Final Project

Clustering the Neighbourhoods with Chinese Restaurants of London

Introduction

This is the final project of the IBM Data Science Course on Coursera. In this project, I have analyzed the scenario of Chinese Restaurants in London. The results and data obtained mighb be used such as information for tourisms guides, neighbourhoods where it is possible to create a new chinese restaurant or areas to avoid in the city for this news Chinese restaurants. Whit these ideas in mind, I have developed this project to get the correct information for tourists or the best locations for new restaurant. The project will be developed with python, Foursquare API and all tools and mechanism that I have learned in this IBM Course.

Business problem

The aim of this project is to help to find the best location for any person that wants establish his Chinese Restaurant in London. In addition, this project can help to ellaborate a tourist guide with the better information about Chinese Restaurants in the capital of United Kingdom.

Data

It is necessary data about the boroughs(or Neighbourhoods), geolocation data of each borough and all the venues in each borough visited by the people in London. When we have obtained this information will be relevant the correct union of valid information in order to cluster the Neighbourhoods and obtain valuables results. We are going to use the following apps or websites pages in order to obtain the primary information:

1. Wikipedia
2. Foursquare API
3. ArcGIS geolocations

Wikipedia

To obtain the data of London's Borough, we have to scrape data from: https://en.wikipedia.org/wiki/List_of_areas_of_London

On this website we will obtain: Borough, Town and Postal code after clean the dataframes.

Foursquare API

We need credentials in order to obtain the information, so first of all we have to register in Foursquare Developer API https://foursquare.com/

All the information about venues location in London will be provided by Foursquare API. This information will be the cornerstone to elaborate this project.

On this website we will obtain: Neighbourhoods, latitude, longitude, venues and venues category.

ArcGIS geolocations

Arcgis is a System of Geographic Information (GIS) that provide to the project the coordinates of each neighbourhood and the city of London in order to obtain the maps and make the cluster with Folium.

Import libraries

Import the python libraries on python

In [1]:

```
import pandas as pd
import numpy as np
import requests
```

```python
import matplotlib.pyplot as plt
%matplotlib inline
```

Scrapping the wikipedia data

The next codes cell will be in ordert to get the dataframe from wikipedia about London Boroughs

In [4]:

```python
wiki_url = "https://en.wikipedia.org/wiki/List_of_areas_of_London"
wiki_page = requests.get(wiki_url)
wiki_data = pd.read_html(wiki_page.text)
wiki_data
```

Out[4]:

```
[                                                   0
 0  Map all coordinates in "Category:Areas of Lond...
 1                 Download coordinates as: KML · GPX,
             Location                     London borough       Post town  \
 0         Abbey Wood              Bexley, Greenwich [7]          LONDON
 1              Acton  Ealing, Hammersmith and Fulham[8]          LONDON
 2          Addington                         Croydon[8]         CROYDON
 3         Addiscombe                         Croydon[8]         CROYDON
 4        Albany Park                             Bexley  BEXLEY, SIDCUP
 ..               ...                                ...             ...
 526         Woolwich                          Greenwich          LONDON
 527   Worcester Park       Sutton, Kingston upon Thames  WORCESTER PARK
 528  Wormwood Scrubs             Hammersmith and Fulham          LONDON
 529          Yeading                         Hillingdon           HAYES
 530         Yiewsley                         Hillingdon    WEST DRAYTON

     Postcode district Dial code OS grid ref
 0                 SE2       020    TQ465785
 1              W3, W4       020    TQ205805
 2                 CR0       020    TQ375645
 3                 CR0       020    TQ345665
 4           DA5, DA14       020    TQ478728
 ..                ...       ...         ...
 526              SE18       020    TQ435795
 527               KT4       020    TQ225655
 528               W12       020    TQ225815
 529               UB4       020    TQ115825
 530               UB7       020    TQ063804

 [531 rows x 6 columns],
     0                                                  1
 0 NaN  Wikimedia Commons has media related to Distric...,
   .mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-output .navbar-collapse{f
loat:left;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:i
nline-block;white-space:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125
em;content:"[ "}.mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:" ]"}.mw-parser-output .na
vbar li{word-spacing:-0.125em}.mw-parser-output .navbar-mini abbr{font-variant:small-caps;border-bottom:none;text
-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .
navbar-ct-mini{font-size:114%;margin:0 4em}.mw-parser-output .infobox .navbar{font-size:100%}.mw-parser-output .n
avbox .navbar{display:block;font-size:100%}.mw-parser-output .navbox-title .navbar{float:left;text-align:left;mar
gin-right:0.5em}vteAreas of London  \
 0                            Central activities zone
 1                                 Town centrenetwork
 2                                      International
 3                                       Metropolitan
 4                                              Major
 5                               Districts(principal)
 6                          Neighbourhoods(principal)
 7                           Lists of areasby borough
 8                                          Fictional

   .mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-output .navbar-collapse{f
loat:left;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:i
nline-block;white-space:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125
em;content:"[ "}.mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:" ]"}.mw-parser-output .na
vbar li{word-spacing:-0.125em}.mw-parser-output .navbar-mini abbr{font-variant:small-caps;border-bottom:none;text
-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .
navbar-ct-mini{font-size:114%;margin:0 4em}.mw-parser-output .infobox .navbar{font-size:100%}.mw-parser-output .n
avbox .navbar{display:block;font-size:100%}.mw-parser-output .navbox-title .navbar{float:left;text-align:left;mar
gin-right:0.5em}vteAreas of London.1
 0  Bloomsbury City of London wards Holborn Maryle...
 1  International Knightsbridge West End Metropoli...
 2                             Knightsbridge West End
 3  Bromley Croydon Ealing Harrow Hounslow Ilford ...
 4  Angel Barking Bexleyheath Brixton Camden Town ...
 5  Acton Beckenham Belgravia Bethnal Green Brentf...
 6  Abbey Wood Alperton Anerley Archway Barnes Bar...
 7  Barking and Dagenham Barnet Bexley Brent Broml...
 8  Canley (borough) (The Bill: TV soap) Charnham ...
,
                            0  \
 0              International
 1               Metropolitan
 2                      Major
 3        Districts(principal)
 4   Neighbourhoods(principal)

                                                    1
 0                             Knightsbridge West End
 1  Bromley Croydon Ealing Harrow Hounslow Ilford ...
 2  Angel Barking Bexleyheath Brixton Camden Town ...
 3  Acton Beckenham Belgravia Bethnal Green Brentf...
 4  Abbey Wood Alperton Anerley Archway Barnes Bar...  ]
```

In [5]:

```
len(wiki_data)
```

Out[5]:

```
5
```

In [6]:

```
type(wiki_data)
```

Out[6]:

```
list
```

Cleanning dataframe and renaming the columns

In [7]:

```
wiki_df = wiki_data[1]
wiki_df.head()
wiki_df
```

Out[7]:

	Location	London borough	Post town	Postcode district	Dial code	OS grid ref
0	Abbey Wood	Bexley, Greenwich [7]	LONDON	SE2	020	TQ465785
1	Acton	Ealing, Hammersmith and Fulham[8]	LONDON	W3, W4	020	TQ205805
2	Addington	Croydon[8]	CROYDON	CR0	020	TQ375645
3	Addiscombe	Croydon[8]	CROYDON	CR0	020	TQ345665
4	Albany Park	Bexley	BEXLEY, SIDCUP	DA5, DA14	020	TQ478728
...	...	...	...	...	...	...
526	Woolwich	Greenwich	LONDON	SE18	020	TQ435795
527	Worcester Park	Sutton, Kingston upon Thames	WORCESTER PARK	KT4	020	TQ225655
528	Wormwood Scrubs	Hammersmith and Fulham	LONDON	W12	020	TQ225815
529	Yeading	Hillingdon	HAYES	UB4	020	TQ115825
530	Yiewsley	Hillingdon	WEST DRAYTON	UB7	020	TQ063804

531 rows × 6 columns

In [8]:

```
df = wiki_df.drop( [ wiki_df.columns[0], wiki_df.columns[4], wiki_df.columns[5] ], axis=1)
df
```

Out[8]:

	London borough	Post town	Postcode district
0	Bexley, Greenwich [7]	LONDON	SE2
1	Ealing, Hammersmith and Fulham[8]	LONDON	W3, W4
2	Croydon[8]	CROYDON	CR0
3	Croydon[8]	CROYDON	CR0
4	Bexley	BEXLEY, SIDCUP	DA5, DA14
...	...	...	...
526	Greenwich	LONDON	SE18
527	Sutton, Kingston upon Thames	WORCESTER PARK	KT4
528	Hammersmith and Fulham	LONDON	W12
529	Hillingdon	HAYES	UB4
530	Hillingdon	WEST DRAYTON	UB7

531 rows × 3 columns

In [9]:

```
df.columns = ['Borough','Town','Post-code']
list(df.columns.values)
```

Out[9]:

```
['Borough', 'Town', 'Post-code']
```

In [10]:

```
df
```

Out[10]:

	Borough	Town	Post-code
0	Bexley, Greenwich [7]	LONDON	SE2
1	Ealing, Hammersmith and Fulham[8]	LONDON	W3, W4
2	Croydon[8]	CROYDON	CR0
3	Croydon[8]	CROYDON	CR0
4	Bexley	BEXLEY, SIDCUP	DA5, DA14
...	...	...	...
526	Greenwich	LONDON	SE18
527	Sutton, Kingston upon Thames	WORCESTER PARK	KT4
528	Hammersmith and Fulham	LONDON	W12
529	Hillingdon	HAYES	UB4
530	Hillingdon	WEST DRAYTON	UB7

531 rows × 3 columns

In [11]:

```
df['Borough'] = df['Borough'].map(lambda x: x.rstrip(']').rstrip('0123456789').rstrip('['))
df
```

Out[11]:

	Borough	Town	Post-code
0	Bexley, Greenwich	LONDON	SE2
1	Ealing, Hammersmith and Fulham	LONDON	W3, W4
2	Croydon	CROYDON	CR0
3	Croydon	CROYDON	CR0
4	Bexley	BEXLEY, SIDCUP	DA5, DA14
...	...	...	...
526	Greenwich	LONDON	SE18
527	Sutton, Kingston upon Thames	WORCESTER PARK	KT4
528	Hammersmith and Fulham	LONDON	W12
529	Hillingdon	HAYES	UB4
530	Hillingdon	WEST DRAYTON	UB7

531 rows × 3 columns

Obtain the shape and info dataframe

In [12]:

```
df.shape
```

Out[12]:

```
(531, 3)
```

In [13]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 531 entries, 0 to 530
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Borough    531 non-null    object
 1   Town       531 non-null    object
 2   Post-code  531 non-null    object
dtypes: object(3)
memory usage: 12.6+ KB
```

Install arcgis and obtain the coords

In [118…

```
!pip install arcgis
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting arcgis
  Downloading arcgis-1.8.3.post2.tar.gz (2.6 MB)
     |████████████████████████████████| 2.6 MB 2.7 MB/s eta 0:00:01
Requirement already satisfied: ipywidgets>=7 in /home/joseph/.local/lib/python3.8/site-packages (from arcgis) (7.
5.1)
Requirement already satisfied: matplotlib in /home/joseph/.local/lib/python3.8/site-packages (from arcgis) (3.3.2
)
Requirement already satisfied: numpy>=1.16.2 in /home/joseph/.local/lib/python3.8/site-packages (from arcgis) (1.
19.4)
Requirement already satisfied: pandas>=1 in /home/joseph/.local/lib/python3.8/site-packages (from arcgis) (1.2.0)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from arcgis) (2.22.0)
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from arcgis) (1.14.0)
Requirement already satisfied: widgetsnbextension>=3 in /home/joseph/.local/lib/python3.8/site-packages (from arc
gis) (3.5.1)
Requirement already satisfied: traitlets>=4.3.1 in /home/joseph/.local/lib/python3.8/site-packages (from ipywidge
ts>=7->arcgis) (5.0.4)
Requirement already satisfied: ipython>=4.0.0 in /home/joseph/.local/lib/python3.8/site-packages (from ipywidgets
>=7->arcgis) (7.18.1)
Requirement already satisfied: nbformat>=4.2.0 in /home/joseph/.local/lib/python3.8/site-packages (from ipywidget
s>=7->arcgis) (5.0.7)
Requirement already satisfied: ipykernel>=4.5.1 in /home/joseph/.local/lib/python3.8/site-packages (from ipywidge
ts>=7->arcgis) (5.3.4)
Requirement already satisfied: tornado>=4.2 in /home/joseph/.local/lib/python3.8/site-packages (from ipykernel>=4
.5.1->ipywidgets>=7->arcgis) (6.0.4)
Requirement already satisfied: jupyter-client in /home/joseph/.local/lib/python3.8/site-packages (from ipykernel>
=4.5.1->ipywidgets>=7->arcgis) (6.1.7)
Requirement already satisfied: jedi>=0.10 in /home/joseph/.local/lib/python3.8/site-packages (from ipython>=4.0.0
->ipywidgets>=7->arcgis) (0.17.2)
Requirement already satisfied: pexpect>4.3 in /usr/lib/python3/dist-packages (from ipython>=4.0.0->ipywidgets>=7-
>arcgis) (4.6.0)
Requirement already satisfied: setuptools>=18.5 in /usr/lib/python3/dist-packages (from ipython>=4.0.0->ipywidget
s>=7->arcgis) (45.2.0)
Requirement already satisfied: pickleshare in /home/joseph/.local/lib/python3.8/site-packages (from ipython>=4.0.
0->ipywidgets>=7->arcgis) (0.7.5)
Requirement already satisfied: pygments in /home/joseph/.local/lib/python3.8/site-packages (from ipython>=4.0.0->
ipywidgets>=7->arcgis) (2.7.1)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /home/joseph/.local/lib/python3.8/
site-packages (from ipython>=4.0.0->ipywidgets>=7->arcgis) (3.0.7)
Requirement already satisfied: backcall in /home/joseph/.local/lib/python3.8/site-packages (from ipython>=4.0.0->
ipywidgets>=7->arcgis) (0.2.0)
Requirement already satisfied: decorator in /home/joseph/.local/lib/python3.8/site-packages (from ipython>=4.0.0-
>ipywidgets>=7->arcgis) (4.4.2)
Requirement already satisfied: parso<0.8.0,>=0.7.0 in /home/joseph/.local/lib/python3.8/site-packages (from jedi>
=0.10->ipython>=4.0.0->ipywidgets>=7->arcgis) (0.7.1)
Collecting keyring>=19
  Using cached keyring-21.8.0-py3-none-any.whl (32 kB)
Collecting jeepney>=0.4.2
  Using cached jeepney-0.6.0-py3-none-any.whl (45 kB)
Requirement already satisfied: ipython-genutils in /home/joseph/.local/lib/python3.8/site-packages (from nbformat
>=4.2.0->ipywidgets>=7->arcgis) (0.2.0)
Requirement already satisfied: jupyter-core in /home/joseph/.local/lib/python3.8/site-packages (from nbformat>=4.
2.0->ipywidgets>=7->arcgis) (4.6.3)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /usr/lib/python3/dist-packages (from nbformat>=4.2.0->i
pywidgets>=7->arcgis) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7.3 in /home/joseph/.local/lib/python3.8/site-packages (from pa
ndas>=1->arcgis) (2.8.1)
Requirement already satisfied: pytz>=2017.3 in /home/joseph/.local/lib/python3.8/site-packages (from pandas>=1->a
rcgis) (2020.4)
Requirement already satisfied: wcwidth in /home/joseph/.local/lib/python3.8/site-packages (from prompt-toolkit!=3
.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=4.0.0->ipywidgets>=7->arcgis) (0.2.5)
Collecting pyshp>=2
  Using cached pyshp-2.1.3-py3-none-any.whl
Collecting SecretStorage>=3.2
  Using cached SecretStorage-3.3.0-py3-none-any.whl (14 kB)
Requirement already satisfied: cryptography>=2.0 in /usr/lib/python3/dist-packages (from SecretStorage>=3.2->keyr
ing>=19->arcgis) (2.8)
Collecting ujson>=3
  Using cached ujson-4.0.1-cp38-cp38-manylinux1_x86_64.whl (181 kB)
Requirement already satisfied: notebook>=4.4.1 in /home/joseph/.local/lib/python3.8/site-packages (from widgetsnb
extension>=3->arcgis) (6.1.4)
Requirement already satisfied: argon2-cffi in /home/joseph/.local/lib/python3.8/site-packages (from notebook>=4.4
.1->widgetsnbextension>=3->arcgis) (20.1.0)
Requirement already satisfied: terminado>=0.8.3 in /home/joseph/.local/lib/python3.8/site-packages (from notebook
>=4.4.1->widgetsnbextension>=3->arcgis) (0.9.1)
Requirement already satisfied: pyzmq>=17 in /home/joseph/.local/lib/python3.8/site-packages (from notebook>=4.4.1
->widgetsnbextension>=3->arcgis) (19.0.2)
Requirement already satisfied: jinja2 in /usr/lib/python3/dist-packages (from notebook>=4.4.1->widgetsnbextension
>=3->arcgis) (2.10.1)
Requirement already satisfied: Send2Trash in /home/joseph/.local/lib/python3.8/site-packages (from notebook>=4.4.
1->widgetsnbextension>=3->arcgis) (1.5.0)
Requirement already satisfied: nbconvert in /home/joseph/.local/lib/python3.8/site-packages (from notebook>=4.4.1
->widgetsnbextension>=3->arcgis) (6.0.7)
Requirement already satisfied: prometheus-client in /home/joseph/.local/lib/python3.8/site-packages (from noteboo
k>=4.4.1->widgetsnbextension>=3->arcgis) (0.8.0)
Requirement already satisfied: ptyprocess in /home/joseph/.local/lib/python3.8/site-packages (from terminado>=0.8
.3->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.6.0)
Requirement already satisfied: cffi>=1.0.0 in /home/joseph/.local/lib/python3.8/site-packages (from argon2-cffi->
notebook>=4.4.1->widgetsnbextension>=3->arcgis) (1.14.3)
Requirement already satisfied: pycparser in /home/joseph/.local/lib/python3.8/site-packages (from cffi>=1.0.0->ar
gon2-cffi->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (2.20)
Collecting jupyterlab
  Using cached jupyterlab-3.0.5-py3-none-any.whl (8.3 MB)
Requirement already satisfied: packaging in /home/joseph/.local/lib/python3.8/site-packages (from jupyterlab->arc
gis) (20.4)
Collecting jupyter-server~=1.2
  Using cached jupyter_server-1.2.2-py3-none-any.whl (184 kB)
Collecting anyio>=2.0.2
  Using cached anyio-2.0.2-py3-none-any.whl (62 kB)
Requirement already satisfied: idna>=2.8 in /usr/lib/python3/dist-packages (from anyio>=2.0.2->jupyter-server~=1.
2->jupyterlab->arcgis) (2.8)
Collecting jupyterlab-server~=2.0
  Using cached jupyterlab_server-2.1.2-py3-none-any.whl (47 kB)
Collecting nbclassic~=0.2
  Using cached nbclassic-0.2.6-py3-none-any.whl (18 kB)
Collecting sniffio>=1.1
  Using cached sniffio-1.2.0-py3-none-any.whl (10 kB)
Collecting tornado>=4.2
  Using cached tornado-6.1-cp38-cp38-manylinux2010_x86_64.whl (427 kB)
Collecting babel
  Using cached Babel-2.9.0-py2.py3-none-any.whl (8.8 MB)
Collecting json5
  Using cached json5-0.9.5-py2.py3-none-any.whl (17 kB)
Collecting lerc
  Using cached lerc-0.1.0-py3-none-any.whl
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in /home/joseph/.local/lib/python3.8/site
-packages (from matplotlib->arcgis) (2.4.7)
Requirement already satisfied: certifi>=2020.06.20 in /home/joseph/.local/lib/python3.8/site-packages (from matpl
otlib->arcgis) (2020.6.20)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/joseph/.local/lib/python3.8/site-packages (from matplot
lib->arcgis) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /home/joseph/.local/lib/python3.8/site-packages (from matplotlib->
arcgis) (0.10.0)
Requirement already satisfied: pillow>=6.2.0 in /home/joseph/.local/lib/python3.8/site-packages (from matplotlib-
>arcgis) (8.0.1)
Requirement already satisfied: defusedxml in /home/joseph/.local/lib/python3.8/site-packages (from nbconvert->not
ebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.6.0)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/lib/python3/dist-packages (from nbconvert->notebook>=4.
4.1->widgetsnbextension>=3->arcgis) (0.3)
Requirement already satisfied: mistune<2,>=0.8.1 in /home/joseph/.local/lib/python3.8/site-packages (from nbconve
rt->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.8.4)
Requirement already satisfied: pandocfilters>=1.4.1 in /home/joseph/.local/lib/python3.8/site-packages (from nbco
nvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (1.4.2)
Requirement already satisfied: jupyterlab-pygments in /home/joseph/.local/lib/python3.8/site-packages (from nbcon
vert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.1.2)
Requirement already satisfied: bleach in /home/joseph/.local/lib/python3.8/site-packages (from nbconvert->noteboo
k>=4.4.1->widgetsnbextension>=3->arcgis) (3.2.1)
Requirement already satisfied: testpath in /home/joseph/.local/lib/python3.8/site-packages (from nbconvert->noteb
ook>=4.4.1->widgetsnbextension>=3->arcgis) (0.4.4)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in /home/joseph/.local/lib/python3.8/site-packages (from nb
convert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.5.0)
Requirement already satisfied: async-generator in /home/joseph/.local/lib/python3.8/site-packages (from nbclient<
0.6.0,>=0.5.0->nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (1.10)
Requirement already satisfied: nest-asyncio in /home/joseph/.local/lib/python3.8/site-packages (from nbclient<0.6
.0,>=0.5.0->nbconvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (1.4.1)
Requirement already satisfied: webencodings in /home/joseph/.local/lib/python3.8/site-packages (from bleach->nbco
nvert->notebook>=4.4.1->widgetsnbextension>=3->arcgis) (0.5.1)
Collecting python-certifi-win32
  Using cached python_certifi_win32-1.6-py2.py3-none-any.whl (7.2 kB)
Requirement already satisfied: wrapt>=1.10.4 in /home/joseph/.local/lib/python3.8/site-packages (from python-cert
ifi-win32->arcgis) (1.12.1)
Collecting requests_ntlm
  Using cached requests_ntlm-1.1.0-py2.py3-none-any.whl (5.7 kB)
Collecting ntlm-auth>=1.0.2
  Using cached ntlm_auth-1.5.0-py2.py3-none-any.whl (29 kB)
```

```
Collecting requests-oauthlib
  Using cached requests_oauthlib-1.3.0-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/lib/python3/dist-packages (from requests-oauthlib->arcgis)
(3.1.0)
Collecting requests_toolbelt
  Using cached requests_toolbelt-0.9.1-py2.py3-none-any.whl (54 kB)
Collecting setuptools-scm
  Using cached setuptools_scm-5.0.1-py2.py3-none-any.whl (28 kB)
Building wheels for collected packages: arcgis
  Building wheel for arcgis (setup.py) ... done
  Created wheel for arcgis: filename=arcgis-1.8.3.post2-py2.py3-none-any.whl size=3484077 sha256=cb97e1bd9aee5938
6906616ab0fe1e7fdcbc2749d9c075e5e1c0f34c6faed833
  Stored in directory: /home/joseph/.cache/pip/wheels/05/58/c2/0c313e9066ce8780c462b195d4c2408fa487c6c106a34b4508
Successfully built arcgis
Installing collected packages: tornado, sniffio, anyio, jupyter-server, json5, jeepney, babel, setuptools-scm, Se
cretStorage, ntlm-auth, nbclassic, jupyterlab-server, ujson, requests-toolbelt, requests-oauthlib, requests-ntlm,
python-certifi-win32, pyshp, lerc, keyring, jupyterlab, arcgis
  Attempting uninstall: tornado
    Found existing installation: tornado 6.0.4
    Uninstalling tornado-6.0.4:
      Successfully uninstalled tornado-6.0.4
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This b
ehaviour is the source of the following dependency conflicts.
launchpadlib 1.10.13 requires testresources, which is not installed.
Successfully installed SecretStorage-3.3.0 anyio-2.0.2 arcgis-1.8.3.post2 babel-2.9.0 jeepney-0.6.0 json5-0.9.5 j
upyter-server-1.2.2 jupyterlab-3.0.5 jupyterlab-server-2.1.2 keyring-21.8.0 lerc-0.1.0 nbclassic-0.2.6 ntlm-auth-
1.5.0 pyshp-2.1.3 python-certifi-win32-1.6 requests-ntlm-1.1.0 requests-oauthlib-1.3.0 requests-toolbelt-0.9.1 se
tuptools-scm-5.0.1 sniffio-1.2.0 tornado-6.1 ujson-4.0.1
```

In [14]:

```
from arcgis.geocoding import geocode
from arcgis.gis import GIS
gis = GIS()
```

In [15]:

```
def get_x_y_london(address):
    lat_coords = 0
    long_coords = 0
    g = geocode(address='{}, London, England, GBR'.format(address))[0]
    long_coords = g['location']['x']
    lat_coords = g['location']['y']
    return str(lat_coords) +","+ str(long_coords)
```

Check the shapes and types of the two dataframes

In [16]:

```
post_code_london = df['Post-code']
post_code_london
```

Out[16]:

```
0             SE2
1          W3, W4
2             CR0
3             CR0
4        DA5, DA14
          ...
526          SE18
527           KT4
528           W12
529           UB4
530           UB7
Name: Post-code, Length: 531, dtype: object
```

In [17]:

```
coordinates_london = post_code_london.apply(lambda x: get_x_y_london(x))
coordinates_london
```

Out[17]:

```
0       51.492450000000076,0.12127000000003818
1        51.51324000000005,-0.2674599999999714
2       51.38475500000004,-0.05149847299992416
3       51.38475500000004,-0.05149847299992416
4        51.50642000000005,-0.1272099999999341
                         ...
526      51.48207000000008,0.07143000000002075
527      51.50642000000005,-0.1272099999999341
528      51.50645000000003,-0.2369099999999662
529      51.50642000000005,-0.1272099999999341
530      51.50642000000005,-0.1272099999999341
Name: Post-code, Length: 531, dtype: object
```

In [18]:

```
lat_uk = coordinates_london.apply(lambda x: x.split(',')[0])
```

```
long_uk = coordinates_london.apply(lambda x: x.split(',')[1])
```

Concat the two dataframes (wikipedia dataframe and arcgis dataframe)

In [19]:

```
london_merged = pd.concat([df,lat_uk.astype(float), long_uk.astype(float)], axis=1)
london_merged.columns= ['Borough','Town','Post-code','Latitude','Longitude']
london_merged
```

Out[19]:

	Borough	Town	Post-code	Latitude	Longitude
0	Bexley, Greenwich	LONDON	SE2	51.492450	0.121270
1	Ealing, Hammersmith and Fulham	LONDON	W3, W4	51.513240	-0.267460
2	Croydon	CROYDON	CR0	51.384755	-0.051498
3	Croydon	CROYDON	CR0	51.384755	-0.051498
4	Bexley	BEXLEY, SIDCUP	DA5, DA14	51.506420	-0.127210
...	...	...	...	...	...
526	Greenwich	LONDON	SE18	51.482070	0.071430
527	Sutton, Kingston upon Thames	WORCESTER PARK	KT4	51.506420	-0.127210
528	Hammersmith and Fulham	LONDON	W12	51.506450	-0.236910
529	Hillingdon	HAYES	UB4	51.506420	-0.127210
530	Hillingdon	WEST DRAYTON	UB7	51.506420	-0.127210

531 rows × 5 columns

In [20]:

```
london_merged.dtypes
```

Out[20]:

```
Borough       object
Town          object
Post-code     object
Latitude     float64
Longitude    float64
dtype: object
```

In [21]:

```
london_merged.shape
```

Out[21]:

```
(531, 5)
```

Obtaining the coords of London city

In [22]:

```
london = geocode(address = 'London, England, GBR')[0]
london
```

Out[22]:

```
{'address': 'London, England',
 'location': {'x': -0.12720999999999341, 'y': 51.50642000000005},
 'score': 100,
 'attributes': {'Loc_name': 'World',
  'Status': 'T',
  'Score': 100,
  'Match_addr': 'London, England',
  'LongLabel': 'London, England, GBR',
  'ShortLabel': 'London',
  'Addr_type': 'Locality',
  'Type': 'City',
  'PlaceName': 'London',
  'Place_addr': 'London, England',
  'Phone': '',
  'URL': '',
  'Rank': 1.75,
  'AddBldg': '',
  'AddNum': '',
  'AddNumFrom': '',
  'AddNumTo': '',
  'AddRange': '',
  'Side': '',
  'StPreDir': '',
  'StPreType': '',
  'StName': '',
  'StType': '',
  'StDir': '',
  'BldgType': '',
  'BldgName': '',
  'LevelType': '',
  'LevelName': '',
  'UnitType': '',
  'UnitName': '',
  'SubAddr': '',
  'StAddr': '',
  'Block': '',
  'Sector': '',
  'Nbrhd': '',
  'District': '',
  'City': 'London',
  'MetroArea': '',
  'Subregion': 'London',
  'Region': 'England',
  'RegionAbbr': 'ENG',
  'Territory': '',
  'Zone': '',
  'Postal': '',
  'PostalExt': '',
  'Country': 'GBR',
  'LangCode': 'ENG',
  'Distance': 0,
  'X': -0.1272099999999341,
  'Y': 51.50642000000005,
  'DisplayX': -0.1272099999999341,
  'DisplayY': 51.50642000000005,
  'Xmin': -0.4402099999999341,
  'Xmax': 0.1857900000000659,
  'Ymin': 51.193420000000046,
  'Ymax': 51.81942000000005,
  'ExInfo': ''},
 'extent': {'xmin': -0.4402099999999341,
  'ymin': 51.193420000000046,
  'xmax': 0.1857900000000659,
  'ymax': 51.81942000000005}}
```

In [23]:

```
london_long_coords = london['location']['x']
london_lat_coords = london['location']['y']
print('The coordinates of London are {}, {}.'.format(london_lat_coords, london_long_coords))
```

```
The coordinates of London are 51.50642000000005, -0.1272099999999341.
```

Create and visualize London's map with Folium

In [24]:

```
import folium
```

In [25]:

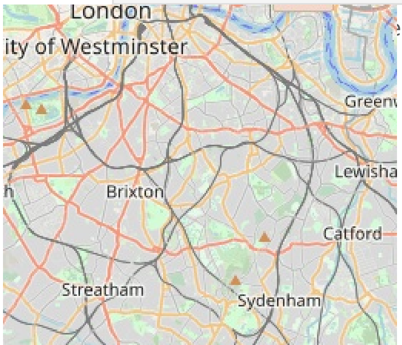
```
map_London = folium.Map(location=[london_lat_coords, london_long_coords], zoom_start=11)
```

Adding markers neighbourhoods to map

In [26]:

```
for latitude, longitude, borough, town in zip(london_merged['Latitude'], london_merged['Longitude'], london_merge
    label = '{}, {}'.format(town, borough)
    label = folium.Popup(label, parse_html = True)
    folium.CircleMarker(
        [latitude, longitude],
        radius = 4,
        popup = label,
        color = 'black',
        fill = True
        ).add_to(map_London)

map_London
```



ed to load map: File -> Trust Notebook

Foursquare API credentials

In [27]:

```
CLIENT_ID = 'NRGQEDUFGDGDUVYO3MEW3QY1E1P51QHT3QBP0MTNJL12ZZ0F'
CLIENT_SECRET = '02GVG22AKZW33XAPG3ZUPVX0BZNCH0DKZJUCPMFH54BC3RTB'
VERSION = '20180605'
```

Getting all venues categories in London

In [28]:

```
Limit=100

def getNearbyVenues(names, latitudes, longitudes, radius=500):

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

        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            long,
            radius,
            Limit
            )

        results = requests.get(url).json()["response"]['groups'][0]['items']

        venues_list.append([(
            name,
            lat,
            long,
            v['venue']['name'],
            v['venue']['categories'][0]['name']) for v in results])

    nearby_venues = pd.DataFrame([element for venue in venues_list for element in venue])
    nearby_venues.columns = ['Neighbourhood',
                  'Neighbourhood Latitude',
                  'Neighbourhood Longitude',
                  'Venue',
                  'Venue Category']

    return(nearby_venues)
```

Venues in London for each Neighbourhood with function created

In [29]:

```
venues_in_london = getNearbyVenues(london_merged['Borough'], london_merged['Latitude'], london_merged['Longitude'
```

```
Bexley, Greenwich
Ealing, Hammersmith and Fulham
Croydon
Croydon
Bexley
Redbridge
City
Westminster
Brent
Bromley
Islington
Bromley
```

```
Islington
Havering
Barnet
Enfield
Wandsworth
Southwark
City
Barking and Dagenham
Redbridge
Bexley
Richmond upon Thames
Bexley
Barnet
Barnet
Islington
Wandsworth
Westminster
Bromley
Newham
Barking and Dagenham
Barking and Dagenham
Sutton
Ealing
Westminster
Lewisham
Harrow
Sutton
Camden
Bexley
Southwark
Kingston upon Thames
Tower Hamlets
Bexley
Bexley
Bromley
Bromley
Bexley
City
Lewisham
Greenwich
Tower Hamlets
Bexley
Camden
Enfield
Haringey
Tower Hamlets
Haringey
Hounslow
Barnet
Brent
Enfield
Lambeth
Lewisham
Bromley
Tower Hamlets
Bromley
Kensington and Chelsea, Hammersmith and Fulham
Brent
Barnet
Enfield
Barnet
Barnet
Southwark
Tower Hamlets
Camden
Tower Hamlets
Waltham Forest
Newham
Islington
Sutton
Richmond upon Thames
Barking and Dagenham
Lewisham
Redbridge, Barking and Dagenham
Camden
Westminster
Greenwich
Havering
Sutton
Kensington and Chelsea
Bromley
Kingston upon Thames
Barnet
```

Westminster
Lewisham
Waltham Forest
Bromley
Hounslow, Ealing, Hammersmith and Fulham
Brent
Barnet
Lambeth, Wandsworth
Islington
Barnet, Enfield
Barnet
Havering
Merton
Barnet
Bexley
Bromley
Croydon
Kingston upon Thames
Croydon
Westminster
Hillingdon
Hounslow
Havering
Bexley
Barking and Dagenham
Enfield
Barnet, Brent, Camden
Lewisham
Bexley
Bexley
Haringey
Croydon
Bromley
Tower Hamlets
Bromley
Newham
Barking and Dagenham
Hackney
Islington
Southwark
Lewisham
Bromley
Brent
Bromley
Lewisham
Southwark
Ealing
Kensington and Chelsea
Wandsworth
Barnet
Hounslow
Southwark
Barnet
Newham
Richmond upon Thames
Bexley
Hillingdon
Bromley
Barnet
Enfield
Richmond upon Thames
Southwark
Havering
Bromley
Bromley
Greenwich
Havering
Enfield
Enfield
Enfield
Enfield
Bexley
Bexley, Greenwich
Islington, City
Hounslow
Barnet
Islington
Haringey, Islington
Camden
Bexley
Newham
Lewisham
Croydon

Haringey
Enfield
Barnet
Camden
Hammersmith and Fulham
Richmond upon Thames
Havering
Redbridge
Havering
Lambeth
Bromley
Barnet
Redbridge
Camden
Barnet
Enfield
Ealing
Greenwich
Hounslow
Lewisham
Hounslow
Hackney
Hackney
Hackney
Hackney
Enfield
Hackney
Redbridge
Barnet
Richmond upon Thames
Hammersmith and Fulham
Camden
Barnet
Richmond upon Thames
Richmond upon Thames
Richmond upon Thames
Ealing
Hounslow
Hillingdon
Brent
Hillingdon
Hillingdon
Havering
Havering
Havering
Haringey
Harrow
Harrow
Harrow
Harrow
Hounslow
Havering
Bromley
Hillingdon
Bromley
Barnet
Lambeth
Hounslow
Waltham Forest
Islington
Camden
Hillingdon
Lewisham
Camden
Kensington and Chelsea
Islington
Hackney
Lewisham
Kingston upon Thames
Havering
Greenwich, Lewisham
Haringey
Hounslow
Hackney
Barnet
Hillingdon
Redbridge
Tower Hamlets
Hounslow
Islington
Croydon
Lambeth, Southwark
Brent

```
Kensington and Chelsea
Camden
Brent, Harrow
Bromley
Richmond upon Thames
Greenwich
Brent, Camden
Camden, Islington
Brent
Kingston upon Thames
Kingston upon Thames
Westminster
Lewisham
Lambeth
Bexley
Hounslow
Hackney
Tower Hamlets
Bromley
Lewisham
Bexley
Lewisham
Waltham Forest
Waltham Forest
Tower Hamlets
Westminster
Newham
Westminster
Bromley
Hillingdon
Bexley
Hackney
Merton
Redbridge
Westminster
Kingston upon Thames
Hackney
Newham
Barking and Dagenham
Newham
Westminster
Westminster
Greenwich
Merton
Greenwich
Tower Hamlets
Barnet
Westminster
Tower Hamlets
Merton
Barnet
Merton
Merton
Richmond upon Thames
Kingston upon Thames
Bromley
Haringey, Barnet
Brent
Croydon
Barnet
Lewisham
Greenwich
Kingston upon Thames
Barnet
Redbridge
Southwark
Wandsworth
Havering
Kingston upon Thames
Croydon
Bexley
Bexley
Barnet
Harrow
Kensington and Chelsea
Havering
Richmond upon Thames
Newham
Ealing
Bexley
Hillingdon
Ealing
Kensington and Chelsea
```

Southwark
Barnet
Croydon
Tower Hamlets
Kingston upon Thames
Hammersmith and Fulham
Bromley
Barnet
Hounslow
Lambeth
Westminster
Enfield
Brent, Ealing
Hammersmith and Fulham
Southwark
Bromley
Islington
Ealing
Richmond upon Thames
Bromley
Westminster
Harrow
Newham
Bromley
Greenwich
Enfield
Tower Hamlets
Bromley
Brent
Camden
Croydon
Wandsworth
Brent
Harrow, Brent
Havering
Tower Hamlets
Harrow
Merton
Redbridge
Richmond upon Thames
Croydon
Wandsworth
Havering
Southwark
Hillingdon
Barking and Dagenham
Bexley, Bromley
Croydon
Hammersmith and Fulham
Croydon
Croydon
Redbridge
Haringey
Hackney
Tower Hamlets
Hammersmith and Fulham
Croydon
Greenwich
Hackney
Bexley
Newham
Hillingdon
Bexley
Redbridge, Waltham Forest
Westminster
Camden
Croydon
Hackney
Harrow
Havering
Kensington and Chelsea
Croydon
Hillingdon
Merton
Redbridge
Haringey
Lewisham
Ealing
Bromley
Wandsworth
Enfield
Tower Hamlets
Merton

Westminster
Richmond upon Thames
Camden
Lewisham
Westminster
Islington
Bromley
Camden
Bromley
Hackney
Harrow
Tower Hamlets
Lambeth
Hackney
Brent
Newham
Richmond upon Thames
Lambeth
Haringey
Brent, Ealing, Harrow
Bromley
Kingston upon Thames
Southwark
Sutton
Camden
Lewisham, Bromley
Lewisham, Southwark
Richmond upon Thames
City, Westminster
Barnet
Bexley, Greenwich
Croydon
Brent
Kingston upon Thames
Wandsworth
Wandsworth
Haringey
Haringey
Haringey
Barnet
Tower Hamlets
Islington, Camden
Lambeth
Haringey
Richmond upon Thames
Havering
Havering
Hackney
Islington
Croydon
Bexley, Bromley
Waltham Forest
Bexley
Newham
Hillingdon
Lambeth
Croydon
Sutton
Waltham Forest
Waltham Forest
Southwark
Wandsworth
Redbridge
Tower Hamlets
Harrow
Greenwich
Bexley
Brent
Brent
Havering
Kensington and Chelsea
Hillingdon
Ealing
Haringey
Hackney
Newham
Camden
Harrow
Bexley
Barnet
Hammersmith and Fulham
Lambeth
Bromley

```
Greenwich
Westminster
Barnet
Hammersmith and Fulham
Tower Hamlets
Bromley
Richmond upon Thames
Brent
Merton
Enfield
Haringey
Redbridge
Redbridge, Waltham Forest
Hounslow
Croydon
Barnet
Greenwich
Sutton, Kingston upon Thames
Hammersmith and Fulham
Hillingdon
Hillingdon
```

In [30]:

```
venues_in_london.shape
```

Out[30]:

```
(21560, 5)
```

In [31]:

```
venues_in_london.head()
```

Out[31]:

	Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
0	Bexley, Greenwich	51.49245	0.12127	Lesnes Abbey	Historic Site
1	Bexley, Greenwich	51.49245	0.12127	Sainsbury's	Supermarket
2	Bexley, Greenwich	51.49245	0.12127	Lidl	Supermarket
3	Bexley, Greenwich	51.49245	0.12127	Abbey Wood Railway Station (ABW)	Train Station
4	Bexley, Greenwich	51.49245	0.12127	Bean @ Work	Coffee Shop

In [35]:

```
b = venues_in_london.groupby('Neighbourhood').count()
b.sort_values('Venue')['Venue']
```

Out[35]:

```
Neighbourhood
Harrow, Brent                                           3
Barnet, Brent, Camden                                   5
Bexley, Greenwich                                       5
Ealing, Hammersmith and Fulham                          7
Bexley, Greenwich                                       8
Brent, Ealing                                           8
Lewisham, Southwark                                     8
Greenwich, Lewisham                                     9
Lewisham, Bromley                                      13
Haringey, Islington                                    26
Islington, Camden                                      31
Brent, Camden                                          34
Hounslow, Ealing, Hammersmith and Fulham               40
Lambeth, Southwark                                     41
Camden, Islington                                      41
Haringey, Barnet                                       45
Redbridge, Barking and Dagenham                        66
Sutton, Kingston upon Thames                           66
Brent, Harrow                                          66
Barnet, Enfield                                        66
Brent, Ealing, Harrow                                  66
Lambeth, Wandsworth                                    73
Islington, City                                        79
Redbridge, Waltham Forest                              94
Kensington and Chelsea, Hammersmith and Fulham         96
City, Westminster                                     100
Bexley, Bromley                                       132
Greenwich                                             151
Enfield                                               224
City                                                  235
Waltham Forest                                        257
Lambeth                                               291
Merton                                                304
Haringey                                              317
```

```
Sutton                                             331
Newham                                             356
Wandsworth                                         370
Lewisham                                           400
Barking and Dagenham                               410
Hammersmith and Fulham                             410
Ealing                                             417
Tower Hamlets                                      469
Brent                                              472
Southwark                                          525
Kensington and Chelsea                             567
Redbridge                                          586
Hackney                                            600
Croydon                                            620
Barnet                                             642
Islington                                          694
Kingston upon Thames                               706
Harrow                                             792
Camden                                             804
Hounslow                                           817
Richmond upon Thames                               885
Hillingdon                                        1122
Bexley                                            1224
Bromley                                           1378
Havering                                          1386
Westminster                                       1570
Name: Venue, dtype: int64
```

In [40]:

```
venues_category = venues_in_london['Venue Category'].unique()
venues_category
```

Out[40]:

```
array(['Historic Site', 'Supermarket', 'Train Station', 'Coffee Shop',
       'Grocery Store', 'Bed & Breakfast', 'Indian Restaurant', 'Park',
       'Breakfast Spot', 'Hotel', 'Plaza', 'Fountain',
       'Outdoor Sculpture', 'Monument / Landmark', 'Art Museum', 'Church',
       'Garden', 'Art Gallery', 'Wine Bar', 'Indie Movie Theater',
       'Thai Restaurant', 'Burger Joint', 'Theater', 'Opera House',
       'Gay Bar', 'Ice Cream Shop', 'Boutique', 'Cocktail Bar',
       'Pharmacy', 'Japanese Restaurant', 'Bar', 'Liquor Store', 'Spa',
       'Sandwich Place', 'Mexican Restaurant', 'Tea Room', 'Bakery',
       'Tour Provider', 'Greek Restaurant', 'Pakistani Restaurant',
       'English Restaurant', 'Steakhouse', 'French Restaurant', 'Lounge',
       'Restaurant', 'Ramen Restaurant', 'Lebanese Restaurant', 'Café',
       'Italian Restaurant', 'Pub', 'Sculpture Garden', 'Scenic Lookout',
       'Salad Place', 'Market', 'Mini Golf', 'Beer Bar',
       'Portuguese Restaurant', 'South American Restaurant',
       'Turkish Restaurant', 'Tapas Restaurant', 'Asian Restaurant',
       'Office', 'Falafel Restaurant', 'Gym / Fitness Center',
       'Street Food Gathering', 'General Entertainment', 'Speakeasy',
       "Men's Store", 'Hotel Bar', 'Argentinian Restaurant', 'Bookstore',
       'Souvenir Shop', 'Sushi Restaurant', 'Food Truck', 'Dessert Shop',
       'Champagne Bar', 'Seafood Restaurant', 'Middle Eastern Restaurant',
       'Event Space', 'Athletics & Sports', 'Other Nightlife',
       'Performing Arts Venue', 'Gastropub', 'History Museum',
       'Comedy Club', 'Motorcycle Shop', 'Gym', 'Clothing Store',
       'Okonomiyaki Restaurant', 'Deli / Bodega', 'Dance Studio',
       'Gelato Shop', 'Gift Shop', 'Shoe Store', 'Arcade',
       'Korean Restaurant', 'Brasserie', 'Accessories Store', 'Pool',
       'Burrito Place', 'Antique Shop', 'Fish & Chips Shop',
       'Fast Food Restaurant', 'Chinese Restaurant', 'Optical Shop',
       'Convenience Store', 'Bike Shop', 'Vietnamese Restaurant',
       'Bagel Shop', 'Taco Place', 'Soup Place', 'Moroccan Restaurant',
       'Museum', 'Concert Hall', 'Modern European Restaurant', 'Gym Pool',
       'Food Stand', 'Pizza Place', 'BBQ Joint', 'Yoga Studio',
       'Kebab Restaurant', 'Farmers Market',
       'Vegetarian / Vegan Restaurant', 'Juice Bar', 'Stationery Store',
       'Salon / Barbershop', 'Pet Store', 'Bus Stop', 'Platform',
       'Spanish Restaurant', 'Beer Store', 'Wine Shop',
       'Caucasian Restaurant', 'Shop & Service', 'Brewery',
       'American Restaurant', 'Residential Building (Apartment / Condo)',
       'Nightclub', 'Music Venue', 'Hostel', 'Cemetery',
       'Department Store', 'Boxing Gym', 'German Restaurant',
       'Whisky Bar', 'Donut Shop', 'Botanical Garden', 'Poke Place',
       'Lake', 'Forest', 'Food & Drink Shop', 'Movie Theater',
       'Nature Preserve', 'Track', 'Sports Club', 'Tennis Court',
       'Recruiting Agency', 'Scandinavian Restaurant', 'Circus School',
       'Mediterranean Restaurant', 'School', 'Sports Bar', 'Bistro',
       'Adult Boutique', 'Costume Shop', 'Filipino Restaurant',
       'Malay Restaurant', 'Canal', 'Chocolate Shop', 'Gourmet Shop',
       'Stables', 'Mobile Phone Shop', 'Dim Sum Restaurant',
       'Tourist Information Center', 'Food Court', 'Science Museum',
       'Pier', 'Diner', 'Beach', 'Playground', 'Housing Development',
       'Organic Grocery', 'Furniture / Home Store', 'Creperie', 'Trail',
       'Sporting Goods Shop', 'Pedestrian Plaza', 'Arts & Crafts Store',
       'Camera Store', 'Cheese Shop', 'Discount Store', 'Shopping Plaza',
       'Bus Station', 'Vape Store', 'Social Club', 'Arepa Restaurant',
       'Thrift / Vintage Store', 'Music Store', 'Distillery',
       'Noodle House', 'Building', 'Library', 'Fried Chicken Joint',
       'Hardware Store', 'Polish Restaurant', 'Light Rail Station',
       'Shopping Mall', 'Szechuan Restaurant', 'Rental Car Location',
       'Locksmith', 'Student Center', 'Irish Pub', 'Warehouse Store',
       'Xinjiang Restaurant', 'Ethiopian Restaurant', 'Flower Shop',
       "Women's Store", 'Kids Store', 'Fish Market', 'Candy Store',
       'Health Food Store', 'Cosmetics Shop', 'Brazilian Restaurant',
       'Doner Restaurant', 'Wings Joint', 'Construction & Landscaping',
       'Zoo Exhibit', 'Metro Station', 'Fruit & Vegetable Store',
       'Snack Place', 'Auto Garage', 'Toy / Game Store', 'Nail Salon',
       'Cycle Studio', 'Electronics Store', 'Lingerie Store',
       'Jewelry Store', 'Cupcake Shop', 'Hookah Bar', 'Skate Park',
       'Auto Workshop', 'Multiplex', 'Kitchen Supply Store',
       'Video Game Store', 'Tram Station', 'Caribbean Restaurant',
       'Record Shop', 'Health & Beauty Service',
       'Paper / Office Supplies Store', 'Butcher', 'River',
       'Arts & Entertainment', 'Persian Restaurant',
       'Grilled Meat Restaurant', 'Peruvian Restaurant', 'Garden Center',
       'Climbing Gym', 'Film Studio', 'Golf Course', 'Home Service',
       'Reservoir', 'Comfort Food Restaurant', 'Recreation Center',
       'Kosher Restaurant', 'Soccer Field', 'South Indian Restaurant',
       'Australian Restaurant', 'Tree', 'Boarding House',
       'Business Center', 'Massage Studio', 'Latin American Restaurant',
       'Herbs & Spices Store', 'Miscellaneous Shop', 'Shaanxi Restaurant',
       'Soccer Stadium', 'Escape Room', 'Other Repair Shop',
       'Event Service', 'Exhibit', 'College Quad', 'University',
       'Convention Center', 'Pilates Studio', 'Recording Studio',
       'Cricket Ground', 'Gas Station', 'African Restaurant',
       'Dry Cleaner', 'Martial Arts School', 'Indie Theater',
       'Hungarian Restaurant', 'Post Office', 'Pie Shop', 'Tanning Salon',
       'Community Center', 'Beer Garden', 'Sake Bar', 'Gaming Cafe',
       'Child Care Service', 'Eastern European Restaurant', 'Bridal Shop',
       'Daycare', 'Betting Shop', 'Flea Market', 'Food Service',
       'Halal Restaurant', 'Jazz Club'], dtype=object)
```

Grouping data by categories

In [46]:

```
Gb_venues_category = venues_in_london.groupby('Venue Category').count()
Gb_venues_category
```

Out[46]:

	Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue
Venue Category				
Accessories Store	4	4	4	4
Adult Boutique	7	7	7	7
African Restaurant	6	6	6	6
American Restaurant	19	19	19	19
Antique Shop	7	7	7	7
...	...	...	...	...
Wings Joint	5	5	5	5
Women's Store	10	10	10	10
Xinjiang Restaurant	3	3	3	3
Yoga Studio	50	50	50	50
Zoo Exhibit	24	24	24	24

308 rows × 4 columns

In [101…

```
g = Gb_venues_category
g['Venue'].loc['Chinese Restaurant']
```

Out[101… 93

In [51]:

```
num_venues = venues_in_london['Venue Category'].value_counts()
```

```
Gb_venues_category[Gb_venues_category['Neighbourhood'] == 4]
```

Out[51]:

	Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue
Venue Category				
Accessories Store	4	4	4	4
Arcade	4	4	4	4
Brasserie	4	4	4	4
Canal	4	4	4	4
Comedy Club	4	4	4	4
Convention Center	4	4	4	4
Escape Room	4	4	4	4
Gas Station	4	4	4	4
Kitchen Supply Store	4	4	4	4
Library	4	4	4	4
Massage Studio	4	4	4	4
Music Store	4	4	4	4
Nature Preserve	4	4	4	4
Okonomiyaki Restaurant	4	4	4	4
Recruiting Agency	4	4	4	4
Salon / Barbershop	4	4	4	4
Snack Place	4	4	4	4
Soccer Field	4	4	4	4
Tram Station	4	4	4	4

In [52]:

```
Gb_neighbourhood = venues_in_london.groupby('Neighbourhood').head()
Gb_neighbourhood
```

Out[52]:

	Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
0	Bexley, Greenwich	51.49245	0.12127	Lesnes Abbey	Historic Site
1	Bexley, Greenwich	51.49245	0.12127	Sainsbury's	Supermarket
2	Bexley, Greenwich	51.49245	0.12127	Lidl	Supermarket
3	Bexley, Greenwich	51.49245	0.12127	Abbey Wood Railway Station (ABW)	Train Station
4	Bexley, Greenwich	51.49245	0.12127	Bean @ Work	Coffee Shop
...	...	...	...	...	...
21333	Sutton, Kingston upon Thames	51.50642	-0.12721	Corinthia Hotel	Hotel
21334	Sutton, Kingston upon Thames	51.50642	-0.12721	Trafalgar Square	Plaza
21335	Sutton, Kingston upon Thames	51.50642	-0.12721	East Trafalgar Square Fountain	Fountain
21336	Sutton, Kingston upon Thames	51.50642	-0.12721	Horse Guards Parade	Plaza
21337	Sutton, Kingston upon Thames	51.50642	-0.12721	Trafalgar Square Lions	Outdoor Sculpture

298 rows × 5 columns

In [53]:

```
venues_in_london['Neighbourhood'].value_counts()
```

Out[53]:

```
Westminster                                         1570
Havering                                            1386
Bromley                                             1378
Bexley                                              1224
Hillingdon                                          1122
Richmond upon Thames                                 885
Hounslow                                             817
Camden                                               804
Harrow                                               792
Kingston upon Thames                                 706
```

```
Islington                                        694
Barnet                                           642
Croydon                                          620
Hackney                                          600
Redbridge                                        586
Kensington and Chelsea                           567
Southwark                                        525
Brent                                            472
Tower Hamlets                                    469
Ealing                                           417
Barking and Dagenham                             410
Hammersmith and Fulham                           410
Lewisham                                         400
Wandsworth                                       370
Newham                                           356
Sutton                                           331
Haringey                                         317
Merton                                           304
Lambeth                                          291
Waltham Forest                                   257
City                                             235
Enfield                                          224
Greenwich                                        151
Bexley, Bromley                                  132
City, Westminster                                100
Kensington and Chelsea, Hammersmith and Fulham    96
Redbridge, Waltham Forest                         94
Islington, City                                   79
Lambeth, Wandsworth                               73
Brent, Harrow                                     66
Brent, Ealing, Harrow                             66
Barnet, Enfield                                   66
Redbridge, Barking and Dagenham                   66
Sutton, Kingston upon Thames                      66
Haringey, Barnet                                  45
Lambeth, Southwark                                41
Camden, Islington                                 41
Hounslow, Ealing, Hammersmith and Fulham          40
Brent, Camden                                     34
Islington, Camden                                 31
Haringey, Islington                               26
Lewisham, Bromley                                 13
Greenwich, Lewisham                                9
Brent, Ealing                                      8
Bexley, Greenwich                                  8
Lewisham, Southwark                                8
Ealing, Hammersmith and Fulham                     7
Barnet, Brent, Camden                              5
Bexley, Greenwich                                  5
Harrow, Brent                                      3
Name: Neighbourhood, dtype: int64
```

One Hot encoding

In [54]:

```python
london_venue_category = pd.get_dummies(venues_in_london[['Venue Category']], prefix = "", prefix_sep = "")
london_venue_category
```

Out[54]:

	Accessories Store	Adult Boutique	African Restaurant	American Restaurant	Antique Shop	Arcade	Arepa Restaurant	Argentinian Restaurant	Art Gallery	Art Museum	...	Vietn Rest
0	0	0	0	0	0	0	0	0	0	0	...	
1	0	0	0	0	0	0	0	0	0	0	...	
2	0	0	0	0	0	0	0	0	0	0	...	
3	0	0	0	0	0	0	0	0	0	0	...	
4	0	0	0	0	0	0	0	0	0	0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
21555	0	0	0	0	0	0	0	0	0	0	...	
21556	0	0	0	0	0	0	0	0	0	0	...	
21557	0	0	0	0	0	0	0	0	0	0	...	
21558	0	0	0	0	0	0	0	0	0	0	...	
21559	0	0	0	0	0	0	0	0	0	0	...	

21560 rows × 308 columns

Adding the neighbourhood to last dataframe and group by neihgbourhood

In [55]:

```
london_venue_category['Neighbourhood'] = venues_in_london['Neighbourhood']

fixed_columns = [london_venue_category.columns[-1]] + list(london_venue_category.columns[:-1])
london_venue_category = london_venue_category[fixed_columns]

london_venue_category.head()
```

Out[55]:

	Neighbourhood	Accessories Store	Adult Boutique	African Restaurant	American Restaurant	Antique Shop	Arcade	Arepa Restaurant	Argentinian Restaurant	Art Gallery	...	Vie Re
0	Bexley, Greenwich	0	0	0	0	0	0	0	0	0	...	
1	Bexley, Greenwich	0	0	0	0	0	0	0	0	0	...	
2	Bexley, Greenwich	0	0	0	0	0	0	0	0	0	...	
3	Bexley, Greenwich	0	0	0	0	0	0	0	0	0	...	
4	Bexley, Greenwich	0	0	0	0	0	0	0	0	0	...	

5 rows × 309 columns

In [56]:

```
Gb_neighbourhood_london = london_venue_category.groupby('Neighbourhood').mean().reset_index().round(8)
Gb_neighbourhood_london.head()
```

Out[56]:

	Neighbourhood	Accessories Store	Adult Boutique	African Restaurant	American Restaurant	Antique Shop	Arcade	Arepa Restaurant	Argentinian Restaurant	Art Gallery	...	Vi R
0	Barking and Dagenham	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.029268	...	
1	Barnet	0.0	0.0	0.0	0.001558	0.0	0.0	0.0	0.006231	0.003115	...	
2	Barnet, Brent, Camden	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	...	
3	Barnet, Enfield	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.030303	...	
4	Bexley	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.029412	...	

5 rows × 309 columns

In [57]:

```
len(Gb_neighbourhood_london[Gb_neighbourhood_london['Chinese Restaurant'] > 0])
```

Out[57]:

```
18
```

In [113…

```
london_chinese_resta = Gb_neighbourhood_london[['Neighbourhood','Chinese Restaurant']]
london_chinese_resta
```

Out[113…

	Neighbourhood	Chinese Restaurant
0	Barking and Dagenham	0.000000
1	Barnet	0.012461
2	Barnet, Brent, Camden	0.000000
3	Barnet, Enfield	0.000000
4	Bexley	0.000000
5	Bexley, Bromley	0.000000
6	Bexley, Greenwich	0.000000
7	Bexley, Greenwich	0.000000
8	Brent	0.014831
9	Brent, Camden	0.000000
10	Brent, Ealing	0.125000
11	Brent, Ealing, Harrow	0.000000
12	Brent, Harrow	0.000000

13	Bromley	0.000000
14	Camden	0.002488
15	Camden, Islington	0.000000
16	City	0.004255
17	City, Westminster	0.000000
18	Croydon	0.000000
19	Ealing	0.000000
20	Ealing, Hammersmith and Fulham	0.000000
21	Enfield	0.000000
22	Greenwich	0.026490
23	Greenwich, Lewisham	0.000000
24	Hackney	0.001667
25	Hammersmith and Fulham	0.012195
26	Haringey	0.000000
27	Haringey, Barnet	0.000000
28	Haringey, Islington	0.000000
29	Harrow	0.000000
30	Harrow, Brent	0.000000
31	Havering	0.000000
32	Hillingdon	0.000000
33	Hounslow	0.000000
34	Hounslow, Ealing, Hammersmith and Fulham	0.000000
35	Islington	0.005764
36	Islington, Camden	0.000000
37	Islington, City	0.012658
38	Kensington and Chelsea	0.003527
39	Kensington and Chelsea, Hammersmith and Fulham	0.000000
40	Kingston upon Thames	0.000000
41	Lambeth	0.006873
42	Lambeth, Southwark	0.000000
43	Lambeth, Wandsworth	0.000000
44	Lewisham	0.010000
45	Lewisham, Bromley	0.000000
46	Lewisham, Southwark	0.000000
47	Merton	0.000000
48	Newham	0.000000
49	Redbridge	0.003413
50	Redbridge, Barking and Dagenham	0.000000
51	Redbridge, Waltham Forest	0.000000
52	Richmond upon Thames	0.000000
53	Southwark	0.007619
54	Sutton	0.000000
55	Sutton, Kingston upon Thames	0.000000
56	Tower Hamlets	0.046908
57	Waltham Forest	0.003891
58	Wandsworth	0.000000
59	Westminster	0.014013

Finding the most common venues

In [59]:

```python
def 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 [60]:

```
import numpy as np
```

Decision of number of common venues due to there are many venues in order to evaluate

In [61]:

```
number_common_venues = 12

ord_number = ['st', 'nd', 'rd']

columns = ['Neighbourhood']
for ordinal in np.arange(number_common_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ordinal+1, ord_number[ordinal]))
    except:
        columns.append('{}th Most Common Venue'.format(ordinal+1))


neighbourhoods_venues_sorted = pd.DataFrame(columns = columns)
neighbourhoods_venues_sorted['Neighbourhood'] = Gb_neighbourhood_london['Neighbourhood']

for i in np.arange(Gb_neighbourhood_london.shape[0]):
    neighbourhoods_venues_sorted.iloc[i, 1:] = most_common_venues(Gb_neighbourhood_london.iloc[i, :], number_comm

neighbourhoods_venues_sorted
```

Out[61]:

	Neighbourhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9 C
0	Barking and Dagenham	Hotel	Plaza	Theater	Garden	Monument / Landmark	Cocktail Bar	Boutique	Ramen Restaurant	Bur
1	Barnet	Coffee Shop	Café	Grocery Store	Pub	Bus Stop	Italian Restaurant	Supermarket	Pharmacy	Re
2	Barnet, Brent, Camden	Hardware Store	Supermarket	Bakery	Clothing Store	Gym / Fitness Center	Office	Noodle House	Okonomiyaki Restaurant	Oper
3	Barnet, Enfield	Hotel	Theater	Garden	Plaza	Monument / Landmark	Burger Joint	Cocktail Bar	Steakhouse	J Re
4	Bexley	Hotel	Garden	Theater	Plaza	Monument / Landmark	Boutique	Cocktail Bar	Sandwich Place	Re
5	Bexley, Bromley	Hotel	Theater	Garden	Plaza	Monument / Landmark	Burger Joint	Cocktail Bar	Steakhouse	J Re
6	Bexley, Greenwich	Daycare	Convenience Store	Golf Course	Park	Massage Studio	Historic Site	Construction & Landscaping	Bus Stop	Re
7	Bexley, Greenwich	Supermarket	Train Station	Coffee Shop	Historic Site	Accessories Store	Other Nightlife	Park	Paper / Office Supplies Store	I Re
8	Brent	Hotel	Theater	Garden	Plaza	Sandwich Place	Monument / Landmark	Pharmacy	Liquor Store	
9	Brent, Camden	Indian Restaurant	Pub	Brazilian Restaurant	Fast Food Restaurant	Supermarket	Doner Restaurant	Grocery Store	Coffee Shop	
10	Brent, Ealing	Convenience Store	Warehouse Store	Grocery Store	Pharmacy	Chinese Restaurant	Liquor Store	Sandwich Place	Fast Food Restaurant	
11	Brent, Ealing, Harrow	Hotel	Theater	Garden	Plaza	Monument / Landmark	Burger Joint	Cocktail Bar	Steakhouse	J Re
12	Brent, Harrow	Hotel	Theater	Garden	Plaza	Monument / Landmark	Burger Joint	Cocktail Bar	Steakhouse	J Re
13	Bromley	Hotel	Garden	Plaza	Theater	Monument / Landmark	Sandwich Place	Cocktail Bar	Burger Joint	I
14	Camden	Pub	Café	Coffee Shop	Bakery	Italian Restaurant	Zoo Exhibit	Hotel	Bookstore	J Re
15	Camden, Islington	Pub	Bookstore	Hotel	Café	Coffee Shop	Garden	Fish & Chips Shop	Gay Bar	Re
16	City	Italian Restaurant	Coffee Shop	Hotel	Gym / Fitness Center	Pub	Restaurant	Wine Bar	Cocktail Bar	
17	City, Westminster	Italian Restaurant	Coffee Shop	Pub	Sandwich Place	Wine Bar	Hotel	Falafel Restaurant	Gym / Fitness Center	Re
						Monument /		Japanese		

18	Croydon	Hotel	Garden	Plaza	Theater	Landmark	Burger Joint	Restaurant	Wine Bar	Re
19	Ealing	Hotel	Plaza	Garden	Theater	Monument / Landmark	Sandwich Place	Pub	Café	Bur
20	Ealing, Hammersmith and Fulham	Grocery Store	Park	Train Station	Indian Restaurant	Breakfast Spot	Bed & Breakfast	Accessories Store	Other Nightlife	
21	Enfield	Pub	Café	Supermarket	Turkish Restaurant	Italian Restaurant	Coffee Shop	Breakfast Spot	Playground	F Re
22	Greenwich	Pub	Grocery Store	Bus Stop	Indian Restaurant	Coffee Shop	Park	Turkish Restaurant	Historic Site	Cons Lanc
23	Greenwich, Lewisham	Pub	Train Station	Café	Fish & Chips Shop	Grocery Store	Hardware Store	Fried Chicken Joint	Italian Restaurant	Re
24	Hackney	Pub	Café	Coffee Shop	Park	Cocktail Bar	Grocery Store	Italian Restaurant	Pizza Place	Ga
25	Hammersmith and Fulham	Pub	Coffee Shop	Café	Grocery Store	Hotel	Gastropub	Pizza Place	Thai Restaurant	
26	Haringey	Pub	Café	Coffee Shop	Grocery Store	Pizza Place	Park	Japanese Restaurant	Bakery	Re
27	Haringey, Barnet	Café	Coffee Shop	Pizza Place	Pub	Japanese Restaurant	Bakery	Deli / Bodega	Grocery Store	Re
28	Haringey, Islington	Hotel	Pub	Fast Food Restaurant	Fish & Chips Shop	Park	Pizza Place	Athletics & Sports	Gym / Fitness Center	Tenr
29	Harrow	Hotel	Theater	Garden	Plaza	Monument / Landmark	Burger Joint	Cocktail Bar	Steakhouse	J Re
30	Harrow, Brent	Bakery	Gym	Construction & Landscaping	Accessories Store	Other Nightlife	Pedestrian Plaza	Park	Paper / Office Supplies Store	F Re
31	Havering	Hotel	Theater	Garden	Plaza	Monument / Landmark	Burger Joint	Cocktail Bar	Steakhouse	J Re
32	Hillingdon	Hotel	Theater	Garden	Plaza	Monument / Landmark	Burger Joint	Cocktail Bar	Steakhouse	J Re
33	Hounslow	Hotel	Garden	Plaza	Theater	Monument / Landmark	Sandwich Place	Burger Joint	Cocktail Bar	
34	Hounslow, Ealing, Hammersmith and Fulham	Café	Pub	Bookstore	Italian Restaurant	Bakery	Coffee Shop	Creperie	Stationery Store	
35	Islington	Coffee Shop	Pub	Café	Food Truck	Vietnamese Restaurant	Italian Restaurant	Park	Cocktail Bar	
36	Islington, Camden	Grocery Store	Pizza Place	Coffee Shop	Pub	Italian Restaurant	Café	Gastropub	Park	Re
37	Islington, City	Coffee Shop	Food Truck	Pub	Hotel	Gym / Fitness Center	Park	Italian Restaurant	Vietnamese Restaurant	B
38	Kensington and Chelsea	Italian Restaurant	Café	Hotel	Bakery	Exhibit	Pub	Science Museum	Bookstore	Coff
39	Kensington and Chelsea, Hammersmith and Fulham	Hotel	Pub	Café	Japanese Restaurant	Coffee Shop	Italian Restaurant	English Restaurant	Plaza	Bo
40	Kingston upon Thames	Hotel	Theater	Plaza	Garden	Monument / Landmark	Sandwich Place	Steakhouse	Japanese Restaurant	I
41	Lambeth	Pub	Café	Coffee Shop	Grocery Store	Bakery	Pizza Place	Park	Fish & Chips Shop	
42	Lambeth, Southwark	Pub	Coffee Shop	Café	Gastropub	Indian Restaurant	Fish & Chips Shop	Grocery Store	Italian Restaurant	Piz
43	Lambeth, Wandsworth	Pub	Café	Burger Joint	Grocery Store	Gym / Fitness Center	Coffee Shop	Pizza Place	Tapas Restaurant	Re
44	Lewisham	Coffee Shop	Café	Pub	Grocery Store	Convenience Store	Clothing Store	Fish & Chips Shop	Train Station	F Re
45	Lewisham, Bromley	Bistro	Gym / Fitness Center	Sandwich Place	Park	Supermarket	Gastropub	Pharmacy	Bakery	Re

46	Lewisham, Southwark	Park	Pub	Wine Shop	Train Station	Tennis Court	Restaurant	Gym / Fitness Center	Flower Shop	
47	Merton	Pub	Coffee Shop	Bar	Café	Clothing Store	Indian Restaurant	Sushi Restaurant	Grocery Store	I
48	Newham	Pub	Café	Hotel	Coffee Shop	Platform	Grocery Store	Fast Food Restaurant	Middle Eastern Restaurant	S
49	Redbridge	Hotel	Plaza	Theater	Garden	Monument / Landmark	Ramen Restaurant	Cocktail Bar	Japanese Restaurant	I
50	Redbridge, Barking and Dagenham	Hotel	Theater	Garden	Plaza	Monument / Landmark	Burger Joint	Cocktail Bar	Steakhouse	J Re
51	Redbridge, Waltham Forest	Café	Hotel	Plaza	Garden	Theater	Bakery	Monument / Landmark	Pub	Pl
52	Richmond upon Thames	Hotel	Plaza	Garden	Theater	Monument / Landmark	Burger Joint	Wine Bar	Sandwich Place	J Re
53	Southwark	Pub	Café	Coffee Shop	Grocery Store	Park	Gym / Fitness Center	Pharmacy	Bar	Re
54	Sutton	Hotel	Theater	Garden	Plaza	Monument / Landmark	Burger Joint	Cocktail Bar	Steakhouse	J Re
55	Sutton, Kingston upon Thames	Hotel	Theater	Garden	Plaza	Monument / Landmark	Burger Joint	Cocktail Bar	Steakhouse	J Re
56	Tower Hamlets	Pub	Coffee Shop	Chinese Restaurant	Café	Park	Bakery	Sandwich Place	Platform	Supe
57	Waltham Forest	Coffee Shop	Café	Pub	Grocery Store	Bakery	Sandwich Place	Pharmacy	Bookstore	Piz
58	Wandsworth	Pub	Coffee Shop	Indian Restaurant	Café	Bar	Supermarket	Portuguese Restaurant	Grocery Store	Bur
59	Westminster	Hotel	Coffee Shop	Sandwich Place	Café	Pub	Italian Restaurant	Hotel Bar	Theater	Re

In [114…

```
'Chinese Restaurant' in neighbourhoods_venues_sorted['3rd Most Common Venue'].unique()
```

Out[114…

```
True
```

In [115…

```
neighbourhoods_venues_sorted['Neighbourhood']

numbers = list(range(1,13))
numbers_str = list(map(str, numbers))
ordinal_numbers = ["st", "nd", "rd", "th", "th", "th", "th", "th", "th", "th", "th", "th"]


for i in range(1, 13):
    a = neighbourhoods_venues_sorted.loc[neighbourhoods_venues_sorted[str(i) + ordinal_numbers[i-1] +' Most Commo
    if ("Chinese Restaurant" in neighbourhoods_venues_sorted[str(i) + ordinal_numbers[i-1] + " Most Common Venue'
        print("A Chinese Restaurant is the", i, ordinal_numbers[i-1], "Most Common Venue in\n", a)
```

```
A Chinese Restaurant is the 3 rd Most Common Venue in
 56    Tower Hamlets
Name: Neighbourhood, dtype: object
A Chinese Restaurant is the 5 th Most Common Venue in
 10    Brent, Ealing
Name: Neighbourhood, dtype: object
```

