Coursera_Capstone_London

June 27, 2021

1 Best places to open an Asian Fine-Dining place in London

By Kartik Patil

1.1 Table of Contents

1.2 Introduction: Business Problem

The aim of the following report is to suggest locations around London which are ideal to open an Asian Fine-Dining Restaurants. An Asian fine-dining chain that serves Ramen, Pad Thai and other assorted dishes from East and South East Asia wants to make its foray into London and wants to find ideal locations for opening its restaurants. The idea behind having a successfull business is that the location should be able to attract a middle-class eating out crowd, while at the same time not facing intense competition from other restaurants, namely other fine-dining and asian restaurants. Therefore, the restaurants are opened based on the following premise: * **The chain wants to open its restaurants in areas which do not already have a wide number of Asian restaurants** * **The restaurants need to be as close to Central London due to ease of commute.** * **The restaurants should be close to as possible to other rich areas, which have malls, business centres or tourists attractions, to attract the appropriate crowd.**

1.3 Data

To appropriately locate our restaurant and solve our business problem we need to find neighbourhoods which are: 1. **Not filled with Asian or Fine-Dining restaurants.** 2. **Are closer to Central London.** 3. **Are in expensive areas or areas with high amounts of movement.** 4. **Are as close to Central London as possible.**

The Data is gathered using the Foursquare API to locate and categorise these restaurants into their respective categories. The data is then wrangled into a Pandas Dataframe for further analysis. We also use the to find the most visited areas in London and add that data to neighbourhoods. Furthermore, we encode——

[1]:
```
import numpy as np # library to handle data in a vectorized manner
from bs4 import BeautifulSoup
import pandas as pd # library for data analsysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

#!conda install -c conda-forge geopy --yes # uncomment this line if you haven't␣
↪completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and␣
↪longitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas␣
↪dataframe

# 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

#!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you␣
↪haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

```
Libraries imported.
```

First we find the co-ordinates for the Centre of London, so as to calculate the distances of different neighbourhoods from the centre of London. In our case, we assume the centre of London to be Trafalgar Square and therefore, we find the co-ordinates for Trafalgar Square using the following code:

[3]:
```
import requests

#define function to get coordinates using the Google Geocoding API
def get_coordinates(api_key, address, verbose=False):
    try:
        url = 'https://maps.googleapis.com/maps/api/geocode/json?
↪key={}&address={}'.format(api_key, address)
        response = requests.get(url).json()
        if verbose:
            print('Google Maps API JSON result =>', response)
        results = response['results']
```

```
        geographical_data = results[0]['geometry']['location'] # get␣
↪geographical coordinates
        lat = geographical_data['lat']
        lon = geographical_data['lng']
        return [lat, lon]
    except:
        return [None, None]
address = 'Trafalgar Square, London, UK'
london_center = get_coordinates(google_api_key, address)
print('Coordinate of {}: {}'.format(address, london_center))
```

```
Coordinate of Trafalgar Square, London, UK: [51.5075873, -0.1278296]
```

Next up we find all the neighborhoods in London as well as their latitudes and longitudes. We scrape this data off Wikipedia, using Beautiful Soup.

[4]:
```
#retrieve London neighborhoods as beautiful soup objects
london_data = pd.DataFrame()
url = 'https://en.wikipedia.org/wiki/List_of_areas_of_London'
london_neigh = requests.get(url).text
soup = BeautifulSoup(london_neigh, 'html5lib')
```

[5]:
```
#find and extract neighborhoods
table = soup.find('table', {'class':'wikitable sortable'}).tbody
# Extracts all "tr" (table rows) within the table above
rows = table.find_all('tr')
# Extracts the column headers, removes and replaces possible '\n' with space␣
↪for the "th" tag
columns = [i.text.replace('\n', '')
           for i in rows[0].find_all('th')]
# Converts columns to pd dataframe
df = pd.DataFrame(columns = columns)
'''
Extracts every row with corresponding columns then appends the values to the␣
↪create pd dataframe "df". The first row (row[0]) is skipped because it is␣
↪already the header
'''
for i in range(1, len(rows)):
    tds = rows[i].find_all('td')
    if len(tds) == 7:
        values = [tds[0].text, tds[1].text, tds[2].text.replace('\n', ''.
↪replace('\xa0','')), tds[3].text, tds[4].text.replace('\n', ''.
↪replace('\xa0','')), tds[5].text.replace('\n', ''.replace('\xa0','')),␣
↪tds[6].text.replace('\n', ''.replace('\xa0',''))]
    else:
        values = [td.text.replace('\n', '').replace('\xa0','') for td in tds]

        df = df.append(pd.Series(values, index = columns), ignore_index = True)
```

```
        df
```

[6]:
```
df.head()
```

[6]:
```
       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

