	Japanese Restaurant	0.04
2	Sandwich Place	0.04
3	Hotel	0.04
4	Italian Restaurant	0.03

----Noho----

	venue	freq
0	Italian Restaurant	0.06
1	French Restaurant	0.05
2	Cocktail Bar	0.04
3	Sushi Restaurant	0.04
4	Art Gallery	0.03

----Roosevelt Island----

	venue	freq
0	Park	0.07
1	Coffee Shop	0.07
2	Sandwich Place	0.07
3	Playground	0.03
4	Gym / Fitness Center	0.03

----Soho----

	venue	freq
0	Clothing Store	0.09
1	Boutique	0.06
2	Women's Store	0.04

3	Shoe Store	0.04
4	Art Gallery	0.04

----Stuyvesant Town----

	venue	freq
0	Park	0.12
1	Bar	0.12
2	Playground	0.12
3	Baseball Field	0.06
4	Farmers Market	0.06

----Sutton Place----

	venue	freq
0	Gym / Fitness Center	0.06
1	Italian Restaurant	0.04
2	Indian Restaurant	0.04
3	Furniture / Home Store	0.04
4	Pizza Place	0.03

----Tribeca----

	venue	freq
0	Italian Restaurant	0.05
1	Café	0.05
2	Boutique	0.05
3	Park	0.05
4	American Restaurant	0.04

----Tudor City----

	venue	freq
0	Mexican Restaurant	0.06
1	Park	0.06
2	Pizza Place	0.05
3	Greek Restaurant	0.05
4	Café	0.05

----Turtle Bay----

	venue	freq
0	Italian Restaurant	0.06
1	Coffee Shop	0.05
2	Sushi Restaurant	0.05
3	Steakhouse	0.05
4	Ramen Restaurant	0.04

----Upper East Side----

	venue	freq
0	Italian Restaurant	0.08
1	Exhibit	0.07
2	Art Gallery	0.05
3	Bakery	0.04
4	Coffee Shop	0.04

----Upper West Side----

	venue	freq
0	Italian Restaurant	0.06
1	Wine Bar	0.04
2	Bar	0.04
3	Cosmetics Shop	0.03
4	Indian Restaurant	0.03

----Washington Heights----

	venue	freq
0	Café	0.06
1	Mobile Phone Shop	0.05
2	Bakery	0.05
3	Grocery Store	0.03
4	Wine Shop	0.02

----West Village----

	venue	freq
0	Italian Restaurant	0.09
1	Cosmetics Shop	0.05
2	New American Restaurant	0.05
3	Park	0.04
4	Wine Bar	0.04

----Yorkville----

	venue	freq
0	Italian Restaurant	0.06
1	Gym	0.06
2	Coffee Shop	0.06
3	Bar	0.05
4	Sushi Restaurant	0.04

Let's put that into a pandas dataframe

```
In [88]: #a function to sort the venues in descending order
```

```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

```
In [89]: #create the new dataframe and display the top 10 venues for each neighborhood.
```

```
num_top_venues = 10
word_string3 = ''
indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Neighborhood'] = ny_grouped['Neighborhood']

for ind in np.arange(ny_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(ny_grouped.iloc[ind, 1:], num_top_venues)
    word_string3 = word_string3 + neighborhoods_venues_sorted.iloc[ind, 1:] + ' '

neighborhoods_venues_sorted.head(10)
```

```
Out [89]:
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	\
0	Battery Park City	Park	Coffee Shop	
1	Carnegie Hill	Coffee Shop	Pizza Place	
2	Central Harlem	African Restaurant	Seafood Restaurant	
3	Chelsea	Coffee Shop	Ice Cream Shop	
4	Chinatown	Chinese Restaurant	Cocktail Bar	
5	Civic Center	Gym / Fitness Center	French Restaurant	
6	Clinton	Theater	Gym / Fitness Center	
7	East Harlem	Mexican Restaurant	Bakery	
8	East Village	Bar	Wine Bar	
9	Financial District	Coffee Shop	Steakhouse	

  

	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	\
0	Hotel	Memorial Site	Gym	
1	Café	Bar	Yoga Studio	

2	American Restaurant	French Restaurant	Public Art
3	Italian Restaurant	Nightclub	Bakery
4	Salon / Barbershop	American Restaurant	Vietnamese Restaurant
5	Sandwich Place	Italian Restaurant	Hotel
6	American Restaurant	Italian Restaurant	Hotel
7	Deli / Bodega	Thai Restaurant	Latin American Restaurant
8	Chinese Restaurant	Mexican Restaurant	Ice Cream Shop
9	Hotel	Gym	Wine Shop

	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue \
0	Boat or Ferry	Wine Shop	Italian Restaurant
1	Bookstore	Japanese Restaurant	Grocery Store
2	Bar	Chinese Restaurant	Dessert Shop
3	Theater	Hotel	American Restaurant
4	Ice Cream Shop	Bubble Tea Shop	Spa
5	Coffee Shop	Yoga Studio	Sporting Goods Shop
6	Coffee Shop	Wine Shop	Spa
7	Chinese Restaurant	Liquor Store	Taco Place
8	Pizza Place	Ramen Restaurant	Coffee Shop
9	Bar	American Restaurant	Event Space

	9th Most Common Venue	10th Most Common Venue
0	Sushi Restaurant	Burger Joint
1	Bakery	Gym
2	Beer Bar	Library
3	Seafood Restaurant	Art Gallery
4	Dumpling Restaurant	Bakery
5	American Restaurant	Spa
6	Sandwich Place	Mediterranean Restaurant
7	Gas Station	Steakhouse
8	Cocktail Bar	Vegetarian / Vegan Restaurant
9	Pizza Place	Cocktail Bar

### 1.4.3 Vizualization with word cloud

```
In [90]: stopwords = set(STOPWORDS)
```

```
In [91]: word_string4 = ''
         for i in range(0, num_top_venues ):
             word_string4 = word_string4 + word_string3[i]
```

```
In [92]: # create the word cloud
         wordcloud = WordCloud(background_color='white').generate(word_string4)

         print('Word cloud created!')

         # display the cloud
         fig = plt.figure()
```



Word cloud created!



```
In [93]: # set number of clusters
kclusters = 5

ny_grouped_clustering = ny_grouped.drop('Neighborhood', 1)

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

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

```
Out[93]: array([2, 1, 1, 2, 1, 2, 2, 0, 1, 1], dtype=int32)
```

65

```
In [94]: # add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

ny_merged = ny_data

# merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
ny_merged = ny_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on=

ny_merged.head() # check the last columns!
```

```
Out[94]:
```

	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	\
0	Manhattan	Marble Hill	40.876551	-73.910660	4	
1	Manhattan	Chinatown	40.715618	-73.994279	1	
2	Manhattan	Washington Heights	40.851903	-73.936900	0	
3	Manhattan	Inwood	40.867684	-73.921210	0	
4	Manhattan	Hamilton Heights	40.823604	-73.949688	0	

  

	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	\
0	Sandwich Place	Coffee Shop	Discount Store	
1	Chinese Restaurant	Cocktail Bar	Salon / Barbershop	
2	Café	Bakery	Mobile Phone Shop	
3	Lounge	Café	Mexican Restaurant	
4	Pizza Place	Coffee Shop	Café	

  

	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	\
0	Yoga Studio	Supplement Shop	Steakhouse	
1	American Restaurant	Vietnamese Restaurant	Ice Cream Shop	
2	Grocery Store	Mexican Restaurant	Supermarket	
3	Deli / Bodega	Pizza Place	Chinese Restaurant	
4	Mexican Restaurant	Yoga Studio	Bakery	

  

	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	\
0	Spa	Shopping Mall	Seafood Restaurant	
1	Bubble Tea Shop	Spa	Dumpling Restaurant	
2	Sandwich Place	Chinese Restaurant	Tapas Restaurant	
3	Wine Bar	American Restaurant	Bakery	
4	Park	Cocktail Bar	Sandwich Place	

  

	10th Most Common Venue
0	Clothing Store
1	Bakery
2	Coffee Shop
3	Park
4	School

Finally, let's visualize the resulting clusters

```
In [95]: # create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)
```

```

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

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(ny_merged['Latitude'], ny_merged['Longitude'], ny_m
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters

```

Out[95]: <folium.folium.Map at 0x7fe50d6d2cf8>

## 1.4.5 Examining the clusters

Let's try and see each cluster and the most common venue among each.

```

In [96]: # For Cluster 0
result = ny_merged.loc[ny_merged['Cluster Labels'] == 0, ny_merged.columns[[1] + list
print("For cluster {}, the distribution of venues is as:\n{}".format(0, result['1st M
result

```

For cluster 0, the distribution of venues is as:

Mexican Restaurant	2
Pizza Place	2
Café	1
Coffee Shop	1
Lounge	1
Park	1

Name: 1st Most Common Venue, dtype: int64

```

Out[96]:
      Neighborhood 1st Most Common Venue 2nd Most Common Venue \
2  Washington Heights      Café      Bakery
3              Inwood      Lounge      Café
4  Hamilton Heights      Pizza Place      Coffee Shop
5  Manhattanville      Mexican Restaurant      Park
7      East Harlem      Mexican Restaurant      Bakery

```

20	Lower East Side	Coffee Shop	Ramen Restaurant
25	Manhattan Valley	Pizza Place	Coffee Shop
36	Tudor City	Park	Mexican Restaurant
	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue \
2	Mobile Phone Shop	Grocery Store	Mexican Restaurant
3	Mexican Restaurant	Deli / Bodega	Pizza Place
4	Café	Mexican Restaurant	Yoga Studio
5	Coffee Shop	Deli / Bodega	Italian Restaurant
7	Deli / Bodega	Thai Restaurant	Latin American Restaurant
20	Chinese Restaurant	Café	Pizza Place
25	Indian Restaurant	Yoga Studio	Thai Restaurant
36	Café	Greek Restaurant	Pizza Place
	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue \
2	Supermarket	Sandwich Place	Chinese Restaurant
3	Chinese Restaurant	Wine Bar	American Restaurant
4	Bakery	Park	Cocktail Bar
5	Seafood Restaurant	Diner	Bike Trail
7	Chinese Restaurant	Liquor Store	Taco Place
20	Art Gallery	Cocktail Bar	Sandwich Place
25	Mexican Restaurant	Café	Bar
36	Deli / Bodega	Hotel	Spa
	9th Most Common Venue	10th Most Common Venue	
2	Tapas Restaurant	Coffee Shop	
3	Bakery	Park	
4	Sandwich Place	School	
5	Lounge	Sushi Restaurant	
7	Gas Station	Steakhouse	
20	Bakery	Japanese Restaurant	
25	Deli / Bodega	Playground	
36	Restaurant	Diner	

```
In [97]: # For Cluster 1
```

```
result = ny_merged.loc[ny_merged['Cluster Labels'] == 1, ny_merged.columns[[1] + list
print("For cluster {}, the distribution of venues is as:\n{}".format(1, result['1st M
result
```

For cluster 1, the distribution of venues is as:

Coffee Shop	4
Bar	2
Italian Restaurant	2
Gym	1
Chinese Restaurant	1
Gym / Fitness Center	1
Sandwich Place	1
Korean Restaurant	1

African Restaurant 1  
 Name: 1st Most Common Venue, dtype: int64

```

Out[97]:
      Neighborhood 1st Most Common Venue 2nd Most Common Venue \
1      Chinatown    Chinese Restaurant    Cocktail Bar
6    Central Harlem    African Restaurant    Seafood Restaurant
9      Yorkville                Gym    Italian Restaurant
10     Lenox Hill        Coffee Shop    Pizza Place
11  Roosevelt Island    Sandwich Place    Park
12   Upper West Side    Italian Restaurant    Wine Bar
16     Murray Hill        Coffee Shop    Sandwich Place
19     East Village                Bar    Wine Bar
27      Gramercy                Bar    Italian Restaurant
29  Financial District    Coffee Shop    Steakhouse
30     Carnegie Hill    Coffee Shop    Pizza Place
33   Midtown South    Korean Restaurant    Hotel
34     Sutton Place  Gym / Fitness Center    Italian Restaurant
35     Turtle Bay    Italian Restaurant    Steakhouse

      3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue \
1    Salon / Barbershop    American Restaurant    Vietnamese Restaurant
6    American Restaurant    French Restaurant    Public Art
9      Coffee Shop                Bar    Sushi Restaurant
10   Italian Restaurant    Sushi Restaurant    Sporting Goods Shop
11      Coffee Shop                Food Truck    Noodle House
12                Bar    Indian Restaurant    Mediterranean Restaurant
16                Hotel    Japanese Restaurant    Bar
19   Chinese Restaurant    Mexican Restaurant    Ice Cream Shop
27      Bagel Shop    American Restaurant    Pizza Place
29                Hotel                Gym    Wine Shop
30                Café                Bar    Yoga Studio
33      Dessert Shop    Cosmetics Shop    Hotel Bar
34   Indian Restaurant    Furniture / Home Store    American Restaurant
35      Coffee Shop    Sushi Restaurant    Ramen Restaurant

      6th Most Common Venue 7th Most Common Venue 8th Most Common Venue \
1      Ice Cream Shop    Bubble Tea Shop    Spa
6                Bar    Chinese Restaurant    Dessert Shop
9      Pizza Place                Wine Shop    Deli / Bodega
10                Gym  Gym / Fitness Center    Burger Joint
11   Greek Restaurant    Liquor Store    Metro Station
12                Bakery    Cosmetics Shop    Coffee Shop
16   Italian Restaurant    French Restaurant  Gym / Fitness Center
19      Pizza Place    Ramen Restaurant    Coffee Shop
27   Thai Restaurant    Coffee Shop    Mexican Restaurant
29                Bar    American Restaurant    Event Space
30      Bookstore    Japanese Restaurant    Grocery Store

```

33	Japanese Restaurant	Coffee Shop	American Restaurant
34	Gym	Juice Bar	Dessert Shop
35	Wine Bar	Indian Restaurant	Japanese Restaurant
	9th Most Common Venue	10th Most Common Venue	
1	Dumpling Restaurant	Bakery	
6	Beer Bar	Library	
9	Mexican Restaurant	Japanese Restaurant	
10	Cosmetics Shop	Deli / Bodega	
11	Supermarket	Bubble Tea Shop	
12	Vegetarian / Vegan Restaurant	Pet Store	
16	Gym	Salon / Barbershop	
19	Cocktail Bar	Vegetarian / Vegan Restaurant	
27	Cocktail Bar	Thrift / Vintage Store	
29	Pizza Place	Cocktail Bar	
30	Bakery	Gym	
33	Boutique	Bakery	
34	Pizza Place	French Restaurant	
35	French Restaurant	Café	

In [98]: # For Cluster 2

```
result = ny_merged.loc[ny_merged['Cluster Labels'] == 2, ny_merged.columns[[1] + list
print("For cluster {}, the distribution of venues is as:\n{}".format(2, result['1st M
result
```

For cluster 2, the distribution of venues is as:

Italian Restaurant	4
Theater	2
American Restaurant	1
Bakery	1
Bookstore	1
Gym	1
Park	1
Hotel	1
Gym / Fitness Center	1
Coffee Shop	1
Boutique	1
Clothing Store	1

Name: 1st Most Common Venue, dtype: int64

Out[98]:	Neighborhood	1st Most Common Venue	2nd Most Common Venue	\
8	Upper East Side	Italian Restaurant	Exhibit	
13	Lincoln Square	Theater	Gym / Fitness Center	
14	Clinton	Theater	Gym / Fitness Center	
15	Midtown	Hotel	Cocktail Bar	
17	Chelsea	Coffee Shop	Ice Cream Shop	
18	Greenwich Village	Italian Restaurant	Clothing Store	

21	Tribeca	Boutique	Park
22	Little Italy	Bakery	Café
23	Soho	Clothing Store	Boutique
24	West Village	Italian Restaurant	Cosmetics Shop
26	Morningside Heights	Bookstore	American Restaurant
28	Battery Park City	Park	Coffee Shop
31	Noho	Italian Restaurant	French Restaurant
32	Civic Center	Gym / Fitness Center	French Restaurant
38	Flatiron	Gym	Gym / Fitness Center
39	Hudson Yards	American Restaurant	Gym / Fitness Center

	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue \
8	Art Gallery	Coffee Shop	Gym / Fitness Center
13	Café	Concert Hall	Plaza
14	American Restaurant	Italian Restaurant	Hotel
15	Clothing Store	Theater	Coffee Shop
17	Italian Restaurant	Nightclub	Bakery
18	Sushi Restaurant	Indian Restaurant	Cosmetics Shop
21	Italian Restaurant	Café	Spa
22	Salon / Barbershop	Italian Restaurant	Sandwich Place
23	Art Gallery	Women's Store	Shoe Store
24	New American Restaurant	Park	Wine Bar
26	Park	Coffee Shop	Food Truck
28	Hotel	Memorial Site	Gym
31	Cocktail Bar	Sushi Restaurant	Pizza Place
32	Sandwich Place	Italian Restaurant	Hotel
38	Japanese Restaurant	Yoga Studio	American Restaurant
39	Café	Italian Restaurant	Hotel

	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue \
8	Bakery	French Restaurant	Spa
13	Italian Restaurant	Performing Arts Venue	Indie Movie Theater
14	Coffee Shop	Wine Shop	Spa
15	Bakery	Spa	Japanese Restaurant
17	Theater	Hotel	American Restaurant
18	Chinese Restaurant	Seafood Restaurant	Café
21	American Restaurant	Wine Shop	Wine Bar
22	Bubble Tea Shop	Clothing Store	Mediterranean Restaurant
23	Men's Store	Italian Restaurant	Hotel
24	American Restaurant	Cocktail Bar	Bakery
26	Tennis Court	Deli / Bodega	Burger Joint
28	Boat or Ferry	Wine Shop	Italian Restaurant
31	Boutique	Coffee Shop	Rock Club
32	Coffee Shop	Yoga Studio	Sporting Goods Shop
38	Café	Salon / Barbershop	New American Restaurant
39	Spanish Restaurant	Restaurant	Theater

9th Most Common Venue      10th Most Common Venue

8	Hotel	Juice Bar
13	Opera House	French Restaurant
14	Sandwich Place	Mediterranean Restaurant
15	Sporting Goods Shop	Steakhouse
17	Seafood Restaurant	Art Gallery
18	French Restaurant	Burger Joint
21	Gym	Coffee Shop
22	Cocktail Bar	Seafood Restaurant
23	Bakery	Sporting Goods Shop
24	Jazz Club	Speakeasy
26	Sandwich Place	New American Restaurant
28	Sushi Restaurant	Burger Joint
31	Mexican Restaurant	Bookstore
32	American Restaurant	Spa
38	Clothing Store	Cosmetics Shop
39	Coffee Shop	Thai Restaurant

In [99]: # For Cluster 3

```
result = ny_merged.loc[ny_merged['Cluster Labels'] == 3, ny_merged.columns[[1] + list
print("For cluster {}, the distribution of venues is as:\n{}".format(3, result['1st M
result
```

For cluster 3, the distribution of venues is as:

Bar 1

Name: 1st Most Common Venue, dtype: int64

```
Out[99]:      Neighborhood 1st Most Common Venue 2nd Most Common Venue \
37 Stuyvesant Town      Bar      Park

      3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue \
37      Playground      Pet Service      Gas Station

      6th Most Common Venue 7th Most Common Venue 8th Most Common Venue \
37      Boat or Ferry      Farmers Market      Basketball Court

      9th Most Common Venue 10th Most Common Venue
37      Baseball Field      Harbor / Marina
```

In [133]: #For Cluster 4

```
#result = ny_merged.loc[ny_merged['Cluster Labels'] == 4, ny_merged.columns[[1] + li
#print("For cluster {}, the distribution of venues is as:\n{}".format(4, result['1st
#result
```

## 1.4.6 Let's explore the Kensington and Chelsea, uk in our dataframe.

Get the neighborhood's name.

In [100]: address = 'Kensington and Chelsea, uk'



```

geolocator = Nominatim(user_agent="uk_explorer")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Kensington and Chelsea, uk {}, {}'.format(latitude, longitude))

```

The geograpical coordinate of Kensington and Chelsea, uk 51.4989948, -0.1991229.

```

In [101]: LIMIT = 200 # limit of number of venues returned by Foursquare API
radius = 1500 # define radius
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&format=json'
      CLIENT_ID,
      CLIENT_SECRET,
      VERSION,
      latitude,
      longitude,
      radius,
      LIMIT)
url # display URL

```

```

Out[101]: 'https://api.foursquare.com/v2/venues/explore?&client_id=B3D1FREXU3FMFKG0XFFFWLZH1UBI&client_secret=...&v=20131013&format=json'

```

```

In [169]: results1 = requests.get(url).json()
#results1

```

```

In [103]: # function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']

```

```

In [104]: venues = results1['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
nearby_venues = nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)

```

```
# clean columns
nearby_venues.columns = [col.split(".")[0] for col in nearby_venues.columns]

nearby_venues.head(10)
```

```
Out[104]:
```

	name	categories	lat \
0	Core Collective	Gym / Fitness Center	51.499589
1	The Design Museum	Museum	51.499785
2	Café Phillies	Café	51.499726
3	Leighton House Museum	History Museum	51.498591
4	The Scarsdale Tavern	Pub	51.496975
5	Stanley Kubrick: The Exhibition	Exhibit	51.499809
6	Cafe Tarte	Café	51.498552
7	Gails Bakery	Bakery	51.498864
8	The Abingdon	Modern European Restaurant	51.497633
9	Byron	Burger Joint	51.499344

```
lng
0 -0.198630
1 -0.199641
2 -0.197747
3 -0.203118
4 -0.199024
5 -0.199966
6 -0.200713
7 -0.199683
8 -0.196110
9 -0.198445
```

```
In [105]: print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))

100 venues were returned by Foursquare.
```

```
In [106]: def getNearbyVenues(names, latitudes, longitudes, radius=1500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&lat={}&lng={}&radius={}&limit={}'
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        lat,
        lng,
        radius,
        LIMIT)
```

```

# make the GET request
results = requests.get(url).json()["response"]["groups"][0]["items"]

# return only relevant information for each nearby venue
venues_list.append([
    name,
    lat,
    lng,
    v['venue']['name'],
    v['venue']['location']['lat'],
    v['venue']['location']['lng'],
    v['venue']['categories'][0]['name']) for v in results])

nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
nearby_venues.columns = ['Neighborhood',
                        'Neighborhood Latitude',
                        'Neighborhood Longitude',
                        'Venue',
                        'Venue Latitude',
                        'Venue Longitude',
                        'Venue Category']

return(nearby_venues)

```

```

In [107]: uk_venues = getNearbyVenues(names=new_data['neighborhoods'],
                                     latitudes=new_data['Latitude'],
                                     longitudes=new_data['Longitude'])

```

```

Brompton
Chelsea
Earls Court
Kensington
South Kensington
West Brompton
Holland Park
North Kensington
Notting Hill

```

```

In [108]: print(uk_venues.shape)

```

```

(900, 7)

```

```

In [109]: uk_venues.head()

```

```

Out[109]:  Neighborhood  Neighborhood Latitude  Neighborhood Longitude \
0      Brompton          51.49014          -0.16248

```

1	Brompton	51.49014	-0.16248
2	Brompton	51.49014	-0.16248
3	Brompton	51.49014	-0.16248
4	Brompton	51.49014	-0.16248

	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Venchi	51.489239	-0.164265	Ice Cream Shop
1	Saturday Farmers' Market	51.490917	-0.160329	Farmers Market
2	Duke of York Square	51.491272	-0.159827	Plaza
3	The Five Fields	51.491770	-0.161191	Restaurant
4	Amorino	51.489455	-0.163803	Ice Cream Shop

```
In [110]: uk_venues.groupby('Neighborhood').count()
```

```
Out[110]:
```

	Neighborhood	Latitude	Longitude	Venue
Neighborhood				
Brompton		100	100	100
Chelsea		100	100	100
Earls Court		100	100	100
Holland Park		100	100	100
Kensington		100	100	100
North Kensington		100	100	100
Notting Hill		100	100	100
South Kensington		100	100	100
West Brompton		100	100	100

  

	Venue Latitude	Venue Longitude	Venue Category
Neighborhood			
Brompton	100	100	100
Chelsea	100	100	100
Earls Court	100	100	100
Holland Park	100	100	100
Kensington	100	100	100
North Kensington	100	100	100
Notting Hill	100	100	100
South Kensington	100	100	100
West Brompton	100	100	100

```
In [111]: print('There are {} unique categories.'.format(len(uk_venues['Venue Category'].unique())))
```

There are 135 unique categories.

```
In [112]: # one hot encoding
uk_onehot = pd.get_dummies(uk_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
uk_onehot['Neighborhood'] = uk_venues['Neighborhood']
```

```
# move neighborhood column to the first column
fixed_columns = [uk_onehot.columns[-1]] + list(uk_onehot.columns[:-1])
uk_onehot = uk_onehot[fixed_columns]
```

```
uk_onehot.head()
```

```
Out[112]:
```

	Neighborhood	American Restaurant	Antique Shop	Argentinian Restaurant	\
0	Brompton	0	0	0	
1	Brompton	0	0	0	
2	Brompton	0	0	0	
3	Brompton	0	0	0	
4	Brompton	0	0	0	

  

	Art Gallery	Art Museum	Asian Restaurant	Australian Restaurant	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

  

	Bagel Shop	Bakery	Bar	Bookstore	Boutique	Boxing Gym	Breakfast Spot	\
0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	

  

	Burger Joint	Butcher	Café	Candy Store	Cheese Shop	Chinese Restaurant	\
0	0	0	0	0	0	0	
1	0	0	0	0	0	0	
2	0	0	0	0	0	0	
3	0	0	0	0	0	0	
4	0	0	0	0	0	0	

  

	Chocolate Shop	Clothing Store	Cocktail Bar	Coffee Shop	College Quad	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

  

	Concert Hall	Cosmetics Shop	Creperie	Cupcake Shop	Dance Studio	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	Deli / Bodega	Department Store	Design Studio	Dessert Shop	Diner	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	Discount Store	Donut Shop	English Restaurant	Ethiopian Restaurant	\
0	0	0	0		0
1	0	0	0		0
2	0	0	0		0
3	0	0	0		0
4	0	0	0		0

	Exhibit	Farmers Market	Fast Food Restaurant	Filipino Restaurant	\
0	0	0	0		0
1	0	1	0		0
2	0	0	0		0
3	0	0	0		0
4	0	0	0		0

	Fish & Chips Shop	Fountain	French Restaurant	Furniture / Home Store	\
0	0	0	0		0
1	0	0	0		0
2	0	0	0		0
3	0	0	0		0
4	0	0	0		0

	Garden	Garden Center	Gastropub	Gift Shop	Gourmet Shop	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

	Greek Restaurant	Grilled Meat Restaurant	Grocery Store	Gym	\
0	0		0	0	0
1	0		0	0	0
2	0		0	0	0
3	0		0	0	0
4	0		0	0	0

	Gym / Fitness Center	Health & Beauty Service	Herbs & Spices Store	\
0	0	0		0
1	0	0		0
2	0	0		0
3	0	0		0
4	0	0		0

	Historic Site	History Museum	Hotel	Ice Cream Shop	Indian Restaurant	\
0	0	0	0	1	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	1	0	

	Indie Movie Theater	Indie Theater	Italian Restaurant	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	Japanese Restaurant	Jazz Club	Jewelry Store	Juice Bar	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Latin American Restaurant	Lebanese Restaurant	Lounge	Malay Restaurant	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Market	Massage Studio	Mediterranean Restaurant	Mexican Restaurant	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	Middle Eastern Restaurant	Modern European Restaurant	Monument / Landmark	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	Moroccan Restaurant	Movie Theater	Museum	Music Venue	Opera House	\
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	

4	0	0	0	0	0
---	---	---	---	---	---

  

	Palace	Park	Pastry Shop	Perfume Shop	Persian Restaurant	Pharmacy \
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0

  

	Piano Bar	Pilates Studio	Pizza Place	Playground	Plaza \
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	1
3	0	0	0	0	0
4	0	0	0	0	0

  

	Polish Restaurant	Portuguese Restaurant	Pub	Ramen Restaurant \
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

  

	Record Shop	Recording Studio	Restaurant	Road	Rock Climbing Spot \
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	1	0	0
4	0	0	0	0	0

  

	Russian Restaurant	Sandwich Place	Science Museum	Seafood Restaurant \
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

  

	Shopping Plaza	Skate Park	Soccer Stadium	Social Club	Speakeasy \
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

  

	Sporting Goods Shop	Sports Bar	Steakhouse	Street Food Gathering \
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0



3		0	0	0	0
4		0	0	0	0

  

	Supermarket	Sushi Restaurant	Tapas Restaurant	Tea Room	Thai Restaurant \
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

  

	Theater	Thrift / Vintage Store	Toy / Game Store	Turkish Restaurant \
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

  

	University	Vietnamese Restaurant	Wine Bar	Wine Shop	Women's Store \
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

  

	Yoga Studio
0	0
1	0
2	0
3	0
4	0

In [113]: uk\_onehot.shape

Out[113]: (900, 136)

In [114]: uk\_grouped = uk\_onehot.groupby('Neighborhood').mean().reset\_index()  
uk\_grouped

	Neighborhood	American Restaurant	Antique Shop \
0	Brompton	0.01	0.00
1	Chelsea	0.01	0.00
2	Earls Court	0.00	0.00
3	Holland Park	0.00	0.00
4	Kensington	0.00	0.00
5	North Kensington	0.00	0.00
6	Notting Hill	0.00	0.01
7	South Kensington	0.00	0.00
8	West Brompton	0.01	0.00

	Argentinian Restaurant	Art Gallery	Art Museum	Asian Restaurant	\
0	0.01	0.02	0.01	0.00	
1	0.01	0.02	0.01	0.00	
2	0.00	0.00	0.00	0.00	
3	0.00	0.01	0.00	0.00	
4	0.01	0.02	0.01	0.00	
5	0.00	0.00	0.00	0.01	
6	0.00	0.02	0.00	0.01	
7	0.01	0.02	0.01	0.00	
8	0.00	0.00	0.00	0.01	

	Australian Restaurant	Bagel Shop	Bakery	Bar	Bookstore	Boutique	\
0	0.00	0.00	0.04	0.00	0.02	0.03	
1	0.00	0.00	0.04	0.00	0.02	0.03	
2	0.00	0.01	0.03	0.00	0.01	0.00	
3	0.00	0.00	0.04	0.00	0.01	0.00	
4	0.00	0.00	0.02	0.00	0.01	0.01	
5	0.00	0.00	0.05	0.03	0.01	0.01	
6	0.01	0.00	0.04	0.00	0.02	0.00	
7	0.00	0.00	0.02	0.00	0.01	0.01	
8	0.00	0.01	0.05	0.00	0.02	0.00	

	Boxing Gym	Breakfast Spot	Burger Joint	Butcher	Café	Candy Store	\
0	0.01	0.01	0.01	0.00	0.07	0.00	
1	0.01	0.01	0.01	0.00	0.07	0.00	
2	0.00	0.00	0.02	0.00	0.03	0.00	
3	0.00	0.01	0.03	0.00	0.04	0.00	
4	0.00	0.00	0.03	0.00	0.07	0.00	
5	0.00	0.02	0.01	0.00	0.04	0.01	
6	0.01	0.03	0.02	0.01	0.03	0.00	
7	0.00	0.00	0.03	0.00	0.07	0.00	
8	0.00	0.00	0.02	0.00	0.06	0.00	

	Cheese Shop	Chinese Restaurant	Chocolate Shop	Clothing Store	\
0	0.01	0.00	0.01	0.01	
1	0.01	0.00	0.01	0.01	
2	0.01	0.01	0.00	0.00	
3	0.00	0.02	0.00	0.01	
4	0.01	0.00	0.01	0.00	
5	0.00	0.00	0.00	0.01	
6	0.01	0.00	0.00	0.02	
7	0.01	0.00	0.01	0.00	
8	0.01	0.00	0.00	0.00	

	Cocktail Bar	Coffee Shop	College Quad	Concert Hall	Cosmetics Shop	\
0	0.03	0.03	0.00	0.01	0.01	
1	0.03	0.03	0.00	0.01	0.01	
2	0.00	0.01	0.00	0.00	0.01	

3	0.00	0.00	0.00	0.01	0.00
4	0.02	0.02	0.01	0.01	0.00
5	0.04	0.02	0.00	0.00	0.00
6	0.03	0.02	0.00	0.00	0.00
7	0.02	0.02	0.01	0.01	0.00
8	0.00	0.02	0.00	0.00	0.01

	Creperie	Cupcake Shop	Dance Studio	Deli / Bodega	Department Store	\
0	0.01	0.01	0.00	0.01	0.02	
1	0.01	0.01	0.00	0.01	0.02	
2	0.00	0.02	0.00	0.00	0.00	
3	0.00	0.02	0.01	0.00	0.00	
4	0.00	0.01	0.00	0.00	0.01	
5	0.00	0.01	0.00	0.01	0.00	
6	0.00	0.01	0.00	0.01	0.00	
7	0.00	0.01	0.00	0.00	0.01	
8	0.01	0.02	0.00	0.00	0.00	

	Design Studio	Dessert Shop	Diner	Discount Store	Donut Shop	\
0	0.01	0.00	0.00	0.00	0.00	
1	0.01	0.00	0.00	0.00	0.00	
2	0.00	0.02	0.00	0.00	0.00	
3	0.00	0.00	0.00	0.00	0.00	
4	0.01	0.02	0.00	0.00	0.00	
5	0.00	0.00	0.01	0.01	0.01	
6	0.00	0.00	0.01	0.00	0.01	
7	0.01	0.02	0.00	0.00	0.00	
8	0.00	0.02	0.01	0.00	0.00	

	English Restaurant	Ethiopian Restaurant	Exhibit	Farmers Market	\
0	0.02	0.00	0.00	0.01	
1	0.02	0.00	0.00	0.01	
2	0.00	0.00	0.01	0.01	
3	0.03	0.00	0.02	0.00	
4	0.01	0.00	0.03	0.00	
5	0.00	0.01	0.00	0.00	
6	0.01	0.00	0.00	0.00	
7	0.01	0.00	0.03	0.00	
8	0.01	0.00	0.00	0.01	

	Fast Food Restaurant	Filipino Restaurant	Fish & Chips Shop	Fountain	\
0	0.00	0.00	0.00	0.00	
1	0.00	0.00	0.00	0.00	
2	0.00	0.00	0.00	0.00	
3	0.01	0.01	0.00	0.00	
4	0.00	0.00	0.00	0.01	
5	0.00	0.00	0.02	0.00	
6	0.00	0.00	0.02	0.00	

7	0.00	0.00	0.00	0.01
8	0.00	0.00	0.00	0.00

	French Restaurant	Furniture / Home	Store	Garden	Garden Center	\
0	0.05		0.00	0.02	0.01	
1	0.05		0.00	0.02	0.01	
2	0.03		0.00	0.02	0.00	
3	0.03		0.00	0.03	0.00	
4	0.01		0.00	0.03	0.00	
5	0.01		0.00	0.00	0.00	
6	0.01		0.00	0.01	0.00	
7	0.01		0.00	0.03	0.00	
8	0.03		0.01	0.02	0.01	

	Gastropub	Gift Shop	Gourmet Shop	Greek Restaurant	\
0	0.03	0.01	0.01	0.00	
1	0.03	0.01	0.01	0.00	
2	0.02	0.00	0.00	0.00	
3	0.01	0.01	0.00	0.01	
4	0.01	0.02	0.01	0.00	
5	0.01	0.00	0.01	0.01	
6	0.01	0.00	0.01	0.01	
7	0.01	0.02	0.01	0.00	
8	0.02	0.00	0.00	0.00	

	Grilled Meat Restaurant	Grocery Store	Gym	Gym / Fitness Center	\
0	0.00	0.01	0.00	0.02	
1	0.00	0.01	0.00	0.02	
2	0.01	0.00	0.00	0.04	
3	0.00	0.01	0.01	0.03	
4	0.00	0.00	0.00	0.01	
5	0.00	0.01	0.02	0.05	
6	0.00	0.02	0.01	0.05	
7	0.00	0.00	0.00	0.01	
8	0.00	0.00	0.00	0.02	

	Health & Beauty Service	Herbs & Spices Store	Historic Site	\
0	0.01	0.00	0.00	
1	0.01	0.00	0.00	
2	0.00	0.00	0.01	
3	0.01	0.00	0.01	
4	0.00	0.00	0.00	
5	0.00	0.00	0.00	
6	0.01	0.01	0.00	
7	0.00	0.00	0.00	
8	0.00	0.00	0.00	

History Museum	Hotel	Ice Cream Shop	Indian Restaurant	\
----------------	-------	----------------	-------------------	---

0	0.00	0.03	0.04	0.01
1	0.00	0.03	0.04	0.01
2	0.00	0.08	0.02	0.02
3	0.01	0.04	0.00	0.02
4	0.00	0.06	0.02	0.01
5	0.00	0.00	0.01	0.00
6	0.00	0.01	0.01	0.01
7	0.00	0.06	0.02	0.01
8	0.00	0.04	0.01	0.01

	Indie Movie Theater	Indie Theater	Italian Restaurant	\
0	0.00	0.00	0.03	
1	0.00	0.00	0.03	
2	0.01	0.01	0.05	
3	0.00	0.00	0.04	
4	0.01	0.00	0.07	
5	0.00	0.00	0.05	
6	0.00	0.00	0.06	
7	0.01	0.00	0.07	
8	0.01	0.01	0.07	

	Japanese Restaurant	Jazz Club	Jewelry Store	Juice Bar	\
0	0.03	0.00	0.00	0.02	
1	0.03	0.00	0.00	0.02	
2	0.02	0.00	0.00	0.01	
3	0.01	0.00	0.00	0.02	
4	0.04	0.00	0.02	0.00	
5	0.01	0.00	0.00	0.02	
6	0.00	0.00	0.00	0.02	
7	0.04	0.00	0.02	0.00	
8	0.02	0.01	0.00	0.02	

	Latin American Restaurant	Lebanese Restaurant	Lounge	Malay Restaurant	\
0	0.00	0.00	0.00	0.00	
1	0.00	0.00	0.00	0.00	
2	0.00	0.00	0.00	0.00	
3	0.00	0.00	0.00	0.00	
4	0.00	0.01	0.00	0.00	
5	0.01	0.00	0.01	0.01	
6	0.01	0.00	0.01	0.00	
7	0.00	0.01	0.00	0.00	
8	0.00	0.00	0.01	0.00	

	Market	Massage Studio	Mediterranean Restaurant	Mexican Restaurant	\
0	0.00	0.00	0.00	0.00	
1	0.00	0.00	0.00	0.00	
2	0.00	0.00	0.01	0.00	
3	0.00	0.00	0.02	0.00	

4	0.00	0.00	0.01	0.00
5	0.01	0.00	0.00	0.00
6	0.01	0.01	0.00	0.01
7	0.00	0.00	0.01	0.00
8	0.00	0.00	0.00	0.00

	Middle Eastern Restaurant	Modern European Restaurant	Monument / Landmark	\
0	0.00	0.00	0.01	
1	0.00	0.00	0.01	
2	0.02	0.01	0.00	
3	0.01	0.01	0.01	
4	0.01	0.00	0.01	
5	0.01	0.01	0.00	
6	0.01	0.00	0.00	
7	0.01	0.00	0.01	
8	0.00	0.01	0.00	

	Moroccan Restaurant	Movie Theater	Museum	Music Venue	Opera House	\
0	0.00	0.00	0.00	0.00	0.00	
1	0.00	0.00	0.00	0.00	0.00	
2	0.00	0.00	0.01	0.00	0.00	
3	0.00	0.01	0.01	0.00	0.01	
4	0.00	0.01	0.00	0.00	0.00	
5	0.01	0.01	0.01	0.01	0.00	
6	0.00	0.02	0.01	0.01	0.00	
7	0.00	0.01	0.00	0.00	0.00	
8	0.00	0.00	0.01	0.01	0.00	

	Palace	Park	Pastry Shop	Perfume Shop	Persian Restaurant	Pharmacy	\
0	0.00	0.01	0.01	0.01	0.00	0.00	
1	0.00	0.01	0.01	0.01	0.00	0.00	
2	0.00	0.02	0.00	0.00	0.01	0.01	
3	0.01	0.02	0.00	0.00	0.01	0.01	
4	0.00	0.00	0.01	0.01	0.00	0.01	
5	0.00	0.03	0.00	0.00	0.00	0.00	
6	0.00	0.03	0.00	0.00	0.00	0.00	
7	0.00	0.00	0.01	0.01	0.00	0.01	
8	0.00	0.02	0.00	0.00	0.00	0.00	

	Piano Bar	Pilates Studio	Pizza Place	Playground	Plaza	\
0	0.00	0.00	0.01	0.00	0.02	
1	0.00	0.00	0.01	0.00	0.02	
2	0.00	0.01	0.05	0.00	0.00	
3	0.01	0.00	0.01	0.01	0.00	
4	0.00	0.00	0.01	0.00	0.00	
5	0.00	0.00	0.04	0.00	0.00	
6	0.00	0.00	0.03	0.00	0.00	
7	0.00	0.00	0.01	0.00	0.00	

8	0.00	0.00	0.04	0.00	0.00
---	------	------	------	------	------

	Polish Restaurant	Portuguese Restaurant	Pub	Ramen Restaurant	\
0	0.00		0.02		0.00
1	0.00		0.02		0.00
2	0.00		0.01	0.04	0.00
3	0.00		0.01	0.07	0.00
4	0.01		0.00	0.01	0.00
5	0.00		0.00	0.10	0.01
6	0.00		0.00	0.08	0.01
7	0.01		0.00	0.01	0.00
8	0.00		0.00	0.03	0.00

	Record Shop	Recording Studio	Restaurant	Road	Rock Climbing Spot	\
0	0.00	0.00	0.02	0.00		0.00
1	0.00	0.00	0.02	0.00		0.00
2	0.00	0.00	0.02	0.00		0.00
3	0.00	0.01	0.04	0.00		0.00
4	0.00	0.00	0.02	0.00		0.00
5	0.02	0.00	0.03	0.01		0.00
6	0.02	0.00	0.02	0.01		0.00
7	0.00	0.00	0.02	0.00		0.00
8	0.00	0.00	0.03	0.00		0.01

	Russian Restaurant	Sandwich Place	Science Museum	Seafood Restaurant	\
0	0.00	0.02		0.00	0.00
1	0.00	0.02		0.00	0.00
2	0.00	0.01		0.02	0.01
3	0.00	0.01		0.01	0.00
4	0.01	0.02		0.04	0.01
5	0.00	0.00		0.00	0.00
6	0.00	0.00		0.00	0.00
7	0.01	0.02		0.04	0.01
8	0.00	0.02		0.00	0.01

	Shopping Plaza	Skate Park	Soccer Stadium	Social Club	Speakeasy	\
0	0.01	0.00	0.00		0.00	0.01
1	0.01	0.00	0.00		0.00	0.01
2	0.00	0.00	0.02		0.01	0.01
3	0.00	0.00	0.00		0.00	0.00
4	0.00	0.00	0.00		0.01	0.01
5	0.00	0.01	0.00		0.00	0.00
6	0.00	0.01	0.00		0.00	0.00
7	0.00	0.00	0.00		0.01	0.01
8	0.01	0.00	0.02		0.01	0.01

	Sporting Goods Shop	Sports Bar	Steakhouse	Street Food Gathering	\
0	0.00	0.00	0.01		0.00

1	0.00	0.00	0.01	0.00
2	0.01	0.00	0.01	0.00
3	0.00	0.00	0.00	0.00
4	0.00	0.00	0.02	0.00
5	0.00	0.00	0.01	0.01
6	0.00	0.00	0.00	0.01
7	0.00	0.00	0.02	0.00
8	0.01	0.01	0.01	0.00

	Supermarket	Sushi Restaurant	Tapas Restaurant	Tea Room	Thai Restaurant	\
0	0.01	0.00	0.00	0.01	0.00	
1	0.01	0.00	0.00	0.01	0.00	
2	0.02	0.01	0.02	0.00	0.03	
3	0.02	0.01	0.00	0.01	0.02	
4	0.01	0.01	0.00	0.00	0.00	
5	0.00	0.00	0.00	0.01	0.01	
6	0.00	0.00	0.00	0.01	0.00	
7	0.01	0.01	0.00	0.00	0.00	
8	0.02	0.01	0.02	0.00	0.00	

	Theater	Thrift / Vintage Store	Toy / Game Store	Turkish Restaurant	\
0	0.01	0.00	0.00	0.00	
1	0.01	0.00	0.00	0.00	
2	0.00	0.00	0.00	0.00	
3	0.00	0.00	0.00	0.00	
4	0.00	0.00	0.01	0.00	
5	0.00	0.01	0.00	0.01	
6	0.00	0.01	0.00	0.01	
7	0.00	0.00	0.01	0.00	
8	0.00	0.00	0.00	0.00	

	University	Vietnamese Restaurant	Wine Bar	Wine Shop	Women's Store	\
0	0.00	0.02	0.00	0.00	0.02	
1	0.00	0.02	0.00	0.00	0.02	
2	0.00	0.01	0.00	0.02	0.00	
3	0.01	0.00	0.01	0.00	0.01	
4	0.01	0.01	0.00	0.01	0.00	
5	0.00	0.00	0.01	0.01	0.00	
6	0.00	0.00	0.01	0.00	0.00	
7	0.01	0.01	0.00	0.01	0.00	
8	0.00	0.02	0.00	0.01	0.01	

Yoga Studio	
0	0.00
1	0.00
2	0.02
3	0.02
4	0.01



5	0.02
6	0.02
7	0.01
8	0.02

```
In [115]: num_top_venues = 5
```

```
for hood in uk_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = uk_grouped[uk_grouped['Neighborhood'] == hood].T.reset_index()
    temp.columns = ['venue', 'freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

```
----Brompton----
```

	venue	freq
0	Café	0.07
1	French Restaurant	0.05
2	Ice Cream Shop	0.04
3	Bakery	0.04
4	Hotel	0.03

```
----Chelsea----
```

	venue	freq
0	Café	0.07
1	French Restaurant	0.05
2	Ice Cream Shop	0.04
3	Bakery	0.04
4	Hotel	0.03

```
----Earls Court----
```

	venue	freq
0	Hotel	0.08
1	Italian Restaurant	0.05
2	Pizza Place	0.05
3	Gym / Fitness Center	0.04
4	Pub	0.04

```
----Holland Park----
```

	venue	freq
0	Pub	0.07
1	Café	0.04

2	Hotel	0.04
3	Bakery	0.04
4	Italian Restaurant	0.04

----Kensington----

	venue	freq
0	Italian Restaurant	0.07
1	Café	0.07
2	Hotel	0.06
3	Japanese Restaurant	0.04
4	Science Museum	0.04

----North Kensington----

	venue	freq
0	Pub	0.10
1	Gym / Fitness Center	0.05
2	Italian Restaurant	0.05
3	Bakery	0.05
4	Cocktail Bar	0.04

----Notting Hill----

	venue	freq
0	Pub	0.08
1	Italian Restaurant	0.06
2	Gym / Fitness Center	0.05
3	Bakery	0.04
4	Café	0.03

----South Kensington----

	venue	freq
0	Italian Restaurant	0.07
1	Café	0.07
2	Hotel	0.06
3	Japanese Restaurant	0.04
4	Science Museum	0.04

----West Brompton----

	venue	freq
0	Italian Restaurant	0.07
1	Café	0.06
2	Bakery	0.05
3	Pizza Place	0.04
4	Hotel	0.04

```
In [116]: #a function to sort the venues in descending order
```

```
def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

```
In [117]: #create the new dataframe and display the top 10 venues for each neighborhood.
```

```
num_top_venues = 10
word_string1 = ''
indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighborhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighborhoods_venues_sorted_uk = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted_uk['Neighborhood'] = uk_grouped['Neighborhood']

for ind in np.arange(uk_grouped.shape[0]):
    neighborhoods_venues_sorted_uk.iloc[ind, 1:] = return_most_common_venues(uk_grouped.iloc[ind, 1:], num_top_venues)
    word_string1 = word_string1 + neighborhoods_venues_sorted_uk.iloc[ind, 1:] + ' '

neighborhoods_venues_sorted_uk
```

```
Out[117]:
```

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	\
0	Brompton	Café	French Restaurant	
1	Chelsea	Café	French Restaurant	
2	Earls Court	Hotel	Pizza Place	
3	Holland Park	Pub	Restaurant	
4	Kensington	Italian Restaurant	Café	
5	North Kensington	Pub	Italian Restaurant	
6	Notting Hill	Pub	Italian Restaurant	
7	South Kensington	Italian Restaurant	Café	
8	West Brompton	Italian Restaurant	Café	

  

	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	\
0	Ice Cream Shop	Bakery	Japanese Restaurant	

1	Ice Cream Shop	Bakery	Japanese Restaurant
2	Italian Restaurant	Pub	Gym / Fitness Center
3	Hotel	Bakery	Café
4	Hotel	Japanese Restaurant	Science Museum
5	Bakery	Gym / Fitness Center	Café
6	Gym / Fitness Center	Bakery	Park
7	Hotel	Japanese Restaurant	Science Museum
8	Bakery	Pizza Place	Hotel

	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	\
0	Coffee Shop	Italian Restaurant	Hotel	
1	Coffee Shop	Italian Restaurant	Hotel	
2	Thai Restaurant	Bakery	Café	
3	Italian Restaurant	Gym / Fitness Center	Burger Joint	
4	Burger Joint	Garden	Exhibit	
5	Pizza Place	Cocktail Bar	Bar	
6	Café	Breakfast Spot	Cocktail Bar	
7	Burger Joint	Garden	Exhibit	
8	Restaurant	Pub	French Restaurant	

	9th Most Common Venue	10th Most Common Venue
0	Cocktail Bar	Boutique
1	Cocktail Bar	Boutique
2	French Restaurant	Ice Cream Shop
3	Garden	French Restaurant
4	Sandwich Place	Steakhouse
5	Restaurant	Park
6	Pizza Place	Fish & Chips Shop
7	Sandwich Place	Steakhouse
8	Yoga Studio	Gym / Fitness Center

```
In [118]: #word_string1[0]
```

```
In [119]: word_string2 = ''
          for i in range(0, num_top_venues ):
              word_string2 = word_string2 + word_string1[i]
          #print (word_string2)
```

```
In [120]: # display the generated text
          #word_string

          # create the word cloud
          wordcloud = WordCloud(background_color='white').generate(word_string2)

          print('Word cloud created!')

          # display the cloud
          fig = plt.figure()
```

Word cloud created!



93

```

Out[122]:      neighborhoods Dialcode      borough posttown postcode \
0      Brompton      020 Kensington and Chelsea LONDON      SW3
1      Chelsea      020 Kensington and Chelsea LONDON      SW3
2      Earls Court      020 Kensington and Chelsea LONDON      SW5
3      Kensington      020 Kensington and Chelsea LONDON      SW7
4      South Kensington      020 Kensington and Chelsea LONDON      SW7

      Latitude Longitude Cluster Labels 1st Most Common Venue \
0      51.49014      -0.16248      2      Café
1      51.49014      -0.16248      2      Café
2      51.49004      -0.18971      3      Hotel
3      51.49807      -0.17404      3      Italian Restaurant
4      51.49807      -0.17404      1      Italian Restaurant

      2nd Most Common Venue 3rd Most Common Venue 4th Most Common Venue \
0      French Restaurant      Ice Cream Shop      Bakery
1      French Restaurant      Ice Cream Shop      Bakery
2      Pizza Place      Italian Restaurant      Pub
3      Café      Hotel      Japanese Restaurant
4      Café      Hotel      Japanese Restaurant

      5th Most Common Venue 6th Most Common Venue 7th Most Common Venue \
0      Japanese Restaurant      Coffee Shop      Italian Restaurant
1      Japanese Restaurant      Coffee Shop      Italian Restaurant
2      Gym / Fitness Center      Thai Restaurant      Bakery
3      Science Museum      Burger Joint      Garden
4      Science Museum      Burger Joint      Garden

      8th Most Common Venue 9th Most Common Venue 10th Most Common Venue
0      Hotel      Cocktail Bar      Boutique
1      Hotel      Cocktail Bar      Boutique
2      Café      French Restaurant      Ice Cream Shop
3      Exhibit      Sandwich Place      Steakhouse
4      Exhibit      Sandwich Place      Steakhouse

```

```

In [123]: # create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

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

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(uk_merged['Latitude'], uk_merged['Longitude'], uk_
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)

```

```

folium.CircleMarker(
    [lat, lon],
    radius=5,
    popup=label,
    color=rainbow[cluster-1],
    fill=True,
    fill_color=rainbow[cluster-1],
    fill_opacity=0.7).add_to(map_clusters)

```

map\_clusters

Out[123]: <folium.folium.Map at 0x7fe50cc6c5f8>

In [124]: # For Cluster 0

```

result = uk_merged.loc[uk_merged['Cluster Labels'] == 0, uk_merged.columns[[1] + list(venue_types.keys())]]
print("For cluster {}, the distribution of venues is as:\n{}".format(0, result['1st Most Common Venue']))
result

```

For cluster 0, the distribution of venues is as:

Italian Restaurant 1

Pub 1

Name: 1st Most Common Venue, dtype: int64

```

Out[124]:
   Dialcode  Latitude  Longitude  Cluster Labels  1st Most Common Venue \
5         020  51.48563   -0.18144             0    Italian Restaurant
6         020  51.50162   -0.19173             0                      Pub

   2nd Most Common Venue 3rd Most Common Venue 4th Most Common Venue \
5                   Café             Bakery          Pizza Place
6             Restaurant             Hotel             Bakery

   5th Most Common Venue 6th Most Common Venue 7th Most Common Venue \
5                   Hotel             Restaurant             Pub
6                   Café    Italian Restaurant  Gym / Fitness Center

   8th Most Common Venue 9th Most Common Venue 10th Most Common Venue
5    French Restaurant             Yoga Studio  Gym / Fitness Center
6      Burger Joint             Garden    French Restaurant

```

In [129]: # For Cluster 1

```

result = uk_merged.loc[uk_merged['Cluster Labels'] == 1, uk_merged.columns[[1] + list(venue_types.keys())]]
print("For cluster {}, the distribution of venues is as:\n{}".format(1, result['1st Most Common Venue']))
result

```

For cluster 1, the distribution of venues is as:

Italian Restaurant 1

Pub 1

Name: 1st Most Common Venue, dtype: int64

```

Out[129]:   Dialãcode  Latitude  Longitude  Cluster Labels  1st Most Common Venue  \
4         020  51.49807   -0.17404                1    Italian Restaurant
7         020  51.52346   -0.21353                1                      Pub

      2nd Most Common Venue 3rd Most Common Venue 4th Most Common Venue  \
4                      Café                Hotel    Japanese Restaurant
7    Italian Restaurant                Bakery  Gym / Fitness Center

      5th Most Common Venue 6th Most Common Venue 7th Most Common Venue  \
4          Science Museum                Burger Joint                Garden
7                      Café                Pizza Place                Cocktail Bar

      8th Most Common Venue 9th Most Common Venue 10th Most Common Venue
4                      Exhibit                Sandwich Place                Steakhouse
7                      Bar                Restaurant                Park

```

```
In [130]: # For Cluster 2
```

```

result = uk_merged.loc[uk_merged['Cluster Labels'] == 2, uk_merged.columns[[1] + list(
print("For cluster {}, the distribution of venues is as:\n{}".format(2, result['1st Most Common Venue'])
result

```

For cluster 2, the distribution of venues is as:

Café 2

Name: 1st Most Common Venue, dtype: int64

```

Out[130]:   Dialãcode  Latitude  Longitude  Cluster Labels  1st Most Common Venue  \
0         020  51.49014   -0.16248                2          Café
1         020  51.49014   -0.16248                2          Café

      2nd Most Common Venue 3rd Most Common Venue 4th Most Common Venue  \
0    French Restaurant                Ice Cream Shop                Bakery
1    French Restaurant                Ice Cream Shop                Bakery

      5th Most Common Venue 6th Most Common Venue 7th Most Common Venue  \
0    Japanese Restaurant                Coffee Shop    Italian Restaurant
1    Japanese Restaurant                Coffee Shop    Italian Restaurant

      8th Most Common Venue 9th Most Common Venue 10th Most Common Venue
0                      Hotel                Cocktail Bar                Boutique
1                      Hotel                Cocktail Bar                Boutique

```

```
In [131]: # For Cluster 3
```

```

result = uk_merged.loc[uk_merged['Cluster Labels'] == 3, uk_merged.columns[[1] + list(
print("For cluster {}, the distribution of venues is as:\n{}".format(3, result['1st Most Common Venue'])
result

```

For cluster 3, the distribution of venues is as:

Hotel 1



```
Italian Restaurant    1
Pub                  1
Name: 1st Most Common Venue, dtype: int64
```

```
Out[131]:  Dialãcode  Latitude  Longitude  Cluster Labels  1st Most Common Venue  \
2           020   51.49004   -0.18971                3                Hotel
3           020   51.49807   -0.17404                3      Italian Restaurant
8           020   51.51244   -0.20639                3                Pub

      2nd Most Common Venue  3rd Most Common Venue  4th Most Common Venue  \
2           Pizza Place      Italian Restaurant                Pub
3                Café                Hotel      Japanese Restaurant
8      Italian Restaurant  Gym / Fitness Center                Bakery

      5th Most Common Venue  6th Most Common Venue  7th Most Common Venue  \
2  Gym / Fitness Center      Thai Restaurant                Bakery
3      Science Museum      Burger Joint                Garden
8                Park                Café      Breakfast Spot

      8th Most Common Venue  9th Most Common Venue  10th Most Common Venue
2                Café      French Restaurant      Ice Cream Shop
3                Exhibit      Sandwich Place                Steakhouse
8      Cocktail Bar      Pizza Place      Fish & Chips Shop
```

## Entertainment in NYC

```
In [134]: #address = '102 North End Ave, New York, NY'
address = '575 5th Ave, New York'
```

```
geolocator = Nominatim(user_agent="foursquare_agent")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print(latitude, longitude)
```

```
40.7563907 -73.9782057
```

```
In [135]: #let's define a query to search for entertainment that is within 2000 metres in nyc.
search_query = 'Escape Games'
radius = 2000
print(search_query + ' .... OK!')

# Define the corresponding URL
url = 'https://api.foursquare.com/v2/venues/search?client_id={} & client_secret={} & ll=
url
```

```
Escape Games ... OK!
```

```
Out[135]: 'https://api.foursquare.com/v2/venues/search?client_id=B3D1FREXU3FMFKGOXFFFWLZH1UBNQ'
```

```
In [170]: # Send the GET Request and examine the results
results = requests.get(url).json()
#results
```

```
In [137]: # assign relevant part of JSON to venues
venues = results['response']['venues']

# transform venues into a dataframe
dataframe = json_normalize(venues)
dataframe.head()
```

```
Out[137]:
```

		categories	hasPerk	\
0	[{'id': '4bf58dd8d48988d1f1931735', 'name': 'G...		False	
1	[{'id': '4bf58dd8d48988d1f1931735', 'name': 'G...		False	
2	[{'id': '4bf58dd8d48988d1ed941735', 'name': 'S...		False	
3	[{'id': '4bf58dd8d48988d1f1931735', 'name': 'G...		False	
4		[]	False	

  

	id	location.address	location.cc	\
0	5bf0c411364d970034315383	265 W 37th St	US	
1	56abc458498ee6c0b6ad7f0a	435 5th Avenue, 4th Floor	US	
2	4bc93a3b3740b713cc825e65	101 W 55th St	US	
3	59c4073de1f2282d14b2cc69	246 W 38th St Fl 7	US	
4	5577e333498ed6ee66399419	50 Central Park S	US	

  

	location.city	location.country	location.crossStreet	location.distance	\
0	New York	United States	NaN	1172	
1	New York	United States	NaN	656	
2	New York	United States	6th Ave, NW corner	767	
3	New York	United States	NaN	1058	
4	New York	United States	NaN	1002	

  

	location.formattedAddress	\
0	[265 W 37th St, New York, NY 10018, United Sta...	
1	[435 5th Avenue, 4th Floor, New York, NY 10016...	
2	[101 W 55th St (6th Ave, NW corner), New York,...	
3	[246 W 38th St Fl 7, New York, NY 10018, Unite...	
4	[50 Central Park S, New York, NY 10019, United...	

  

	location.labeledLatLngs	location.lat	\
0	[{'label': 'display', 'lat': 40.754016, 'lng':...	40.754016	
1	[{'label': 'display', 'lat': 40.75139156486494...	40.751392	
2	[{'label': 'display', 'lat': 40.76327188497643...	40.763272	
3	[{'label': 'display', 'lat': 40.75409, 'lng': ...	40.754090	
4	[{'label': 'display', 'lat': 40.76525, 'lng': ...	40.765250	

	location.lng	location.neighborhood	location.postalCode	location.state	\
0	-73.991748		10018	NY	
1	-73.982329	Garment District	10016	NY	
2	-73.978747		10019	NY	
3	-73.990390		10018	NY	
4	-73.976040		10019	NY	

	name	referralId	venuePage.id
0	Mission Escape Games	v-1564735154	NaN
1	Riddle Me Out Escape Games NYC	v-1564735154	152966516
2	Escape Day Spa & Skin Care	v-1564735154	NaN
3	Exit Escape Room NYC	v-1564735154	453995455
4	Escape Models VIP	v-1564735154	NaN

```
In [138]: # keep only columns that include venue name, and anything that is associated with lo
filtered_columns = ['name', 'categories'] + [col for col in dataframe.columns if col
dataframe_filtered = dataframe.loc[:, filtered_columns]
```

```
# function that extracts the category of the venue
```

```
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

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

```
# filter the category for each row
```

```
dataframe_filtered['categories'] = dataframe_filtered.apply(get_category_type, axis=
```

```
# clean column names by keeping only last term
```

```
dataframe_filtered.columns = [column.split('.')[0] for column in dataframe_filtered
```

```
dataframe_filtered.head()
```

```
Out[138]:
```

	name	categories	\
0	Mission Escape Games	General Entertainment	
1	Riddle Me Out Escape Games NYC	General Entertainment	
2	Escape Day Spa & Skin Care	Spa	
3	Exit Escape Room NYC	General Entertainment	
4	Escape Models VIP	None	

	address	cc	city	country	crossStreet	\
0	265 W 37th St	US	New York	United States	NaN	
1	435 5th Avenue, 4th Floor	US	New York	United States	NaN	

2	101 W 55th St	US	New York	United States	6th Ave, NW corner
3	246 W 38th St Fl 7	US	New York	United States	NaN
4	50 Central Park S	US	New York	United States	NaN

	distance	formattedAddress	\
0	1172	[265 W 37th St, New York, NY 10018, United Sta...	
1	656	[435 5th Avenue, 4th Floor, New York, NY 10016...	
2	767	[101 W 55th St (6th Ave, NW corner), New York,...	
3	1058	[246 W 38th St Fl 7, New York, NY 10018, Unite...	
4	1002	[50 Central Park S, New York, NY 10019, United...	

	labeledLatLngs	lat	lng	\
0	[{'label': 'display', 'lat': 40.754016, 'lng': ...	40.754016	-73.991748	
1	[{'label': 'display', 'lat': 40.75139156486494...	40.751392	-73.982329	
2	[{'label': 'display', 'lat': 40.76327188497643...	40.763272	-73.978747	
3	[{'label': 'display', 'lat': 40.75409, 'lng': ...	40.754090	-73.990390	
4	[{'label': 'display', 'lat': 40.76525, 'lng': ...	40.765250	-73.976040	

	neighborhood	postalCode	state
0	NaN	10018	NY
1	Garment District	10016	NY
2	NaN	10019	NY
3	NaN	10018	NY
4	NaN	10019	NY

```
In [139]: dataframe_filtered1 = dataframe_filtered[dataframe_filtered.categories == 'General Entertainment']
dataframe_filtered1.head()
```

```
Out[139]:
```

	name	categories	\
0	Mission Escape Games	General Entertainment	
1	Riddle Me Out Escape Games NYC	General Entertainment	
3	Exit Escape Room NYC	General Entertainment	
6	Escape Room Madness	General Entertainment	
7	Escape Entertainment	General Entertainment	

	address	cc	city	country	\
0	265 W 37th St	US	New York	United States	
1	435 5th Avenue, 4th Floor	US	New York	United States	
3	246 W 38th St Fl 7	US	New York	United States	
6	38 West 32nd Street, 5th Floor, Ste 500	US	New York	United States	
7	4th Floor, 39 W 32nd St	US	New York	United States	

	crossStreet	distance	\
0	NaN	1172	
1	NaN	656	
3	NaN	1058	
6	32nd street & Broadway	1212	
7	NaN	1214	

	formattedAddress	\
0	[265 W 37th St, New York, NY 10018, United Sta...	
1	[435 5th Avenue, 4th Floor, New York, NY 10016...	
3	[246 W 38th St Fl 7, New York, NY 10018, Unite...	
6	[38 West 32nd Street, 5th Floor, Ste 500 (32nd...	
7	[4th Floor, 39 W 32nd St, New York, NY 10001, ...	

  

	labeledLatLngs	lat	lng	\
0	[{'label': 'display', 'lat': 40.754016, 'lng': ...	40.754016	-73.991748	
1	[{'label': 'display', 'lat': 40.75139156486494...	40.751392	-73.982329	
3	[{'label': 'display', 'lat': 40.75409, 'lng': ...	40.754090	-73.990390	
6	[{'label': 'display', 'lat': 40.74803054394054...	40.748031	-73.987428	
7	[{'label': 'display', 'lat': 40.74810986880671...	40.748110	-73.987582	

  

	neighborhood	postalCode	state
0	NaN	10018	NY
1	Garment District	10016	NY
3	NaN	10018	NY
6	NaN	10001	NY
7	NaN	10001	NY

```
In [140]: venues_ett_map = folium.Map(location=[latitude, longitude], zoom_start=13) # generat
```

```
# add the Italian restaurants as blue circle markers
for lat, lng, label in zip(dataframe_filtered.lat, dataframe_filtered.lng, dataframe.
    folium.features.CircleMarker(
        [lat, lng],
        radius=3,
        color='green',
        popup=label,
        fill = True,
        fill_color='green',
        fill_opacity=0.6
    ).add_to(venues_ett_map)

# add the Italian restaurants as blue circle markers
for lat, lng, label in zip(dataframe_filtered1.lat, dataframe_filtered1.lng, datafram
    folium.features.CircleMarker(
        [lat, lng],
        radius=3,
        color='red',
        popup=label,
        fill = True,
        fill_color='red',
        fill_opacity=0.6
    ).add_to(venues_ett_map)
```

```
# display map
venues_ett_map
```

```
Out[140]: <folium.folium.Map at 0x7fe50d76e898>
```

## 1.5 Results

Analyzing the results we can see that people in different borough of London and NYC often visit identical places, such as Italian Restaurant, Coffee Shop, Park, Pizza Place, Hotel, Gym, Fitness Center, But there are also differences in preferences, such as French Restaurant, Pub, Japanese Restaurant, Cocktail Bar, Boutique for borough of London and for borough of NYC - American Restaurant, Wine Shop, Chinese Restaurant, Sushi Restaurant, Taco Place. Also using Foursquare API and visualization we can easily see the information that we need, for example, the placement of Escape room. They are popular now, and how we can see on map there are only a few in Manhattan.

## 1.6 Discussion

Based on our result, we can conclude that there are few Escape rooms and if we want to open a Escape room then the best place in Manhattan is near the center and above. If to analyze Brooklyn we can see only 2 of them. That is a good decision and we are not occupied with a place at this moment.

Also analyzing area of London and New York, we see that the British prefer French Restaurant, Pub, Japanese Restaurant while the American prefer Mexican Restaurant, Chinese Restaurant, Sushi Restaurant, Taco Place.

## 1.7 Conclusion

Using Foursquare API, we can captured data of common places all around the world. Using it, we refer back to our main objectives, which is to determine;

the similarity or dissimilarirty of both cities classification of area located inside the city whether it is residential, tourism places, or others In conclusion, both borough cities London and NYC