Battle of Neighborhoods – Toronto City

Business Problem

The objective of this project is to find a most suitable neighborhood in Toronto for starting an Indian restaurant. Using clustering technique like K means, this project aims to answer the question " Which neighborhoods should owner consider opening an Indian restaurant in Toronto?"

Target Audience

An entrepreneur who wants to open Indian restaurant in Toronto but is uncertain about which neighborhood.

Data requirement

- I will need an entire list of neighborhoods in Toronto
- Their latitudes and longitudes
- All the restaurants and their types within a 500 M radius in those neighborhoods

Data collection

- I used Wikipedia page for creating the data frame of neighborhoods
- For latitude and longitude of neighborhoods, Geocoder package is used
- Finally, restaurants and their types in a 500-meter radius of a neighborhood were extracted using Foursquare API.

Methodology

First and foremost, I need the data. Wikipedia page https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M has all the postal codes and their respective neighborhoods for Toronto City.

There are various ways of scraping a web page, I will show two of them here.

1. Web Scrapping using BeautifulSoup.

```
url = 'https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M'
response = requests.get(url)
soup = BeautifulSoup(response.content,'lxml')

```

Notice how we have the elements of interest from soup in tags <b> and <span> inside <table>

```
<table cellpadding="2" cellspacing="0" rules="all" style="width:100%; border-collapse:collapse; border:1px solid #ccc;">
<tbody><tr>
<td style="width:11%; vertical-align:top; color:#ccc;">
<p><b>M1A</b><br/><span style="font-size:80%;"><i>Not assigned</i></span>
</p>
</td>
<td style="width:11%; vertical-align:top; color:#ccc;">
<p><b>M2A</b><br/><span style="font-size:80%;"><i>Not assigned</i></span>
</p>
</td>
<td style="width:11%; vertical-align:top;">
<p><b>M3A</b><br/><span style="font-size:80%;"><a href="/wiki/North_York" title="North York">North York</a><br/>(<a href="/wiki/Parkwoods" title="Parkwoods">Parkwoods</a>)</span>
```

I created an empty dataframe called 'neighborhoods' and populated the same using the data from tags above, like this,

```python
# define the dataframe columns
column_names = ['Postal Code', 'Bourough', 'Neighborhood']

# instantiate the dataframe
neighborhoods = pd.DataFrame(columns=column_names)
neighborhoods
```

Postal Code	Bourough	Neighborhood

```python
for b, s in zip(soup.find('table').find_all('b'),soup.find('table').find_all('span')):
    if 'Not assigned' in (b.get_text(),s.get_text()):
        continue
    else:
        pc = b.get_text()
        ne = s.get_text().rsplit('(')[-1].replace(' /',',').replace(')','')
        br = s.get_text().rsplit('(')[0]

        neighborhoods = neighborhoods.append({'Postal Code': pc,
                                              'Bourough': br,
                                              'Neighborhood':ne},ignore_index=True)
```

If done correctly, the output should be as below. Beautiful, isn't it?

	Postal Code	Bourough	Neighborhood
6	M1B	Scarborough	Malvern, Rouge
12	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek
18	M1E	Scarborough	Guildwood, Morningside, West Hill
22	M1G	Scarborough	Woburn
26	M1H	Scarborough	Cedarbrae

2 . Web scrapping using Pandas,

The wepage of wikipedia had 4 tables, the first table has the information we need. Though it is in a bit weird format. When I scrapped the table from page, it looks like this

```python
df = pd.read_html('https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M')
df1=df[0]
```

We have the table in its raw form,

```python
df1.head()
```

	0	1	2	3	4	5	6	7	8
0	M1ANot assigned	M2ANot assigned	M3ANorth York(Parkwoods)	M4ANorth York(Victoria Village)	M5ADowntown Toronto(Regent Park / Harbourfront)	M6ANorth York(Lawrence Manor / Lawrence Heights)	M7AQueen's Park / Ontario Provincial Government	M8ANot assigned	M9AEtobicoke(Islington Avenue)
1	M1BScarborough(Malvern / Rouge)	M2BNot assigned	M3BNorth York(Don Mills)North	M4BEast York(Parkview Hill / Woodbine Gardens)	M5BDowntown Toronto(Garden District, Ryerson)	M6BNorth York(Glencairn)	M7BNot assigned	M8BNot assigned	M9BEtobicoke(West Deane Park / Princess Garden...
2	M1CScarborough(Rouge Hill / Port Union / Highl...	M2CNot assigned	M3CNorth York(Don Mills)South(Flemingdon Park)	M4CEast York(Woodbine Heights)	M5CDowntown Toronto(St. James Town)	M6CYork(Humewood-Cedarvale)	M7CNot assigned	M8CNot assigned	M9CEtobicoke(Eringate / Bloordale Gardens / Ol...
3	M1EScarborough(Guildwood / Morningside / West ...	M2ENot assigned	M3ENot assigned	M4EEast Toronto(The Beaches)	M5EDowntown Toronto(Berczy Park)	M6EYork(Caledonia-Fairbanks)	M7ENot assigned	M8ENot assigned	M9ENot assigned
4	M1GScarborough(Woburn)	M2GNot assigned	M3GNot assigned	M4GEast York(Leaside)	M5GDowntown Toronto(Central Bay Street)	M6GDowntown Toronto(Christie)	M7GNot assigned	M8GNot assigned	M9GNot assigned

Every cell here has three components. Lets take (1,1) 'M1BScarborough(Malvern/Rouge)' in consideration. First three characters signify postal code, M1B. Scarborough is a borough and (Malvern/Rouge) are neighborhoods in postal code M1B.

Data wrangling on the above table fetches the below result.

```python
for j in range(df1.shape[1]):
    for i in range(len(df1)):
        if 'Not assigned' in df1.iloc[i,j]:
            continue
        else:
            pc = df1.iloc[i,j][:3]
            ne = df1.iloc[i,j].rsplit('(')[-1].replace(' /',',').replace(')','')
            br = df1.iloc[i,j][3:].rsplit('(')[0]

        neighborhoods = neighborhoods.append({'Postal Code': pc,
                                              'Bourough': br,
                                              'Neighborhood':ne},ignore_index=True)
```

```python
neighborhoods.shape
```

```
(103, 3)
```

```python
neighborhoods.head()
```

	Postal Code	Bourough	Neighborhood
0	M1B	Scarborough	Malvern, Rouge
1	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek
2	M1E	Scarborough	Guildwood, Morningside, West Hill
3	M1G	Scarborough	Woburn
4	M1H	Scarborough	Cedarbrae

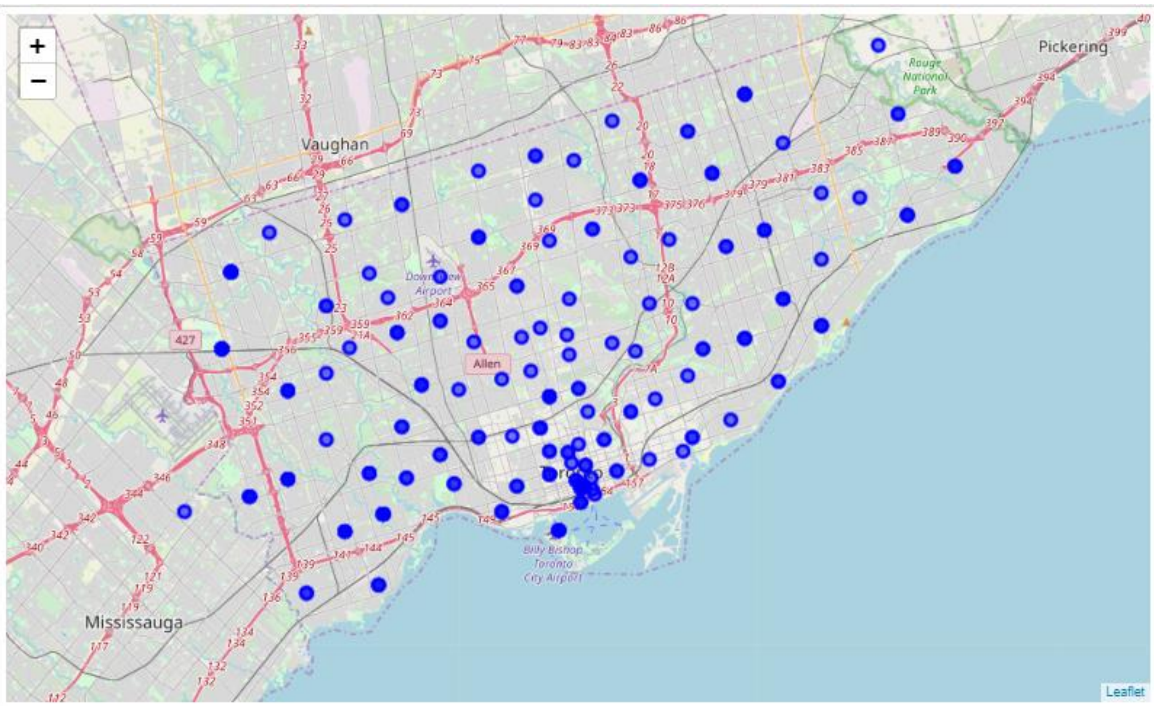
Now for the next part, getting latitudes and longitudes for every neighborhood. I used Geocoder from Geopy. The dataframe with coordinates is ready.

```
import geopy
for i in range(len(neighborhoods)):
    add = neighborhoods.loc[i,'Neighborhood'] + ' Toronto'

    geolocator = geopy.Nominatim(user_agent='ca_explorer')
    location = geolocator.geocode(add)
    #print(add)
    try:
        neighborhoods.loc[i,'Latitude'] = location.latitude
        neighborhoods.loc[i,'Longitude'] = location.longitude
    except:
        continue
```

	Postal Code	Bourough	Latitude	Longitude	Neighborhood
0	M1B	Scarborough	43.806686	-79.194353	Malvern
1	M1B	Scarborough	43.806686	-79.194353	Rouge
2	M1C	Scarborough	43.784535	-79.160497	Rouge Hill
3	M1C	Scarborough	43.784535	-79.160497	Port Union
4	M1C	Scarborough	43.784535	-79.160497	Highland Creek

We can visualize the map of Toronto and its neighborhoods using package Folium.

Out[27]:



Now for the final part, getting lists of all the restaurants and their types for every neighborhood in a 500 meter radius.

I won't go into the nity-gritty details of how I used Foursquare API, but anyone is welcome to post a query in the comment. Here's a snippet for the function that got me my list of restaurants for every neighborhood.

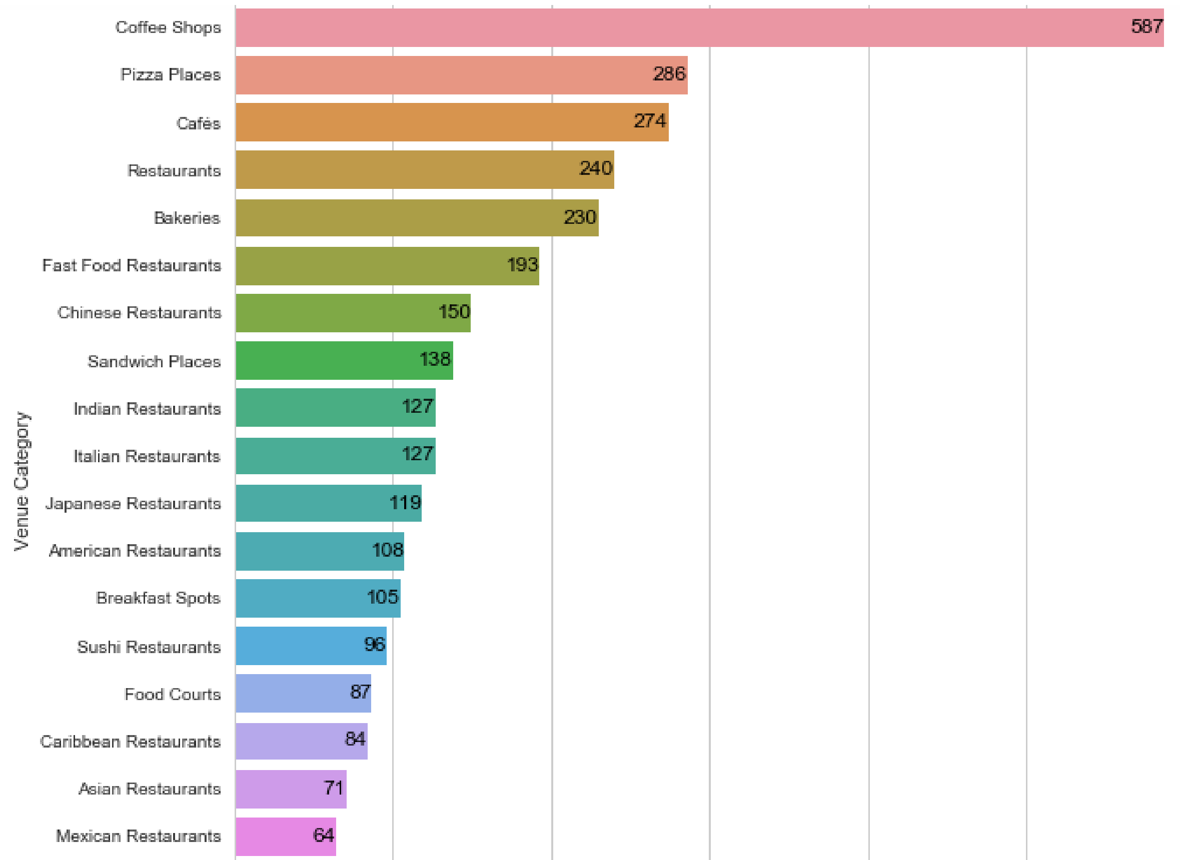
```
def getNearbyResto(names, latitudes, longitudes, radius=500, query='Food', categoryID = '4d4b7105d754a06374d81259'):

    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/search?&client_id={}&client_secret={}&v={}&ll={},{}&q={}&radius={}&limit={
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        lat,
        lng,
        query,
        radius,
        LIMIT,
        categoryID)

        # make the GET request
        results = requests.get(url).json()["response"]['venues']

        # return only relevant information for each nearby venue
        venues_list.append([(
            name,
            lat,
            lng,
            v['name'],
            v['location']['lat'],
            v['location']['lng'],
            v['categories'][0]['pluralName']) 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)
```

```
Toronto_restaurants.head(3)
