Course 9/9 Week - Coursea/IBM Data Science Certificate

In [375]:
```
#import librarys
import pandas as pd #pandas
import numpy as np #numpy
import matplotlib.pyplot as plt # For graphics
import seaborn as sns
import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe
from geopy.geocoders import Nominatim
import folium

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

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

Import csv files for London geospatal data

In [376]:
```
LON_GEO = pd.read_csv("Cam_geo.csv")
df_lon = pd.DataFrame (LON_GEO, columns = ['CITY','LOCALITY','NEIGHBOURHOOD','LATITUDE','LONGI

print(df_lon.shape)
print(type(df_lon))
df_lon.head()
```

```
(17, 5)
<class 'pandas.core.frame.DataFrame'>
```

Out[376]:

	CITY	LOCALITY	NEIGHBOURHOOD	LATITUDE	LONGITUDE
0	LONDON	CAMDEN	Belsize Park	51.545045	-0.165609
1	LONDON	CAMDEN	Bloomsbury	51.526342	-0.120229
2	LONDON	CAMDEN	Camden Town	51.544545	-0.133901
3	LONDON	CAMDEN	Chalk Farm	51.543966	-0.154115
4	LONDON	CAMDEN	Fitzrovia	51.518530	-0.137848

Visualising Camden's Localities

In [377]:
```
#return the coordinates for Camden, London
address = 'Camden, London'

geolocator = Nominatim(user_agent="can_explorer")
location = geolocator.geocode(address)
lon_latitude = location.latitude
lon_longitude = location.longitude
print('The geograpical coordinate of Camden in London are {}, {}.'.format(lon_latitude, lon_lo
```

```
The geograpical coordinate of Camden in London are 51.5423045, -0.1395604.
```

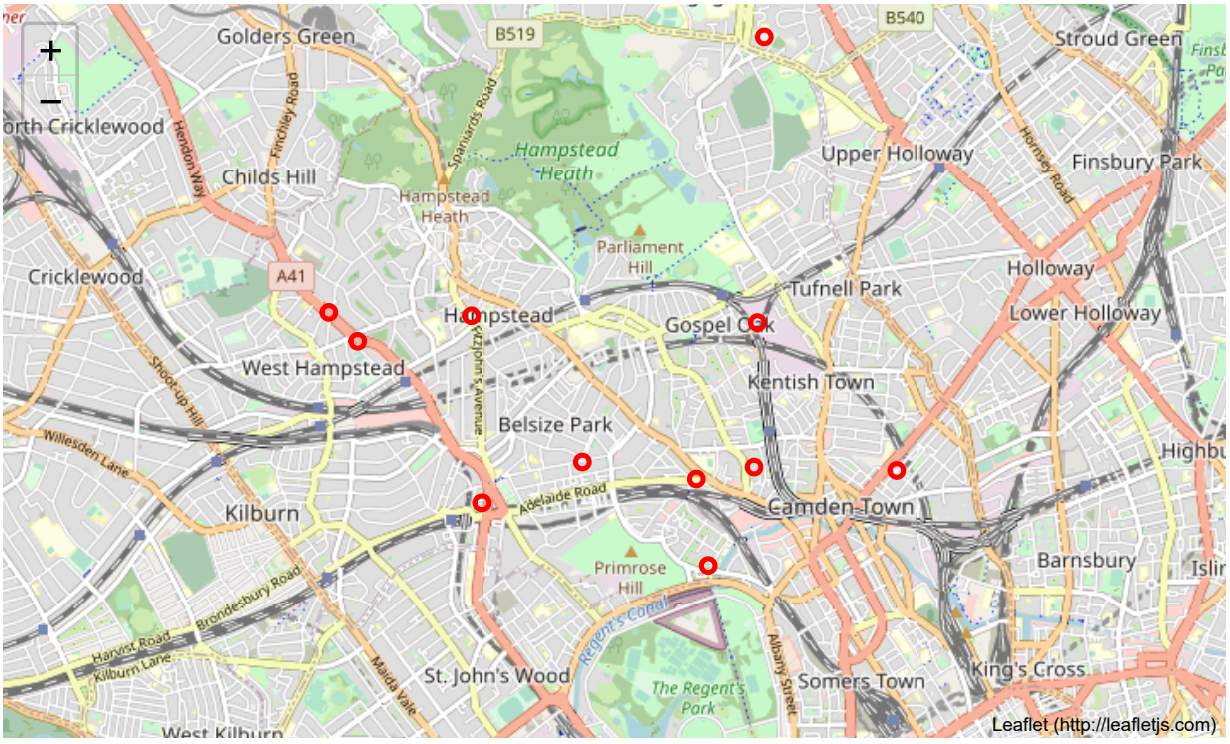
In [378]:
```
#visualising our London locations

london_map = folium.Map(location=[lon_latitude, lon_longitude], zoom_start=13)


for lat, lng, Locality, neighborhood in zip(df_lon['LATITUDE'], df_lon['LONGITUDE'], df_lon['L
    label = '{}, {}'.format(neighborhood, Locality)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=4,
        popup=label,
        color='red',
        fill=True,
        fill_color=0,
        fill_opacity=0.0,
        parse_html=False).add_to(london_map)

