Premium Fitness Centre Location Recommender (Using K-Means)

Introduction- Business Problem

The problem at hand is to find an optimal location for a restaurant. Specifically, this report will be targeted to stakeholders interested in opening a Premium Fitness Centre or a Franchise Gym in Bangalore, India. Finding a suitable location for gym in major cities like Bangalore proves to be a daunting task. Various factors such as over-saturation or no demand ,for the gym/fitness centre that the customer wants to open, effect the success or failure of the centre. Hence, prospective investors can bolster their decisions using the descriptive and predictive capabilities of data science.We need to find locations (Neighbourhood) that have a potentially unfulfilled demand for a Premium Fitness Centre like Golds, Cult, Icon Fitness…etc. Also, we need locations that have a relatively higher Average Income and population. We would also prefer location as close to popular city Neighbourhood, assuming the first two conditions are met. We will use our data science powers to generate a few most promising neighbourhoods based on this criteria. Advantages of each area will then be clearly Expressed so that best possible final location can be chosen by stakeholders.

Data Acquisition and Preparation

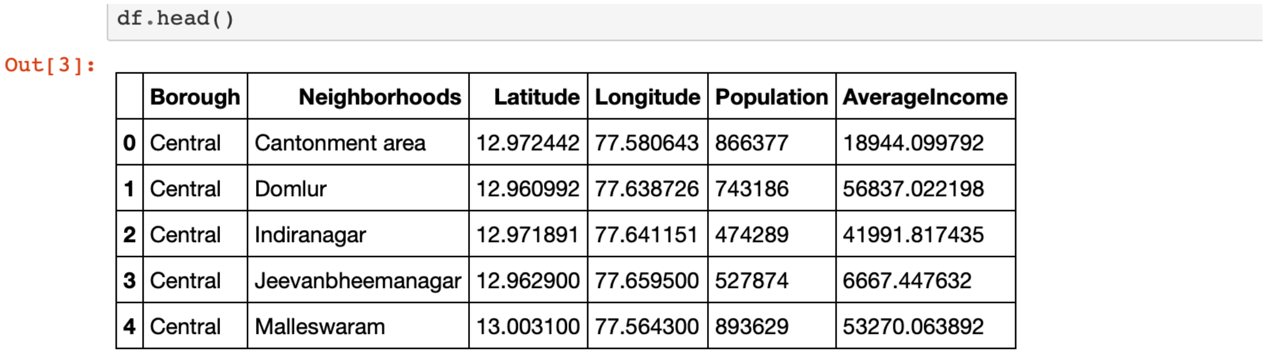
Based on definition of our problem, factors that will influence our decision are:

- Number of existing gyms in the neighbourhood or nearby neighbourhood
- Population of the neighbourhood
- Average Income of the neighbourhood

In our Project we will :

- Acquire the names and boroughs of the neighbourhoods by scrapping a wikipedia page.
- After we have got the names of all the neighbourhoods, we will geocode them using the library geopy.geocoder (Nominatim) to get the latitude, longitude of each neighbourhood.
- Use Kaggle and internet to get Population data of each neighbourhood of Bangalore.
- Similarly we will get Average Income of each Neighbourhood.

After we combine and merge all data into one DataFrame, we have to clean it, process it to make it useful for Data Understanding and further processes ahead. So here is a sample of how our clean DataSet looks like on which we going to work.



```
df.head()
```

Out[3]:

	Borough	Neighborhoods	Latitude	Longitude	Population	AverageIncome
0	Central	Cantonment area	12.972442	77.580643	866377	18944.099792
1	Central	Domlur	12.960992	77.638726	743186	56837.022198
2	Central	Indiranagar	12.971891	77.641151	474289	41991.817435
3	Central	Jeevanbheemanagar	12.962900	77.659500	527874	6667.447632
4	Central	Malleswaram	13.003100	77.564300	893629	53270.063892

Top 5 rows of the Dataframe displayed

Data Understanding

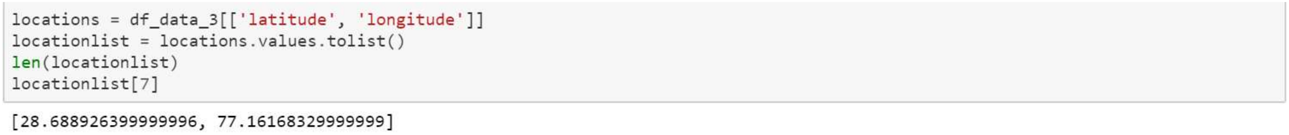
Boroughs vs Number of Neighborhoods

```
In [6]: df_1 = df[['Borough','Neighborhoods']]
        df_grp1 = df_1.groupby(['Borough'], as_index = False).count() #arranging no of neighborhoods and boroughs
        df_grp1
```

Out[6]:

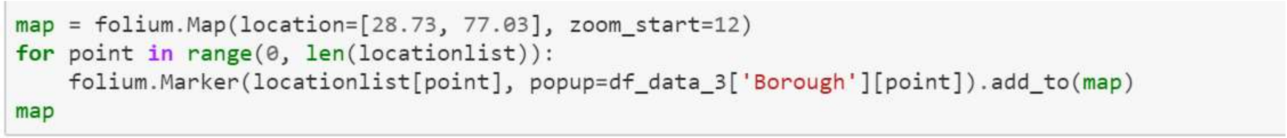
	Borough	Neighborhoods
0	Central	12
1	Eastern	8
2	NorthEastern	6
3	Northern	8
4	SouthEastern	7
5	Southern	8
6	SouthernSuburbs	6
7	Western	9

? Visualising the obtained dataset

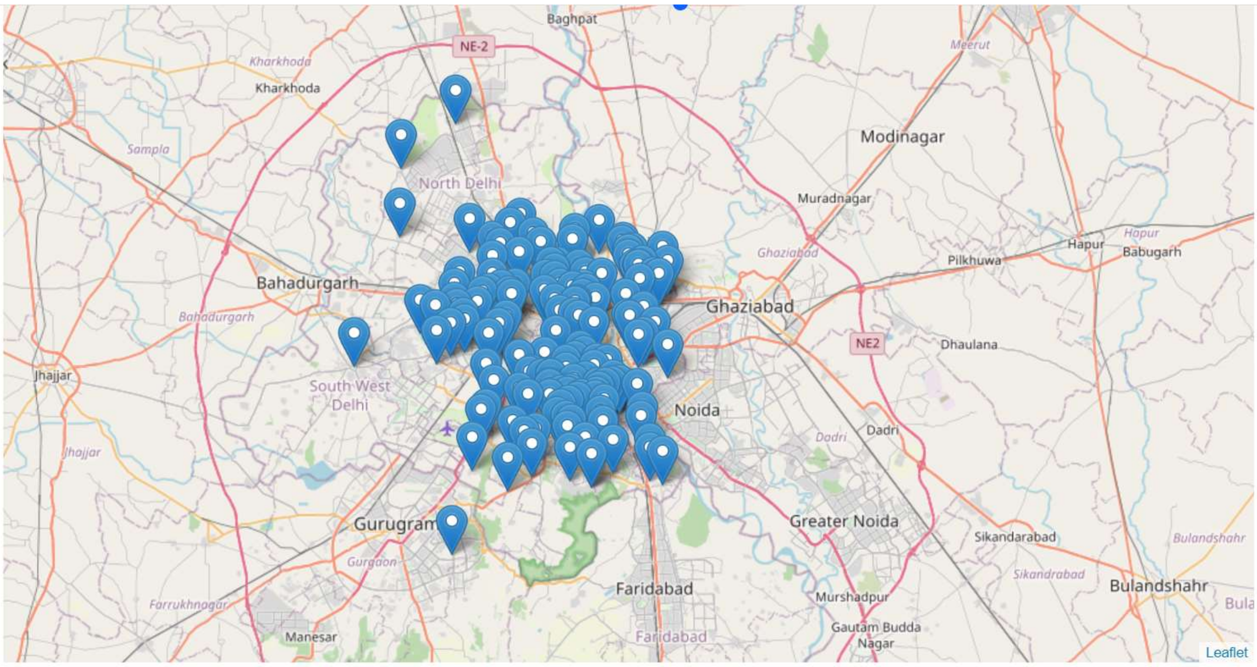


```
locations = df_data_3[['latitude', 'longitude']]
locationlist = locations.values.tolist()
len(locationlist)
locationlist[7]
```

```
[28.688926399999996, 77.16168329999999]
```