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Malvern	43.806686	-79.194353	Meena's Fine Foods	43.804476	-79.199753	Indian Restaurants
1	Malvern	43.806686	-79.194353	Wendy's	43.807448	-79.199056	Fast Food Restaurants
2	Malvern	43.806686	-79.194353	Second Cup	43.802165	-79.196114	Coffee Shops

The data set is now complete, I have all the information I need. Neighborhoods, their locations, the list of restaurants in neighborhoods and their category.

On further analysis, I see that there are 141 unique categories of restaurants. Any guesses which one would be the most frequent type?



Not bad for Indian restaurants either, they're 9th most frequent type of restaurants in Toronto.

The data set isn't ready for clustering just yet, I need to convert the categorical column 'Venue Category' to dummy vectors using one hot encoding. Let's check the shape and head of the resulting data frame.

Out[45]:

	Neighborhood	Afghan Restaurants	African Restaurants	American Restaurants	Argentinian Restaurants	Asian Restaurants	Australian Restaurants	BBQ Joints	Bagel Shops	Bakeries	...	Tapas Restaurants	Tea Rooms	Thi Restaurant
0	Malvern	0	0	0	0	0	0	0	0	0	...	0	0	
1	Malvern	0	0	0	0	0	0	0	0	0	...	0	0	
2	Malvern	0	0	0	0	0	0	0	0	0	...	0	0	
3	Rouge	0	0	0	0	0	0	0	0	0	...	0	0	
4	Rouge	0	0	0	0	0	0	0	0	0	...	0	0	

5 rows × 142 columns

In [46]:
```
Toronto_onehot.shape
```

Out[46]: (4839, 142)

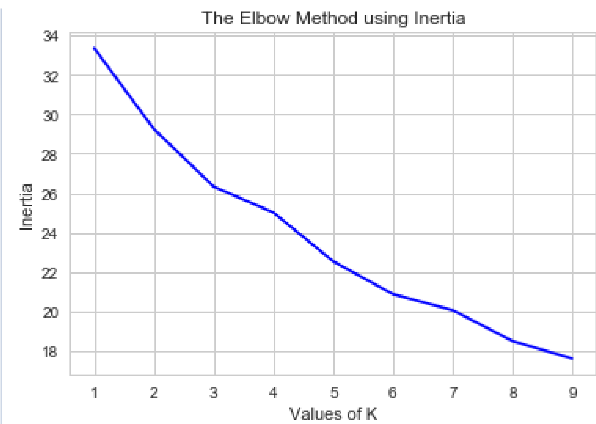
142 variables? most of them with no information for every row, that's one too many.

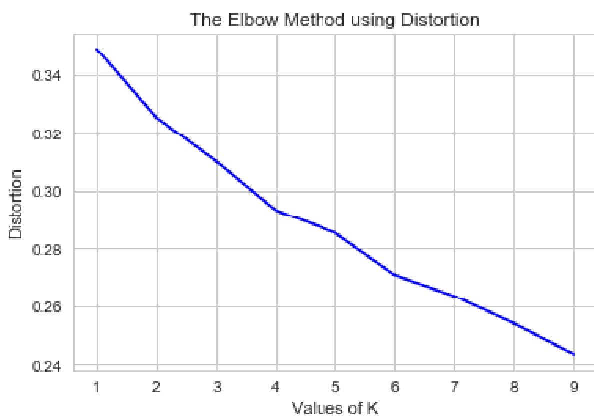
I will find top 10 most common restaurant types for every neighborhood. I get below results.

```
neighborhoods_venues_sorted.head(11)
```

	Neighborhood	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	9th Most Common Venue	10th Most Common Venue
0	Adelaide	Coffee Shops	Food Courts	Restaurants	American Restaurants	Cafés	Ramen Restaurants	Bars	Fast Food Restaurants	Japanese Restaurants	Vegetarian / Vegan Restaurants
1	Agincourt North	Chinese Restaurants	BBQ Joints	Asian Restaurants	Bakeries	Fast Food Restaurants	Pizza Places	Dumpling Restaurants	Coffee Shops	Food Courts	Chinese Breakfast Places
2	Albion Gardens	Pizza Places	Bakeries	Caribbean Restaurants	Indian Restaurants	Sandwich Places	Fast Food Restaurants	Chinese Restaurants	Coffee Shops	Bubble Tea Shops	Food Trucks
3	Bathurst Quay	Coffee Shops	Bars	American Restaurants	Tapas Restaurants	Empanada Restaurants	Egyptian Restaurants	Eastern European Restaurants	Dumpling Restaurants	Donut Shops	Diners
4	Beaumond Heights	Pizza Places	Bakeries	Caribbean Restaurants	Indian Restaurants	Sandwich Places	Fast Food Restaurants	Chinese Restaurants	Coffee Shops	Bubble Tea Shops	Food Trucks
5	Bloordale Gardens	Pizza Places	Fried Chicken Joints	Coffee Shops	Mediterranean Restaurants	Cafés	Dessert Shops	Dumpling Restaurants	Donut Shops	Diners	Dim Sum Restaurants
6	Cabbagetown	Cafés	Coffee Shops	Pizza Places	Restaurants	Breakfast Spots	Chinese Restaurants	BBQ Joints	Japanese Restaurants	Thai Restaurants	Gastropubs
7	Chinatown	Coffee Shops	Chinese Restaurants	Cafés	Vietnamese Restaurants	Dumpling Restaurants	Bubble Tea Shops	Fast Food Restaurants	Pizza Places	Ramen Restaurants	Burger Joints

Alright, time to cluster these neighborhoods and find out areas of interest. Though, how many clusters should I make? I ran Kmeans in range(1:10) and visualized the elbow point in inertia and distortion. Using below two charts, the elbow point isn't very clear but if I had to pick, I'd pick K = 4.





Here are the four clusters on a map,

Out[62]:

Results and Recommendation

Clusters are ready, lets examine each of them.

Cluster 0 - Percentage of Indian restaurants in cluster 0 is 30%. This number is pretty high, so I further hypothesized that maybe residents of this cluster are inclined towards food that uses similar ingredients or spice levels. As someone from Indian origin myself, I think whoever likes Indian food, generally likes Mexican and Chinese as well. Let's see if these types are in high number as well.

In [68]:

```
Indian_resto_count_0=0
for i in range(3,len(cluster0.columns)):
    Indian_resto_count_0= Indian_resto_count_0 + cluster0[cluster0.columns[i]].str.count('Indian Restauran
print('Indian, Chinese and Mexican Restaurants in Cluster 0 are ',Indian_resto_count_0)
```

```
Indian, Chinese and Mexican Restaurants in Cluster 0 are  102
```

In [69]:

```
print('Percentage of Indian, Chinese and Mexican Restaurants restaurants in Cluster 0 is {0:.2f}%'.format(
```

```
Percentage of Indian, Chinese and Mexican Restaurants restaurants in Cluster 0 is 85.00%
```

As expected, people in cluster 0 are inclined towards aforementioned cuisines, now these neighborhoods might already be saturated with enough options for people looking to dine in an Indian restaurants, so far these neighborhoods don't seem like an ideal option

Cluster 1 - Cluster 1 has 0% of Indian, Chinese and Mexican restaurants. Either it is an untapped market for such cuisines or maybe people simple don't have the taste buds for them. Let's look at other options.

Cluster 2 - It is the same as Cluster 1, No Indian, Mexican or Chinese Restaurants here.

Cluster 3 - Cluster 3 has 18% Indian restaurants and 36% Mexican, Chinese and Indian restaurants cumulative. This seems like the cluster of interest, suggesting people have the taste bud for these cuisines and the cluster isn't already saturated with options for customer like cluster 0.

Finally let's narrow down to the list of neighborhoods in cluster 3 that don't already have Indian restaurant as one of the top 10 most common venues.

[92]:

```
NPrefer=[]
for j in range(len(cluster3)):
    if cluster3.iloc[j,:].str.contains('Indian Restaurants').any():
        continue
    else:
        NPrefer.append(cluster3.iloc[j,1])
print(NPrefer)
```

```
['Rouge Hill', ' Port Union', ' Highland Creek', 'Agincourt', 'Hillcrest Village', 'Parkwoods', 'CFB Toronto', 'DownsviewWest',
'Victoria Village', 'The Danforth West', ' Riverdale', 'Moore Park', ' Summerhill East', 'Rosedale', 'St. James Town', 'Church
and Wellesley', 'Regent Park', ' Harbourfront', 'Garden District', ' Ryerson', 'St. James Town', 'Berczy Park', 'Central Bay St
reet', 'Richmond', ' Adelaide', ' King', 'Harbourfront East', ' Union Station', ' Toronto Islands', 'Toronto Dominion Centre',
' Design Exchange', 'Commerce Court', ' Victoria Hotel', 'Kensington Market', ' Chinatown', ' Grange Park', 'CN Tower', ' King
and Spadina', ' Railway Lands', ' Harbourfront West', ' Bathurst Quay', ' South Niagara', ' Island airport', 'Enclave of M5E',
'First Canadian Place', ' Underground city', 'Little Portugal', ' Trinity', 'Brockton', ' Parkdale Village', ' Exhibition Plac
e', "M7AQueen's Park", ' Ontario Provincial Government', 'Mimico NW', ' The Queensway West', ' South of Bloor', ' Kingsway Park
South West', ' Royal York South West']
```

Limitation and scope for further research

Here I only take into consideration the occurrence and frequency of Indian Restaurants in the neighborhood to drive the insights. There are numerous more factors that can affect the decision to open a new restaurant such as population density of the neighborhood, ethnicity, income of residents, real estate prices, etc. Here I'm relying solely on an assumptions while deciding on the clusters due to lack of information about the demographics. It is very much possible that cluster 1 and 2 have people of Asian and South Asian cultures, in that case those clusters would be the best bet to invest.

Conclusion

In this project, we have gone through the process of identifying the business problem, specifying the data required, extracting and preparing the data, performing the machine learning by utilizing k-means clustering and providing recommendation to the stakeholder.