  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
```

[7]:
```
df.shape
```

[7]:
```
(531, 6)
```

Once we have all the neighbourhoods, we get the latitudes and longitudes for each of the neighborhoods using the Google Maps geocoding API as well as the function we wrote above.

[8]:
```
#get the latitudes and longitudes for all neighborhoods
lat_list = {}
for location in df['Location']:
    locale = location+", London, UK"
    lat_long=get_coordinates(google_api_key, locale)
    lat_list[location]={}
    lat_list[location]['latitude']=lat_long[0]
    lat_list[location]['longitude']=lat_long[1]
print("complete")
```

```
complete
```

[9]:
```
#get the latitudes and longitudes in a dataframe
lat_list_df = pd.DataFrame()
for ind, location in enumerate(lat_list):
    latitude = lat_list[location]['latitude']
    longitude = lat_list[location]['longitude']
    lat_list_df=lat_list_df.append({"Location": location,
                       "Latitude": latitude,
                       "Longitude": longitude},
                        ignore_index=True)
lat_list_df.head()
```

```
[9]:     Latitude     Location  Longitude
     0  51.492612   Abbey Wood   0.118818
     1  51.508372        Acton  -0.274440
     2  51.358673    Addington  -0.031254
     3  51.380550   Addiscombe  -0.072274
     4  51.426316  Albany Park   0.102809
```

We now clean the data to make it workable. Firstly, we do not need the Dial code or OS grid ref so we remove these two columns from our data frame. Furthermore, we remove the numbers next to London boroughs as these are the references from Wikipedia. Then we calculate the distance from centre for either of these neighborhoods and add the column to our dataframe. Lastly, London is big and therefore, it isn't feasible to get venues in all neighbourhoods. Since, our client wants the restaurants to be situated closer to the centre we only keep those neighborhoods that are within 15 kms of Trafalgar Square. Once the data is all cleaned up we retrieve the venue details for each neighborhood using the Foursquare API.

```
[12]: #import geopy for distance
from geopy import distance
london_data = df.merge(lat_list_df, how='inner', on='Location')
london_data = london_data.drop(london_data.columns[[4,5]], axis=1)
london_data['London\xa0borough'] = london_data['London\xa0borough'].map(lambda␣
 ↪x: x.rstrip('\]0123456789\['))
distance_list = {}
for location, lat, lng in zip(london_data['Location'], london_data['Latitude'],␣
 ↪london_data['Longitude']):
    distance_list[location] = {}
    loc_lat_lng = (lat, lng)
    distance_list[location] = distance.distance(loc_lat_lng, london_center).km
distance_df = pd.DataFrame()
for ind, location in enumerate(distance_list):
    distance = distance_list[location]
    distance_df = distance_df.append({'Location':location, 'Distance from␣
 ↪Center': distance}, ignore_index=True)
london_data = london_data.merge(distance_df, how='inner', on='Location')
drop_index = london_data[london_data['Distance from Center'] > 15].index
london_data = london_data.drop(drop_index)
london_data.head()
```

```
[12]:     Location                   London borough Post town Postcode district  \
     1       Acton  Ealing, Hammersmith and Fulham    LONDON            W3, W4
     3  Addiscombe                         Croydon   CROYDON               CR0
     6     Aldgate                            City    LONDON               EC3
     7     Aldwych                     Westminster    LONDON               WC2
     8    Alperton                           Brent   WEMBLEY               HA0

        Latitude  Longitude  Distance from Center
     1  51.508372  -0.274440             10.179297
```

```
3  51.380550  -0.072274            14.652051
6  51.513438  -0.077171             3.576673
7  51.513266  -0.117183             0.972334
8  51.539601  -0.298837            12.391598
```

[14]:

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

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

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

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

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

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

    print("complete")
    return(nearby_venues)
```