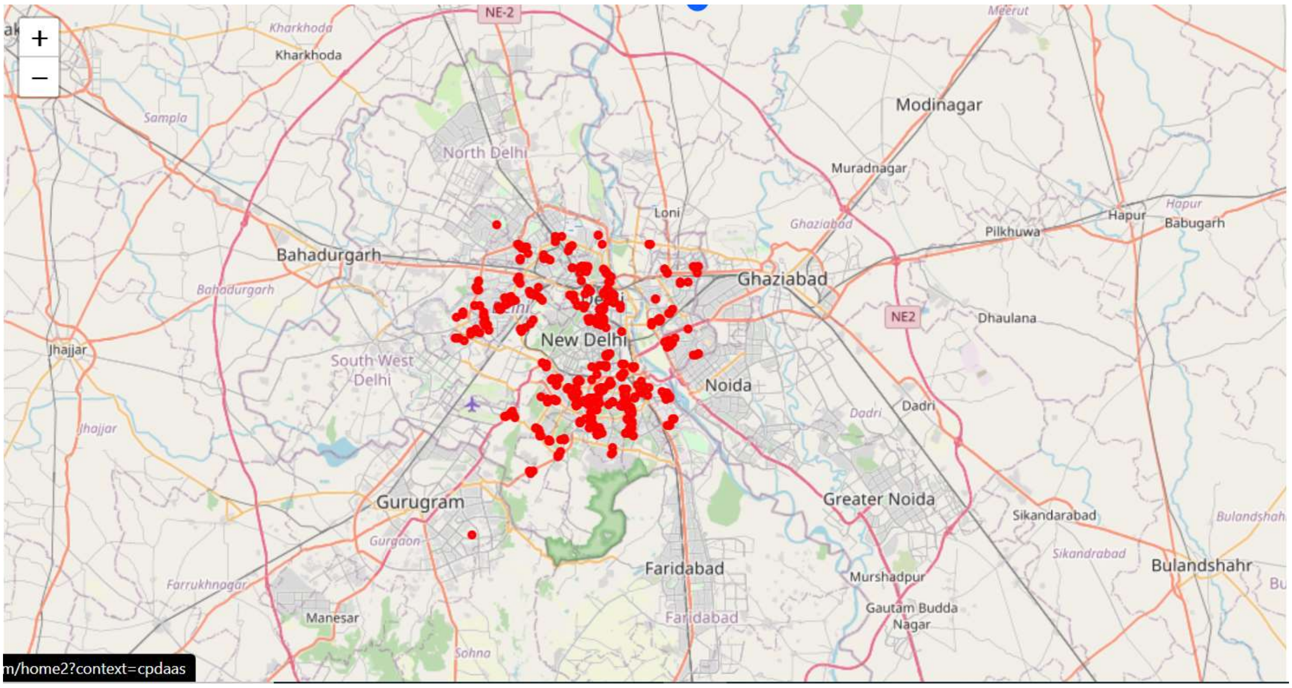


```
map = folium.Map(location=[28.73, 77.03], zoom_start=12)
for point in range(0, len(locationlist)):
    folium.Marker(locationlist[point], popup=df_data_3['Borough'][point]).add_to(map)
map
```



Now that we had marked all the concerned neighbourhoods, we will use foursquare api to locate the nearby restaurants.

Here we have used the explore API call and filtered the search to find venues designated as restaurants.



Clustering and Analysis

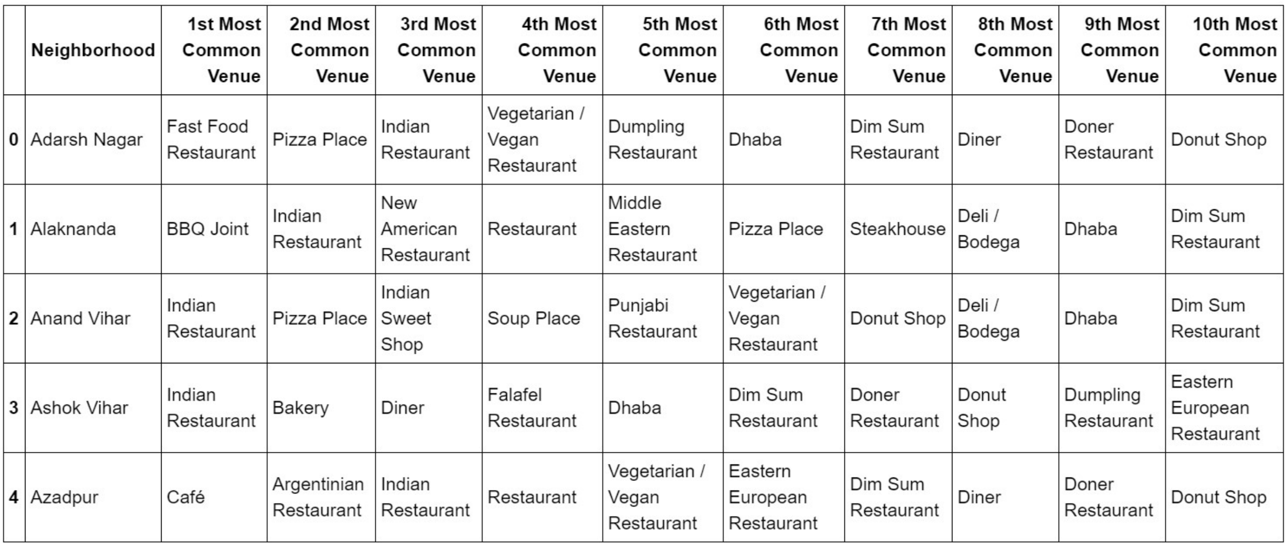
The task at hand was not only to find the neighbourhoods with low density of Indian restaurants, but also the ones which have the potential for growth.

So we searched for neighbourhoods with similar pattern of restaurant trends.

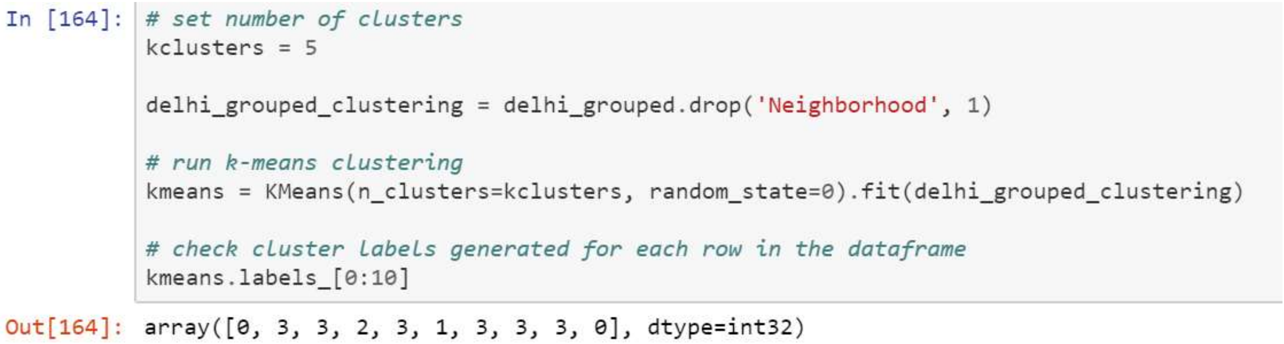
This was achieved by clustering of neighbourhoods on the basis of restaurant data we had acquired. Clustering is a predominant algorithm of unsupervised Machine Learning. It is used to segregate data entries in cluster depending of the similarity of their attributes, calculated by using the simple formula of Euclidian distance.

We then analysed these clusters separately and used those clusters that showed high trends of Indian restaurants.

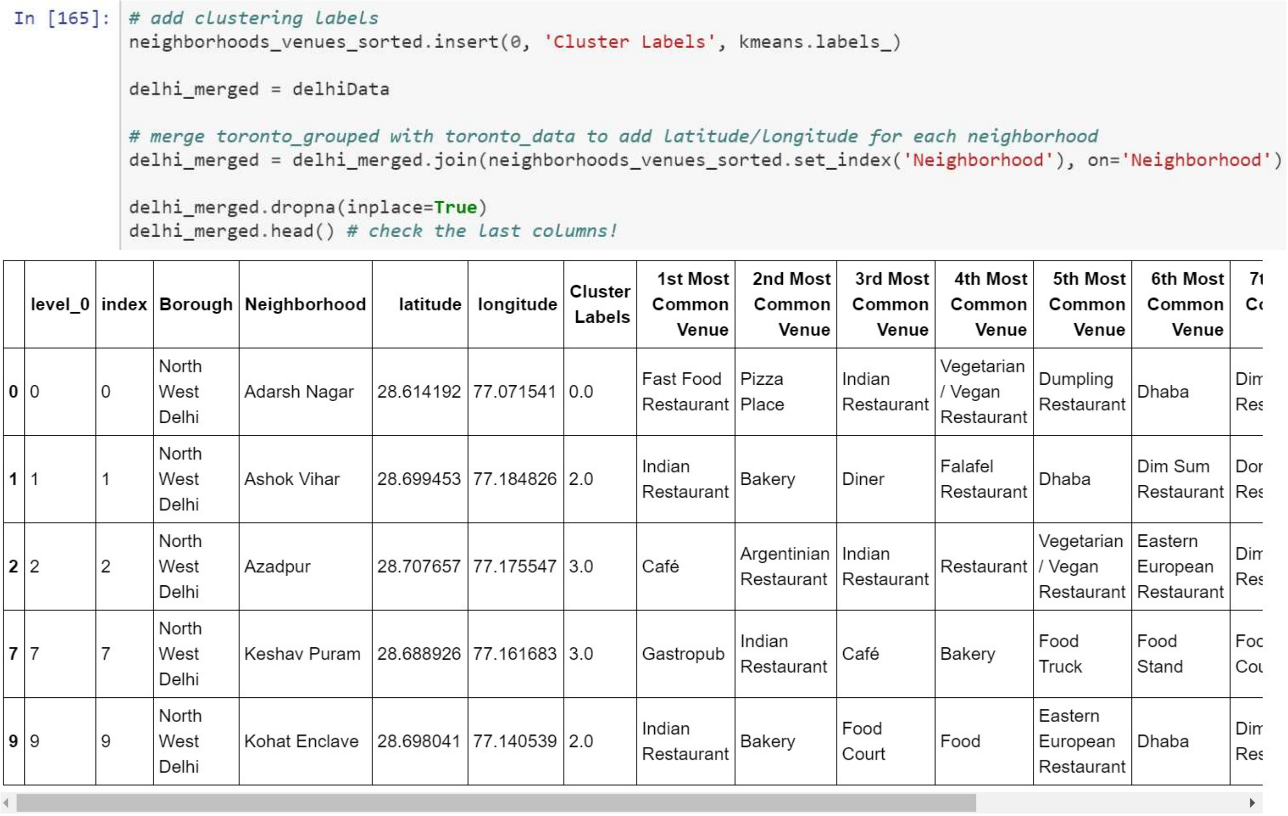
Applying the clustering algorithm



	Neighborhood	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
0	Adarsh Nagar	Fast Food Restaurant	Pizza Place	Indian Restaurant	Vegetarian / Vegan Restaurant	Dumpling Restaurant	Dhaba	Dim Sum Restaurant	Diner	Doner Restaurant	Donut Shop
1	Alaknanda	BBQ Joint	Indian Restaurant	New American Restaurant	Restaurant	Middle Eastern Restaurant	Pizza Place	Steakhouse	Deli / Bodega	Dhaba	Dim Sum Restaurant
2	Anand Vihar	Indian Restaurant	Pizza Place	Indian Sweet Shop	Soup Place	Punjabi Restaurant	Vegetarian / Vegan Restaurant	Donut Shop	Deli / Bodega	Dhaba	Dim Sum Restaurant
3	Ashok Vihar	Indian Restaurant	Bakery	Diner	Falafel Restaurant	Dhaba	Dim Sum Restaurant	Doner Restaurant	Donut Shop	Dumpling Restaurant	Eastern European Restaurant
4	Azadpur	Café	Argentinian Restaurant	Indian Restaurant	Restaurant	Vegetarian / Vegan Restaurant	Eastern European Restaurant	Dim Sum Restaurant	Diner	Doner Restaurant	Donut Shop



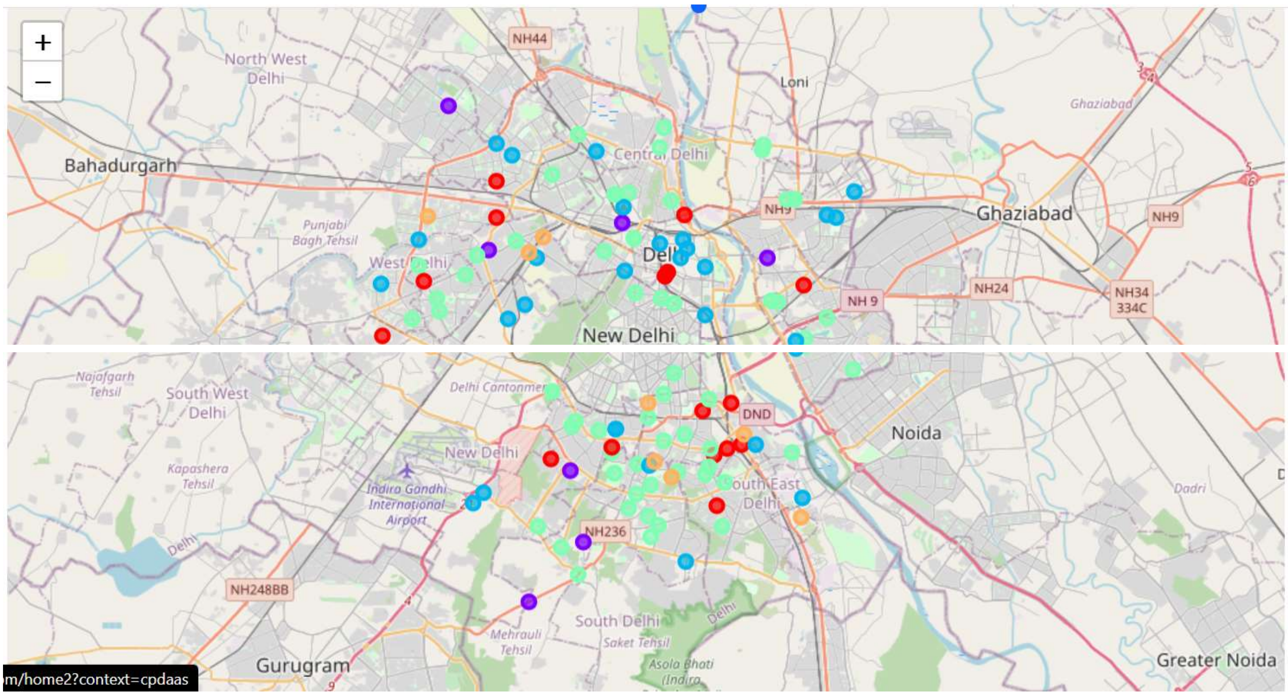
```
In [164]: # set number of clusters
          kclusters = 5

          delhi_grouped_clustering = delhi_grouped.drop('Neighborhood', 1)

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

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

Out[164]: array([0, 3, 3, 2, 3, 1, 3, 3, 3, 0], dtype=int32)
```



