Capstone Project-The-Battle-of-Neighborhoods|

Exploring Chennai, Tamil Nadu, India

Table of contents:

1. Introduction:

Chennai, the capital city of Tamil Nadu- India, attracts many visitors either as tourists or as part of its large workforce. The vast majority claim Chennai is one of the best cities in India. We know that Chennai is popular for IIT and for being an IT and industrial hub. The city is also renowned for its passion for music. But, there's more to the city than you think. From its varied culture and tradition, vibrant festivals, dainty delicacies to its quintessential "Chennai Tamil", this city doesn't fail to mesmerize the locals and the outsiders living here.

1.1. Business Problem:

The expectation of visitors to Chennai could be stated as follows:
a. What are the local food/ native cuisine available from restaurants in and around Chennai?

b. What services or value addition does the stop-over at a restaurant bring him, other than enjoying good food?

1.2 Target Audience:

1.2.1 The goal of this exercise is to give a simple recommendation to visitors of Chennai, Tamil Nadu: in which area they will find a large number or concentration of which types of restaurants.

1.2.2 The target audience are investors who would like to start a group or chain of restaurants in and around Chennai. This analysis will give an idea, which area is crowded with restaurants and where is it beneficial to open a restaurant around Chennai.

1.2.3 Road Travelers, to find reasonable refreshment joint where they can dine and also get along to refresh themselves in an amusement park – rest and refresh during their long road trip.

2. Data

I will use foursquare API to collect data about restaurants in Chennai. I need data about different venues around Chennai.

In order to gain that information we will use “Foursquare” locational information. Foursquare is a location data provider with information about all manner of venues and events within an area of interest. Such information includes venue names, locations, menus and even photos. As such, the foursquare location platform will be used as the sole data source since all the stated required information can be obtained through the API.

After finding the list of neighborhoods, we then connect to the Foursquare API to gather information about venues inside each and every neighborhood.

The data retrieved from Foursquare contained information of venues within a specified distance of the longitude and latitude of the postcodes.

Interactive maps are useful for data exploration and communicating research. Folium package will be used to:

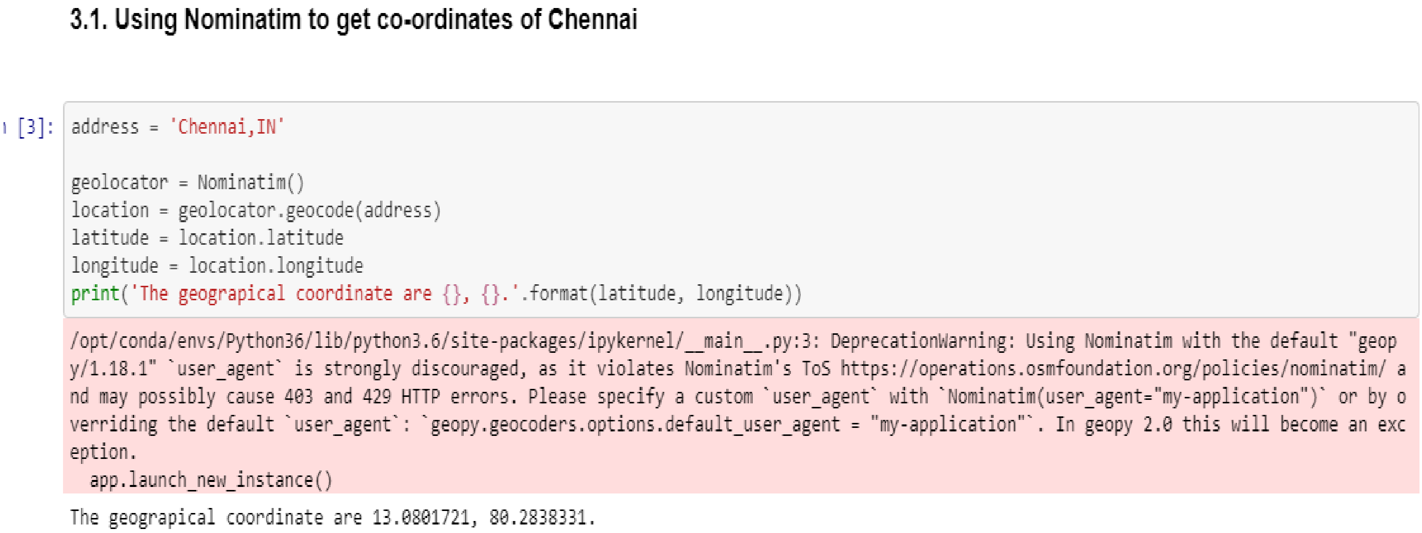
a. Create a map centered at an inputted location

b. Create marker on the map

Markers can be extremely useful for storing information about locations on the map such as cross streets, building information, etc.

