Exploring Neighborhoods of Hyderabad, India –

To decide on ideal location for Food Court

Tirthankar Baidya
June 15, 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 6th 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) 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

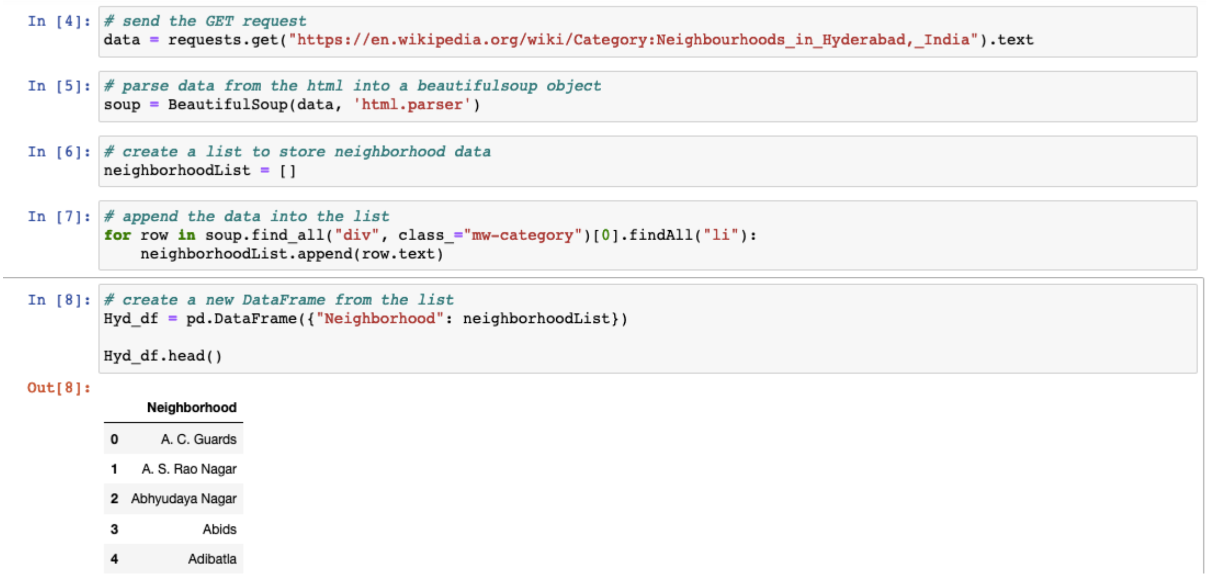
2.3 **Data extraction**

- Web scraping techniques to extract data from Wikipedia page with the help of Python requests and Beautifulsoup packages. Geographical co-ordinates 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

3. Methodology

3.1 Web scraping using python to extract list of neighborhoods

- The **Wikipedia** page (https://en.wikipedia.org/wiki/Category:Neighbourhoods_in_Hyderabad,_India) contains a list of neighborhoods in Hyderabad, India.
- Web scarping from above mentioned Wiki page is done to extract list of neighbor hoods.
- Code with screen shot