london_map
```

Out[378]:



Repeating the process for Sydney

Import csv files for Sydney geospatal data

In [379]:
```
SYD_GEO = pd.read_csv("Syd_geo.csv")
df_syd = pd.DataFrame (SYD_GEO, columns = ['CITY','LOCALITY','NEIGHBOURHOOD','LATITUDE','LONGI

print(df_syd.shape)
print(type(df_syd))
df_syd.head()
```

```
(17, 5)
<class 'pandas.core.frame.DataFrame'>
```

Out[379]:

	CITY	LOCALITY	NEIGHBOURHOOD	LATITUDE	LONGITUDE
0	SYDNEY	CITY OF SYDNEY	Broadway	-33.883514	151.200287
1	SYDNEY	CITY OF SYDNEY	Central	-33.885746	151.204895
2	SYDNEY	CITY OF SYDNEY	Central Park	-33.885256	151.200317
3	SYDNEY	CITY OF SYDNEY	Chinatown	-33.878502	151.204453
4	SYDNEY	CITY OF SYDNEY	Circular Quay	-33.862087	151.211000

In [380]:
```
#return the coordinates for City of Sydney, Sydney
address = 'City of Sydney, Sydney'

geolocator = Nominatim(user_agent="can_explorer")
location = geolocator.geocode(address)
syd_latitude = location.latitude
syd_longitude = location.longitude
print('The geograpical coordinate of the City of Sydney in Sydney are {}, {}.'.format(syd_lati
```

```
The geograpical coordinate of the City of Sydney in Sydney are -33.8853222, 151.2065221.
```

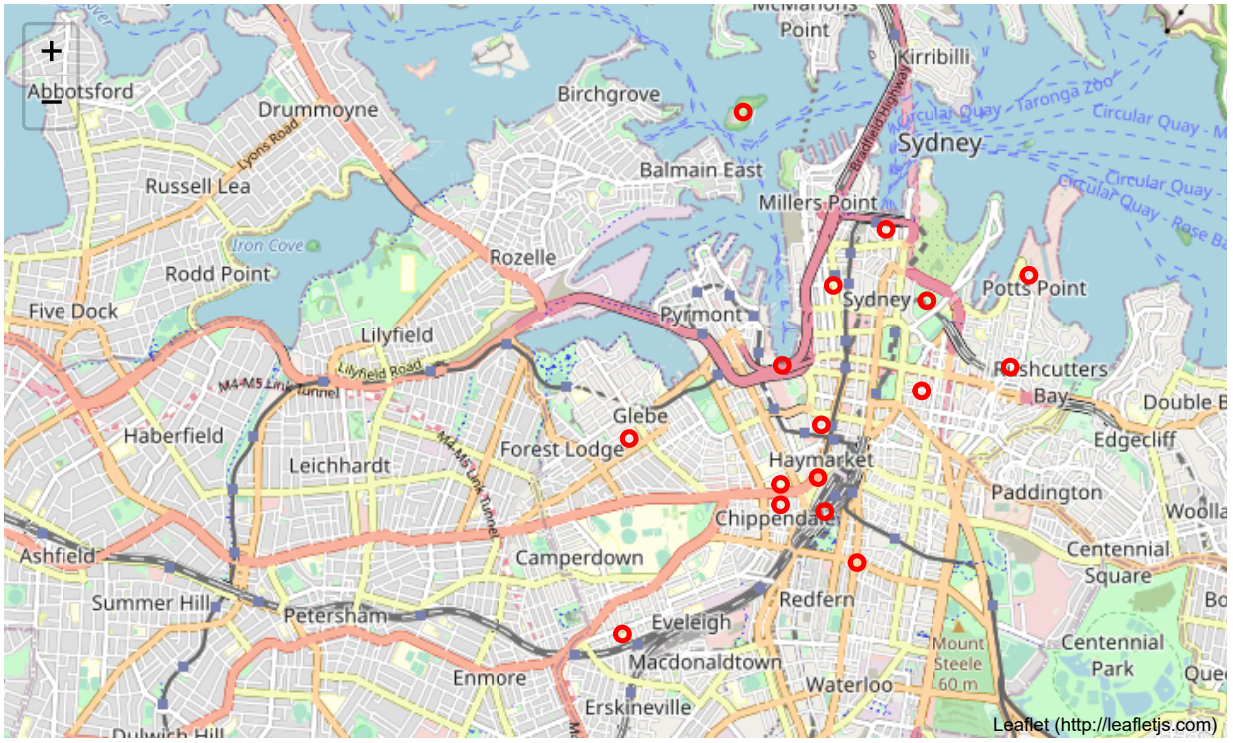
In [381]:
```
#visualising our Sydney locations

sydney_map = folium.Map(location=[syd_latitude, syd_longitude], zoom_start=13)


for lat, lng, Locality, neighborhood in zip(df_syd['LATITUDE'], df_syd['LONGITUDE'], df_syd['L
    label = '{}, {}'.format(neighborhood, Locality)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=4,
        popup=label,
        color='red',
        fill=True,
        fill_color='',
        fill_opacity=0.0,
        parse_html=False).add_to(sydney_map)

sydney_map
```

Out[381]:



Now I need to "append" the two tables into one dataframe

In [382]:
```
frames = [df_lon, df_syd]
df = df_lon
df = df_lon.append(df_syd, ignore_index=True)

print(df.shape)
df
```

```
(34, 5)
```

Out[382]:

	CITY	LOCALITY	NEIGHBOURHOOD	LATITUDE	LONGITUDE
0	LONDON	CAMDEN	Belsize Park	51.545045	-0.165609
1	LONDON	CAMDEN	Bloomsbury	51.526342	-0.120229
2	LONDON	CAMDEN	Camden Town	51.544545	-0.133901
3	LONDON	CAMDEN	Chalk Farm	51.543966	-0.154115
4	LONDON	CAMDEN	Fitzrovia	51.518530	-0.137848
5	LONDON	CAMDEN	Frognal	51.552593	-0.188385
6	LONDON	CAMDEN	Gospel Oak	51.553760	-0.147949
7	LONDON	CAMDEN	Hampstead	51.554212	-0.176781
8	LONDON	CAMDEN	Highgate	51.571734	-0.147219
9	LONDON	CAMDEN	Holborn	51.517355	-0.120599
10	LONDON	CAMDEN	Kentish Town	51.544774	-0.148314
11	LONDON	CAMDEN	Primrose Hill	51.538551	-0.152893
12	LONDON	CAMDEN	Somerstown	51.526572	-0.134636
13	LONDON	CAMDEN	St Giles	51.517355	-0.120599
14	LONDON	CAMDEN	St Pancras	51.526342	-0.120229
15	LONDON	CAMDEN	Swiss Cottage	51.542507	-0.175806
16	LONDON	CAMDEN	West Hampstead	51.554435	-0.191197
17	SYDNEY	CITY OF SYDNEY	Broadway	-33.883514	151.200287
18	SYDNEY	CITY OF SYDNEY	Central	-33.885746	151.204895
19	SYDNEY	CITY OF SYDNEY	Central Park	-33.885256	151.200317
20	SYDNEY	CITY OF SYDNEY	Chinatown	-33.878502	151.204453
21	SYDNEY	CITY OF SYDNEY	Circular Quay	-33.862087	151.211000
22	SYDNEY	CITY OF SYDNEY	Darling Harbour	-33.873585	151.200485
23	SYDNEY	CITY OF SYDNEY	The Domain	-33.868187	151.215057
24	SYDNEY	CITY OF SYDNEY	East Sydney	-33.875622	151.214615
25	SYDNEY	CITY OF SYDNEY	Garden Island	-33.866000	151.225388
26	SYDNEY	CITY OF SYDNEY	Goat Island	-33.852329	151.196564
27	SYDNEY	CITY OF SYDNEY	Green Square	-33.906000	151.203000
28	SYDNEY	CITY OF SYDNEY	Kings Cross	-33.873730	151.223570
29	SYDNEY	CITY OF SYDNEY	Macdonaldtown	-33.896053	151.184326
30	SYDNEY	CITY OF SYDNEY	Railway Square	-33.882900	151.204193
31	SYDNEY	CITY OF SYDNEY	Strawberry Hills	-33.890141	151.208099
32	SYDNEY	CITY OF SYDNEY	St James	-33.879631	151.185104
33	SYDNEY	CITY OF SYDNEY	Wynyard	-33.866798	151.205750

logging into Foursquare API

In [383]:
```
CLIENT_ID = 'LND32RIPQGS5ADMZUNTHJ3GB4ZYSWQRZRPZSE5OVMUT0UPZL'
CLIENT_SECRET = '5OASDGHSYF0B45EHZOYTRPESPILXZFQC3PBOLFC4DSTU2XOQ'
VERSION = '20200126'

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

```
Your credentails:
CLIENT_ID: LND32RIPQGS5ADMZUNTHJ3GB4ZYSWQRZRPZSE5OVMUT0UPZL
CLIENT_SECRET:5OASDGHSYF0B45EHZOYTRPESPILXZFQC3PBOLFC4DSTU2XOQ
```

In [384]:
```
#what is the 1st entry in the pdmerge dataset?

df.loc[0, 'NEIGHBOURHOOD']
```

Out[384]: 'Belsize Park'

In [385]:
```
#Fetch Json file for venues

limit = 100
LIMIT = 100
radius = 500

neighbourhood_latitude = df.loc[0, 'LATITUDE'] # neighborhood latitude value
neighbourhood_longitude = df.loc[0, 'LONGITUDE'] # neighborhood longitude value
neighbourhood_name = df.loc[0, 'NEIGHBOURHOOD'] # neighborhood name


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

results = requests.get(url).json()
```

In [386]:
```
# 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 [387]:
```
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.location.ln
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()
```

