

# **Exploring Neighborhoods of Hyderabad, India – To decide on ideal location for Food Court**

Tirthankar Baidya

June 14, 2021



# **1. Introduction**

## **1.1 Background**

Hyderabad is considered as India's Best City to Live as per Mercer's Quality of Living rankings 2020, for consecutive 6<sup>th</sup> time. With growing IT sector, along with other industries like Pharmaceuticals and Automobiles, it has become a hub of investment.

There is a huge boom in real sector, and the city is becoming more and more cosmopolitans. While there is an even distribution of population demographics, a spike in young IT professional is observed . This crowd loves to freak out and ideal place to hang around will be food court. While there are quite a few varieties of restaurant across the city, there is lack of food courts at certain places in the city. This places are having unexplored potential.

The purpose of this project is to locate certain areas in Hyderabad, which is having a huge potential for Food court Business with good foot fall.

## **1.2 Interested audience**

The target audience for such a project is twofold.

- Any Food court joints or aggregators who is looking for lucrative investment in Hyderabad
- Any real estate developers, who is looking for Commercial investment

## 2. Data

### 2.1 Data Points

- List of Neighborhoods in Hyderabad, India (scope of this project is limited to Hyderabad)
- Latitude & Longitude coordinates of those neighborhoods. Purpose of this is to plot the map and also to get the venue data
- Venue data related to Food court, which will be used for clustering of neighborhoods.

### 2.2 Data Sources

- The **Wikipedia** page ([https://en.wikipedia.org/wiki/Category:Neighbourhoods\\_in\\_Hyderabad,\\_India](https://en.wikipedia.org/wiki/Category:Neighbourhoods_in_Hyderabad,_India)) contains a list of neighborhoods in Hyderabad, India
- **Foursquare** API is used to fetch categories of venue data. Foursquare has one of the largest data base with 105+million places and used by over 125,000 developers across the Globe

### 2.3 Data extraction

- Web scraping techniques to extract data from Wikipedia page with the help of Python requests and BeautifulSoup packages. Geographical coordinates of the neighborhoods are derived using Python Geocoder package which will give us latitude and longitude coordinates of neighborhoods.
- Foursquare API will be used to get venue data for those neighborhoods. The categories of venue data related to Food court will be used for project

# Codes and Out put

## 1. Web scraping from Wikipedia page using BeautifulSoup

```
In [4]: # send the GET request
data = requests.get("https://en.wikipedia.org/wiki/Category:Neighbourhoods_in_Hyderabad,_India").text

In [5]: # parse data from the html into a beautifulsoup object
soup = BeautifulSoup(data, 'html.parser')

In [6]: # create a list to store neighborhood data
neighborhoodList = []

In [7]: # append the data into the list
for row in soup.find_all("div", class_="mw-category")[0].findAll("li"):
    neighborhoodList.append(row.text)

In [8]: # create a new DataFrame from the list
Hyd_df = pd.DataFrame({"Neighborhood": neighborhoodList})

Hyd_df.head()
```

Out[8]:

	Neighborhood
0	A. C. Guards
1	A. S. Rao Nagar
2	Abhyudaya Nagar
3	Abids
4	Adibatla

## 2. Geographical co-ordinates of the neighborhoods are derived using Python Geocoder package

```
In [10]: # define a function to get coordinates
def get_latlng(neighborhood):
    # initialize your variable to None
    lat_lng_coors = None
    # loop until you get the coordinates
    while(lat_lng_coors is None):
        g = geocoder.arcgis('{} , Hyderabad, India'.format(neighborhood))
        lat_lng_coors = g.latlng
    return lat_lng_coors

In [11]: # call the function to get the coordinates, store in a new list using list comprehension
coords = [ get_latlng(neighborhood) for neighborhood in Hyd_df["Neighborhood"].tolist() ]
df_coors = pd.DataFrame(coords, columns=['Latitude', 'Longitude'])

# merge the coordinates into the original dataframe
Hyd_df['Latitude'] = df_coors['Latitude']
Hyd_df['Longitude'] = df_coors['Longitude']

# check the neighborhoods and the coordinates
print(Hyd_df.shape)
Hyd_df
```

	Neighborhood	Latitude	Longitude
0	A. C. Guards	17.395015	78.459812
1	A. S. Rao Nagar	17.411200	78.508240
2	Abhyudaya Nagar	17.337650	78.564140
3	Abids	17.389800	78.476580

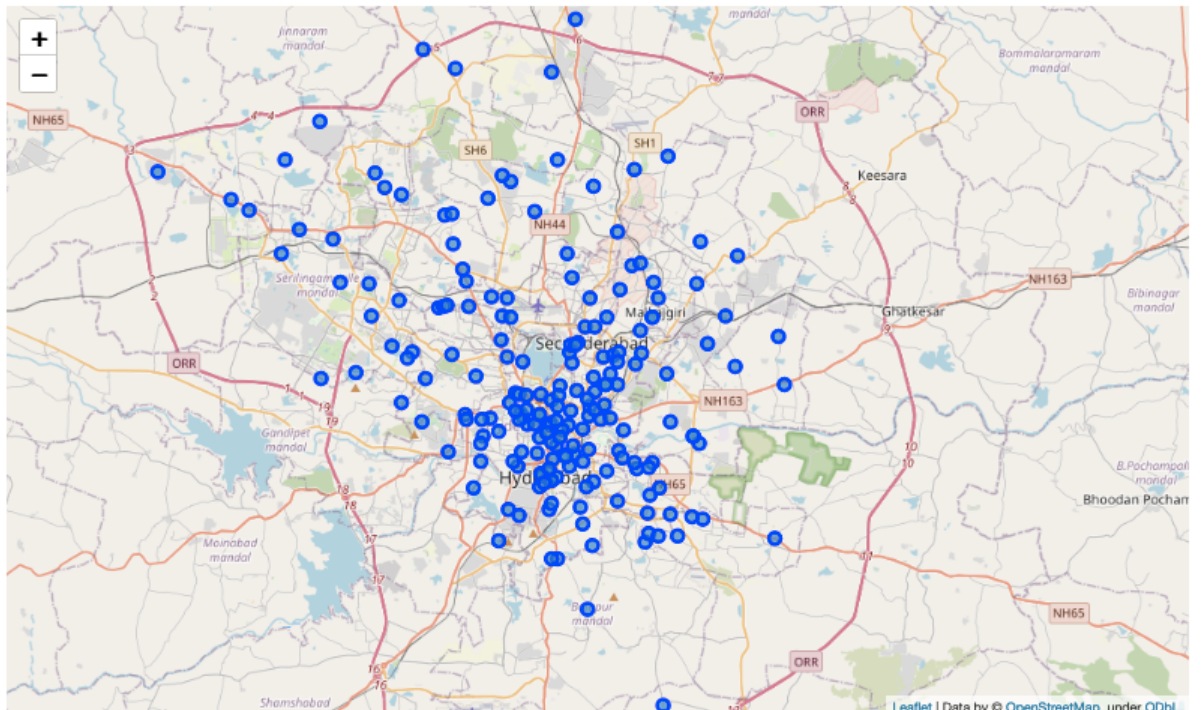
### 3. Hyderabad map plotted with Co-ordinates

```
In [13]: # create map of Hyderabad using latitude and longitude values
map_Hyd = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, neighborhood in zip(Hyd_df['Latitude'], Hyd_df['Longitude'], Hyd_df['Neighborhood']):
    label = '{}'.format(neighborhood)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7).add_to(map_Hyd)
```

map\_Hyd

Out[13]:



#### 4. Using Foursquare API for details

```
In [14]: # define Foursquare Credentials and Version
CLIENT_ID = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX' # your Foursquare ID
CLIENT_SECRET = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX' # your Foursquare Secret
VERSION = '20180605' # Foursquare API version

print('Your credentials:')
print('CLIENT_ID: ' + CLIENT_ID)
print('CLIENT_SECRET: ' + CLIENT_SECRET)

Your credentials:
CLIENT_ID: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
CLIENT_SECRET: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

In [15]: ### Now, let's get the top 100 venues that are within a radius of 2000 meters.
radius = 2000
LIMIT = 100

venues = []

for lat, long, neighborhood in zip(Hyd_df['Latitude'], Hyd_df['Longitude'], Hyd_df['Neighborhood']):

    # create the API request URL
    url = "https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&v={}≪={},{}&radius={}&limit={}"
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        lat,
        long,
        radius,
        LIMIT)

    # make the GET request
    results = requests.get(url).json()['response']['groups'][0]['items']

    # return only relevant information for each nearby venue
    for venue in results:
        venues.append({
            neighborhood,
            lat,
            long,
            venue['venue']['name'],
            venue['venue']['location']['lat'],
            venue['venue']['location']['lng'],
            venue['venue']['categories'][0]['name']})

In [16]: # convert the venues list into a new DataFrame
Hyd_new_df = pd.DataFrame(venues)

# define the column names
Hyd_new_df.columns = ['Neighborhood', 'Latitude', 'Longitude', 'VenueName', 'VenueLatitude', 'VenueLongitude', 'VenueCategories']

print(Hyd_new_df.shape)
Hyd_new_df.head()

(5651, 7)
```

Out[16]:	Neighborhood	Latitude	Longitude	VenueName	VenueLatitude	VenueLongitude	VenueCategory	
	0	A. C. Guards	17.395015	78.459812	Chicha's	17.403255	78.460152	Hyderabadi Restaurant
	1	A. C. Guards	17.395015	78.459812	Subhan Bakery	17.392412	78.464712	Bakery
	2	A. C. Guards	17.395015	78.459812	Jewel Of Nizam	17.403869	78.461194	Middle Eastern Restaurant
	3	A. C. Guards	17.395015	78.459812	Nizam club	17.403221	78.468729	Lounge