

The Battle of Neighborhoods

Introduction to project

In this project, we will be working on New Delhi dataset. New Delhi is the Capital of India and has a population of about 2.18 crore (21.8 million). It has a diversity of religions and a lot of people from different parts of the country come here for work. It is a central hub of various business and political work. New Delhi are used interchangeably to refer to the National Capital Territory of Delhi (NCT), these are two distinct entities, with New Delhi forming a small part of Delhi. The National Capital Region is a much larger entity comprising the entire NCT along with adjoining districts in neighboring states.

There are various types of restaurants and famous for variety of food.

In this project we will visualize hot spots and various neighboring restaurants of Delhi.

Raw Data

To do this project, we will use the following data sets.

New Delhi Restaurants data that contains list Locality, Restaurant name, Rating along with their latitude and longitude.

- Data source : kaggle dataset
- Description : This data set contains the required information. And we will use this data set to explore various locality of new delhi city.

The link to the data is : <https://www.kaggle.com/shrutihehta/zomato-restaurants-data> (<https://www.kaggle.com/shrutihehta/zomato-restaurants-data>)

Data Creation and Gathering

Nearby places in each locality of new delhi city.

- Data source : Foursquare API
- Description : By using this api we will get all the venues in each neighborhood.

Procedure

- Collect the new delhi city data from Zomato kaggel dataset
- Using FourSquare API we will find all venues for each neighborhood.
- Filter out all venues that are nearby by locality.
- Using aggregative rating for each resturant to find the best places.
- Visualize the Ranking of neighborhoods using folium library(python)

LIBRARIES

```
In [1]: import types
import pandas as pd
from botocore.client import Config
import ibm_boto3
import numpy as np
import requests
from pandas.io.json import json_normalize
import matplotlib.cm as cm
import matplotlib.colors as colors
from sklearn.cluster import KMeans
#!/conda install -c conda-forge folium=0.5.0 --yes ]
import folium
#!/ pip install geocoder
import geocoder
```

Solving environment: done

Package Plan

environment location: /opt/conda/envs/Python36

added / updated specs:

- folium=0.5.0

The following packages will be downloaded:

package	build		
-----	-----		
branca-0.4.0	py_0	26 KB	conda-forge
python_abi-3.6	1_cp36m	4 KB	conda-forge
openssl-1.1.1f	h516909a_0	2.1 MB	conda-forge
ca-certificates-2020.4.5.1	hecc5488_0	146 KB	conda-forge
vincent-0.4.4	py_1	28 KB	conda-forge
altair-4.1.0	py_1	614 KB	conda-forge
folium-0.5.0	py_0	45 KB	conda-forge
certifi-2020.4.5.1	py36h9f0ad1d_0	151 KB	conda-forge
-----	-----		
Total:		3.1 MB	

The following NEW packages will be INSTALLED:

altair:	4.1.0-py_1	conda-forge
branca:	0.4.0-py_0	conda-forge
folium:	0.5.0-py_0	conda-forge
python_abi:	3.6-1_cp36m	conda-forge
vincent:	0.4.4-py_1	conda-forge

The following packages will be UPDATED:

ca-certificates:	2020.1.1-0	--> 2020.4.5.1-hecc5488_0	conda-forge
certifi:	2020.4.5.1-py36_0	--> 2020.4.5.1-py36h9f0ad1d_0	conda-forge
openssl:	1.1.1f-h7b6447c_0	--> 1.1.1f-h516909a_0	conda-forge

Downloading and Extracting Packages

branca-0.4.0	26 KB	#####	100%
python_abi-3.6	4 KB	#####	100%

```

openssl-1.1.1f      | 2.1 MB | ##### | 100%
ca-certificates-2020 | 146 KB | ##### | 100%
vincent-0.4.4       | 28 KB  | ##### | 100%
altair-4.1.0        | 614 KB | ##### | 100%
folium-0.5.0        | 45 KB  | ##### | 100%
certifi-2020.4.5.1  | 151 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Collecting geocoder
  Downloading https://files.pythonhosted.org/packages/4f/6b/13166c909ad2f2d76b929a4227c952630ebaf0d729f6317eb09cbceccbab/geocoder-1.38.1-py2.py3-none-any.whl (98kB)
    |#####| 102kB 7.0MB/s ta 0:00:011
Requirement already satisfied: future in /opt/conda/envs/Python36/lib/python3.6/site-packages (from geocoder) (0.17.1)
Requirement already satisfied: six in /opt/conda/envs/Python36/lib/python3.6/site-packages (from geocoder) (1.12.0)
Requirement already satisfied: click in /opt/conda/envs/Python36/lib/python3.6/site-packages (from geocoder) (7.0)
Requirement already satisfied: requests in /opt/conda/envs/Python36/lib/python3.6/site-packages (from geocoder) (2.21.0)
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 /opt/conda/envs/Python36/lib/python3.6/site-packages (from requests->geocoder) (2020.4.5.1)
Requirement already satisfied: idna<2.9,>=2.5 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from requests->geocoder) (2.8)
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from requests->geocoder) (1.24.1)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from requests->geocoder) (3.0.4)
Requirement already satisfied: decorator in /opt/conda/envs/Python36/lib/python3.6/site-packages (from ratelim->geocoder) (4.3.2)
Installing collected packages: ratelim, geocoder
Successfully installed geocoder-1.38.1 ratelim-0.1.6

