**Capstone project: Battle of neighbourhoods; final presentation.**

**Presenter: Sri Datta Boggaram**

**A: Introduction/Business Problem**

**Introduction:** Indians are among the largest immigrants to Canada. Generally people migrating to new place will look for a decently paid job , then the locality they live in as to what it offers such as security, frequency of public of transportation, grocery stores, restaurants fitness centres, schools etc.,

This study aims to filter out a place for new migrants moving to Toronto who are strict about their physical routine, including swimming, and are looking to settle near by a grocery store or an Indian restaurant.

**Data acquisition and cleaning:**

There are different sources from the data was collected for different purpose.

1. List of postal Codes for Canada:-

* Fetched the postal code of the neighbourhoods in Canada from Wikipedia.
* Link — <https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M>

2. Geographical Co-ordinates:-

* A CVS file which contains latitude and longitude of the neighborhoods in Canada, Toronto, instead of geocoder is used, as the data sometimes is inconsistent while using live.
* Link for CVS — [http://cocl.us/Geospatial\_data](https://cocl.us/Geospatial_data)

3. Fetching Details of the venue:

* Foursquare API for fetching the details and location of the venues.
* Venue ratings as a benchmark.
* Visualization using Folium.
* From Foursquare API (<https://developer.foursquare.com/docs>),

4. Following data is retrieved the following for each venue:

* Name: The name of the venue.
* Category: The category type as defined by the API.
* Latitude: The latitude value of the venue.
* Longitude: The longitude value of the venue.
* Likes: Likes of the venue, that the user liked the restaurant.
* Rating: Rating of the venue.
* Tips: Tips given by the users.

**Data Cleaning**

Cleaning the Postal Code data

* The data frame will consist of three columns: Postal Code, Borough, and Neighborhood
* Only process the cells that have an assigned borough. Ignore cells with a borough that is ‘**Not assigned’.**
* More than one neighborhood can exist in one postal code area. For example, in the table on the Wikipedia page, you will notice that **M5A** is listed twice and has two neighborhoods: **Harbourfront**and **Regent Park**. These two rows will be combined into one row with the neighborhoods separated with a comma.
* If a cell has a borough but a ‘**Not assigned’**neighborhood, then the neighborhood will be the same as the borough.

**Methodology in detail results discussion:**

* The required data was Scraped from Wikipedia and converted to a data frame initially.
* Then thus obtained data was cleaned into a more refined data frame by removing all the null values and redundant values in the rows and this stage can be termed as data cleaning per se and then the data frames obtained after cleaning are collapsed into one data frame. The code is depicted below.

#### Collapse the data

**df\_merge = pd.merge(df, temp\_df, on='Postalcode')**

**df\_merge.drop(['Neighborhood'],axis=1,inplace=True)**

**df\_merge.drop\_duplicates(inplace=True)**

**df\_merge.rename(columns={'Neighborhood\_joined':'Neighborhood'},inplace=True)**

**df\_merge.head()**

* Adding latitude and longitude coordinates to the above data frame gives a scope to visualize the data frame if need be.

**from** **geopy.geocoders** **import** Nominatim

**def** get\_geocode(postal\_code):

#### Then the above data frame was cleaned to obtain boroughs that contain word Toronto in the name, as we are looking at obtaining a neighbourhood around the Downtown, so finally 40 boroughs are filtered to be looked at for the desirable location.

#### The foursquare data is used to data frame out the venues in the chosen neighbourhood.

**CLIENT\_ID = 'QEF4SFROUUVOQKNPAFNNRDUW4ACAWSYYG312LM3BKEDDKPIZ' *# Foursquare ID***

**CLIENT\_SECRET = 'CCBFW1WDVTUYDLPFX3SR0QBB5R2UKRYOZ3F1JDJ3PLQMPDRM' *# Foursquare Secret code***

**VERSION = '20201004’**

* **Finding out the number of unique categories** **and carving out Indian restraint and pool values from the rows and forming a new data frame.**

print('There are **{}** uniques categories.'.format(len(toronto\_venues['Venue\_Category'].unique())))

df.Indian\_Restaurant=toronto\_venues[toronto\_venues.Venue\_Category=="Indian Restaurant"]

**Same process for the pool**

df.SwimmingPool=toronto\_venues[toronto\_venues.Venue\_Category=="Pool"

#### Then a new data frame is made the above two dfs

df.Result = df.Indian\_Restaurant.append( df.SwimmingPool, sort=**False**)

df.Result.head(50)

#### Creating a new data frame or just the neighbourhood and venue category

#### df.new\_result = df.Result[['Neighborhood','Venue\_Category']] df.new\_result.head(50)

* **Prepere a new datafarame that has largest value in the above data**

df.neighbourhood = df.Result.loc[df.Result['Neighborhood'] == 'India Bazaar, The Beaches West']

df.neighbourhood.head(20)



#### Discussion:

#### Machine learning was not incorporated as the need to employ one did not occur. All the data cleaning, data wrangling, faming new data sets from the previously made was done suing ‘pandas’ alone. Foursquare data was just used to add the venue locations to the data frame in order to find out what places of interest were present in the given neighbourhood.

#### Thus, by merely applying Pandas, and other simple python packages, like geocoder, the desired result was obtained.

#### Conclusion:

#### From the above data frame India Bazaar, The Beaches West has 8 Indian restaurants and 1 pool and thus we can conclude the neighbourhood " India Bazaar, The Beaches West " is the ideal place for our new immigrant to settle in...