```
C:\Anaconda\lib\site-packages\ipykernel_launcher.py:3: FutureWarning: pandas.io.json.json_nor
malize is deprecated, use pandas.json_normalize instead
  This is separate from the ipykernel package so we can avoid doing imports until
```

Out[387]:

	name	categories	lat	lng
0	Chamomile	Café	51.545729	-0.162398
1	Sable D'or	Café	51.545990	-0.162048
2	The Washington	Pub	51.545467	-0.162768
3	Black Truffle	Deli / Bodega	51.545977	-0.162530
4	Starbucks	Coffee Shop	51.545459	-0.162607

In [388]:
```
print('{} venues were returned by Foursquare.'.format(nearby_venues.shape[0]))
```

```
27 venues were returned by Foursquare.
```

Define as a process

In [389]:
```
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={}&v={
            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 = ['Neighbourhood',
                  'Neighbourhood Latitude',
                  'Neighbourhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

In [390]:
```
#repeat for all neighbourhoods
all_venues = getNearbyVenues(names=df['NEIGHBOURHOOD'],
                                     latitudes=df['LATITUDE'],
                                     longitudes=df['LONGITUDE'])
```

```
Belsize Park
Bloomsbury
Camden Town
Chalk Farm
Fitzrovia
Frognal
Gospel Oak
Hampstead
Highgate
Holborn
Kentish Town
Primrose Hill
Somerstown
St Giles
St Pancras
Swiss Cottage
West Hampstead
Broadway
Central
Central Park
Chinatown
Circular Quay
Darling Harbour
The Domain
East Sydney
Garden Island
Goat Island
Green Square
Kings Cross
Macdonaldtown
Railway Square
Strawberry Hills
St James
Wynyard
```

In [391]:
```
print(all_venues.shape)
all_venues.head()
```

```
(1931, 7)
```

Out[391]:

	Neighbourhood	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Belsize Park	51.545045	-0.165609	Chamomile	51.545729	-0.162398	Café
1	Belsize Park	51.545045	-0.165609	Sable D'or	51.545990	-0.162048	Café
2	Belsize Park	51.545045	-0.165609	The Washington	51.545467	-0.162768	Pub
3	Belsize Park	51.545045	-0.165609	Black Truffle	51.545977	-0.162530	Deli / Bodega
4	Belsize Park	51.545045	-0.165609	Starbucks	51.545459	-0.162607	Coffee Shop

In [392]:
```
print('There are {} uniques categories.'.format(len(all_venues['Venue Category'].unique())))
```

```
There are 245 uniques categories.
```

In [393]:
```
# one hot encoding
all_onehot = pd.get_dummies(all_venues[['Venue Category']], prefix="", prefix_sep="")

# add Neighbourhood column back to dataframe
all_onehot['Neighbourhood'] = all_venues['Neighbourhood']

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

all_onehot.head()
```

Out[393]:

	Neighbourhood	Accessories Store	African Restaurant	American Restaurant	Aquarium	Arcade	Argentinian Restaurant	Art Gallery	Art Museum	Arts & Crafts Store
0	Belsize Park	0	0	0	0	0	0	0	0	0
1	Belsize Park	0	0	0	0	0	0	0	0	0
2	Belsize Park	0	0	0	0	0	0	0	0	0
3	Belsize Park	0	0	0	0	0	0	0	0	0
4	Belsize Park	0	0	0	0	0	0	0	0	0

5 rows × 246 columns

In [394]:
```
all_onehot.shape
```

Out[394]: (1931, 246)

In [395]:
```
all_grouped = all_onehot.groupby('Neighbourhood').mean().reset_index()
all_grouped
```

Out[395]:

	Neighbourhood	Accessories Store	African Restaurant	American Restaurant	Aquarium	Arcade	Argentinian Restaurant	Art Gallery	Art Museum	A
0	Belsize Park	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
1	Bloomsbury	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.010000	0.000000	0.0
2	Broadway	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.018182	0.000000	0.0
3	Camden Town	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
4	Central	0.000000	0.000000	0.000000	0.000000	0.000000	0.020408	0.020408	0.000000	0.0
5	Central Park	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.017544	0.000000	0.0
6	Chalk Farm	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.012987	0.000000	0.0
7	Chinatown	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
8	Circular Quay	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.010000	0.0
9	Darling Harbour	0.000000	0.000000	0.019608	0.019608	0.000000	0.000000	0.000000	0.000000	0.0
10	East Sydney	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
11	Fitzrovia	0.000000	0.000000	0.010000	0.000000	0.000000	0.000000	0.010000	0.000000	0.0
12	Frognal	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.030303	0.000000	0.0
13	Garden Island	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
14	Goat Island	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
15	Gospel Oak	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.052632	0.000000	0.0
16	Green Square	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.041667	0.000000	0.0
17	Hampstead	0.000000	0.000000	0.023810	0.000000	0.000000	0.023810	0.000000	0.000000	0.0
18	Highgate	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
19	Holborn	0.012346	0.000000	0.000000	0.000000	0.012346	0.012346	0.000000	0.000000	0.0
20	Kentish Town	0.000000	0.000000	0.022222	0.000000	0.000000	0.000000	0.022222	0.000000	0.0
21	Kings Cross	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
22	Macdonaldtown	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
23	Primrose Hill	0.000000	0.000000	0.000000	0.015873	0.000000	0.000000	0.000000	0.015873	0.0
24	Railway Square	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
25	Somerstown	0.000000	0.014085	0.000000	0.000000	0.000000	0.000000	0.014085	0.000000	0.0
26	St Giles	0.012346	0.000000	0.000000	0.000000	0.012346	0.012346	0.000000	0.000000	0.0
27	St James	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
28	St Pancras	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.010000	0.000000	0.0
29	Strawberry Hills	0.000000	0.000000	0.000000	0.000000	0.000000	0.012987	0.000000	0.000000	0.0
30	Swiss Cottage	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
31	The Domain	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.034483	0.034483	0.0
32	West Hampstead	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
33	Wynyard	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0