```

LOADING AND FILTERING DATA

```
In [2]: def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_afe55041fc984c29ae3da0f99b4efe8b = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='xxxxxxxxxxxxxxxxxxxx',
    ibm_auth_endpoint="xxxxxxxxxxxxxxxxxxxxxxxx",
    config=Config(signature_version='oauth'),
    endpoint_url='xxxxxxxxxxxxxxxxxxxxxxxx')

body = client_afe55041fc984c29ae3da0f99b4efe8b.get_object(Bucket='xxxxxxxxxxxxxxxxxxxxxxxx',Key='Delhi_Data.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

# If you are reading an Excel file into a pandas DataFrame, replace `read_csv` by `read_excel` in the next statement.
rawdf = pd.read_csv(body,encoding='ISO-8859-1')
rawdf.head()
```

Out[2]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City	Century City Mall, Poblacion, Makati City, Mak...	121.027535	14.565443	French, Japanese, Desserts	...	Botswana Pula(P)
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City	Little Tokyo, Legaspi Village, Makati City, Ma...	121.014101	14.553708	Japanese	...	Botswana Pula(P)
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City	Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...	121.056831	14.581404	Seafood, Asian, Filipino, Indian	...	Botswana Pula(P)
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.056475	14.585318	Japanese, Sushi	...	Botswana Pula(P)
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.057508	14.584450	Japanese, Korean	...	Botswana Pula(P)

5 rows × 21 columns



```
In [3]: df=rawdf.loc[rawdf['City']=='New Delhi']
df.head()
```

Out[3]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Has Table booking
2560	18287358	Food Cloud	1	New Delhi	Aaya Nagar, New Delhi	Aaya Nagar	Aaya Nagar, New Delhi	0.000000	0.000000	Cuisine Varies	...	Indian Rupees(Rs.)	No
2561	18216944	Burger.in	1	New Delhi	84, Near Honda Showroom, Adchini, New Delhi	Adchini	Adchini, New Delhi	77.196923	28.535382	Fast Food	...	Indian Rupees(Rs.)	No
2562	313333	Days of the Raj	1	New Delhi	81/3, 1st Floor, Qutub Residency, Adchini, New...	Adchini	Adchini, New Delhi	77.197475	28.535493	North Indian, Seafood, Continental	...	Indian Rupees(Rs.)	Yes
2563	18384127	Dilli Ka Dhaba	1	New Delhi	66 A, Ground Floor, Sri Aurobindo Marg, Adchin...	Adchini	Adchini, New Delhi	77.198033	28.537547	South Indian, North Indian	...	Indian Rupees(Rs.)	No
2564	582	Govardhan	1	New Delhi	84, Adjacent Hero Motor Bike Showroom, Main Me...	Adchini	Adchini, New Delhi	77.196924	28.535523	South Indian, North Indian, Chinese	...	Indian Rupees(Rs.)	No

5 rows × 21 columns




```
In [4]: rdf= df[df.Longitude !=0.000000][['Restaurant Name','Locality','Longitude','Latitude','Cuisines','Aggregate rating','Rating text','Votes']]
rdf = rdf[rdf['Aggregate rating'] !=0.0]
rdf.head()
```

Out[4]:

	Restaurant Name	Locality	Longitude	Latitude	Cuisines	Aggregate rating	Rating text	Votes
2561	Burger.in	Adchini	77.196923	28.535382	Fast Food	3.2	Average	46
2562	Days of the Raj	Adchini	77.197475	28.535493	North Indian, Seafood, Continental	3.4	Average	45
2563	Dilli Ka Dhaba	Adchini	77.198033	28.537547	South Indian, North Indian	2.6	Average	11
2564	Govardhan	Adchini	77.196924	28.535523	South Indian, North Indian, Chinese	3.4	Average	238
2565	Mezbaan Grills	Adchini	77.198122	28.538134	Mughlai	3.1	Average	8

MAP to show restraunts clusters

```
In [5]: New_Delhi_Rest = folium.Map(location=[28.52, 77.25], zoom_start=12)

df_Res=rdf
X = df_Res['Latitude']
Y = df_Res['Longitude']
Z = np.stack((X, Y), axis=1)

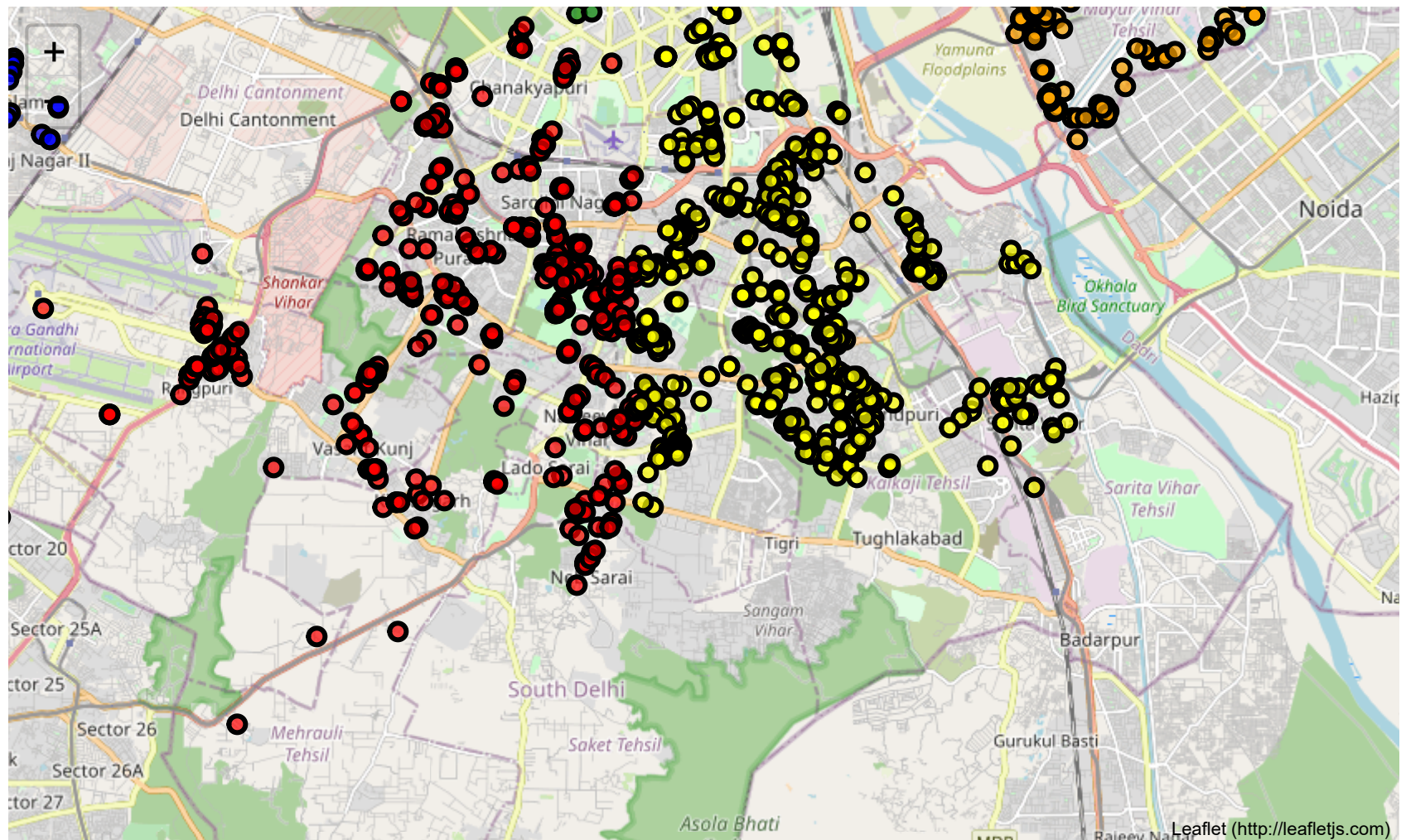
kmeans = KMeans(n_clusters=5, random_state=0).fit(Z)

clusters = kmeans.labels_
colors = ['red', 'green', 'blue', 'yellow', 'orange']
df_Res['Cluster'] = clusters

for latitude, longitude, Locality, cluster in zip(df_Res['Latitude'], df_Res['Longitude'], df_Res['Locality'], df_Res['Cluster']):
    label = folium.Popup(Locality, parse_html=True)
    folium.CircleMarker(
        [latitude, longitude],
        radius=5,
        popup=label,
        color='black',
        fill=True,
        fill_color=colors[cluster],
        fill_opacity=0.7).add_to(New_Delhi_Rest)

New_Delhi_Rest
```

Out[5]:



DATA TRANSFORMATION

```
In [6]: df_Res_Loc = df_Res.groupby('Locality').count()['Restaurant Name'].to_frame()
df_Res_rating= df_Res.groupby('Locality')['Aggregate rating'].mean().to_frame()
d_Cuisines = df_Res.groupby(['Locality'])['Cuisines'].agg(', '.join).reset_index()
d_R = df_Res.groupby(['Locality'])['Rating text'].unique().agg(', '.join).reset_index()
d_V = df_Res.groupby(['Locality'])['Votes'].sum().to_frame()
d_Lat = df_Res.groupby('Locality').mean()['Latitude'].to_frame()
d_Lng = df_Res.groupby('Locality').mean()['Longitude'].to_frame()
df_final = pd.merge(d_Lat,d_Lng,on='Locality').merge(df_Res_Loc, on='Locality').merge(d_Cuisines, on='Locality').merge(df_Res_rating,on = 'Locality').merge(d_R, on = 'Locality').merge(d_V, on = 'Locality')
df_final = df_final[df_final['Aggregate rating'] != 0.000000]
df_final.columns =['Locality','Lat','Lng', 'No_of_Restaurant','Cusines', 'Agg_Rating','Comments' , 'No_of_Votes']
df_final.head()
```

Out[6]:

	Locality	Lat	Lng	No_of_Restaurant	Cusines	Agg_Rating	Comments	No_of_Votes
0	ARSS Mall, Paschim Vihar	28.668945	77.101544	1	North Indian, South Indian, Chinese, Mithai, F...	3.100000	Average	117
1	Adchini	28.537063	77.197808	13	Fast Food, North Indian, Seafood, Continental,...	3.292308	Average, Good, Poor, Very Good	1560
2	Aditya Mega Mall, Karkardooma	28.656131	77.301266	4	Finger Food, North Indian, Mughlai, Pizza, Fas...	3.275000	Average, Good	434
3	Aerocity	28.553077	77.104270	2	Fast Food, Italian, Pizza, North Indian, Conti...	3.200000	Average	59
4	Aggarwal City Mall, Pitampura	28.690020	77.134650	3	North Indian, Chinese, Street Food, Mithai, No...	3.033333	Average	126

Define Foursquare Credentials and Find neighbors

```
In [7]: CLIENT_ID = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
CLIENT_SECRET = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
VERSION = 'xxxxxxx'

def getNearbyVenues(names, latitudes, longitudes, radius=500, LIMIT = 100):
    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)
        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)
        results = requests.get(url).json()["response"]['groups'][0]['items']
        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 = ['Locality',
                            'Locality Latitude',
                            'Locality Longitude',
                            'Venue',
                            'Venue Latitude',
                            'Venue Longitude',
                            'Venue Category']

    return(nearby_venues)

new_Delhi_venues = getNearbyVenues(names=df_final['Locality'],
                                    latitudes=df_final['Lat'],
                                    longitudes=df_final['Lng']
                                    )
```

ARSS Mall, Paschim Vihar
Adchini
Aditya Mega Mall, Karkardooma
Aerocity
Aggarwal City Mall, Pitampura
Aggarwal City Plaza, Rohini
Alaknanda
Ambience Mall, Vasant Kunj
Anand Lok
Anand Vihar
Andaz Delhi, Aerocity
Ansal Plaza Mall, Khel Gaon Marg
Asaf Ali Road
Ashok Vihar Phase 1
Ashok Vihar Phase 2
Ashok Vihar Phase 3
Barakhamba Road
Basant Lok Market, Vasant Vihar
Bellagio, Ashok Vihar Phase 2
Best Western Taurus Hotel, Mahipalpur
Bhikaji Cama Place
Chanakypuri
Chander Nagar
Chandni Chowk
Chawri Bazar
Chhatarpur
Chittaranjan Park
City Centre Mall, Rohini
City Square Mall, Rajouri Garden
Civil Lines
Community Centre, New Friends Colony
Connaught Place
Crescent Square Mall, Rohini
Cross River Mall, Karkardooma
Crowne Plaza Hotel, Rohini
Crowne Plaza, Mayur Vihar Phase 1
D Mall, Netaji Subhash Place
DDA Market, Kalu Sarai, Hauz Khas
DLF City Centre Mall, Shalimar Bagh
DLF Emporio Mall, Vasant Kunj
DLF Place Mall, Saket
DLF Promenade Mall, Vasant Kunj
DLF South Square, Sarojini Nagar

Daryaganj
Defence Colony
Delhi University-GTB Nagar
Dilli Haat, INA
Dilshad Garden
District Centre, Janakpuri
Dr. Zakir Hussain Marg
East Patel Nagar
East of Kailash
Epicuria Food Mall, Nehru Place
Eros Hotel, Nehru Place
Essex Farms
Feroze Shah Road
Friends Colony
GTB Nagar
Garden of Five Senses, Saket
Geeta Colony
Ginger Hotel, Vivek Vihar
Gourmet Hub, Pashim Vihar
Greater Kailash (GK) 1
Greater Kailash (GK) 2
Greater Kailash (GK) 3
Green Park
Gujranwala Town
Hauz Khas
Hauz Khas Village
Holiday Inn, Aerocity
Holiday Inn, Mayur Vihar
Hotel Broadway, Daryaganj
Hotel City Park, Pitampura
Hotel Regent Grand, Karol Bagh
Hotel The Royal Plaza, Janpath
Hyatt Regency, Bhikaji Cama Place
INA
IP Extension
ITC Maurya, Chanakyapuri
ITO
India Gate
JMD Kohinoor Mall, Greater Kailash
JNU
JW Marriott New Delhi
Jail Road
Jama Masjid

Janakpuri
Jangpura
Janpath
Jasola
Jaypee Siddharth, Rajendra Place
Jaypee Vasant Continental, Vasant Vihar
Jor Bagh
Kailash Colony
Kalkaji
Kamla Nagar
Kapashera
Karampura
Karkardooma
Karol Bagh
Kasbah, Greater Kailash (GK) 1
Kashmiri Gate
Khan Market
Kirti Nagar
Krishna Nagar
Lado Sarai
Lajpat Nagar 1
Lajpat Nagar 2
Lajpat Nagar 4
Lawrence Road
Laxmi Nagar
Le Meridien, Janpath
Lemon Tree Premier, Aerocity
Living Style Mall, Jasola
Lodhi Colony
Lodhi Road
MG Road
MGF Metropolitan Mall, Saket
MGM Club, Daryaganj
Mahipalpur
Maidens Hotel, Civil Lines
Majnu ka Tila
Malviya Nagar
Mandi House
Mathura Road
Mayapuri Phase 2
Mayur Vihar Phase 1
Mayur Vihar Phase 2
Mayur Vihar Phase 3

Mehrauli
Metro Walk Mall, Rohini
Model Town 1
Model Town 2
Model Town 3
Moments Mall, Kirti Nagar
Moti Bagh
Moti Nagar
Mukherjee Nagar
Munirka
Najafgarh
Naraina
Nehru Place
Netaji Subhash Place
New Friends Colony
Nizamuddin
Okhla Phase 1
Okhla Phase 2
PVR Anupam Complex
Pacific Mall, Tagore Garden
Paharganj
Palam
Palate of Delhi, Chanakyapuri
Panchsheel Park
Pandara Road Market
Pandav Nagar
Paschim Vihar
Patparganj
Piccadily Hotel, Janakpuri
Pitampura
Pragati Maidan
Prashant Vihar
Preet Vihar
Premier Inn, Shalimar Bagh
Pride Plaza Hotel, Aerocity
Punjabi Bagh
Qutab Institutional Area
R K Puram
Race Course
Radisson Blu Plaza Delhi, Mahipalpur
Radisson Blu, Paschim Vihar
Rajendra Place
Rajinder Nagar

Rajouri Garden
Rohini
Roseate House, Aerocity
SDA
Safdarjung
Sainik Farms
Saket
Sangam Courtyard, RK Puram
Sarita Vihar
Sarojini Nagar
Satyaniketan
Sector 15, Dwarka
Select Citywalk Mall, Saket
Shahdara
Shahpur Jat
Shakarpur
Shalimar Bagh
Shangri La's - Eros hotel, Janpath
Shanti Niketan Marg
Sheikh Sarai
Sheraton New Delhi Hotel, Saket
South Extension 1
South Extension 2
Southern Park Mall, Saket
Spark Mall, Kamla Nagar
Star City Mall, Mayur Vihar Phase 1
Subhash Nagar
Sunder Nagar
T3 Domestic Arrival, Aerocity
TDI Mall, Rajouri Garden
Tagore Garden
Taj Vivanta, Khan Market
The Ashok, Chanakyapuri
The Claridges, Aurangzeb Road
The Grand New Delhi, Vasant Kunj
The Imperial, Janpath
The India Mall, New Friends Colony
The Lalit New Delhi, Barakhamba Road
The Leela Ambience Convention Hotel
The Leela Palace, Chanakyapuri
The Lodhi, Lodhi Road
The Park, Connaught Place
The Suryaa New Delhi, New Friends Colony

The Taj Mahal Hotel, Mansingh Road
The Taj Palace Hotel, Chanakyapuri
The Uppal, Aerocity
The Village Restaurant Complex, Khel Gaon Marg
Tilak Nagar
Tughlakabad Institutional Area
Uday Park
Unity One Mall, Janakpuri
Uttam Nagar
V3S Mall, Laxmi Nagar
Vasant Kunj
Vasant Square Mall, Vasant Kunj
Vasant Vihar
Vasundhara Enclave
Vijay Nagar
Vikas Marg
Vikaspuri
Vivek Vihar
Wazirpur
West End Mall, Janak Puri
West Gate Mall, Rajouri Garden
West Patel Nagar
Worldmark 1, Aerocity
Yusuf Sarai
ibis New Delhi, Aerocity

In [9]: `new_Delhi_venues.head()`

Out[9]:

	Locality	Locality Latitude	Locality Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	ARSS Mall, Paschim Vihar	28.668945	77.101544	Subway	28.669999	77.102546	Sandwich Place
1	ARSS Mall, Paschim Vihar	28.668945	77.101544	Café Coffee Day	28.670009	77.102480	Coffee Shop
2	ARSS Mall, Paschim Vihar	28.668945	77.101544	Pizza Hut	28.670321	77.103853	Pizza Place
3	ARSS Mall, Paschim Vihar	28.668945	77.101544	Baljeet's Amritsari Koolcha	28.665768	77.100481	Indian Restaurant
4	ARSS Mall, Paschim Vihar	28.668945	77.101544	Little Chef	28.670000	77.101459	Chinese Restaurant

Using one hot encoder and grouping data together

```
In [10]: # one hot encoding
new_Delhi_onehot = pd.get_dummies(new_Delhi_venues[['Venue Category']], prefix="", prefix_sep="")

# add Locality column back to dataframe
new_Delhi_onehot['Locality'] = new_Delhi_venues['Locality']

# move Locality column to the first column
column_list = new_Delhi_onehot.columns.tolist()
column_number = int(column_list.index('Locality'))
column_list = [column_list[column_number]] + column_list[:column_number] + column_list[column_number+1:]
new_Delhi_onehot = new_Delhi_onehot[column_list]
New_Delhi_grouped = new_Delhi_onehot.groupby('Locality').mean().reset_index()
New_Delhi_grouped.head()
```

Out[10]:

	Locality	ATM	Accessories Store	Airport	Airport Food Court	Airport Lounge	Airport Service	Airport Terminal	American Restaurant	Arcade	...	Track Stadium	Trail	Train Station	I
0	ARSS Mall, Paschim Vihar	0.111111	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	
1	Adchini	0.000000	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	
2	Aditya Mega Mall, Karkardooma	0.000000	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	
3	Aerocity	0.000000	0.0	0.0	0.0	0.142857	0.0	0.142857	0.0	0.0	...	0.0	0.0	0.0	
4	Aggarwal City Mall, Pitampura	0.000000	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	...	0.0	0.0	0.0	

5 rows × 208 columns

Create the new dataframe and display the top 10 venues for each Locality.

```
In [12]: 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]

num_top_venues = 10
indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Locality']
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
Locality_venues_sorted = pd.DataFrame(columns=columns)
Locality_venues_sorted['Locality'] = New_Delhi_grouped['Locality']

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

Locality_venues_sorted.head()
```

Out[12]:

	Locality	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
0	ARSS Mall, Paschim Vihar	Indian Restaurant	Sandwich Place	Market	Multicuisine Indian Restaurant	Pizza Place	Coffee Shop	Chinese Restaurant	ATM	American Restaurant	Donut Shop
1	Adchini	Indian Restaurant	Café	Parsi Restaurant	Pub	Restaurant	Women's Store	Fish Market	Frozen Yogurt Shop	Fried Chicken Joint	French Restaurant
2	Aditya Mega Mall, Karkardooma	Indian Restaurant	Pizza Place	Shopping Mall	Multiplex	Hotel	Café	Women's Store	Flea Market	Frozen Yogurt Shop	Fried Chicken Joint
3	Aerocity	Hotel	Rental Car Location	Airport Lounge	Airport Terminal	Fast Food Restaurant	Coffee Shop	Flea Market	Gaming Cafe	Furniture / Home Store	Frozen Yogurt Shop
4	Aggarwal City Mall, Pitampura	Department Store	Fast Food Restaurant	Café	Chinese Restaurant	Eastern European Restaurant	Electronics Store	Garden	Gaming Cafe	Furniture / Home Store	Frozen Yogurt Shop

Clustering the locality

```
In [13]: kclusters = 5

New_Delhi_clustering = New_Delhi_grouped.drop('Locality', 1)

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

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

# add clustering labels
New_Delhi_merged = df_final.head(239)
New_Delhi_merged['Cluster Labels'] = kmeans.labels_

# merge New_Delhi_grouped with df_Chinese to add Latitude/Longitude for each Locality
New_Delhi_merged = New_Delhi_merged.join(Locality_venues_sorted.set_index('Locality'), on='Locality')

New_Delhi_merged.head()
```



```
/opt/conda/envs/Python36/lib/python3.6/site-packages/ipykernel/__main__.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

Out[13]:

	Locality	Lat	Lng	No_of_Restaurant	Cusines	Agg_Rating	Comments	No_of_Votes	Cluster Labels	1st Most Common Venue	2nd Most Common Venue
0	ARSS Mall, Paschim Vihar	28.668945	77.101544	1	North Indian, South Indian, Chinese, Mithai, F...	3.100000	Average	117	0	Indian Restaurant	Sar
1	Adchini	28.537063	77.197808	13	Fast Food, North Indian, Seafood, Continental,...	3.292308	Average, Good, Poor, Very Good	1560	0	Indian Restaurant	
2	Aditya Mega Mall, Karkardooma	28.656131	77.301266	4	Finger Food, North Indian, Mughlai, Pizza, Fas...	3.275000	Average, Good	434	0	Indian Restaurant	
3	Aerocity	28.553077	77.104270	2	Fast Food, Italian, Pizza, North Indian, Conti...	3.200000	Average	59	1	Hotel	Ren' Lc
4	Aggarwal City Mall, Pitampura	28.690020	77.134650	3	North Indian, Chinese, Street Food, Mithai, No...	3.033333	Average	126	4	Department Store	Fas Rest

Mapping the clusters

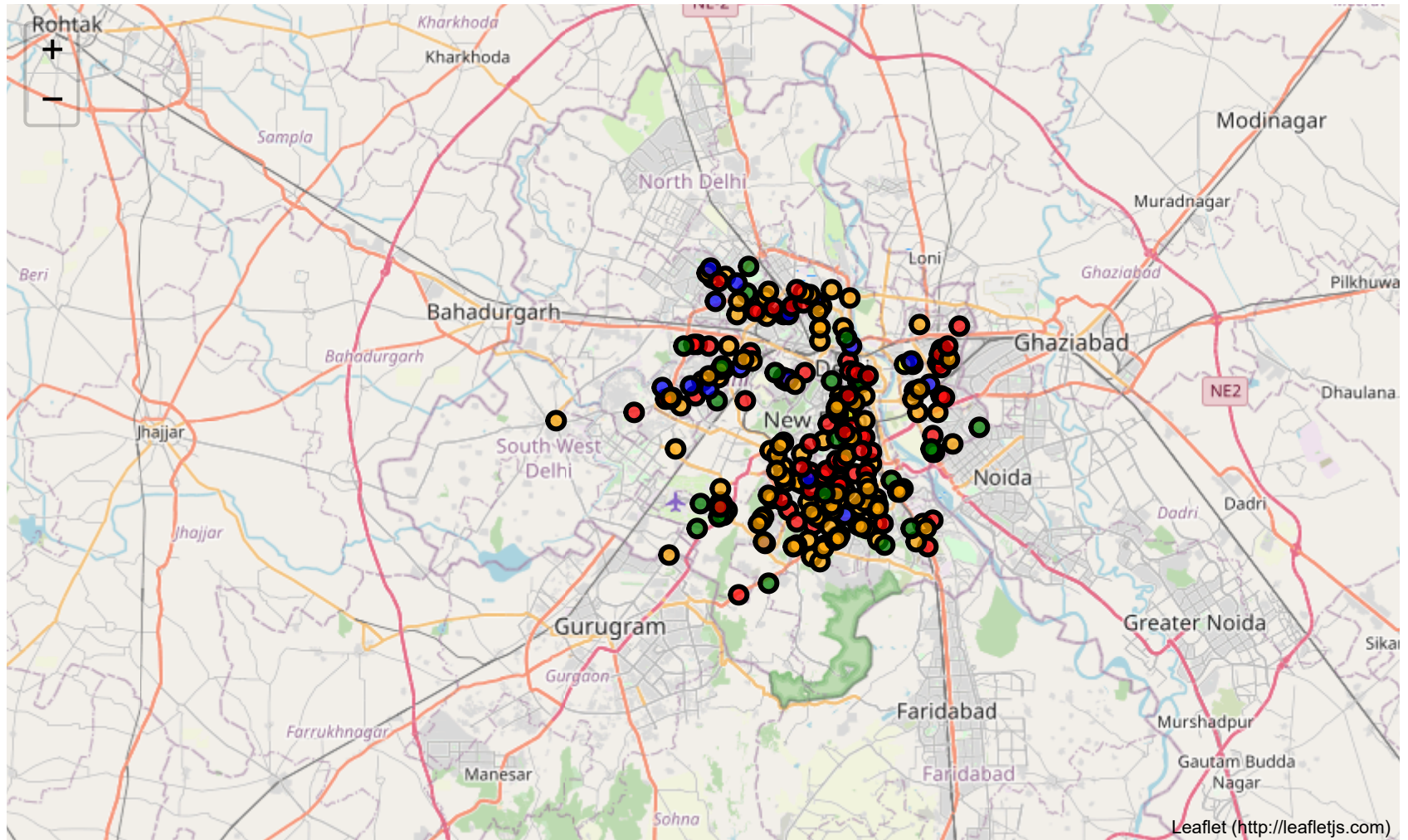
```

In [16]: map_clusters = folium.Map(location=[latitude, longitude], zoom_start=10)
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]
colors = ['red', 'green', 'blue', 'yellow', 'orange']

markers_colors = []
for lat, lon, poi, cluster in zip(New_Delhi_merged['Lat'], New_Delhi_merged['Lng'], New_Delhi_merged['Localit
y'], New_Delhi_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color='black',
        fill=True,
        fill_color=colors[cluster],
        fill_opacity=0.7).add_to(map_clusters)
map_clusters

```

Out[16]:



Examining the clusters

```
In [14]: New_Delhi_merged.loc[New_Delhi_merged['Cluster Labels'] == 0, New_Delhi_merged.columns[[1] + list(range(5, New_Delhi_merged.shape[1]))]].head()
```

Out[14]:

	Lat	Agg_Rating	Comments	No_of_Votes	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
0	28.668945	3.100000	Average	117	0	Indian Restaurant	Sandwich Place	Market	Multicuisine Indian Restaurant	Pizza Place	Coffee Shop	Cafe
1	28.537063	3.292308	Average, Good, Poor, Very Good	1560	0	Indian Restaurant	Café	Parsi Restaurant	Pub	Restaurant	Women's Store	
2	28.656131	3.275000	Average, Good	434	0	Indian Restaurant	Pizza Place	Shopping Mall	Multiplex	Hotel	Café	W
13	28.690182	3.090000	Average, Good, Poor	971	0	Indian Restaurant	Smoke Shop	Pizza Place	BBQ Joint	Market	Snack Place	I
14	28.693969	3.378571	Average, Good	2343	0	Pizza Place	Indian Restaurant	South Indian Restaurant	Sandwich Place	Coffee Shop	Asian Restaurant	

```
In [15]: New_Delhi_merged.loc[New_Delhi_merged['Cluster Labels'] == 1, New_Delhi_merged.columns[[1] + list(range(5, New_Delhi_merged.shape[1]))]].head()
```

Out[15]:

	Lat	Agg_Rating	Comments	No_of_Votes	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
3	28.553077	3.20	Average	59	1	Hotel	Rental Car Location	Airport Lounge	Airport Terminal	Fast Food Restaurant	Coffee Shop	Flea Market
10	28.554807	3.00	Average	4	1	Hotel	Coffee Shop	Indian Restaurant	Gym / Fitness Center	Food Court	Cocktail Bar	Acces
12	28.641698	3.05	Average	17	1	Hotel	Movie Theater	Indian Restaurant	Hostel	Flea Market	Garden	G
19	28.548827	2.70	Average	9	1	Hotel	Italian Restaurant	Buffet	Bed & Breakfast	Hotel Bar	Indian Restaurant	Acces
25	28.495624	2.70	Average	23	1	Japanese Restaurant	Indian Restaurant	Donut Shop	Hotel	Women's Store	Flea Market	G

```
In [17]: New_Delhi_merged.loc[New_Delhi_merged['Cluster Labels'] == 2, New_Delhi_merged.columns[[1] + list(range(5, New_Delhi_merged.shape[1]))]].head()
```

Out[17]:

	Lat	Agg_Rating	Comments	No_of_Votes	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
5	28.700516	3.040000	Average, Good, Poor	594	2	Fast Food Restaurant	Pizza Place	Gym / Fitness Center	Shopping Mall	Multiplex	Electronics Store	Si
15	28.691136	3.390909	Average, Good, Very Good	485	2	Gym / Fitness Center	Pizza Place	South Indian Restaurant	Fast Food Restaurant	BBQ Joint	Food	
22	28.654105	3.085714	Average	80	2	Pizza Place	Women's Store	Flea Market	Garden	Gaming Cafe	Furniture / Home Store	
57	28.699837	3.376471	Average, Good, Very Good	2460	2	Pizza Place	Food Truck	Women's Store	Flea Market	Garden	Gaming Cafe	Fu
101	28.667644	3.300000	Average	132	2	Pizza Place	American Restaurant	Bus Station	Metro Station	Fast Food Restaurant	Historic Site	Ele

```
In [18]: New_Delhi_merged.loc[New_Delhi_merged['Cluster Labels'] ==3 , New_Delhi_merged.columns[[1] + list(range(5, New_Delhi_merged.shape[1]))]].head()
```

Out[18]:

	Lat	Agg_Rating	Comments	No_of_Votes	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
55	28.617102	4.200000	Very Good	3010	3	Indian Restaurant	Women's Store	Flea Market	Garden	Gaming Cafe	Furniture / Home Store	Frozen Yogurt Shop
59	28.654883	3.053846	Average, Good	262	3	Indian Restaurant	Women's Store	Flea Market	Garden	Gaming Cafe	Furniture / Home Store	Frozen Yogurt Shop

```
In [19]: New_Delhi_merged.loc[New_Delhi_merged['Cluster Labels'] == 4, New_Delhi_merged.columns[[1] + list(range(5, New_Delhi_merged.shape[1]))]].head()
```

Out[19]:

	Lat	Agg_Rating	Comments	No_of_Votes	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 Common Venue
4	28.690020	3.033333	Average	126	4	Department Store	Fast Food Restaurant	Café	Chinese Restaurant	Eastern European Restaurant	Electronics Store	G
6	28.527088	3.117391	Average, Good, Poor	1012	4	Market	Chinese Restaurant	Hotel	Bar	Steakhouse	Coffee Shop	G
7	28.541298	3.425000	Average, Good, Very Good	2460	4	Coffee Shop	Fast Food Restaurant	Asian Restaurant	Clothing Store	Italian Restaurant	Café	Sho
8	28.555599	3.800000	Average, Very Good	2411	4	Stadium	Café	Dessert Shop	Golf Course	Other Great Outdoors	Coffee Shop	\
9	28.659870	3.281481	Average, Good	1309	4	Café	ATM	Burger Joint	Park	Burrito Place	Pharmacy	Hot

Thank you!