

Description of the data and how it will be used to solve the problem

Data Acquisition and Data Cleaning:

1. Data Acquisition

The data acquired for the project is the combination data from three sources. The first source of the project uses [London Coronavirus Infected cases](#) in London.

The dataset contains following columns:

- **Borough:** Common Name of London Borough
- **Confirmed Cases:** No. of Coronavirus infected patients in a Borough

The second source of data is scraped from the Wikipedia page that contains the list of London Boroughs. The page contains additional information about the borough. The dataset contains the following columns:

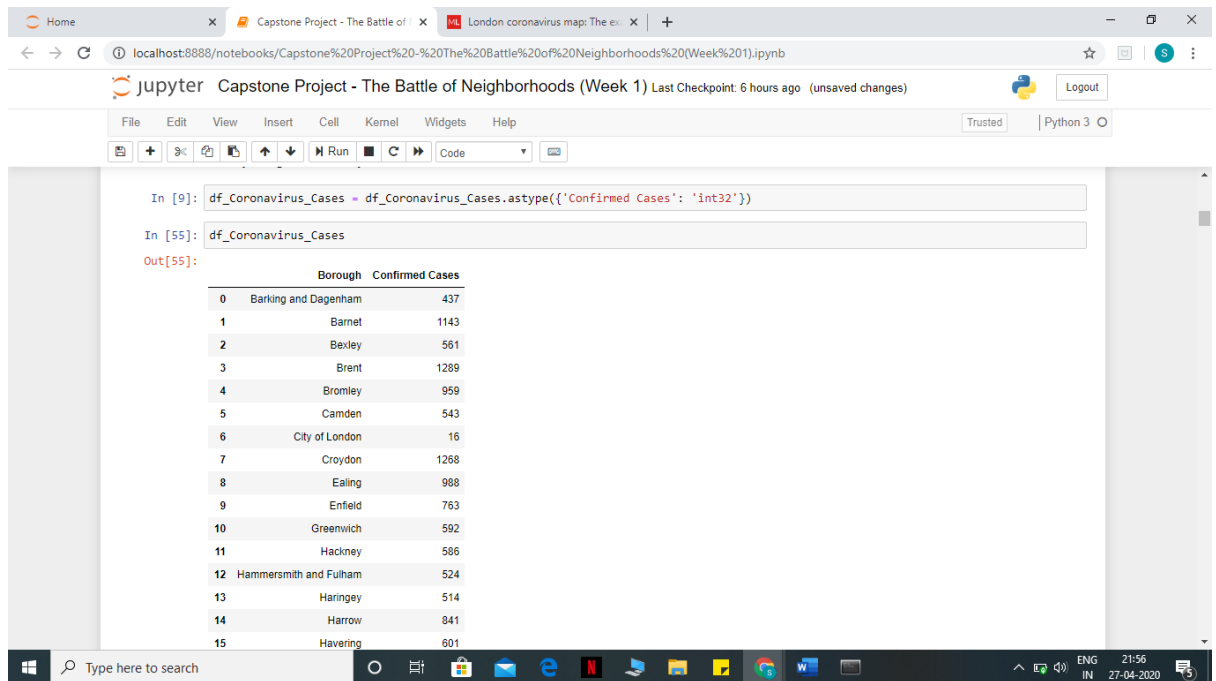
- **Borough:** The names of the 33 London boroughs.
- **Inner:** Categorizing the borough as an Inner London borough or an Outer London Borough.
- **Status:** Categorizing the borough as Royal, City or other borough.
- **Local authority:** The local authority assigned to the borough.
- **Political control:** The political party that control the borough.
- **Headquarters:** Headquarters of the Boroughs.
- **Area (sq mi):** Area of the borough in square miles.
- **Population (2013 est)[1]:** The population in the borough recorded during the year 2013.
- **Co-ordinates:** The latitude and longitude of the boroughs.
- **Nr. in map:** The number assigned to each borough to represent visually on a map.

The third data source is the list of Neighbourhoods in the Royal Borough of Kingston upon Thames as found on a Wikipedia page. This dataset is created from scratch using the list of neighbourhoods available on the site, the following are columns:

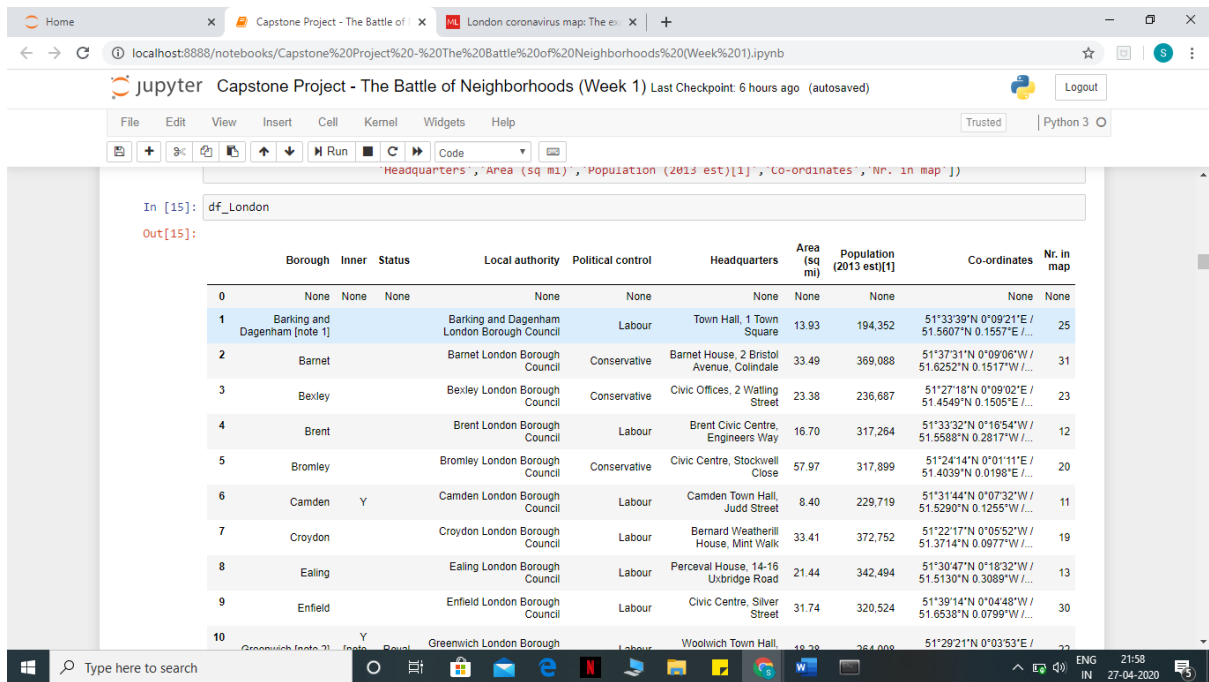
- **Neighbourhood:** Name of the neighbourhood in the Borough.
- **Borough:** Name of the Borough.
- **Latitude:** Latitude of the Borough.
- **Longitude:** Longitude of the Borough.

2. Data Cleaning

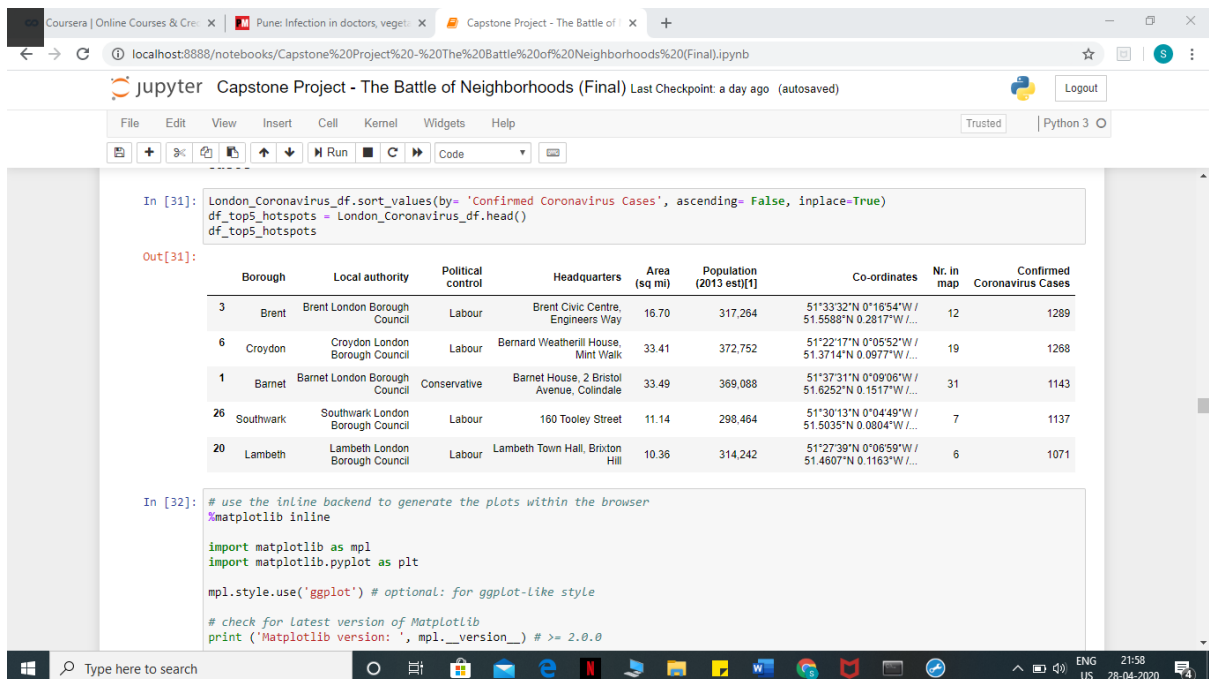
The data preparation for each of the three sources of data is done separately. Initially, we took the data from showing total number of confirmed infections in each borough as shown in the fig.



The second data is scraped from a Wikipedia page using the BeautifulSoup library in python. Using this library, we can extract the data in the tabular format as shown in the website. After the web scraping, string manipulation is required to get the names of the boroughs in the correct form. This is important because we will be merging the two datasets together using the Borough names.

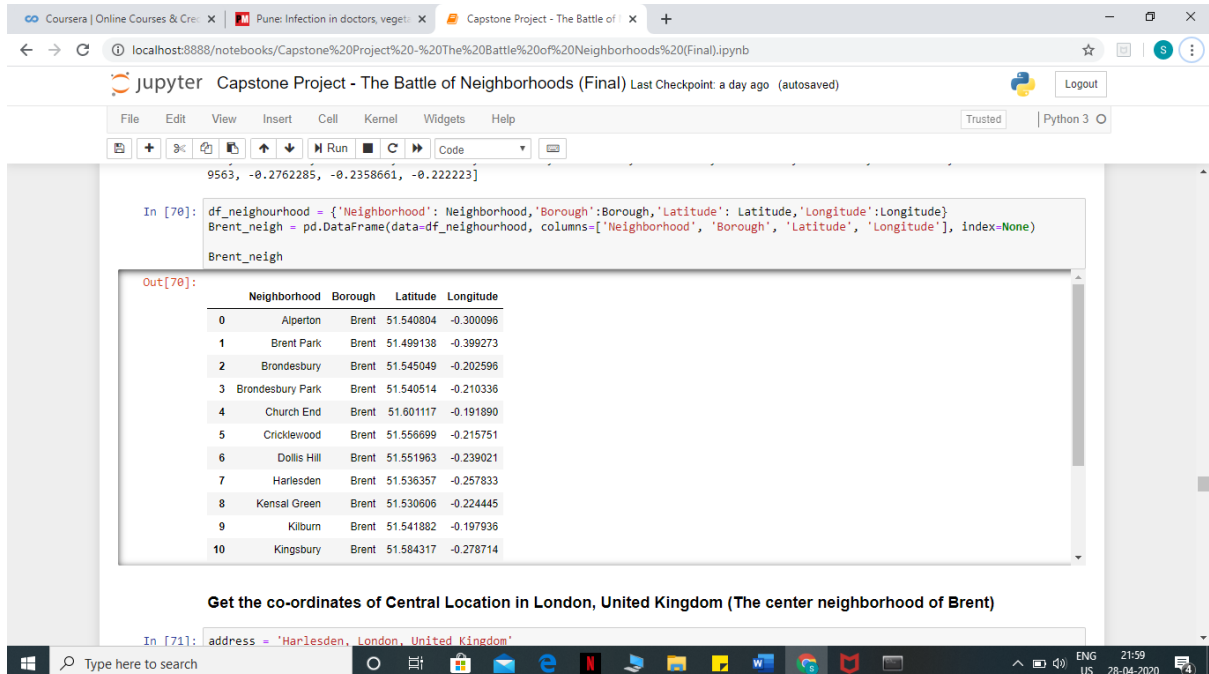


The two datasets are merged on the Borough names to form a new dataset that combines the necessary information in one dataset. The purpose of this dataset is to visualize the coronavirus infection rates in each borough and identify the worst effected borough with the maximum number of coronavirus cases recorded so far.



After visualizing the coronavirus cases in each borough, we can find the borough with the maximum number of cases and hence tag that borough as the worst

effected borough. The third source of data is acquired from the list of neighbourhoods in the worst affected borough on Wikipedia. This dataset is created from scratch, the pandas data frame is created with the names of the neighbourhoods and the name of the borough with the latitude and longitude are obtained using Google Maps API geocoding to get the final dataset.



The screenshot shows a Jupyter Notebook titled "Capstone Project - The Battle of Neighborhoods (Final)". The code in the cell defines a DataFrame with neighborhood data for Brent. The output displays a table with 11 rows of neighborhood data.

```
In [70]: df_neighbourhood = {'Neighborhood': Neighborhood, 'Borough':Borough, 'Latitude': Latitude, 'Longitude':Longitude}
Brent_neigh = pd.DataFrame(data=df_neighbourhood, columns=['Neighborhood', 'Borough', 'Latitude', 'Longitude'], index=None)
Brent_neigh
```

```
Out[70]:
```

	Neighborhood	Borough	Latitude	Longitude
0	Alperton	Brent	51.540804	-0.300096
1	Brent Park	Brent	51.499138	-0.399273
2	Brondesbury	Brent	51.545049	-0.202596
3	Brondesbury Park	Brent	51.540514	-0.210336
4	Church End	Brent	51.601117	-0.191890
5	Cricklewood	Brent	51.556699	-0.215751
6	Dollis Hill	Brent	51.551963	-0.239021
7	Harlesden	Brent	51.536357	-0.257833
8	Kensal Green	Brent	51.530606	-0.224445
9	Kilburn	Brent	51.541882	-0.197936
10	Kingsbury	Brent	51.584317	-0.278714

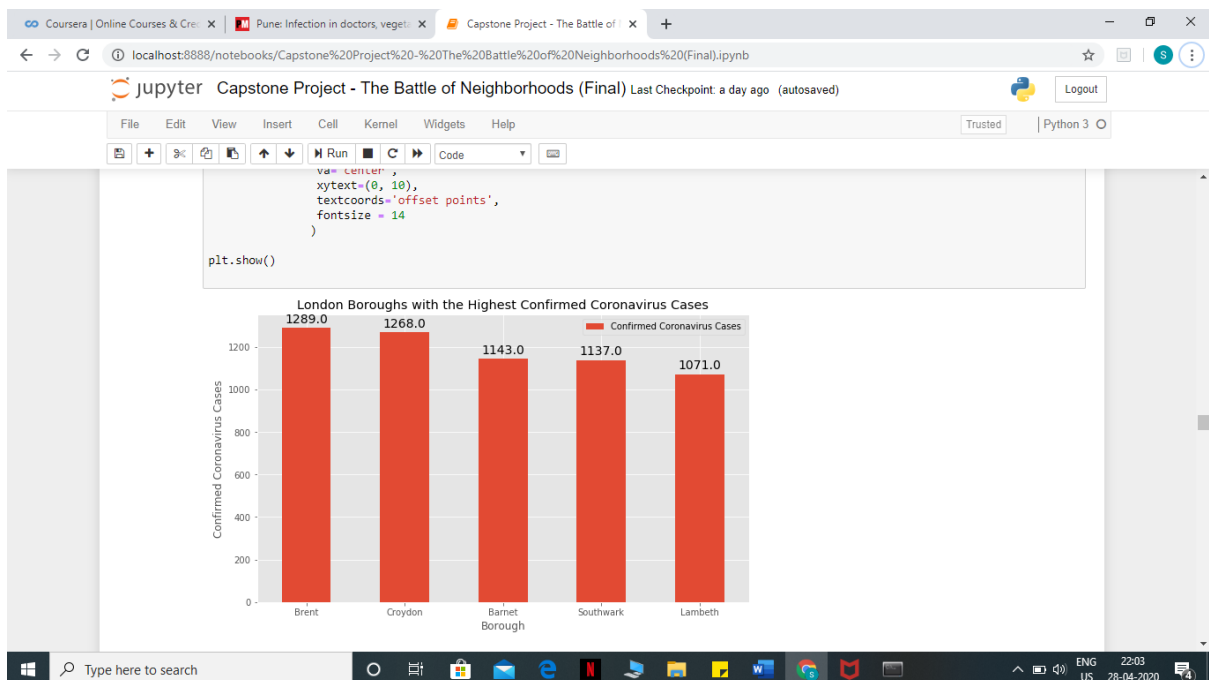
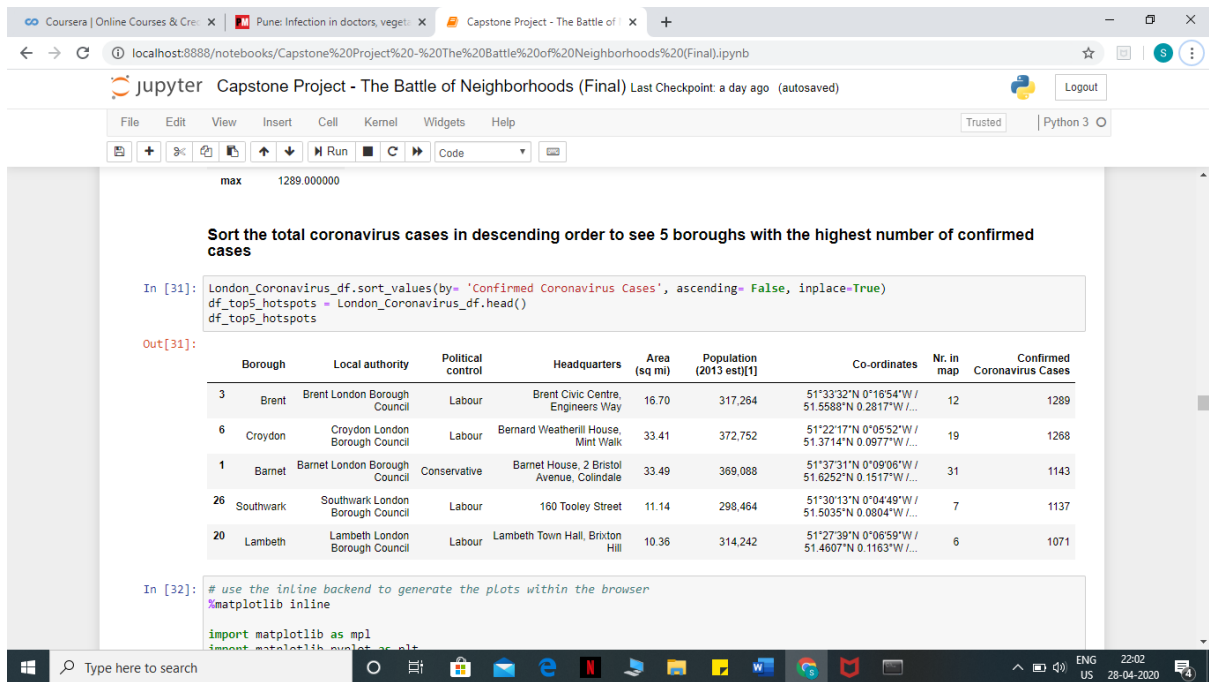
Get the co-ordinates of Central Location in London, United Kingdom (The center neighborhood of Brent)

```
In [71]: address = "Harlesden, London, United Kingdom"
```

Solution of the Problem:

This is a critical problem that happened globally so the only solution suggested by the experts to maintain social distancing and to exhibit lockdown in the cities which are worst affected by the pandemic. This lockdown prohibits the unnecessary travel of the people as well as going out of the homes for the non-essential reasons. Also, building of hospitals along with quarantine facilities and employing the healthcare workers so that critical patients effected by the disease can be better taken care of.

One way is to analyse the worst regions of London and practice strict Lockdown there. Along with that the regions have to be critically monitored for the spread.



People should not be allowed to go out without masks and only one person per house should be permitted to go out for grocery items. No unnecessary travels should be available for citizens of that region. The infected people should be home quarantined and asked to stay away from other members. Senior citizens and children below age of 5 should be given extra precautions of that region. The complete region should be declared a red zone and strict lockdown should be enforced in the region.

The government should observe serious measures to handle the situation by maintaining required number of hospitals and medical facilities for the severely and critically ill patients infected by the disease. Currently, whatever number of hospitals and health facilities are there should be dedicated mostly to take care of above Covid-19 patients until unless there is an emergency of some other serious illness.

```
Capstone Project - The Battle of Neighborhoods (Final) Last Checkpoint: a day ago (autosaved)

In [133]: print(Brent_venues.shape)
          Brent_venues.head()

Out[133]:
(3, 7)

          Neighborhood  Neighborhood Latitude  Neighborhood Longitude  Venue  Venue Latitude  Venue Longitude  Venue Category
0  Brondesbury         51.545049          -0.202596  Brondesbury medical center  51.543601      -0.200030      Hospital
1  Kilburn             51.541882          -0.197936  Brondesbury medical center  51.543601      -0.200030      Hospital
2  Willesden           51.546622          -0.235866  Willesden Centre for Health Care  51.542861      -0.235756      Hospital

In [134]: Brent_venues.groupby('Neighborhood').count()

Out[134]:
          Neighborhood Latitude  Neighborhood Longitude  Venue  Venue Latitude  Venue Longitude  Venue Category
Neighborhood
Brondesbury              1              1              1              1              1              1
Kilburn                  1              1              1              1              1              1
Willesden                1              1              1              1              1              1

In [135]: print('There are {} uniques categories.'.format(len(Brent_venues['Venue Category'].unique())))
          There are 1 uniques categories.

In [136]: def addToMap(df, color, existingMap):
          for lat, lng, local, venue, venueCat in zip(df['Venue Latitude'], df['Venue Longitude'], df['Neighborhood'], df['Venue'], df[
          label = '{} ({} - {})'.format(venue, venueCat, local)
          label = folium.Popup(label, parse_html=True)
```

```
Capstone Project - The Battle of Neighborhoods (Final) Last Checkpoint: a day ago (autosaved)

In [137]: Brent_onehot = pd.get_dummies(Brent_venues[['Venue Category']], prefix="", prefix_sep="")
          # add neighborhood column back to dataframe
          Brent_onehot['Neighborhood'] = Brent_venues['Neighborhood']

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

          Brent_onehot.head()

Out[138]:
          Neighborhood  Hospital
0  Brondesbury         1
1  Kilburn             1
2  Willesden           1

In [139]: Brent_grouped = Brent_onehot.groupby('Neighborhood').mean().reset_index()
          Brent_grouped

Out[139]:
          Neighborhood  Hospital
0  Brondesbury         1
1  Kilburn             1
2  Willesden           1

In [140]: Brent_grouped.columns
```

Also, where home quarantine for some people is not possible their institutional quarantine should be provided so as to curb down the further spread of the coronavirus infections. Therefore, certain places like malls and restaurants which are closed for now can temporarily be converted to quarantine places or temporary hospitals where large number of patients can be taken care.