Data Science Capstone Project

Introduction/Business problem

Metro Manila, simply Manila, is the National Capital Region and the Philippines' prime tourist destination. Manila comprises 17 cities and municipalities, including the capital city, Manila City. Though it is the smallest region in the country, Metro Manila is the most populous of the twelve defined metropolitan areas in the Philippines and the 19th most populous in the world (Koop, 2021). Being the capital, Manila is considered to be the center of commerce, education, and entertainment of the country.

In this Capstone Project, I want to know the most common places available to Manila's people. After I determine the Top 3 Most Common Places, I will cluster the type of business and determine what kind of establishment is best to set up in a particular city. Lastly, I also want to know the most common cuisines people prefer to eat here in the Metro.

Description of the data

The different districts present in Metro Manila were obtained from the Philippines Statistics Authority records (PSA). Every city has been divided into separate smaller neighborhoods or "barangays," as termed in the Philippines. To make the data processing easier, Microsoft Excel is used to clean the PSA data while also retaining the different Metro Manila cities.

Each city's central location's latitude and longitude is requested from Google Cloud's Geocoding API. These coordinates will then be used as input in the Foursquare API, which is used to obtain the different venues present at each city within a 3000-meter radius relative to its coordinates.

Lastly, from the Foursquare API's response, the result will be parsed to obtain necessary values such as each venue's coordinates, name, venue sub-category, and main category. The primary type is how Foursquare classifies the venue. To name a few, this can be food, arts, and education. Venue sub-category is the specific category a venue is classified as, i.e., Japanese Restaurant for Food.

Methodology

This section will describe the data analysis and how I used the data to yield the results.

I cleaned the PSA data and loaded it to the Jupyter Notebook as a Pandas data frame. For this, I used the pandas read function. Once I loaded the data, I had to clean it further by renaming two (2) columns for Latitude and Longitude while removing an extra ghost column imported from the file. The table below shows the processed data frame named "NCR_data."

Table 1 NCR_data data frame

	City	2015 Population	Latitude	Longtitude
0	CITY OF MANILA	1780148.0	NaN	NaN
1	CITY OF MANDALUYONG	386276.0	NaN	NaN
2	CITY OF MARIKINA	450741.0	NaN	NaN
3	CITY OF PASIG	755300.0	NaN	NaN
4	QUEZON CITY	2936116.0	NaN	NaN
5	CITY OF SAN JUAN	122180.0	NaN	NaN
6	CALOOCAN CITY	1583978.0	NaN	NaN
7	CITY OF MALABON	365525.0	NaN	NaN
8	CITY OF NAVOTAS	249463.0	NaN	NaN
9	CITY OF VALENZUELA	620422.0	NaN	NaN
10	CITY OF LAS PIÑAS	588894.0	NaN	NaN
11	CITY OF MAKATI	582602.0	NaN	NaN
12	CITY OF MUNTINLUPA	504509.0	NaN	NaN
13	CITY OF PARAÑAQUE	665822.0	NaN	NaN
14	PASAY CITY	416522.0	NaN	NaN
15	PATEROS	63840.0	NaN	NaN
16	TAGUIG CITY	804915.0	NaN	NaN

As discussed, each city's central location's latitude and longitude is requested from Google Cloud's Geocoding API. The data has provisions for the Latitude and Longitude but had NaN values. Using the Geocoding API documentation, I looked up each city and directly added the values to the NCR_data table.

In [6]:
```python
#API Keys for Google Geocoders and Foursquare
geocoders_APIkey = ipython_config.geocoders_APIkey
foursquare_ID = ipython_config.foursquare_ID
foursquare_secret= ipython_config.foursquare_secret
foursquare_version = '20210315'
foursquare_limit = 100
```

In [7]:
```python
for ind, row in NCR_data.iterrows():
    address = str(NCR_data.at[ind, 'City']) + ", Philippines"
    parameters ={
    "key": geocoders_APIkey,
    "address": address
    }
    response = requests.get("https://maps.googleapis.com/maps/api/geocode/json?",params = parameters)

    data = json.loads(response.text)["results"][0]["geometry"]
    lat = data["location"]["lat"]
    lng = data["location"]["lng"]

    NCR_data.at[ind, 'Latitude'] = lat
    NCR_data.at[ind, 'Longtitude'] = lng
```

In [8]:
```python
NCR_data.head()
```

Out[8]:

	City	2015 Population	Latitude	Longtitude
0	CITY OF MANILA	1780148.0	14.599512	120.984219
1	CITY OF MANDALUYONG	386276.0	14.579444	121.035917
2	CITY OF MARIKINA	450741.0	14.650730	121.102855
3	CITY OF PASIG	755300.0	14.576377	121.085110
4	QUEZON CITY	2936116.0	14.676041	121.043700

Figure 1 Code used to collect coordinates and directly append to the main data frame

I used Folium to plot the data points, embedded on an interactive Map, to verify the collected coordinates.

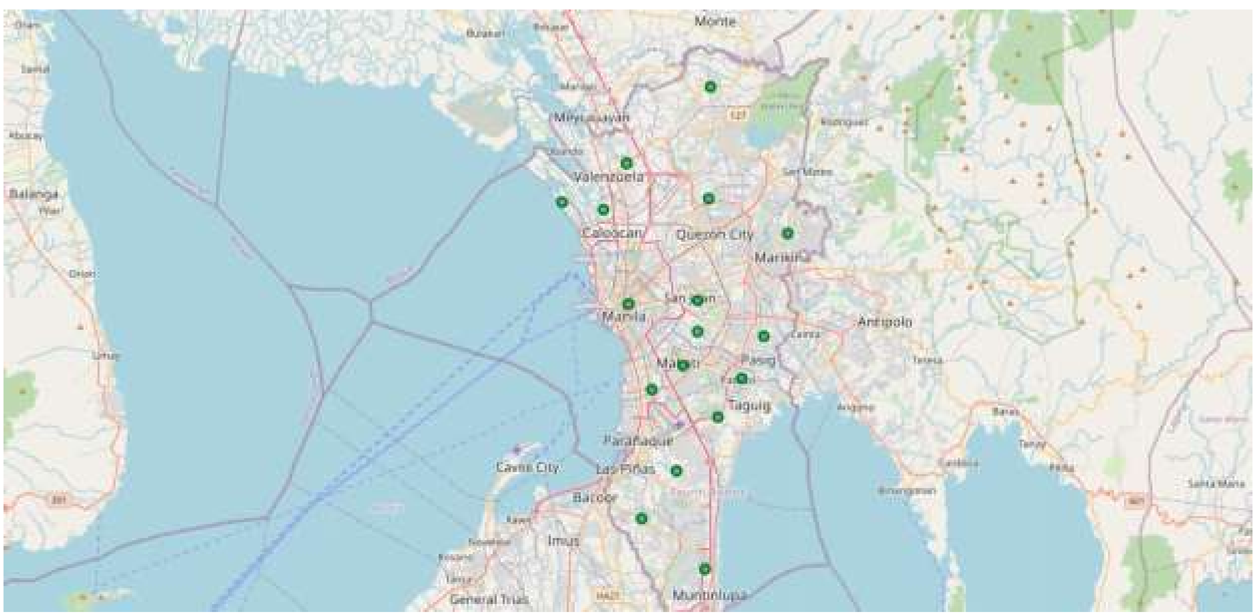


Figure 2 Plotted points in Folium map

Since the coordinates have been collected, I was able to proceed to use the Foursquare API. A search radius of 3000 meters was used to return 1625 data points from each city's center. It is important to note that Foursquare does not directly return the main category (named 'Short Category' in the table) a venue is classified under.

Table 2 NCR_venue Data Frame with data from Foursquare

	City	City Latitude	City Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	Short Category
0	CITY OF MANILA	14.599512	120.984219	Krispy Kreme	14.601195	120.982774	Donut Shop	food
1	CITY OF MANILA	14.599512	120.984219	BonChon Chicken	14.601194	120.982791	Fried Chicken Joint	food
2	CITY OF MANILA	14.599512	120.984219	98B	14.598836	120.979435	Public Art	arts_entertainment
3	CITY OF MANILA	14.599512	120.984219	The Den	14.598827	120.979450	Coffee Shop	food
4	CITY OF MANILA	14.599512	120.984219	Minor Basilica of St. Lorenzo Ruiz of Manila (...	14.599935	120.974646	Church	building

There are two (2) main reasons as to why I added the Short Category column. The first reason is I will be able to determine the top places available in Metro Manila. Coincidentally, this method is more efficient in clustering cities with the same kinds of places/businesses. Secondly, adding the Short Category will also allow me to select only the data I need for a specific type; in this case, I needed the 'food' category.

Using Pandas to manipulate the data, each venue type's sum is outputted in descending order along with its total.

```
In [16]: #Obtain the Number of Main categories for the whole Metro Manila
         NCR_venues[['Short Category']].value_counts()

Out[16]: Short Category
         food                  1082
         shops                  324
         arts_entertainment      56
         building                45
         parks_outdoors          44
         travel                  36
         nightlife               35
         education                3
         dtype: int64
```

It will be a good idea to visualize the data. However, with the values returned by the system, it is clear what the top 3 types of venues are.

In order to obtain a data frame that contains the totaled summary of available places/businesses in each city, the data frame is first transformed by one-hot encoding (0/1) the venue types and then adding up the values per city.

Table 3 Total Location Types per City

	arts_entertainment	building	education	food	nightlife	parks_outdoors	shops	travel
City								
CALOOCAN CITY	3	1	0	59	1	0	36	0
CITY OF LAS PIÑAS	2	3	0	72	5	1	17	0
CITY OF MAKATI	1	3	0	62	2	1	24	7
CITY OF MALABON	3	0	0	63	2	4	21	0
CITY OF MANDALUYONG	5	3	0	67	4	0	18	3

The summarized data frame is further processed using the MinMax Scaler to obtain values between 0 and 1. It is essential to note the arrangement of the index due to the merging of tables later on.

Table 4 Scaled Table of Location Type per City

	arts_entertainment	building	education	food	nightlife	parks_outdoors	shops	travel
0	0.4	0.142857	0.0	0.605263	0.2	0.000	1.000000	0.000
1	0.2	0.428571	0.0	0.947368	1.0	0.125	0.366667	0.000
2	0.0	0.428571	0.0	0.684211	0.4	0.125	0.600000	0.875
3	0.4	0.000000	0.0	0.710526	0.4	0.500	0.500000	0.000
4	0.8	0.428571	0.0	0.815789	0.8	0.000	0.400000	0.375

With the data transformed to 1s and 0s, it is now possible to cluster the dataset. However, there are methods, like the Elbow method and the Silhouette score, to determine the optimal number of clusters (k) to analyze the data. For this analysis, the Silhouette score was used because the sample size is less than 18, which is the minimum for the Elbow method.

An advantage of using the Silhouette score in determining the optimal number of clusters is that this test readily shows the optimum number. In previous tests, I observed that the optimal number of groups is 4. However, due to further data processing, the code has gotten mixed up and returned two (2) as the optimum number of clusters.

```
In [31]: #Determine optimal number of clusters
         #  Silhouette Score for K means
         #Import ElbowVisualizer
         from yellowbrick.cluster import KElbowVisualizer
         model = KMeans()
         # k is range of number of clusters
         NCR_main_cat_clustering = NCR_total_short_cat_scaled.copy()
         visualizer = KElbowVisualizer(model, k=(2,16),metric='silhouette', timings= True)
         visualizer.fit(NCR_main_cat_clustering)       # Fit the data to the visualizer
         visualizer.show()       # Finalize and render the figure
```

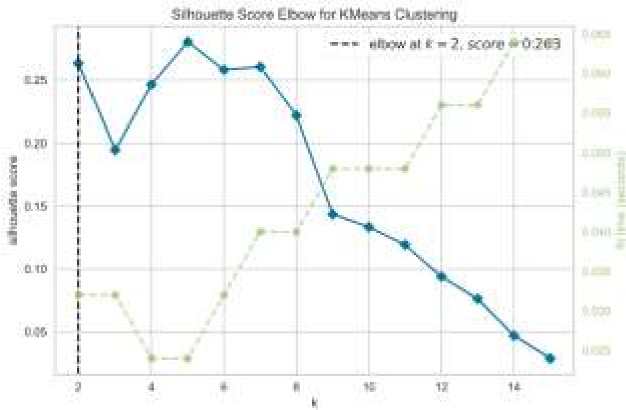


Figure 3 Silhouette Score Method to Determine Optimum Number of Groups

I ran an unsupervised machine learning algorithm with all this data, specifically a k-means clustering algorithm from the Scikit-learn package. One could use the elbow method to define the k value systematically. Still, I chose k to be four since most of my tests returned four as the optimum number.

The same steps were recreated to determine the clusters of restaurants for each city. However, in selecting the datapoints, only those with the 'food' location type were selected. After saving this to a data frame, the data was visualized through a bar graph and histogram to determine the distribution of the data set.

```
In [48]: #obtain data for food related locations
         NCR_cuisines = NCR_venues[NCR_venues['Short Category'] == 'food']
         NCR_cuisines = NCR_cuisines.reset_index(drop=True)
         NCR_cuisines.head()
```

Out[48]:

	City	City Latitude	City Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	Short Category
0	CITY OF MANILA	14.599512	120.984219	Krispy Kreme	14.601195	120.982774	Donut Shop	food
1	CITY OF MANILA	14.599512	120.984219	BonChon Chicken	14.601194	120.982791	Fried Chicken Joint	food
2	CITY OF MANILA	14.599512	120.984219	The Den	14.598827	120.979450	Coffee Shop	food
3	CITY OF MANILA	14.599512	120.984219	Wai Ying Fastfood (嶸榮小食館)	14.601992	120.975985	Chinese Restaurant	food
4	CITY OF MANILA	14.599512	120.984219	Ying Ying Tea House	14.597885	120.976729	Chinese Restaurant	food

```
In [49]: f"There are {NCR_cuisines[['Short Category']].value_counts().sum()} items relating to food"
Out[49]: 'There are 1082 items relating to food'

In [50]: cuisines_data = NCR_cuisines['Venue Category'].value_counts(sort=True).to_frame().reset_index()
         cuisines_data.rename(index = str, columns ={'index':'Cuisine','Venue Category':'Count'}, inplace=True )

In [51]: cuisines_data.head()
```

Out[51]:

	Cuisine	Count
0	Coffee Shop	107
1	Fast Food Restaurant	78
2	Café	76
3	Filipino Restaurant	74
4	Chinese Restaurant	60

Figure 4 Code to Filter Food Location Type Data

I had determined that the majority of the food-venue category has a value of 1 to 11.6. As much as possible, the data should be flat because I am concerned only with the top cuisines available in Manila.

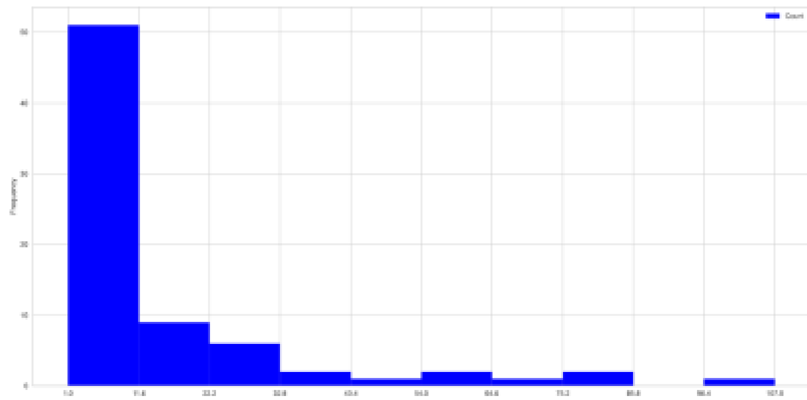


Figure 5 Histogram of Cuisines available in Metro Manila

In order to easily remove the unwanted cuisines in the dataset, I used the code in the figure below. The logic behind this is that I collect the unwanted cuisines and save this as a list. From the filtered food data frame, I only selected the rows, *not in* the list (seen in line 60).

```
In [56]: #obtain rows for venue category with countss less than 11.7
         cuisines_to_remove = cuisines_data[cuisines_data['Count'] < 11.7]

In [57]: cuisines_to_remove.shape

Out[57]: (51, 2)

In [58]: remove_rows = cuisines_to_remove['Cuisine'].tolist()

In [59]: f"Number of Rows to be removed is {cuisines_to_remove['Count'].sum()}"

Out[59]: 'Number of Rows to be removed is 206'

In [60]: NCR_cuisines_cleaned = NCR_cuisines[~NCR_cuisines['Venue Category'].isin(remove_rows)]

In [61]: f"Original Number of columns from NCR_venues DataFrame is {NCR_cuisines.shape[0]}"

Out[61]: 'Original Number of columns from NCR_venues DataFrame is 1082'

In [62]: if (NCR_cuisines_cleaned.shape[0] == (NCR_cuisines.shape[0]-cuisines_to_remove['Count'].sum())):
             print("Successfully removed rows")

         Successfully removed rows
```

Figure 6 Code to Clean Unwanted Data

The data is the one-hot encoded using the mean and was processed using the k-means algorithm with the optimum number of clusters obtained from the Silhouette score.

Results

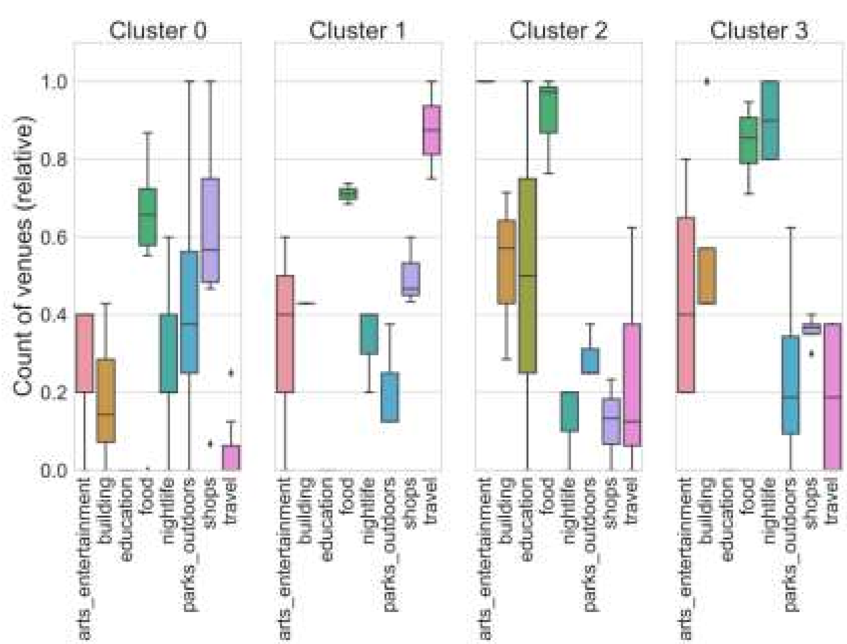
```
In [16]: #Obtain the Number of Main categories for the whole Metro Manila
         NCR_venues[['Short Category']].value_counts()

Out[16]: Short Category
         food                 1082
         shops                 324
         arts_entertainment     56
         building               45
         parks_outdoors         44
         travel                 36
         nightlife              35
         education               3
         dtype: int64
```

The picture above shows that the top most-common venue types in Metro Manila are Food, Shops, and Art and Entertainment.

Venue Type Cluster

Four (4) clusters of location type concentrations for Metro Manila are shown in the box plot below. The different clusters assigned from 0 to 3 show the most common venue type available in each city. However, it is still difficult to discern the depicted data, so I provided more context on the next page.



For simplicity's sake, Clusters 0 to 3 have been renamed to Clusters 1 to 4. The analysis has outputted groups that are divided into shops, travel, arts and entertainment, and buildings. The clusters have similarities, but some refinement can still be done to represent more accurate data.

Cluster 1: Shops

	City	Cluster	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue
0	CALOOCAN CITY	0	shops	food	arts_entertainment
3	CITY OF MALABON	0	food	parks_outdoors	shops
8	CITY OF NAVOTAS	0	nightlife	parks_outdoors	shops
10	CITY OF PASIG	0	food	shops	building
12	CITY OF VALENZUELA	0	shops	parks_outdoors	food
15	QUEZON CITY	0	shops	food	nightlife
16	TAGUIG CITY	0	parks_outdoors	food	shops

Cluster 2: Travel

	City	Cluster	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue
2	CITY OF MAKATI	1	travel	food	shops
7	CITY OF MUNTINLUPA	1	travel	food	shops
13	PASAY CITY	1	travel	food	arts_entertainment

Cluster 3: Arts and Entertainment

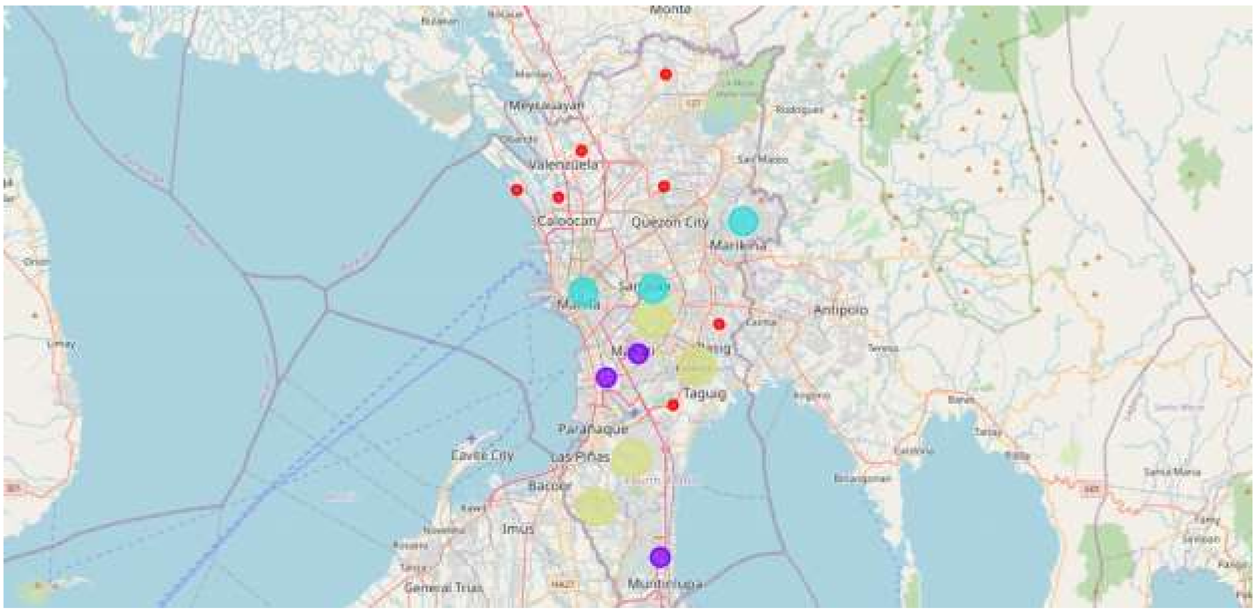
	City	Cluster	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue
5	CITY OF MANILA	2	arts_entertainment	education	food
6	CITY OF MARIKINA	2	arts_entertainment	food	education
11	CITY OF SAN JUAN	2	arts_entertainment	food	building

Cluster 4: Building and Nightlife

	City	Cluster	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue
1	CITY OF LAS PIÑAS	3	nightlife	food	building
4	CITY OF MANDALUYONG	3	food	arts_entertainment	nightlife
9	CITY OF PARAÑAQUE	3	building	food	nightlife
14	PATEROS	3	nightlife	food	parks_outdoors

We can now use the cluster labels to show the city districts marked with a cluster-specific color on a map, again using Folium.

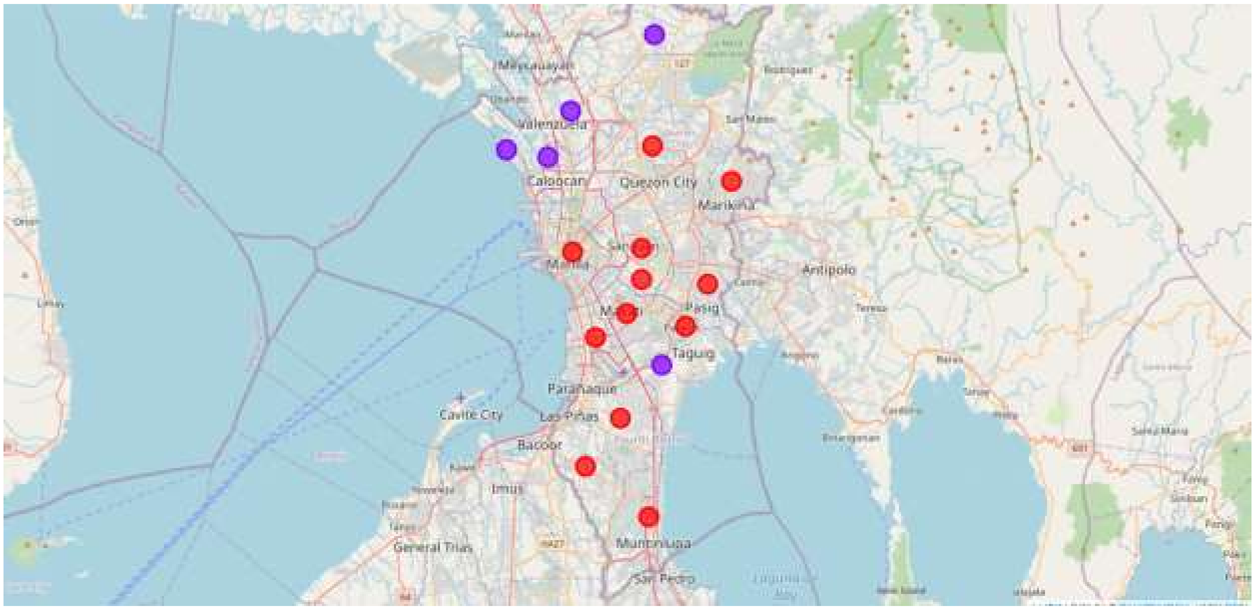
The different clusters are shown on the map. For easier visualization, I have used different sizes for each cluster – this, however, does not accurately depict the size of the cluster – to differentiate each of them easily. There are seventeen (17) bubbles for the seventeen (17) cities, with four (4) different colors for the four (4) clusters.



With the analysis complete, we can now define where a person can set up an establishment or business with the different clusters in Metro Manila.

Cuisine Cluster

All of the significant cuisines are represented thanks to the K Means clustering method. Cluster 0, where people have more choices for fast food. Cluster 1 is where Restaurants and Cafés are the 1st choices for people.



Cluster 0: Fast Food

	City	Cluster	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
0	CALOOCAN CITY	0	Fast Food Restaurant	Coffee Shop	Pizza Place	Filipino Restaurant	Bakery	Café	Burger Joint	Ice Cream Shop
3	CITY OF MALABON	0	Fast Food Restaurant	Chinese Restaurant	Café	Asian Restaurant	Bubble Tea Shop	Pizza Place	Diner	Tea Room
8	CITY OF NAVOTAS	0	Filipino Restaurant	Fast Food Restaurant	Bubble Tea Shop	Steakhouse	Café	Chinese Restaurant	Dessert Shop	Tea Room
12	CITY OF VALENZUELA	0	Fast Food Restaurant	Chinese Restaurant	Café	Coffee Shop	Donut Shop	Pizza Place	Bubble Tea Shop	Burger Joint
16	TAGUIG CITY	0	Fast Food Restaurant	Coffee Shop	Pizza Place	Café	Restaurant	Chinese Restaurant	Filipino Restaurant	Japanese Restaurant

Cluster 1: Coffee Shop and Restaurants

	City	Cluster	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
1	CITY OF LAS PIÑAS	1	Filipino Restaurant	Japanese Restaurant	Coffee Shop	Tea Room	Fast Food Restaurant	Pizza Place	Bubble Tea Shop	Café
2	CITY OF MAKATI	1	Café	Coffee Shop	Japanese Restaurant	Italian Restaurant	Filipino Restaurant	Korean Restaurant	Bakery	Restaurant
4	CITY OF MANDALUYONG	1	Filipino Restaurant	Japanese Restaurant	Café	Coffee Shop	Italian Restaurant	Bakery	Chinese Restaurant	Restaurant
5	CITY OF MANILA	1	Chinese Restaurant	Filipino Restaurant	Coffee Shop	Bakery	Ice Cream Shop	Bubble Tea Shop	Café	Tea Room
6	CITY OF MARIKINA	1	Filipino Restaurant	Coffee Shop	Café	Diner	Restaurant	BBQ Joint	Wings Joint	Bubble Tea Shop
7	CITY OF MUNTINLUPA	1	Coffee Shop	Café	Filipino Restaurant	Italian Restaurant	Pizza Place	Chinese Restaurant	Diner	Restaurant
9	CITY OF PARAÑAQUE	1	Coffee Shop	Filipino Restaurant	Bubble Tea Shop	Fast Food Restaurant	Japanese Restaurant	Asian Restaurant	Chinese Restaurant	Diner
10	CITY OF PASIG	1	Coffee Shop	Café	Fast Food Restaurant	Bakery	Restaurant	Japanese Restaurant	Filipino Restaurant	Pizza Place
11	CITY OF SAN JUAN	1	Coffee Shop	Filipino Restaurant	Chinese Restaurant	Japanese Restaurant	Fast Food Restaurant	Ice Cream Shop	Café	Pizza Place
13	PASAY CITY	1	Café	Japanese Restaurant	Coffee Shop	Filipino Restaurant	Pizza Place	Dessert Shop	Steakhouse	Snack Place
14	PATEROS	1	Coffee Shop	Steakhouse	Café	Ice Cream Shop	Italian Restaurant	Bakery	Burger Joint	Restaurant
15	QUEZON CITY	1	Coffee Shop	Pizza Place	Café	Japanese Restaurant	Bakery	Fast Food Restaurant	Bubble Tea Shop	Burger Joint

Conclusion

From the results, we can see the readily available venue type for people residing in Metro Manila. Subsequently, we can also know what type of business is best established per city while even knowing where to eat. Although the analysis is not perfect and further refinement is needed, it is a good stepping stone in learning data science.

Appendix

The publications of Dr. Johannes Wagner and Kristian Jackson have helped me in various stages of this project. Additionally, the IBM Labs resources that were taken up to this point have helped in reviewing codes needed to analyze and visualize the data.

Link to Dr. Johannes Wagner's blog post: https://www.linkedin.com/pulse/applied-data-science-capstone-project-restaurant-wagner-mba/
Link to Kristian Jackson's Git Hub Repository: https://github.com/kristianjackson/Coursera_Capstone
Link to Koop, 2021: https://www.visualcapitalist.com/most-populous-cities-in-the-world/
Link to the PSA Data: https://psa.gov.ph/sites/default/files/attachments/hsd/pressrelease/2015_Table%201_Legislative%20Districts.xlsx