Predicting the Best Place for Chinese Restaurant in Toronto

Steve Daniels

November 27, 2020

1. Introduction

1.1 Background

Chinese cuisine is a great food for people in Toronto, especially during winter season, because it contained various spices. In reality, there are only small number of restaurant that served Chinese cuisine there. So, it is a good idea to open an Chinese restaurant in Toronto. It is quite challenging to find a place or area to open the Chinese restaurant. This project will help the enterpreneur to find the most suitable location.

1.2 Problem

The main problem is to find the most suitable location based on the density of Chinese restautant in the area.

1.3 Interest

Enterpreneur(s) who wants to open Chinese restaurant in Toronto, Canada.

2. Data

2.1 Data needed

Data needed for this project are shown below :

- List of neighborhood in Toronto, Canada

	Postal Code	Borough	Neighbourhood
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

- Latitude and Longitude data for every neighborhood in Toronto, Canada

	Postal Code	Latitude	Longitude
0	M1B	43.806686	-79.194353
1	M1C	43.784535	-79.160497
2	M1E	43.763573	-79.188711
3	M1G	43.770992	-79.216917
4	M1H	43.773136	-79.239476

- Venue data related to restaurant in neighborhoods of Toronto, Canada

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	The Beaches	43.676357	-79.293031	Glen Manor Ravine	43.676821	-79.293942	Trail
1	The Beaches	43.676357	-79.293031	The Big Carrot Natural Food Market	43.678879	-79.297734	Health Food Store
2	The Beaches	43.676357	-79.293031	Grover Pub and Grub	43.679181	-79.297215	Pub
3	The Beaches	43.676357	-79.293031	Upper Beaches	43.680563	-79.292869	Neighborhood
4	The Beaches	43.676357	-79.293031	Seaspray Restaurant	43.678888	-79.298167	Asian Restaurant

2.2 Data Extraction

The extraction of data needed are shown below:

- Scrapping data of Toronto neighborhoods via Wikipedia and stored into dataframe

```
[ ]  url  = "https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M"
     page = requests.get(url)
```

```
[ ]  df_html = pd.read_html(url, header=0, na_values = ['Not assigned'])[0]
     df_html.head()
```

- Getting location coordinates via Geospacial Data given by Coursera

```
[ ]  url_csv = 'http://cocl.us/Geospatial_data'
     df_coordinates = pd.read_csv(url_csv)
```

- Getting the venue data via API call to FourSquare API

```
[ ]  def getNearbyVenues(names, latitudes, longitudes, radius=500):

        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']
```

3. Methodology

3.1 Scrapping Toronto Neighborhoods Data

The Toronto Neighborhood data was scrapped from Wikipedia with pandas library in Python.

```
url  = "https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M"
df_html = pd.read_html(url, header=0, na_values = ['Not assigned'])[0]
df_html.head()
```

	Postal Code	Borough	Neighbourhood
0	M1A	NaN	NaN
1	M2A	NaN	NaN
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Regent Park, Harbourfront

3.2 Cleaning Toronto Neighborhoods Data

The Toronto neighborhood contained “NaN” value in Borough and Neighborhood. Delete all rows contained “NaN” values of Borough and Neighborhood.

```
[ ] df_html.dropna(subset=['Borough'], inplace=True)
    df_html.head()
```

	Postal Code	Borough	Neighborhood
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Regent Park, Harbourfront
5	M6A	North York	Lawrence Manor, Lawrence Heights
6	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government

Group the dataframe with the same borough.

```
[ ] df_postcodes = df_html.groupby(['Postal Code','Borough']).Neighborhood.agg([('Neighbourhood', ', '.join)])
    df_postcodes.reset_index(inplace=True)
    df_postcodes.head(5)
```

	Postal Code	Borough	Neighbourhood
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

Save the cleaned Toronto neighborhood data into csv file

```
[ ] df_postcodes.to_csv("torontodata.csv")
```

3.3 Combining Cleanned Data with Geospacial Data

Call the geospacial data with the link given in Coursera and store into dataframe

```
[ ] url_csv = 'http://cocl.us/Geospatial_data'
    df_coordinates = pd.read_csv(url_csv)
```

Call the Toronto Neighborhood data from csv file and store into dataframe

```
[ ]  df_neighborhoods = pd.read_csv("torontodata.csv",index_col=[0])
     df_neighborhoods.head()
```

Merge both datasets with Pandas

```
[ ]  df_neighborhoods_coordinates = pd.merge(df_neighborhoods, df_coordinates, on='Postal Code')
     df_neighborhoods_coordinates.head()
```

	Postal Code	Borough	Neighbourhood	Latitude	Longitude
0	M1B	Scarborough	Malvern, Rouge	43.806686	-79.194353
1	M1C	Scarborough	Rouge Hill, Port Union, Highland Creek	43.784535	-79.160497
2	M1E	Scarborough	Guildwood, Morningside, West Hill	43.763573	-79.188711
3	M1G	Scarborough	Woburn	43.770992	-79.216917
4	M1H	Scarborough	Cedarbrae	43.773136	-79.239476

Filter the data only with the Borough contained “Toronto” value

```
[ ]  df_toronto = dfa[dfa['Borough'].str.contains('Toronto')]
     df_toronto.reset_index(inplace=True)
     df_toronto.drop('index', axis=1, inplace=True)
     df_toronto.tail()
```

```
[ ]  print(df_toronto.groupby('Borough').count()['Neighbourhood'])
```

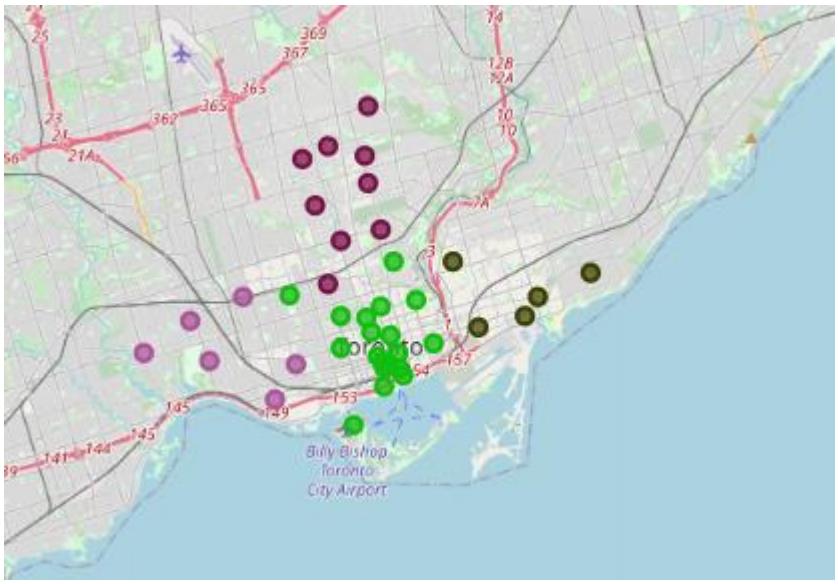
```
Borough
Central Toronto      9
Downtown Toronto    19
East Toronto         5
West Toronto         6
Name: Neighbourhood, dtype: int64
```

Show the coordinates of toronto

```
[ ]  lat_toronto = df_toronto['Latitude'].mean()
     lon_toronto = df_toronto['Longitude'].mean()
     print('The geographical coordinates of Toronto are {}, {}'.format(lat_toronto, lon_toronto))
```

```
The geographical coordinates of Toronto are 43.66713498717948, -79.38987324871795
```

Show the map of Toronto with Folium



3.4 Using Foursquare API to Explore Neighborhood

Define the credential and create a function to call Foursquare API

```
[ ]  CLIENT_ID = #inputclientid
     CLIENT_SECRET = #inputclientsecret
     VERSION = #YYYYMMDD
     LIMIT = 100
     radius = 500
```

Get the top 100 venues within 500m radius

```
[ ]  def getNearbyVenues(names, latitudes, longitudes, radius=500):

         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)
```

```
[ ]  toronto_venues = getNearbyVenues(names=df_toronto['Neighbourhood'],
                                      latitudes=df_toronto['Latitude'],
                                      longitudes=df_toronto['Longitude'])
```

Store the data from Foursquare API to dataframe

```
[ ] toronto_venues.to_csv("APIfq1.csv")
```

```
[ ] df_hasilfq = pd.read_csv("APIfq1.csv",index_col=0)
    df_hasilfq
```

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	The Beaches	43.676357	-79.293031	Glen Manor Ravine	43.676821	-79.293942	Trail
1	The Beaches	43.676357	-79.293031	The Big Carrot Natural Food Market	43.678879	-79.297734	Health Food Store
2	The Beaches	43.676357	-79.293031	Grover Pub and Grub	43.679181	-79.297215	Pub
3	The Beaches	43.676357	-79.293031	Upper Beaches	43.680563	-79.292869	Neighborhood
4	The Beaches	43.676357	-79.293031	Seaspray Restaurant	43.678888	-79.298167	Asian Restaurant

Create a dataframe of neighborhoods and Chinese Restaurant occurence.

```
to_chin = to_grouped[["Neighborhoods","Chinese Restaurant"]]
to_chin.head()
```

	Neighborhoods	Chinese Restaurant
0	Berczy Park	0.0
1	Brockton, Parkdale Village, Exhibition Place	0.0
2	Business reply mail Processing Centre, South C...	0.0
3	CN Tower, King and Spadina, Railway Lands, Har...	0.0
4	Central Bay Street	0.0

3.5 Clustering

Use the Kmeans for clustering. Kmeans used for clustering because this project aimed to find the best location based on density of neighborhood with restaurant. Kmeans clustering will use coordinates data from filtered neighborhood data.

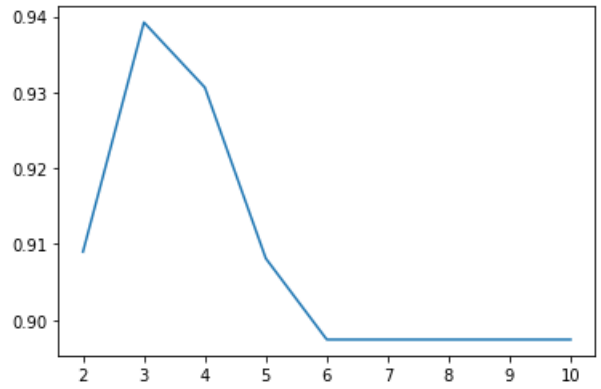
Prepare the dataframe for fitting in Kmeans clustering

```
X = to_chin.drop(['Neighborhood'], axis=1)
```

In order to find the best number of K, simulate Kmeans with K value ranged from 2-10, and check the silhouette score to decide number or K

```
sil = []
kmax = 10
# dissimilarity would not be defined for a single cluster, thus, minimum number of clusters should be 2
for k in range(2, kmax+1):
  kmeansx = KMeans(n_clusters = k).fit(X)
  labels = kmeansx.labels_
  sil.append(silhouette_score(X, labels, metric = 'euclidean'))

plt.plot (list(range(2,11)),sil)
```

```
[<matplotlib.lines.Line2D at 0x7f68f98f9748>]
```



From the graph shown above, K=3 showed the best result based on Silhouette Score, so later K=3 will be used for Kmeans clustering

```
# set number of clusters
K = 3

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

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

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

Add the cluster label to dataframe

```
to_merged = to_chin.copy()

# add clustering labels
to_merged["Cluster Labels"] = kmeans.labels_
to_merged.tail()
```

	Neighborhood	Chinese Restaurant	Cluster Labels
34	The Annex, North Midtown, Yorkville	0.00	0
35	The Beaches	0.00	0
36	The Danforth West, Riverdale	0.00	0
37	Toronto Dominion Centre, Design Exchange	0.01	2
38	University of Toronto, Harbord	0.00	0

Merge with another dataframe to get latitude/longitude for each neighborhood

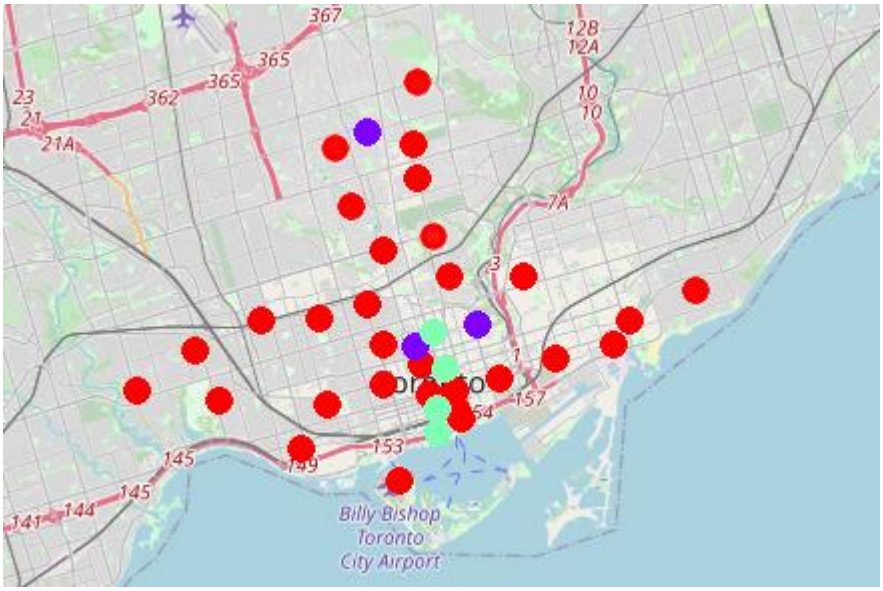
```
# merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
to_merged = to_merged.join(df_hasilfq.set_index("Neighborhood"), on="Neighborhood")

