REPORT

Capstone Project - The Battle of Neighbourhoods

(Finding nearby places for tourist in Mumbai)

INTRODUCTION:

Tourism in Mumbai is an industry that attracts almost 6 million tourists per year, making it the 30th-most visited location worldwide. According to United Nations, as of 2018, Mumbai was the second most populous city in India after Delhi and the seventh most populous city in the world with a population of 19.98 million.

Mumbai offers natural heritage and modern entertainment including leisure spots, beaches, cinemas, studios, holy places, amusement parks and historical monuments. Transport options include air, road, train and ship.

As tourists would be interested in exploring places they would need a list of places and an itinerary. In order to make it easier they can use this program to identify all the places of interest in no time and it would be reliable as the places are located nearby. Thus, this program will help to find the underlisted places, nearby:

1. Restaurants
2. Parks
3. Hotels
4. Cafeteria
5. Shopping

PROBLEMS:

To find the answers to the following questions

1. List and visualise all major restaurants in Mumbai.
2. List and visualise all major Parks in Mumbai
3. List and visualise all major Hotels in Mumbai
4. List and visualise all major Cafeteria in Mumbai
5. List and visualise all major Shopping in Mumbai

DATA DESCRIPTION

For this project we need the following data:

1. Restaurants in Mumbai (India)
   1.1. Data source: Foursquare API
   1.2. Description: By using API we will get all restaurants in neighbourhood. We can filter these restaurants to only get nearby restaurants.
2. Parks in Mumbai (India)
   2.1 Data source: Foursquare API
   2.2 Description: By using API we will get all hotels in neighbourhood. We can filter these restaurants to only get nearby parks.
3. Hotels in Mumbai (India)
   3.1. Data source: Foursquare API
   3.2. Description: By using API we will get all hotels in neighbourhood. We can filter these restaurants to only get nearby hotels.
4. Cafeteria in Mumbai (India)
   4.1. Data source: Foursquare API
   4.2. Description: By using API we will get all cafeteria in neighbourhood. We can filter these restaurants to only get nearby cafeteria.
5. Shopping in Mumbai (India)
   5.1. Data source: Foursquare API
   5.2. Description: By using API we will get all shopping in neighbourhood. We can filter these restaurants to only get nearby shopping.

METHODOLOGY

1. We begin by searching the city, its longitude and latitude:

```
: #define city location, latitude, longitude
  city = 'Mumbai'
  geolocator = Nominatim(user_agent="foursquare_agent")
  location = geolocator.geocode(city)
  latitude = location.latitude
  longitude = location.longitude
  print(latitude, longitude)
```

```
18.9387711 72.8353355
```

2. Then searching the Hotels from the searched city:

	id	name	categories	referralId	hasPerk	location.address	location.crossStreet	location.lat	location.lng	loca
0	4e135a03a809291902c7d32f	Hotel Traveller's Inn	[{'id': '4bf58dd8d48988d1f8931735', 'name': 'B...	v-1590405162	False	26, ADI Marzban Path	Behind Cafe Universal, Ballard Estate, Fort	18.936039	72.838000	
1	4bcf4de40ffdce72215fb2c0	Residency Hotel	[{'id': '4bf58dd8d48988d1fa931735', 'name': 'H...	v-1590405162	False	26, Corner of D.N. Road and Rustom Sidhwa Marg...	NaN	18.934978	72.833481	
2	4cbc460f9552b60c8d81e48b	City Palace Hotel	[{'id': '4bf58dd8d48988d1fa931735', 'name': 'H...	v-1590405162	False	NaN	NaN	18.938970	72.835610	
3	4ffaf206e4b0a0306dd1271d	Hotel City Palace	[]	v-1590405162	False	Near CST	NaN	18.938474	72.834599	
4	5321eed6498e273e35971c04	sheel hotel	[{'id': '4bf58dd8d48988d1fa931735', 'name': 'H...	v-1590405162	False	NaN	NaN	18.938211	72.835725	

3. Now cleaning the search result ie. Deleting unnecessary columns, deleting rows with none values, deleting rows which its category is not Hotel, deleting rows which has duplicate hotel's name.

```
In [112]: # choose the hotel which has the same postalCode with the event space
          df_hotel = df_hotels[df_hotels.postalCode == '400001']
          df_hotel
```

Out[112]:

	name	categories	address	lat	lng	postalCode	state
1	Residency Hotel	Hotel	26, Corner of D.N. Road and Rustom Sidhwa Marg...	18.934978	72.833481	400001	Mahārāshtra
21	Benazeer Hotel	Hotel	16 Gunbow Street	18.935140	72.833948	400001	Mahārāshtra

4. Now searching the park in Mumbai:

	id	name	categories	referralId	hasPerk	location.address	location.crossStreet	location.lat	location.lng
0	511e3da9e4b092d8057a44b6	Harish Mahindra Children's Park	[{'id': '4bf58dd8d48988d163941735', 'name': 'P...	v-1590405164	False	Breach Candy, Behind American Counsulate	Bhulabhai Desai Road	18.914867	72.823688
1	52f2771d11d2afc107a8206d	Traditions at Reagan Park	[{'id': '5267e4d9e4b0ec79466e48c6', 'name': 'C...	v-1590405164	False	1176 Kingwood Drive	NaN	18.949804	72.834699
2	4c84af7ed34ca14342c74080	Park Avenue	[{'id': '4bf58dd8d48988d103951735', 'name': 'C...	v-1590405164	False	in High Street Phoenix,	NaN	18.996986	72.826677
3	5b430c7f2b274a002c36a513	Rivali Park	[{'id': '5032885091d4c4b30a586d66', 'name': 'R...	v-1590405164	False	Rivali Park, Western Express Highway,Borivali ...	NaN	18.929599	72.828181
4	4c9174ec238c6dcb067ec055	Priyadarshini Park	[{'id': '4bf58dd8d48988d163941735', 'name': 'P...	v-1590405164	False	Nepean Sea Road	NaN	18.957515	72.799614

5. Cleaning the collected park dataframe:

In [119]:
```
# delete rows which its category is not Park
df_park = clean_park_dataframe3[clean_park_dataframe3.categories == 'Park']
df_park
```

Out[119]:

	name	categories	address	lat	lng	postalCode	state
0	Harish Mahindra Children's Park	Park	Breach Candy, Behind American Counsulate	18.914867	72.823688	400026	Mahārāshtra
4	Priyadarshini Park	Park	Nepean Sea Road	18.957515	72.799614	400006	Mahārāshtra

6. Now searching the nearby restaurants in Mumbai:

```
use pandas.json_normalize instead
  """
```

Out[122]:

	id	name	categories	referralId	hasPerk	location.address	location.crossStreet	location.lat	location.lng	loc
0	5263f0f5498edc1ad5f3270f	Nanumal Bhojraj Restaurant - Fort	[{'id': '4bf58dd8d48988d10f941735', 'name': 'I...	v-1590405165	False	Near GPO, Fort, VT	S. Bhagat Singh Road	18.937974	72.837663	
1	4e4c0a4122713bd908c641ff	Farhang Restaurant	[{'id': '4bf58dd8d48988d1cc941735', 'name': 'S...	v-1590405165	False	Opposite GPO	Ballard Estate	18.938165	72.837917	
2	4fb5276be4b0b34b9ac27a1b	Iran Like Restaurant	[{'id': '4bf58dd8d48988d10f941735', 'name': 'I...	v-1590405165	False	Palton Road, Near Haj House	NaN	18.945343	72.836809	
3	4d9f2bfe9b91a1cda6c365c0	New Majestic Restaurant	[{'id': '4bf58dd8d48988d10f941735', 'name': 'I...	v-1590405165	False	Opp CST	NaN	18.938972	72.835517	
4	4f489f23e4b018635344413a	Panchratna restaurant & bar	[{'id': '4bf58dd8d48988d10f941735', 'name': 'I...	v-1590405165	False	Opp. Metro Cinema	Marine Lines	18.942900	72.826820	

7. Cleaning the collected restaurant dataframe:

```
In [125]: # delete rows with none values
df_Restaurant = clean_Restaurant_dataframe2.dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)
df_Restaurant
```

Out[125]:

	name	categories	address	lat	lng	postalCode	state
0	Nanumal Bhojraj Restaurant - Fort	Indian Restaurant	Near GPO, Fort, VT	18.937974	72.837663	400001	Mahārāshtra
2	Iran Like Restaurant	Indian Restaurant	Palton Road, Near Haj House	18.945343	72.836809	400001	Mahārāshtra
5	Stadium Restaurant & Stores	Indian Restaurant	Veer Nariman Road	18.933173	72.826929	400023	Mahārāshtra
9	Nanumal Bhojraj Restaurant - Masjid Bunder	Indian Restaurant	Gaumukh Bhawan, Masjid Bunder West, Near Masji...	18.953200	72.837577	400009	Mahārāshtra
18	Lalit restaurant	Bar	Cawasji Patel road	18.935002	72.833791	400001	Mahārāshtra
21	Pine Restaurant	Indian Restaurant	Nawab Tank Bunder, Dockyard	18.964799	72.844015	400010	Mahārāshtra
22	Excellensea Restaurant	BBQ Joint	Mint Road	18.935266	72.837240	400001	Mahārāshtra
24	National Restaurant	Food	opp fort market	18.935112	72.836896	400001	Mahārāshtra
27	Fountain Sizzlers Restaurant	Steakhouse	57 Mahatma Gandhi Road	18.931366	72.831462	400 023	Mahārāshtra

```
In [ ]: #Cafeteria
```

8. Now collecting the nearby cafeteria in Mumbai:

```
"""
```

	id	name	categories	referralId	hasPerk	location.address	location.lat	location.lng	location.labeledLatLngs
0	4e81776030f86985e40abb54	Cafeteria	[{'id': '4bf58dd8d48988d16d941735', 'name': 'C...	v-1590405168	False	Breach Candy Hospital	18.971833	72.804981	[{'label': 'display', 'lat': 18.97183316652913...
1	50050da9e4b0d73a8dfa7ee9	Cafeteria at Tata Capital	[{'id': '4bf58dd8d48988d120951735', 'name': 'F...	v-1590405168	False	One Forbes	18.930769	72.835434	[{'label': 'display', 'lat': 18.930769, 'lng':...
2	4e2943f02271752a459cb122	Mumbai High Court Cafeteria	[{'id': '4bf58dd8d48988d120951735', 'name': 'F...	v-1590405168	False	Fort	18.933198	72.829170	[{'label': 'display', 'lat': 18.93319755884302...
3	4bdfcc5affdec9286e31eda1	Cafeteria at Indian Hotels	[]	v-1590405168	False	Oxford Towers	18.922937	72.832849	[{'label': 'display', 'lat': 18.922937, 'lng':...
4	4d53497cd4a6721ec05589ac	Cafeteria	[]	v-1590405168	False	Mumbai central station	18.970839	72.820636	[{'label': 'display', 'lat': 18.97083858982881...

9. Cleaning the cafeteria dataframe:

```
In [131]: # delete rows with none values
df_Cafeteria = clean_Cafeteria_dataframe2.dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)
df_Cafeteria
```

Out[131]:

	name	categories	address	lat	lng	postalCode	state
0	Cafeteria	Café	Breach Candy Hospital	18.971833	72.804981	400026	Mahārāshtra
5	KES College of Commerce & Arts Cafeteria	College Cafeteria	Parekh lane	18.948587	72.829424	400067	Mahārāshtra
6	Cafeteria - P2 Building	Cafeteria	Next to Phoenix Building	18.992866	72.823965	400013	Mahārāshtra
7	Cafeteria, Lodha Excelus	Cafeteria	9th Floor, Lodha Excelus,	18.994027	72.834380	400011	Mahārāshtra
11	Jai Hind College	College Academic Building	A Road, Churchgate	18.934689	72.824822	400 020	Mahārāshtra
13	Leo Burnett Cafeteria	Cafeteria	36-A Big Apple	18.998785	72.840376	400012	Mahārāshtra
17	Mahindra Towers Cafeteria	Cafeteria	Worli	19.006044	72.820630	400030	Mahārāshtra
18	Mahindra Holidays Cafeteria	Snack Place	Mahindra Holidays, Mahindra Towers, G.M. Bhosl...	19.005692	72.822309	400018	Mahārāshtra
19	burhani college	College Arts Building	Mazagoan	18.970446	72.835390	400010	Mahārāshtra
20	VIT Executive Cafeteria	College Cafeteria	Vidyalankar Institute Of Technology	19.021675	72.870364	400037	Mahārāshtra
25	Indigo Consulting	Building	The Big Apple	18.998891	72.840350	400012	Mahārāshtra
26	BloombergUTV	Food Court	Parijaat House, 1076, Dr. E Moses Road,	18.996397	72.819472	400018	Mahārāshtra

```
In [ ]: #Shopping
```

10. Now collecting the nearby shopping:

In [134]:
```
# assign relevant part of JSON to venues
venues = sresults['response']['venues']

# tranform venues into a dataframe
Shopping_dataframe = json_normalize(venues)
Shopping_dataframe.head()
```

```
C:\Users\Acer\anaconda3\lib\site-packages\ipykernel_launcher.py:5: FutureWarning: pandas.io.json.json_normalize is deprecated,
use pandas.json_normalize instead
  """
```

Out[134]:

	id	name	categories	referralId	hasPerk	location.lat	location.lng	location.labeledLatLngs	location.distance
0	4e54cc34ae60e3d6b7f18c28	Ashoka Shopping Centre	[{'id': '4bf58dd8d48988d124941735', 'name': 'O...	v-1590405170	False	18.945182	72.832121	[{'label': 'display', 'lat': 18.94518213019515...	789
1	527b9fd8498e3c08ce24052d	Ashrafi Shopping Centre	[{'id': '4bf58dd8d48988d124941735', 'name': 'O...	v-1590405170	False	18.948688	72.836013	[{'label': 'display', 'lat': 18.9486880047746,...	1106
2	4ecf3a6ce5faa5ec01659919	Bhangwadi Shopping Arcade	[{'id': '4bf58dd8d48988d124941735', 'name': 'O...	v-1590405170	False	18.946791	72.827999	[{'label': 'display', 'lat': 18.94679133422852...	1180

11. Now cleaning the shopping dataframe:

In [136]:
```
# delete unnecessary columns
clean_Shopping_dataframe2= clean_Shopping_dataframe.drop(['cc', 'city', 'country', 'distance', 'formattedAddress',\
                                    'labeledLatLngs', 'id'], axis=1)
clean_Shopping_dataframe2
```

Out[136]:

	name	categories	lat	lng	state
0	Ashoka Shopping Centre	Office	18.945182	72.832121	Mahārāshtra
1	Ashrafi Shopping Centre	Office	18.948688	72.836013	NaN
2	Bhangwadi Shopping Arcade	Office	18.946791	72.827999	NaN

12. Now creating the dataframe of hotels, shopping stores and cafeteria

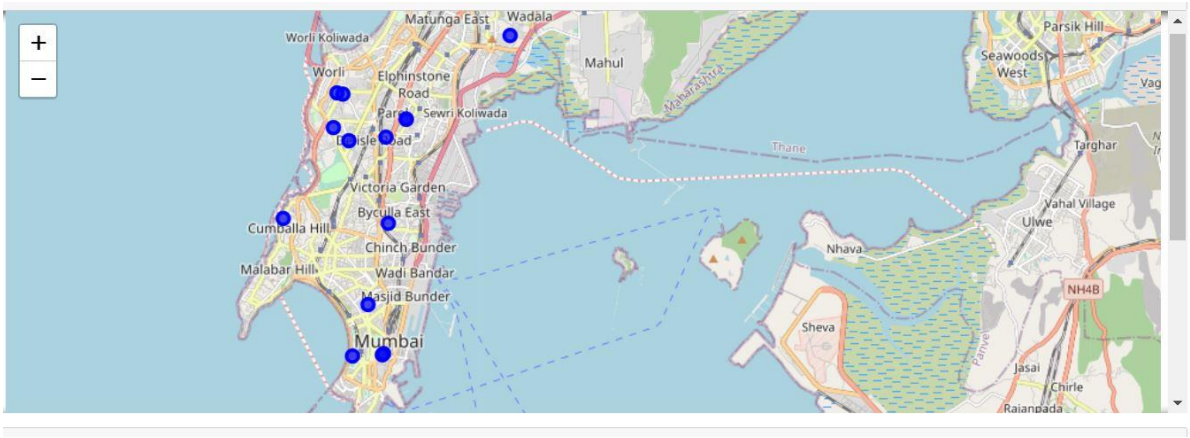
In [ ]:
```
# create dataframe of hotels, shopping stores and Cafeteria
```

In [137]:
```
hotel_neighbourhood_df = pd.concat([df_hotel, df_Cafeteria, df_Shopping], ignore_index=True)
hotel_neighbourhood_df
```

Out[137]:

	name	categories	address	lat	lng	postalCode	state
0	Residency Hotel	Hotel	26, Corner of D.N. Road and Rustom Sidhwa Marg...	18.934978	72.833481	400001	Mahārāshtra
1	Benazeer Hotel	Hotel	16 Gunbow Street	18.935140	72.833948	400001	Mahārāshtra
2	Cafeteria	Café	Breach Candy Hospital	18.971833	72.804981	400026	Mahārāshtra
3	KES College of Commerce & Arts Cafeteria	College Cafeteria	Parekh lane	18.948587	72.829424	400067	Mahārāshtra
4	Cafeteria - P2 Building	Cafeteria	Next to Phoenix Building	18.992866	72.823965	400013	Mahārāshtra
5	Cafeteria, Lodha Excelus	Cafeteria	9th Floor, Lodha Excelus,	18.994027	72.834380	400011	Mahārāshtra
6	Jai Hind College	College Academic Building	A Road, Churchgate	18.934689	72.824822	400 020	Mahārāshtra
7	Leo Burnett Cafeteria	Cafeteria	36-A Big Apple	18.998785	72.840376	400012	Mahārāshtra
8	Mahindra Towers Cafeteria	Cafeteria	Worli	19.006044	72.820630	400030	Mahārāshtra
9	Mahindra Holidays Cafeteria	Snack Place	Mahindra Holidays, Mahindra Towers, G.M. Bhosl...	19.005692	72.822309	400018	Mahārāshtra
10	burhani college	College Arts Building	Mazagoan	18.970446	72.835390	400010	Mahārāshtra
11	VIT Executive Cafeteria	College Cafeteria	Vidyalankar Institute Of Technology	19.021675	72.870364	400037	Mahārāshtra
12	Indigo Consulting	Building	The Big Apple	18.998891	72.840350	400012	Mahārāshtra
13	BloombergUTV	Food Court	Parijaat House, 1076, Dr. E Moses Road,	18.996397	72.819472	400018	Mahārāshtra

13. Visualization the above dataframe ie. Of hotels, shopping and cafeteria:



14. Making dataframe of park,restaurant and cafeteria:

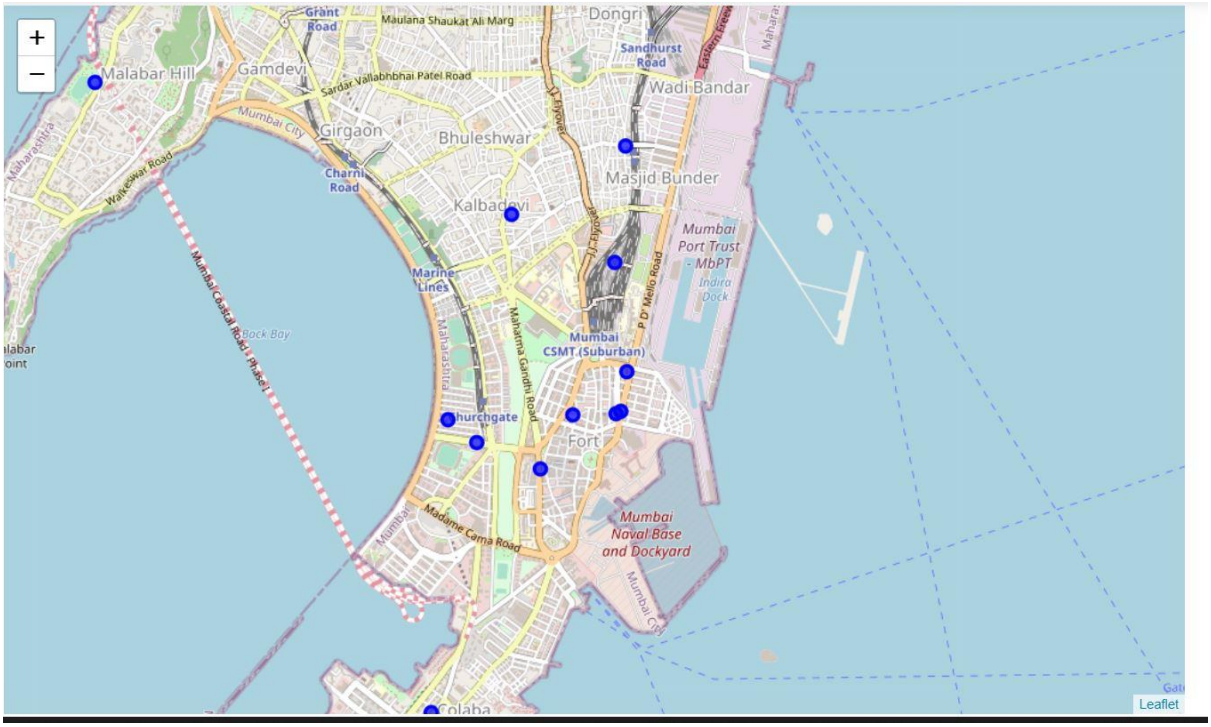
```
In [ ]: #create dataframe of Park, Restaurant and Cafeteria

In [139]:
park_neighbourhood_df = pd.concat([df_park, df_Restaurant, df_Cafeteria,], ignore_index=True)
park_neighbourhood_df
```

Out[139]:

	name	categories	address	lat	lng	postalCode	state
0	Harish Mahindra Children's Park	Park	Breach Candy, Behind American Counsulate	18.914867	72.823688	400026	Mahārāshtra
1	Priyadarshini Park	Park	Nepean Sea Road	18.957515	72.799614	400006	Mahārāshtra
2	Nanumal Bhojraj Restaurant - Fort	Indian Restaurant	Near GPO, Fort, VT	18.937974	72.837663	400001	Mahārāshtra
3	Iran Like Restaurant	Indian Restaurant	Palton Road, Near Haj House	18.945343	72.836809	400001	Mahārāshtra
4	Stadium Restaurant & Stores	Indian Restaurant	Veer Nariman Road	18.933173	72.826929	400023	Mahārāshtra
5	Nanumal Bhojraj Restaurant - Masjid Bunder	Indian Restaurant	Gaumukh Bhawan, Masjid Bunder West, Near Masji...	18.953200	72.837577	400009	Mahārāshtra
6	Lalit restaurant	Bar	Cawasji Patel road	18.935002	72.833791	400001	Mahārāshtra
7	Pine Restaurant	Indian Restaurant	Nawab Tank Bunder, Dockyard	18.964799	72.844015	400010	Mahārāshtra
8	Excellensea Restaurant	BBQ Joint	Mint Road	18.935266	72.837240	400001	Mahārāshtra
9	National Restaurant	Food	opp fort market	18.935112	72.836896	400001	Mahārāshtra
10	Fountain Sizzlers Restaurant	Steakhouse	57 Mahatma Gandhi Road	18.931366	72.831462	400 023	Mahārāshtra
11	Cafeteria	Café	Breach Candy Hospital	18.971833	72.804981	400026	Mahārāshtra
12	KES College of Commerce & Arts Cafeteria	College Cafeteria	Parekh lane	18.948587	72.829424	400067	Mahārāshtra
13	Cafeteria - P2 Building	Cafeteria	Next to Phoenix Building	18.992866	72.823965	400013	Mahārāshtra
14	Cafeteria, Lodha Excelus	Cafeteria	9th Floor, Lodha Excelus,	18.994027	72.834380	400011	Mahārāshtra
15	Jai Hind College	College Academic	A Road, Churchgate	18.934689	72.824822	400 020	Mahārāshtra

15. Visualization of dataframe of park,restaurant and cafeteria:



RESULT:

So now we can answer the questions asked above in Question section:

Answers:

1. The nearest Hotel are:

```
In [112]: # choose the hotel which has the same postalCode with the event space
          df_hotel = df_hotels[df_hotels.postalCode == '400001']
          df_hotel
```

Out[112]:

	name	categories	address	lat	lng	postalCode	state
1	Residency Hotel	Hotel	26, Corner of D.N. Road and Rustom Sidhwa Marg...	18.934978	72.833481	400001	Mahārāshtra
21	Benazeer Hotel	Hotel	16 Gunbow Street	18.935140	72.833948	400001	Mahārāshtra

2. The nearest park are:

In [119]:
```
# delete rows which its category is not Park
df_park = clean_park_dataframe3[clean_park_dataframe3.categories == 'Park']
df_park
```

Out[119]:

	name	categories	address	lat	lng	postalCode	state
0	Harish Mahindra Children's Park	Park	Breach Candy, Behind American Counsulate	18.914867	72.823688	400026	Mahārāshtra
4	Priyadarshini Park	Park	Nepean Sea Road	18.957515	72.799614	400006	Mahārāshtra

3. The nearest restaurant are:

In [125]:
```
# delete rows with none values
df_Restaurant = clean_Restaurant_dataframe2.dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)
df_Restaurant
```

Out[125]:

	name	categories	address	lat	lng	postalCode	state
0	Nanumal Bhojraj Restaurant - Fort	Indian Restaurant	Near GPO, Fort, VT	18.937974	72.837663	400001	Mahārāshtra
2	Iran Like Restaurant	Indian Restaurant	Palton Road, Near Haj House	18.945343	72.836809	400001	Mahārāshtra
5	Stadium Restaurant & Stores	Indian Restaurant	Veer Nariman Road	18.933173	72.826929	400023	Mahārāshtra
9	Nanumal Bhojraj Restaurant - Masjid Bunder	Indian Restaurant	Gaumukh Bhawan, Masjid Bunder West, Near Masji...	18.953200	72.837577	400009	Mahārāshtra
18	Lalit restaurant	Bar	Cawasji Patel road	18.935002	72.833791	400001	Mahārāshtra
21	Pine Restaurant	Indian Restaurant	Nawab Tank Bunder, Dockyard	18.964799	72.844015	400010	Mahārāshtra
22	Excellensea Restaurant	BBQ Joint	Mint Road	18.935266	72.837240	400001	Mahārāshtra
24	National Restaurant	Food	opp fort market	18.935112	72.836896	400001	Mahārāshtra
27	Fountain Sizzlers Restaurant	Steakhouse	57 Mahatma Gandhi Road	18.931366	72.831462	400 023	Mahārāshtra

In [ ]:
```
#Cafeteria
```

4. The nearest Cafeteria are:

In [131]:
```
# delete rows with none values
df_Cafeteria = clean_Cafeteria_dataframe2.dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)
df_Cafeteria
```

Out[131]:

	name	categories	address	lat	lng	postalCode	state
0	Cafeteria	Café	Breach Candy Hospital	18.971833	72.804981	400026	Mahārāshtra
5	KES College of Commerce & Arts Cafeteria	College Cafeteria	Parekh lane	18.948587	72.829424	400067	Mahārāshtra
6	Cafeteria - P2 Building	Cafeteria	Next to Phoenix Building	18.992866	72.823965	400013	Mahārāshtra
7	Cafeteria, Lodha Excelus	Cafeteria	9th Floor, Lodha Excelus,	18.994027	72.834380	400011	Mahārāshtra
11	Jai Hind College	College Academic Building	A Road, Churchgate	18.934689	72.824822	400 020	Mahārāshtra
13	Leo Burnett Cafeteria	Cafeteria	36-A Big Apple	18.998785	72.840376	400012	Mahārāshtra
17	Mahindra Towers Cafeteria	Cafeteria	Worli	19.006044	72.820630	400030	Mahārāshtra
18	Mahindra Holidays Cafeteria	Snack Place	Mahindra Holidays, Mahindra Towers, G.M. Bhosl...	19.005692	72.822309	400018	Mahārāshtra
19	burhani college	College Arts Building	Mazagoan	18.970446	72.835390	400010	Mahārāshtra
20	VIT Executive Cafeteria	College Cafeteria	Vidyalankar Institute Of Technology	19.021675	72.870364	400037	Mahārāshtra
25	Indigo Consulting	Building	The Big Apple	18.998891	72.840350	400012	Mahārāshtra
26	BloombergUTV	Food Court	Parijaat House, 1076, Dr. E Moses Road,	18.996397	72.819472	400018	Mahārāshtra

In [ ]:
```
#Shopping
```

5. The nearest shopping center are:

In [136]:
```
# delete unnecessary columns
clean_Shopping_dataframe2= clean_Shopping_dataframe.drop(['cc', 'city', 'country', 'distance', 'formattedAddress',\
                                    'labeledLatLngs', 'id'], axis=1)
clean_Shopping_dataframe2
```

Out[136]:

	name	categories	lat	lng	state
0	Ashoka Shopping Centre	Office	18.945182	72.832121	Mahārāshtra
1	Ashrafi Shopping Centre	Office	18.948688	72.836013	NaN
2	Bhangwadi Shopping Arcade	Office	18.946791	72.827999	NaN

CONCLUSION

There is always room for improvement and hence the above solution I have provided can also be improved for best results depending upon the data we have.