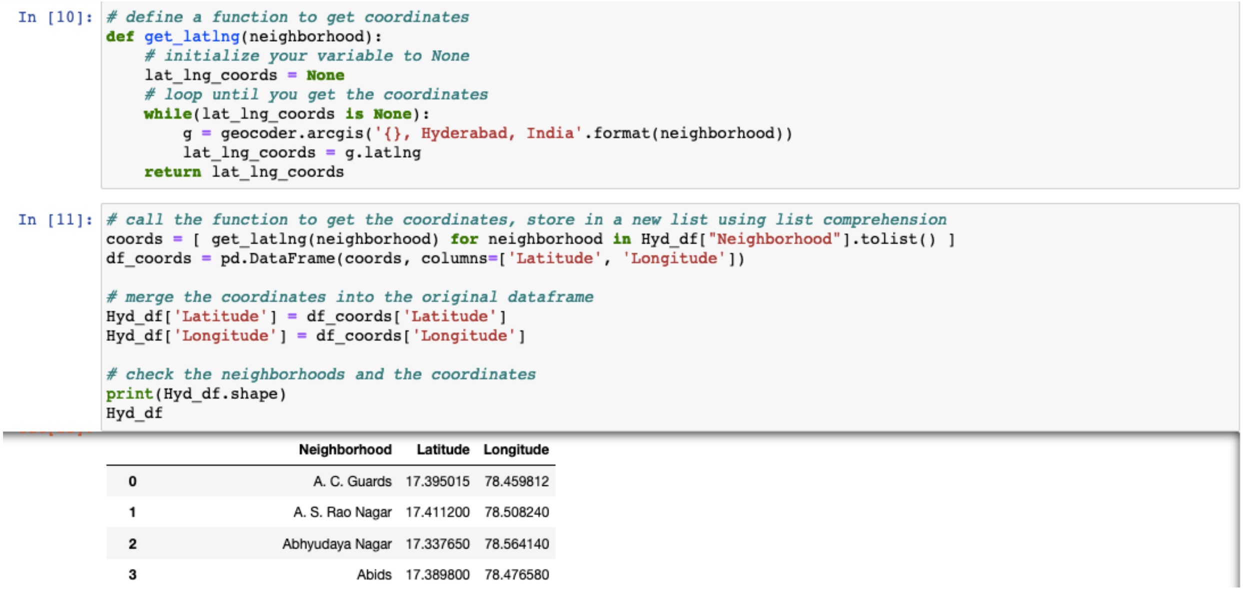


```
In [4]: # send the GET request
        data = requests.get("https://en.wikipedia.org/wiki/Category:Neighbourhoods_in_Hyderabad,_India").text

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

3.2 Deriving Geographical coordinates for this List of Neighborhoods

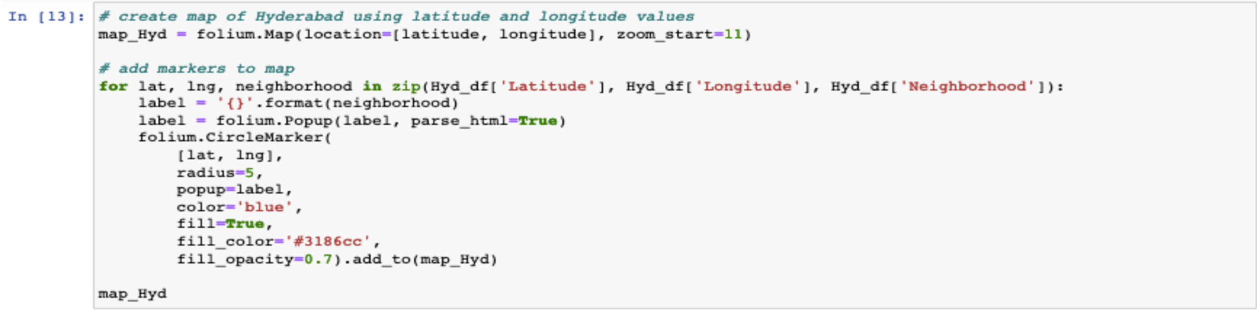
- The geographical coordinates are required for this neighborhoods in order to use Foursquare API, which further helps to deep dive and analyze data
- To do this Geocoder package is used to link this addresses with geographical coordinates (Latitude & Longitude)
- The data is populated in pandas Data-Frame and visualized in map. Folium package is used to visualize this data
- Code with screenshot



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

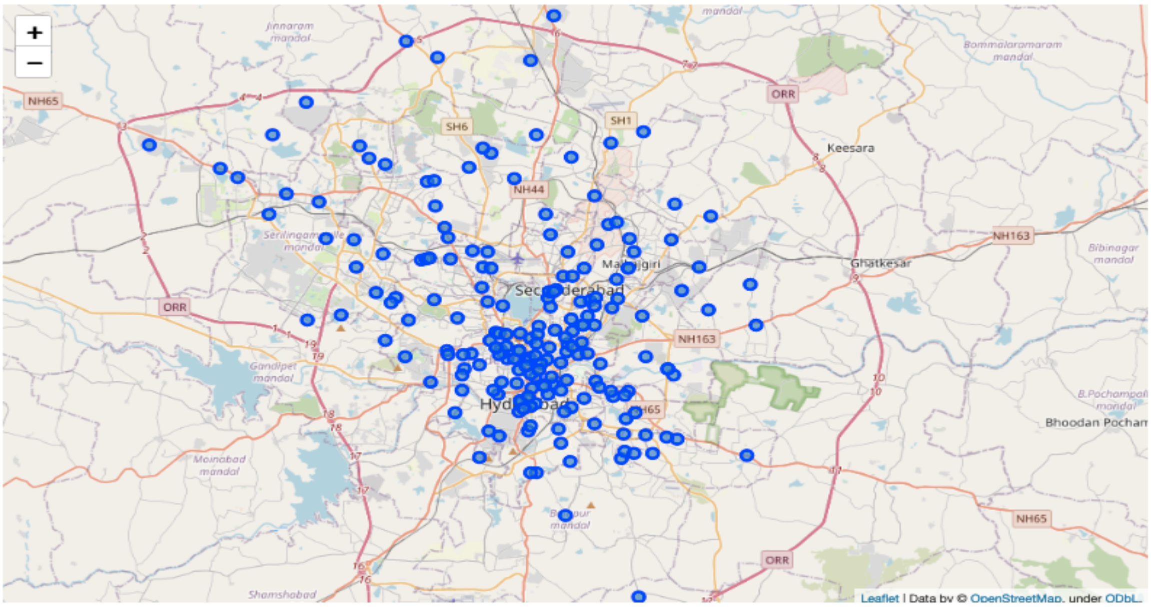
```
In [11]: # call the function to get the coordinates, store in a new list using list comprehension
         coords = [ get_latlng(neighborhood) for neighborhood in Hyd_df["Neighborhood"].tolist() ]
         df_coords = pd.DataFrame(coords, columns=['Latitude', 'Longitude'])

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



```
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
```

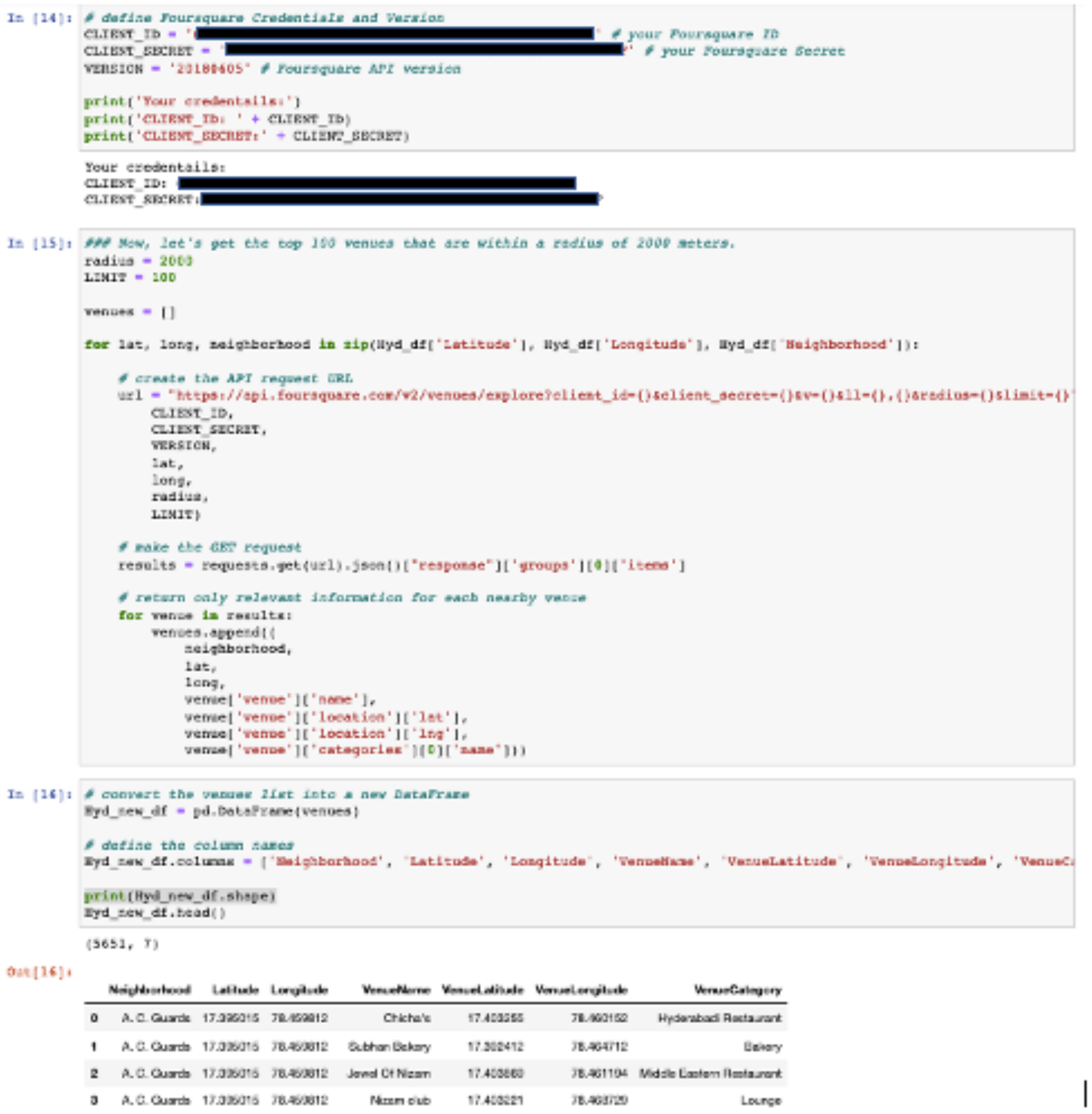
Out[13]:



3.3 Derive the top 100 venues for each neighborhood within 2km radius and filter those with "Food court" as criteria

- API calls to Foursquare, using geographical coordinates of neighborhoods and fetch venue data

- Venue data includes name, category, latitude and longitude
- With this data, number of venues related to each and every neighborhood and unique categories can be derived.
- Venue category is now filtered to Food court, to derive final clustering of Neighborhood, as this project is related to Food court availability
- Code with screenshot

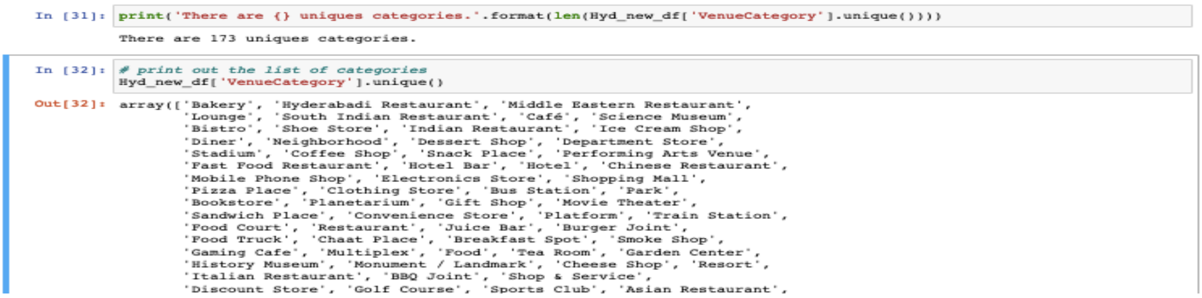


```
In [14]: # define Foursquare Credentials and Version
         CLIENT_ID = '[redacted]' # your Foursquare ID
         CLIENT_SECRET = '[redacted]' # your Foursquare Secret
         VERSION = '20180405' # Foursquare API version

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

         Your credentails:
         CLIENT_ID: [redacted]
         CLIENT_SECRET:[redacted]
```

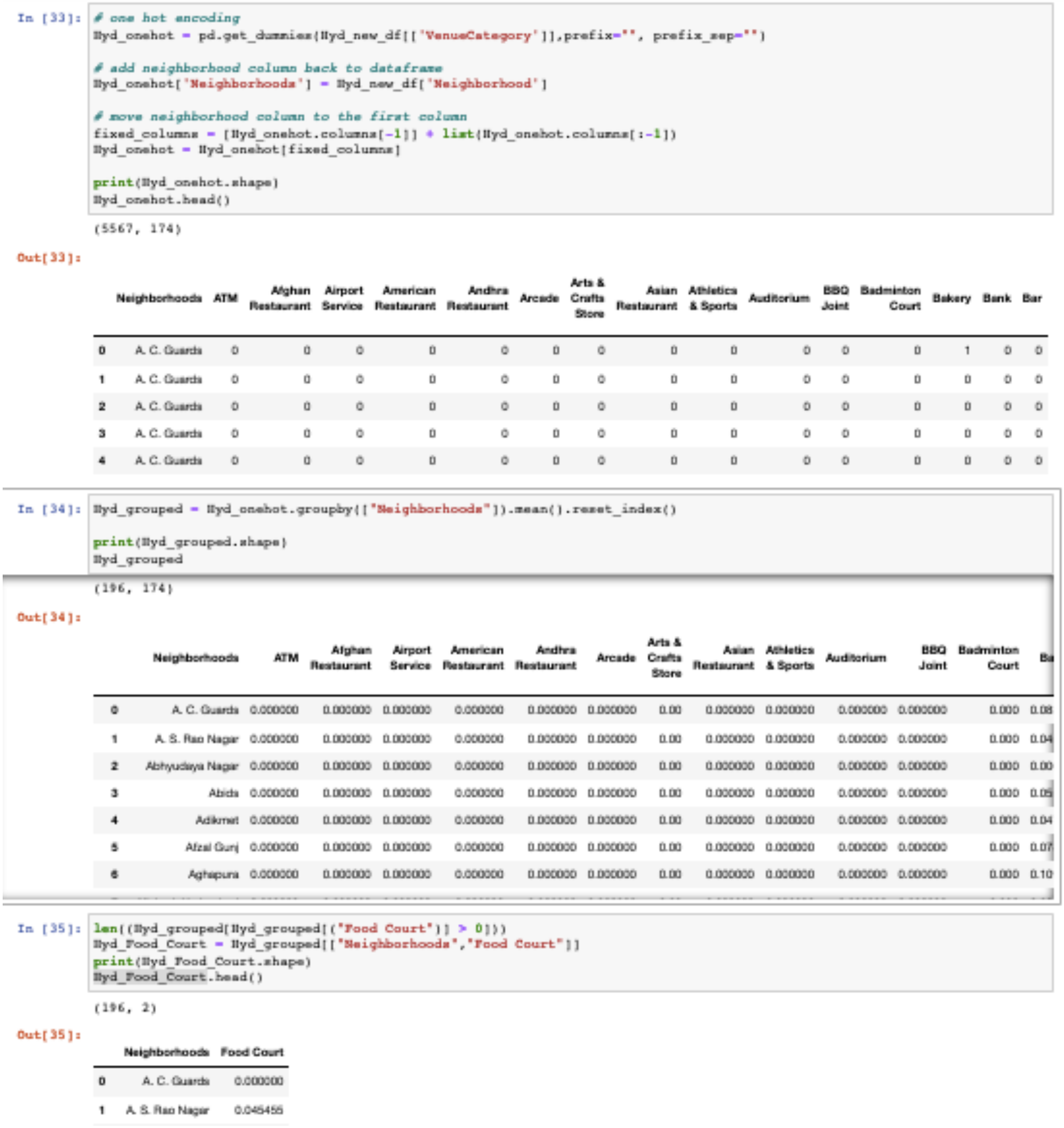
```
In [15]: ### Now, let's get the top 100 venues that are within a radius of 2000 meters.
         radius = 2000
         LIMIT = 100

         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={}&ll={},{}&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', 'VenueC

         print(Hyd_new_df.shape)
         Hyd_new_df.head()

         (5651, 7)
Out[16]:
```

	Neighborhood	Latitude	Longitude	VenueName	VenueLatitude	VenueLongitude	VenueCategory
0	A. C. Guards	17.396015	78.459812	Chicha's	17.403255	78.460152	Hyderabadi Restaurant
1	A. C. Guards	17.396015	78.459812	Subhan Bakery	17.392412	78.464712	Bakery
2	A. C. Guards	17.396015	78.459812	Jewel Of Nizam	17.403863	78.461104	Middle Eastern Restaurant
3	A. C. Guards	17.396015	78.459812	Nizam club	17.403221	78.463729	Lounge



```
In [31]: print('There are {} uniques categories.'.format(len(Hyd_new_df['VenueCategory'].unique())))

         There are 173 uniques categories.

In [32]: # print out the list of categories
         Hyd_new_df['VenueCategory'].unique()

Out[32]: array(['Bakery', 'Hyderabadi Restaurant', 'Middle Eastern Restaurant',
                'Lounge', 'South Indian Restaurant', 'Café', 'Science Museum',
                'Bistro', 'Shoe Store', 'Indian Restaurant', 'Ice Cream Shop',
                'Diner', 'Neighborhood', 'Dessert Shop', 'Department Store',
                'Stadium', 'Coffee Shop', 'Snack Place', 'Performing Arts Venue',
                'Fast Food Restaurant', 'Hotel Bar', 'Hotel', 'Chinese Restaurant',
                'Mobile Phone Shop', 'Electronics Store', 'Shopping Mall',
                'Pizza Place', 'Clothing Store', 'Bus Station', 'Park',
                'Bookstore', 'Planetarium', 'Gift Shop', 'Movie Theater',
                'Sandwich Place', 'Convenience Store', 'Platform', 'Train Station',
                'Food Court', 'Restaurant', 'Juice Bar', 'Burger Joint',
                'Food Truck', 'Chaat Place', 'Breakfast Spot', 'Smoke Shop',
                'Gaming Cafe', 'Multiplex', 'Food', 'Tea Room', 'Garden Center',
                'History Museum', 'Monument / Landmark', 'Cheese Shop', 'Resort',
                'Italian Restaurant', 'BBQ Joint', 'Shop & Service',
                'Discount Store', 'Golf Course', 'Sports Club', 'Asian Restaurant',
```



In [33]:
```
# one hot encoding
Hyd_onehot = pd.get_dummies(Hyd_new_df[['VenueCategory']],prefix="", prefix_sep="")

# add neighborhood column back to dataframe
Hyd_onehot['Neighborhoods'] = Hyd_new_df['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [Hyd_onehot.columns[-1]] + list(Hyd_onehot.columns[:-1])
Hyd_onehot = Hyd_onehot[fixed_columns]

print(Hyd_onehot.shape)
Hyd_onehot.head()
```

```
(5567, 174)
```

Out[33]:

	Neighborhoods	ATM	Afghan Restaurant	Airport Service	American Restaurant	Andhra Restaurant	Arcade	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Auditorium	BBQ Joint	Badminton Court	Bakery	Bank	Bar
0	A. C. Guards	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	A. C. Guards	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	A. C. Guards	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	A. C. Guards	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	A. C. Guards	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In [34]:
```
Hyd_grouped = Hyd_onehot.groupby(["Neighborhoods"]).mean().reset_index()

print(Hyd_grouped.shape)
Hyd_grouped
```

```
(196, 174)
```

Out[34]:

	Neighborhoods	ATM	Afghan Restaurant	Airport Service	American Restaurant	Andhra Restaurant	Arcade	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	Auditorium	BBQ Joint	Badminton Court	Ba
0	A. C. Guards	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000	0.08
1	A. S. Rao Nagar	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000	0.04
2	Abhyudaya Nagar	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000	0.00
3	Abids	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000	0.05
4	Adikmet	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000	0.04
5	Afzal Gunj	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000	0.07
6	Aghapura	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00	0.000000	0.000000	0.000000	0.000000	0.000	0.10

In [35]:
```
len((Hyd_grouped[Hyd_grouped[("Food Court")] > 0]))
Hyd_Food_Court = Hyd_grouped[["Neighborhoods","Food Court"]]
print(Hyd_Food_Court.shape)
Hyd_Food_Court.head()
```

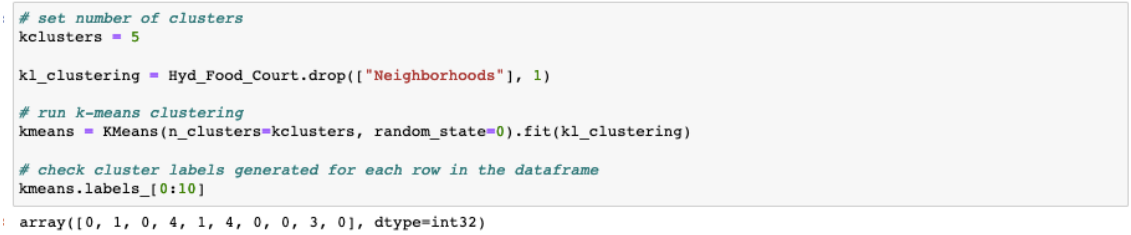
```
(196, 2)
```

Out[35]:

	Neighborhoods	Food Court
0	A. C. Guards	0.000000
1	A. S. Rao Nagar	0.045455

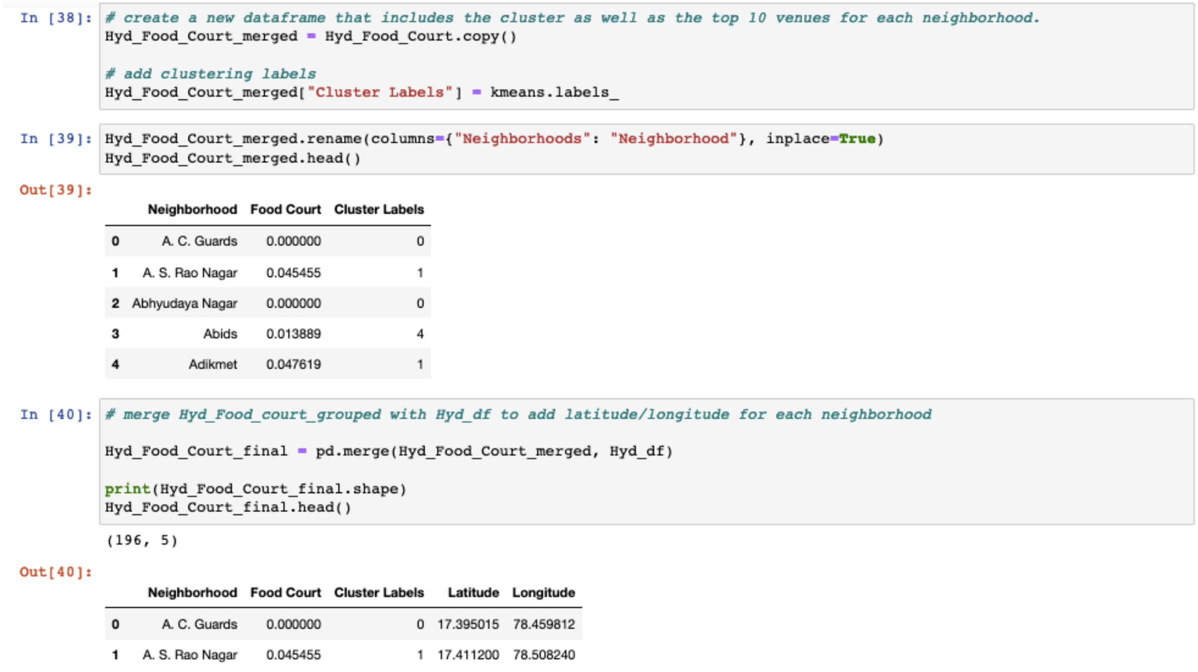
3.4 Clustering of data using K-mean clustering

- K- mean clustering identifies K number of centroids and ever data point gets allocated to nearest cluster, while keep centroid as small as possible
- Popular un-supervised Machine Learning, helps to solve this problem
- 5 clusters created based on frequency of occurrence of “Food court”.
- Code with screenshot



```
# set number of clusters
kclusters = 5

kl_clustering = Hyd_Food_Court.drop(["Neighborhoods"], 1)

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

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

```
array([0, 1, 0, 4, 1, 4, 0, 0, 3, 0], dtype=int32)
```



```
In [38]: # create a new dataframe that includes the cluster as well as the top 10 venues for each neighborhood.
         Hyd_Food_Court_merged = Hyd_Food_Court.copy()

         # add clustering labels
         Hyd_Food_Court_merged["Cluster Labels"] = kmeans.labels_
```

```
In [39]: Hyd_Food_Court_merged.rename(columns={"Neighborhoods": "Neighborhood"}, inplace=True)
         Hyd_Food_Court_merged.head()
```

Out[39]:

	Neighborhood	Food Court	Cluster Labels
0	A. C. Guards	0.000000	0
1	A. S. Rao Nagar	0.045455	1
2	Abhyudaya Nagar	0.000000	0
3	Abids	0.013889	4
4	Adikmet	0.047619	1

```
In [40]: # merge Hyd_Food_court_grouped with Hyd_df to add latitude/longitude for each neighborhood

         Hyd_Food_Court_final = pd.merge(Hyd_Food_Court_merged, Hyd_df)

         print(Hyd_Food_Court_final.shape)
         Hyd_Food_Court_final.head()
```

```
(196, 5)
```

Out[40]:

	Neighborhood	Food Court	Cluster Labels	Latitude	Longitude
0	A. C. Guards	0.000000	0	17.395015	78.459812
1	A. S. Rao Nagar	0.045455	1	17.411200	78.508240

3.5 Concluding the exploration, based on cluster properties related to frequency of Food court occurrence

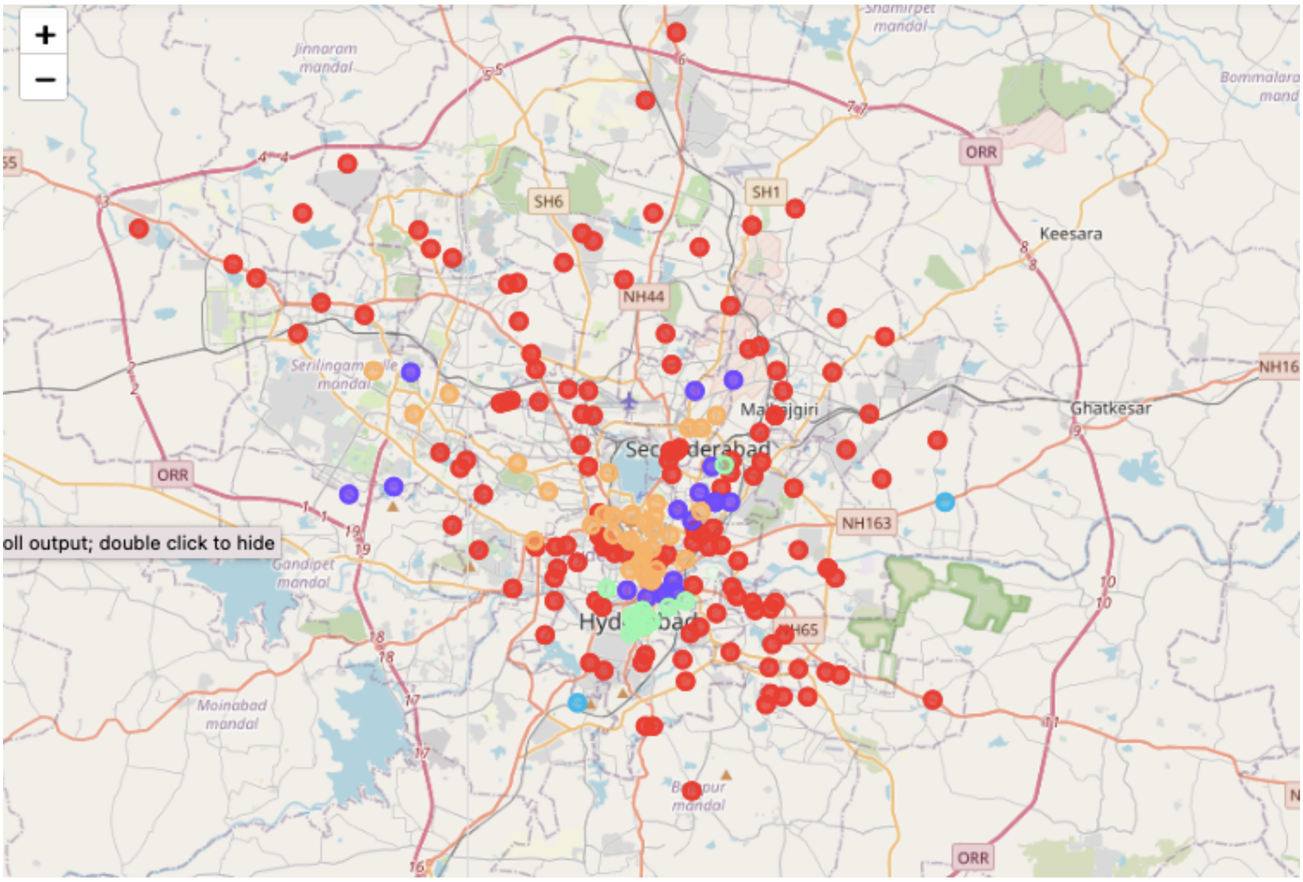
- 5 clusters provides indication of Food Court presence in those clusters.
- The places where limited or zero presence are suggestive places, whereas places with high density of Food courts presence should be avoided.

4. Results

- The neighborhoods of Hyderabad is clustered in 5 Clusters, based on frequency of occurrence of Food Court.

1. Cluster 0 (Red dots in map)- Neighborhood with **zero or minimum** number of Food court
2. Cluster 1 (Violet dots in map)- Neighborhood with **moderate** presence of Food court
3. Cluster 2 (Turquoise dots in map)- Neighborhood with **limited** number of Food court
4. Cluster 3 (Green dots in map)- Neighborhood with **high density** of Food court
5. Cluster 4 (Amber dots in map)-- Neighborhood with **moderate** presence of Food court

- Based on the this above depiction, it is observed there is scope of having good Food court Business in Cluster 0, followed by Cluster 2.
- However this is an exploratory research, and there are other associated factors to take conclusion



5. Discussion & Limitation

It is observed that most of the food courts are located towards center of the city, whereas frequency is less towards outskirts. There is a potential for opening Food court in quite a number of places as Cluster zero is having largest number of Neighborhoods. This can be a lucrative option.

However there is a limitation of this study. This is an exploratory study, where objective is to identify neighborhoods, where there is a scope of building Food court. There should be other study conducted before concluding on single location. This is study includes demographics division, population density, spending capability etc. Also standalone restaurant will indirectly compete with Food court. The frequency of occurrence Restaurants also need to be studied

6. Conclusion

In this project we have Identified a Business Problem for Real Estate investors / Food court joints, and address it with suggestive action.
We have identified source of data, extracted data, prepare the data, and perform Clustering of data.
We have concluded with result, provided appropriate solution and suggesting further scope of study.
Business Problem - To locate certain areas in Hyderabad, which is having a huge potential for Food court Business with good foot fall
Result- It is observed there is scope of having good Food court Business in Cluster 0, followed by Cluster 2.