34 rows × 246 columns

In [396]:
```
num_top_venues = 5

for hood in all_grouped['Neighbourhood']:
    print("----"+hood+"----")
    temp = all_grouped[all_grouped['Neighbourhood'] == 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')
```

```
----Belsize Park----
                venue  freq
0               Café  0.19
1              Hotel  0.07
2  Italian Restaurant  0.07
3                Pub  0.07
4   Convenience Store  0.07


----Bloomsbury----
                venue  freq
0              Hotel  0.10
1                Pub  0.10
2        Coffee Shop  0.08
3               Café  0.06
4  Italian Restaurant  0.05


----Broadway----
```

In [397]:
```
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 [398]:
```
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Neighbourhood']
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['Neighbourhood'] = all_grouped['Neighbourhood']

for ind in np.arange(all_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(all_grouped.iloc[ind

neighborhoods_venues_sorted.head()
```

Out[398]:

	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	9th Co
0	Belsize Park	Café	Hotel Bar	Hotel	Pub	Italian Restaurant	Convenience Store	Market	Deli / Bodega	
1	Bloomsbury	Hotel	Pub	Coffee Shop	Café	Italian Restaurant	Bookstore	Bakery	Burger Joint	I M
2	Broadway	Café	Thai Restaurant	Bar	Coffee Shop	Pub	Wine Bar	Dumpling Restaurant	Hotel	Art (
3	Camden Town	Pub	Café	Italian Restaurant	Garden Center	Park	Grocery Store	Event Space	Caribbean Restaurant	
4	Central	Café	Thai Restaurant	Coffee Shop	Indonesian Restaurant	Bar	Hostel	Pub	Comedy Club	Du Rest

In [399]:
```
num_top_venues = 10

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

# create columns according to number of top venues
columns = ['Neighbourhood']
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
heatmapdata = pd.DataFrame(columns=columns)
heatmapdata['Neighbourhood'] = all_grouped['Neighbourhood']

for ind in np.arange(all_grouped.shape[0]):
    heatmapdata.iloc[ind, 1:] = return_most_common_venues(all_grouped.iloc[ind, :], num_top_ve

heatmapdata.head()
```

Out[399]:

	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	9th Co
0	Belsize Park	Café	Hotel Bar	Hotel	Pub	Italian Restaurant	Convenience Store	Market	Deli / Bodega	
1	Bloomsbury	Hotel	Pub	Coffee Shop	Café	Italian Restaurant	Bookstore	Bakery	Burger Joint	I M
2	Broadway	Café	Thai Restaurant	Bar	Coffee Shop	Pub	Wine Bar	Dumpling Restaurant	Hotel	Art (
3	Camden Town	Pub	Café	Italian Restaurant	Garden Center	Park	Grocery Store	Event Space	Caribbean Restaurant	
4	Central	Café	Thai Restaurant	Coffee Shop	Indonesian Restaurant	Bar	Hostel	Pub	Comedy Club	Du Rest

In [400]:
```
# set number of clusters
kclusters = 4

all_grouped_clustering = all_grouped.drop('Neighbourhood', 1)

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

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

Out[400]:
```
array([3, 0, 2, 0, 2, 2, 3, 3, 3, 3])
```

In [401]:
```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
neighborhoods_venues_sorted
```

Out[401]:

	Cluster Labels	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 M Comn Ve
0	3	Belsize Park	Café	Hotel Bar	Hotel	Pub	Italian Restaurant	Convenience Store	Ma
1	0	Bloomsbury	Hotel	Pub	Coffee Shop	Café	Italian Restaurant	Bookstore	Bak
2	2	Broadway	Café	Thai Restaurant	Bar	Coffee Shop	Pub	Wine Bar	Dump Restau
3	0	Camden Town	Pub	Café	Italian Restaurant	Garden Center	Park	Grocery Store	Event Sp
4	2	Central	Café	Thai Restaurant	Coffee Shop	Indonesian Restaurant	Bar	Hostel	I
5	2	Central Park	Café	Bar	Bakery	Thai Restaurant	Pub	Hotel	Wine