print(to_merged.shape)
to_merged.head()
```

```
(1639, 9)
```

	Neighborhood	Chinese Restaurant	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Berczy Park	0.0	0	43.644771	-79.373306	The Keg Steakhouse + Bar - Esplanade	43.646712	-79.374768	Restaurant
0	Berczy Park	0.0	0	43.644771	-79.373306	LCBO	43.642944	-79.372440	Liquor Store
0	Berczy Park	0.0	0	43.644771	-79.373306	Fresh On Front	43.647815	-79.374453	Vegetarian / Vegan Restaurant
0	Berczy Park	0.0	0	43.644771	-79.373306	Meridian Hall	43.646292	-79.376022	Concert Hall
0	Berczy Park	0.0	0	43.644771	-79.373306	Goose Island Brewhouse	43.647329	-79.373541	Beer Bar

3.6 Clustering Visualization

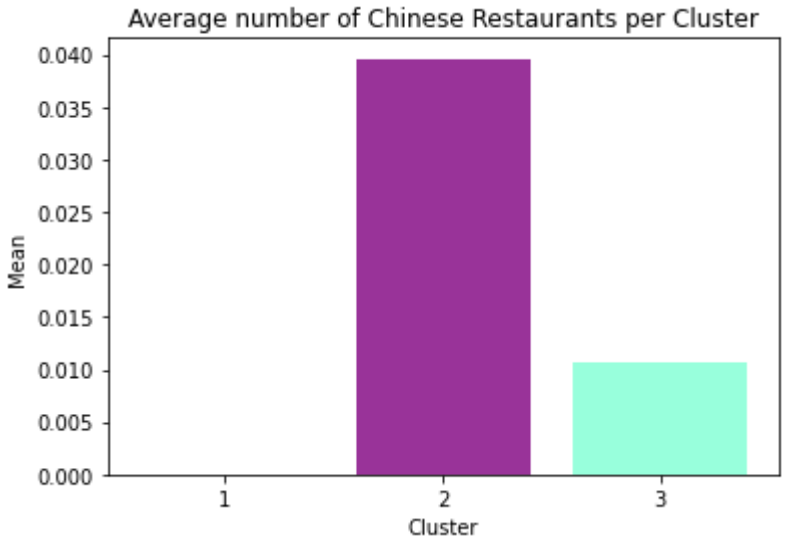
For the visualization, use the folium to show the clusters with different color each cluster



3.8 Clusters Examination

```
to_chin['Cluster Labels'].value_counts()
```

```
0    32
2     4
1     3
Name: Cluster Labels, dtype: int64
```



Cluster 1 = Label 0
Cluster 2 = Label 1
Cluster 3 = Label 2

4. Conclusion and Recommendation

The conclusion is, the best location to open Chinese Restaurant is in cluster 1 around The Beaches, Stn A PO Boxes, Central Bay Street, and other location located in cluster 1 because cluster 1 has the least dense Chinese restaurant in the area. There are only several Chinese Restaurant in Toronto, so I think it is a good idea to present restaurant that serve Chinese cuisine, especially in neighborhood without Chinese Restaurant to eliminate competition.