Capstone Project - The Battle of Neighborhoods

Thai restaurants in Toronto

Introduction

Thai immigrants to Canada are primarily well-educated professionals working as bankers, dentists, doctors, nurses, computer technicians and engineers. Many of them closed businesses such as Thai restaurants or Muay Thai gyms, which has become popular in Canada.

Thai cooking places emphasis on lightly prepared dishes with strong aromatic components and a spicy edge. Thai chef McDang characterises Thai food as demonstrating "intricacy; attention to detail; texture; color; taste; and the use of ingredients with medicinal benefits, as well as good flavor", as well as care being given to the food's appearance, smell and context.

So as part of this project, we will list and visualize all major parts of Toronto that has poppular Thai restaurants.

Data

For this project we need the following data :

- Toronto,CA data that contains list Boroughs, Neighborhoods along with their latitude and longitude.
    - Data source : https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M and get latitude,longitude from https://cocl.us/Geospatial_data'
    - Description : This data set contains the required information. And we will use this data set to explore various neighborhoods of new york city.
- Thai resturants in each neighborhood of Toronto city.
    - Data source : Fousquare API
    - Description : By using this api we will get all the venues in each neighborhood. We can filter these venues to get only indian resturants.
- GeoSpace data
    - Data source : Generating GeoJSON File for Toronto FSAs https://medium.com/dataexplorations/generating-geojson-file-for-toronto-fsas-9b478a059f04
    - Description : By using this geo space data we will get the Toronto Borough boundaries that will help us visualize choropleth map.

Approach

- Collect the postal code,borough,neighborhood of Toronto city data from wikipedia

- Collect latitude,longitude data from https://cocl.us/Geospatial_data
- Using FourSquare API we will find all venues for each neighborhood.
- Filter out all venues that are Thai Resturants.
- Find rating , tips and like count for each Thai Resturants using FourSquare API.
- Using rating for each resturant , we will sort that data.
- Visualize the Ranking of neighborhoods using folium library(python)

Defined the problems

As a Thai food business, seeing the picture in business for Thai restuarnats, so we would like to find out about the business </br> -Best Places in Toronto city for Thai cuisine </br> -which areas have the potential to market for Thai Resturant </br> -which is the best place to stay if I like Thai food

Stockholder

Owner Thai Cusine.
Some one who want to eat Thai Food.

Analysis

We will import the required libraries for python.

- pandas and numpy for handling data.
- request module for using FourSquare API.
- folium to visualize the results on a map

In [ ]:
```
import pandas as pd
import numpy as np
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
import requests
from bs4 import BeautifulSoup
import os
import folium # map rendering library
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors
from geopy.geocoders import Nominatim
%matplotlib inline


print('Libraries imported.')
```
In [ ]:
```
CLIENT_ID = '' # your Foursquare ID
CLIENT_SECRET = '' # your Foursquare Secret
VERSION = '20180604'
LIMIT = 100
radius = 1000
```

```
category='4d4b7105d754a06374d81259'
print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET:' + CLIENT_SECRET)
```

We define a function to intract with FourSquare API and get top 100 venues within a radius of 1000 metres for a given latitude and longitude. Below function will return us the venue id , venue name and category.

In [ ]:

```
def geo_location(address):
    # get geo location of address
    geolocator = Nominatim(user_agent="ny_explorer")
    location = geolocator.geocode(address)
    latitude = location.latitude
    longitude = location.longitude
    return latitude,longitude
```

In [ ]:

```
def get_venues(lat,lng):


    #url to fetch data from foursquare api
    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,
            category)

    # get all the data
    results = requests.get(url).json()
    venue_data=results["response"]['groups'][0]['items']
    venue_details=[]
    for row in venue_data:
        try:
            venue_id=row['venue']['id']
            venue_name=row['venue']['name']
            venue_category=row['venue']['categories'][0]['name']
            venue_details.append([venue_id,venue_name,venue_category])
        except KeyError:
            pass

    column_names=['ID','Name','Category']
    df = pd.DataFrame(venue_details,columns=column_names)
    return df
```

Now we will define a function to get venue details like like count , rating , tip counts for a given venue id. This will be used for ranking.

In [ ]:
```
def get_venue_details(venue_id):
    #url to fetch data from foursquare api
    url =
'https://api.foursquare.com/v2/venues/{}?&client_id={}&client_secret={}&v={}'
.format(
            venue_id,
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION)

    # get all the data
    results = requests.get(url).json()

    venue_data=results['response']['venue']
    venue_details=[]
    try:
        venue_id=venue_data['id']
        venue_name=venue_data['name']
        venue_likes=venue_data['likes']['count']
        venue_rating=venue_data['rating']
        venue_tips=venue_data['tips']['count']

venue_details.append([venue_id,venue_name,venue_likes,venue_rating,venue_tips
])
    except KeyError:
        pass

    column_names=['ID','Name','Likes','Rating','Tips']
    df = pd.DataFrame(venue_details,columns=column_names)
    return df
```

Now we get the Toronto city data such as Boroughs, Neighborhoods along with their latitude and longitude.

We will get the Toronto city data from wikipedia and get latitude and longitude from geodata.

In [ ]:
```
import requests
import pandas as pd
from bs4 import BeautifulSoup
website_url =
requests.get("https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M
").text
soup = BeautifulSoup(website_url,'lxml')
table = soup.find('table',{'class':'wikitable sortable'})
table_rows = table.find_all('tr')
res = []
for tr in table_rows:
    td = tr.find_all('td')
    row = [tr.text.strip() for tr in td if tr.text.strip()]

    if row:
        # ignore cell with boroage is nan
        if row[1] != 'Not assigned':
            if row[2]=='Not assigned':
               row[2]=row[1]
            res.append(row)
df = pd.DataFrame(res, columns=["Postcode", "Borough", "Neighborhood"])
df_group = df.groupby(['Postcode', 'Borough']).apply(lambda group:
','.join(group['Neighborhood']))
df_postcode = df_group.to_frame().reset_index()
df_postcode.columns = ['Postcode', 'Borough', 'Neighborhood']
locgeo_df = pd.read_csv('https://cocl.us/Geospatial_data', index_col='Postal
Code')
toronto_data = df_postcode.join(locgeo_df, on='Postcode')
toronto_data.head()
```

So there are total of 103 different Neighborhoods in Toronto

In [ ]:
```
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('Number of Neighborhood for each Borough in Totonto City')
#On x-axis
plt.xlabel('Borough', fontsize = 15)
#On y-axis
plt.ylabel('No.of Neighborhood', fontsize=15)
#giving a bar plot
toronto_data.groupby('Borough')['Neighborhood'].count().plot(kind='bar')
#legend
plt.legend()
#displays the plot
plt.show()
```

We see that North York has highest number of neighborhoods

Now we will collect Thai resturants for each Neighborhood

In [ ]:
```
# prepare neighborhood list that contains indian resturants
column_names=['Postcode','Borough', 'Neighborhood', 'ID','Name']
thai_rest_toronto=pd.DataFrame(columns=column_names)
count=1
for row in toronto_data.values.tolist():
    Postcode,Borough, Neighborhood, Latitude, Longitude=row
    venues = get_venues(Latitude,Longitude)
    thai_resturants=venues[venues['Category']=='Thai Restaurant']
    print('(',count,'/',len(toronto_data),')','Thai Resturants in
'+Neighborhood+', '+Borough+':'+str(len(thai_resturants)))
    for resturant_detail in thai_resturants.values.tolist():
        id, name , category=resturant_detail
        thai_rest_toronto = thai_rest_toronto.append({'Postcode': Postcode,
                                                'Borough': Borough,
                                                'Neighborhood': Neighborhood,
                                                'ID': id,
                                                'Name' : name
                                               }, ignore_index=True)
    count+=1
```

Now that we have got all the Thai resturants in Toronto,CA , we will analyze it

In [ ]:
```
thai_rest_toronto.shape
```

We got 58 Thai Resturants across Toronto City

In [ ]:
```
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('Number of Thai Resturants for each Borough in Toronto')
#On x-axis
plt.xlabel('Borough', fontsize = 15)
#On y-axis
plt.ylabel('No.of Thai Resturants', fontsize=15)
#giving a bar plot
thai_rest_toronto.groupby('Borough')['ID'].count().plot(kind='bar')
#legend
plt.legend()
#displays the plot
plt.show()
```

We see that Downtown Toronto has the largest number of Thai restuarants

In [ ]:
```
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('Number of Thai Resturants for each Neighborhood in Toronto')
#On x-axis
plt.xlabel('Neighborhood', fontsize = 15)
#On y-axis
plt.ylabel('No.of Thai Resturants', fontsize=15)
#giving a bar plot
thai_rest_toronto.groupby('Neighborhood')['ID'].count().nlargest(5).plot(kind
='bar')
#legend
plt.legend()
#displays the plot
plt.show()
```

we see that for each Neighborhood is similar number of Thai Restaurant.

Now we will get the ranking of each resturant for further analysis.

In [ ]:
```
# prepare neighborhood list that contains indian resturants
column_names=['Postcode','Lat','lng','Borough', 'Neighborhood',
'ID','Name','Likes','Rating','Tips']
thai_rest_stats_toronto=pd.DataFrame(columns=column_names)
count=1
for row in thai_rest_toronto.values.tolist():
    Postcode,Borough,Neighborhood,ID,Name=row
    try:
        venue_details=get_venue_details(ID)
        print(venue_details)
        id,name,likes,rating,tips=venue_details.values.tolist()[0]
    except IndexError:
        print('No data available for id=',ID)
        # we will assign 0 value for these resturants as they may have been
        #recently opened or details does not exist in FourSquare Database
        id,name,likes,rating,tips=[0]*5
    print('(',count,'/',len(thai_rest_toronto),')','processed')
    thai_rest_stats_toronto = thai_rest_stats_toronto.append({'Postcodd':
Postcode,
                                                'latitude' : Latitude,
                                                'longitude' : Longitude,
                                                'Borough': Borough,
                                                'Neighborhood': Neighborhood,
                                                'ID': id,
                                                'Name' : name,
                                                'Likes' : likes,
                                                'Rating' : rating,
                                                'Tips' : tips
                                               }, ignore_index=True)
    count+=1
```

So we got data for all resturants Now lets save this data to a csv sheet. In case we by mistake modify it. As the number of calls to get details for venue are premium call and have limit of 500 per day, we will refer to saved data sheet csv if required

In [ ]:
```
# thai_rest_stats_toronto.to_csv('thai_rest_stats_toronto.csv', index=False)
thai_rest_stats_toronto.to_csv('thai_rest_stats_toronto.csv')
```

Lets verify the data from saved csv file

In [ ]:
```
thai_rest_stats_toronto_csv=pd.read_csv('thai_rest_stats_toronto.csv')
```
In [ ]:
```
thai_rest_stats_toronto_csv=pd.read_csv('thai_rest_stats_toronto.csv')
thai_rest_stats_toronto_csv =
thai_rest_stats_toronto_csv.drop(thai_rest_stats_toronto_csv.columns[[0, 1,
3]], axis=1)  # df.columns is zero-based pd.Index
thai_rest_stats_toronto.head()
```

We see that values like Likes, Tips are strig values. We would need to convert them into float for further analysis

In [ ]:
```
thai_rest_stats_toronto.info()
```
In [ ]:
```
thai_rest_stats_toronto['Likes']=thai_rest_stats_toronto['Likes'].astype('flo
at64')
```
In [ ]:
```
thai_rest_stats_toronto['Tips']=thai_rest_stats_toronto['Tips'].astype('float
64')
```

Now the data types looks correct

In [ ]:
```
# Resturant with maximum Likes
thai_rest_stats_toronto.iloc[thai_rest_stats_toronto['Likes'].idxmax()]
```
In [ ]:
```
# Resturant with maximum Rating
thai_rest_stats_toronto["Rating"] =
pd.to_numeric(thai_rest_stats_toronto_csv["Rating"])
thai_rest_stats_toronto.iloc[thai_rest_stats_toronto['Rating'].idxmax()]
```
In [ ]:
```
# Resturant with maximum Tips
thai_rest_stats_toronto.iloc[thai_rest_stats_toronto_csv['Tips'].idxmax()]
```

Now lets visualize neighborhood with maximum average rating of resturants

In [ ]:
```
toronto_neighborhood_stats=thai_rest_stats_toronto_csv.groupby('Neighborhood'
,as_index=False).mean()[['Neighborhood','Rating']]
# toronto_neighborhood_stats.columns=['Neighborhood','Average Rating']
toronto_neighborhood_stats.head()
# toronto_data.head()
df_new=pd.merge(toronto_data, toronto_neighborhood_stats,
left_on='Neighborhood', right_on='Neighborhood')
```
In [ ]:
```
df_new.to_csv('df_new.csv')
df_new.head(100)
```
In [ ]:
```
toronto_neighborhood_stats.sort_values(['Rating'],ascending=False).head()
```

Above are the top neighborhoods with top average rating of Thai resturants

In [ ]:
```
toronto_borough_stats=thai_rest_stats_toronto.groupby('Borough',as_index=Fals
e).mean()[['Borough','Rating']]
toronto_borough_stats.columns=['Borough','Average Rating']
```
In [ ]:
```
toronto_borough_stats.sort_values(['Average Rating'],ascending=False).head()
```

Similarly these are the average rating of Thai Resturants for each Borough

Lets visualize it

In [ ]:
```
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('Average rating of Thai Resturants for each Borough')
#On x-axis
plt.xlabel('Borough', fontsize = 15)
#On y-axis
plt.ylabel('Average Rating', fontsize=15)
#giving a bar plot
```

```
thai_rest_stats_toronto.groupby('Borough').mean()['Rating'].plot(kind='bar')
#legend
plt.legend()
#displays the plot
plt.show()
```

We will consider all the neighborhoods with average rating greater or equal 7.0 to visualize on map

In [ ]:
```
toronto_neighborhood_stats.sort_values(['Rating'],ascending=False).head()
# toronto_neighborhood_stats.head()
```

We will join this dataset to original Toronto data to get lonitude and latitude

In [ ]:
```
toronto_neighborhood_stats=pd.merge(toronto_neighborhood_stats,toronto_data,
on='Neighborhood')
```
In [ ]:
```
toronto_neighborhood_stats=toronto_neighborhood_stats[['Borough','Neighborhoo
d','Latitude','Longitude','Rating']]
toronto_neighborhood_stats.sort_values(['Rating'],ascending=False).head()
toronto_neighborhood_stats.head()
```

Now we will show this data on a map

In [ ]:
```
# create map and display it
toronto_map = folium.Map(location=geo_location('Toronto,CA'), zoom_start=14)
toronto_neighborhood_stats.to_csv('toronto_neighborhood_vis.csv',
index=False)
toronto_neighborhood_vis = pd.read_csv("toronto_neighborhood_vis.csv")
toronto_neighborhood_vis.to_csv('toronto_neighborhood_vis.csv')
```
In [ ]:
```
# instantiate a feature group for the incidents in the dataframe
incidents = folium.map.FeatureGroup()

for lat, lng, in toronto_neighborhood_stats[['Latitude','Longitude']].values:
    incidents.add_child(
        folium.CircleMarker(
            [lat, lng],
            radius=10, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='blue',
            fill_opacity=0.6
        )
    )
```

Lets add a new field to dataframe for labeling purpose

In [ ]:

```
toronto_neighborhood_stats.head()
```
In [ ]:
```
toronto_neighborhood_stats['Label']=toronto_neighborhood_stats['Neighborhood'
]+',
'+toronto_neighborhood_stats['Borough']+'('+toronto_neighborhood_stats['Ratin
g'].map(str)+')'
```
In [ ]:
```
toronto_neighborhood_vis.to_csv('toronto_neighborhood_vis.csv')
```
In [ ]:
```
GeoJson = folium.Map(location=[43.6532, -79.3832], zoom_start=14, tiles =
'cartodbpositron')
```
In [ ]:
```
import time
# Function to change the marker color
# according to the elevation of volcano
def color(avg):
    if 6 <= avgr <= 7:
        col = 'red'
    elif 7 <= avgr <= 8:
        col = 'yellow'
    elif 8 <= avgr <= 9:
        col = 'green'
    else:
        col='blue'
    return col
torontoMap = folium.Map(location=[43.6532, -79.3832], zoom_start=14)
toronto_geo = 'CanadianBound.geojson'
GeoJson = folium.Map(location=[43.6532, -79.3832], zoom_start=14, tiles =
'cartodbpositron')

# add pop-up text to each marker on the map
for lat, lng,avgr in
toronto_neighborhood_vis[['Latitude','Longitude','Rating']].values:
    folium.Marker([lat, lng],
popup=repr(avgr),icon=folium.Icon(color=color(avgr))).add_to(GeoJson)
```
In [ ]:
```
toronto_borough_stats=toronto_neighborhood_stats.groupby('Borough',as_index=F
alse).mean()[['Borough','Latitude','Longitude','Rating']]
toronto_borough_stats.columns=['Borough','Latitude','Longitude','Rating']
toronto_borough_stats.head()
```
In [ ]:
```
toronto_borough_stats=toronto_neighborhood_stats.groupby('Borough',as_index=F
alse).mean()[['Borough','Latitude','Longitude','Rating']]
toronto_borough_stats.columns=['Borough','Latitude','Longitude','Rating']
toronto_map = folium.Map(location=geo_location('Toronto'), zoom_start=14)
toronto_geo = r'CanadianBound1.geojson'
GeoJson.choropleth(geo_data=toronto_geo,
    data = df_new,
    columns=['Postcode', 'Rating'],
    key_on='feature.properties.CFSAUID',
    fill_color='YlOrRd',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Rating')
```

`GeoJson`

Now that we have visualized the Neighborhoods.
In green maker show the Restaurants is rating between 8-9 point
in Red maker show the Restaurants is Rating Between 0-7 Point

Conclusion

The best place to run the Thai food business is located in the Down Town area in Toronto, as it is considered rating, which means a lot of people who have been eating Thai food, heading Down to Town Down Town, to go to those Thai cusine.

Limitations

- The ranking is purely on basis of rating of resturants
- The accuracy of data depends purely depends on the data provided by FourSquare