In [402]:
```
finaldf = df.join(neighborhoods_venues_sorted.set_index('Neighbourhood'), on='NEIGHBOURHOOD')
finaldf.head()
```

Out[402]:

	CITY	LOCALITY	NEIGHBOURHOOD	LATITUDE	LONGITUDE	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th M Comm Ve
0	LONDON	CAMDEN	Belsize Park	51.545045	-0.165609	3	Café	Hotel Bar	Hotel	F
1	LONDON	CAMDEN	Bloomsbury	51.526342	-0.120229	0	Hotel	Pub	Coffee Shop	C
2	LONDON	CAMDEN	Camden Town	51.544545	-0.133901	0	Pub	Café	Italian Restaurant	Gar Cer
3	LONDON	CAMDEN	Chalk Farm	51.543966	-0.154115	3	Café	Bar	Italian Restaurant	F
4	LONDON	CAMDEN	Fitzrovia	51.518530	-0.137848	0	Coffee Shop	Clothing Store	Pizza Place	H

In [403]:
```
cluster_0 = finaldf.loc[finaldf['Cluster Labels'] == 0, finaldf.columns[[1] + list(range(2, fi
cluster_1 = finaldf.loc[finaldf['Cluster Labels'] == 1, finaldf.columns[[1] + list(range(2, fi
cluster_2 = finaldf.loc[finaldf['Cluster Labels'] == 2, finaldf.columns[[1] + list(range(2, fi
cluster_3 = finaldf.loc[finaldf['Cluster Labels'] == 3, finaldf.columns[[1] + list(range(2, fi
```

In [404]:
```
print(cluster_0.shape)
cluster_0
```

```
(14, 15)
```

Out[404]:

	LOCALITY	NEIGHBOURHOOD	LATITUDE	LONGITUDE	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	
1	CAMDEN	Bloomsbury	51.526342	-0.120229	0	Hotel	Pub	Coffee Shop	Café	R
2	CAMDEN	Camden Town	51.544545	-0.133901	0	Pub	Café	Italian Restaurant	Garden Center	
4	CAMDEN	Fitzrovia	51.518530	-0.137848	0	Coffee Shop	Clothing Store	Pizza Place	Hotel	
5	CAMDEN	Frognal	51.552593	-0.188385	0	Pub	Bakery	Café	Chinese Restaurant	
6	CAMDEN	Gospel Oak	51.553760	-0.147949	0	Gym / Fitness Center	Pool	Vietnamese Restaurant	Farm	
7	CAMDEN	Hampstead	51.554212	-0.176781	0	Bakery	Pub	Café	Italian Restaurant	
8	CAMDEN	Highgate	51.571734	-0.147219	0	Pub	Coffee Shop	Indian Restaurant	Deli / Bodega	
9	CAMDEN	Holborn	51.517355	-0.120599	0	Pub	Sandwich Place	Theater	Hotel	R
11	CAMDEN	Primrose Hill	51.538551	-0.152893	0	Zoo Exhibit	Coffee Shop	Italian Restaurant	Pub	
12	CAMDEN	Somerstown	51.526572	-0.134636	0	Coffee Shop	Gym / Fitness Center	Café	Indian Restaurant	I
13	CAMDEN	St Giles	51.517355	-0.120599	0	Pub	Sandwich Place	Theater	Hotel	R
14	CAMDEN	St Pancras	51.526342	-0.120229	0	Hotel	Pub	Coffee Shop	Café	R
15	CAMDEN	Swiss Cottage	51.542507	-0.175806	0	Coffee Shop	Café	Italian Restaurant	Grocery Store	R
16	CAMDEN	West Hampstead	51.554435	-0.191197	0	Indian Restaurant	Café	Pub	Breakfast Spot	I

In [405]:
```
HMAP0 = cluster_0[['1st Most Common Venue','2nd Most Common Venue','3rd Most Common Venue','4t
cor = HMAP0.corr() #Calculate the correlation of the above variables
#sns.heatmap(cor, square = True) #Plot the correlation as heat map
cor
```

Out[405]:

In [406]:
```
print(cluster_1.shape)
cluster_1
```

```
(1, 15)
```

Out[406]:

	LOCALITY	NEIGHBOURHOOD	LATITUDE	LONGITUDE	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th M Comn Ve
26	CITY OF SYDNEY	Goat Island	-33.852329	151.196564	1	Park	Pier	Bus Station	Harbor / Marina	Boa F

In [407]:
```
print(cluster_2.shape)
cluster_2
```

```
(8, 15)
```

Out[407]:

	LOCALITY	NEIGHBOURHOOD	LATITUDE	LONGITUDE	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue
17	CITY OF SYDNEY	Broadway	-33.883514	151.200287	2	Café	Thai Restaurant	Bar	Coffee Shop
18	CITY OF SYDNEY	Central	-33.885746	151.204895	2	Café	Thai Restaurant	Coffee Shop	Indonesian Restaurant
19	CITY OF SYDNEY	Central Park	-33.885256	151.200317	2	Café	Bar	Bakery	Thai Restaurant
27	CITY OF SYDNEY	Green Square	-33.906000	151.203000	2	Café	Coffee Shop	Sporting Goods Shop	Electronics Store
29	CITY OF SYDNEY	Macdonaldtown	-33.896053	151.184326	2	Thai Restaurant	Café	Bar	Pub
30	CITY OF SYDNEY	Railway Square	-33.882900	151.204193	2	Thai Restaurant	Café	Ice Cream Shop	Coffee Shop
31	CITY OF SYDNEY	Strawberry Hills	-33.890141	151.208099	2	Café	Pub	Japanese Restaurant	Coffee Shop
32	CITY OF SYDNEY	St James	-33.879631	151.185104	2	Café	Pub	Pizza Place	Indian Restaurant

In [408]:
```
print(cluster_3.shape)
cluster_3
```

```
(11, 15)
```

Out[408]:

	LOCALITY	NEIGHBOURHOOD	LATITUDE	LONGITUDE	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue
0	CAMDEN	Belsize Park	51.545045	-0.165609	3	Café	Hotel Bar	Hotel	Pub
3	CAMDEN	Chalk Farm	51.543966	-0.154115	3	Café	Bar	Italian Restaurant	Pub
10	CAMDEN	Kentish Town	51.544774	-0.148314	3	Café	Pub	Market	Coffee Shop
20	CITY OF SYDNEY	Chinatown	-33.878502	151.204453	3	Thai Restaurant	Japanese Restaurant	Hotel	Korean Restaurant
21	CITY OF SYDNEY	Circular Quay	-33.862087	151.211000	3	Café	Hotel	Cocktail Bar	Steakhouse
22	CITY OF SYDNEY	Darling Harbour	-33.873585	151.200485	3	Hotel	Café	Japanese Restaurant	Thai Restaurant
23	CITY OF SYDNEY	The Domain	-33.868187	151.215057	3	Café	Sandwich Place	Fountain	Steakhouse
24	CITY OF SYDNEY	East Sydney	-33.875622	151.214615	3	Café	Japanese Restaurant	Pizza Place	Bakery
25	CITY OF SYDNEY	Garden Island	-33.866000	151.225388	3	Café	Chinese Restaurant	Park	Australian Restaurant
28	CITY OF SYDNEY	Kings Cross	-33.873730	151.223570	3	Café	Italian Restaurant	Coffee Shop	Australian Restaurant
33	CITY OF SYDNEY	Wynyard	-33.866798	151.205750	3	Café	Bar	Cocktail Bar	Coffee Shop

In [409]:
```
#a quick check to ensure that the total number of results in all of my clusters matched that o
14+1+8+11
```

Out[409]: 34

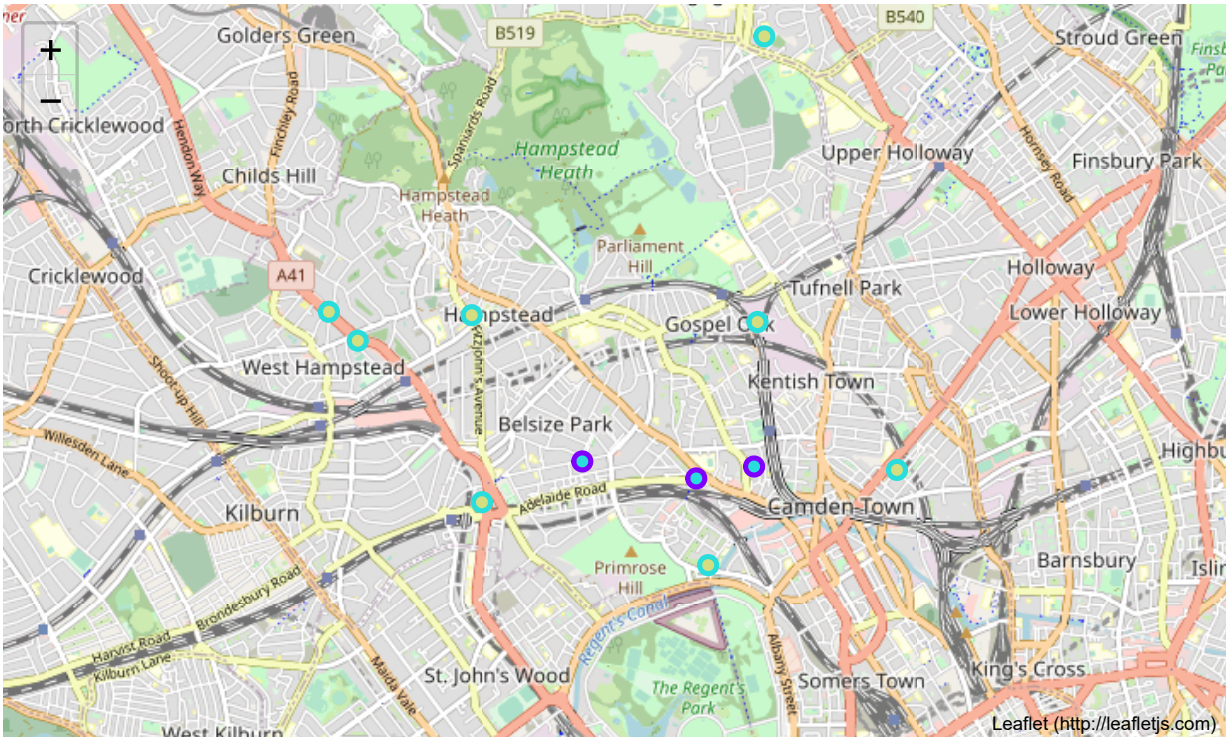
In [410]:
```
# create london map
map_lonclusters = folium.Map(location=[lon_latitude, lon_longitude], zoom_start=13)

# 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(finaldf['LATITUDE'], finaldf['LONGITUDE'],finaldf['NEIGHBOUR
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color = rainbow[int(cluster)-3],
        fill=True,
        fill_color=rainbow[int(cluster)-2],
        fill_opacity=1).add_to(map_lonclusters)

map_lonclusters
```

