Introduction: As a part of my online course, I have to work on a project, where I have to apply my Data Science and Machine Learning knowledge and skills in solving a problem. The problem I have chosen is to explore and compare the neighborhoods of London city and use this information in identifying a best location to open an Asian restaurant in London, where the asian population is highest.

For this project I am using the K-means Clustering algorithm to cluster the neighborhoods. This is one of the simplest algorithms which is vastly used for clustering in many data science applications, especially useful if you need to quickly discover insights from the unlabeled data.

Data: For this solution, I need the data of all the 32 boroughs and the neighborhoods of these 32 boroughs. I need the names of the boroughs, the area names of each borough, the geo-coordinates, the venues in each neighborhood and so on. Using the web scraping libraries of Python, I downloaded the boroughs and the geo-coordinates data from Wikipedia pages into a structured dataframe.

To get the population of London boroughs, I have used the dataset available here. From these datasets, the Newham borough is having the highest asian population, so I am selecting this borough as my prime target location for the restaurant, and will use the neighborhood of this borough for my clustering solution.

Import the necessary libraries

```
In [ ]: import requests
        import pandas as pd
        import numpy as np
        from bs4 import BeautifulSoup
        from urllib.request import urlopen
        import re
        from geopy.geocoders import Nominatim # convert an address into latitude and longitude values
```

Importing the required libraries

After performing the web scraping of the Wikipedia page, the data is retrieved and saved in a Pandas dataframe as below.

```
In [6]: df_boroughs.head()
```

Out[6]:

	Borough	Latitude	Longitude
0	Barking and Dagenham	51.5607	0.1557
1	Barnet	51.6252	-0.1517
2	Bexley	51.4549	0.1505
3	Brent	51.5588	-0.2817
4	Bromley	51.4039	0.0198

Pandas dataframe

Get the geo-coordinates of London city using the geopy library of python.

```
In [82]: from geopy.geocoders import Nominatim # convert an address into latitude and longitude values
         address = 'London, UK'
         geolocator = Nominatim()
         location = geolocator.geocode(address)
         latitude = location.latitude
         longitude = location.longitude
         print('The geograpical coordinate of London City are {}, {}.'.format(latitude, longitude))

         The geograpical coordinate of London City are 51.5073219, -0.1276474.
```

Create a map of London city with all the boroughs superimposed on it.

```
In [86]: import folium

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

         # add markers to map
         for lat, lng, borough in zip(df_boroughs['Latitude'], df_boroughs['Longitude'], df_boroughs['Borough']):
             label = '{}'.format(borough)
             label = folium.Popup(label, parse_html=True)
             folium.CircleMarker(
                 [lat, lng],
                 radius=5,
                 popup=label,
                 color='blue',
                 fill=True,
                 fill_color='#3186cc',
         fill_opacity=0.7).add_to(map_london)
         map_london
```

Using folium create a map of London

London Map with all the 32 boroughs superimposed

Extract the geo-coordinates data of all the neighborhoods in Newham borough using the Python web-scraping libraries and save the data into a dataframe.

In [42]:
```
Newham_borough
```

Out[42]:

	Borough	Neighborhood	Latitude	Longitude
0	Newham	Beckton	51.514642	0.067375
1	Newham	Canning Town	51.515396	0.024169
2	Newham	Custom House	51.508133	0.028171
3	Newham	East Ham	51.532867	0.053782
4	Newham	Forest Gate	51.551339	0.025765
5	Newham	Little Ilford	51.550584	0.069004
6	Newham	Manor Park	51.550838	0.054591
7	Newham	Maryland	51.546294	0.005349
8	Newham	North Woolwich	51.496671	0.066561
9	Newham	Plaistow	51.524382	0.024568
10	Newham	Silvertown	51.497175	0.037769
11	Newham	Stratford	51.542847	-0.003456
12	Newham	Upton Park	51.535165	0.025046
13	Newham	West Ham	51.535165	0.025046

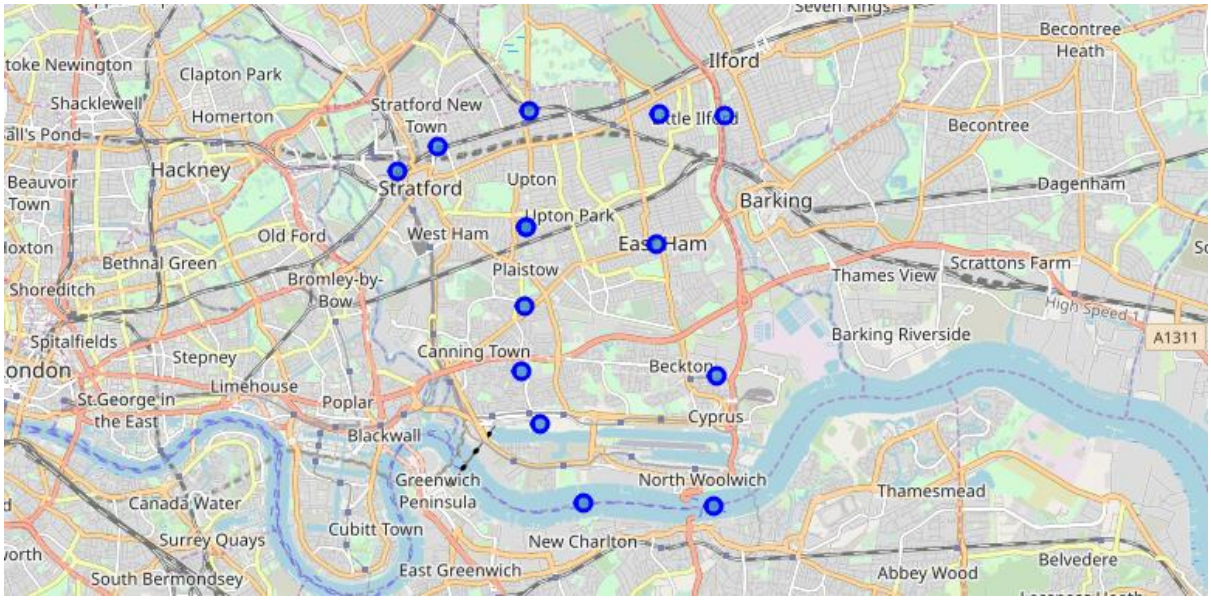
Neighborhoods of Newham borough and their geo-coordinates data

Visualize all the neighborhoods of Newham borough in a map using Folium.

In [45]:
```
# create map of Newham using latitude and longitude values
map_Newham = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(Newham_borough['Latitude'], Newham_borough['Longitude'], Newham_borough['Neighborhood']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7).add_to(map_Newham)

map_Newham
```



Map of Newham with all the 13 neighborhoods marked on it

Explore the first neighborhood Beckton of Newham borough. We construct a GET url and send this as a request to the Foursquare API to search and explore the neighborhood to get the venues of that neighborhood and the venue categories.

Note: To make calls to the Foursquare API, you need to register and get your credentials to use here as client id and client secret.

```
In [52]: LIMIT = 100 # limit of number of venues returned by Foursquare API
radius = 500 # define radius
# create URL
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret=\
       {}&v={}&ll={},{}&radius={}&limit={}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    neighborhood_latitude,
    neighborhood_longitude,
    radius,
    LIMIT)
url # display URL

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

GET url request to read the data from Foursquare data location provider

Once we get the json format results from the GET request URL of Foursquare API, we will structure the data into a dataframe

which contains the Venue name, Venue category and lat and long of that venue.

```
In [55]: results = requests.get(url).json()

         venues = results['response']['groups'][0]['items']
         nearby_venues = json_normalize(venues) # flatten JSON

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

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

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

Out[55]:

	name	categories	lat	lng
0	Lituanica	Grocery Store	51.516442	0.062927
1	Home Bargains	Discount Store	51.517190	0.062754
2	Premier Inn London Beckton	Hotel	51.515017	0.060978
3	Dreams Beckton	Furniture / Home Store	51.516101	0.063028
4	Beckton DLR Station	Light Rail Station	51.514365	0.061460

Venue data of first neighborhood Beckton in Newham borough

Create a function, which takes all the neighborhoods names, their geo-coordinates data and returns all the venues of each neighborhood and also their lat and long data along with the venue categories.

```
In [106]: Newham_venues = getNearbyVenues(names=Newham_borough['Neighborhood'],
                                          latitudes=Newham_borough['Latitude'],
                                          longitudes=Newham_borough['Longitude']
                                         )
```

```
In [98]: Newham_venues.shape
```

Out[98]: (208, 7)

```
In [99]: Newham_venues.head()
```

Out[99]:

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Beckton	51.514642	0.067375	Lituanica	51.516442	0.062927	Grocery Store
1	Beckton	51.514642	0.067375	Home Bargains	51.517190	0.062754	Discount Store
2	Beckton	51.514642	0.067375	Premier Inn London Beckton	51.515017	0.060978	Hotel
3	Beckton	51.514642	0.067375	Dreams Beckton	51.516101	0.063028	Furniture / Home Store
4	Beckton	51.514642	0.067375	Beckton DLR Station	51.514365	0.061460	Light Rail Station

Dataframe with all the venues data of each neighborhood in Newham borough

Using One Hot encoding on the above Newham_venues dataframe, create another dataframe and then group all the neighborhoods based on the frequency of occurrence of each venue category.

In [65]:
```
# one hot encoding
Newham_onehot = pd.get_dummies(Newham_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
Newham_onehot['Neighborhood'] = Newham_venues['Neighborhood']

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

Newham_onehot.head()
```

Out[65]:

	Neighborhood	Accessories Store	American Restaurant	Art Gallery	Asian Restaurant	Bagel Shop	Bakery	Bar	Boat or Ferry	Bookstore	...	Tapas Restaurant	Tennis Court	Thai Restaurant	Theater	Toy / Game Store	Train Station	T
0	Beckton	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
1	Beckton	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
2	Beckton	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
3	Beckton	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
4	Beckton	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	

In [107]:
```
Newham_grouped = Newham_onehot.groupby('Neighborhood').mean().reset_index()
Newham_grouped.head()
```

Out[107]:

	Neighborhood	Accessories Store	American Restaurant	Art Gallery	Asian Restaurant	Bagel Shop	Bakery	Bar	Boat or Ferry	Bookstore	...	Tapas Restaurant	Tennis Court	Thai Restaurant
0	Beckton	0.0	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.0	0.00	...	0.000000	0.00	0.0
1	Canning Town	0.0	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.0	0.00	...	0.000000	0.25	0.0
2	Custom House	0.0	0.038462	0.0	0.0	0.038462	0.038462	0.0	0.0	0.00	...	0.038462	0.00	0.0
3	East Ham	0.0	0.000000	0.0	0.0	0.000000	0.050000	0.0	0.0	0.05	...	0.000000	0.00	0.0
4	Forest Gate	0.0	0.000000	0.0	0.0	0.000000	0.111111	0.0	0.0	0.00	...	0.000000	0.00	0.0

Get the top 10 venues in each neighborhood

In [71]:
```
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]
```

Function that returns most common venues

In [109]:
```
num_top_venues = 10
indicators = ['st', 'nd', 'rd']

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

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

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

Top 10 venues that are most common in the neighborhoods

```
In [110]: neighborhoods_venues_sorted.head()
```

Out[110]:

	Neighborhood	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 Most Common Venue	10th Most Common Venue
0	Beckton	Hotel	Furniture / Home Store	Clothing Store	Pub	Discount Store	Light Rail Station	Shopping Plaza	Grocery Store	General Entertainment	Eastern European Restaurant
1	Canning Town	Convenience Store	Tennis Court	Gas Station	Park	Greek Restaurant	Go Kart Track	Creperie	Dance Studio	Department Store	Dessert Shop
2	Custom House	Hotel	Pub	Wine Bar	English Restaurant	Light Rail Station	Japanese Restaurant	Italian Restaurant	Gym / Fitness Center	Convenience Store	Salad Place
3	East Ham	Fast Food Restaurant	Clothing Store	Park	Sporting Goods Shop	Pub	Sandwich Place	Café	Chinese Restaurant	Shopping Mall	Grocery Store

Apply the K-means clustering on the Newham_grouped dataframe.

```
In [73]: # set number of clusters
kclusters = 5
Newham_grouped_clustering = Newham_grouped.drop('Neighborhood', 1)

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

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

```
Out[73]: array([1, 0, 1, 1, 1, 3, 4, 1, 1, 0], dtype=int32)
```

```
In [114]: kmeans
```

```
Out[114]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=5, n_init=10, n_jobs=1, precompute_distances='auto',
       random_state=0, tol=0.0001, verbose=0)
```

Create a new dataframe , that contains the Cluster and the top 10 most common venues.

```
In [74]: Newham_merged = Newham_borough
# add clustering labels
Newham_merged['Cluster Labels'] = kmeans.labels_

# merge Neighborhoods dataframe with Newham borough dataframe to add latitude/longitude for each neighborhood
Newham_merged = Newham_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

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

Out[74]:

	Borough	Neighborhood	Latitude	Longitude	Cluster Labels	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 Most Common Venue	1
0	Newham	Beckton	51.514642	0.067375	1	Hotel	Furniture / Home Store	Clothing Store	Pub	Discount Store	Light Rail Station	Shopping Plaza	Grocery Store	General Entertainment	
1	Newham	Canning Town	51.515396	0.024169	0	Convenience Store	Tennis Court	Gas Station	Park	Greek Restaurant	Go Kart Track	Creperie	Dance Studio	Department Store	
2	Newham	Custom House	51.508133	0.028171	1	Hotel	Pub	Wine Bar	English Restaurant	Light Rail Station	Japanese Restaurant	Italian Restaurant	Gym / Fitness Center	Convenience Store	

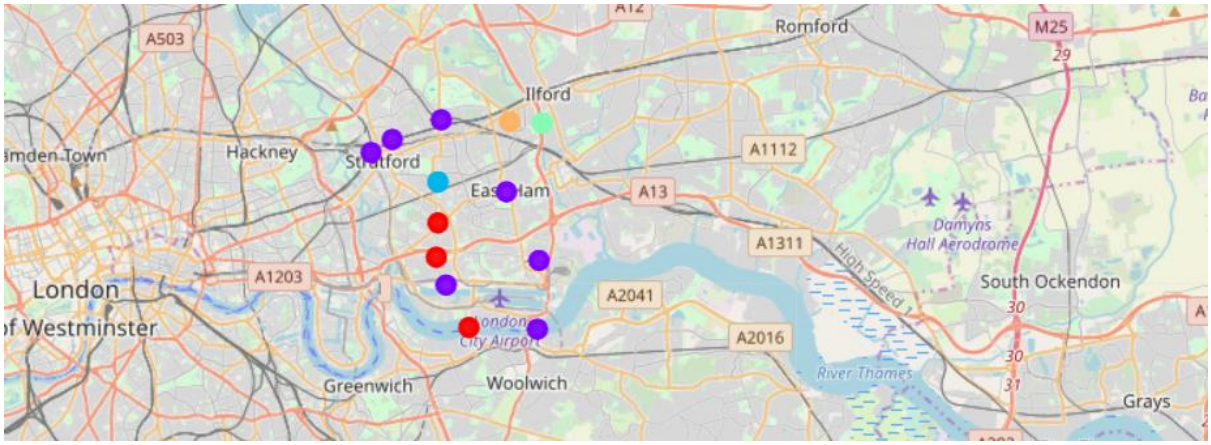
Visualize the Clusters on the map.

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

# set color scheme for the clusters
x = np.arange(kclusters)
y = [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]
```

In [115]:
```
# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(Newham_merged['Latitude'], Newham_merged['Longitude'], Newham_merged['Neighborhood'],
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```



Examine the clusters.

In [76]:
```
Newham_merged.loc[Newham_merged['Cluster Labels'] == 0, Newham_merged.columns[[1] + list(range(5, Newham_merged.shape[1]))]]
```

Out[76]:

	Neighborhood	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 Most Common Venue	10th Most Common Venue
1	Canning Town	Convenience Store	Tennis Court	Gas Station	Park	Greek Restaurant	Go Kart Track	Creperie	Dance Studio	Department Store	Dessert Shop
9	Plaistow	Park	Café	Gym / Fitness Center	Indian Restaurant	Grocery Store	Bus Stop	English Restaurant	Dance Studio	Department Store	Dessert Shop
10	Silvertown	Gym / Fitness Center	Theater	Construction & Landscaping	Museum	Café	Park	Paintball Field	Go Kart Track	General Entertainment	Discount Store

Examine the first cluster

```
In [77]: Newham_merged.loc[Newham_merged['Cluster Labels'] == 1, Newham_merged.columns[[1] + list(range(5, Newham_merged.shape[1]))]]
```

Out[77]:

	Neighborhood	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 Most Common Venue	10th Most Common Venue
0	Beckton	Hotel	Furniture / Home Store	Clothing Store	Pub	Discount Store	Light Rail Station	Shopping Plaza	Grocery Store	General Entertainment	Eastern European Restaurant
2	Custom House	Hotel	Pub	Wine Bar	English Restaurant	Light Rail Station	Japanese Restaurant	Italian Restaurant	Gym / Fitness Center	Convenience Store	Salad Place
3	East Ham	Fast Food Restaurant	Clothing Store	Park	Sporting Goods Shop	Pub	Sandwich Place	Café	Chinese Restaurant	Shopping Mall	Grocery Store
4	Forest Gate	Grocery Store	Train Station	Moving Target	Bakery	Italian Restaurant	Pub	Café	Fast Food Restaurant	Wine Bar	Electronics Store
7	Maryland	Hotel	Pub	Bus Stop	Grocery Store	Supermarket	Liquor Store	Portuguese Restaurant	Café	Sculpture Garden	Dance Studio
8	North Woolwich	Pier	History Museum	Clothing Store	Scenic Lookout	Gym / Fitness Center	Hotel	Italian Restaurant	Outdoor Sculpture	Pharmacy	Chinese Restaurant
11	Stratford	Pub	Sandwich Place	Café	Cosmetics Shop	Pizza Place	Bookstore	Burger Joint	Bar	Coffee Shop	Toy / Game Store

Examine the second cluster

Conclusion: After examining the above 5 clusters, we can recommend that Beckton,Custom House, Maryland, Eastham and Manor Park are the best neighborhoods in Newham borough, to open their asian restaurant. This is because in these areas, the most common venue visited by the public is the restaurants and as these areas have highest asian population, opening an asian restaurant would definitely be a good business idea.