3. Methodology - Data Visualization and Exploration

3.1 Used Nominatim to get co-ordinates of Chennai using geocoder

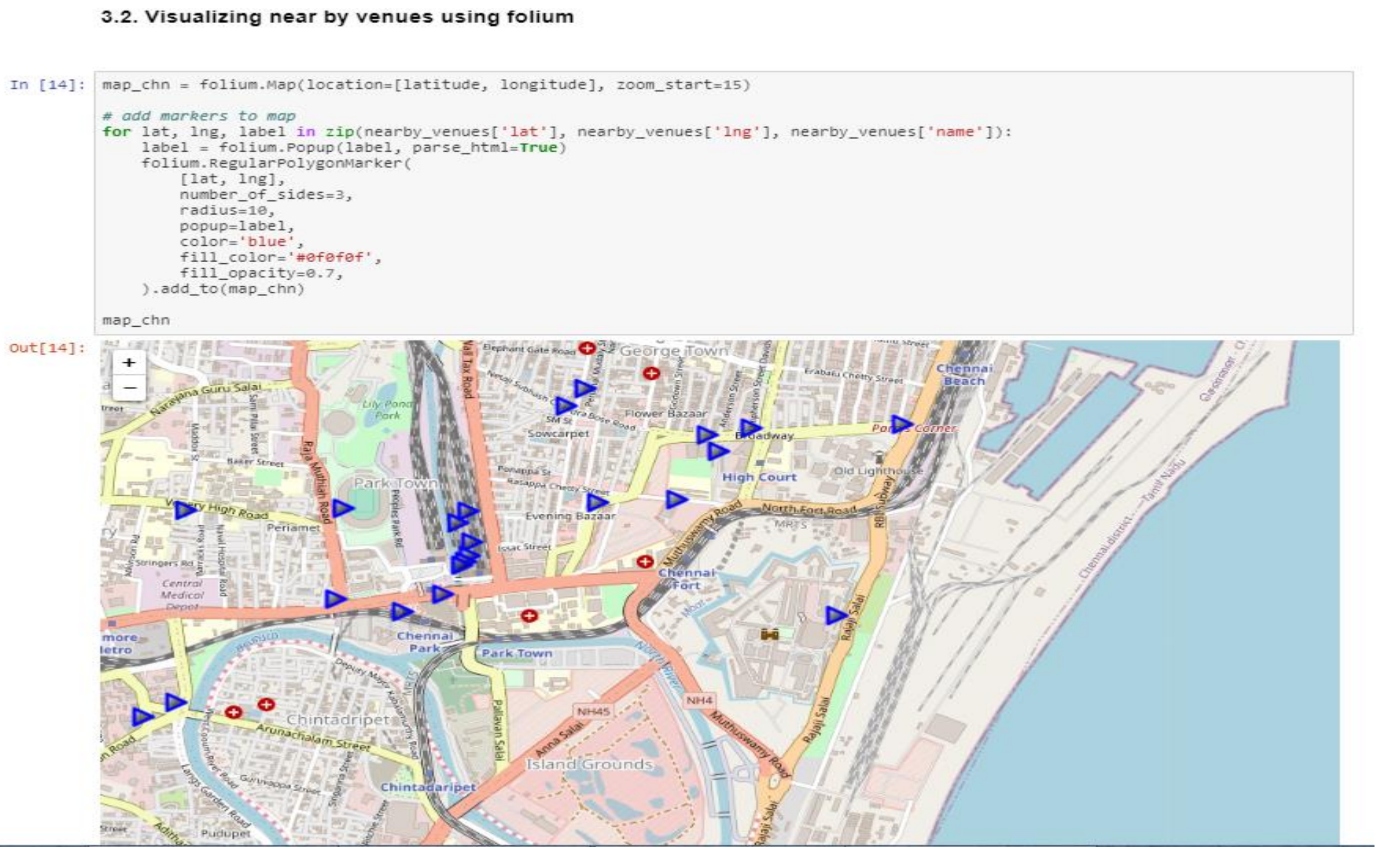


3.1. Using Nominatim to get co-ordinates of Chennai

```
[3]: address = 'Chennai,IN'

     geolocator = Nominatim()
     location = geolocator.geocode(address)
     latitude = location.latitude
     longitude = location.longitude
     print('The geograpical coordinate are {}, {}.'.format(latitude, longitude))
```

```
/opt/conda/envs/Python36/lib/python3.6/site-packages/ipykernel/__main__.py:3: DeprecationWarning: Using Nominatim with the default "geop
y/1.18.1" `user_agent` is strongly discouraged, as it violates Nominatim's ToS https://operations.osmfoundation.org/policies/nominatim/ a
nd may possibly cause 403 and 429 HTTP errors. Please specify a custom `user_agent` with `Nominatim(user_agent="my-application")` or by o
verriding the default `user_agent`: `geopy.geocoders.options.default_user_agent = "my-application"`. In geopy 2.0 this will become an exc
eption.
  app.launch_new_instance()
The geograpical coordinate are 13.0801721, 80.2838331.
```

3.2 Used Folium package to visualize the near-by venues.

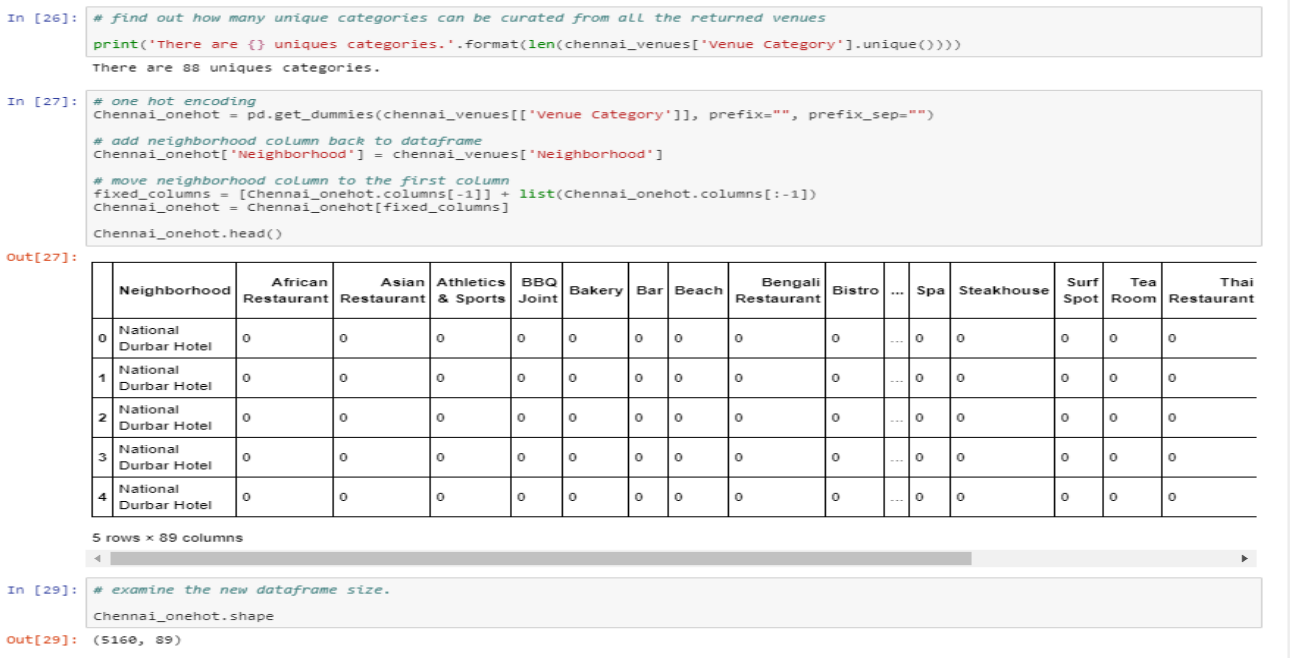


3.2. Visualizing near by venues using folium

In [14]:
```python
map_chn = folium.Map(location=[latitude, longitude], zoom_start=15)

# add markers to map
for lat, lng, label in zip(nearby_venues['lat'], nearby_venues['lng'], nearby_venues['name']):
    label = folium.Popup(label, parse_html=True)
    folium.RegularPolygonMarker(
        [lat, lng],
        number_of_sides=3,
        radius=10,
        popup=label,
        color='blue',
        fill_color='#0f0f0f',
        fill_opacity=0.7,
    ).add_to(map_chn)

map_chn
```

Out[14]:

3.3 Used one hot coding to group neighborhood by venue categories. Retrieved the most common venue by the frequency.



In [26]:
```python
# find out how many unique categories can be curated from all the returned venues

print('There are {} uniques categories.'.format(len(chennai_venues['Venue Category'].unique())))
```

```
There are 88 uniques categories.
```

In [27]:
```python
# one hot encoding
Chennai_onehot = pd.get_dummies(chennai_venues[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
Chennai_onehot['Neighborhood'] = chennai_venues['Neighborhood']

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

Chennai_onehot.head()
```

Out[27]:

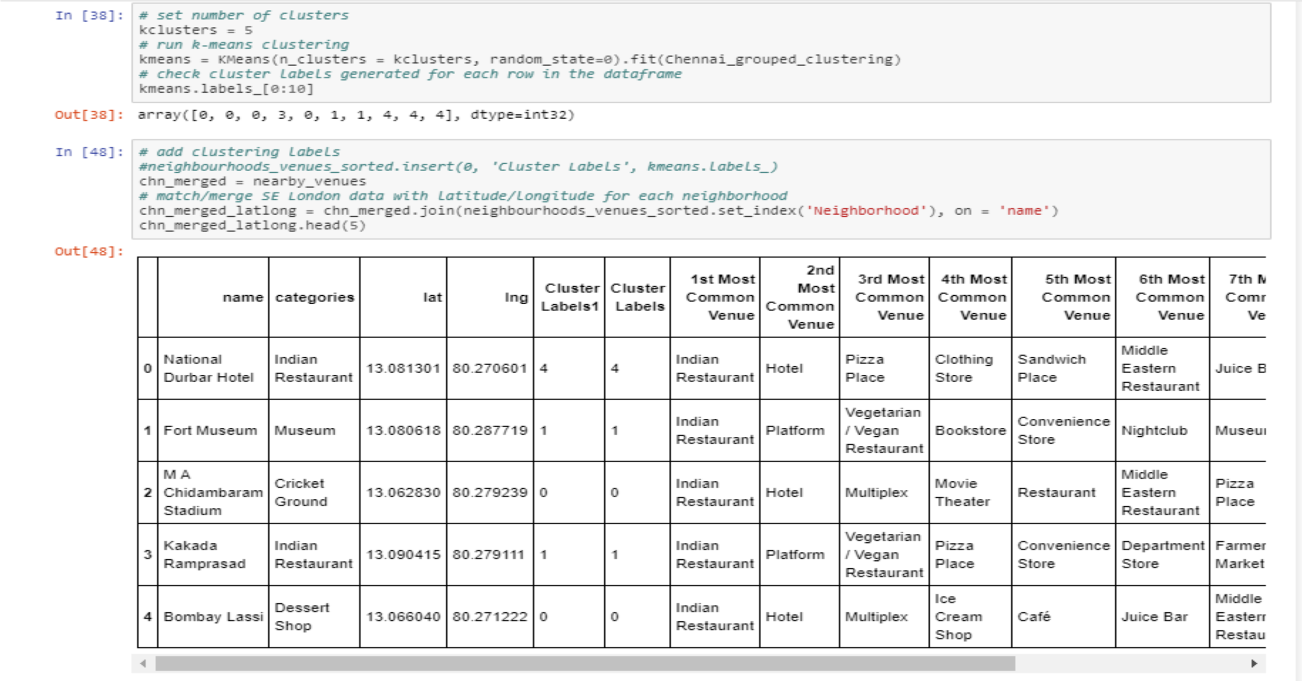
	Neighborhood	African Restaurant	Asian Restaurant	Athletics & Sports	BBQ Joint	Bakery	Bar	Beach	Bengali Restaurant	Bistro	...	Spa	Steakhouse	Surf Spot	Tea Room	Thai Restaurant
0	National Durbar Hotel	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
1	National Durbar Hotel	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
2	National Durbar Hotel	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
3	National Durbar Hotel	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
4	National Durbar Hotel	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0

5 rows × 89 columns

In [29]:
```python
# examine the new dataframe size.

Chennai_onehot.shape
```

Out[29]:
```
(5160, 89)
```

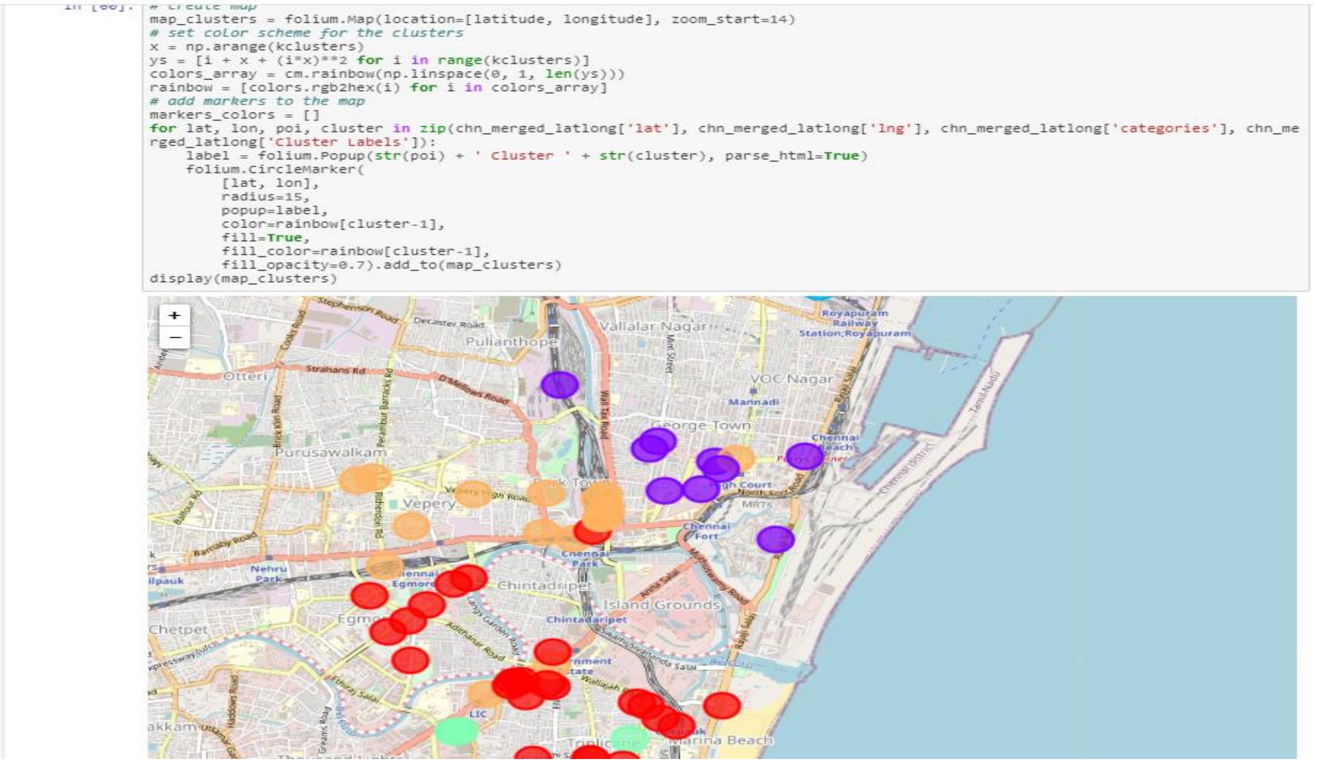
3.4 Used Kclusters to group the neighborhood by venues categories



```
In [38]: # set number of clusters
         kclusters = 5
         # run k-means clustering
         kmeans = KMeans(n_clusters = kclusters, random_state=0).fit(Chennai_grouped_clustering)
         # check cluster labels generated for each row in the dataframe
         kmeans.labels_[0:10]

Out[38]: array([0, 0, 0, 3, 0, 1, 1, 4, 4, 4], dtype=int32)

In [48]: # add clustering labels
         #neighbourhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)
         chn_merged = nearby_venues
         # match/merge SE London data with latitude/longitude for each neighborhood
         chn_merged_latlong = chn_merged.join(neighbourhoods_venues_sorted.set_index('Neighborhood'), on = 'name')
         chn_merged_latlong.head(5)

Out[48]:
```

	name	categories	lat	lng	Cluster Labels1	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th M Comr Ve
0	National Durbar Hotel	Indian Restaurant	13.081301	80.270601	4	4	Indian Restaurant	Hotel	Pizza Place	Clothing Store	Sandwich Place	Middle Eastern Restaurant	Juice B
1	Fort Museum	Museum	13.080618	80.287719	1	1	Indian Restaurant	Platform	Vegetarian / Vegan Restaurant	Bookstore	Convenience Store	Nightclub	Museu
2	M A Chidambaram Stadium	Cricket Ground	13.062830	80.279239	0	0	Indian Restaurant	Hotel	Multiplex	Movie Theater	Restaurant	Middle Eastern Restaurant	Pizza Place
3	Kakada Ramprasad	Indian Restaurant	13.090415	80.279111	1	1	Indian Restaurant	Platform	Vegetarian / Vegan Restaurant	Pizza Place	Convenience Store	Department Store	Farmer Market
4	Bombay Lassi	Dessert Shop	13.066040	80.271222	0	0	Indian Restaurant	Hotel	Multiplex	Ice Cream Shop	Café	Juice Bar	Middle Easterr Restau

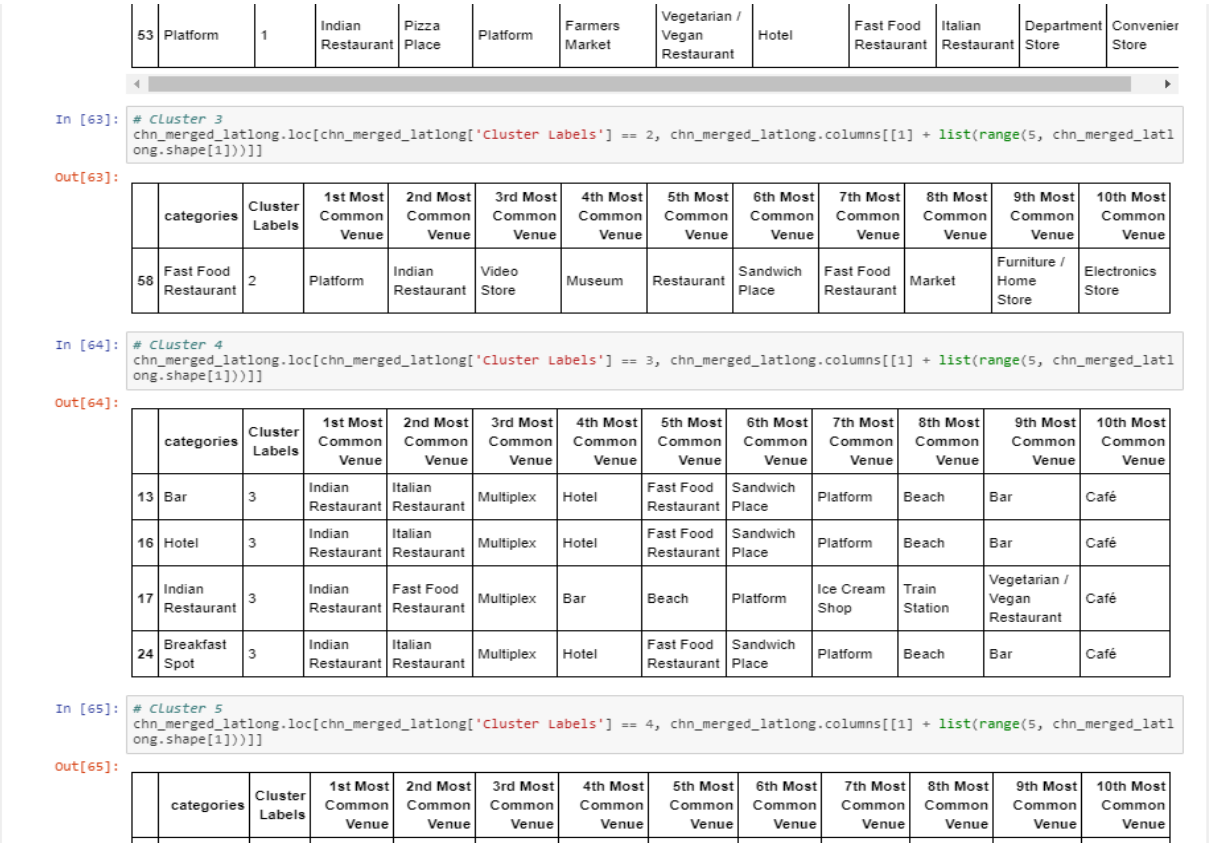
3.5 Folium package is used to visualize the clusters



```
In [60]: # create map
         map_clusters = folium.Map(location=[latitude, longitude], zoom_start=14)
         # set color scheme for the clusters
         x = np.arange(kclusters)
         ys = [i + x + (i*x)**2 for i in range(kclusters)]
         colors_array = cm.rainbow(np.linspace(0, 1, len(ys)))
         rainbow = [colors.rgb2hex(i) for i in colors_array]
         # add markers to the map
         markers_colors = []
         for lat, lon, poi, cluster in zip(chn_merged_latlong['lat'], chn_merged_latlong['lng'], chn_merged_latlong['categories'], chn_me
         rged_latlong['Cluster Labels']):
             label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
             folium.CircleMarker(
                 [lat, lon],
                 radius=15,
                 popup=label,
                 color=rainbow[cluster-1],
                 fill=True,
                 fill_color=rainbow[cluster-1],
                 fill_opacity=0.7).add_to(map_clusters)
         display(map_clusters)
```

4. Results

Snapshot of the cluster is shown below:



53	Platform	1	Indian Restaurant	Pizza Place	Platform	Farmers Market	Vegetarian / Vegan Restaurant	Hotel	Fast Food Restaurant	Italian Restaurant	Department Store	Convenier Store

```
In [63]: # Cluster 3
         chn_merged_latlong.loc[chn_merged_latlong['Cluster Labels'] == 2, chn_merged_latlong.columns[[1] + list(range(5, chn_merged_latl
         ong.shape[1]))]]
```

Out[63]:

	categories	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
58	Fast Food Restaurant	2	Platform	Indian Restaurant	Video Store	Museum	Restaurant	Sandwich Place	Fast Food Restaurant	Market	Furniture / Home Store	Electronics Store

```
In [64]: # Cluster 4
         chn_merged_latlong.loc[chn_merged_latlong['Cluster Labels'] == 3, chn_merged_latlong.columns[[1] + list(range(5, chn_merged_latl
         ong.shape[1]))]]
```

Out[64]:

	categories	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
13	Bar	3	Indian Restaurant	Italian Restaurant	Multiplex	Hotel	Fast Food Restaurant	Sandwich Place	Platform	Beach	Bar	Café
16	Hotel	3	Indian Restaurant	Italian Restaurant	Multiplex	Hotel	Fast Food Restaurant	Sandwich Place	Platform	Beach	Bar	Café
17	Indian Restaurant	3	Indian Restaurant	Fast Food Restaurant	Multiplex	Bar	Beach	Platform	Ice Cream Shop	Train Station	Vegetarian / Vegan Restaurant	Café
24	Breakfast Spot	3	Indian Restaurant	Italian Restaurant	Multiplex	Hotel	Fast Food Restaurant	Sandwich Place	Platform	Beach	Bar	Café

```
In [65]: # Cluster 5
         chn_merged_latlong.loc[chn_merged_latlong['Cluster Labels'] == 4, chn_merged_latlong.columns[[1] + list(range(5, chn_merged_latl
         ong.shape[1]))]]
```

Out[65]:

	categories	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue

The following are derived based on the results of 5 clusters:

1. As discussed in the introduction, data also supports the fact that Chennai is famous for food and movie.

2. Although, the Clusters have variations, the most common venue is the Indian Restaurants.

5. Discussion and Conclusion

Visitors of Chennai can enjoy Indian cuisine in all the 5 clusters. If the visitor is interested in visiting other venues other than restaurant then cluster 3 is the best suited to enjoy movie with food.