Part 2: Data acquisition and preparation

To carry out this project two types of data are needed:

- A list of the neighborhoods in each borough, including geo location; and
- The venues data for each neighborhood.

The geolocation data for each neighborhood will be used to extract a list of venues in their vicinity. The venues data will be used to do a K-means clustering analysis. The objective is to cluster groups of neighborhoods based on their venue proximity and composition.

2.1 Neighborhood Data

The data for the neighborhoods for each borough can be extracted from Mexico City Data Portal (https://datos.cdmx.gob.mx/explore/dataset/coloniascdmx/table/). The city has an open database that includes the list of neighborhoods by borough along with the geolocation. The data can be consumed through an API or directly downloading the data in any of the file formats they offer (csv, json, or excel). For ease of processing, the direct download in csv will be used.

Since the full dataset includes more information than what is necessary for this analysis, the data will need to be filtered and adjusted to fit our needs. This includes renaming the columns, splitting the Geo location data from one column ("Geo Point") to separate columns for Latitude and Longitude, converting Latitude and Longitude data types from Object to float. Figure 1: shows the code to complete these steps.

Code

```
## Read csv file into a pandas dataframe
link='https://datos.cdmx.gob.mx/explore/dataset/coloniascdmx/download/?format=csv&timezone=America/Guatemala&lang=es&use_labels_for_header=true&csv_separator=%2C'
data=pd.read_csv(link)
data.head()

## We can see that the latitude and longitude data is stored in a column labeled "Geo Point". We will need to split this column in two to and store it in a temporary dataframe

Lat_Long=data['Geo Point'].str.split(',', expand=True)

## Now we will create a new dataframe using only the columns Colonia (Neighborhood), and Alcaldia (Burough)

Mexico_city=data[['ALCALDIA','COLONIA']].copy()

## Next we will add the latitude and longitude from the temporary dataframe and rename the columns.
Mexico_city['Latitude']=Lat_Long[0]
Mexico_city['Longitude']=Lat_Long[1]
Mexico_city.columns=['Borough','Neighborhood','Latitude','Longitude']

## We'll need to convert Latitude and Longitud to real numbers to avoid problems down the road
```

```
Mexico_city['Latitude'] = Mexico_city['Latitude'].astype(float)
Mexico_city['Longitude'] = Mexico_city['Longitude'].astype(float)

## Because we will be using only two boroughs from the whole dataset we will create dataframe
for each borough. MH for Miguel Hidalgo and BJ for Benito Juarez

MH_data = Mexico_city[Mexico_city['Borough'].isin(['MIGUEL HIDALGO'])].reset_index(drop=True)
BJ_data = Mexico_city[Mexico_city['Borough'].isin(['BENITO JUAREZ'])].reset_index(drop=True)
```

Figure 1: Code to extract, clean and prepare data.

The final step is to create two separate dataframes, one for Benito Juarez (`BJ_data`) and one for Miguel Hidalgo (`MH_data`), by filtering the "Borough" column in the Mexico City dataframe. Figures 2 and 3 show some basic information of these two dataframes, including the number of records for each borough.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 88 entries, 0 to 87
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Borough       88 non-null     object
 1   Neighborhood  88 non-null     object
 2   Latitude      88 non-null     float64
 3   Longitude     88 non-null     float64
dtypes: float64(2), object(2)
memory usage: 2.1+ KB
```

Figure 2: Benito Juarez dataframe information.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64 entries, 0 to 63
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Borough       64 non-null     object
 1   Neighborhood  64 non-null     object
 2   Latitude      64 non-null     float64
 3   Longitude     64 non-null     float64
dtypes: float64(2), object(2)
memory usage: 1.6+ KB
```

Figure 3: Miguel Hidalgo dataframe information.

Having the neighborhood data with the coordinates allows us to draw a map of each borough using Folium Python package. Figure 4 shows the map generated of Benito Juarez alongside the borough boundaries taken from Mexico City Data Portal. Figure 5 shows the map of the data extracted for Miguel Hidalgo.

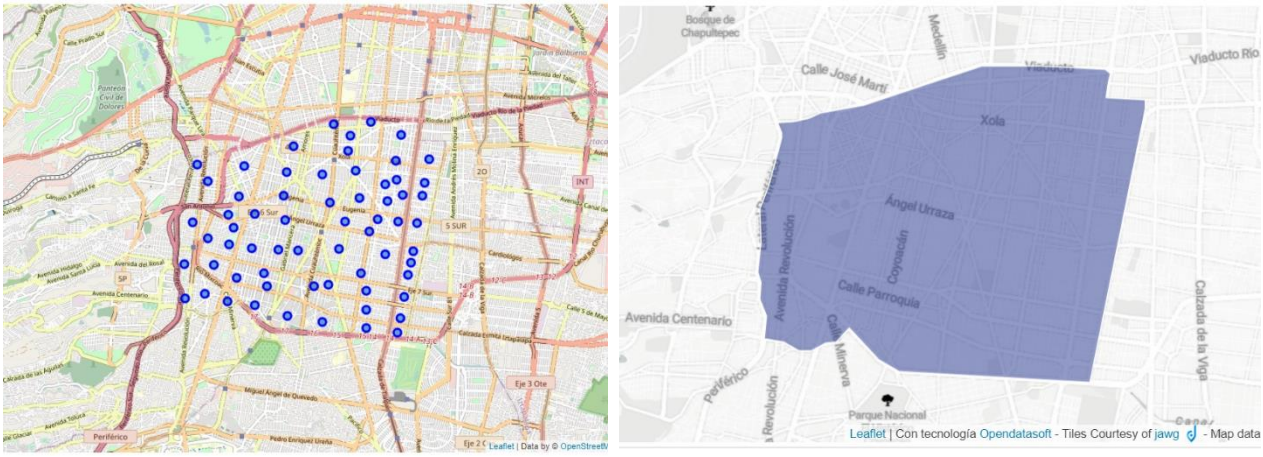

Figure 4: Map of Benito Juarez neighborhoods location and boundaries.

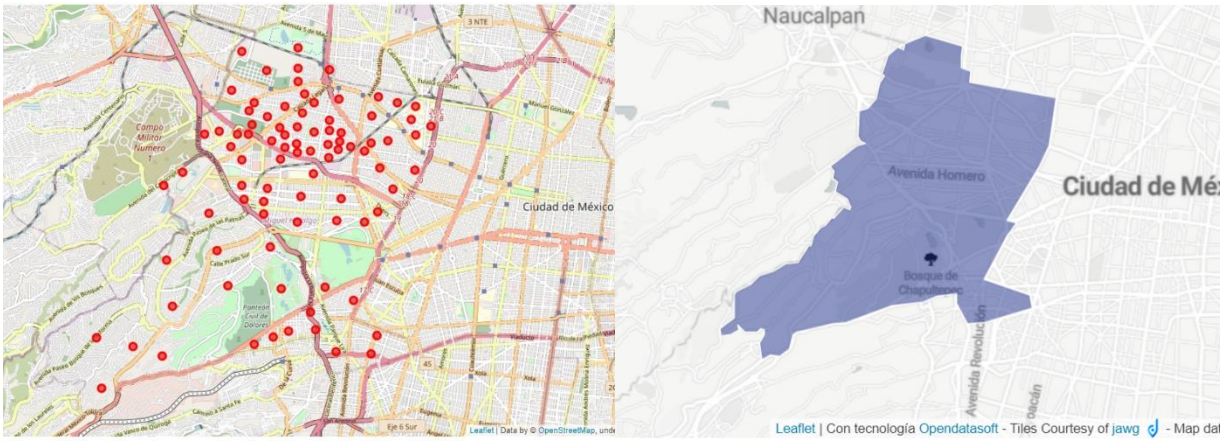

Figure 5: Map of Miguel Hidalgo neighborhoods location and boundaries.

2.2 Venues Data

We will extract the venues data from Foursquare. The Places API offers real-time access to Foursquare's global database of rich venue data. The venue data is obtained by passing the required parameters for each neighborhood to the Places API. We will create a dataframe for each borough to contain the extracted venue data.

Figure 6 shows the code used to create a function that takes the neighborhood names, latitudes and longitudes as inputs, creates the get request for Places API, and it returns a dataframe with the list of venues.

Code

```
def getNearbyVenues(names, latitudes, longitudes, radius=1000):

    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={}&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 = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

Figure 6: Function to Get nearby venues for each neighborhood.

After using this function for each Borough dataframe (`BJ_data` and `MH_data`), we get the following results:

- Benito Juarez: 6,400 venues with 225 venue categories.
- Miguel Hidalgo: 7,817 venues with 273 venues categories.