[15]:
```python
london_venues = getNearbyVenues(names=london_data['Location'],
                                 latitudes=london_data['Latitude'],
                                 longitudes=london_data['Longitude']
                                )
london_venues.head()
```

```
complete
```

[15]:
```
  Location  Location Latitude  Location Longitude  \
0    Acton          51.508372            -0.27444
1    Acton          51.508372            -0.27444
2    Acton          51.508372            -0.27444
3    Acton          51.508372            -0.27444
4    Acton          51.508372            -0.27444

                                 Venue  Venue Latitude  Venue Longitude  \
0                    London Star Hotel       51.509624        -0.272456
1                          The Aeronaut       51.508376        -0.275216
2                     Amigo's Peri Peri       51.508396        -0.274561
3  Dragonfly Brewery at George & Dragon       51.507378        -0.271702
4                North China Restaurant       51.508251        -0.277435

        Venue Category
0                Hotel
1                  Pub
2  Fast Food Restaurant
3              Brewery
4   Chinese Restaurant
```

[16]:
```python
london_venues = london_venues.merge(distance_df, how='inner', on='Location')
london_venues.head()
```

[16]:
```
  Location  Location Latitude  Location Longitude  \
0    Acton          51.508372            -0.27444
1    Acton          51.508372            -0.27444
2    Acton          51.508372            -0.27444
3    Acton          51.508372            -0.27444
4    Acton          51.508372            -0.27444

                                 Venue  Venue Latitude  Venue Longitude  \
0                    London Star Hotel       51.509624        -0.272456
1                          The Aeronaut       51.508376        -0.275216
2                     Amigo's Peri Peri       51.508396        -0.274561
3  Dragonfly Brewery at George & Dragon       51.507378        -0.271702
4                North China Restaurant       51.508251        -0.277435

        Venue Category  Distance from Center
0                Hotel             10.179297
```

```
1                      Pub             10.179297
2   Fast Food Restaurant             10.179297
3                  Brewery             10.179297
4     Chinese Restaurant             10.179297
```

1.4 Methodology

We now use our use the Pandas dataframe to find the best locations to open a Asian fine dining restaurant. In order to do so, we first do one hot encoding to convert all venue categories in dummies and then use clustering to find the similar neighborhoods. We use the one hot encoding to find the most popular venues in each neighborhood and then cluster regions into 5 clusters. We then observe the clusters for similarity.

[17]:
```python
# one hot encoding
london_onehot = pd.get_dummies(london_venues[['Venue Category']], prefix="",␣
 ↪prefix_sep="")

# add neighborhood column back to dataframe
london_onehot['Location'] = london_venues['Location']

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

london_onehot.head()
```

[17]:
```
  Location  Accessories Store  Adult Boutique  Afghan Restaurant  \
0    Acton                  0               0                  0
1    Acton                  0               0                  0
2    Acton                  0               0                  0
3    Acton                  0               0                  0
4    Acton                  0               0                  0

   African Restaurant  Airport Service  Airport Terminal  American Restaurant  \
0                   0                0                 0                    0
1                   0                0                 0                    0
2                   0                0                 0                    0
3                   0                0                 0                    0
4                   0                0                 0                    0

   Animal Shelter  Antique Shop  Aquarium  Arcade  Arepa Restaurant  \
0               0             0         0       0                 0
1               0             0         0       0                 0
2               0             0         0       0                 0
3               0             0         0       0                 0
4               0             0         0       0                 0
```

```
   Veneto Restaurant  Video Game Store  Vietnamese Restaurant  \
0                0.0               0.0               0.000000
1                0.0               0.0               0.000000
2                0.0               0.0               0.011765
3                0.0               0.0               0.000000
4                0.0               0.0               0.000000

   Warehouse Store  Waterfront  Whisky Bar  Wine Bar  Wine Shop  Winery  \
0              0.0         0.0         0.0  0.000000   0.000000     0.0
1              0.0         0.0         0.0  0.000000   0.000000     0.0
2              0.0         0.0         0.0  0.011765   0.011765     0.0
3              0.0         0.0         0.0  0.010000   0.010000     0.0
4              0.0         0.0         0.0  0.000000   0.000000     0.0

   Wings Joint  Women's Store  Xinjiang Restaurant  Yoga Studio  Zoo Exhibit
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
```

[20]:

```
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]
#top 10 most common venues
num_top_venues = 10

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


columns = ['Location']
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))

neighborhoods_venues_sorted = pd.DataFrame(columns=columns)
neighborhoods_venues_sorted['Location'] = london_grouped['Location']

for ind in np.arange(london_grouped.shape[0]):
    neighborhoods_venues_sorted.iloc[ind, 1:] =␣
 ↪return_most_common_venues(london_grouped.iloc[ind, :], num_top_venues)

neighborhoods_venues_sorted.head()
```

[20]:
```
      Location 1st Most Common Venue 2nd Most Common Venue  \
0        Acton                   Pub  Gym / Fitness Center
1   Addiscombe                  Park                Bakery
2      Aldgate                 Hotel           Coffee Shop
3      Aldwych               Theater                   Pub
4     Alperton     Indian Restaurant           Supermarket

  3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
0              Bus Stop    Chinese Restaurant         Grocery Store
1         Grocery Store     Indian Restaurant          Tram Station
2  Gym / Fitness Center           Pizza Place          Cocktail Bar
3           Coffee Shop            Restaurant                 Hotel
4  Gym / Fitness Center           Bus Station      Asian Restaurant

  6th Most Common Venue 7th Most Common Venue 8th Most Common Venue  \
0     Athletics & Sports           Supermarket    Turkish Restaurant
1     Chinese Restaurant                  Café  Fast Food Restaurant
2     English Restaurant            Restaurant                   Pub
3     Italian Restaurant        Sandwich Place          Cocktail Bar
4     Falafel Restaurant   Empanada Restaurant    English Restaurant

  9th Most Common Venue 10th Most Common Venue
0           Coffee Shop   Fast Food Restaurant
1         Cosmetics Shop         Historic Site
2     Italian Restaurant    Japanese Restaurant
3      French Restaurant     Seafood Restaurant
4            Escape Room   Ethiopian Restaurant
```

[21]:
```
kclusters = 5

london_grouped_clustering = london_grouped.drop('Location', 1)

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

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

[21]:
```
array([3, 1, 3, 3, 2, 1, 3, 0, 2, 3])
```

[22]:
```
# add clustering labels
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

london_merged = london_data
```

```python
# merge london_grouped with london_data to add latitude/longitude for each␣
↪neighborhood
london_merged = london_merged.join(neighborhoods_venues_sorted.
↪set_index('Location'), on='Location')

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

[22]:

```
      Location                   London borough Post town Postcode district  \
1        Acton  Ealing, Hammersmith and Fulham    LONDON            W3, W4
3   Addiscombe                         Croydon   CROYDON               CR0
6      Aldgate                            City    LONDON               EC3
7      Aldwych                     Westminster    LONDON               WC2
8     Alperton                           Brent   WEMBLEY               HA0

    Latitude  Longitude  Distance from Center  Cluster Labels  \
1  51.508372  -0.274440             10.179297               3
3  51.380550  -0.072274             14.652051               1
6  51.513438  -0.077171              3.576673               3
7  51.513266  -0.117183              0.972334               3
8  51.539601  -0.298837             12.391598               2

  1st Most Common Venue 2nd Most Common Venue 3rd Most Common Venue  \
1                   Pub  Gym / Fitness Center              Bus Stop
3                  Park               Bakery         Grocery Store
6                 Hotel           Coffee Shop  Gym / Fitness Center
7               Theater                   Pub           Coffee Shop
8     Indian Restaurant           Supermarket  Gym / Fitness Center

  4th Most Common Venue 5th Most Common Venue 6th Most Common Venue  \
1    Chinese Restaurant         Grocery Store     Athletics & Sports
3     Indian Restaurant          Tram Station    Chinese Restaurant
6           Pizza Place          Cocktail Bar    English Restaurant
7            Restaurant                 Hotel    Italian Restaurant
8           Bus Station     Asian Restaurant    Falafel Restaurant

  7th Most Common Venue 8th Most Common Venue 9th Most Common Venue  \
1           Supermarket    Turkish Restaurant           Coffee Shop
3                  Café  Fast Food Restaurant        Cosmetics Shop
6            Restaurant                   Pub    Italian Restaurant
7        Sandwich Place          Cocktail Bar     French Restaurant
8   Empanada Restaurant    English Restaurant           Escape Room

  10th Most Common Venue
1   Fast Food Restaurant
3          Historic Site
6    Japanese Restaurant
7     Seafood Restaurant
```

```
8   Ethiopian Restaurant
```

[23]:
```
london_merged = london_merged.dropna(axis=0)
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

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]


markers_colors = []
for lat, lon, poi, cluster in zip(london_merged['Latitude'],␣
 ↪london_merged['Longitude'], london_merged['Location'],␣
 ↪london_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[int(cluster)-1],
        fill=True,
        fill_color=rainbow[int(cluster)-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

[23]:
```
<folium.folium.Map at 0x156ea16d280>
```

We now find the most common venues as well as reduce the distance from center. Since the number of venues is huge we only consider those venues which are less than 10km away from the city center. We observe that cluseter 3 is closer to hotesl and movie theatersand therefore more likely to attract a fine-dining crowd. Therefore, we selct those locations for our restaurant that are in cluster 3 and less than 10 kms away from the center.

[24]:
```
london_merged.loc[london_merged['Cluster Labels'] == 3, london_merged.
 ↪columns[[1] + list(range(5, london_merged.shape[1]))]].head()
```

[24]:
```
                       London borough  Longitude  Distance from Center   \
1   Ealing, Hammersmith and Fulham  -0.274440             10.179297
6                             City  -0.077171              3.576673
7                      Westminster  -0.117183              0.972334
10                       Islington  -0.104290              3.428213
16                      Wandsworth  -0.149411              7.232908

    Cluster Labels 1st Most Common Venue 2nd Most Common Venue  \
1                3                   Pub  Gym / Fitness Center
6                3                 Hotel           Coffee Shop
```

```
7                  3              Theater                  Pub
10                 3                  Pub          Coffee Shop
16                 3          Coffee Shop                  Pub

   3rd Most Common Venue 4th Most Common Venue 5th Most Common Venue  \
1               Bus Stop     Chinese Restaurant         Grocery Store
6    Gym / Fitness Center           Pizza Place          Cocktail Bar
7            Coffee Shop             Restaurant                 Hotel
10                  Café     French Restaurant            Restaurant
16           Pizza Place    Italian Restaurant           Supermarket

   6th Most Common Venue 7th Most Common Venue 8th Most Common Venue  \
1      Athletics & Sports           Supermarket    Turkish Restaurant
6      English Restaurant            Restaurant                   Pub
7      Italian Restaurant         Sandwich Place         Cocktail Bar
10        Thai Restaurant    Italian Restaurant     Sushi Restaurant
16                Bakery                   Bar     Indian Restaurant

   9th Most Common Venue 10th Most Common Venue
1            Coffee Shop   Fast Food Restaurant
6      Italian Restaurant    Japanese Restaurant
7       French Restaurant     Seafood Restaurant
10      Korean Restaurant      Indian Restaurant
16         Sandwich Place             Steakhouse
```

[25]:
```python
london_merged.loc[london_merged['Cluster Labels'] == 0, london_merged.
↪columns[[1] + list(range(5, london_merged.shape[1]))]].head()
```

[25]:
```
   London borough  Longitude  Distance from Center  Cluster Labels  \
12      Islington  -0.132381              6.383041               0
26      Islington  -0.118345              3.704740               0
27     Wandsworth  -0.165547              4.728954               0
43  Tower Hamlets  -0.048080              6.122627               0
51      Greenwich   0.018833             10.747825               0

   1st Most Common Venue 2nd Most Common Venue 3rd Most Common Venue  \
12                   Pub         Grocery Store           Coffee Shop
26                  Café         Grocery Store               Brewery
27                   Pub              Bus Stop                  Park
43                  Café                   Pub       Harbor / Marina
51                   Pub                  Café         Grocery Store

   4th Most Common Venue 5th Most Common Venue 6th Most Common Venue  \
12     Italian Restaurant                  Café  Fast Food Restaurant
26            Coffee Shop  Ethiopian Restaurant         Train Station
27          Grocery Store       Thai Restaurant                  Café
43             Restaurant           Coffee Shop          Cocktail Bar
```

```
51                   Spa              Bus Stop     Electronics Store

   7th Most Common Venue     8th Most Common Venue 9th Most Common Venue  \
12     Indian Restaurant  Bike Rental / Bike Share     Seafood Restaurant
26  Caucasian Restaurant               Escape Room              Nightclub
27             Mini Golf                       Gym  Gym / Fitness Center
43                  Park            Farmers Market             Food Truck
51   Empanada Restaurant        English Restaurant            Escape Room

       10th Most Common Venue
12             Sandwich Place
26                   Bus Stop
27                     Bakery
43  Middle Eastern Restaurant
51       Ethiopian Restaurant
```

[26]:
```
restaurant_filtered = london_merged.loc[(london_merged['Distance from Center']
↪< 10)&(london_merged['Cluster Labels'] ==3)]
```

[27]:
```
restaurant_filtered.head()
```

[27]:
```
    Location London borough Post town Postcode district   Latitude  Longitude  \
6    Aldgate           City    LONDON               EC3  51.513438  -0.077171
7    Aldwych    Westminster    LONDON               WC2  51.513266  -0.117183
10     Angel      Islington    LONDON           EC1, N1  51.534676  -0.104290
16    Balham     Wandsworth    LONDON              SW12  51.443989  -0.149411
17  Bankside      Southwark    LONDON               SE1  51.508137  -0.095184

    Distance from Center  Cluster Labels 1st Most Common Venue  \
6               3.576673               3                 Hotel
7               0.972334               3               Theater
10              3.428213               3                   Pub
16              7.232908               3           Coffee Shop
17              2.267391               3           Coffee Shop

   2nd Most Common Venue 3rd Most Common Venue 4th Most Common Venue  \
6            Coffee Shop  Gym / Fitness Center           Pizza Place
7                    Pub           Coffee Shop            Restaurant
10           Coffee Shop                  Café     French Restaurant
16                   Pub           Pizza Place    Italian Restaurant
17                   Bar    Italian Restaurant    Seafood Restaurant

   5th Most Common Venue 6th Most Common Venue 7th Most Common Venue  \
6           Cocktail Bar    English Restaurant            Restaurant
7                  Hotel    Italian Restaurant        Sandwich Place
10            Restaurant       Thai Restaurant    Italian Restaurant
16           Supermarket                Bakery                   Bar
```

```
17                   Café             Art Museum                Bakery

   8th Most Common Venue 9th Most Common Venue 10th Most Common Venue
6                    Pub    Italian Restaurant    Japanese Restaurant
7           Cocktail Bar     French Restaurant     Seafood Restaurant
10      Sushi Restaurant     Korean Restaurant      Indian Restaurant
16     Indian Restaurant        Sandwich Place             Steakhouse
17              Wine Bar                   Pub  Portuguese Restaurant
```

[28]:

```python
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

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]


markers_colors = []
for lat, lon, poi, cluster in zip(restaurant_filtered['Latitude'],␣
 ↪restaurant_filtered['Longitude'], restaurant_filtered['Location'],␣
 ↪restaurant_filtered['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[int(cluster)-1],
        fill=True,
        fill_color=rainbow[int(cluster)-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

[28]: <folium.folium.Map at 0x156ec641430>

1.5 Results and Discussion

Our analysis shows that there are quite a few areas in London near the city center i.e. Trafalgar Square, which are in localities that are around hotels, theaters and tourist attractions and therefore capable of attracting a huge number of people looking to fine-dine. Furthermore, we find that most of the areas in cluster 3 which match the kind of neighborhoods we are looking for also lack other Asian fine-dining restaurants therefore, these are ideal to open branches of our restaurant. The map above shows these locations.

We approached our problem by first gathering data about different neighborhoods and how far they are from the center of London. Then we filtered and cleaned our data and used the Foursquare API to collect data about venues and venue categories. We then use one hot encoding to get the dummies for each venue categories and finally we use kmeans clustering to cluster neighborhoods and find the most suitable areas for opening our locations and as mentioned above cluster 3 best suits our preferences. While these might not neccessary be the best locations for opening branches of our fine-dine restaurant, they surely exhibit potential. More nuanced analysis is needed to find further shortlist locations from those mentioned in the above map.

1.6 Conclusion

The following project tried to find the best locations to establis branches for our fine-dine restaurant chain. We settled on cluster 3 neighborhoods that are less than 10km away from the city center. This should be the starting point for deciding the suitable locations for our restaurant and further analysis is needed to better shortlist locations.