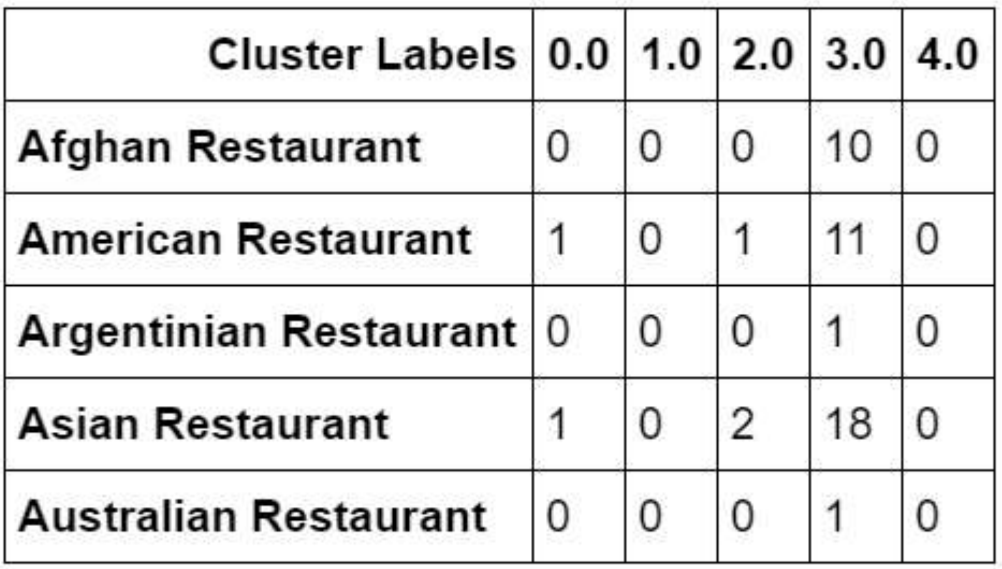
```
In [165]: # add clustering labels
          neighborhoods_venues_sorted.insert(0, 'Cluster Labels', kmeans.labels_)

          delhi_merged = delhiData

          # merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
          delhi_merged = delhi_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighborhood')

          delhi_merged.dropna(inplace=True)
          delhi_merged.head() # check the last columns!
```

	level_0	index	Borough	Neighborhood	latitude	longitude	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	7t Co
0	0	0	North West Delhi	Adarsh Nagar	28.614192	77.071541	0.0	Fast Food Restaurant	Pizza Place	Indian Restaurant	Vegetarian / Vegan Restaurant	Dumpling Restaurant	Dhaba	Dim Res
1	1	1	North West Delhi	Ashok Vihar	28.699453	77.184826	2.0	Indian Restaurant	Bakery	Diner	Falafel Restaurant	Dhaba	Dim Sum Restaurant	Dor Res
2	2	2	North West Delhi	Azadpur	28.707657	77.175547	3.0	Café	Argentinian Restaurant	Indian Restaurant	Restaurant	Vegetarian / Vegan Restaurant	Eastern European Restaurant	Dim Res
7	7	7	North West Delhi	Keshav Puram	28.688926	77.161683	3.0	Gastropub	Indian Restaurant	Café	Bakery	Food Truck	Food Stand	Foc Cou
9	9	9	North West Delhi	Kohat Enclave	28.698041	77.140539	2.0	Indian Restaurant	Bakery	Food Court	Food	Eastern European Restaurant	Dhaba	Dim Res

Cluster Visualisation



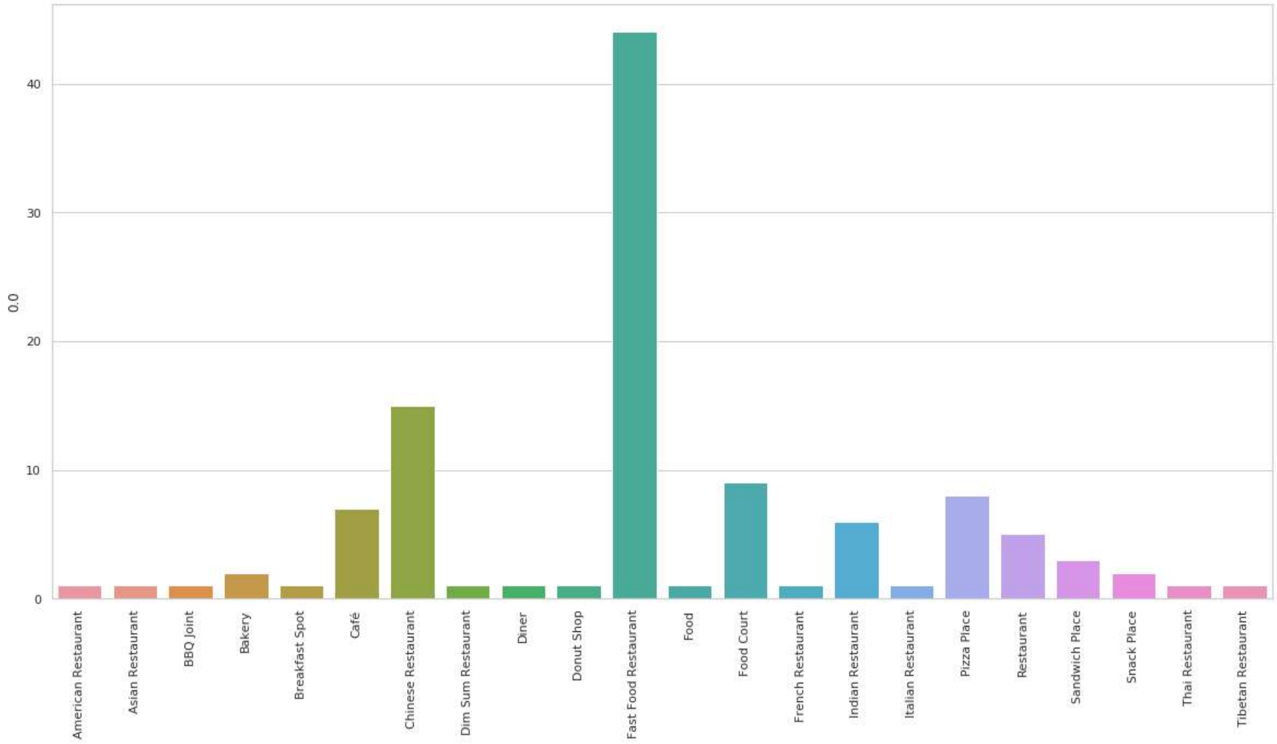
Out[167]:



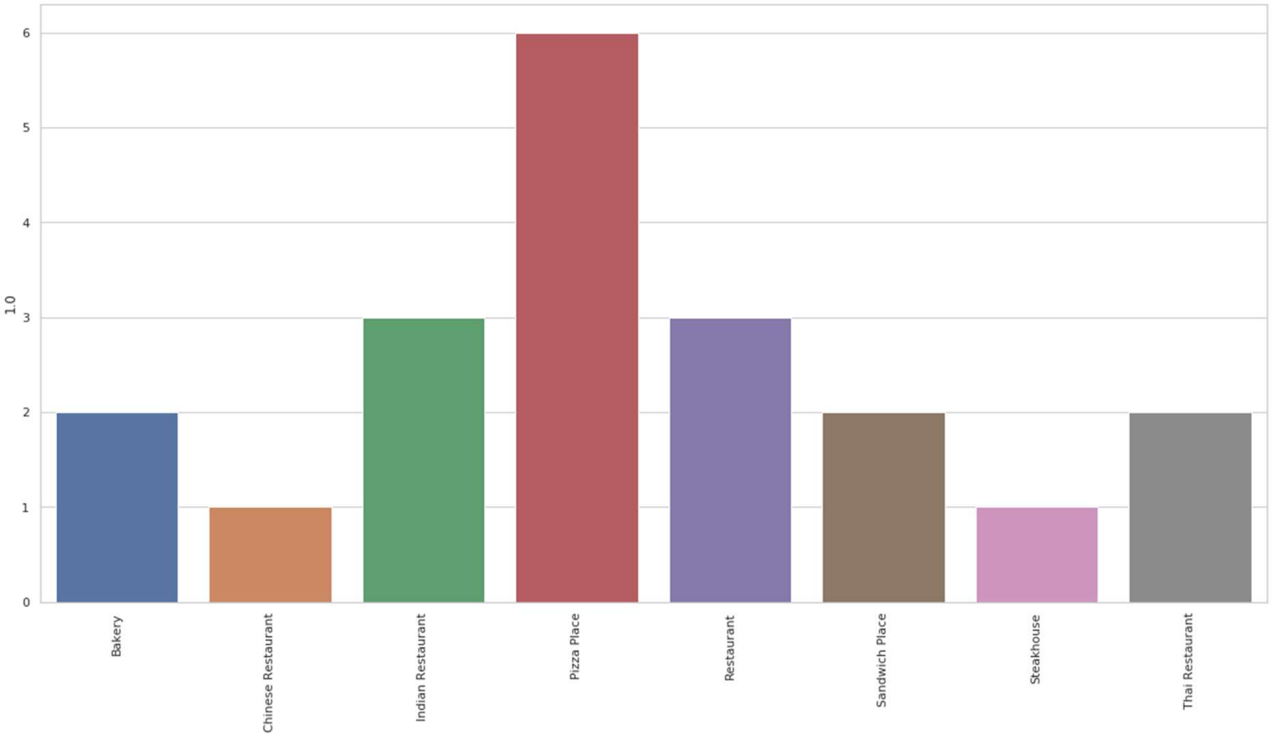
Cluster Labels	0.0	1.0	2.0	3.0	4.0
Afghan Restaurant	0	0	0	10	0
American Restaurant	1	0	1	11	0
Argentinian Restaurant	0	0	0	1	0
Asian Restaurant	1	0	2	18	0
Australian Restaurant	0	0	0	1	0

Analysing clusters

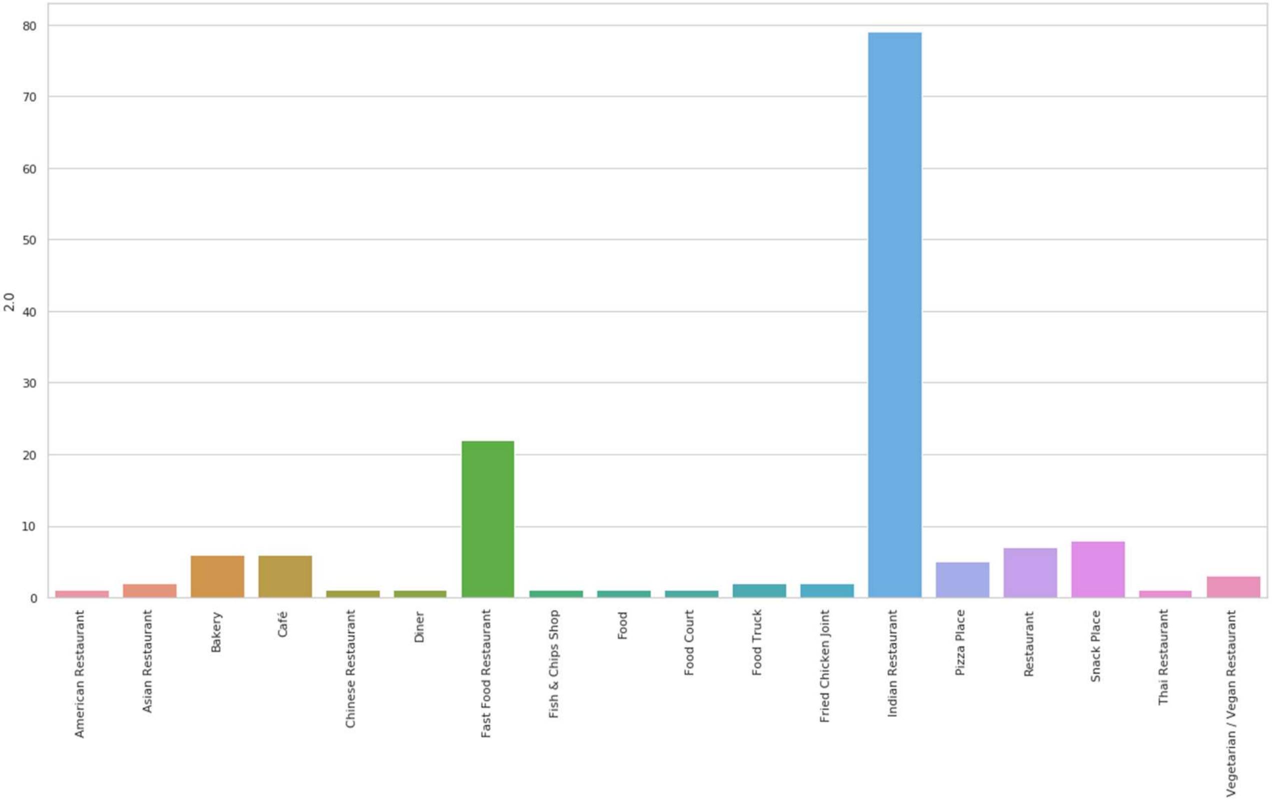
Plot_bar (0)



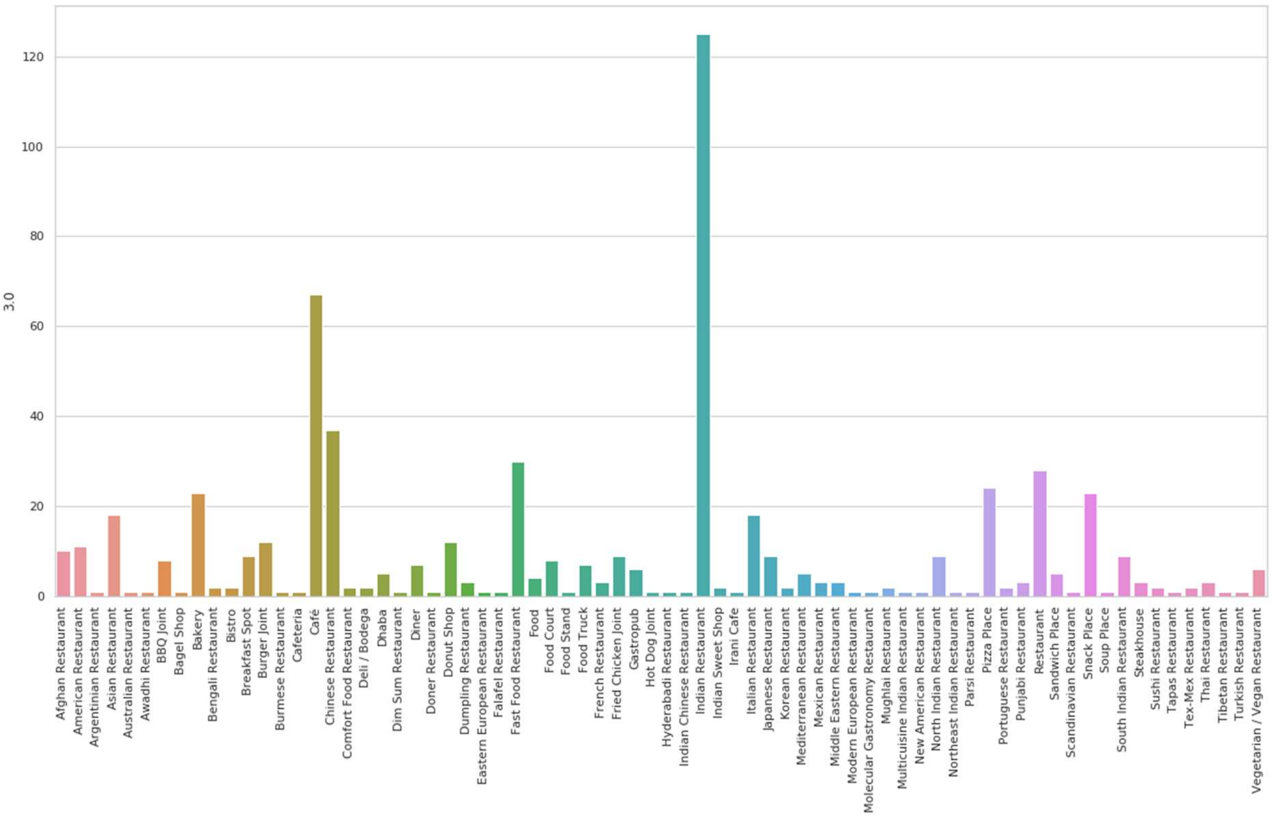
Plot_bar(1)



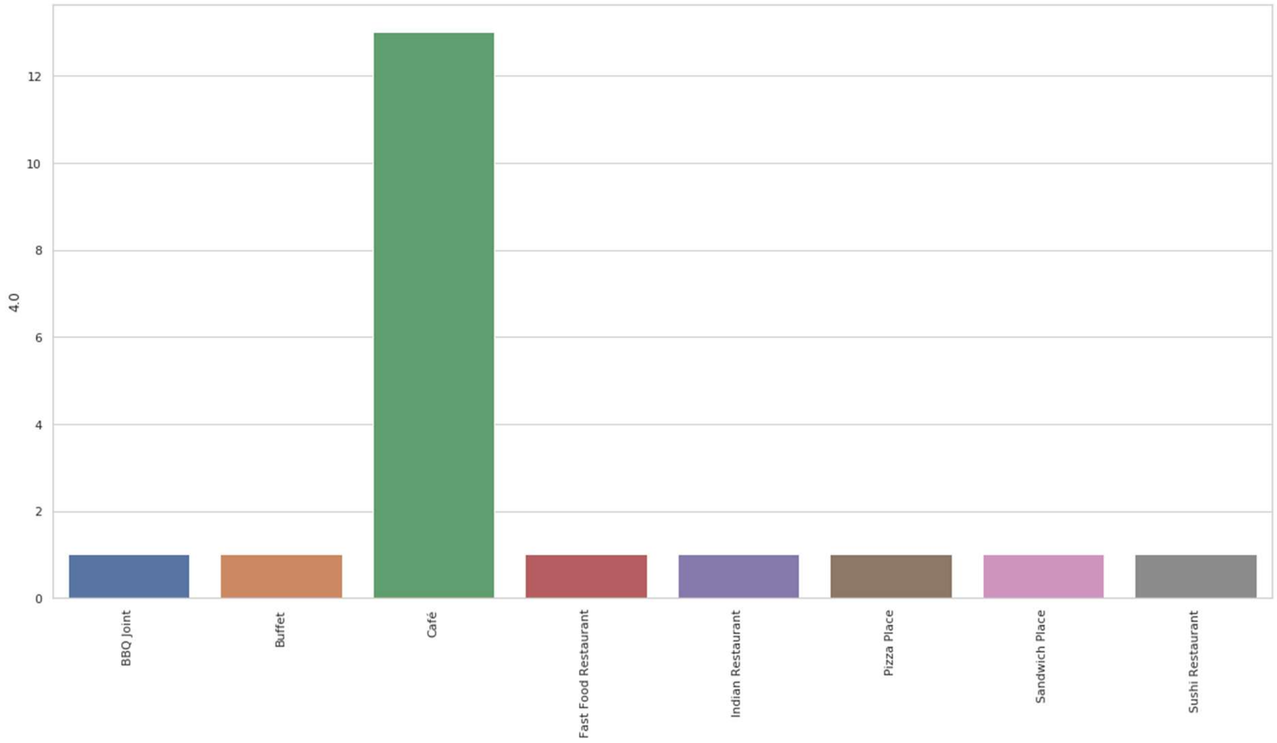
Plot_bar(2)



Plot_bar(3)



Plot_bar(4)



From the graphs it is evident that plots 1 and 2 have a high demand for Indian restaurants.

Recommendation

- ⍰ we have first, analysed the density of the Indian Restaurants in generally for each neighbourhood.

- ⍰ Then we eliminated the neighbourhoods that in the highest 70 percentile of density

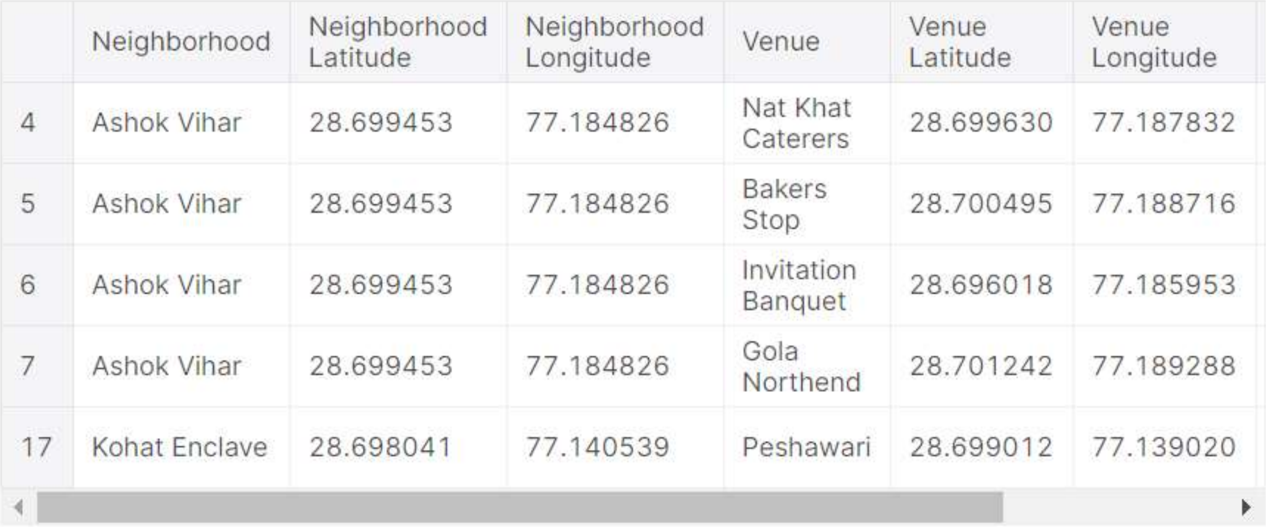
- ⍰ Found out the most popular neighbourhoods

- ⍰ Then tried to find remaining neighbourhoods that are close to them

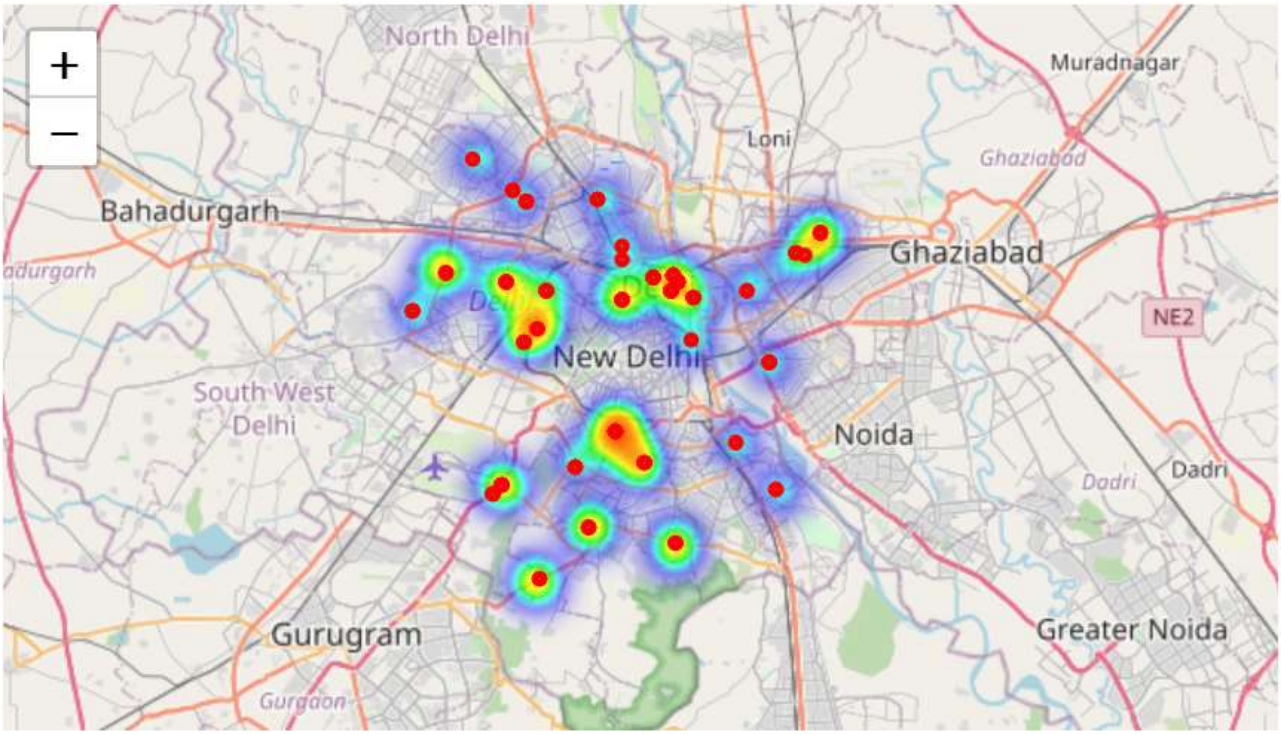
- ⍰ Provided the a detailed recommendation of top 10 neighbourhoods

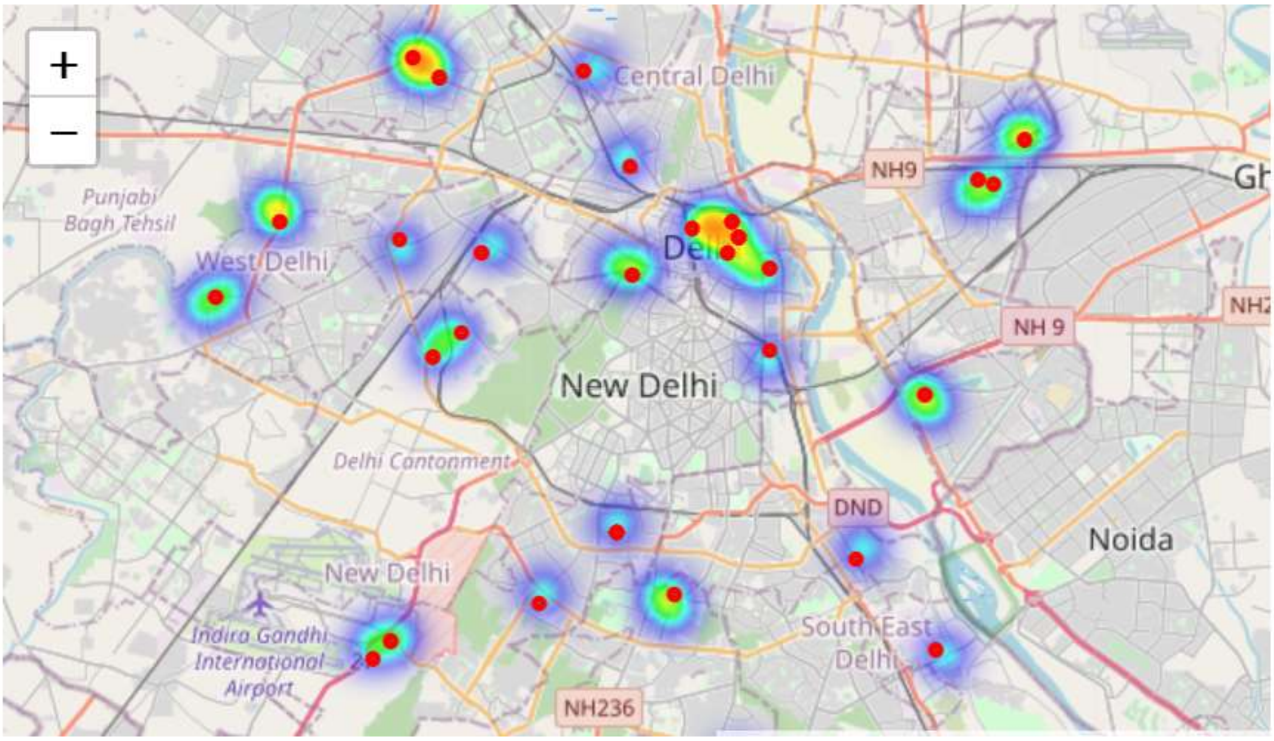
We know that when we were clustering the neighborhoods the data used contained the mean of all types of restaurants present in the particular neighborhood. Therefore, we can say that the neighborhoods are clustered on their restaurant trends.

Now, clusters 2 and 3 may collectively have the highest number of indian restaurant but there will be some neighborhoods in these clusters which would have a demand for Indian Restaurants, as these neighborhoods are in the same cluster, but would not have enough supply.



	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude
4	Ashok Vihar	28.699453	77.184826	Nat Khat Caterers	28.699630	77.187832
5	Ashok Vihar	28.699453	77.184826	Bakers Stop	28.700495	77.188716
6	Ashok Vihar	28.699453	77.184826	Invitation Banquet	28.696018	77.185953
7	Ashok Vihar	28.699453	77.184826	Gola Northend	28.701242	77.189288
17	Kohat Enclave	28.698041	77.140539	Peshawari	28.699012	77.139020

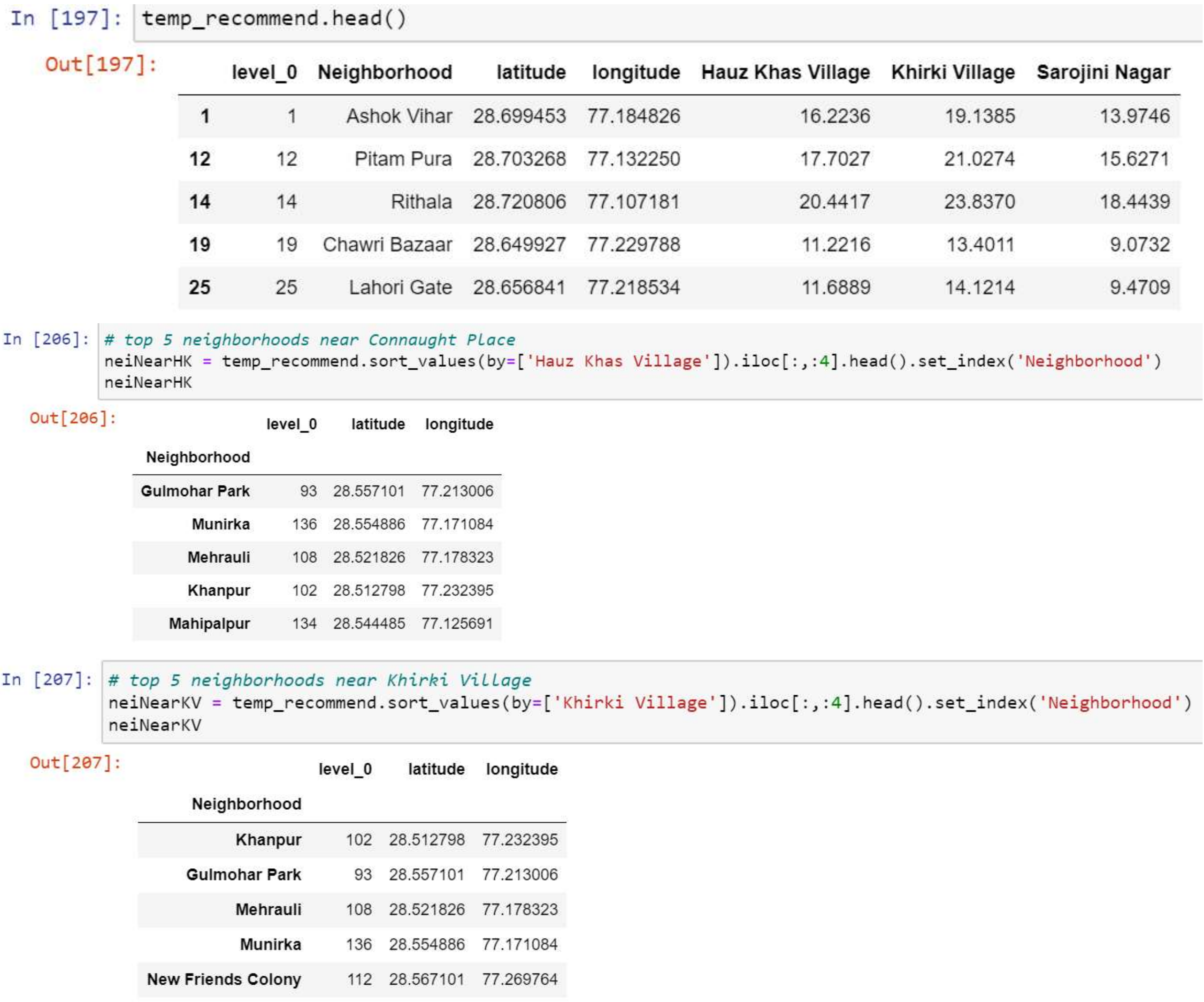




now we will remove all neighbourhoods with the following conditions:

- ⍰ Number of Indian restaurants >30%

- ⍰ Number of all restaurants >60% '%' here refers to percentile



```
In [197]: temp_recommend.head()
```

Out[197]:

	level_0	Neighborhood	latitude	longitude	Hauz Khas Village	Khirki Village	Sarojini Nagar
1	1	Ashok Vihar	28.699453	77.184826	16.2236	19.1385	13.9746
12	12	Pitam Pura	28.703268	77.132250	17.7027	21.0274	15.6271
14	14	Rithala	28.720806	77.107181	20.4417	23.8370	18.4439
19	19	Chawri Bazaar	28.649927	77.229788	11.2216	13.4011	9.0732
25	25	Lahori Gate	28.656841	77.218534	11.6889	14.1214	9.4709

```
In [206]: # top 5 neighborhoods near Connaught Place
          neiNearHK = temp_recommend.sort_values(by=['Hauz Khas Village']).iloc[:,:4].head().set_index('Neighborhood')
          neiNearHK
```

Out[206]:

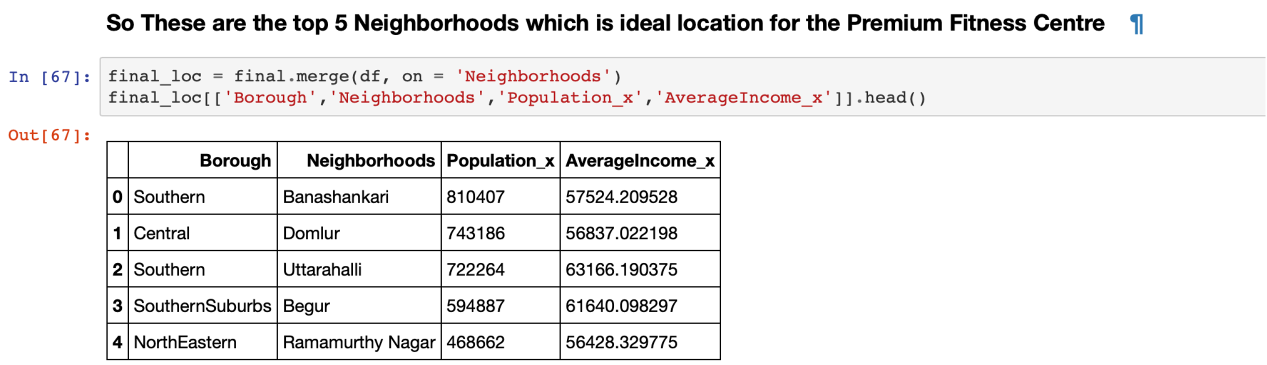
	level_0	latitude	longitude
Neighborhood			
Gulmohar Park	93	28.557101	77.213006
Munirka	136	28.554886	77.171084
Mehrauli	108	28.521826	77.178323
Khanpur	102	28.512798	77.232395
Mahipalpur	134	28.544485	77.125691

```
In [207]: # top 5 neighborhoods near Khirki Village
          neiNearKV = temp_recommend.sort_values(by=['Khirki Village']).iloc[:,:4].head().set_index('Neighborhood')
          neiNearKV
```

Out[207]:

	level_0	latitude	longitude
Neighborhood			
Khanpur	102	28.512798	77.232395
Gulmohar Park	93	28.557101	77.213006
Mehrauli	108	28.521826	77.178323
Munirka	136	28.554886	77.171084
New Friends Colony	112	28.567101	77.269764

Results and Discussions



So These are the top 5 Neighborhoods which is ideal location for the Premium Fitness Centre ¶

```
In [67]: final_loc = final.merge(df, on = 'Neighborhoods')
         final_loc[['Borough','Neighborhoods','Population_x','AverageIncome_x']].head()
```

Out[67]:

	Borough	Neighborhoods	Population_x	AverageIncome_x
0	Southern	Banashankari	810407	57524.209528
1	Central	Domlur	743186	56837.022198
2	Southern	Uttarahalli	722264	63166.190375
3	SouthernSuburbs	Begur	594887	61640.098297
4	NorthEastern	Ramamurthy Nagar	468662	56428.329775

Banashankari in Central Bengaluru is an ideal location for opening a new Premium Fitness Centre or a gym Franchise.