Clustering Neighbourhoods k-means

In [64]:

```
from sklearn.cluster import KMeans
```

In [116…

```
k_num = 5

london_clustering = london_chinese_resta.drop(['Neighbourhood'], 1)

k_means = KMeans(n_clusters = k_num, random_state=0).fit(london_clustering)
k_means
```

Out[116…

```
KMeans(n_clusters=5, random_state=0)
```

In [117…

```
k_means.labels_[0:]
```

Out[117…

```
array([2, 0, 2, 2, 2, 2, 2, 2, 0, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       4, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 2, 2, 2, 0, 2, 2,
       0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 3, 2, 2, 0], dtype=int32)
```

In [118…

```
london_chinese_resta.insert(2, 'Cluster Labels', k_means.labels_)
london_chinese_resta
```

Out[118…

	Neighbourhood	Chinese Restaurant	Cluster Labels
0	Barking and Dagenham	0.000000	2
1	Barnet	0.012461	0
2	Barnet, Brent, Camden	0.000000	2
3	Barnet, Enfield	0.000000	2
4	Bexley	0.000000	2
5	Bexley, Bromley	0.000000	2
6	Bexley, Greenwich	0.000000	2
7	Bexley, Greenwich	0.000000	2
8	Brent	0.014831	0
9	Brent, Camden	0.000000	2
10	Brent, Ealing	0.125000	1
11	Brent, Ealing, Harrow	0.000000	2
12	Brent, Harrow	0.000000	2
13	Bromley	0.000000	2
14	Camden	0.002488	2
15	Camden, Islington	0.000000	2
16	City	0.004255	2
17	City, Westminster	0.000000	2
18	Croydon	0.000000	2
19	Ealing	0.000000	2
20	Ealing, Hammersmith and Fulham	0.000000	2
21	Enfield	0.000000	2
22	Greenwich	0.026490	4
23	Greenwich, Lewisham	0.000000	2
24	Hackney	0.001667	2
25	Hammersmith and Fulham	0.012195	0
26	Haringey	0.000000	2
27	Haringey, Barnet	0.000000	2
28	Haringey, Islington	0.000000	2
29	Harrow	0.000000	2
30	Harrow, Brent	0.000000	2
31	Havering	0.000000	2
32	Hillingdon	0.000000	2
33	Hounslow	0.000000	2
34	Hounslow, Ealing, Hammersmith and Fulham	0.000000	2
35	Islington	0.005764	0
36	Islington, Camden	0.000000	2
37	Islington, City	0.012658	0
38	Kensington and Chelsea	0.003527	2
39	Kensington and Chelsea, Hammersmith and Fulham	0.000000	2
40	Kingston upon Thames	0.000000	2
41	Lambeth	0.006873	0

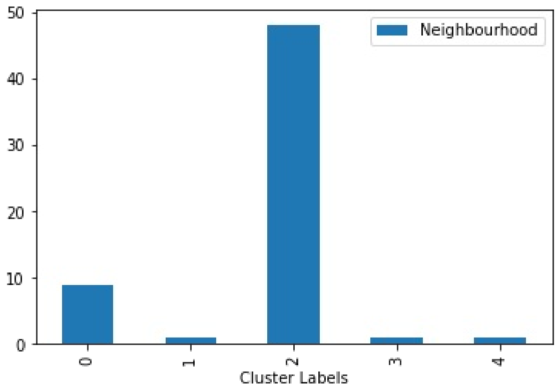
42	Lambeth, Southwark	0.000000	2
43	Lambeth, Wandsworth	0.000000	2
44	Lewisham	0.010000	0
45	Lewisham, Bromley	0.000000	2
46	Lewisham, Southwark	0.000000	2
47	Merton	0.000000	2
48	Newham	0.000000	2
49	Redbridge	0.003413	2
50	Redbridge, Barking and Dagenham	0.000000	2
51	Redbridge, Waltham Forest	0.000000	2
52	Richmond upon Thames	0.000000	2
53	Southwark	0.007619	0
54	Sutton	0.000000	2
55	Sutton, Kingston upon Thames	0.000000	2
56	Tower Hamlets	0.046908	3
57	Waltham Forest	0.003891	2
58	Wandsworth	0.000000	2
59	Westminster	0.014013	0

In [119…

```
data_plot_1 = london_chinese_resta.drop([london_chinese_resta.columns[1]], axis = 1)
data_plot_1.groupby('Cluster Labels').count().plot.bar()
```

Out[119…

```
<AxesSubplot:xlabel='Cluster Labels'>
```

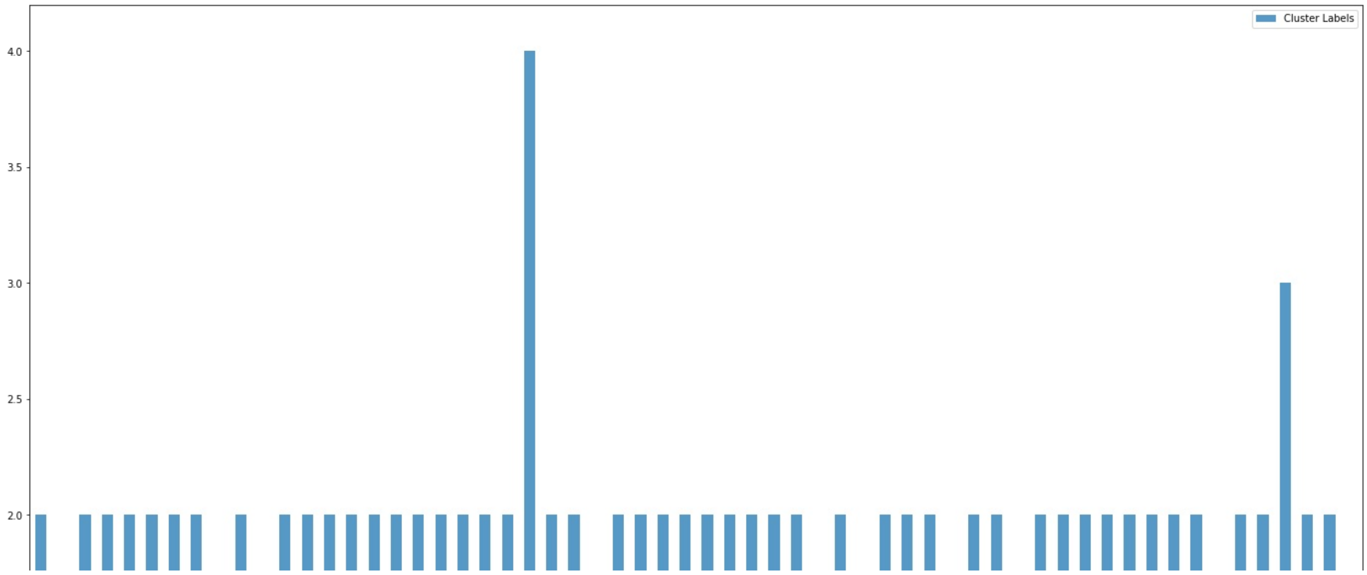


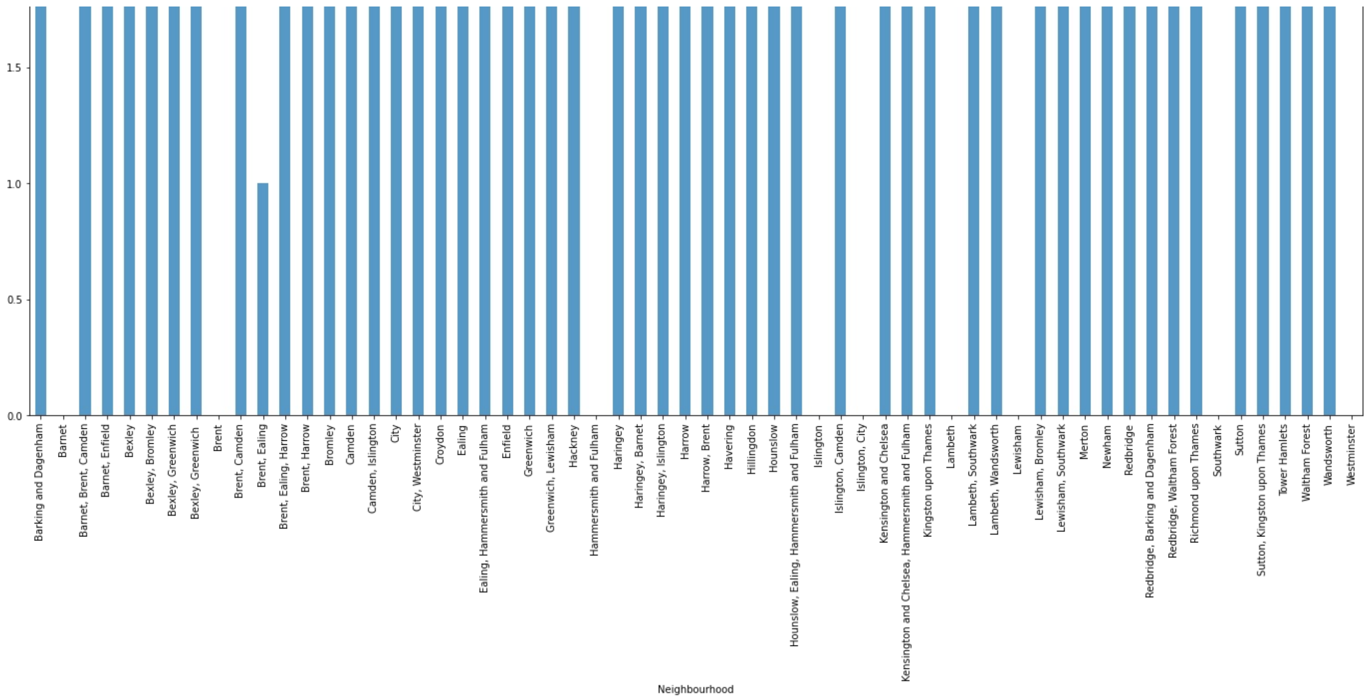
In [120…

```
data_plot_1.set_index('Neighbourhood',inplace=True,drop=True)
data_plot_1.plot(kind='bar',figsize=(24,18),alpha=0.75)
```

Out[120…

```
<AxesSubplot:xlabel='Neighbourhood'>
```

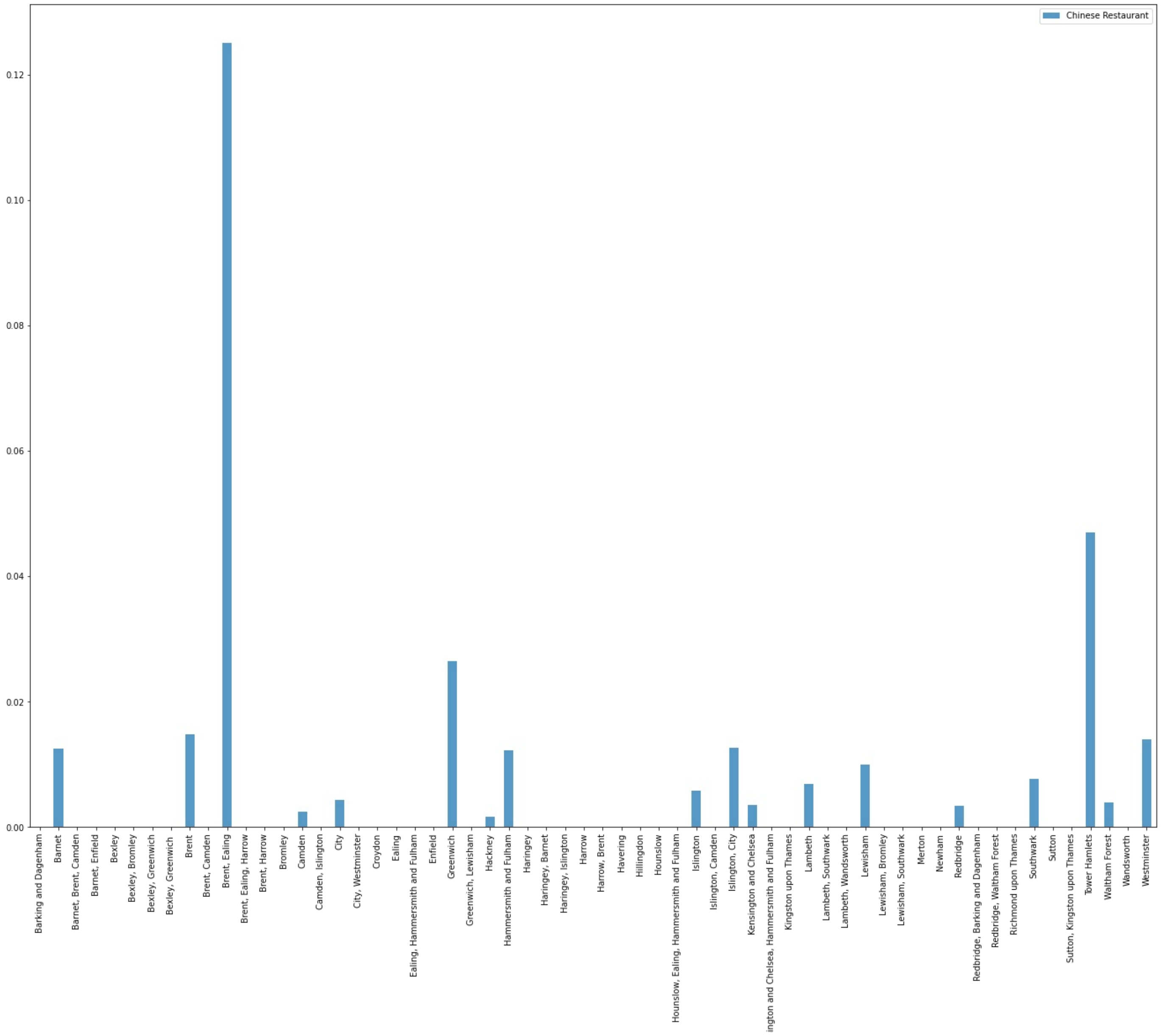




```
In [121… data_plot_2 = london_chinese_resta.drop([london_chinese_resta.columns[2]], axis = 1)
```

```
In [122… data_plot_2.set_index('Neighbourhood',inplace=True,drop=True)
         data_plot_2.plot(kind='bar',figsize=(24,18),alpha=0.75)
```

```
Out[122… <AxesSubplot:xlabel='Neighbourhood'>
```



Neighbourhood

Kens

In [90]:

```
london_merged.columns = ['Neighbourhood','Town','Post-code', 'Latitude', 'Longitude']
london_merged
```

Out[90]:

	Neighbourhood	Town	Post-code	Latitude	Longitude
0	Bexley, Greenwich	LONDON	SE2	51.492450	0.121270
1	Ealing, Hammersmith and Fulham	LONDON	W3, W4	51.513240	-0.267460
2	Croydon	CROYDON	CR0	51.384755	-0.051498
3	Croydon	CROYDON	CR0	51.384755	-0.051498
4	Bexley	BEXLEY, SIDCUP	DA5, DA14	51.506420	-0.127210
...	...	...	...	...	...
526	Greenwich	LONDON	SE18	51.482070	0.071430
527	Sutton, Kingston upon Thames	WORCESTER PARK	KT4	51.506420	-0.127210
528	Hammersmith and Fulham	LONDON	W12	51.506450	-0.236910
529	Hillingdon	HAYES	UB4	51.506420	-0.127210
530	Hillingdon	WEST DRAYTON	UB7	51.506420	-0.127210

531 rows × 5 columns

In [123…

```
final_merged = london_chinese_resta.join(venues_in_london.set_index("Neighbourhood"), on="Neighbourhood")
final_merged
```

Out[123…

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
0	Barking and Dagenham	0.000000	2	51.53312	0.084077	Lidl	Supermarket
0	Barking and Dagenham	0.000000	2	51.53312	0.084077	Tesco Express	Grocery Store
0	Barking and Dagenham	0.000000	2	51.53312	0.084077	Tesco Express	Grocery Store
0	Barking and Dagenham	0.000000	2	51.53312	0.084077	Barking Bathhouse	Spa
0	Barking and Dagenham	0.000000	2	51.53312	0.084077	Greatfields Park	Park
...	...	...	...	...	...	...	...
59	Westminster	0.014013	0	51.49713	-0.138290	The Jugged Hare	Pub
59	Westminster	0.014013	0	51.49713	-0.138290	Laos Cafe	Restaurant
59	Westminster	0.014013	0	51.49713	-0.138290	Subway	Sandwich Place
59	Westminster	0.014013	0	51.49713	-0.138290	Nando's	Portuguese Restaurant
59	Westminster	0.014013	0	51.49713	-0.138290	Loco Mexicano	Mexican Restaurant

21560 rows × 7 columns

In [124…

```
final_merged.sort_values(["Cluster Labels"], inplace=True)
final_merged
```

Out[124…

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
59	Westminster	0.014013	0	51.49713	-0.13829	Loco Mexicano	Mexican Restaurant
35	Islington	0.005764	0	51.56393	-0.12945	Starbucks	Coffee Shop
35	Islington	0.005764	0	51.56393	-0.12945	Costa Coffee	Coffee Shop
35	Islington	0.005764	0	51.56393	-0.12945	Il Mio Mosaic	Italian Restaurant
35	Islington	0.005764	0	51.56393	-0.12945	The Landseer	Pub
...	...	...	...	...	...	...	...

22	Greenwich	0.026490	4	51.48454	0.00275	Star Express	Café
22	Greenwich	0.026490	4	51.48454	0.00275	Gurkha's Inn	Indian Restaurant
22	Greenwich	0.026490	4	51.48454	0.00275	Co-op Food	Grocery Store
22	Greenwich	0.026490	4	51.48454	0.00275	Tyler Street Bus Stop	Bus Stop
22	Greenwich	0.026490	4	51.48454	0.00275	Maze Hill Railway Station (MZH)	Train Station

21560 rows × 7 columns

In [125…

```
map_clusters = folium.Map(location=[london_lat_coords, london_long_coords],zoom_start=10.5)

markers_colors={}
markers_colors[0] = 'red'
markers_colors[1] = 'blue'
markers_colors[2] = 'green'
markers_colors[3] = 'yellow'
markers_colors[4] = 'orange'
markers_colors[5] = 'black'
for lat, lon, cluster in zip(final_merged['Neighbourhood Latitude'], final_merged['Neighbourhood Longitude'], fir


    folium.features.CircleMarker(
        [lat, lon],
        radius=5,

        color =markers_colors[cluster],
        fill_color=markers_colors[5],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

ed to load map: File -> Trust Notebook

Preparing the dataframe in order to plot it

Plot the clusters

Cluster 1 (Red)

In [126…

```
london_chinese_rest_cluster_0 = final_merged.loc[(final_merged['Cluster Labels'] == 0) & (final_merged['Venue Cat
london_chinese_rest_cluster_0.drop_duplicates()
```

Out[126…

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category

35	Islington	0.005764	0	51.52361	-0.09877	New East House	Chinese Restaurant
8	Brent	0.014831	0	51.53938	-0.25205	Good Taste	Chinese Restaurant
53	Southwark	0.007619	0	51.47480	-0.09313	Tasty House	Chinese Restaurant
53	Southwark	0.007619	0	51.47480	-0.09313	Lamoon	Chinese Restaurant
25	Hammersmith and Fulham	0.012195	0	51.53938	-0.25205	Good Taste	Chinese Restaurant
25	Hammersmith and Fulham	0.012195	0	51.47772	-0.20145	Royal China	Chinese Restaurant
44	Lewisham	0.010000	0	51.46268	-0.03558	Bamboo Garden	Chinese Restaurant
25	Hammersmith and Fulham	0.012195	0	51.49617	-0.22935	Steam Restaurant	Chinese Restaurant
41	Lambeth	0.006873	0	51.47480	-0.09313	Lamoon	Chinese Restaurant
41	Lambeth	0.006873	0	51.47480	-0.09313	Tasty House	Chinese Restaurant
37	Islington, City	0.012658	0	51.52361	-0.09877	New East House	Chinese Restaurant
44	Lewisham	0.010000	0	51.47489	-0.04038	Yao Kee	Chinese Restaurant
59	Westminster	0.014013	0	51.49713	-0.13829	A Wong	Chinese Restaurant
59	Westminster	0.014013	0	51.49713	-0.13829	Dragon Inn Club	Chinese Restaurant
59	Westminster	0.014013	0	51.51651	-0.11968	Kam Fung	Chinese Restaurant
59	Westminster	0.014013	0	51.51651	-0.11968	Canton Element	Chinese Restaurant
59	Westminster	0.014013	0	51.52587	-0.19526	Mayflower	Chinese Restaurant
59	Westminster	0.014013	0	51.52587	-0.19526	Gourmet Oriental	Chinese Restaurant
1	Barnet	0.012461	0	51.61568	-0.24511	The Good Earth	Chinese Restaurant
1	Barnet	0.012461	0	51.58918	-0.22805	Jun Peking Chinese Restaurant -	Chinese Restaurant
1	Barnet	0.012461	0	51.60104	-0.19401	Man Chui	Chinese Restaurant

Cluster 2 (Blue)

In [127…

```
london_chinese_rest_cluster_1 = final_merged.loc[(final_merged['Cluster Labels'] == 1) & (final_merged['Venue Cat
london_chinese_rest_cluster_1.drop_duplicates()
```

Out[127…

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
10	Brent, Ealing	0.125	1	51.53938	-0.25205	Good Taste	Chinese Restaurant

Cluster 3 (Green)

In [128…

```
london_chinese_rest_cluster_2 = final_merged.loc[(final_merged['Cluster Labels'] == 2) & (final_merged['Venue Cat
london_chinese_rest_cluster_2.drop_duplicates()
```

Out[128…

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
38	Kensington and Chelsea	0.003527	2	51.48563	-0.18144	New Culture Revolution, SW3	Chinese Restaurant
57	Waltham Forest	0.003891	2	51.55885	-0.00733	Fortune House Chinese Takeaways	Chinese Restaurant
							Chinese

49	Redbridge	0.003413	2	51.58977	0.03052	Wing Sing	Restaurant
16	City	0.004255	2	51.51841	-0.08815	Yauatcha	Chinese Restaurant
14	Camden	0.002488	2	51.51651	-0.11968	Canton Element	Chinese Restaurant
14	Camden	0.002488	2	51.51651	-0.11968	Kam Fung	Chinese Restaurant
24	Hackney	0.001667	2	51.55885	-0.00733	Fortune House Chinese Takeaways	Chinese Restaurant

Cluster 4 (Yellow)

In [129…

```
london_chinese_rest_cluster_3 = final_merged.loc[(final_merged['Cluster Labels'] == 3) & (final_merged['Venue Cat
london_chinese_rest_cluster_3.drop_duplicates()
```

Out[129…

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
56	Tower Hamlets	0.046908	3	51.51122	-0.01264	Sichuan Kitchen	Chinese Restaurant
56	Tower Hamlets	0.046908	3	51.52022	-0.05431	Sinh Le	Chinese Restaurant
56	Tower Hamlets	0.046908	3	51.52022	-0.05431	Tian Tian	Chinese Restaurant

Cluster 5 (Orange)

In [130…

```
london_chinese_rest_cluster_4 = final_merged.loc[(final_merged['Cluster Labels'] == 4) & (final_merged['Venue Cat
london_chinese_rest_cluster_4.drop_duplicates()
```

Out[130…

	Neighbourhood	Chinese Restaurant	Cluster Labels	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Category
22	Greenwich	0.02649	4	51.48747	0.02795	Dragon & Phoenix	Chinese Restaurant
22	Greenwich	0.02649	4	51.48207	0.07143	Capital Noodle Bar	Chinese Restaurant

In [ ]:

Loading [MathJax]/extensions/Safe.js