

Analysis of the Restaurant Market in South Mumbai

Applied Data Science Capstone

Introduction

```
In [249]: from IPython.display import Image  
Image("marinedrive.jpg")
```

Out[249]:



Mumbai is the financial, commercial and entertainment capital of India and also one of the world's top ten hubs of commerce in terms of global financial flow. Mumbai's business and investment opportunities, as well as its potential to offer a higher standard of living, attracts migrants not from only India but also from other countries, making the city a multicultural. It is a major hub for world trade, jewelry, finance and banking, film industry, arts, information technology, healthcare, media, entertainment, fashion, tourism etc.

“South Mumbai “is the Mumbai City district which is the southernmost precinct of Greater Mumbai. It extends from Colaba in the south to Mahim and Sion in the north. It comprises the city's main business localities, making it the wealthiest urban precinct in India. But at the same time high business opportunities and development makes South Mumbai as the extremely competitive. In this fierce market, the risk volume is even higher for new start-ups or business enterprises or for making an expansion of business especially for small and medium enterprises.

Due to this, any new business plan or expansion needs to be analyzed carefully. The insight derived from analysis will give better understanding of business environment which help in maneuvering the market.

In a case of Restaurant industry, it is evident that to start a restaurant in right location is very effective and wise strategic plan, but many people especially who haven't much knowledge and exploration about demographics of South Mumbai are facing many issues to find the suitable restaurant location. Apart from location, many restaurateurs and investors don't know what kind of restaurant they need to start. Those question are needed to be solved by analysis of the Restaurant Business in South Mumbai.

In []:

1. Data Section

- importing libraries

```
In [1]: !pip install geocoder

from bs4 import BeautifulSoup
import requests
import numpy as np

import pandas as pd
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

import json # library to handle JSON files

import urllib
import geocoder

!conda install -c conda-forge geopy --yes # uncomment this line if you haven't completed the Foursquare API lab
from geopy.geocoders import Nominatim # convert an address into latitude and longitude values

import requests # library to handle requests
from pandas.io.json import json_normalize # tranform JSON file into a pandas dataframe

# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors

# import k-means from clustering stage
from sklearn.cluster import KMeans

!conda install -c conda-forge folium=0.5.0 --yes # uncomment this line if you haven't completed the Foursquare API lab
import folium # map rendering library

print('Libraries imported.')
```

Collecting geocoder

Downloading <https://files.pythonhosted.org/packages/4f/6b/13166c909ad2f2d76b929a4227c952630ebaf0d729f6317eb09cbceccbab/geocoder-1.38.1-py2.py3-none-any.whl> (98kB)

100% |██| 102kB 16.8MB/s

Requirement already satisfied: future in /home/jupyterlab/conda/lib/python3.6/site-packages (from geocoder) (0.17.1)

Requirement already satisfied: six in /home/jupyterlab/conda/lib/python3.6/site-packages (from geocoder) (1.11.0)

Requirement already satisfied: click in /home/jupyterlab/conda/lib/python3.6/site-packages (from geocoder) (7.0)

Requirement already satisfied: requests in /home/jupyterlab/conda/lib/python3.6/site-packages (from geocoder) (2.20.1)

Collecting ratelim (from geocoder)

Downloading <https://files.pythonhosted.org/packages/f2/98/7e6d147fd16a10a5f821db6e25f192265d6ecca3d82957a4fdd592cad49c/ratelim-0.1.6-py2.py3-none-any.whl>

Requirement already satisfied: certifi>=2017.4.17 in /home/jupyterlab/conda/lib/python3.6/site-packages (from requests->geocoder) (2018.11.29)

Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /home/jupyterlab/conda/lib/python3.6/site-packages (from requests->geocoder) (3.0.4)

Requirement already satisfied: urllib3<1.25,>=1.21.1 in /home/jupyterlab/conda/lib/python3.6/site-packages (from requests->geocoder) (1.23)

Requirement already satisfied: idna<2.8,>=2.5 in /home/jupyterlab/conda/lib/python3.6/site-packages (from requests->geocoder) (2.7)

Requirement already satisfied: decorator in /home/jupyterlab/conda/lib/python3.6/site-packages (from ratelim->geocoder) (4.3.0)

Installing collected packages: ratelim, geocoder

Successfully installed geocoder-1.38.1 ratelim-0.1.6

Solving environment: done

Package Plan

environment location: /home/jupyterlab/conda

added / updated specs:

- geopy

The following packages will be downloaded:

package	build		
geopy-1.18.1	py_0	51 KB	conda-forge
geographiclib-1.49	py_0	32 KB	conda-forge
Total:		84 KB	

The following NEW packages will be INSTALLED:

geographiclib-1.49-py_0 conda-forge

- Fetching data from CSV file downloaded from Open Government Data (OGD) Platform India (Digital India Initiative)

```
In [2]: url = "mumbai_neighbourhoods.csv"
mumbai_data = pd.read_csv(url)
mumbai_data.tail()
```

Out[2]:

	Neighbourhood	Pincode	officeType	Deliverystatus	regionname	circlename	Taluk	Districtname	statename	Telephone	Related Suboffice	Borough
1002	Wada	421303	NaN	Delivery	Mumbai	Maharashtra	Wada	Thane	MAHARASHTRA	02526-271413	NaN	South Mumbai
1003	Waki B.O	401602	B.O	Delivery	Mumbai	Maharashtra	Dahanu	Thane	MAHARASHTRA	NaN	Dahanu Road	South Mumbai
1004	Waki B.O	421303	B.O	Delivery	Mumbai	Maharashtra	Vikramgad	Thane	MAHARASHTRA	NaN	Wada	South Mumbai
1005	Zap B.O	401603	B.O	Delivery	Mumbai	Maharashtra	Jawhar	Thane	MAHARASHTRA	NaN	Jawhar	South Mumbai
1006	Zari B.O	401606	B.O	Delivery	Mumbai	Maharashtra	Talasari	Thane	MAHARASHTRA	NaN	Talasari	South Mumbai

