Introduction

For my first data science project, I am presenting a scenario for an entrepreneur who wants to open a Pakistani restaurant in Dubai.

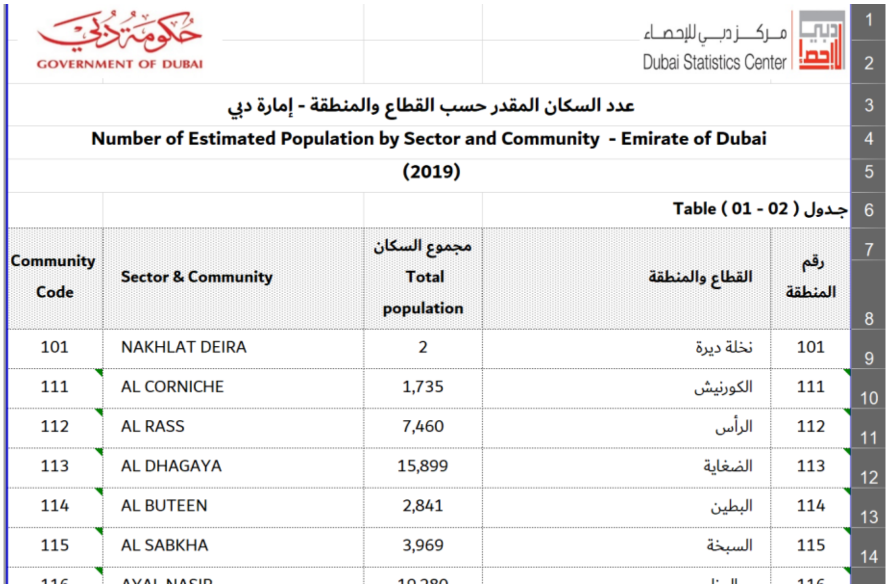
I've chosen Dubai first because I've spent a fair amount of time there, and I know how Dubai's localities are segmented. Second, being a melting pot of 200+ nationalities, it offers world-class dining opportunities for people visiting or residing in Dubai.

Due to its world-class status as a tourism destination, Dubai is also the best contender for the Foursquare platform, an essential requirement for this project.

Data

In our hypothetical scenario, we are looking for an optimal place wherein Dubai we can open our restaurant. For this purpose, we need to see how Dubai is segmented into different communities and neighborhoods.

To know about the communities in Dubai, I required a list of communities from a credible source. I came across the Dubai Statistics Center website, which is a government organization. Due to the OpenData policies of the Dubai government, they publish vital data on their website. I managed to found a list of communities in Dubai, along with population details circa 2019.



GOVERNMENT OF DUBAI

مـركـــز دبـــي للإحصـاء
Dubai Statistics Center

عدد السكان المقدر حسب القطاع والمنطقة - إمارة دبي

Number of Estimated Population by Sector and Community - Emirate of Dubai

(2019)

Table (01 - 02) جـدول

Community Code	Sector & Community	مجموع السكان Total population	القطاع والمنطقة	رقم المنطقة
101	NAKHLAT DEIRA	2	نخلة ديرة	101
111	AL CORNICHE	1,735	الكورنيش	111
112	AL RASS	7,460	الرأس	112
113	AL DHAGAYA	15,899	الضغاية	113
114	AL BUTEEN	2,841	البطين	114
115	AL SABKHA	3,969	السبخة	115
116	AYAL NASIR	10,280	[illegible]	116

The above data can be found here.

Another critical piece of information required was location data for each community, i.e., latitudes and longitudes. This piece of information was not available at the source, so we had to use Nominatim through Geocoder to retrieve the latitudes and longitude for each community.

Step 2.2: GeoPy

[14]:

```
# importing library

from geopy.geocoders import Nominatim
```

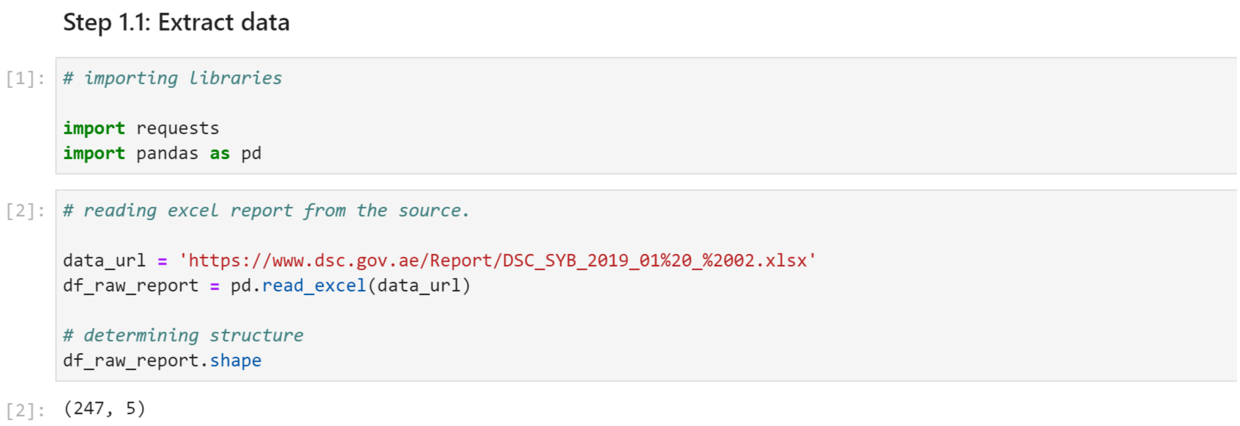
[15]:

```
# defining function to get coordinates based on community name

def get_latitude_longitude(community_name):
    # initialize your variable to None
    lat_lng_coords = None

    # loop until you get the coordinates
    #while(location is None):
    geolocator = Nominatim(user_agent="waqa5_ahm3d_capstone")
    location = geolocator.geocode('{}, Dubai, United Arab Emirates'.format(community_name))

    latitude = location.latitude
    longitude = location.longitude

    return latitude, longitude
```

Methodology

To approach this scenario, I followed the below methodology:

1. Extract all communities along with their population from the Dubai Statistics Center



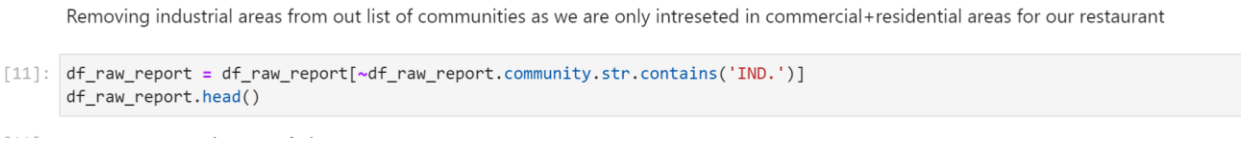
Step 1.1: Extract data

```
[1]: # importing libraries

     import requests
     import pandas as pd
```

```
[2]: # reading excel report from the source.

     data_url = 'https://www.dsc.gov.ae/Report/DSC_SYB_2019_01%20_%2002.xlsx'
     df_raw_report = pd.read_excel(data_url)

     # determining structure
     df_raw_report.shape
```

```
[2]: (247, 5)
```

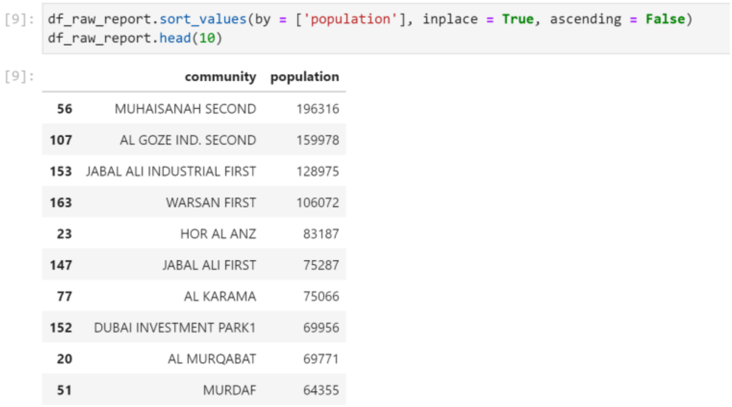
2. Clean and purify the data based on the following set of rules:
   a. Remove all industrial locations from the list, as we only want to open a restaurant in residential or commercial localities. These locations were identified based on overall knowledge about Dubai and indicated in the data source with a suffix like Industrial, IND, etc.



Removing industrial areas from out list of communities as we are only intreseted in commercial+residential areas for our restaurant

```
[11]: df_raw_report = df_raw_report[~df_raw_report.community.str.contains('IND.')]
      df_raw_report.head()
```

   b. Opening a restaurant only in a populated area, therefore, the cleansed list was sorted by population, and then only the top 100 communities were selected for this analysis.

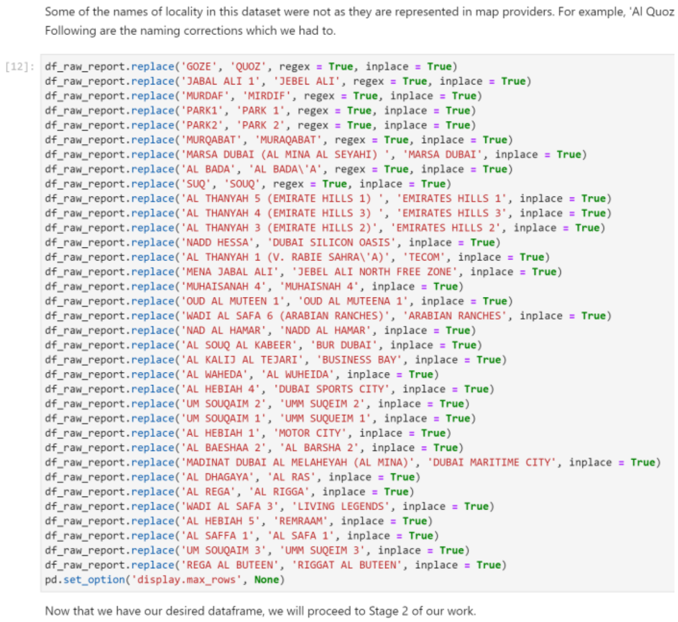


```
[9]: df_raw_report.sort_values(by = ['population'], inplace = True, ascending = False)
     df_raw_report.head(10)
```

[9]:

	community	population
56	MUHAISANAH SECOND	196316
107	AL GOZE IND. SECOND	159978
153	JABAL ALI INDUSTRIAL FIRST	128975
163	WARSAN FIRST	106072
23	HOR AL ANZ	83187
147	JABAL ALI FIRST	75287
77	AL KARAMA	75066
152	DUBAI INVESTMENT PARK1	69956
20	AL MURQABAT	69771
51	MURDAF	64355

3. Shortlisted community names were then verified with Nominatim using its website here. It was necessary because the primary data source's spelling was different from the listing in OpenStreetMaps. For example, Al Quoz is spelled Al Goze in the data source, due to direct translation from an Arabic dialect.



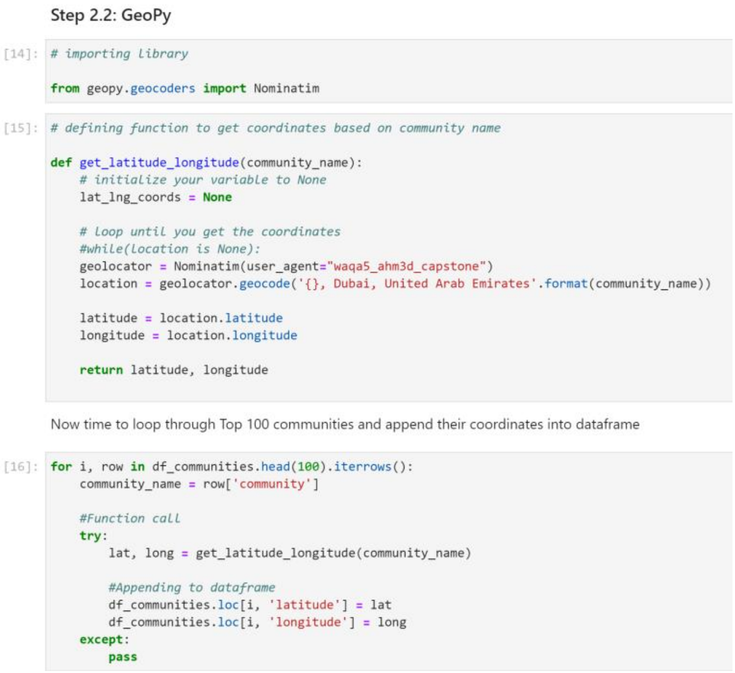
Some of the names of locality in this dataset were not as they are represented in map providers. For example, 'Al Quoz
Following are the naming corrections which we had to.

[12]:

```
df_raw_report.replace('GOZE', 'QUOZ', regex = True, inplace = True)
df_raw_report.replace('JABAL ALI 1', 'JEBEL ALI', regex = True, inplace = True)
df_raw_report.replace('MURDAF', 'MIRDIF', regex = True, inplace = True)
df_raw_report.replace('PARK1', 'PARK 1', regex = True, inplace = True)
df_raw_report.replace('PARK2', 'PARK 2', regex = True, inplace = True)
df_raw_report.replace('MURQABAT', 'MURAQABAT', regex = True, inplace = True)
df_raw_report.replace('MARSA DUBAI (AL MINA AL SEYAHI) ', 'MARSA DUBAI', inplace = True)
df_raw_report.replace('AL BADA', 'AL BADA\'A', regex = True, inplace = True)
df_raw_report.replace('SUQ', 'SOUQ', regex = True, inplace = True)
df_raw_report.replace('AL THANYAH 5 (EMIRATE HILLS 1) ', 'EMIRATES HILLS 1', inplace = True)
df_raw_report.replace('AL THANYAH 4 (EMIRATE HILLS 3) ', 'EMIRATES HILLS 3', inplace = True)
df_raw_report.replace('AL THANYAH 3 (EMIRATE HILLS 2)', 'EMIRATES HILLS 2', inplace = True)
df_raw_report.replace('NADD HESSA', 'DUBAI SILICON OASIS', inplace = True)
df_raw_report.replace('AL THANYAH 1 (V. RABIE SAHRA\'A)', 'TECOM', inplace = True)
df_raw_report.replace('MENA JABAL ALI', 'JEBEL ALI NORTH FREE ZONE', inplace = True)
df_raw_report.replace('MUHAISANAH 4', 'MUHAISNAH 4', inplace = True)
df_raw_report.replace('OUD AL MUTEEN 1', 'OUD AL MUTEENA 1', inplace = True)
df_raw_report.replace('WADI AL SAFA 6 (ARABIAN RANCHES)', 'ARABIAN RANCHES', inplace = True)
df_raw_report.replace('NAD AL HAMAR', 'NADD AL HAMAR', inplace = True)
df_raw_report.replace('AL SOUQ AL KABEER', 'BUR DUBAI', inplace = True)
df_raw_report.replace('AL KALIJ AL TEJARI', 'BUSINESS BAY', inplace = True)
df_raw_report.replace('AL WAHEDA', 'AL WUHEIDA', inplace = True)
df_raw_report.replace('AL HEBIAH 4', 'DUBAI SPORTS CITY', inplace = True)
df_raw_report.replace('UM SOUQAIM 2', 'UMM SUQEIM 2', inplace = True)
df_raw_report.replace('UM SOUQAIM 1', 'UMM SUQUEIM 1', inplace = True)
df_raw_report.replace('AL HEBIAH 1', 'MOTOR CITY', inplace = True)
df_raw_report.replace('AL BAESHAA 2', 'AL BARSHA 2', inplace = True)
df_raw_report.replace('MADINAT DUBAI AL MELAHEYAH (AL MINA)', 'DUBAI MARITIME CITY', inplace = True)
df_raw_report.replace('AL DHAGAYA', 'AL RAS', inplace = True)
df_raw_report.replace('AL REGA', 'AL RIGGA', inplace = True)
df_raw_report.replace('WADI AL SAFA 3', 'LIVING LEGENDS', inplace = True)
df_raw_report.replace('AL HEBIAH 5', 'REMRAAM', inplace = True)
df_raw_report.replace('AL SAFFA 1', 'AL SAFA 1', inplace = True)
df_raw_report.replace('UM SOUQAIM 3', 'UMM SUQEIM 3', inplace = True)
df_raw_report.replace('REGA AL BUTEEN', 'RIGGAT AL BUTEEN', inplace = True)
pd.set_option('display.max_rows', None)
```

Now that we have our desired dataframe, we will proceed to Stage 2 of our work.

4. After the naming corrections, extract the coordinates for each community correctly using Nominatim through geocoder libraries.



Step 2.2: GeoPy

[14]:

```
# importing library

from geopy.geocoders import Nominatim
```

[15]:

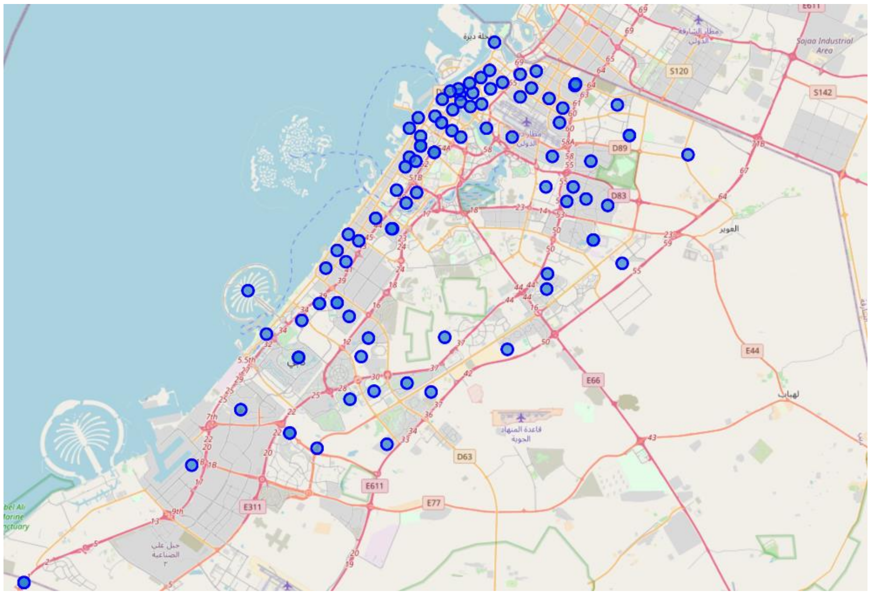
```
# defining function to get coordinates based on community name

def get_latitude_longitude(community_name):
    # initialize your variable to None
    lat_lng_coords = None

    # loop until you get the coordinates
    #while(location is None):
    geolocator = Nominatim(user_agent="waqa5_ahm3d_capstone")
    location = geolocator.geocode('{}, Dubai, United Arab Emirates'.format(community_name))

    latitude = location.latitude
    longitude = location.longitude

    return latitude, longitude
```

Now time to loop through Top 100 communities and append their coordinates into dataframe

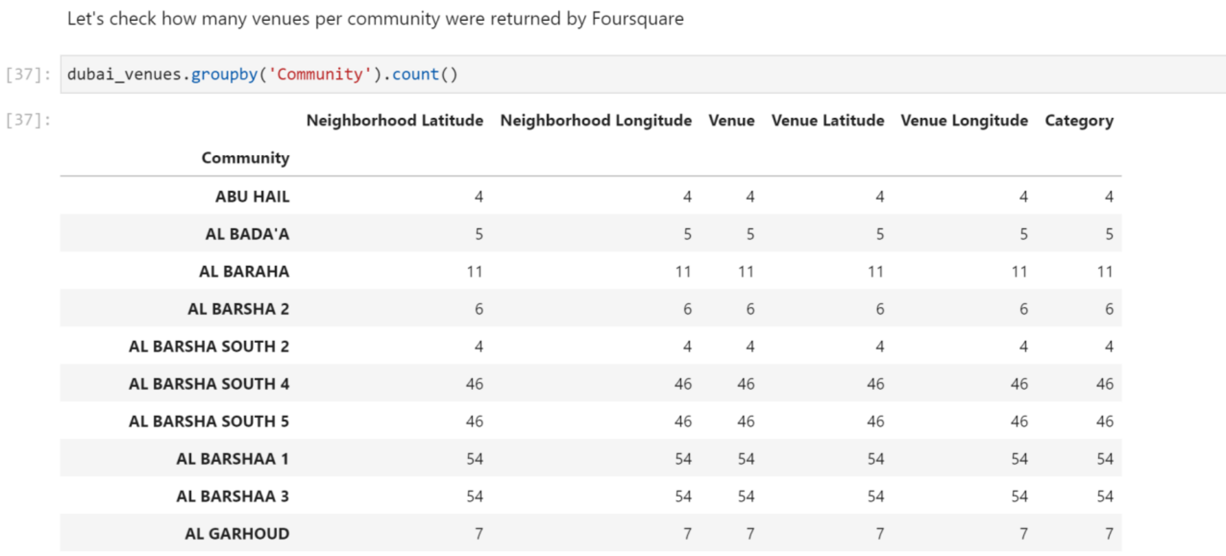
[16]:

```
for i, row in df_communities.head(100).iterrows():
    community_name = row['community']

    #Function call
    try:
        lat, long = get_latitude_longitude(community_name)

        #Appending to dataframe
        df_communities.loc[i, 'latitude'] = lat
        df_communities.loc[i, 'longitude'] = long
    except:
        pass
```

5. Visualize the extracted coordinates on the map using Folium to verify the information retrieved.



6. Access Foursquare APIs to retrieve a list of venues for each community. Perform exploratory analysis by visualizing:
    a. Number of venues per community



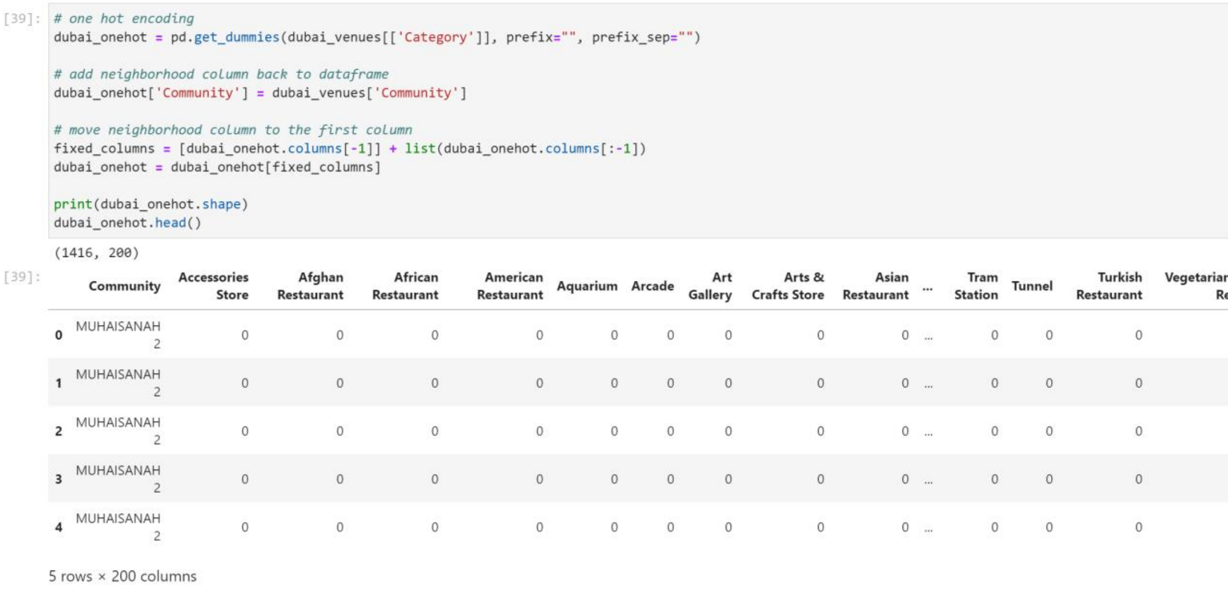
Let's check how many venues per community were returned by Foursquare

```
[37]: dubai_venues.groupby('Community').count()
```

[37]:

Community	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Category
ABU HAIL	4	4	4	4	4	4
AL BADA'A	5	5	5	5	5	5
AL BARAHA	11	11	11	11	11	11
AL BARSHA 2	6	6	6	6	6	6
AL BARSHA SOUTH 2	4	4	4	4	4	4
AL BARSHA SOUTH 4	46	46	46	46	46	46
AL BARSHA SOUTH 5	46	46	46	46	46	46
AL BARSHAA 1	54	54	54	54	54	54
AL BARSHAA 3	54	54	54	54	54	54
AL GARHOUD	7	7	7	7	7	7

    b. Summize each community based on type or category of venues available



```
[39]: # one hot encoding
      dubai_onehot = pd.get_dummies(dubai_venues[['Category']], prefix="", prefix_sep="")

      # add neighborhood column back to dataframe
      dubai_onehot['Community'] = dubai_venues['Community']

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

      print(dubai_onehot.shape)
      dubai_onehot.head()

      (1416, 200)
```

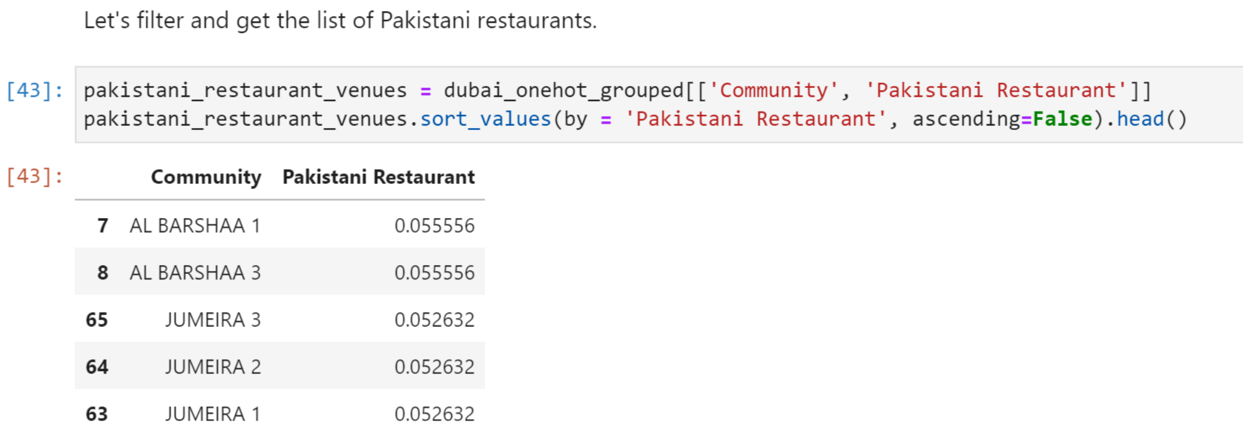
[39]:

	Community	Accessories Store	Afghan Restaurant	African Restaurant	American Restaurant	Aquarium	Arcade	Art Gallery	Arts & Crafts Store	Asian Restaurant	...	Tram Station	Tunnel	Turkish Restaurant	Vegetarian Re
0	MUHAISANAH 2	0	0	0	0	0	0	0	0	0	...	0	0	0	
1	MUHAISANAH 2	0	0	0	0	0	0	0	0	0	...	0	0	0	
2	MUHAISANAH 2	0	0	0	0	0	0	0	0	0	...	0	0	0	
3	MUHAISANAH 2	0	0	0	0	0	0	0	0	0	...	0	0	0	
4	MUHAISANAH 2	0	0	0	0	0	0	0	0	0	...	0	0	0	

5 rows × 200 columns

c. List down venues for the targeted category, i.e., a Pakistani restaurant, for each community.

7. Filter out noise from the data by excluding all venues belonging categories other than Pakistani restaurant.



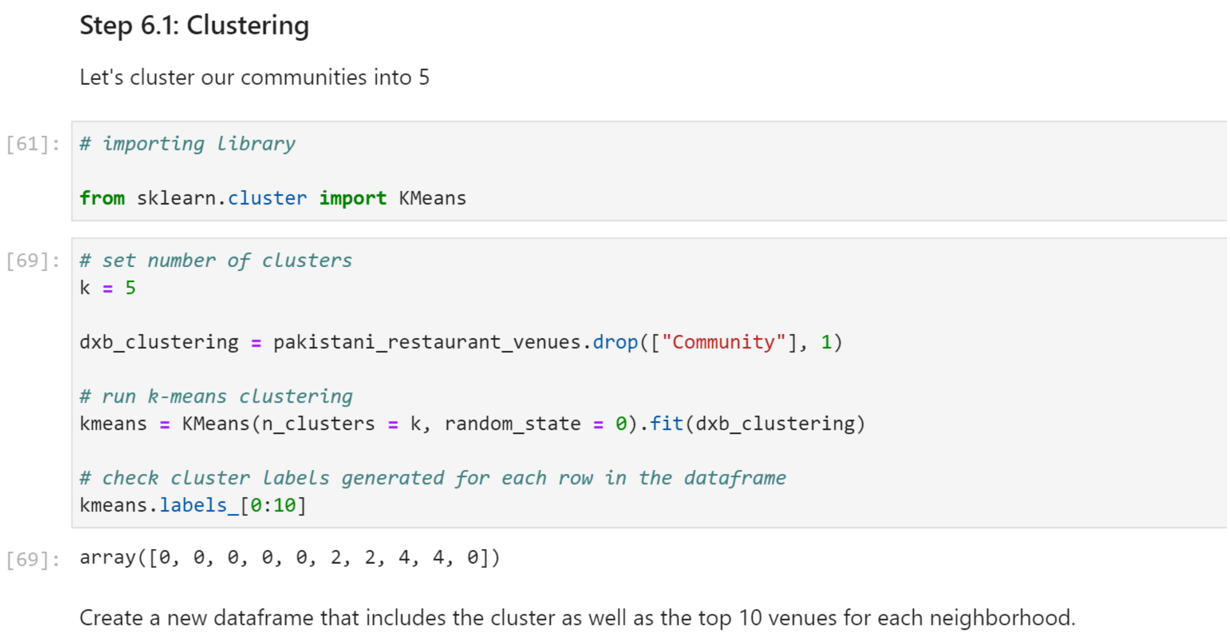
Let's filter and get the list of Pakistani restaurants.

```
[43]: pakistani_restaurant_venues = dubai_onehot_grouped[['Community', 'Pakistani Restaurant']]
      pakistani_restaurant_venues.sort_values(by = 'Pakistani Restaurant', ascending=False).head()
```

[43]:

	Community	Pakistani Restaurant
7	AL BARSHAA 1	0.055556
8	AL BARSHAA 3	0.055556
65	JUMEIRA 3	0.052632
64	JUMEIRA 2	0.052632
63	JUMEIRA 1	0.052632

8. Perform KMean clustering to identify and segment communities into clusters where Pakistani restaurants are listed and where not.



Step 6.1: Clustering

Let's cluster our communities into 5

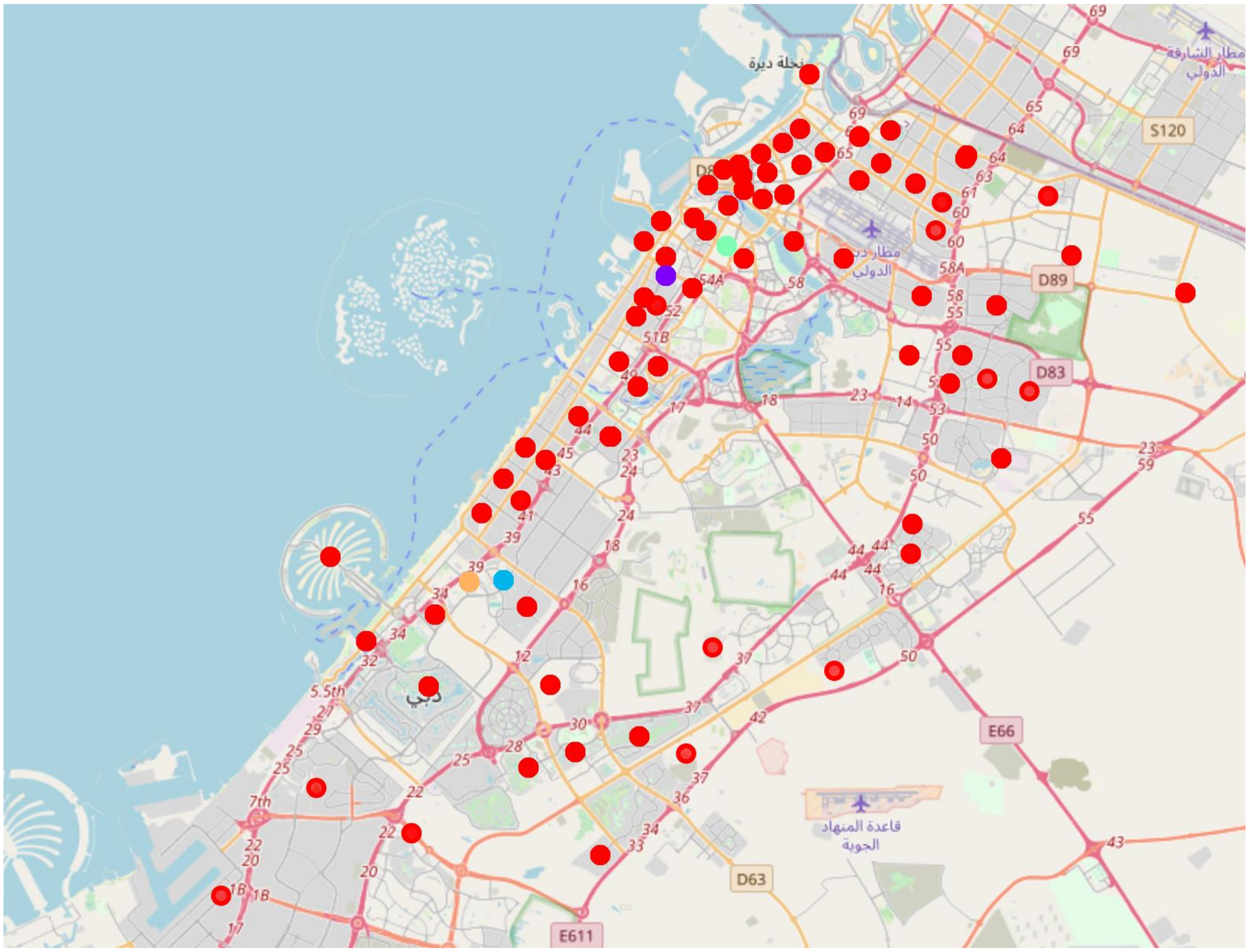
```
[61]: # importing library

      from sklearn.cluster import KMeans
```

```
[69]: # set number of clusters
      k = 5

      dxb_clustering = pakistani_restaurant_venues.drop(["Community"], 1)

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

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

```
[69]: array([0, 0, 0, 0, 0, 2, 2, 4, 4, 0])
```

Create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.

9. List down clusters to explore which communities are ideal for opening a business.



Findings

Based on our above approach, we found out that:

- Data wrangling and correction is a crucial aspect of achieving results. The key is to find the correct data source and transform it into a form that you can use to achieve your objectives.
- We found out that there were only eight restaurants in Foursquare categorized as Pakistani in 100 communities across Dubai (Possible causes define in below section)
- Because there were such low number of restaurants listed, the majority of the clusters were rendered empty

Shortcoming

Certain shortcomings were identified throughout data extracting and analysis, which impacted the results and decision making.

- The only information available to us was a list of the community name, and it's population. Although we all know that when trying to decide where to open a specific ethnic restaurant, we have to see those areas' demographics. In our case, if we had information regarding communities with a high population of Pakistanis, that would have made a significant impact on decision making.
- Another weakness in data was identified when we retrieved the list of venues from Foursquare. We have only managed to pull out eight places marked as 'Pakistani Restaurant' in our top 100 communities based on their total population. This low number of restaurants could be due to mis categorization or mere the fact that Pakistani restaurants are not extensively listed on Foursquare.

Conclusion

This scenario can easily be applied to any business you are planning to open in any part of the world. The approach will pre-dominantly remain the same. The only adjustment in the data selection and some parameter adjustment based on geographical location will be required.

So, Let it be about opening a grocery store, a pharmacy, or even a barbershop.

Utilize Foursquare API to retrieve venues and then perform exploratory and machine learning analysis to answer the question of where to start your business?