Finalproject

April 24, 2020

1 Introduction

The aim of the project is to identify venues in Mumbai, India based on their rating and average prices. In this notebook, we will identify various venues in the city of Mumbai, India, using Foursquare API and Zomato API, to help visitors select the restaurants that suit them the best. Whenever a user is visiting a city they start looking for places to visit during their stay. They primarily look for places based on the venue ratings across all venues and the average prices such that the locations fits in their budget. Here, we'll identify places that are fit for various individuals based on the information collected from the two APIs and Data Science. Once we have the plot with the venues, any company can launch an application using the same data and suggest users such information.

2 Data Collection from APIs

To begin with, we will take a look at Mumbai on the Map using the folium library. We will also fetch the data from two different APIs. Foursquare API: We will use the Foursquare API to fetch venues in Mumbai starting from the middle upto 44 Kilometers in each direction. Zomato API: The Zomato API provides information about various venues including the complete address, user ratings, price for two people, price range and a lot more.

[1]:
```
CHD_LATITUDE = 19.076
CHD_LONGITUDE = 72.877
print('The geograpical coordinates of Mumbai are {}, {}.'.format(CHD_LATITUDE,␣
 ↪CHD_LONGITUDE))
```

```
The geograpical coordinates of Mumbai are 19.076, 72.877.
```

[3]:
```
import folium

Mumbai_map = folium.Map(location = [CHD_LATITUDE, CHD_LONGITUDE], zoom_start =␣
 ↪13)
folium.Marker([CHD_LATITUDE, CHD_LONGITUDE]).add_to(Mumbai_map)

Mumbai_map
```

[3]:
```
<folium.folium.Map at 0x7faa940e19e8>
```

[4]:
```
RADIUS = 4000 # 4 Km
NO_OF_VENUES = 100

#set credintials
FOURSQUARE_CLIENT_ID = 'IPTYUZQHVW5OCDTT331BXA1SFQCJ3QCNQ2NVFZHQI5M4ZJLY' #␣
 ↪your Foursquare ID
FOURSQUARE_CLIENT_SECRET = '4ARF5SHATZIHJ2FJURJBFIUBZWYKROUZ4FP5XHAGRE4BCZJ1' #␣
 ↪your Foursquare Secret
VERSION = '20200424' # Foursquare API version
```

[5]:
```
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']
```

2.1 using foursquare api

[6]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as colors

from pandas.io.json import json_normalize
import requests

pd.set_option('display.max_rows', None)

offset = 0
total_venues = 0
foursquare_venues = pd.DataFrame(columns = ['name', 'categories', 'lat', 'lng'])

while (True):
    url = ('https://api.foursquare.com/v2/venues/explore?client_id={}'
           '&client_secret={}&v={}&ll={},{}&radius={}&limit={}&offset={}').
 ↪format(FOURSQUARE_CLIENT_ID,
                                                                    ␣
 ↪FOURSQUARE_CLIENT_SECRET,
                                                                     VERSION,
```

```
                                                                        ␣
↪CHD_LATITUDE,
                                                                        ␣
↪CHD_LONGITUDE,
                                                                         RADIUS,
                                                                        ␣
↪NO_OF_VENUES,
                                                                         offset)
    result = requests.get(url).json()
    venues_fetched = len(result['response']['groups'][0]['items'])
    total_venues = total_venues + venues_fetched
    print("Total {} venues fetched within a total radius of {} Km".
↪format(venues_fetched, RADIUS/1000))

    venues = result['response']['groups'][0]['items']
    venues = json_normalize(venues)

    # Filter the columns
    filtered_columns = ['venue.name', 'venue.categories', 'venue.location.lat',␣
↪'venue.location.lng']
    venues = venues.loc[:, filtered_columns]

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

    # Clean all column names
    venues.columns = [col.split(".")[-1] for col in venues.columns]
    foursquare_venues = pd.concat([foursquare_venues, venues], axis = 0, sort =␣
↪False)

    if (venues_fetched < 100):
        break
    else:
        offset = offset + 100

foursquare_venues = foursquare_venues.reset_index(drop = True)
print("\nTotal {} venues fetched".format(total_venues))
```

```
Total 100 venues fetched within a total radius of 4.0 Km
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-
packages/ipykernel_launcher.py:33: FutureWarning: pandas.io.json.json_normalize
is deprecated, use pandas.json_normalize instead
```

```
Total 100 venues fetched within a total radius of 4.0 Km
Total 31 venues fetched within a total radius of 4.0 Km

Total 231 venues fetched
```

2.1.1 using zomato Api

[7]:
```
headers = {'user-key': '57c7bf3391e8f795ff01b534595a1fc4'}
venues_information = []

for index, row in foursquare_venues.iterrows():
    print("Fetching data for venue: {}".format(index + 1))
    venue = []
    url = ('https://developers.zomato.com/api/v2.1/search?q={}' +
           '&start=0&count=1&lat={}&lon={}&sort=real_distance').
 ↪format(row['name'], row['lat'], row['lng'])
    result = requests.get(url, headers = headers).json()
    if (len(result['restaurants']) > 0):
        venue.append(result['restaurants'][0]['restaurant']['name'])
        venue.
 ↪append(result['restaurants'][0]['restaurant']['location']['latitude'])
        venue.
 ↪append(result['restaurants'][0]['restaurant']['location']['longitude'])
        venue.
 ↪append(result['restaurants'][0]['restaurant']['average_cost_for_two'])
        venue.append(result['restaurants'][0]['restaurant']['price_range'])
        venue.
 ↪append(result['restaurants'][0]['restaurant']['user_rating']['aggregate_rating'])
        venue.
 ↪append(result['restaurants'][0]['restaurant']['location']['address'])
        venues_information.append(venue)
    else:
        venues_information.append(np.zeros(6))

zomato_venues = pd.DataFrame(venues_information,
                                  columns = ['venue', 'latitude',
                                             'longitude', 'price_for_two',
                                             'price_range', 'rating',␣
 ↪'address'])
```

```
Fetching data for venue: 1
Fetching data for venue: 2
Fetching data for venue: 3
Fetching data for venue: 4
Fetching data for venue: 5
Fetching data for venue: 6
Fetching data for venue: 7
Fetching data for venue: 8
Fetching data for venue: 9
Fetching data for venue: 10
Fetching data for venue: 11
Fetching data for venue: 12
```

```
Fetching data for venue: 13
Fetching data for venue: 14
Fetching data for venue: 15
Fetching data for venue: 16
Fetching data for venue: 17
Fetching data for venue: 18
Fetching data for venue: 19
Fetching data for venue: 20
Fetching data for venue: 21
Fetching data for venue: 22
Fetching data for venue: 23
Fetching data for venue: 24
Fetching data for venue: 25
Fetching data for venue: 26
Fetching data for venue: 27
Fetching data for venue: 28
Fetching data for venue: 29
Fetching data for venue: 30
Fetching data for venue: 31
Fetching data for venue: 32
Fetching data for venue: 33
Fetching data for venue: 34
Fetching data for venue: 35
Fetching data for venue: 36
Fetching data for venue: 37
Fetching data for venue: 38
Fetching data for venue: 39
Fetching data for venue: 40
Fetching data for venue: 41
Fetching data for venue: 42
Fetching data for venue: 43
Fetching data for venue: 44
Fetching data for venue: 45
Fetching data for venue: 46
Fetching data for venue: 47
Fetching data for venue: 48
Fetching data for venue: 49
Fetching data for venue: 50
Fetching data for venue: 51
Fetching data for venue: 52
Fetching data for venue: 53
Fetching data for venue: 54
Fetching data for venue: 55
Fetching data for venue: 56
Fetching data for venue: 57
Fetching data for venue: 58
Fetching data for venue: 59
Fetching data for venue: 60
```

```
Fetching data for venue: 61
Fetching data for venue: 62
Fetching data for venue: 63
Fetching data for venue: 64
Fetching data for venue: 65
Fetching data for venue: 66
Fetching data for venue: 67
Fetching data for venue: 68
Fetching data for venue: 69
Fetching data for venue: 70
Fetching data for venue: 71
Fetching data for venue: 72
Fetching data for venue: 73
Fetching data for venue: 74
Fetching data for venue: 75
Fetching data for venue: 76
Fetching data for venue: 77
Fetching data for venue: 78
Fetching data for venue: 79
Fetching data for venue: 80
Fetching data for venue: 81
Fetching data for venue: 82
Fetching data for venue: 83
Fetching data for venue: 84
Fetching data for venue: 85
Fetching data for venue: 86
Fetching data for venue: 87
Fetching data for venue: 88
Fetching data for venue: 89
Fetching data for venue: 90
Fetching data for venue: 91
Fetching data for venue: 92
Fetching data for venue: 93
Fetching data for venue: 94
Fetching data for venue: 95
Fetching data for venue: 96
Fetching data for venue: 97
Fetching data for venue: 98
Fetching data for venue: 99
Fetching data for venue: 100
Fetching data for venue: 101
Fetching data for venue: 102
Fetching data for venue: 103
Fetching data for venue: 104
Fetching data for venue: 105
Fetching data for venue: 106
Fetching data for venue: 107
Fetching data for venue: 108
```

```
Fetching data for venue: 109
Fetching data for venue: 110
Fetching data for venue: 111
Fetching data for venue: 112
Fetching data for venue: 113
Fetching data for venue: 114
Fetching data for venue: 115
Fetching data for venue: 116
Fetching data for venue: 117
Fetching data for venue: 118
Fetching data for venue: 119
Fetching data for venue: 120
Fetching data for venue: 121
Fetching data for venue: 122
Fetching data for venue: 123
Fetching data for venue: 124
Fetching data for venue: 125
Fetching data for venue: 126
Fetching data for venue: 127
Fetching data for venue: 128
Fetching data for venue: 129
Fetching data for venue: 130
Fetching data for venue: 131
Fetching data for venue: 132
Fetching data for venue: 133
Fetching data for venue: 134
Fetching data for venue: 135
Fetching data for venue: 136
Fetching data for venue: 137
Fetching data for venue: 138
Fetching data for venue: 139
Fetching data for venue: 140
Fetching data for venue: 141
Fetching data for venue: 142
Fetching data for venue: 143
Fetching data for venue: 144
Fetching data for venue: 145
Fetching data for venue: 146
Fetching data for venue: 147
Fetching data for venue: 148
Fetching data for venue: 149
Fetching data for venue: 150
Fetching data for venue: 151
Fetching data for venue: 152
Fetching data for venue: 153
Fetching data for venue: 154
Fetching data for venue: 155
Fetching data for venue: 156
```

```
Fetching data for venue: 157
Fetching data for venue: 158
Fetching data for venue: 159
Fetching data for venue: 160
Fetching data for venue: 161
Fetching data for venue: 162
Fetching data for venue: 163
Fetching data for venue: 164
Fetching data for venue: 165
Fetching data for venue: 166
Fetching data for venue: 167
Fetching data for venue: 168
Fetching data for venue: 169
Fetching data for venue: 170
Fetching data for venue: 171
Fetching data for venue: 172
Fetching data for venue: 173
Fetching data for venue: 174
Fetching data for venue: 175
Fetching data for venue: 176
Fetching data for venue: 177
Fetching data for venue: 178
Fetching data for venue: 179
Fetching data for venue: 180
Fetching data for venue: 181
Fetching data for venue: 182
Fetching data for venue: 183
Fetching data for venue: 184
Fetching data for venue: 185
Fetching data for venue: 186
Fetching data for venue: 187
Fetching data for venue: 188
Fetching data for venue: 189
Fetching data for venue: 190
Fetching data for venue: 191
Fetching data for venue: 192
Fetching data for venue: 193
Fetching data for venue: 194
Fetching data for venue: 195
Fetching data for venue: 196
Fetching data for venue: 197
Fetching data for venue: 198
Fetching data for venue: 199
Fetching data for venue: 200
Fetching data for venue: 201
Fetching data for venue: 202
Fetching data for venue: 203
Fetching data for venue: 204
```

```
Fetching data for venue: 205
Fetching data for venue: 206
Fetching data for venue: 207
Fetching data for venue: 208
Fetching data for venue: 209
Fetching data for venue: 210
Fetching data for venue: 211
Fetching data for venue: 212
Fetching data for venue: 213
Fetching data for venue: 214
Fetching data for venue: 215
Fetching data for venue: 216
Fetching data for venue: 217
Fetching data for venue: 218
Fetching data for venue: 219
Fetching data for venue: 220
Fetching data for venue: 221
Fetching data for venue: 222
Fetching data for venue: 223
Fetching data for venue: 224
Fetching data for venue: 225
Fetching data for venue: 226
Fetching data for venue: 227
Fetching data for venue: 228
Fetching data for venue: 229
Fetching data for venue: 230
Fetching data for venue: 231
```

[8]:
```
Mumbai_map = folium.Map(location = [CHD_LATITUDE, CHD_LONGITUDE], zoom_start =␣
↪13)
for name, latitude, longitude in zip(foursquare_venues['name'],␣
↪foursquare_venues['lat'], foursquare_venues['lng']):
    label = '{}'.format(name)
    label = folium.Popup(label, parse_html = True)
    folium.CircleMarker(
        [latitude, longitude],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(Mumbai_map)


Mumbai_map
```

[8]:
```
<folium.folium.Map at 0x7faa6a2e5780>
```

[9]:
```
zomato_venu2=zomato_venues.astype({'latitude': 'float64','longitude':'float64'})
```

[10]:
```
Mumbai_map = folium.Map(location = [CHD_LATITUDE, CHD_LONGITUDE], zoom_start =␣
↪13)

for venue, address, latitude, longitude in zip(zomato_venu2['venue'],␣
↪zomato_venu2['address'],
                                               zomato_venu2['latitude'],␣
↪zomato_venu2['longitude']):
    label = '{}, {}'.format(name, address)
    label = folium.Popup(label, parse_html = True)
    folium.CircleMarker(
        [latitude, longitude],
        radius = 5,
        popup = label,
        color = 'red',
        fill = True,
        fill_color = '#cc3535',
        fill_opacity = 0.7,
        parse_html = False).add_to(Mumbai_map)

Mumbai_map
```

[10]:
```
<folium.folium.Map at 0x7faa63baf198>
```

[11]:
```
foursquare_venues['lat'] = foursquare_venues['lat'].apply(lambda lat:␣
↪round(float(lat), 4))
foursquare_venues['lng'] = foursquare_venues['lng'].apply(lambda lng:␣
↪round(float(lng), 4))
zomato_venues['latitude'] = zomato_venues['latitude'].apply(lambda lat:␣
↪round(float(lat), 4))
zomato_venues['longitude'] = zomato_venues['longitude'].apply(lambda lng:␣
↪round(float(lng), 4))
```

[12]:
```
dataset = pd.concat([foursquare_venues, zomato_venues], axis = 1)
dataset['lat_diff'] = dataset['latitude'] - dataset['lat']
dataset['lng_diff'] = dataset['longitude'] - dataset['lng']
```

I'll drop the venues which have 0.0 rating as it means it's not been rated yet.

[13]:
```
selected_venues = dataset[(abs(dataset['lat_diff']) <= 0.0004) &␣
↪(abs(dataset['lng_diff']) <= 0.0004)].reset_index(drop = True)
selected_venues
```

[13]:

	name	categories
0	Sofitel Mumbai BKC	Hotel
1	Trident	Hotel
2	Masala Library	Indian Restaurant
3	Natural's Ice Cream	Ice Cream Shop
4	Smoke House Deli (BKC)	Deli / Bodega
5	Theobrama	Dessert Shop
6	Delhi Zaika	Indian Restaurant
7	SodaBottleOpenerWala	Irani Cafe
8	Starbucks	Coffee Shop
9	Naaz Hotel	Indian Restaurant
10	The Irish House Phoenix Marketcity	Beer Garden
11	IVY Restaurant & Banquets	Restaurant
12	Maya	Indian Restaurant
13	Darshan	Chinese Restaurant
14	JW Marriott Mumbai Sahar	Hotel
15	Phoenix Marketcity	Shopping Mall
16	212 All Day	Café
17	Cream Centre	Restaurant
18	Chembur Post Office Wada Pav	Snack Place
19	Sadguru Juice and Snacks	Indian Restaurant
20	the chocolate room	Dessert Shop
21	The Royal Orchid	Asian Restaurant
22	Wok Express BKC	Noodle House
23	Vikrant Circle	Sculpture Garden
24	Natural Ice Cream	Dessert Shop
25	L5 Restaurant & Lounge	Lounge
26	Mani's Lunch Home's Satvika	Indian Restaurant
27	Lamba Da Dhaba	Punjabi Restaurant
28	Kandoi haribhai damodar mithaiwala	Dessert Shop
29	MIG Cricket Club	Sports Club
30	GD's Chinese Corner	Chinese Restaurant
31	Sadguru Veg. Diet	Restaurant
32	Chakra Restaurant & Bar	Indian Restaurant
33	Starbucks Coffee : A TATA Alliance	Coffee Shop
34	Ambedkar Udyan	Garden
35	British Brewing Company	Brewery
36	Le Café	Café
37	Meghdoot	Indian Restaurant
38	Udipi Hotel	Fast Food Restaurant
39	The Lalit	Hotel
40	Sai Swad Dosa	Fast Food Restaurant
41	Blossom Juice Center	Juice Bar
42	Amoeba	Bowling Alley
43	Pizza Express	Italian Restaurant
44	Opa! Bar & Cafe	Lounge
45	Gurukrupa	Indian Restaurant

46	Rainforest Resto-Bar	Restaurant
47	Burger King	Fast Food Restaurant
48	Harry's	Bar
49	Sai Sagar Ice Cream Falooda	Ice Cream Shop
50	Achija	Vegetarian / Vegan Restaurant
51	Vienna Bakery	Bakery
52	Curry Twist	Vegetarian / Vegan Restaurant
53	Wild Orchid	Lounge
54	New Achija	Restaurant
55	Geeta Bhavan	Indian Restaurant
56	Ranjit Daa Dhabha	Indian Restaurant
57	Noodle Bar	Chinese Restaurant
58	Pop Tate's	American Restaurant
59	Vig Refreshment	Fast Food Restaurant
60	Spirit Kitchen & Bar	Indian Restaurant
61	The Beer Café	Beer Garden
62	Sadguru	Vegetarian / Vegan Restaurant
63	Mirage Hotel Mumbai	Hotel
64	Hit & Run	Falafel Restaurant
65	Wok Hei	Chinese Restaurant
66	MIG Club Cloud 9 Bar	Bar
67	Vaishali Garden	Seafood Restaurant
68	Mahesh Lunch Home	Seafood Restaurant
69	California Pizza Kitchen	Pizza Place
70	Odeon	Snack Place
71	Barista	Café
72	Terminal f	Food Court
73	Urban Tadka	Indian Restaurant
74	Asia Kitchen & Bar	Indian Chinese Restaurant
75	Persian Darbar	Indian Restaurant
76	Sadichha	Indian Restaurant
77	Grand Central	Indian Restaurant
78	Hotel Pooja	Vegetarian / Vegan Restaurant
79	Bawa International Hotel Mumbai	Hotel
80	South of Vindhyas	Indian Restaurant
81	Popular Hotel	Indian Restaurant
82	Domino's Pizza	Pizza Place
83	Cafe Rumours	Fast Food Restaurant
84	Shree Ganesh Sagar	Indian Restaurant
85	Blend n Brew	Bar
86	cafe coffee day	Café
87	Grilled House Kabab Corner	Indian Restaurant
88	Pizza Etc	Pizza Place
89	Highway Gomantak	Seafood Restaurant
90	Cafe Udipi	Indian Restaurant
91	Tastings	Ice Cream Shop
92	New Cafe Marol	Indian Restaurant

	lat	lng	venue	latitude	\
0	19.0674	72.8690	Tuskers - Sofitel	19.0673	
1	19.0668	72.8675	O22 - Trident	19.0672	
2	19.0689	72.8697	Masala Library	19.0690	
3	19.0776	72.8630	Natural Ice Cream	19.0776	
4	19.0688	72.8697	Smoke House Deli	19.0688	
5	19.0635	72.8617	Theobroma	19.0637	
6	19.0771	72.8783	Delhi Zaika	19.0770	
7	19.0636	72.8624	SodaBottleOpenerWala	19.0638	
8	19.0866	72.8898	Starbucks Coffee	19.0865	
9	19.0848	72.8892	Naaz Hotel	19.0846	
10	19.0864	72.8888	Le Paan	19.0863	
11	19.0697	72.9005	Ivy Restaurant and Banquets	19.0696	
12	19.0669	72.8670	BOX8- Desi Meals	19.0667	
13	19.0827	72.8853	Hotel Darshan	19.0828	
14	19.1025	72.8782	Love & Latte	19.1022	
15	19.0859	72.8887	Secondo - The Continental Chicken	19.0862	
16	19.0863	72.8891	Shizusan	19.0864	
17	19.0866	72.8898	Punjabi Kulfi	19.0865	
18	19.0569	72.8981	Oven Hot The Bake Shop	19.0569	
19	19.0622	72.9005	Sadguru Veg Diet	19.0622	
20	19.0752	72.9052	Chocday	19.0752	
21	19.0617	72.9016	Orchids	19.0616	
22	19.0654	72.8618	McCafe by McDonald's	19.0653	
23	19.0750	72.9052	Mi Favorito	19.0750	
24	19.0526	72.8998	Tastings	19.0527	
25	19.0691	72.9004	Flaming 5 Fine Dine & Lounge	19.0690	
26	19.0752	72.9047	The Food Blog	19.0751	
27	19.0526	72.8994	Lamba Restaurant & Bar	19.0525	
28	19.0822	72.9046	Purshottam Kandoi Haribhai Damodar Mithaiwala	19.0825	
29	19.0575	72.8477	Maratha Darbar	19.0576	
30	19.0713	72.8972	GD's Chinese Corner	19.0714	
31	19.0611	72.8997	Hotel Saroj & Sweet Mart	19.0615	
32	19.1035	72.8862	Chakra	19.1036	
33	19.0866	72.8898	Starbucks Coffee	19.0865	
34	19.0596	72.9005	Vijay Bhelpuri	19.0592	
35	19.1075	72.8835	British Brewing Company	19.1077	
36	19.0618	72.8995	Le Café	19.0617	
37	19.0872	72.9086	Meghdoot	19.0871	
38	19.0801	72.9041	Udipi Refreshment	19.0801	
39	19.1054	72.8757	R K Food Plaza	19.1053	
40	19.0804	72.9043	The London Shakes	19.0804	
41	19.0752	72.9052	Blossom- The Fusion Spot	19.0751	
42	19.0869	72.8891	Amoeba Sports Bar	19.0870	
43	19.1074	72.8835	British Brewing Company	19.1077	
44	19.1025	72.8886	The Stables Beer Garden	19.1024	

45	19.0798	72.9061	Guru Krupa	19.0798
46	19.0865	72.8891	BAR BAR	19.0865
47	19.1076	72.8828	Burger King	19.1077
48	19.0866	72.8898	McDonald's	19.0867
49	19.0456	72.8941	Sai Sagar	19.0456
50	19.0800	72.9061	Achija	19.0801
51	19.0793	72.8521	Vienna Bakery	19.0795
52	19.1033	72.8862	Curry Twist	19.1032
53	19.0617	72.9017	Wild Orchids	19.0616
54	19.0745	72.9052	Swamy - South Indian Food Express	19.0749
55	19.0607	72.9005	Geeta Bhavan	19.0607
56	19.0452	72.8948	Ranjeet Daa Dhaba	19.0453
57	19.0866	72.8899	Balaji Pure Veg	19.0866
58	19.1039	72.8867	KFC	19.1036
59	19.0457	72.8954	Vig	19.0457
60	19.1068	72.8812	Spirit	19.1067
61	19.0996	72.8753	The Beer Cafe	19.0999
62	19.0616	72.8995	Sadguru Veg Diet	19.0616
63	19.1089	72.8773	Magnolia - Hotel Mirage	19.1091
64	19.1078	72.8633	Hit & Run	19.1077
65	19.0526	72.8997	Chinese Wok- Wok Express	19.0524
66	19.0574	72.8477	Maratha Darbar	19.0576
67	19.0618	72.9012	Vaishali Garden Restaurant	19.0619
68	19.1069	72.8845	Mahesh Lunch Home	19.1070
69	19.0550	72.8506	Pa Pa Ya	19.0550
70	19.0791	72.9094	Hotel Prakash	19.0790
71	19.0521	72.9008	Quenchers by Barista	19.0521
72	19.0927	72.8539	Kailash Parbat	19.0926
73	19.1037	72.8866	Urban Tadka	19.1036
74	19.1047	72.8864	Asia Kitchen by Mainland China	19.1046
75	19.1096	72.8758	Persian Darbar	19.1097
76	19.0581	72.8477	Sadichha	19.0581
77	19.0614	72.9006	Hotel Grand Central	19.0614
78	19.0812	72.9043	Hotel Pooja	19.0813
79	19.0965	72.8534	Spice It - IBIS	19.0966
80	19.0972	72.8549	Boulevard - The Orchid	19.0974
81	19.0822	72.9097	Hotel Popular	19.0822
82	19.0520	72.9010	Domino's Pizza	19.0521
83	19.0775	72.8991	Baithack Taste Of Kulhad	19.0777
84	19.0510	72.9043	Shree Ganesh Sagar	19.0509
85	19.0607	72.9000	Blend N Brew	19.0608
86	19.0615	72.9004	Cafe Udipi	19.0616
87	19.0590	72.9010	Grilled House	19.0592
88	19.0825	72.9085	Supriya Fast Food	19.0825
89	19.0568	72.8470	Highway Gomantak	19.0568
90	19.0615	72.9007	Cafe Udipi	19.0616
91	19.0526	72.8998	Tastings	19.0527

92 19.1116 72.8787   Fatty Patty   19.1117

	longitude	price_for_two	price_range	rating	\
0	72.8692	3300.0	4.0	4.0	
1	72.8675	4000.0	4.0	4.0	
2	72.8696	5000.0	4.0	4.4	
3	72.8628	300.0	1.0	4.1	
4	72.8695	2500.0	4.0	4.3	
5	72.8621	900.0	2.0	4.2	
6	72.8784	700.0	2.0	3.8	
7	72.8620	1500.0	3.0	4.1	
8	72.8898	700.0	2.0	4.1	
9	72.8893	700.0	2.0	4.1	
10	72.8888	200.0	1.0	0	
11	72.9004	1600.0	3.0	4.1	
12	72.8667	400.0	1.0	4.2	
13	72.8851	250.0	1.0	3.5	
14	72.8782	900.0	2.0	3.9	
15	72.8889	500.0	2.0	0	
16	72.8892	1800.0	3.0	4.0	
17	72.8898	150.0	1.0	3.5	
18	72.8981	200.0	1.0	3.7	
19	72.9005	700.0	2.0	3.8	
20	72.9049	350.0	1.0	3.7	
21	72.9016	1200.0	3.0	4.0	
22	72.8615	500.0	2.0	3.9	
23	72.9052	500.0	2.0	3.8	
24	72.8998	550.0	2.0	3.5	
25	72.9004	1400.0	3.0	3.6	
26	72.9047	1000.0	3.0	3.5	
27	72.8994	750.0	2.0	3.7	
28	72.9046	200.0	1.0	4.2	
29	72.8478	600.0	2.0	3.4	
30	72.8972	700.0	2.0	3.8	
31	72.8997	500.0	2.0	3.8	
32	72.8863	1100.0	3.0	3.9	
33	72.8898	700.0	2.0	4.1	
34	72.9005	100.0	1.0	3.3	
35	72.8834	2000.0	4.0	4.0	
36	72.8997	1500.0	3.0	4.2	
37	72.9086	1000.0	3.0	3.8	
38	72.9042	500.0	2.0	3.6	
39	72.8759	300.0	1.0	0	
40	72.9043	600.0	2.0	3.8	
41	72.9051	800.0	2.0	3.9	
42	72.8893	1100.0	3.0	3.8	
43	72.8834	2000.0	4.0	4.0	

44	72.8882	1000.0	3.0	4.2
45	72.9061	600.0	2.0	3.8
46	72.8893	1500.0	3.0	3.8
47	72.8825	500.0	2.0	4.0
48	72.8900	400.0	1.0	4.1
49	72.8941	250.0	1.0	3.9
50	72.9057	800.0	2.0	4.1
51	72.8522	250.0	1.0	3.9
52	72.8863	1600.0	3.0	3.9
53	72.9016	2000.0	4.0	3.8
54	72.9050	200.0	1.0	0
55	72.9004	450.0	1.0	3.8
56	72.8947	900.0	2.0	3.8
57	72.8900	700.0	2.0	3.2
58	72.8865	450.0	1.0	4.1
59	72.8953	400.0	1.0	4.0
60	72.8813	850.0	2.0	3.8
61	72.8751	2400.0	4.0	4.0
62	72.8996	700.0	2.0	3.9
63	72.8771	1500.0	3.0	3.7
64	72.8633	700.0	2.0	4.1
65	72.8995	800.0	2.0	3.6
66	72.8478	600.0	2.0	3.4
67	72.9010	1000.0	3.0	3.8
68	72.8845	1700.0	3.0	4.1
69	72.8506	2600.0	4.0	4.2
70	72.9093	500.0	2.0	3.4
71	72.9007	300.0	1.0	0
72	72.8538	1000.0	3.0	2.5
73	72.8865	1700.0	3.0	4.0
74	72.8866	1800.0	3.0	4.2
75	72.8758	1500.0	3.0	4.1
76	72.8477	500.0	2.0	3.8
77	72.9005	1200.0	3.0	3.8
78	72.9042	750.0	2.0	3.5
79	72.8533	1700.0	3.0	3.4
80	72.8548	2600.0	4.0	4.7
81	72.9098	450.0	1.0	3.8
82	72.9014	400.0	1.0	3.8
83	72.8992	500.0	2.0	0
84	72.9040	550.0	2.0	3.6
85	72.9000	1700.0	3.0	3.8
86	72.9007	500.0	2.0	3.7
87	72.9011	550.0	2.0	3.7
88	72.9086	550.0	2.0	3.6
89	72.8470	600.0	2.0	3.7
90	72.9007	500.0	2.0	3.7

```
91    72.8998          400.0          1.0      0
92    72.8787          500.0          2.0    3.6
```

	address	lat_diff	lng_diff
0	Sofitel Hotel, C 57, Bandra Kurla Complex, Mumbai	-0.0001	0.0002
1	Trident Hotel, C 56, G Block, Bandra Kurla Com…	0.0004	0.0000
2	Ground Floor, First International Financial Ce…	0.0001	-0.0001
3	3, Gokul Harmony, Kalina Market, Sunder Nagar,…	0.0000	-0.0002
4	3A, Ground Floor, 1st International Financial …	0.0000	-0.0002
5	The Capital, Ground Floor, Bandra Kurla Comple…	0.0002	0.0004
6	5/6, Near Kamran Kalpana Theater, LBS Marg, Ku…	-0.0001	0.0001
7	Ground Floor, The Capital Building, G Block, B…	0.0002	-0.0004
8	First Floor, Phoenix Market City, LBS Marg, Ku…	-0.0001	0.0000
9	Kantharia Mahal, LBS Road, Opposite Phoenix Ma…	-0.0002	0.0001
10	Phoenix Mall, Lbs Road, Kamani, Kurla, Mumbai	-0.0001	0.0000
11	Above Shopper's Stop, Amar Mahal, Chembur, Mumbai	-0.0001	-0.0001
12	1st Floor, Near Trident Hotel, Bandra Kurla Co…	-0.0002	-0.0003
13	Opposite Kohinoor City Gate, Kurla West, Near …	0.0001	-0.0002
14	The ORB, Retail Arcade, Unit G2 & G3, Ground F…	-0.0003	0.0000
15	Unit S-32, 2nd Floor, Phoenix Market City, LBS…	0.0003	0.0002
16	F-97, Phoenix Marketcity, Near Kurla Fire Stat…	0.0001	0.0001
17	Phoenix Marketcity, LBS Road, Kurla, Mumbai	-0.0001	0.0000
18	4 & 5, Ganesh Kutir Building, Near Laxmi Store…	0.0000	0.0000
19	Janata Market, Near Chembur Station, Chembur, …	0.0000	0.0000
20	Unit 15, Vikrant Building, Vikrant Circle, Gha…	0.0000	-0.0003
21	Hotel Royal Orchid, NG Acharya Marg, Chembur, …	-0.0001	0.0000
22	Shop 1, Urban Plaza, Plot C5, GN Block, Bandra…	-0.0001	-0.0003
23	Shop 11, Vikrant Building, Tilak Road, Jhulela…	0.0000	0.0000
24	Opposite Diamond Garden, Central Avenue Road, …	0.0001	0.0000
25	Level 5, The Destination, Near Shopper's Stop,…	-0.0001	0.0000
26	Shop 24, Vikrant Building, Vikrant Circle, Gha…	-0.0001	0.0000
27	15/16, Chhadva Apartments, Sion Trombay Road, …	-0.0001	0.0000
28	Gayatri Dham, M.G. Road, Ghatkopar East, Mumbai	0.0003	0.0000
29	Nishigandha CHS, Opposite MIG Club, Gandhi Nag…	0.0001	0.0001
30	3, Building 106, Kambodi CHS, Tilak Nagar, Che…	0.0001	0.0000
31	10th Road, Near Chembur Post Office, Chembur, …	0.0004	0.0000
32	Sakinaka Junction, Andheri Kurla Road, Sakinak…	0.0001	0.0001
33	First Floor, Phoenix Market City, LBS Marg, Ku…	-0.0001	0.0000
34	Central Avenue Road, Near Ambedkar Garden, Che…	-0.0004	0.0000
35	Ground Floor, Times Square Tech Park, Andheri …	0.0002	-0.0001
36	1st Road, Opposite B.M.C Office, Chembur, Mumbai	-0.0001	0.0002
37	Guruprasad Building, JV Road, Beside Neo Welco…	-0.0001	0.0000
38	MG Road, Rajawadi, Ghatkopar East, Mumbai	0.0000	0.0001
39	Marol Naka, International Airport Road, Chabhi…	-0.0001	0.0002
40	Shop 4, Jalaram CHS, Khau Galli, Ghatkopar Eas…	0.0000	0.0000
41	Vikrant Circle, Tilak Road, Ghatkopar East, Mu…	-0.0001	-0.0001
42	F 33, 1st Floor, Phoenix Market City, LBS Marg…	0.0001	0.0002

43	Ground Floor, Times Square Tech Park, Andheri …	0.0003	-0.0001
44	The Peninsula Redpine, Near Airport Road Metro…	-0.0001	-0.0004
45	1, Kondeshwar Bhuvan, Tilak Road, Ghatkopar Ea…	0.0000	0.0000
46	1st Floor, Phoenix Marketcity, Lal Bahadur Sha…	0.0000	0.0002
47	Next to Times Square Building, Marol, Mumbai	0.0001	-0.0003
48	2nd Floor, Phoenix Marketcity Mall, L B S Marg…	0.0001	0.0002
49	20, Dr. C.G.Road, Chembur Colony, Chembur, Mumbai	0.0000	0.0000
50	13, Vasant Nivas, Tilak Road Naka, Ghatkopar E…	0.0001	-0.0004
51	125, Vakola Pipeline, Behind St. Anthonys Chur…	0.0002	0.0001
52	Sagar Pallazio, Andheri Kurla Road, Opposite C…	-0.0001	0.0001
53	Hotel Royal Orchid, NG Acharya Marg, Chembur, …	-0.0001	-0.0001
54	Shop 17, Vikrant Building, Next To New York Bu…	0.0004	-0.0002
55	Central Avenue Road, Chembur, Mumbai	0.0000	-0.0001
56	Dr CG Mark, Chembur-E Mumbai, Chembur, Mumbai	0.0001	-0.0001
57	Phoenix Market City, LBS Road, Kurla, Mumbai	0.0000	0.0001
58	Sagar Pallazio, Andheri-Kurla Road, Safed Pul,…	-0.0003	-0.0002
59	CG Road, Chembur Camp, Chembur East, Chembur, …	0.0000	-0.0001
60	Near Mittal Estate Hindsaurastra, Wellington B…	-0.0001	0.0001
61	MLCP P10, Landside, Departure Terminal 2, Inte…	0.0003	-0.0002
62	Jewel Of Chembur Building, BMC Office, Chembur…	0.0000	0.0001
63	Hotel Mirage, International Approach Road, Mar…	0.0002	-0.0002
64	3, Tarun Bharat Road, Near Akal Society, Chaka…	-0.0001	0.0000
65	K-Star Mall, Unit 10, 3rd Road, V.N Purav Marg…	-0.0002	-0.0002
66	Nishigandha CHS, Opposite MIG Club, Gandhi Nag…	0.0002	0.0001
67	1st Floor, Navketan Building, Govandi Road, Op…	0.0001	-0.0002
68	Andheri Kurla Road, Andheri East, Sakinaka, …	0.0001	0.0000
69	G-2, Ground Floor, North Avenue, Maker Maxity,…	0.0000	0.0000
70	R Odeon Mall, Vallabh Baug Lane, Ghatkopar Eas…	-0.0001	-0.0001
71	Shop 1-2-3, Garden Apartments, Diamond Garden …	0.0000	-0.0001
72	Terminal 1B, Ground Floor, Arrival Plaza, Sant…	-0.0001	-0.0001
73	1st Floor, Sagar Pallazio, Sakinaka Junction, …	-0.0001	-0.0001
74	Ground Floor, Neelkanth Udyog Bavan, Near Saki…	-0.0001	0.0002
75	Shop 101 & 102, Noori Baug, Andheri Kurla Road…	0.0001	0.0000
76	Opposite MIG Cricket Club, Gandhi Nagar, BKC, …	0.0000	0.0000
77	70 C, Shilp Building, Central Avenue Road, Che…	0.0000	-0.0001
78	MG Road, Ghatkopar East, Mumbai	0.0001	-0.0001
79	IBIS Mumbai Airport, Junction Of Nehru Road & …	0.0001	-0.0001
80	The Orchid, 70 C, Nehru Road, Vile Parle East,…	0.0002	-0.0001
81	Hing Wala Lane, Ghatkopar East, Mumbai	0.0000	0.0001
82	16 & 17, 3rd Floor, K Star Mall, Sion Trombay …	0.0001	0.0004
83	Shop 3, Surya House, Station Road, Vidhya Viha…	0.0002	0.0001
84	Atur Park, Sion Trombay Road, Chembur, Mumbai	-0.0001	-0.0003
85	Hotel Plaza, 70, Central Avenue Road, Chembur …	0.0001	0.0000
86	72, Hira Baug, Opposite Canara Bank, Chembur, …	0.0001	0.0003
87	10, Eric House, 16th Road, Chembur, Mumbai	0.0002	0.0001
88	D D Rashtryashala School Building, Hingwala La…	0.0000	0.0001
89	44/2179, Gandhi Nagar, Service Road, Bandra Ea…	0.0000	0.0000

```
90  72, Hira Baug, Opposite Canara Bank, Chembur, …    0.0001    0.0000
91  Shop 1, Suswagtam Building, Opposite Diamond G…    0.0001    0.0000
92  Shop 10, Opposite Silver Inn Hotel, Marol Maro…    0.0001    0.0000
```

Category 1: There are venues that have specific restaurants/cafes inside them as provided by Zomato API (Pizza Hut in Elante Mall).

Category 2: Two locations are so close by that they have practically same latitude and longitude values (The Pizza Kitchen and Zara).

Category 3: Some have been replaced with new restaurants (Uderdoggs has now been replaced by The Brew Estate).

The venues which belong to category 1 and category 3 are alright to keep, the venues that fall in category 2 should be removed.

[14]:
```
selected_venues = selected_venues.
↪drop([12,14,15,16,17,22,24,26,34,43,44,48,58,66,70,72,92]).reset_index(drop␣
↪= True)
```

I'll now select the venue name from Zomato API. I'll also get the average price per person by dividing the column price_for_two by 2 and removing this column from the dataset along with other unnecessary columns.

[15]:
```
selected_venues['average_price'] = selected_venues['price_for_two']/2
selected_venues = selected_venues.drop(columns = ['name', 'lat', 'lng',␣
↪'lat_diff', 'lng_diff', 'price_for_two'])
```

[16]:
```
selected_venues.head(5)
```

[16]:

	categories	venue	latitude	longitude	price_range	rating	address	average_price
0	Hotel	Tuskers - Sofitel	19.0673	72.8692	4.0	4.0	Sofitel Hotel, C 57, Bandra Kurla Complex, Mumbai	1650.0
1	Hotel	022 - Trident	19.0672	72.8675	4.0	4.0	Trident Hotel, C 56, G Block, Bandra Kurla Com…	2000.0
2	Indian Restaurant	Masala Library	19.0690	72.8696	4.0	4.4	Ground Floor, First International Financial Ce…	2500.0
3	Ice Cream Shop	Natural Ice Cream	19.0776	72.8628	1.0	4.1	3, Gokul Harmony, Kalina Market, Sunder Nagar,…	150.0
4	Deli / Bodega	Smoke House Deli	19.0688	72.8695	4.0	4.3	3A, Ground Floor, 1st International Financial …	1250.0

I'll drop the venues which have 0.0 rating as it means it's not been rated yet.

[17]:
```
selected_venues = selected_venues[selected_venues['rating'] != 0.0]
print("Total venues available: {}".format(selected_venues.shape[0]))
```

```
Total venues available: 70
```

2.1.2 methodology

This project aims at identifying the venues in mumbai based on their rating and average costs. This would enable any visitor to identify the venues he/she wants to visit based on their rating and cost preference.

As a first step, we retrieved the data from two APIs (Foursquare and Zomato). We extract venue information from the center of Mumbai, upto a distance of 4 Km. The latitude and longitude values are then used to fetch venue rating and price from Zomato.

Secondly, we then explored the data retrieved from the two APIs on the map and identified the top category types. The data from the two sources is carefully combined based on the name, latitude and longitude values from the two sources. The final dataset would include the rating and price values for each venue.

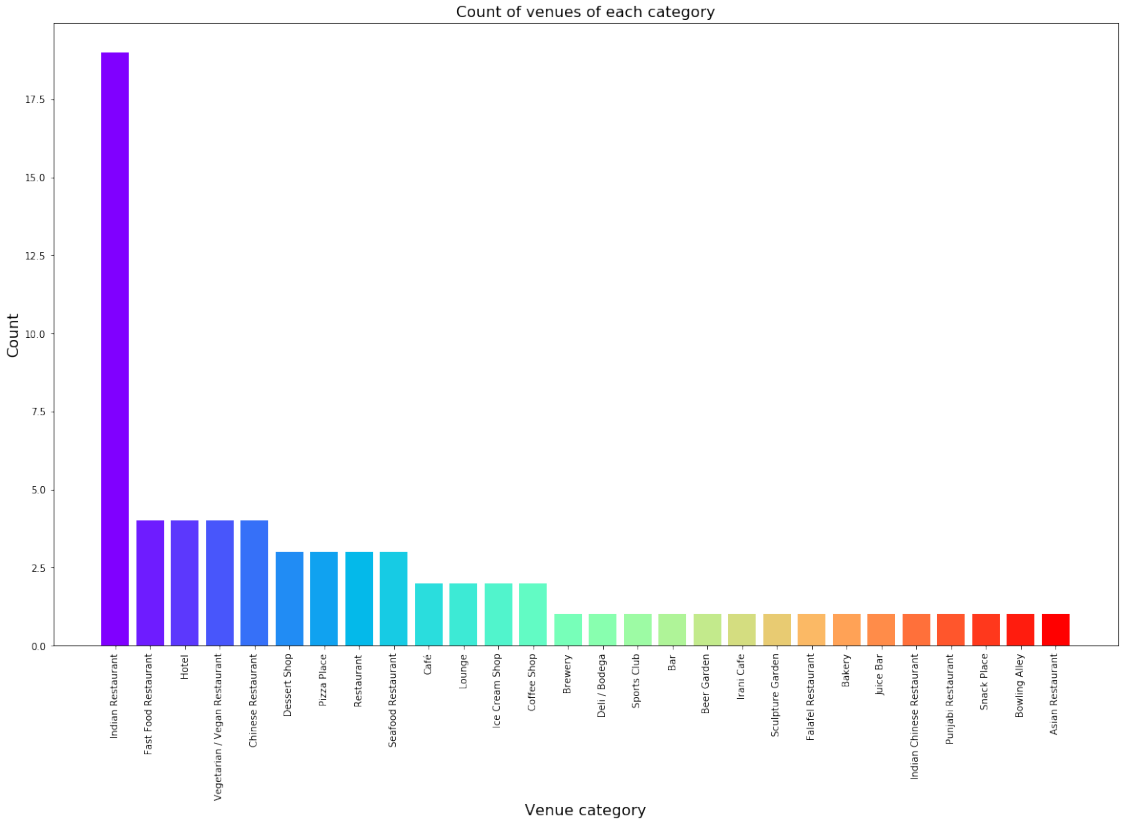
Next, we'll analyse the data that we created based on the ratings and price of each venue. We'll identify places where many venues are located so that any visitor can go to one place and enjoy the option to choose amongst many venue options. We'll also explore areas that are high rated and those that are low rated while also plotting the map of high and low priced venues. Lastly, we'll cluster the venues based on the available information of each venue. This will allow us to clearly identify which venues can be recommended and with what characteristics.

Finally, we'll discuss and conclude which venues to be explored based on visitor requirement of rating and cost.

3 Analysis

[83]:
```
venue_distribution = selected_venues['categories'].value_counts()
colors = cm.rainbow(np.linspace(0, 1, len(venue_distribution.index)))
plt.figure(figsize = (20, 12))
plt.xticks(rotation = 90)
plt.xlabel("Venue category", fontsize = 16)
plt.ylabel("Count", fontsize = 16)
plt.title("Count of venues of each category", fontsize = 16)
plt.bar(venue_distribution.index, venue_distribution.values, color = colors)
```

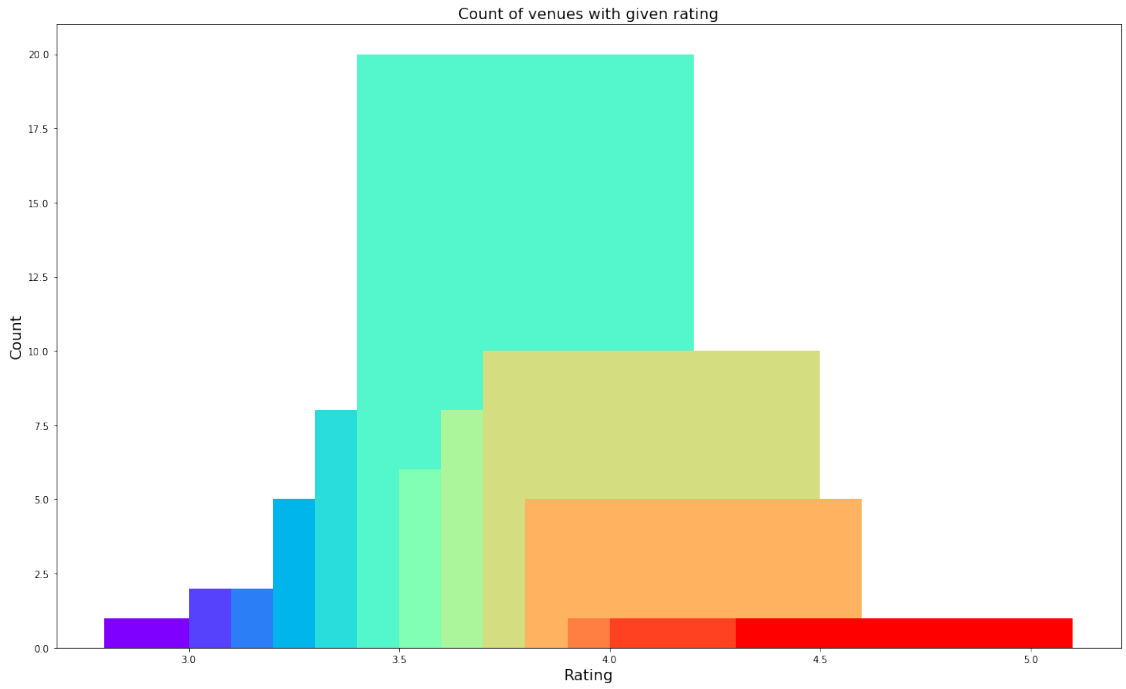
[83]:
```
<BarContainer object of 28 artists>
```



3.0.1 based on ratings

[84]:
```
selected_venues['rating'] = selected_venues['rating'].astype(float)
rating = selected_venues['rating'].value_counts().sort_index()
plt.figure(figsize = (20, 12))
plt.bar(rating.index, rating.values, color = cm.rainbow(np.linspace(0, 1,␣
↪len(rating.index))))
plt.xlabel("Rating", fontsize = 16)
plt.ylabel("Count", fontsize = 16)
plt.title("Count of venues with given rating", fontsize = 16)
```

[84]:
```
Text(0.5, 1.0, 'Count of venues with given rating')
```



From the plot above, it is clear that majority venues have their rating close to 4.

Let's create bins for various ratings and plot them in different colors on the map. The ratings will be divided between 4 bins: 3 to 3.5
3.5 to 4
4 to 4.5
4.5 to 5

[88]:
```
bins = [3.0, 3.5, 4.0, 4.5, 5.0]
labels = ['Low', 'Okay', 'Good', 'Very good']
selected_venues['rating_bin'] = pd.cut(selected_venues['rating'].astype(float),␣
↪bins = bins, labels = labels, include_lowest = True)
```

[89]:
```
color_map = {'Low': 'red', 'Okay': 'orange', 'Good': 'green', 'Very good':␣
↪'darkgreen'}

Mumbai_map = folium.Map(location = [CHD_LATITUDE, CHD_LONGITUDE], zoom_start =␣
↪13)

for name, address, latitude, longitude, rating_bin in␣
↪zip(selected_venues['venue'],

                                                    ␣
↪selected_venues['address'],
```

```
                                                                    ␣
↪selected_venues['latitude'],
                                                                    ␣
↪selected_venues['longitude'],
                                                                    ␣
↪selected_venues['rating_bin']):
    label = '{}, {}'.format(name, address)
    label = folium.Popup(label, parse_html = True)
    folium.Marker(
        [latitude, longitude],
        icon = folium.Icon(color = color_map[rating_bin]),
        popup = label).add_to(Mumbai_map)


Mumbai_map
```

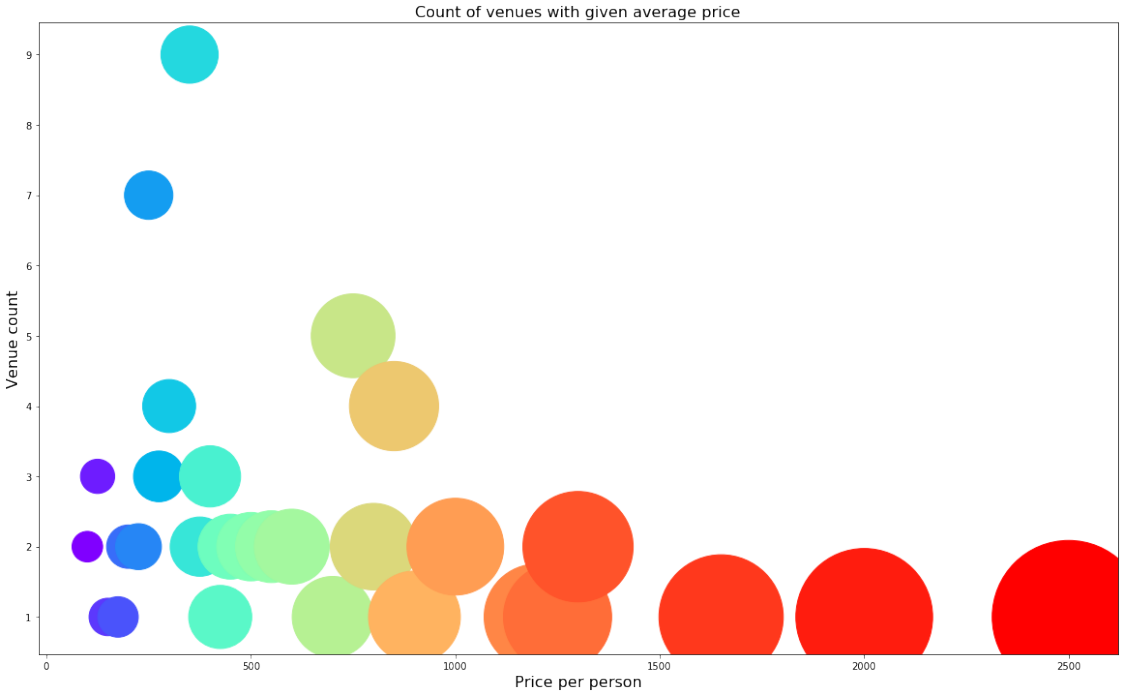
[89]:
```
<folium.folium.Map at 0x7f6864ceed30>
```

3.1 Price

We will now take a look the venues based on the price values. We have two price features for our venues, one is average_price which defines the average cost for one person and the other is price_range which determines the price range as defined by Zomato.
We will first explore the average_price using a scatter plot between the price and the count of venues with that average price. We'll size the points based on the price to highlight their price.

[90]:
```
average_prices = selected_venues['average_price'].value_counts().sort_index()
plt.figure(figsize = (20, 12))
plt.scatter(average_prices.index,
            average_prices.values,
            s = average_prices.index*10,
            c = cm.rainbow(np.linspace(0, 1, len(average_prices.index))))
plt.xlabel("Price per person", fontsize = 16)
plt.ylabel("Venue count", fontsize = 16)
plt.title("Count of venues with given average price", fontsize = 16)
```

[90]:
```
Text(0.5, 1.0, 'Count of venues with given average price')
```



we can see thier are more affordabe hotels in this area and less expensive one

4 clustering

We will now cluster all these venues based on their price range, location and more to identify similar venues and the relationship amongst them. We'll cluster the venues into two separate groups.

[109]:
```
from sklearn.cluster import KMeans

NO_OF_CLUSTERS = 3

clustering = selected_venues.drop(['venue', 'address', 'rating_bin',␣
↪'categories'], 1)
kMeans = KMeans(n_clusters = NO_OF_CLUSTERS, random_state = 0).fit(clustering)
selected_venues.insert(0, 'cluster_labels', kMeans.labels_)
selected_venues.head(11)
```

[109]:

	cluster_labels	categories	venue	latitude	\
0	2	Hotel	Tuskers - Sofitel	19.0673	
1	2	Hotel	022 - Trident	19.0672	
2	2	Indian Restaurant	Masala Library	19.0690	
3	0	Ice Cream Shop	Natural Ice Cream	19.0776	
4	1	Deli / Bodega	Smoke House Deli	19.0688	
5	0	Dessert Shop	Theobroma	19.0637	
6	0	Indian Restaurant	Delhi Zaika	19.0770	

```
7                1         Irani Cafe         SodaBottleOpenerWala   19.0638
8                0        Coffee Shop             Starbucks Coffee   19.0865
9                0  Indian Restaurant                   Naaz Hotel   19.0846
11               1         Restaurant  Ivy Restaurant and Banquets   19.0696

    longitude  price_range  rating  \
0     72.8692          4.0     4.0
1     72.8675          4.0     4.0
2     72.8696          4.0     4.4
3     72.8628          1.0     4.1
4     72.8695          4.0     4.3
5     72.8621          2.0     4.2
6     72.8784          2.0     3.8
7     72.8620          3.0     4.1
8     72.8898          2.0     4.1
9     72.8893          2.0     4.1
11    72.9004          3.0     4.1

                                              address  average_price  \
0   Sofitel Hotel, C 57, Bandra Kurla Complex, Mumbai         1650.0
1   Trident Hotel, C 56, G Block, Bandra Kurla Com…         2000.0
2   Ground Floor, First International Financial Ce…         2500.0
3   3, Gokul Harmony, Kalina Market, Sunder Nagar,…          150.0
4   3A, Ground Floor, 1st International Financial …         1250.0
5   The Capital, Ground Floor, Bandra Kurla Comple…          450.0
6   5/6, Near Kamran Kalpana Theater, LBS Marg, Ku…          350.0
7   Ground Floor, The Capital Building, G Block, B…          750.0
8   First Floor, Phoenix Market City, LBS Marg, Ku…          350.0
9   Kantharia Mahal, LBS Road, Opposite Phoenix Ma…          350.0
11  Above Shopper's Stop, Amar Mahal, Chembur, Mumbai          800.0

   rating_bin
0        Okay
1        Okay
2        Good
3        Good
4        Good
5        Good
6        Okay
7        Good
8        Good
9        Good
11       Good
```

[114]:
```
Mumbai_map = folium.Map(location = [CHD_LATITUDE, CHD_LONGITUDE],␣
 ↪zoom_start = 13)
color_map = { 0: 'blue', 1: 'red',2:'darkgreen'}
```

```
# add venues to the map
markers_colors = []
for venue, address, cluster, latitude, longitude in␣
 ↪zip(selected_venues['venue'],
                                                    ␣
 ↪selected_venues['address'],
                                                    ␣
 ↪selected_venues['cluster_labels'],
                                                    ␣
 ↪selected_venues['latitude'],
                                                    ␣
 ↪selected_venues['longitude']):
    label = folium.Popup(str(venue) + ', ' + str(address), parse_html = True)
    folium.CircleMarker(
        [latitude, longitude],
        radius = 5,
        popup = label,
        color = color_map[cluster],
        fill = True,
        fill_color = color_map[cluster],
        fill_opacity = 0.7).add_to(mumbai_map)

# add cluster centers to the map
for index, cluster in enumerate(kMeans.cluster_centers_):
    latitude = cluster[0]
    longitude = cluster[1]
    label = folium.Popup("Cluster: " + str(index), parse_html = True)
    folium.CircleMarker(
        [latitude, longitude],
        radius = 10,
        popup = label,
        color = color_map[index],
        fill = True,
        fill_color = color_map[index],
        fill_opacity = 0.7).add_to(mumbai_map)


mumbai_map
```

[114]: <folium.folium.Map at 0x7f67e9b8a668>

From the map, we see the three clusters:
The first cluster have all the expensive restaurant
The second cluster is spread across the whole city and includes the majority venues.
The third cluster is very cheap restaurant spread and has very limited venues.
I'll also check the venues of the 3 clusters

[131]:
```
result = selected_venues[selected_venues['cluster_labels'] == 0]
print("Cluster 0")
result.head(10).reset_index(drop = True)
```

```
Cluster 0
```

[131]:

	cluster_labels	categories	venue	latitude	longitude	price_range	rating	address	average_price	rating_bin
0	0	Ice Cream Shop	Natural Ice Cream	19.0776	72.8628	1.0	4.1	3, Gokul Harmony, Kalina Market, Sunder Nagar,…	150.0	Good
1	0	Dessert Shop	Theobroma	19.0637	72.8621	2.0	4.2	The Capital, Ground Floor, Bandra Kurla Comple…	450.0	Good
2	0	Indian Restaurant	Delhi Zaika	19.0770	72.8784	2.0	3.8	5/6, Near Kamran Kalpana Theater, LBS Marg, Ku…	350.0	Okay
3	0	Coffee Shop	Starbucks Coffee	19.0865	72.8898	2.0	4.1	First Floor, Phoenix Market City, LBS Marg, Ku…	350.0	Good
4	0	Indian Restaurant	Naaz Hotel	19.0846	72.8893	2.0	4.1	Kantharia Mahal, LBS Road, Opposite Phoenix Ma…	350.0	Good
5	0	Chinese Restaurant	Hotel Darshan	19.0828	72.8851	1.0	3.5	Opposite Kohinoor City Gate, Kurla West, Near …	125.0	Low
6	0	Snack Place	Oven Hot The Bake Shop	19.0569	72.8981	1.0	3.7	4 & 5, Ganesh Kutir Building, Near Laxmi Store…	100.0	Okay
7	0	Indian Restaurant	Sadguru Veg Diet	19.0622	72.9005	2.0	3.8	Janata Market, Near Chembur Station, Chembur, …	350.0	Okay
8	0	Dessert Shop	Chocday	19.0752	72.9049	1.0	3.7	Unit 15, Vikrant Building, Vikrant Circle, Gha…	175.0	Okay
9	0	Asian Restaurant	Orchids	19.0616	72.9016	3.0	4.0	Hotel Royal Orchid, NG Acharya Marg, Chembur, …	600.0	Okay

[132]:
```
print("These venues for cluster 0 have mean price range of {:.02f} and rating␣
↪spread around {:.02f} and avg price is{:.02f}".
      format(result['price_range'].mean(), result['rating'].astype(float).
↪mean(),result['average_price'].astype(float).mean()))
```

```
These venues for cluster 0 have mean price range of 1.90 and rating spread
```

around 3.81 and avg price is317.71

[126]:
```
result = selected_venues[selected_venues['cluster_labels'] == 1]
print("Cluster 1")
result.head(10).reset_index(drop = True)
```

Cluster 1

[126]:

	cluster_labels	categories	venue	latitude	longitude	price_range	rating	address	average_price	rating_bin
0	1	Deli / Bodega	Smoke House Deli	19.0688	72.8695	4.0	4.3	3A, Ground Floor, 1st International Financial …	1250.0	Good
1	1	Irani Cafe	SodaBottleOpenerWala	19.0638	72.8620	3.0	4.1	Ground Floor, The Capital Building, G Block, B…	750.0	Good
2	1	Restaurant	Ivy Restaurant and Banquets	19.0696	72.9004	3.0	4.1	Above Shopper's Stop, Amar Mahal, Chembur, Mumbai	800.0	Good
3	1	Lounge	Flaming 5 Fine Dine & Lounge	19.0690	72.9004	3.0	3.6	Level 5, The Destination, Near Shopper's Stop,…	700.0	Okay
4	1	Brewery	British Brewing Company	19.1077	72.8834	4.0	4.0	Ground Floor, Times Square Tech Park, Andheri …	1000.0	Okay
5	1	Café	Le Café	19.0617	72.8997	3.0	4.2	1st Road, Opposite B.M.C Office, Chembur, Mumbai	750.0	Good
6	1	Restaurant	BAR BAR	19.0865	72.8893	3.0	3.8	1st Floor, Phoenix Marketcity, Lal Bahadur Sha…	750.0	Okay
7	1	Vegetarian / Vegan Restaurant	Curry Twist	19.1032	72.8863	3.0	3.9	Sagar Pallazio, Andheri Kurla Road, Opposite C…	800.0	Okay
8	1	Lounge	Wild Orchids	19.0616	72.9016	4.0	3.8	Hotel Royal Orchid, NG Acharya Marg, Chembur, …	1000.0	Okay
9	1	Beer Garden	The Beer Cafe	19.0999	72.8751	4.0	4.0	MLCP P10, Landside, Departure Terminal 2, Inte…	1200.0	Okay

[127]:
```
print("These venues for cluster 1 have mean price range of {:.02f} and rating␣
↪spread around {:.02f} and avg price is{:.02f}".
```

```
    format(result['price_range'].mean(), result['rating'].astype(float).
↪mean(),result['average_price'].astype(float).mean()))
```

These venues for cluster 1 have mean price range of 3.32 and rating spread around 4.00 and avg price is915.79

[128]:
```
result = selected_venues[selected_venues['cluster_labels'] == 2]
print("Cluster 1")
result.head(10).reset_index(drop = True)
```

Cluster 1

[128]:

	cluster_labels	categories	venue	latitude	longitude	\
0	2	Hotel	Tuskers - Sofitel	19.0673	72.8692	
1	2	Hotel	022 - Trident	19.0672	72.8675	
2	2	Indian Restaurant	Masala Library	19.0690	72.8696	

	price_range	rating	address	\
0	4.0	4.0	Sofitel Hotel, C 57, Bandra Kurla Complex, Mumbai	
1	4.0	4.0	Trident Hotel, C 56, G Block, Bandra Kurla Com…	
2	4.0	4.4	Ground Floor, First International Financial Ce…	

	average_price	rating_bin
0	1650.0	Okay
1	2000.0	Okay
2	2500.0	Good

[129]:
```
print("These venues for cluster 2 have mean price range of {:.02f} and rating␣
↪spread around {:.02f} and avg price is{:.02f}".
    format(result['price_range'].mean(), result['rating'].astype(float).
↪mean(),result['average_price'].astype(float).mean()))
```

These venues for cluster 2 have mean price range of 4.00 and rating spread around 4.13 and avg price is2050.00

4.0.1 results and discussion

we see that it is you get very marginal rating increase after you reach meadium price range
we see that cluster 0 gets low rating and they are cheap to dine.
we see that if we are willing to pay extra we will get assurance that we will get good dining

5 Conclusion

The purpose of this project was to explore the places that a person visiting Mumbai could visit. The venues have been identified using Foursquare and Zomato API and have been plotted on the map.
we conluded you will need arount 1000 rs to have a good dining in mumbai per person

[ ]: