Capstone Project - The Battle of Neighborhoods

1. Introduction/Business Understanding

1.1 Description of the problem

The business problem that we make is facing the biggest soccer league event in Europe, the UEFA Champions League Final 2020 scheduled to be played at the Atatürk Olympic Stadium in Istanbul, Turkey, how could we provide support to different visitors that coming from many countries to list and visualize Istanbul districts that fit their needs in term of culinary/ food venues. So specifically, this report will be targeted to stakeholders who want to visit the Atatürk Olympic Stadium in Istanbul, Turkey.

1.2 Discussion of the background

With a population of 15.5 million people and one of the largest cities in the world established on two continental continents, namely Asia and Europe, estimates that hundreds of thousands of visitors from local and abroad will be crowded in the city of Istanbul to attend the biggest football event this coming Europe.

Besides that, with this very vital city location, you can see a modern western city combined with a traditional eastern city where many cultural differences, multi-religious, historic buildings, shopping centers, a lively nightlife scene, real estate investment, and entertainment venues making the interest of many tourists to visit this historic city in Turkey as a tourist destination.

Istanbul has a lot of culinary types that offer a variety of domestic food and also a choice of food from around the world and places to eat in 39 different districts consisting of 25 districts sit on the European side, and 14 rest on the Asian side. Because of the enormous size of these districts, to get to know them all would take a lifetime so it is very difficult for newcomers especially those who want to go to restaurants to choose the type of restaurant that suits their tastes from the wide variety of information in the media.

In this Capstone Project, the problem solving steps will be carried out on how to utilize Foursquare location data and use one of the machine learning techniques, namely clustering to make an analysis and decision to determine which neighborhoods are in accordance with the tastes of customers in the city of Istanbul when the big event was held.

2. Data

In this Capstone Project we need the data needed to be processed and analyzed, including

- **List of districts of Istanbul**
  Source : https://en.wikipedia.org/wiki/List_of_districts_of_Istanbul
  The list of districts of Istanbul data in the form of this table will be taken from Wikipedia through the scrapping method then that table will be cleaned, explored, and processed then to add coordinates in each district in Istanbul can automatically use the geocoder class of Geopy client.

- **Food Venue/Restaurants in each neighborhoods of Istanbul City**
  Source : Foursquare APIs
  By using this Foursquare APIs after logging in using registered credentials, we will get all the closest places from all neighborhoods in Istanbul City so that we can get restaurants by filtering the data of the closest places from all the neighborhoods.

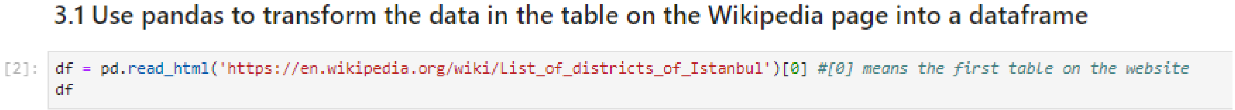
By using this Foursquare APIs after logging in using registered credentials, we will get all the closest places from all neighborhoods in Istanbul City so that we can get restaurants by filtering the data of the closest places from all the neighborhoods.

3. Methodology

3.1 Data Preparation

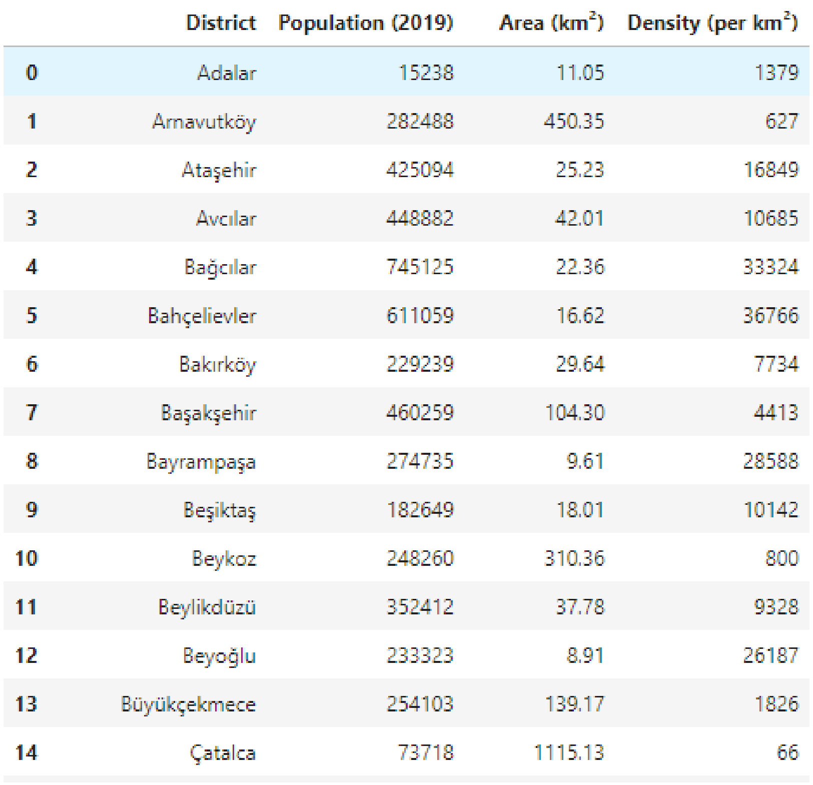
3.1.1 Scraping List of Districts of Istanbul Table from Wikipedia

First, use List of Districts of Istanbul page from Wikipedia to scrap the table to create a dataframe. For this, I've used pandas to transform the data in the table on the Wikipedia page into a dataframe containing name of the 39 districts of Istanbul, Population, Area, and Density for each district. Below is the method and result of the scrapping table method from Wikipedia.



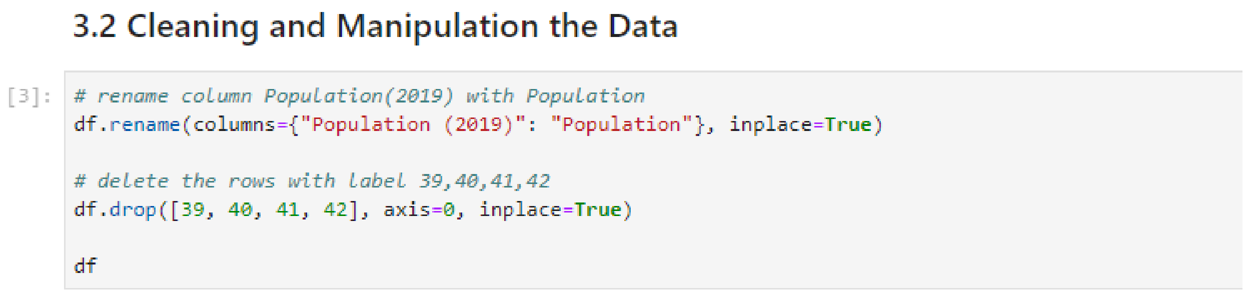
3.1 Use pandas to transform the data in the table on the Wikipedia page into a dataframe

```
[2]: df = pd.read_html('https://en.wikipedia.org/wiki/List_of_districts_of_Istanbul')[0] #[0] means the first table on the website
     df
```



	District	Population (2019)	Area (km^2)	Density (per km^2)
0	Adalar	15238	11.05	1379
1	Arnavutköy	282488	450.35	627
2	Ataşehir	425094	25.23	16849
3	Avcılar	448882	42.01	10685
4	Bağcılar	745125	22.36	33324
5	Bahçelievler	611059	16.62	36766
6	Bakırköy	229239	29.64	7734
7	Başakşehir	460259	104.30	4413
8	Bayrampaşa	274735	9.61	28588
9	Beşiktaş	182649	18.01	10142
10	Beykoz	248260	310.36	800
11	Beylikdüzü	352412	37.78	9328
12	Beyoğlu	233323	8.91	26187
13	Büyükçekmece	254103	139.17	1826
14	Çatalca	73718	1115.13	66

then performed Cleaning and Manipulation of the Data by rename column Population (2019) with Population and delete the rows with label 39, 40, 41, 42 which is not really needed for further analysis and detection of missing data in each row to ensure complete data. Below are the methods for cleaning and manipulating the data

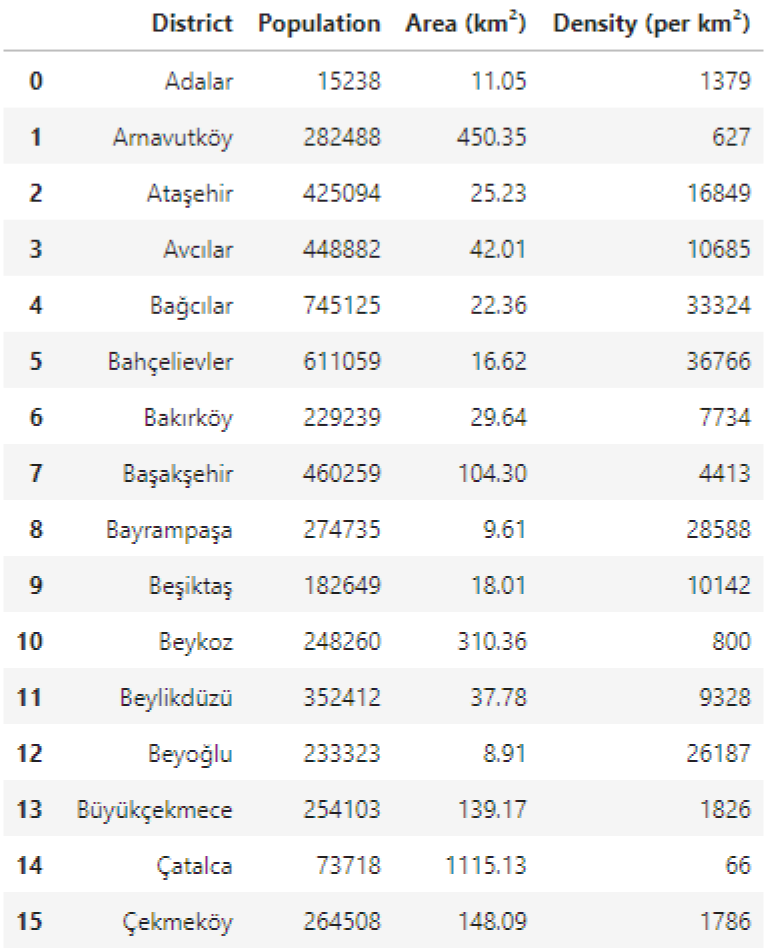


3.2 Cleaning and Manipulation the Data

[3]:

```
# rename column Population(2019) with Population
df.rename(columns={"Population (2019)": "Population"}, inplace=True)

# delete the rows with label 39,40,41,42
df.drop([39, 40, 41, 42], axis=0, inplace=True)

df
```

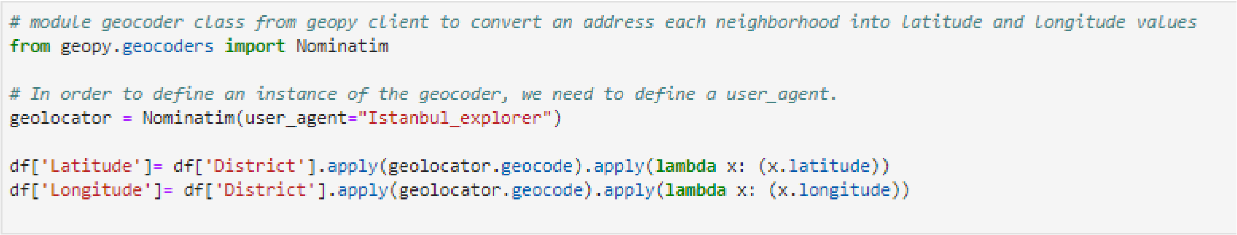
After little manipulation, the data-frame is obtained as below



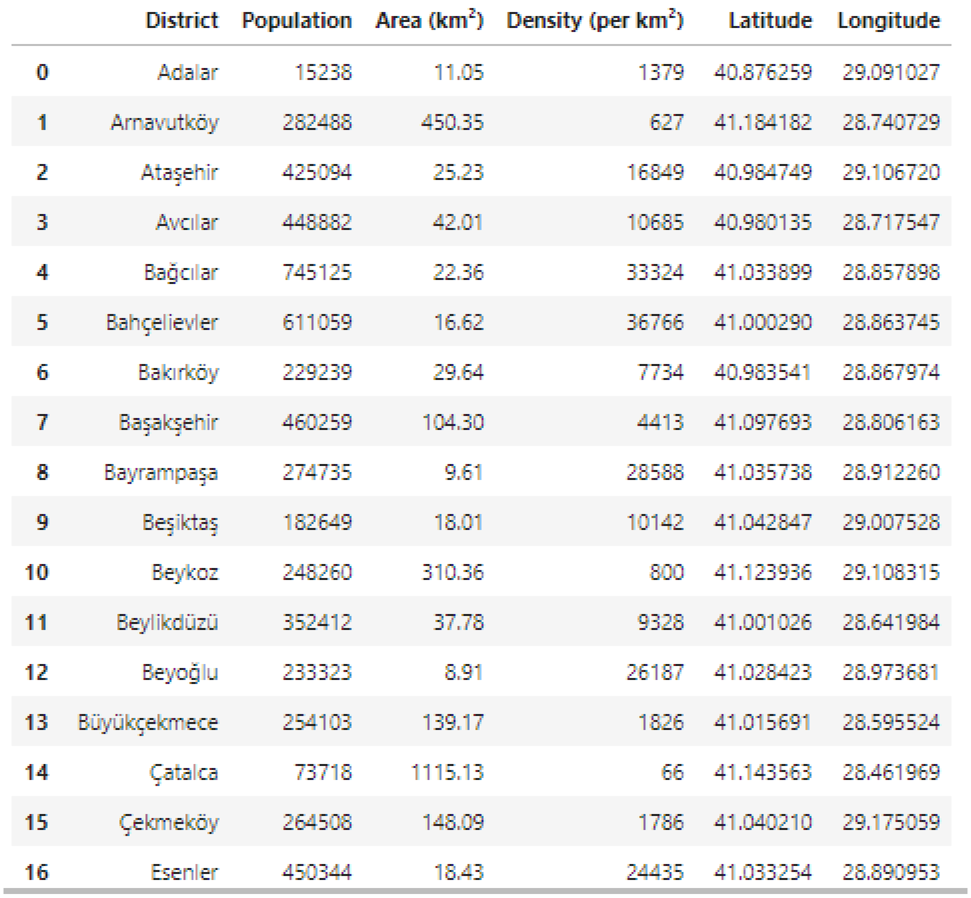
	District	Population	Area (km^2)	Density (per km^2)
0	Adalar	15238	11.05	1379
1	Arnavutköy	282488	450.35	627
2	Ataşehir	425094	25.23	16849
3	Avcılar	448882	42.01	10685
4	Bağcılar	745125	22.36	33324
5	Bahçelievler	611059	16.62	36766
6	Bakırköy	229239	29.64	7734
7	Başakşehir	460259	104.30	4413
8	Bayrampaşa	274735	9.61	28588
9	Beşiktaş	182649	18.01	10142
10	Beykoz	248260	310.36	800
11	Beylikdüzü	352412	37.78	9328
12	Beyoğlu	233323	8.91	26187
13	Büyükçekmece	254103	139.17	1826
14	Çatalca	73718	1115.13	66
15	Çekmeköy	264508	148.09	1786

3.1.2 Getting Coordinates of Districts using Geopy Client

Next, get the coordinates of these 39 districts in Istanbul using geocoder class of Geopy client as follow

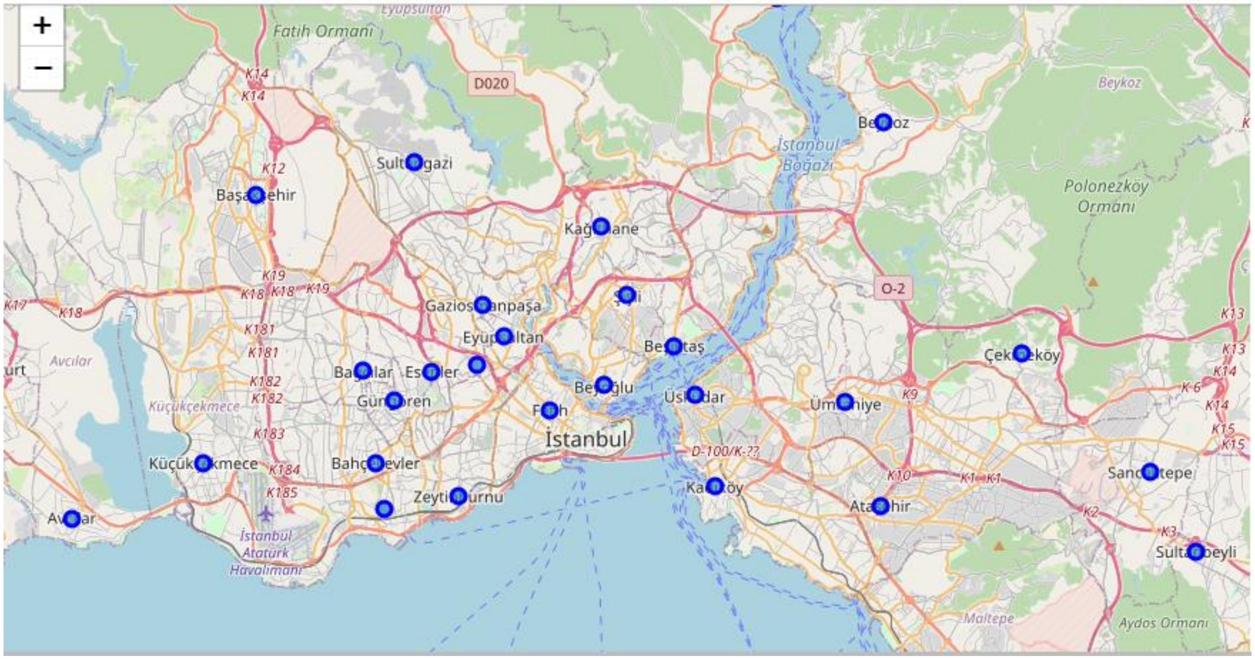


```
# module geocoder class from geopy client to convert an address each neighborhood into latitude and longitude values
from geopy.geocoders import Nominatim

# In order to define an instance of the geocoder, we need to define a user_agent.
geolocator = Nominatim(user_agent="Istanbul_explorer")

df['Latitude']= df['District'].apply(geolocator.geocode).apply(lambda x: (x.latitude))
df['Longitude']= df['District'].apply(geolocator.geocode).apply(lambda x: (x.longitude))
```



	District	Population	Area (km²)	Density (per km²)	Latitude	Longitude
0	Adalar	15238	11.05	1379	40.876259	29.091027
1	Arnavutköy	282488	450.35	627	41.184182	28.740729
2	Ataşehir	425094	25.23	16849	40.984749	29.106720
3	Avcılar	448882	42.01	10685	40.980135	28.717547
4	Bağcılar	745125	22.36	33324	41.033899	28.857898
5	Bahçelievler	611059	16.62	36766	41.000290	28.863745
6	Bakırköy	229239	29.64	7734	40.983541	28.867974
7	Başakşehir	460259	104.30	4413	41.097693	28.806163
8	Bayrampaşa	274735	9.61	28588	41.035738	28.912260
9	Beşiktaş	182649	18.01	10142	41.042847	29.007528
10	Beykoz	248260	310.36	800	41.123936	29.108315
11	Beylikdüzü	352412	37.78	9328	41.001026	28.641984
12	Beyoğlu	233323	8.91	26187	41.028423	28.973681
13	Büyükçekmece	254103	139.17	1826	41.015691	28.595524
14	Çatalca	73718	1115.13	66	41.143563	28.461969
15	Çekmeköy	264508	148.09	1786	41.040210	29.175059
16	Esenler	450344	18.43	24435	41.033254	28.890953

I used python **folium** library to visualize geographic details of Istanbul and its 39 districts and I created a map of Istanbul with neighborhood superimposed on top. I used latitude and longitude values to get the visual as below:

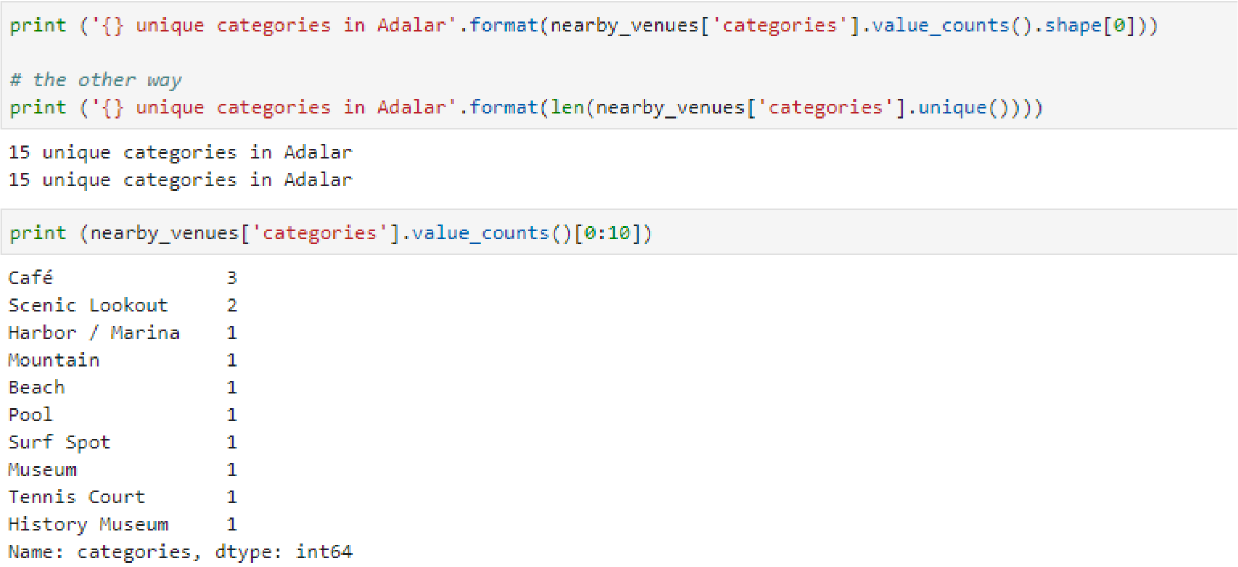


3.2 Exploratory Data Analysis

I will use exploratory data analysis (EDA) to give properties of each neighborhood in Istanbul, especially the food venues/restaurant of each neighborhood and provide useful insights to the people who visited the area.

3.2.1 Using Foursquare Location Data

Finally, let’s make utilizing the Foursquare API to explore the neighborhoods and get the top 100 venues that are in Adalar within a radius of 500 meters. After send the GET request, we will get numbers of unique venue categories in Adalar. And we will see numbers of each venue categories using value_counts.



```
print ('{} unique categories in Adalar'.format(nearby_venues['categories'].value_counts().shape[0]))

# the other way
print ('{} unique categories in Adalar'.format(len(nearby_venues['categories'].unique())))
```

```
15 unique categories in Adalar
15 unique categories in Adalar
```

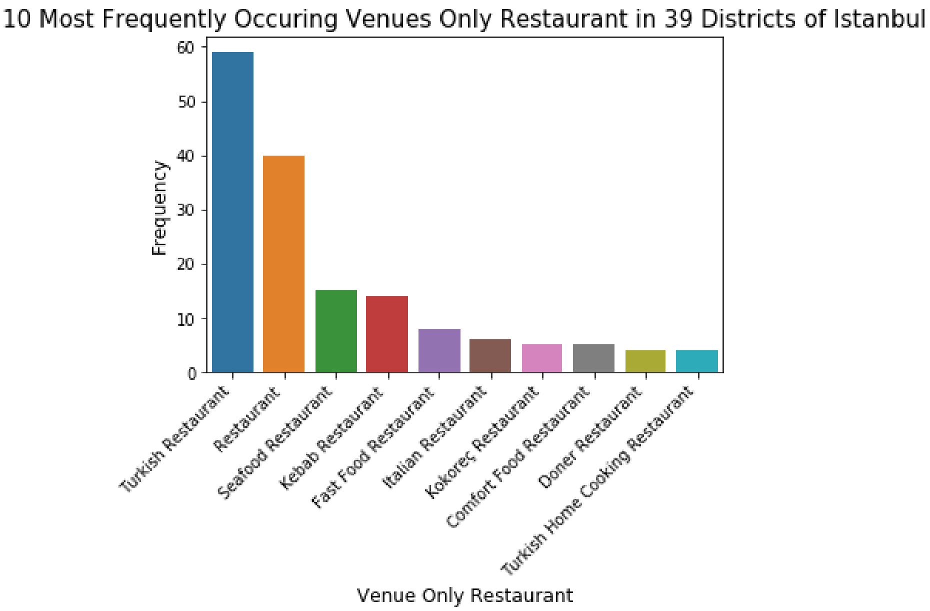
```
print (nearby_venues['categories'].value_counts()[0:10])
```

```
Café                3
Scenic Lookout      2
Harbor / Marina     1
Mountain            1
Beach               1
Pool                1
Surf Spot           1
Museum              1
Tennis Court        1
History Museum      1
Name: categories, dtype: int64
```

We notice that 15 unique venue categories were returned by Foursquare and Café in the top of venue categories as we can see above.

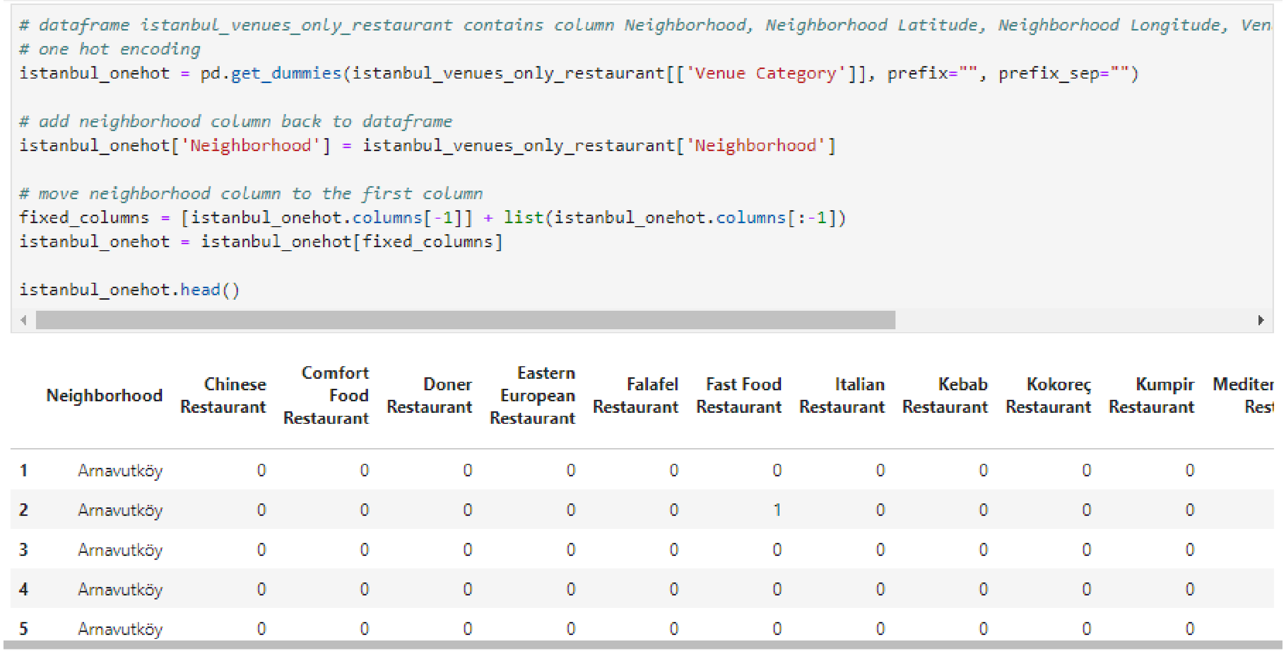
Then I will concentrate in Restaurant Category only and explore all the 39 districts in Istanbul.

We find out 18 unique venue categories that only restaurant in Istanbul and Turkish Restaurants top the charts as we can see in the plot below



So, you need to try the delicious turkish food if you visit Istanbul, but in which district Turkish restaurant are more common? Let's get back to exploring the data a little more. Let's analyze each neighborhood to know about the top 5 venues of each one. So, we proceed as follows:

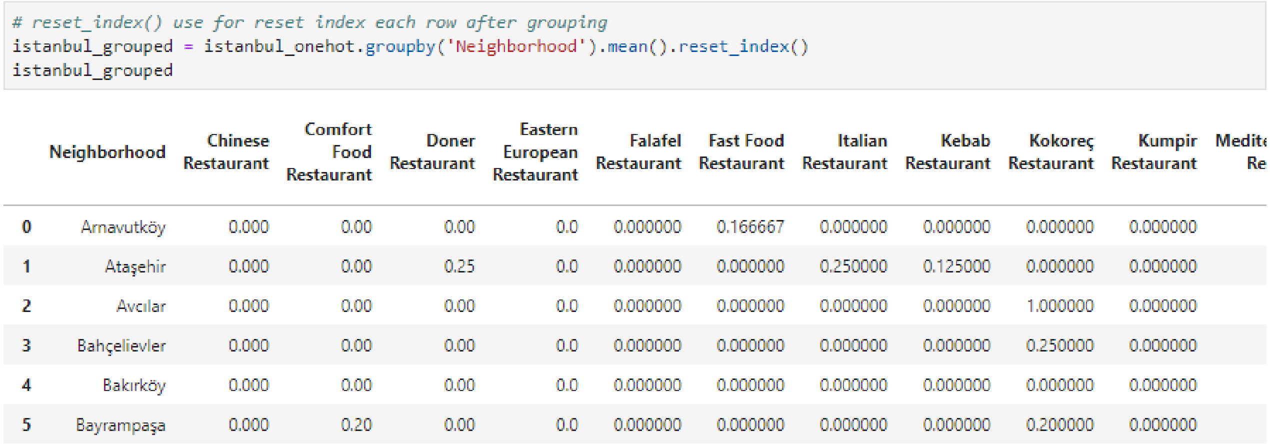
1. Create a data-frame with pandas one hot encoding for the venue categories that only contain restaurant.



```
# dataframe istanbul_venues_only_restaurant contains column Neighborhood, Neighborhood Latitude, Neighborhood Longitude, Ven
# one hot encoding
istanbul_onehot = pd.get_dummies(istanbul_venues_only_restaurant[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
istanbul_onehot['Neighborhood'] = istanbul_venues_only_restaurant['Neighborhood']

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

istanbul_onehot.head()
```

	Neighborhood	Chinese Restaurant	Comfort Food Restaurant	Doner Restaurant	Eastern European Restaurant	Falafel Restaurant	Fast Food Restaurant	Italian Restaurant	Kebab Restaurant	Kokoreç Restaurant	Kumpir Restaurant	Mediter Rest
1	Arnavutköy	0	0	0	0	0	0	0	0	0	0	
2	Arnavutköy	0	0	0	0	0	1	0	0	0	0	
3	Arnavutköy	0	0	0	0	0	0	0	0	0	0	
4	Arnavutköy	0	0	0	0	0	0	0	0	0	0	
5	Arnavutköy	0	0	0	0	0	0	0	0	0	0	

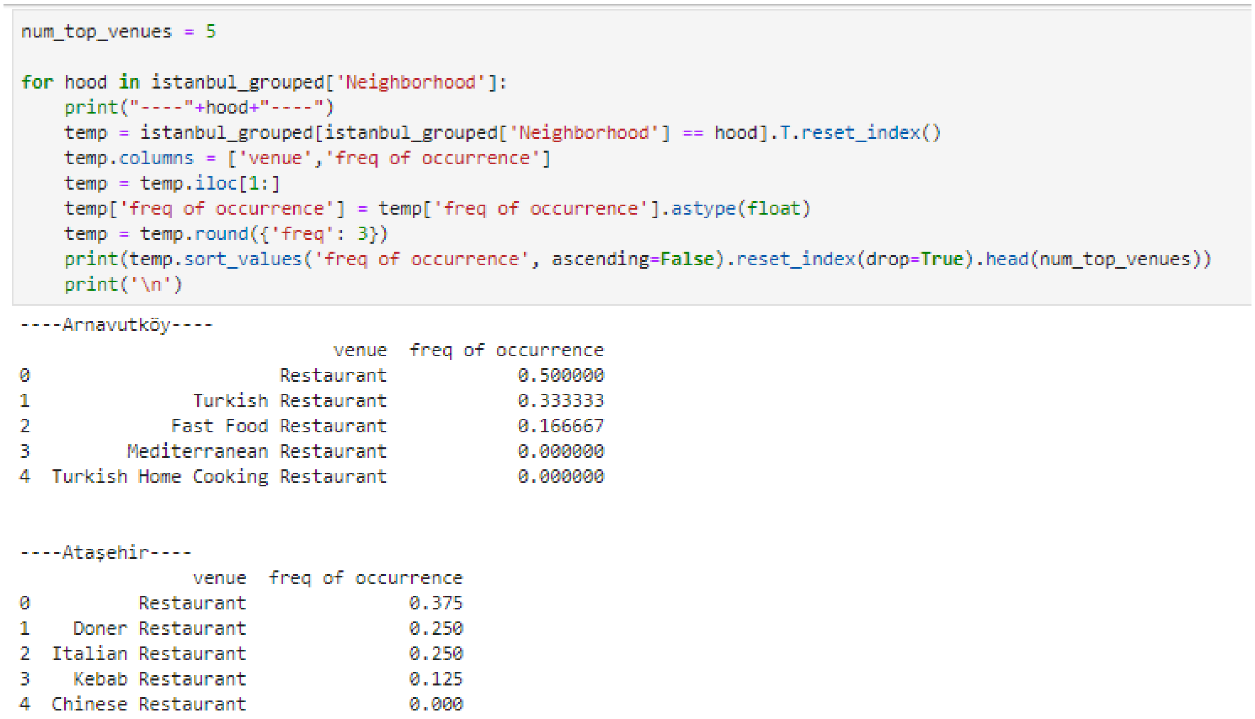
2. Use pandas groupby on neighborhood column and calculate the mean of the frequency of occurrence of each venue category.



```
# reset_index() use for reset index each row after grouping
istanbul_grouped = istanbul_onehot.groupby('Neighborhood').mean().reset_index()
istanbul_grouped
```

	Neighborhood	Chinese Restaurant	Comfort Food Restaurant	Doner Restaurant	Eastern European Restaurant	Falafel Restaurant	Fast Food Restaurant	Italian Restaurant	Kebab Restaurant	Kokoreç Restaurant	Kumpir Restaurant	Medite Re
0	Arnavutköy	0.000	0.00	0.00	0.0	0.000000	0.166667	0.000000	0.000000	0.000000	0.000000	
1	Ataşehir	0.000	0.00	0.25	0.0	0.000000	0.000000	0.250000	0.125000	0.000000	0.000000	
2	Avcılar	0.000	0.00	0.00	0.0	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	
3	Bahçelievler	0.000	0.00	0.00	0.0	0.000000	0.000000	0.000000	0.000000	0.250000	0.000000	
4	Bakırköy	0.000	0.00	0.00	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
5	Bayrampaşa	0.000	0.20	0.00	0.0	0.000000	0.000000	0.000000	0.000000	0.200000	0.000000	

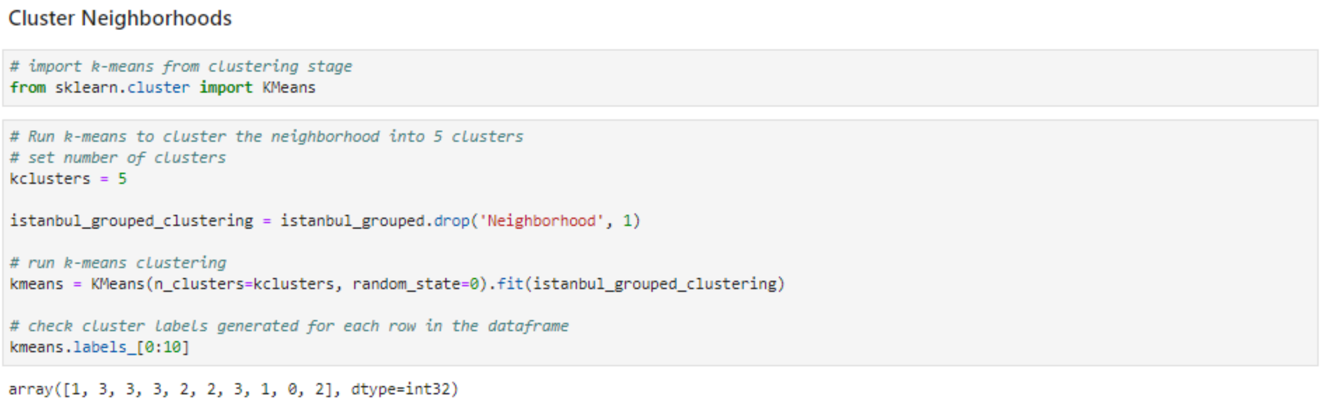
3. Output each neighborhood along with the top 5 most common venues categories that only restaurant in Istanbul.



```
num_top_venues = 5

for hood in istanbul_grouped['Neighborhood']:
    print("----"+hood+"----")
    temp = istanbul_grouped[istanbul_grouped['Neighborhood'] == hood].T.reset_index()
    temp.columns = ['venue','freq of occurrence']
    temp = temp.iloc[1:]
    temp['freq of occurrence'] = temp['freq of occurrence'].astype(float)
    temp = temp.round({'freq': 3})
    print(temp.sort_values('freq of occurrence', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

```
----Arnavutköy----
                         venue  freq of occurrence
0                   Restaurant            0.500000
1           Turkish Restaurant            0.333333
2         Fast Food Restaurant            0.166667
3      Mediterranean Restaurant            0.000000
4  Turkish Home Cooking Restaurant            0.000000


----Ataşehir----
                venue  freq of occurrence
0          Restaurant               0.375
1    Doner Restaurant               0.250
2  Italian Restaurant               0.250
3    Kebab Restaurant               0.125
4  Chinese Restaurant               0.000
```

To help a tourist decide a location to go for a restaurant/food venue, we will use clustering method to clustering these 34 districts based on the venue categories that only contain restaurant and we will use k-Means clustering. So, our expectation would be based on the similarities of venue categories, these districts will be clustered. I have used the code below

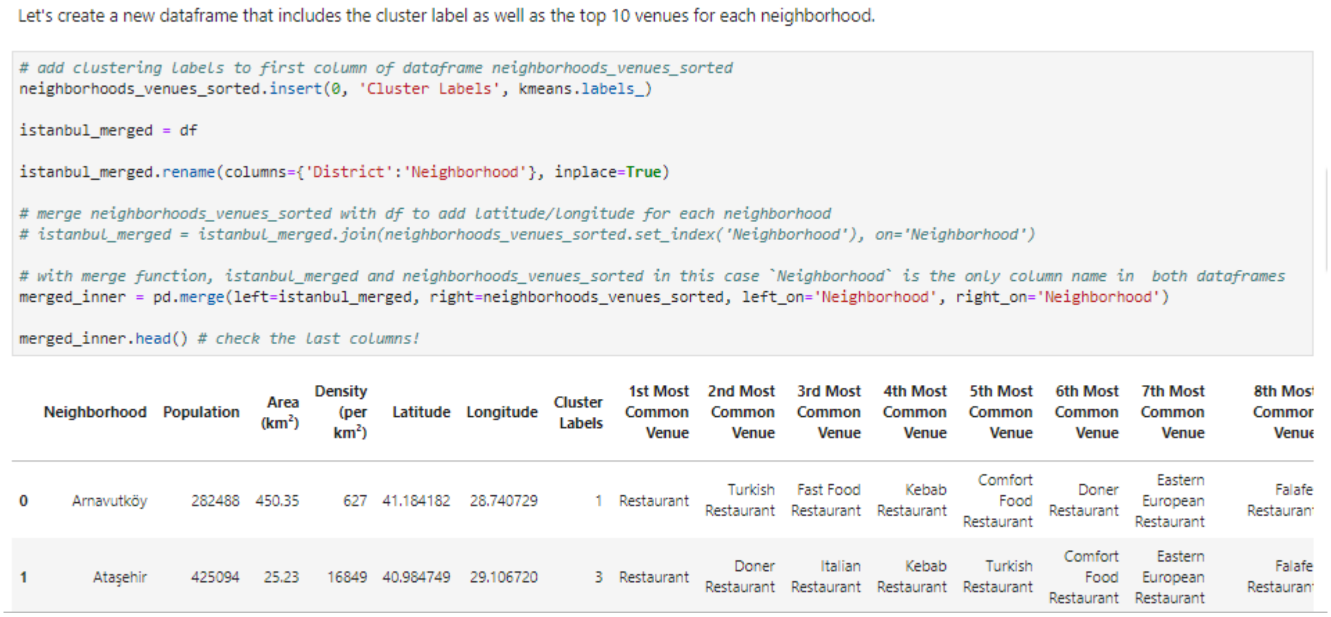


Cluster Neighborhoods

```
# import k-means from clustering stage
from sklearn.cluster import KMeans
```

```
# Run k-means to cluster the neighborhood into 5 clusters
# set number of clusters
kclusters = 5

istanbul_grouped_clustering = istanbul_grouped.drop('Neighborhood', 1)

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

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

```
array([1, 3, 3, 3, 2, 2, 3, 1, 0, 2], dtype=int32)
```

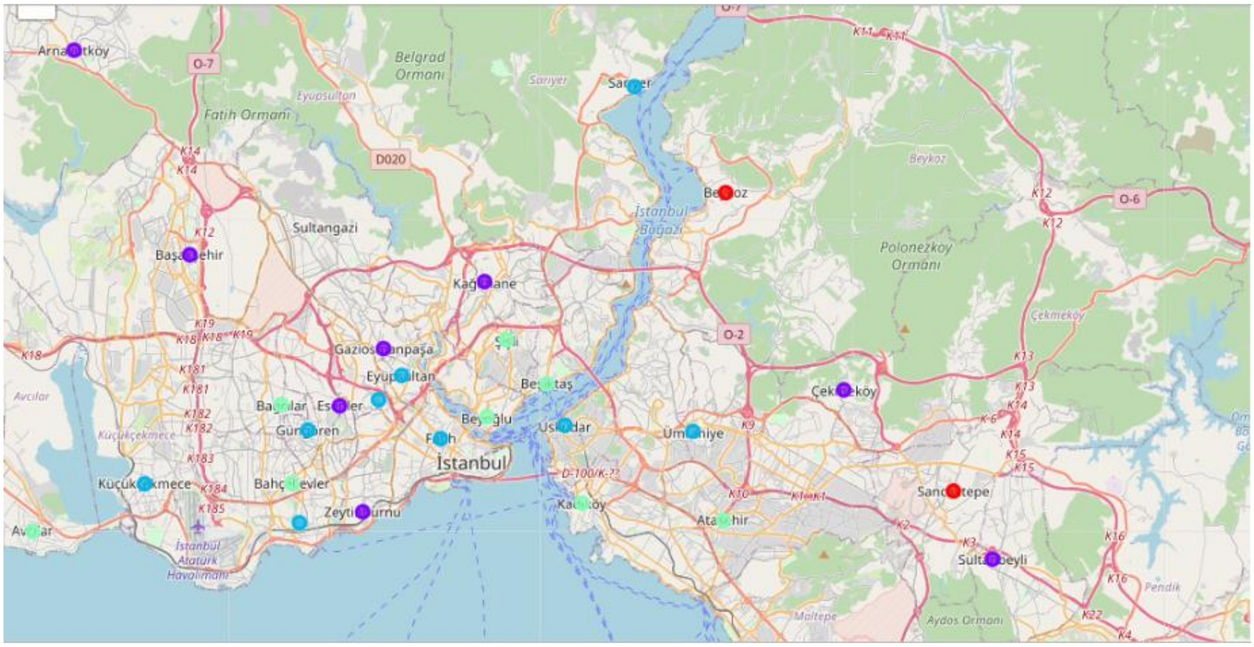


Let's create a new dataframe that includes the cluster label as well as the top 10 venues for each neighborhood.

```
# add clustering labels to first column of dataframe neighborhoods_venues_sorted
neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

istanbul_merged = df

istanbul_merged.rename(columns={'District':'Neighborhood'}, inplace=True)

# merge neighborhoods_venues_sorted with df to add latitude/longitude for each neighborhood
# istanbul_merged = istanbul_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

# with merge function, istanbul_merged and neighborhoods_venues_sorted in this case `Neighborhood` is the only column name in  both dataframes
merged_inner = pd.merge(left=istanbul_merged, right=neighborhoods_venues_sorted, left_on='Neighborhood', right_on='Neighborhood')

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

	Neighborhood	Population	Area (km²)	Density (per km²)	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 Mos Commor Venue
0	Arnavutköy	282488	450.35	627	41.184182	28.740729	1	Restaurant	Turkish Restaurant	Fast Food Restaurant	Kebab Restaurant	Comfort Food Restaurant	Doner Restaurant	Eastern European Restaurant	Falafe Restauran
1	Ataşehir	425094	25.23	16849	40.984749	29.106720	3	Restaurant	Doner Restaurant	Italian Restaurant	Kebab Restaurant	Turkish Restaurant	Comfort Food Restaurant	Eastern European Restaurant	Falafe Restauran

We can visualize the resulting clusters by represent these 5 clusters using Folium library as below



4. Results & Discussion

We got a glimpse of the Restaurants in Istanbul and were able to find out some interesting insights which might be useful to travelers who want to visit Istanbul as well as people with business interests. Let’s summarize our findings:

- Turkish Restaurant is the most frequently occuring venues that only restaurant in the 39 Districts of Istanbul.

- The types of restaurants outside Turkey such as Thai Restaurant and Chinese Restaurant are the least frequently occurring venues that only restaurant in the 39 Districts of Istanbul.

- Neighborhood Güngören and Silivri have the highest number of restaurants/food venues in Istanbul.

- Neighborhood Avcılar, Bakırköy, Bağcılar, Beykoz, Kartal, and Tuzla has the least number of restaurants in Istanbul.

- Cluster 1, cluster 2, and cluster 3 which are marked with a circle maker in red, purple and blue in folium map are in the downtown area of Istanbul so they can choose from many types of restaurants, especially Turkish restaurants.

- Istanbul's strategic position has a variety of land lines and many rail lines that connect between neighborhoods making it easy for every tourist to travel, especially to find a place to eat.

- If tourists want to find a place to eat with a taste of Turkey outside such as European restaurants, they can travel in Clusters 4 and Clusters 5 a bit far from downtown Istanbul.

The clustering is completely based on the most common venues obtained from Foursquare data, especially in this clustering is done on the most common venues data that contains the word restaurant in each neighborhood in Istanbul. In this clustering analysis, we make a number of assumptions, including food venue/restaurant data that appears in each neighborhood based on the closest distance from the center of the neighborhood not from the Atatürk Olympic Stadium, ignoring the price range of each restaurant, the cleanliness and hygiene of food from each restaurant, restaurant services and etc. Since we don't have such data and it would be difficult to farm it for a small exploratory study like ours. Hence, our analysis only helps tourists to get an overview of food venue/restaurants distribution by categories in the 39 districts of Istanbul.

5. Conclusion

Many problems in real life where the data associated with these problems can be used to find solutions. For example the problem above we want to help every visitor from various countries to list and visualize Istanbul districts that fit their needs in terms of culinary/food venues. Existing data is used for segmenting and clustering each neighborhood in Istanbul based on the most common venue that contains a word restaurant. From this data will help each tourist determine which neighborhood has a restaurant that suits their interests.

Data manipulation starts from scrapping tables from wikipedia, cleaning and manipulation of the data, getting coordinates of districts using geopy client, using foursquare location data to explore the neighborhoods to get venues that only restaurants and finally use clustering methods to clustering these 34 districts based on the venue categories that only contain restaurant and we can visualize the results using the folium leaflet map.

From the results of clustering 34 neighborhoods in Istanbul we grouped them into five clusters. The first, second, and third clusters located in the center of Istanbul have many types of restaurants, especially the highest number of Turkish restaurants and also foreign restaurants such as Fast Food Restaurant and Italian Restaurant. These three clusters are also closer to the Atatürk Olympic Stadium. The fourth and fifth clusters are located a bit far from downtown Istanbul if tourists want to find a place to eat with a taste of Turkey outside such as European restaurants.