Most suitable place to open a restaurant a South Asian (particularly Bangladeshi) restaurant in Toronto Pedal Cycle Accident Severity

Qazi Mussabbir

9th Dec 2020

Contents

Figures

1. Introduction

1.1 Background

Toronto is one of the most cosmopolitan cities in the world. It has a very diverse population with over 160 difference languages spoken. On the global front, it is a very important business and financial hub. Each year Canada, and in particular Toronto draws an increased number of immigration from across the world due it's open and liberal approach to welcoming immigrants in search for better lives. As of 2016, Canada has 1,963,330 Canadians with South Asian geographical origins, constituting 5.6% of the Canadian population and 32% of Canada's Asian Canadian population. The population of South Asians in Toronto in 2016 stood at 995,125 and continuing to increase every year. In particular, the Bangladeshi community is steadily growing and has a population base of close 40,000. Many of the newly arrived immigrants from Bangladesh are very entrepreneurial and looking at ways to establish a successful business by establishing a South Asian restaurant, serving Bangladeshi cuisine in particular by identifying a particular gap in the market. In this assignment, we use location based data (from Foursquare), to identify the best borough and neighborhood to set up a Bangladeshi restaurant.

1.2 Problem

Setting up any new business comes with its own risk. In a bustling city like Toronto, the food scene can be very competitive, and opening a restaurant is a moderately risky venture. It will be prudent for any budding restaurateur, to seek data driven insights to understand the right location to setup the desired type of restaurant by looking at a numbers of parameters like cuisine concentration, trends, reviews and proximity to other revenues like malls, cinemas, etc. The project aims to develop a clustering model for identifying the right borough and neighborhood for setting up a Bangladeshi restaurant.

1.3 Interested Parties

The output of the model can be used by any entrepreneur for identifying the right location for setting up a Bangladeshi restaurant in Toronto. Additionally, in the future, the data collected could lead to insights that could be used by South Asian communities for lifestyle purposes for finding various venue information like restaurants, cafes, spiritual centres, clothing stores etc.

2. Data acquisition and cleaning

2.1 Data sources

The dataset used for this project is taken from a Wikipedia page (https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M) which contains a table consisting of all the postal codes, boroughs and neighbourhoods in Toronto. The required data is scraped using a web scraper tool called 'Beautiful Soup' and converted into a *pandas* dataframe like the one shown below

	Postal Code	Borough	Neighbourhood
0	M1A	Not assigned	Not assigned
1	M2A	Not assigned	Not assigned
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Regent Park, Harbourfront
...	...	...	...
175	M5Z	Not assigned	Not assigned
176	M6Z	Not assigned	Not assigned
177	M7Z	Not assigned	Not assigned
178	M8Z	Etobicoke	Mimico NW, The Queensway West, South of Bloor,...
179	M9Z	Not assigned	Not assigned

180 rows × 3 columns

Figure 1: Dataframe composed from the scraped data

The primary objective of the data is to provide a tabular view on all the post codes, boroughs and neighborhood across Toronto. The dataframe has 180 rows and 3 columns as can be seen from the figure above.

2.2 Data cleaning

The datasets in its original form needed cleaning and transformation to be rendered fit for use by a model. To start, there were quite a few 'Not assigned' rows across the 'Borough' and 'Neighbourhood' columns. For each of the 'Not assigned' values for Borough, there are 77 'Not assigned' values for Neighbourhood column too**.** In total there were 77 rows which were dropped.

```
df[df['Borough'] == 'Not assigned'].count()
```

```
Postal Code      77
Borough          77
Neighbourhood    77
dtype: int64
```

```
In [21]: #Dropping rows in Borough column with 'Not assigned' values

         df.drop(df[df['Borough'] == 'Not assigned'].index, inplace = True)
```

```
In [13]: #checking if all the 'Not assigned' rows have indeed been dropped
         df[df['Borough'] == 'Not assigned']
```

Out[13]:

Postal Code	Borough	Neighbourhood

Figure 2: Dropped rows containing 'Not assigned' values

It was mentioned in the assignment questionnaire that If a cell has a borough but a 'Not assigned' neighborhood, then the neighborhood will need to be the same as the borough and in this respects visual checks on the dataframe were done and no values equal to 'Not assigned' were found to

meet this condition. However, the following code was executed to ensure that if there were any 'Not assigned' neighborhoods, then the neighborhood will equal to the name of the borough.

```
#If a cell has a borough but a Not assigned neighborhood, then the neighborhood will be the same as the borough
for a in df_group['Borough']:
    if a != 'Not assigned':
        Borough_na = a
        #print(a)
        for i in df_group['Neighbourhood']:
            #print(i)
            y = 0
            #neighbourhood_na = i
            #print(neighbourhood_na)
            if i == 'Not assigned':
                #print("I am here")
                neighbourhood_na = Borough_na
                #print(i)
                df_group.loc[y]['Neighbourhood'] = neighbourhood_na
                print(df_group.loc[y]['Neighbourhood'])
                y += 1
```

Figure 3: Assigning 'Non assigned' neighbourhoods with valid values

2.2.1Dealing with duplicate data (combining data rows)

As mentioned in the assignment questionnaire more than one neighborhood can exist in one postal code area. For example, it is mentioned that M5A is listed twice and has two neighborhoods: Harbourfront and Regent Park. However, these two rows are already combined into one row with the neighborhoods separated with a comma. In order to ensure that there were no other duplicate values for the 'Postal Code' column, the following test was executed

```
: pd.Series(df['Postal Code']).is_unique
: True
: df[df['Postal Code'] == 'M5A']
:
```

	Postal Code	Borough	Neighbourhood
4	M5A	Downtown Toronto	Regent Park, Harbourfront

Figure 4: Checks for identifying any duplicate values for Postal Code

2.2.2 Geospatial data for Toronto

For the purpose of the getting the latitude and longitude data for all the neighbourhoods in Toronto, a csv file that has the geographical coordinates of each postal code from the link http://cocl.us/Geospatial_data has been used. The data from the csv file is converted to a pandas dataframe, which is then merged using a 'join' operation with the Toronto neighbourhood dataframe as seen below:

```
#Exploring the geospatial dataframe with the grouped dataframe
df_group_merged = pd.merge(df_group,df_geo)
```

```
df_group_merged
```

	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
...	...	...	...	...	...
98	M9N	York	Weston	43.706876	-79.518188
99	M9P	Etobicoke	Westmount	43.696319	-79.532242
100	M9R	Etobicoke	Kingsview Village, St. Phillips, Martin Grove ...	43.688905	-79.554724
101	M9V	Etobicoke	South Steeles, Silverstone, Humbergate, Jamest...	43.739416	-79.588437
102	M9W	Etobicoke	Northwest, West Humber - Clairville	43.706748	-79.594054

103 rows × 5 columns

Figure 5: Merged dataframe consisting for Geospatial data

2.2.3 Venue data from Four Square

In order to understand the potential for setting up a new restaurant in a particular area, one needs to analyze the different types of venues near a particular neighbourhood like popular restaurants, cafes, bars, social clubs, venue trends etc. For retrieving such venue information, Four Square APIs will be used. In order to use the Four Square API, a developer account has been created with necessary credentials. The response from the API call is a json file which contains the requested venue data.

```
CLIENT_ID = 'DJ3WT0BU4G01WASVHSNKMJME24MVFZNPXUGULBP30MRNXZAL' # your Foursquare ID
CLIENT_SECRET = 'VHRAWJ3PLFPUSI5HEPHUXQSQU510SFS5PSVG05VCGFXEFRKV' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version
LIMIT = 100 # A default Foursquare API limit value

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

```
Your credentails:
CLIENT_ID: DJ3WT0BU4G01WASVHSNKMJME24MVFZNPXUGULBP30MRNXZAL
CLIENT_SECRET:VHRAWJ3PLFPUSI5HEPHUXQSQU510SFS5PSVG05VCGFXEFRKV
```

Figure 6: Four Square developer account and credentials