- Project is focus on 'South Mumbai' so data filtering as per requirement

```
In [3]: mumbai_data = mumbai_data[mumbai_data.Borough=='South Mumbai']
mumbai_data.head()
```

Out[3]:

	Neighbourhood	Pincode	officeType	Deliverystatus	regionname	circlename	Taluk	Districtname	statename	Telephone	Related Suboffice	Borough
0	B.P.Lane	400003	NaN	Non-Delivery	Mumbai	Maharashtra	NaN	Mumbai	MAHARASHTRA	022-23421653	NaN	South Mumbai
1	BEST Staff Quarters	400012	NaN	Non-Delivery	Mumbai	Maharashtra	NaN	Mumbai	MAHARASHTRA	022-24180776	NaN	South Mumbai
2	Chamalbaug	400012	NaN	Non-Delivery	Mumbai	Maharashtra	Mumbai	Mumbai	MAHARASHTRA	022-24705407	NaN	South Mumbai
3	Chinchbunder	400009	NaN	Delivery	Mumbai	Maharashtra	NaN	Mumbai	MAHARASHTRA	022-23771116	NaN	South Mumbai
4	Cotton Exchange	400033	NaN	Non-Delivery	Mumbai	Maharashtra	Mumbai	Mumbai	MAHARASHTRA	022-23742035	NaN	South Mumbai

- Data cleaning as removing redundancy and unwanted duplicity

1.Join all the AreaName by Pincode

```
In [4]: def neighbourhood_list(grouped):
        return ', '.join(sorted(grouped['Neighbourhood'].tolist()))

grp = mumbai_data.groupby(['Pincode', 'Borough'])
south_mumbai_data = grp.apply(neighbourhood_list).reset_index(name='Neighbourhood')
```

2.Displaying south_mumbai_data

```
In [5]: south_mumbai_data.head()
```

```
Out[5]:
```

	Pincode	Borough	Neighbourhood
0	400001	South Mumbai	Bazargate , CSMT , M.P.T. , Stock Exchange , T...
1	400002	South Mumbai	Kalbadevi , Ramwadi , S. C. Court , Thakurdwar
2	400003	South Mumbai	B.P.Lane , Mandvi (Mumbai), Masjid , Null Bazar
3	400004	South Mumbai	Ambewadi (Mumbai), Charni Road , Chaupati , G...
4	400005	South Mumbai	Asvini , Colaba , Colaba Bazar , Holiday Camp ...

3.Getting Coordinates as per pincode

```
In [6]: def get_latlng(postal_code):
        # initialize your variable to None
        lat_lng_coords = None
        # loop until you get the coordinates
        while(lat_lng_coords is None):
            g = geocoder.arcgis('{} , Mumbai, Maharashtra'.format(postal_code))
            lat_lng_coords = g.latlng
        return lat_lng_coords
```

e.g, For Pincode 400001

```
In [7]: get_latlng('400001')
```

```
Out[7]: [18.938919252000062, 72.837470000000005]
```

Now applying this functions for all pincode of areas of South Mumbai

```
In [8]: latitude = []
longitude = []
for row in south_mumbai_data['Pincode']:
    coordinate = get_latlng(row)
    latitude.append(coordinate[0])
    longitude.append(coordinate[1])

south_mumbai_data['Latitude']=latitude
south_mumbai_data['Longitude']=longitude
south_mumbai_data.head()
```

Out[8]:

	Pincode	Borough	Neighbourhood	Latitude	Longitude
0	400001	South Mumbai	Bazargate , CSMT , M.P.T. , Stock Exchange , T...	18.938919	72.837470
1	400002	South Mumbai	Kalbadevi , Ramwadi , S. C. Court , Thakurdwar	18.947760	72.826815
2	400003	South Mumbai	B.P.Lane , Mandvi (Mumbai), Masjid , Null Bazar	18.951733	72.835129
3	400004	South Mumbai	Ambewadi (Mumbai), Charni Road , Chaupati , G...	18.955285	72.820930
4	400005	South Mumbai	Asvini , Colaba , Colaba Bazar , Holiday Camp ...	18.904399	72.810195

- Loading property data which are researched from different Indian propert portals and property news and professional realtors which stored in CSV file

```
In [9]: property_url = 'South_Mumbai_Real_Estate.csv'
property_data = pd.read_csv(property_url)
property_data.head()
```

Out[9]:

	Pincode	Neighbourhood	Avg. Price Range (Rs. per sq. ft.)
0	400001	Bazargate , CSMT , M.P.T. , Stock Exchange , T...	68817.0
1	400002	Kalbadevi , Ramwadi , S. C. Court , Thakurdwar	49502.0
2	400003	B.P.Lane , Mandvi (Mumbai), Masjid , Null Bazar	16666.0
3	400004	Ambewadi (Mumbai), Charni Road , Chaupati , G...	32926.0
4	400005	Asvini , Colaba , Colaba Bazar , Holiday Camp ...	68817.0

- South Mumbai ward wise average property price

```
In [10]: southmumbai_propertyrate = 'BMCWardproperties.csv'
southmumbai_propertyrate_data = pd.read_csv(southmumbai_propertyrate)
southmumbai_propertyrate_data.head()
```

Out[10]:

	Ward	Avg Price
0	A	125761
1	B	45900
2	C	65800
3	D	105753
4	E	55800

- Fetching details from wikipedia page of Administrative divisions of Mumbai 1.Fetch data from wikipedia

```
In [12]: # fetching data from wikipedia page and storing into m_wiki_data
url = "https://en.wikipedia.org/wiki/Administrative_divisions_of_Mumbai"
m_wiki_data = requests.get(url).text
interm = m_wiki_data[m_wiki_data.find("<table"):m_wiki_data.find("</table>")+8]
population_sm = pd.read_html(interm, header = 0)[0]
population_sm.head()
```

Out[12]:

	Ward	Population
0	A	210926
1	B	140480
2	C	190670
3	D	378610
4	E	439390

- Next, we are going to start utilizing the Foursquare API to explore the South Mumbai for data collection

```
In [139]: CLIENT_ID = 'XXXXXX' # your Foursquare ID
CLIENT_SECRET = 'XXXX' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

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

```
Your credentails:
CLIENT_ID: XXXXXX
CLIENT_SECRET:XXXX
```

1.Taking first area of South Mumbai and creating url

```
In [14]: first_lat = south_mumbai_data['Latitude'][0]
first_long = south_mumbai_data['Longitude'][0]
radius = 500
LIMIT = 100
url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}
{}'.format(
    CLIENT_ID,
    CLIENT_SECRET,
    VERSION,
    first_lat,
    first_long,
    radius,
    LIMIT)
```

2.Getting results

```
In [15]: results = requests.get(url).json()
```

```
In [16]: # function that extracts the category of the venue
def get_category_type(row):
    try:
        categories_list = row['categories']
    except:
        categories_list = row['venue.categories']

    if len(categories_list) == 0:
        return None
    else:
        return categories_list[0]['name']
```

3. Now we are ready to clean the json and structure it into a *pandas* dataframe.

```
In [17]: venues = results['response']['groups'][0]['items']

nearby_venues = json_normalize(venues) # flatten JSON

# filter columns
filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat', 'venue.location.lng']
nearby_venues = nearby_venues.loc[:, filtered_columns]

# filter the category for each row
nearby_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)

# clean columns
nearby_venues.columns = [col.split(".")[-1] for col in nearby_venues.columns]

nearby_venues.head()
```

Out[17]:

	name	categories	lat	lng
0	Sher-E-Punjab	Indian Restaurant	18.937944	72.837853
1	Chhatrapati Shivaji Maharaj Terminus	Train Station	18.940297	72.835384
2	Royal China	Chinese Restaurant	18.938715	72.832933
3	Dakshin Bar And Kitchen	Seafood Restaurant	18.936489	72.837490
4	Town House Cafe	Bar	18.938550	72.833464

2. Methodology section

- Creating South Mumbai Map

```
In [18]: address = 'Mumbai, Maharashtra'

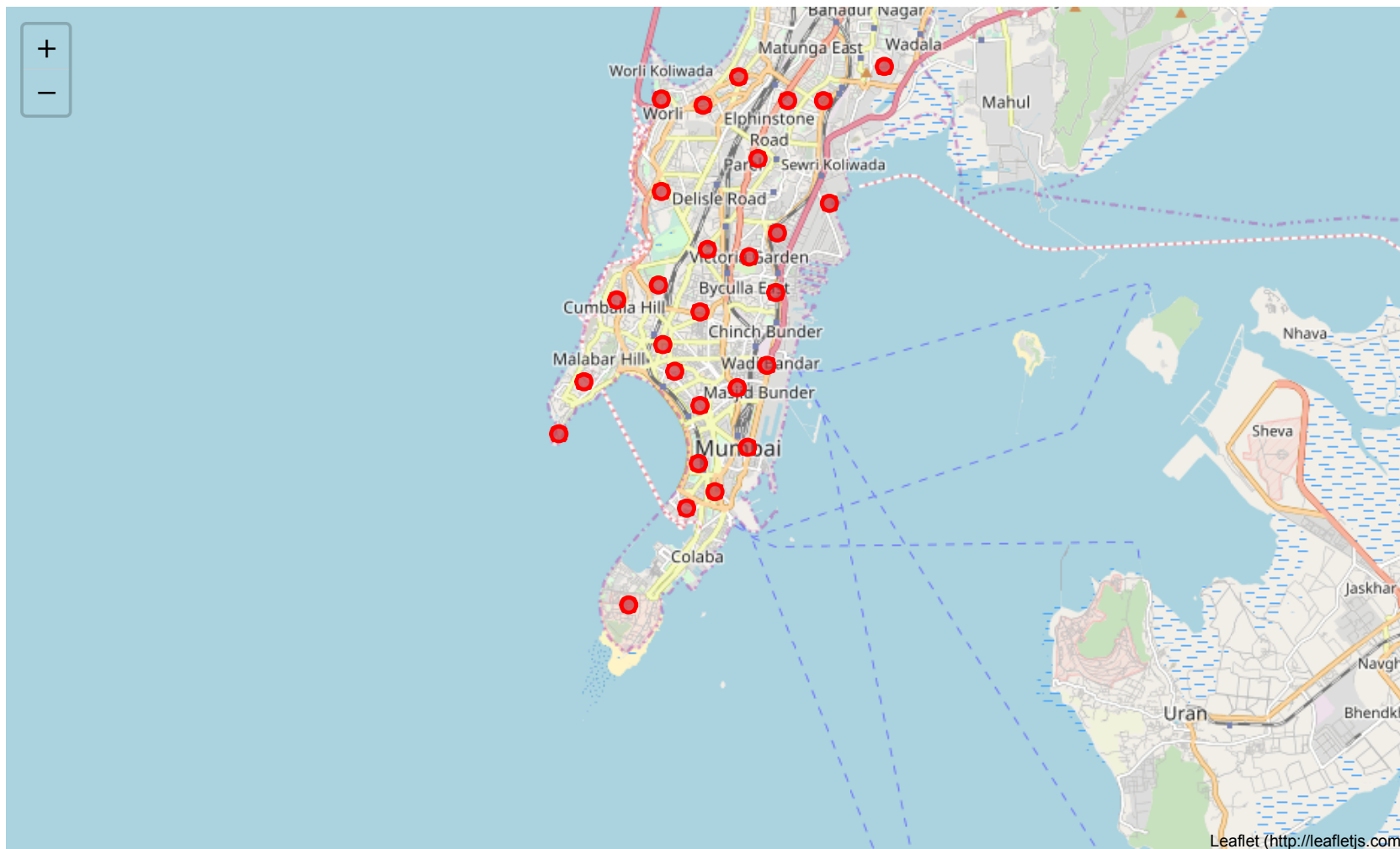
geolocator = Nominatim(user_agent="userid")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude

# create map of South Mumbai using latitude and longitude values
map_south_mumbai = folium.Map(location=[latitude, longitude], zoom_start=11.5)