Out[410]:



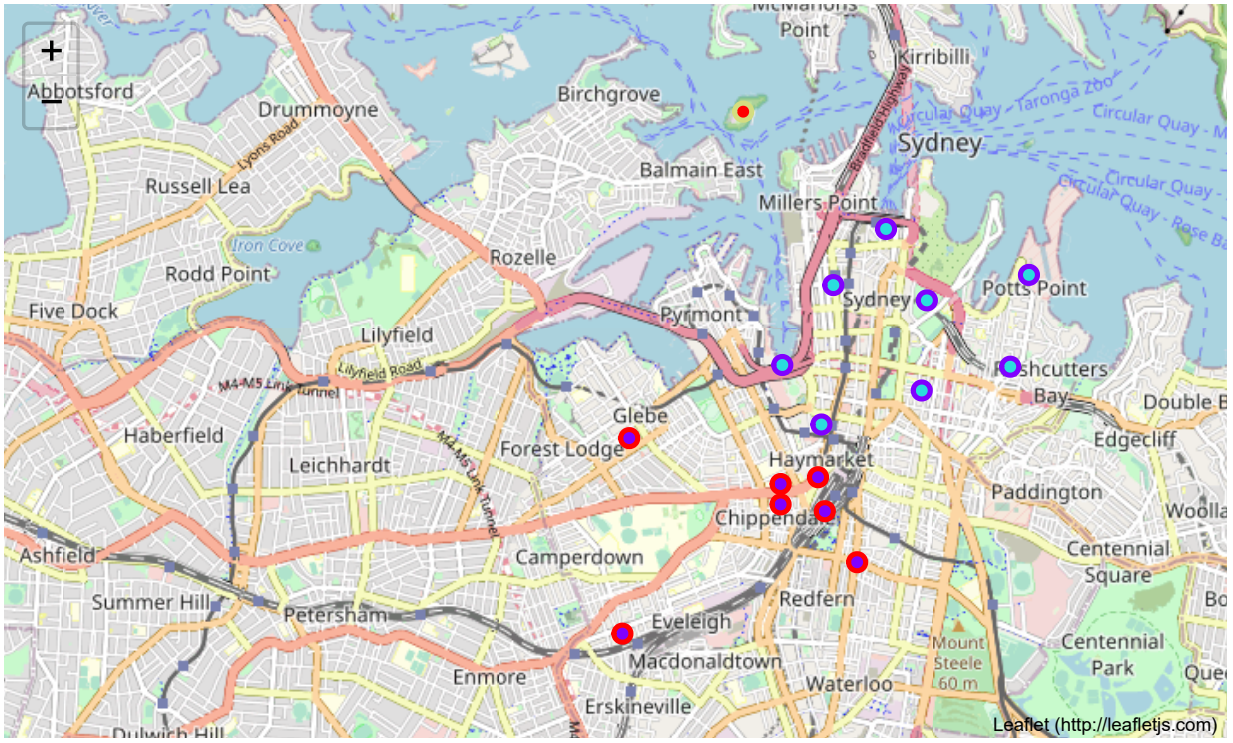
In [411]:
```
# create Sydney map
map_sydclusters = folium.Map(location=[syd_latitude, syd_longitude], zoom_start=13)

# 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(finaldf['LATITUDE'], finaldf['LONGITUDE'],finaldf['NEIGHBOUR
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color = rainbow[int(cluster)-3],
        fill=True,
        fill_color=rainbow[int(cluster)-2],
        fill_opacity=1).add_to(map_sydclusters)

map_sydclusters
```

Out[411]:



High level conclusions

There is only one cluster which groups neighbourhoods from both Sydney and London - Cluster 3 (11 of which are in Sydney and 3 are in Camden) the analysis suggests that these area's should be visited first as to whether they are suitiable for re-location.

Out of the 5 area's mapped, 1 area is totally unqiue and shares few characteristics with any other neighbourhood - Goat Island (Cluster 1)

Two area's are unique to their related Cities (Cluster 0 in London and Cluster 2 in Sydney)

In [ ]:

In [ ]: