Project Title:

Restaurants and Eateries information

from Hinjewadi Phase-I, Pune

1. Introduction :

Whenever people move to any other place, they explore the place and try to fetch as much information as possible about it. It can be the neighbourhood, locality, market, price of the place and many more factors including neighbourhood analysis.

This project aims to create information on the restaurants/Hotel information for the people visiting Hinjewadi Phase-I for the purpose of official work. On completion of official work, People generally visit nearby places/restaurants/Hotels for the purpose of Leisure/food. The project aims at providing the meaningful information on the availability of such avenues to help the executives visiting Hinjewadi Area. For the purpose of implementation, tt is assumed that the customer is visiting the Hinjewadi Phase-I in Infosys DC, Pune for two to three days to complete some official work. His preference for stay is "Hotel Lemon Tree", which is near to Infosys Development Centre, Hinjewadi Phase-I, Pune.

1. Data Sets and API:

• Foursquare API:

This API has a database of more than 105 million places. This project would use Four-square API as its prime data gathering source. Many organizations are using to geo-tag their photos with detailed info about a destination, while also serving up contextually relevant locations for those who are searching for a place to eat, drink or explore. This API provides the ability to perform location search, location sharing and details about a business. Foursquare users can also use photos, tips and reviews in many productive ways to add value to the results. This API will be used to explore the information of various restaurants available in the Hinjewadi Area.

• Folium:

Python visualization library would be used to visualize the neighbourhoods cluster distribution of Hinjewadi Area, Pune to derive the desirable insights from the outcomes using python’s scientific libraries Pandas, NumPy.

1. Methodology :

The Steps used for the the purpose of solving the defined problem is given as below:

1. Importing the important Libraries
2. Loading data from Foursquare app
3. Getting location information
4. Getting list of Hotels and Their ratings from Foursquare app
5. Exploring the nearby area for restaurants and eateries

The results obtained from the application of methodology and further discussion is given in the coming sections.

1. Results :
2. Importing important libraries

import requests # library to handle requests

import pandas as pd # library for data analsysis

import numpy as np # library to handle data in a vectorized manner

import random # library for random number generation

!conda install -c conda-forge geopy --yes

from geopy.geocoders import Nominatim # module to convert an address into latitude and longitude values

# libraries for displaying images

from IPython.display import Image

from IPython.core.display import HTML

# tranforming json file into a pandas dataframe library

from pandas.io.json import json\_normalize

!conda install -c conda-forge folium=0.5.0 --yes

import folium # plotting library

1. Loading data from FourSquare App

# your Foursquare ID

CLIENT\_ID = 'RO4LIOP5MMIONX2KITRQ3QEBS2FZYWW13MAGJRJZL0T3PBVK'

# your Foursquare Secret

CLIENT\_SECRET = '5XKOATUE1FBD0QP5UHXJSSAZFRBZ1LCCYLH1Z1PQW0C1KMTE'

VERSION = '20180604'

LIMIT = 30

print('Your credentails:')

print('CLIENT\_ID: ' + CLIENT\_ID)

print('CLIENT\_SECRET:' + CLIENT\_SECRET)

1. Getting Location Information

address = 'Lemon Tree Hotel, Pune'

geolocator = Nominatim(user\_agent="foursquare\_agent")

location = geolocator.geocode(address)

latitude = location.latitude

longitude = location.longitude

print(latitude, longitude)

1. Getting list of Hotels and Their ratings from Foursquare app

# Search Nearby Restaurants withn 2 km range

search\_query = 'Indian Restaurants'

radius = 2000 # within the duration of two kilometer

print(search\_query + ' .... OK!')

#Get Data from FourSquare app

url = 'https://api.foursquare.com/v2/venues/search?client\_id={}&client\_secret={}&ll={},{}&v={}&query={}&radius={}&limit={}'.format(CLIENT\_ID, CLIENT\_SECRET, latitude, longitude, VERSION, search\_query, radius, LIMIT)

# Get the results

results = requests.get(url).json()

# assign relevant part of JSON to venues

venues = results['response']['venues']

# tranform venues into a dataframe

dataframe = json\_normalize(venues)

dataframe.head(10)

# keep only columns that include venue name, and anything that is associated with location

filtered\_columns = ['name', 'categories'] + [col for col in dataframe.columns if col.startswith('location.')] + ['id']

dataframe\_filtered = dataframe.loc[:, filtered\_columns]

# function that extracts the category of the venue

def get\_category\_type(row):

try:

categories\_list = row['categories']

except:

categories\_list = row['venue.categories']

if len(categories\_list) == 0:

return None

else:

return categories\_list[0]['name']

# filter the category for each row

dataframe\_filtered['categories'] = dataframe\_filtered.apply(get\_category\_type, axis=1)

# clean column names by keeping only last term

dataframe\_filtered.columns = [column.split('.')[-1] for column in dataframe\_filtered.columns]

dataframe\_filtered.name

venues\_map = folium.Map(location=[latitude, longitude], zoom\_start=13) # generate map centred around the Hotel Lemon Tree

# add a red circle marker to represent the Conrad Hotel

folium.features.CircleMarker(

[latitude, longitude],

radius=10,

color='red',

popup = 'Lemon Tree Hotel, Pune',

fill = True,

fill\_color = 'red',

fill\_opacity = 0.6

).add\_to(venues\_map)

# add the Italian restaurants as blue circle markers

for lat, lng, label in zip(dataframe\_filtered.lat, dataframe\_filtered.lng, dataframe\_filtered.categories):

folium.features.CircleMarker(

[lat, lng],

radius=5,

color='blue',

popup=label,

fill = True,

fill\_color='blue',

fill\_opacity=0.6

).add\_to(venues\_map)

# display map

venues\_map

# ID of Lamon Tree Restaurant

venue\_id = '4bb8aee51261d13a0bcee898'

url = 'https://api.foursquare.com/v2/venues/{}?client\_id={}&client\_secret={}&v={}'.format(venue\_id, CLIENT\_ID, CLIENT\_SECRET, VERSION)

url

try:

print(result['response']['venue']['rating'])

except:

print('This venue has not been rated yet.')

venue\_id = '4bb8aee51261d13a0bcee898' # ID of Lamon Tree Restaurant = 'https://api.foursquare.com/v2/venues/{}?client\_id={}&client\_secret={}&v={}'.format(venue\_id, CLIENT\_ID, CLIENT\_SECRET, VERSION)

url = 'https://api.foursquare.com/v2/venues/{}?client\_id={}&client\_secret={}&v={}'.format(venue\_id, CLIENT\_ID, CLIENT\_SECRET, VERSION)

result = requests.get(url).json()

# result['response']['venue']['tips']['count']

1. Exploring the nearby area for restaurants and eateries

## Ecco Tips

limit = 15 # set limit to be greater than or equal to the total number of tips

url = 'https://api.foursquare.com/v2/venues/{}/tips?client\_id={}&client\_secret={}&v={}&limit={}'.format(venue\_id, CLIENT\_ID, CLIENT\_SECRET, VERSION, limit)

results = requests.get(url).json()

#results

#tips = results['response']['tips']['items']

tip = results['response']['tips']['items'][0]

tip.keys()

pd.set\_option('display.max\_colwidth', -1)

tips\_df = json\_normalize(tips) # json normalize tips

# columns to keep

filtered\_columns = ['text', 'agreeCount', 'disagreeCount', 'id', 'user.firstName', 'user.lastName', 'user.gender', 'user.id']

tips\_filtered = tips\_df.loc[:, filtered\_columns]

# display tips

tips\_filtered

import requests

results = requests.get(url).json()

'There are {} around Ecco restaurant.'.format(len(results['response']['groups'][0]['items']))

items = results['response']['groups'][0]['items']

items[0]

dataframe = json\_normalize(items) # flatten JSON

# filter columns

filtered\_columns = ['venue.name', 'venue.categories'] + [col for col in dataframe.columns if col.startswith('venue.location.')] + ['venue.id']

dataframe\_filtered = dataframe.loc[:, filtered\_columns]

# filter the category for each row

dataframe\_filtered['venue.categories'] = dataframe\_filtered.apply(get\_category\_type, axis=1)

# clean columns

dataframe\_filtered.columns = [col.split('.')[-1] for col in dataframe\_filtered.columns]

dataframe\_filtered.head(28)

venues\_map = folium.Map(location=[latitude, longitude], zoom\_start=15) # generate map centred around Ecco

# add Ecco as a red circle mark

folium.features.CircleMarker(

[latitude, longitude],

radius=10,

popup='Ecco',

fill=True,

color='red',

fill\_color='red',

fill\_opacity=0.6

).add\_to(venues\_map)

# add popular spots to the map as blue circle markers

for lat, lng, label in zip(dataframe\_filtered.lat, dataframe\_filtered.lng, dataframe\_filtered.categories):

folium.features.CircleMarker(

[lat, lng],

radius=5,

popup=label,

fill=True,

color='blue',

fill\_color='blue',

fill\_opacity=0.6

).add\_to(venues\_map)

# display map

venues\_map

Fig 1. Map showing Good Restaurants within the distance of two Kilometres from Hotel Lemon Tree

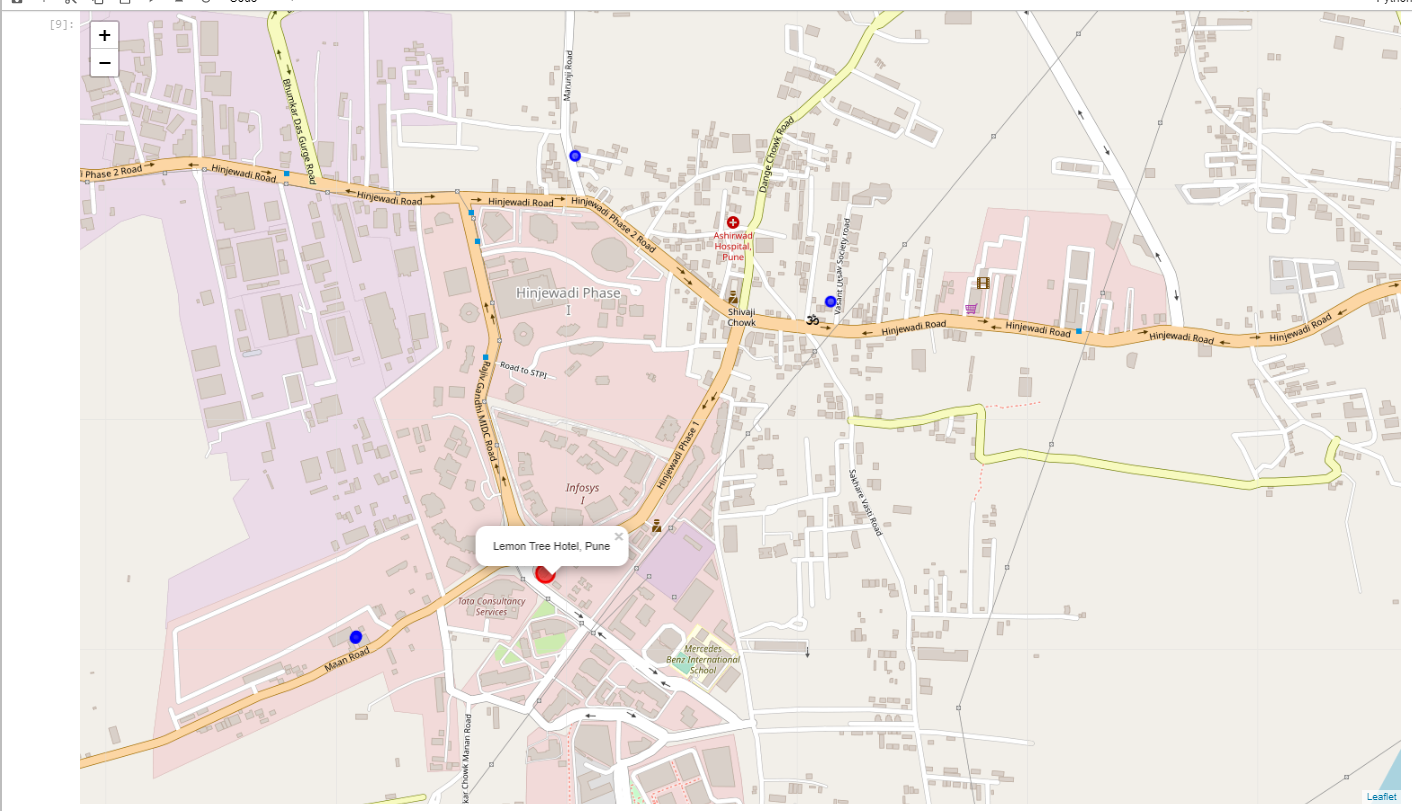
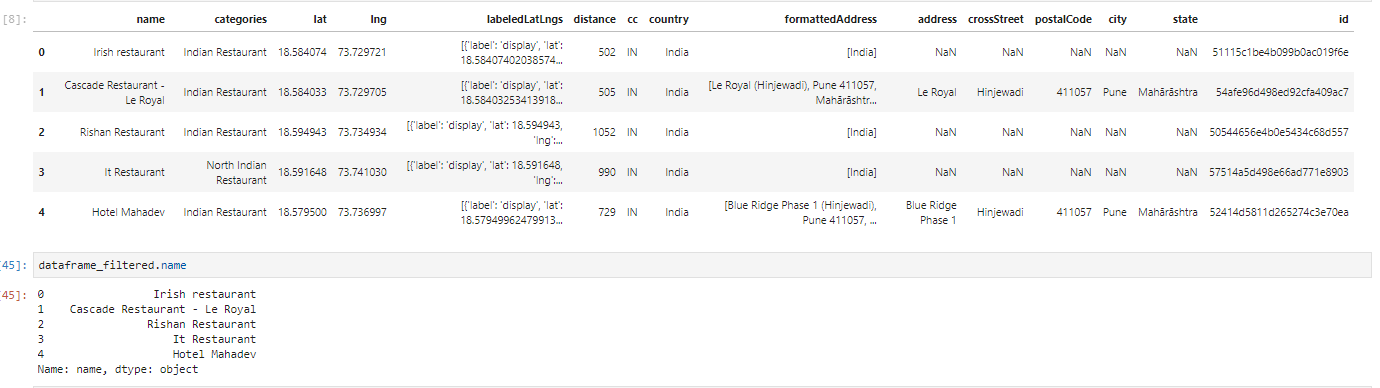


Fig 2. Details of Good Restaurants within the distance of two Kilometres from Hotel Lemon Tree



According to data received from FourSquare app Two Restaurants from the list viz

1. “Irish Restaurant” and
2. “Cascade Restaurant-Le-Royal”

Have rating of 07 out of 10 also the rating for the “Hotel Lemon Tree” has rating of 07 out of 10.

Fig3. List of Restaurants/Eateries/Places to visit near hotel Lemon Tree within the limit of two kilometres from Hotel Lemon Tree

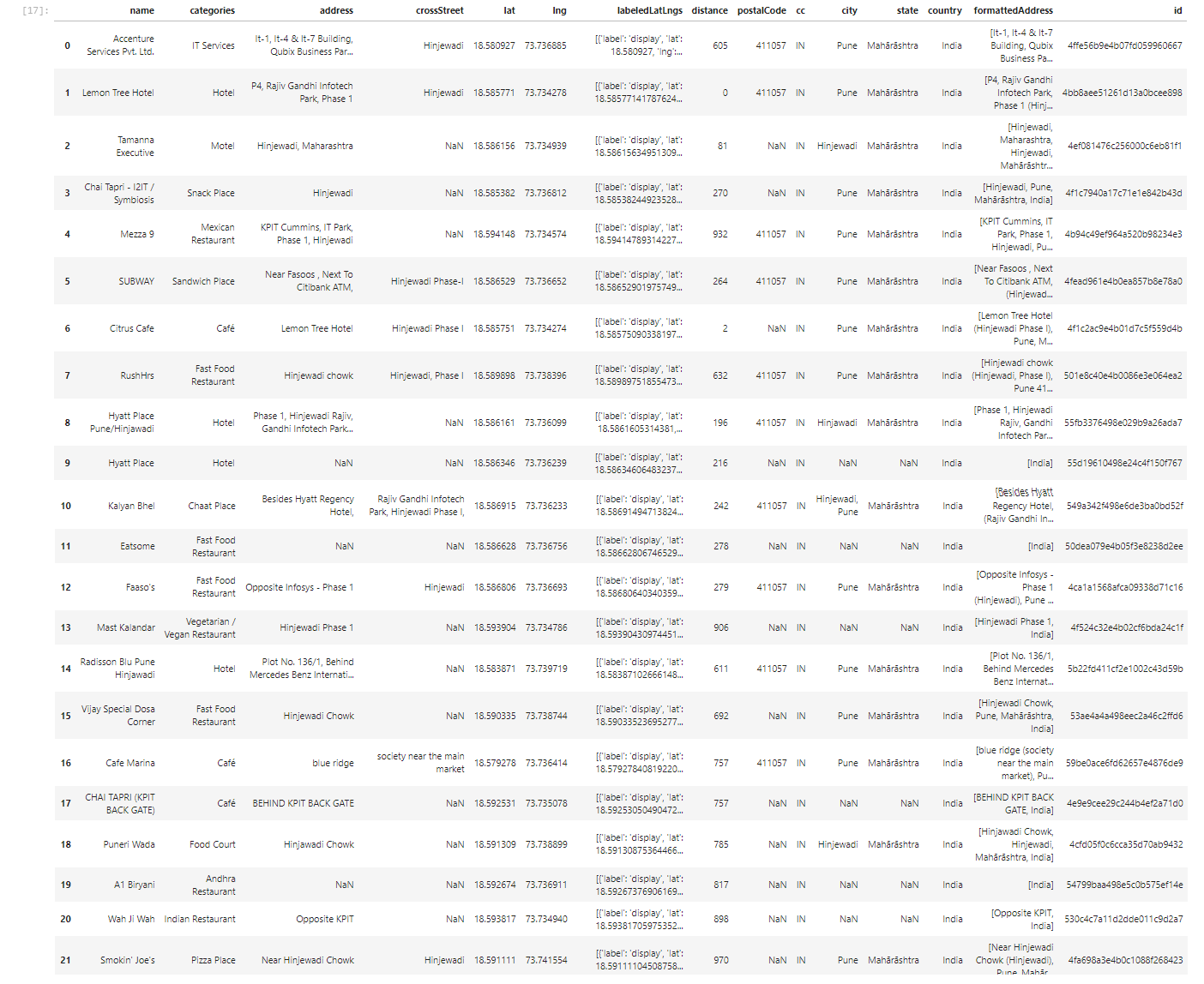
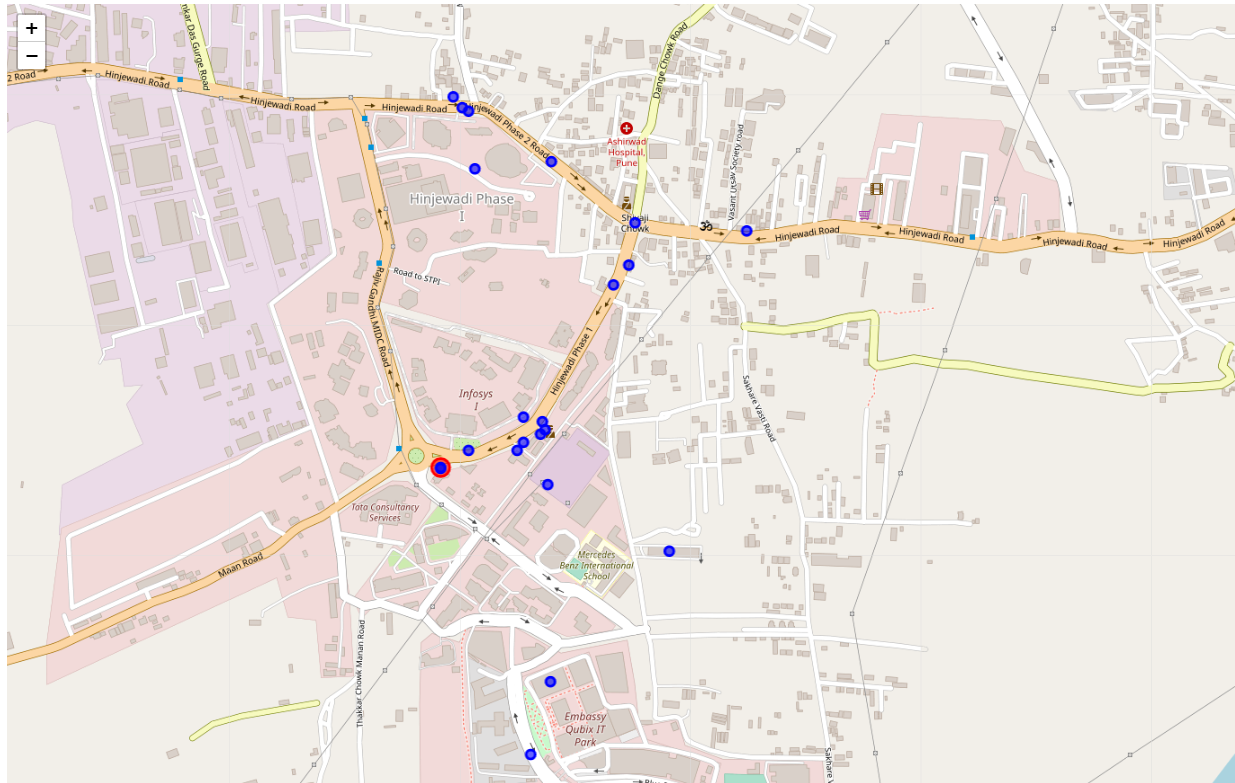


Fig4. Mapsshowing Restaurants/Eateries/Places to visit near hotel Lemon Tree within the limit of two kilometres from Hotel Lemon Tree.



1. Discussion :

From the data explored from Foursquare app and its representation on the Folium map it is found that there are three good Hotels/Restaurants with Rating 07 out of 10 in the Hinjewadi Area those are:

1. Hotel Lemon Tree
2. “Irish Restaurant” and
3. “Cascade Restaurant-Le-Royal”

These restaurants are excellent for Staying as well as having dinner/Lunch.

The people visiting this area can also enjoy evening snacks or alternate food in the nearby area where around 15-20 places/eateries/small size restaurants are available.

1. Conclusion :

The client’’s decision to stay at “Hotel Lemon Tree” is good, as it has good rating as well as it is available nearby to Infosys Development Center, Pune. He can also have relaxed time in the evening and enjoy the delicious food in nearby restaurants in evening.