# add markers to map
for lat, lng, borough, neighborhood in zip(south_mumbai_data['Latitude'], south_mumbai_data['Longitude'], south_mumbai_data['Borough'], south_mumbai_data['Neighbourhood']):
    label = '{} , {}'.format(neighborhood, borough)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='red',
        fill=True,
        fill_color='#cc3139',
        fill_opacity=0.7,
        parse_html=False).add_to(map_south_mumbai)

map_south_mumbai
```

Out[18]:



- Choropleth Maps of Properties of South Mumbai

```
In [19]: #South Mumbai Coordinates
SM_COORDINATES = (18.96, 72.82)

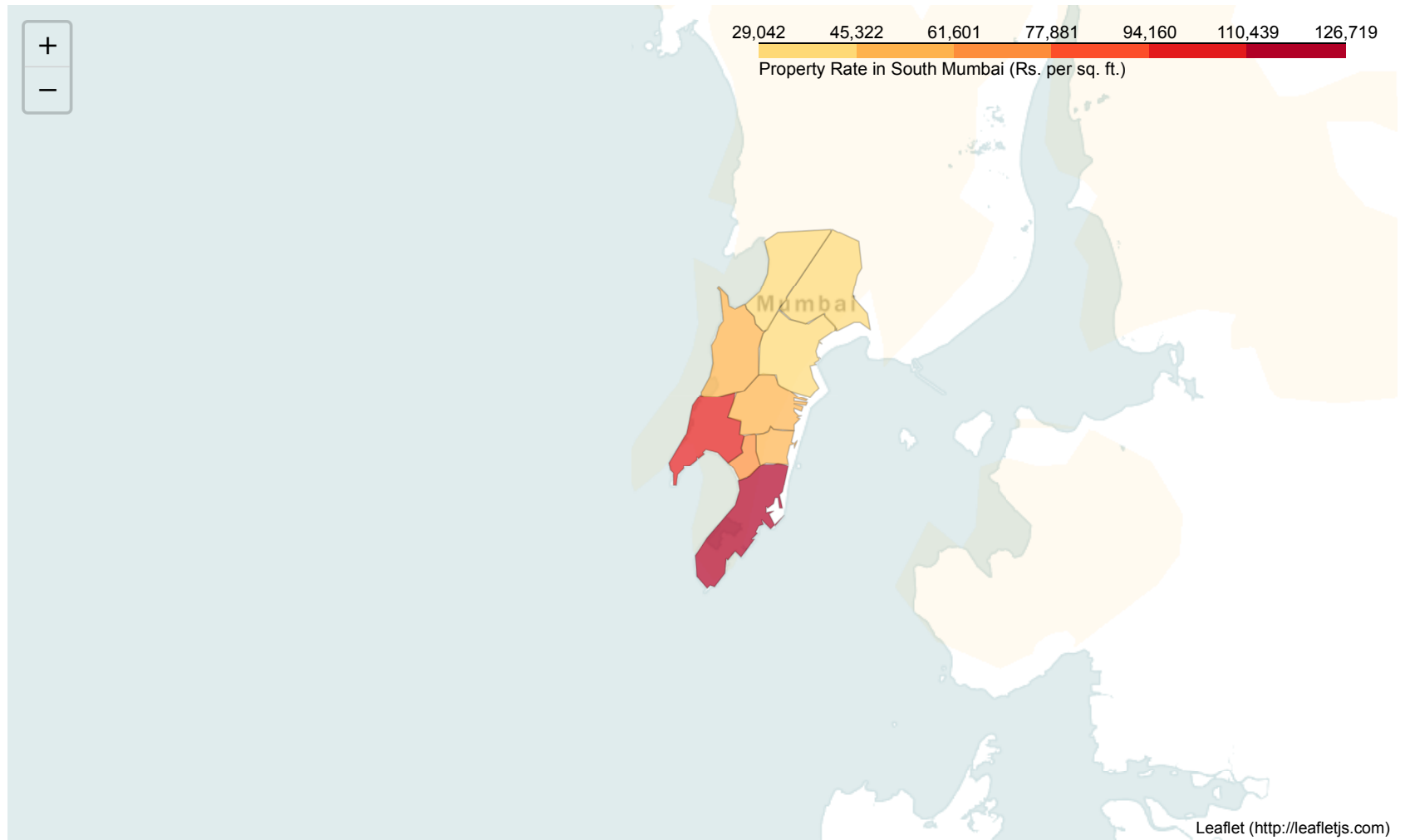
southmumbai_geo = r'southmumbai.geojson'

# create empty map zoomed in on South Mumbai
map = folium.Map(location=SM_COORDINATES,tiles="Mapbox Bright", zoom_start=11)

# creation of the choropleth
map_sc = folium.Map(location=SM_COORDINATES,tiles="Mapbox Bright", zoom_start=11)

map_sc.choropleth(geo_data = southmumbai_geo,
                  name='choropleth',
                  data = southmumbai_propertyrate_data,
                  columns = ['Ward', 'Avg Price'],
                  key_on = 'feature.properties.Ward',
                  fill_color = 'YlOrRd',
                  fill_opacity = 0.7,
                  line_opacity = 0.2,
                  legend_name = 'Property Rate in South Mumbai (Rs. per sq. ft.)')

display(map_sc)
```



- Choropleth Maps of Population of South Mumbai


```
In [20]: #South Mumbai Coordinates
SM_COORDINATES = (18.96, 72.82)

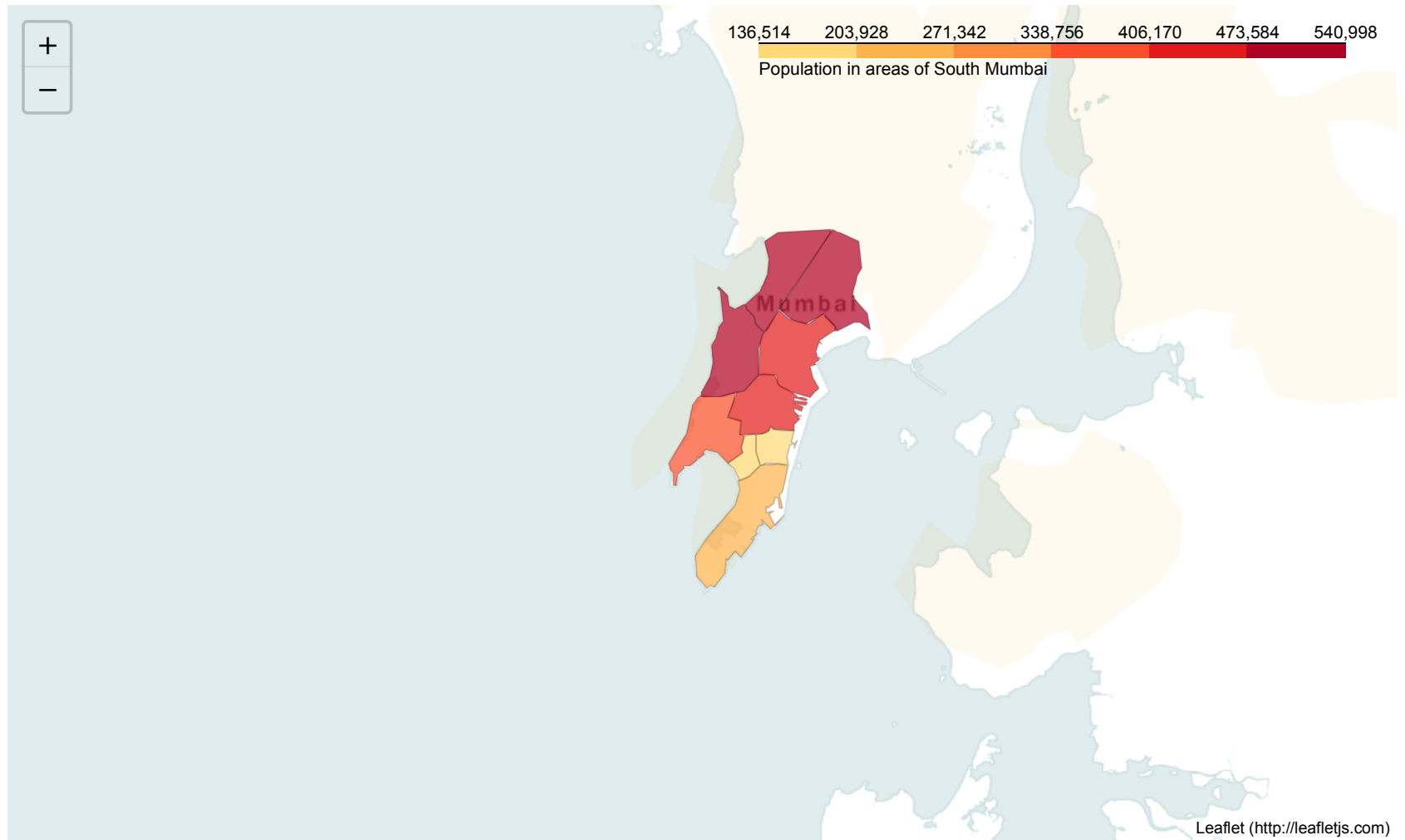
southmumbai_geo = r'southmumbai.geojson'

# create empty map zoomed in on South Mumbai
map = folium.Map(location=SM_COORDINATES,tiles="Mapbox Bright", zoom_start=11)

# creation of the choropleth
map_sc = folium.Map(location=SM_COORDINATES,tiles="Mapbox Bright", zoom_start=11)

map_sc.choropleth(geo_data = southmumbai_geo,
                  name='choropleth',
                  data = population_sm,
                  columns = ['Ward', 'Population'],
                  key_on = 'feature.properties.Ward',
                  fill_color = 'YlOrRd',
                  fill_opacity = 0.7,
                  line_opacity = 0.2,
                  legend_name = 'Population in areas of South Mumbai')

display(map_sc)
```



- Finding the number of restaurants by the areas of South Mumbai

```
In [21]: 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']

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

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighbourhood',
                            'Neighbourhood Latitude',
                            'Neighbourhood Longitude',
                            'Venue',
                            'Venue Latitude',
                            'Venue Longitude',
                            'Venue Category']

    return(nearby_venues)
```

```
In [22]: southmumbai_venues = getNearbyVenues(names=south_mumbai_data['Neighbourhood'],
                                             latitudes=south_mumbai_data['Latitude'],
                                             longitudes=south_mumbai_data['Longitude']
                                             )
```

Bazargate , CSMT , M.P.T. , Stock Exchange , Tajmahal , Town Hall (Mumbai)
 Kalbadevi , Ramwadi , S. C. Court , Thakurdwar
 B.P.Lane , Mandvi (Mumbai), Masjid , Null Bazar
 Ambewadi (Mumbai), Charni Road , Chaupati , Girgaon , Madhavbaug , Opera House
 Asvini , Colaba , Colaba Bazar , Holiday Camp , V.W.T.C.
 Malabar Hill
 Bharat Nagar (Mumbai), Grant Road , N.S.Patkar Marg , S V Marg , Tardeo
 Falkland Road , J.J.Hospital , Kamathipura , M A Marg , Mumbai Central
 Chinchbunder , Noor Baug , Princess Dock
 Dockyard Road , Mazgaon , Mazgaon Dock , Mazgaon Road , V K Bhavan
 Agripada , BPC Jacob Circle , Chinchpokli , Haines Road , Jacob Circle
 BEST Staff Quarters , Chamarbaug , Haffkin Institute , Lal Baug , Parel , Parel Naka , Parel Rly Work Shop
 Dadar , Dadar Colony , Naigaon (Mumbai)
 Sewri
 Worli , Worli Naka
 Central Building , Churchgate , Marine Lines
 Nariman Point , New Yogakshema
 Chunabhatti , Raoli Camp , Sion , Transit Camp
 New Prabhadevi Road , Prabhadevi
 Cumballa Hill , Cumballa Sea Face , Dr Deshmukh Marg , Gowalia Tank
 V J B Udyan
 Ranade Road , S V S Marg , Shivaji Park (Mumbai)
 Worli Colony , Worli Police Camp , Worli Sea Face
 Kidwai Nagar (Mumbai), Wadala , Wadala Rs
 High Court Building (Mumbai), Mantralaya (Mumbai), Secretariate
 Cotton Exchange , Kalachowki , L B S N E collage , Reay Road , Tank Road
 Haji Ali , Tulsiwadi
 Rajbhavan (Mumbai)
 Antop Hill , B P T Colony , C G S Colony , Wadala Truck Terminal
 Sakinaka , Vihar Road

'Top 5 regions of South Mumbai with Highest Number of Restaurants'

```
In [23]: areawise_venueus = southmumbai_venues.groupby('Neighbourhood').count()
areawise_venueus.to_csv('South_Mumbai_Restaurants_Count.csv', index=False)
areawise_venueus.sort_values(by=['Venue'], ascending=False).head()
```

Out [23]:

	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighbourhood						
High Court Building (Mumbai), Mantralaya (Mumbai), Secretariate	74	74	74	74	74	74
Central Building , Churchgate , Marine Lines	50	50	50	50	50	50
Nariman Point , New Yogakshema	36	36	36	36	36	36
Cumballa Hill , Cumballa Sea Face , Dr Deshmukh Marg , Gowalia Tank	23	23	23	23	23	23
Ambewadi (Mumbai), Charni Road , Chaupati , Girgaon , Madhavbaug , Opera House	20	20	20	20	20	20

Top 5 regions of South Mumbai with Lowest Number of Restaurants'

```
In [24]: areawise_venueus.sort_values(by=['Venue']).head()
```

Out [24]:

	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighbourhood						
Antop Hill , B P T Colony , C G S Colony , Wadala Truck Terminal	2	2	2	2	2	2
Chunabhatti , Raoli Camp , Sion , Transit Camp	3	3	3	3	3	3
Dockyard Road , Mazgaon , Mazgaon Dock , Mazgaon Road , V K Bhavan	4	4	4	4	4	4
Cotton Exchange , Kalachowki , L B S N E collage , Reay Road , Tank Road	4	4	4	4	4	4
Chinchbunder , Noor Baug , Princess Dock	4	4	4	4	4	4

Let's find out how many unique category of restaurants from South Mumbai

```
In [25]: print('There are {} uniques categories.'.format(len(areawise_venue['Venue Category'].unique())))
```

There are 17 uniques categories.

```
In [26]: def getNearbyVenuesNames(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):

        # 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']

        # return only relevant information for each nearby venue
        venues_list.append([

            v['venue']['name']
            ) for v in results])

    return(venues_list)
```

```
In [ ]: southmumbai_rest = getNearbyVenuesNames(names=south_mumbai_data['Neighbourhood'],
                                                latitudes=south_mumbai_data['Latitude'],
                                                longitudes=south_mumbai_data['Longitude']
                                                )

southmumbai_rest
```

```
In [30]: change=[]
for i in range(0, len(southmumbai_rest)-1):
    change = change + southmumbai_rest[i]
```

Top of 5 QSR Franchises in South Mumbai

```
In [31]: df = pd.DataFrame({'col':change})
df['col'].value_counts().head()
```

```
Out[31]: Cafe Coffee Day      13
Subway                        8
Domino's Pizza               5
McDonald's                   3
Monginis Cake Shop          3
Name: col, dtype: int64
```

```
In [32]: def getNearbyVenuesCat(names, latitudes, longitudes, radius=500):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):

        # 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"]

        # return only relevant information for each nearby venue
        venues_list.append([(
            v['venue']['categories'][0]['name']
        ) for v in results])

    return(venues_list)
```

```
In [37]: southmumbai_rest_cat = getNearbyVenuesCat(names=south_mumbai_data['Neighbourhood'],
                                                  latitudes=south_mumbai_data['Latitude'],
                                                  longitudes=south_mumbai_data['Longitude']
                                                  )
```

```
In [38]: totalcat=[]
         for i in range(0, len(southmumbai_rest_cat)-1):
             totalcat = totalcat + southmumbai_rest_cat[i]
```

Top 5 type of Restaurant of service types

```
In [39]: dfcat = pd.DataFrame({'col':totalcat})
         dfcat['col'].value_counts().head(5)
```

```
Out[39]: Indian Restaurant      47
         Café                    29
         Fast Food Restaurant    20
         Chinese Restaurant      17
         Bakery                  15
         Name: col, dtype: int64
```

```
In [254]:
```

```
In [40]: southmumbai_ves = getNearbyVenues(names=south_mumbai_data['Neighbourhood'],
                                             latitudes=south_mumbai_data['Latitude'],
                                             longitudes=south_mumbai_data['Longitude']
                                             )
```



```
In [41]: southmumbai_ves.groupby('Neighbourhood').count()
```

Out [41]:

	Neighbourhood Latitude	Neighbourhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Neighbourhood						
Agripada , BPC Jacob Circle , Chinchpokli , Haines Road , Jacob Circle	6	6	6	6	6	6
Ambewadi (Mumbai), Charni Road , Chaupati , Girgaon , Madhavbaug , Opera House	20	20	20	20	20	20
Antop Hill , B P T Colony , C G S Colony , Wadala Truck Terminal	2	2	2	2	2	2
Asvini , Colaba , Colaba Bazar , Holiday Camp , V.W.T.C.	5	5	5	5	5	5
B.P.Lane , Mandvi (Mumbai), Masjid , Null Bazar	19	19	19	19	19	19
BEST STaff Quarters , Chamarbaug , Haffkin Institute , Lal Baug , Parel , Parel Naka , Parel Rly Work Shop	13	13	13	13	13	13
Bazargate , CSMT , M.P.T. , Stock Exchange , Tajmahal , Town Hall (Mumbai)	19	19	19	19	19	19
Bharat Nagar (Mumbai), Grant Road , N.S.Patkar Marg , S V Marg , Tardeo	11	11	11	11	11	11
Central Building , Churchgate , Marine Lines	50	50	50	50	50	50
Chinchbunder , Noor Baug , Princess Dock	4	4	4	4	4	4
Chunabhatti , Raoli Camp , Sion , Transit Camp	3	3	3	3	3	3
Cotton Exchange , Kalachowki , L B S N E collage , Reay Road , Tank Road	4	4	4	4	4	4
Cumballa Hill , Cumballa Sea Face , Dr Deshmukh Marg , Gowalia Tank	23	23	23	23	23	23
Dadar , Dadar Colony , Naigaon (Mumbai)	19	19	19	19	19	19
Dockyard Road , Mazgaon , Mazgaon Dock , Mazgaon Road , V K Bhavan	4	4	4	4	4	4
Falkland Road , J.J.Hospital , Kamathipura , M A Marg , Mumbai Central	9	9	9	9	9	9
Haji Ali , Tulsiwadi	11	11	11	11	11	11
High Court Building (Mumbai), Mantralaya (Mumbai), Secretariate	74	74	74	74	74	74
Kalbadevi , Ramwadi , S. C. Court , Thakurdwar	9	9	9	9	9	9
Kidwai Nagar (Mumbai), Wadala , Wadala Rs	10	10	10	10	10	10
Malabar Hill	5	5	5	5	5	5
Nariman Point , New Yogakshema	36	36	36	36	36	36
New Prabhadevi Road , Prabhadevi	16	16	16	16	16	16

```
In [66]: # one hot encoding
sm_onehot = pd.get_dummies(southmumbai_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
sm_onehot['Neighbourhood'] = southmumbai_venues['Neighbourhood']

# move neighborhood column to the first column
fixed_columns = [sm_onehot.columns[-1]] + list(sm_onehot.columns[:-1])
sm_onehot = sm_onehot[fixed_columns]

sm_grouped = sm_onehot.groupby('Neighbourhood').mean().reset_index()
```

```
In [67]: def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]
```

```
In [68]: num_top_venues = 20

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Neighbourhood']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
neighbourhoods_venues_sorted = pd.DataFrame(columns=columns)
neighbourhoods_venues_sorted['Neighbourhood'] = sm_grouped['Neighbourhood']

for ind in np.arange(sm_grouped.shape[0]):
    neighbourhoods_venues_sorted.iloc[ind, 1:] = return_most_common_venues(sm_grouped.iloc[ind, :], num_top_venues
)

neighbourhoods_venues_sorted
```

Out [68]:

	Neighbourhood	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	11th Most Common Venue
0	Agripada , BPC Jacob Circle , Chinchpokli , H...	Gym	Restaurant	Coffee Shop	Indian Restaurant	Racetrack	Tea Room	Zoo	Donut Shop	Farmers Market	Electronics Store	Dessert Shop
1	Ambewadi (Mumbai), Charni Road , Chaupati , G...	Indian Restaurant	Train Station	Juice Bar	Pizza Place	Snack Place	Italian Restaurant	Café	Restaurant	Sandwich Place	Breakfast Spot	Indie Movie Theatre
2	Antop Hill , B P T Colony , C G S Colony , Wad...	Smoke Shop	Gym / Fitness Center	Food	Dance Studio	Deli / Bodega	Department Store	Dessert Shop	Diner	Donut Shop	Electronics Store	Farmers Market
3	Asvini , Colaba , Colaba Bazar , Holiday Camp ...	Cricket Ground	Gym	Indian Restaurant	Bakery	Garden	Food	Deli / Bodega	Department Store	Dessert Shop	Diner	Donut Shop
4	B.P.Lane , Mandvi (Mumbai), Masjid , Null Bazar	Indian Restaurant	American Restaurant	Market	Dessert Shop	Convenience Store	Electronics Store	Middle Eastern Restaurant	Chinese Restaurant	Rest Area	Restaurant	Café
5	BEST Staff Quarters , Chamarbaug , Haffkin Ins...	Indian Restaurant	Chinese Restaurant	Playground	Bus Station	Restaurant	Sporting Goods Shop	Coffee Shop	Hotel	Diner	Field	Dessert Shop
6	Bazargate , CSMT , M.P.T. , Stock Exchange , T...	Indian Restaurant	Café	Irani Cafe	Seafood Restaurant	Fast Food Restaurant	Lounge	Clothing Store	Chinese Restaurant	Sandwich Place	Food Truck	Multiple
7	Bharat Nagar (Mumbai), Grant Road , N.S.Patka...	Electronics Store	Nightclub	Hotel	Fast Food Restaurant	Chinese Restaurant	Snack Place	Bakery	Lounge	Train Station	Vegetarian / Vegan Restaurant	Field
8	Central Building , Churchgate , Marine Lines	Fast Food Restaurant	Ice Cream Shop	Indian Restaurant	Café	Cricket Ground	Hotel	Italian Restaurant	Movie Theater	Train Station	Bakery	Coffee Shop
9	Chinchbunder , Noor Baug , Princess Dock	Harbor / Marina	Indian Restaurant	Boat or Ferry	Furniture / Home Store	Zoo	Food	Deli / Bodega	Department Store	Dessert Shop	Diner	Donut Shop
10	Chunabhatti , Raoli Camp , Sion , Transit Camp	Dance Studio	Mountain	Seafood Restaurant	Zoo	Flea Market	Deli / Bodega	Department Store	Dessert Shop	Diner	Donut Shop	Electronic Store

Cluster Areas

Run *k*-means to cluster the areas into 5 clusters.

```
In [123]: # set number of clusters
kclusters = 5

sm_grouped_clustering = sm_grouped.drop('Neighbourhood', 1)

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

# check cluster labels generated for each row in the dataframe
kmeans.labels_[0:20]
```

Out[123]: array([0, 1, 2, 1, 1, 1, 1, 1, 3, 4, 1, 1, 1, 1, 1, 1, 1, 1],
dtype=int32)

Let's create a new dataframe that includes the cluster as well as the top 10 venues for each area.

```
In [124]: sm_merged = south_mumbai_data

# add clustering labels
sm_merged['Cluster Labels'] = kmeans.labels_[1]

sm_merged = sm_merged.join(neighbourhoods_venues_sorted.set_index('Neighbourhood'), on='Neighbourhood')
```

Finally, let's visualize the resulting clusters

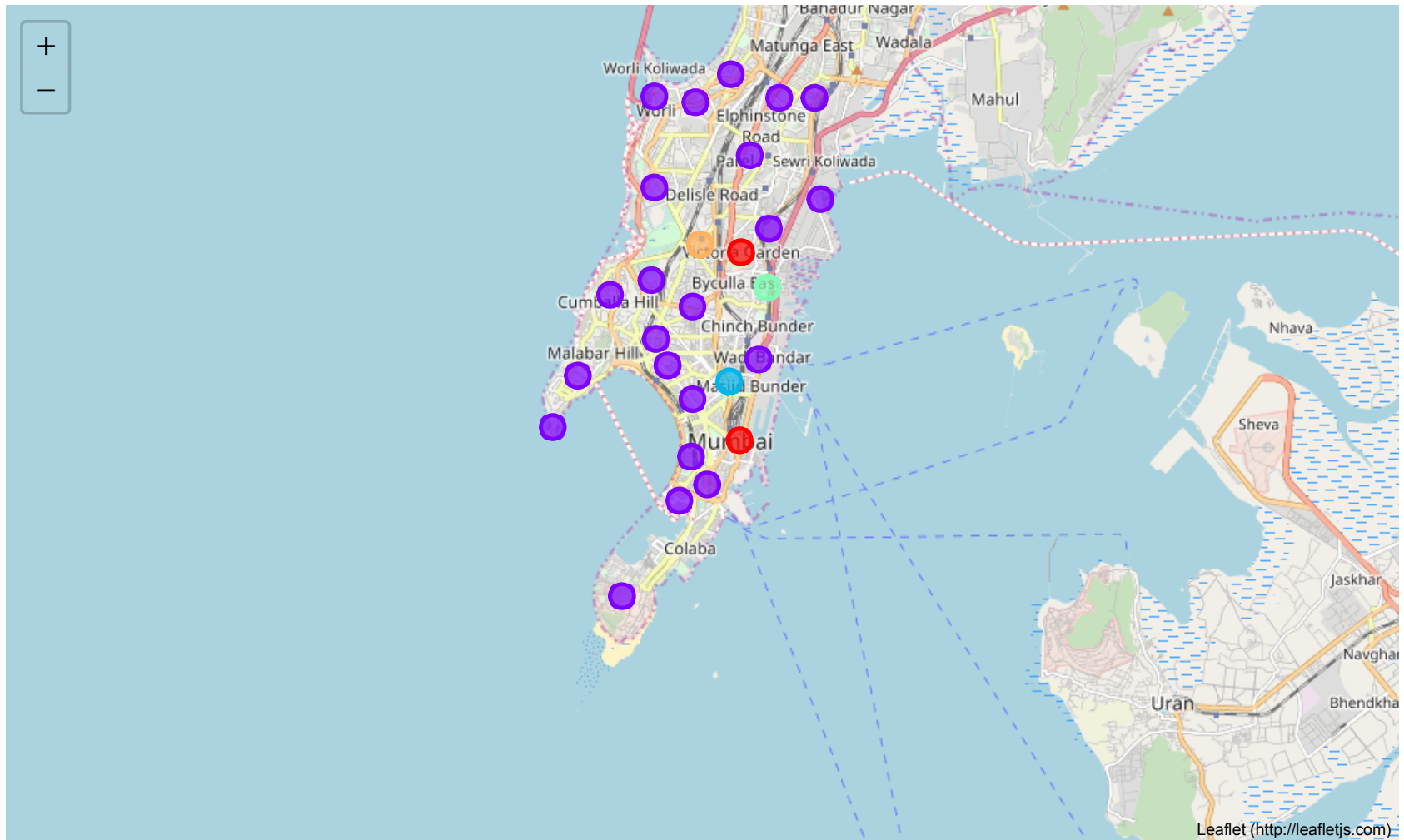
```
In [126]: # create map
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11.5)

# set color scheme for the clusters
x = np.arange(kclusters)
ys = [i+x+(i*x)**2 for i in range(kclusters)]
colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
rainbow = [colors.rgb2hex(i) for i in colors_array]

# add markers to the map
markers_colors = []
for lat, lon, poi, cluster in zip(sm_merged['Latitude'], sm_merged['Longitude'], sm_merged['Neighbourhood'], kmeans.labels_):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=8,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

Out[126]:



As majority of area of South Mumbai fall into Cluster 1 Cluster 1 details are useful to know which type of Restaurant service are popular in majority of South Mumbai


```
In [138]: sm_merged.loc[sm_merged['Cluster Labels'] == 1, sm_merged.columns[[1] + list(range(5, sm_merged.shape[1]))]].head(
)
```

Out[138]:

	Borough	Cluster Labels	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	11th Most Common Venue	12th Most Common Venue
0	South Mumbai	1	Indian Restaurant	Café	Irani Cafe	Seafood Restaurant	Fast Food Restaurant	Lounge	Clothing Store	Chinese Restaurant	Sandwich Place	Food Truck	Multiplex	
1	South Mumbai	1	Indian Restaurant	Café	Bar	Jewelry Store	Cheese Shop	Snack Place	Train Station	Fast Food Restaurant	Flea Market	Field	Zoo	Farm
2	South Mumbai	1	Indian Restaurant	American Restaurant	Market	Dessert Shop	Convenience Store	Electronics Store	Middle Eastern Restaurant	Chinese Restaurant	Rest Area	Restaurant	Café	BBQ
3	South Mumbai	1	Indian Restaurant	Train Station	Juice Bar	Pizza Place	Snack Place	Italian Restaurant	Café	Restaurant	Sandwich Place	Breakfast Spot	Indie Movie Theater	Fast Food Restaurant
4	South Mumbai	1	Cricket Ground	Gym	Indian Restaurant	Bakery	Garden	Food	Deli / Bodega	Department Store	Dessert Shop	Diner	Donut Shop	Electronics Store

In majority of part of South Mumbai "Indian Restaurant" is the most popular type of restaurant service.

In []: