IBM DATA SCIENCE PROFFESIONAL CERTIFICATE CAPSTONE PROJECT

ASHISH KUMAR GAUTAM

29-JULY-2020

TOPIC: **LOCATING THE BEST PLACE IN DELHI NCR REGION (INDIA) TO OPEN A CHINESE FOOD JOINT.**

INTRODUCTION:

The inspiration behind this problem is pretty simple. I'm in love with Chinese cuisine. So I kind of thought of a way to incorporate this final report in and around that. The idea is to find a few Chinese Food Spots in and around town and classify them with parameters that are available.

BUSINESS PROBLEM:

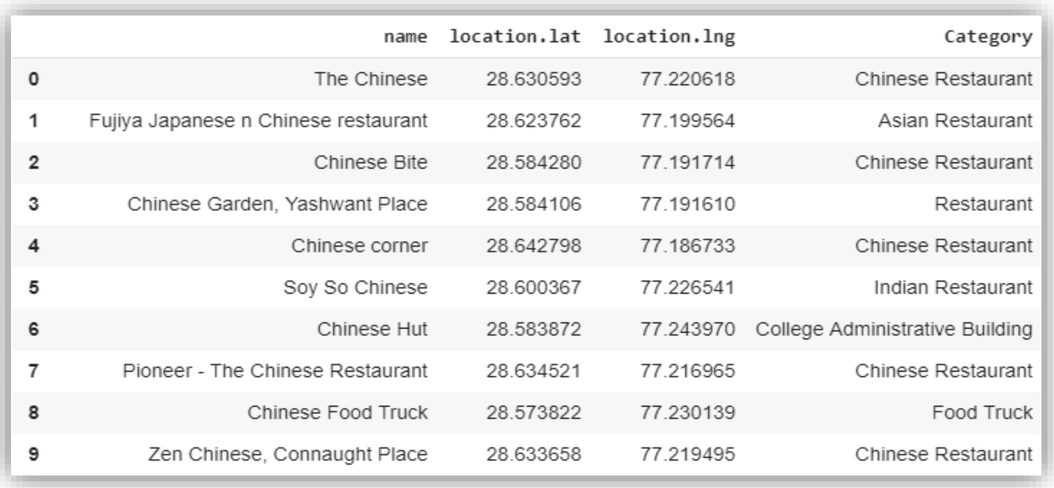
Identifying critical parameters to open a new Chinese Food Joint in Delhi-NCR area.

Data Used:

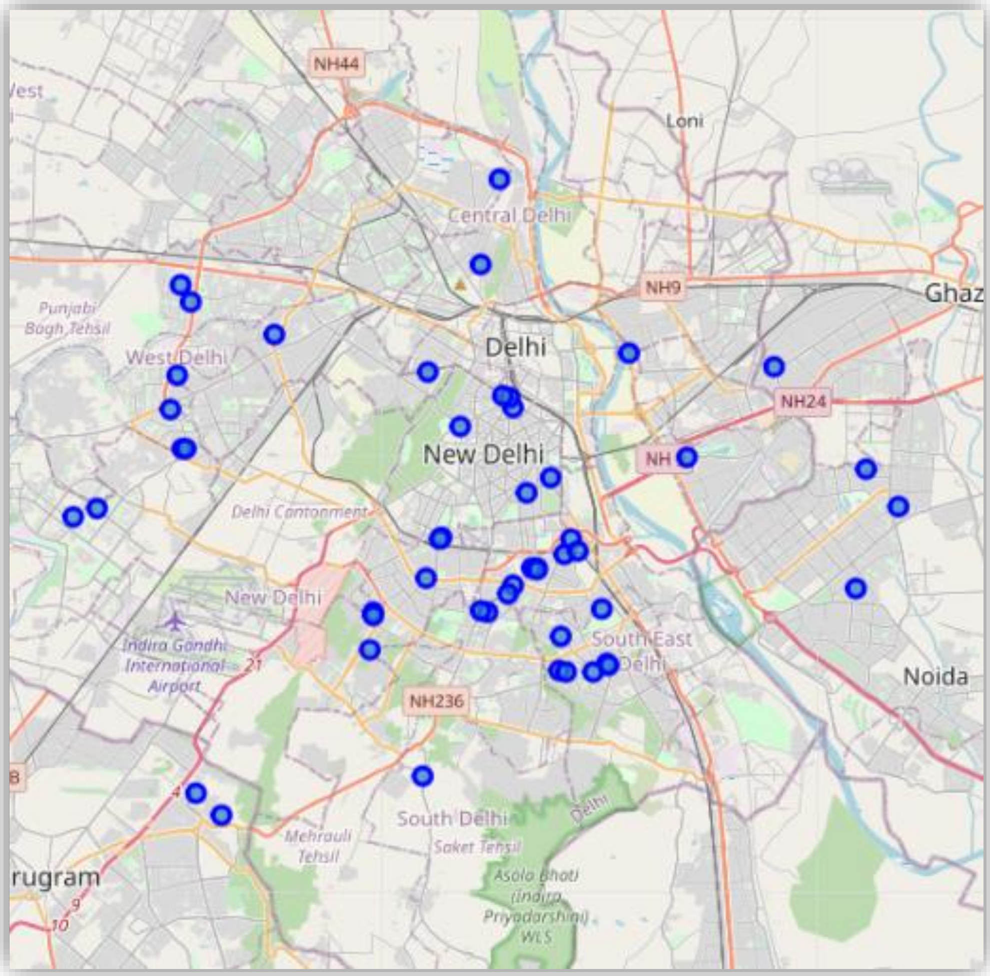
The Data is retrieved from Foursquare API for locations and names of the places [constraint in the project, the project could have been done with the help of Zomato API itself]. The other parameters like Price, Average rating from users are obtained from the ZOMATO API (a popular food delivery service in India).

Methodology:

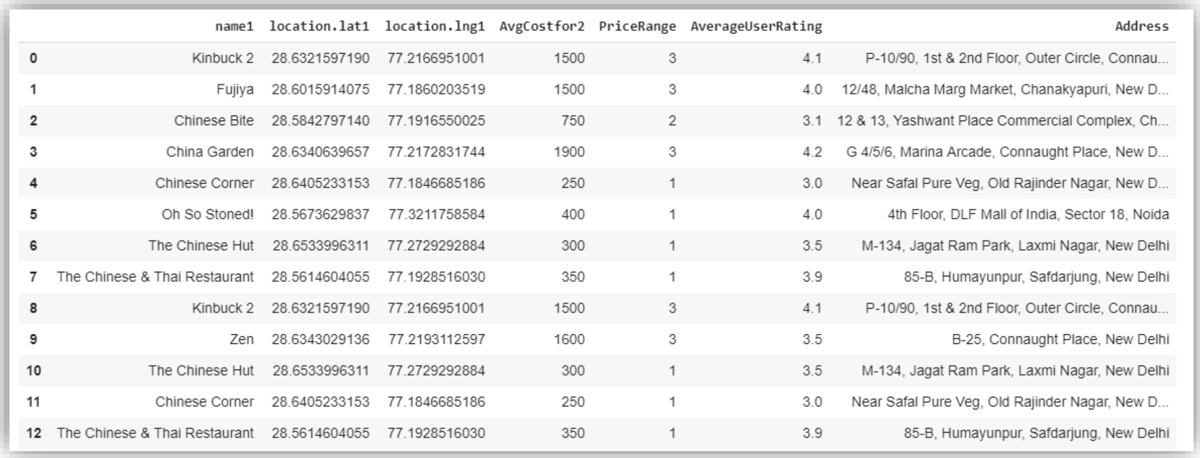
Initially, I queried 'Chinese' and obtained results from the foursquare api in and around Delhi-NCR. The radius was kept 5000 (large to get enough results for a conclusion and not too large that we deviate too far from city centre). Relevant results were stored on a Data Frame. With the latitude & longitude of these places, markers were plotted on a map using folium.



	name	location.lat	location.lng	Category
0	The Chinese	28.630593	77.220618	Chinese Restaurant
1	Fujiya Japanese n Chinese restaurant	28.623762	77.199564	Asian Restaurant
2	Chinese Bite	28.584280	77.191714	Chinese Restaurant
3	Chinese Garden, Yashwant Place	28.584106	77.191610	Restaurant
4	Chinese corner	28.642798	77.186733	Chinese Restaurant
5	Soy So Chinese	28.600367	77.226541	Indian Restaurant
6	Chinese Hut	28.583872	77.243970	College Administrative Building
7	Pioneer - The Chinese Restaurant	28.634521	77.216965	Chinese Restaurant
8	Chinese Food Truck	28.573822	77.230139	Food Truck
9	Zen Chinese, Connaught Place	28.633658	77.219495	Chinese Restaurant

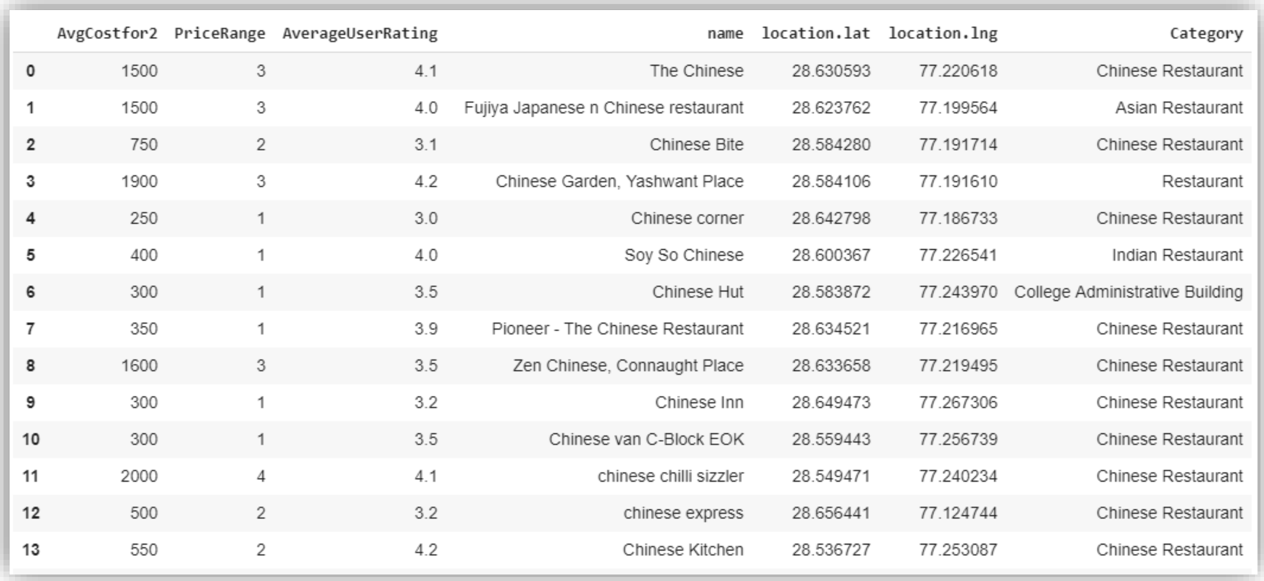


Next Data from Zomato API is collected using the same latitude longitudes of the obtained results from the foursquare API



	name1	location.lat1	location.lng1	AvgCostfor2	PriceRange	AverageUserRating	Address
0	Kinbuck 2	28.6321597190	77.2166951001	1500	3	4.1	P-10/90, 1st & 2nd Floor, Outer Circle, Connau...
1	Fujiya	28.6015914075	77.1860203519	1500	3	4.0	12/48, Malcha Marg Market, Chanakyapuri, New D...
2	Chinese Bite	28.5842797140	77.1916550025	750	2	3.1	12 & 13, Yashwant Place Commercial Complex, Ch...
3	China Garden	28.6340639657	77.2172831744	1900	3	4.2	G 4/5/6, Marina Arcade, Connaught Place, New D...
4	Chinese Corner	28.6405233153	77.1846685186	250	1	3.0	Near Safal Pure Veg, Old Rajinder Nagar, New D...
5	Oh So Stoned!	28.5673629837	77.3211758584	400	1	4.0	4th Floor, DLF Mall of India, Sector 18, Noida
6	The Chinese Hut	28.6533996311	77.2729292884	300	1	3.5	M-134, Jagat Ram Park, Laxmi Nagar, New Delhi
7	The Chinese & Thai Restaurant	28.5614604055	77.1928516030	350	1	3.9	85-B, Humayunpur, Safdarjung, New Delhi
8	Kinbuck 2	28.6321597190	77.2166951001	1500	3	4.1	P-10/90, 1st & 2nd Floor, Outer Circle, Connau...
9	Zen	28.6343029136	77.2193112597	1600	3	3.5	B-25, Connaught Place, New Delhi
10	The Chinese Hut	28.6533996311	77.2729292884	300	1	3.5	M-134, Jagat Ram Park, Laxmi Nagar, New Delhi
11	Chinese Corner	28.6405233153	77.1846685186	250	1	3.0	Near Safal Pure Veg, Old Rajinder Nagar, New D...
12	The Chinese & Thai Restaurant	28.5614604055	77.1928516030	350	1	3.9	85-B, Humayunpur, Safdarjung, New Delhi

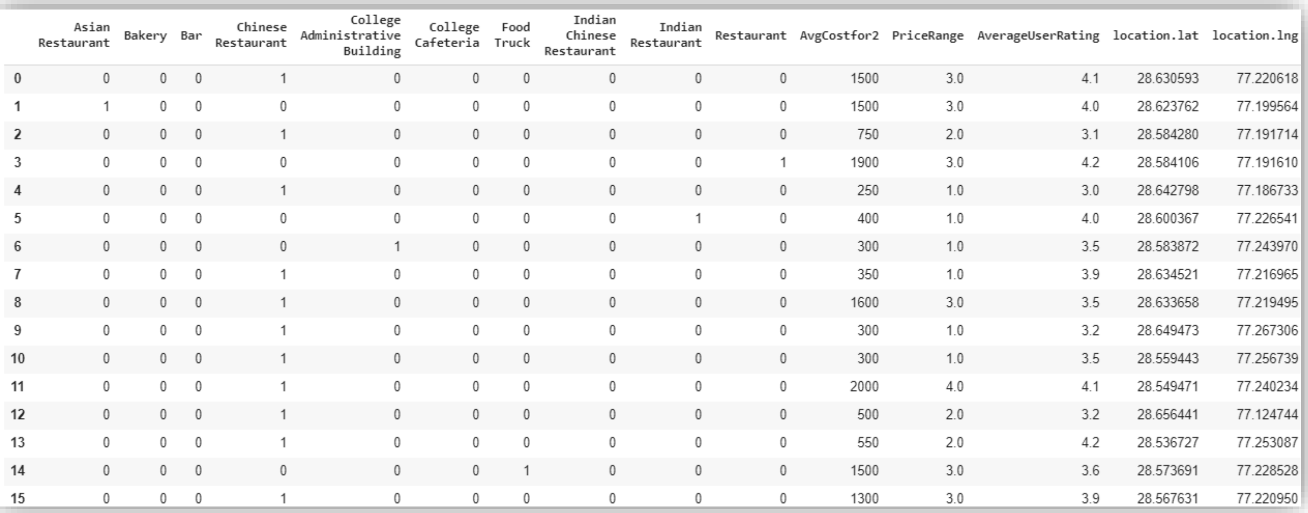
As it can be observed that there are some duplicate entries on the basis of longitude and latitudes we have provided so we need to clean this data. After cleaning we need to merge this data with the Foursquare API data that we retrieved. After merging we have data frame as shown below.



	AvgCostfor2	PriceRange	AverageUserRating	name	location.lat	location.lng	Category
0	1500	3	4.1	The Chinese	28.630593	77.220618	Chinese Restaurant
1	1500	3	4.0	Fujiya Japanese n Chinese restaurant	28.623762	77.199564	Asian Restaurant
2	750	2	3.1	Chinese Bite	28.584280	77.191714	Chinese Restaurant
3	1900	3	4.2	Chinese Garden, Yashwant Place	28.584106	77.191610	Restaurant
4	250	1	3.0	Chinese corner	28.642798	77.186733	Chinese Restaurant
5	400	1	4.0	Soy So Chinese	28.600367	77.226541	Indian Restaurant
6	300	1	3.5	Chinese Hut	28.583872	77.243970	College Administrative Building
7	350	1	3.9	Pioneer - The Chinese Restaurant	28.634521	77.216965	Chinese Restaurant
8	1600	3	3.5	Zen Chinese, Connaught Place	28.633658	77.219495	Chinese Restaurant
9	300	1	3.2	Chinese Inn	28.649473	77.267306	Chinese Restaurant
10	300	1	3.5	Chinese van C-Block EOK	28.559443	77.256739	Chinese Restaurant
11	2000	4	4.1	chinese chilli sizzler	28.549471	77.240234	Chinese Restaurant
12	500	2	3.2	chinese express	28.656441	77.124744	Chinese Restaurant
13	550	2	4.2	Chinese Kitchen	28.536727	77.253087	Chinese Restaurant

Then Data Wrangling is done to use most of the data, eg One Hot Encoding of Category, Names are dropped as they are of no use etc.

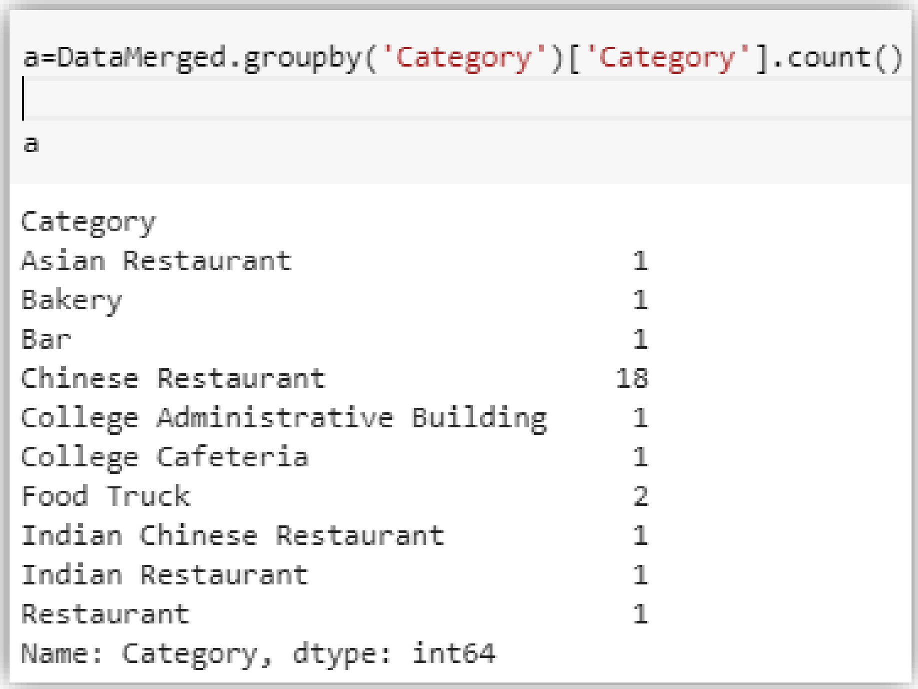
The final Data looks something like this below.



	Asian Restaurant	Bakery	Bar	Chinese Restaurant	College Administrative Building	College Cafeteria	Food Truck	Indian Chinese Restaurant	Indian Restaurant	Restaurant	AvgCostfor2	PriceRange	AverageUserRating	location.lat	location.lng
0	0	0	0	1	0	0	0	0	0	0	1500	3.0	4.1	28.630593	77.220618
1	1	0	0	0	0	0	0	0	0	0	1500	3.0	4.0	28.623762	77.199564
2	0	0	0	1	0	0	0	0	0	0	750	2.0	3.1	28.584280	77.191714
3	0	0	0	0	0	0	0	0	0	1	1900	3.0	4.2	28.584106	77.191610
4	0	0	0	1	0	0	0	0	0	0	250	1.0	3.0	28.642798	77.186733
5	0	0	0	0	0	0	0	0	1	0	400	1.0	4.0	28.600367	77.226541
6	0	0	0	0	1	0	0	0	0	0	300	1.0	3.5	28.583872	77.243970
7	0	0	0	1	0	0	0	0	0	0	350	1.0	3.9	28.634521	77.216965
8	0	0	0	1	0	0	0	0	0	0	1600	3.0	3.5	28.633658	77.219495
9	0	0	0	1	0	0	0	0	0	0	300	1.0	3.2	28.649473	77.267306
10	0	0	0	1	0	0	0	0	0	0	300	1.0	3.5	28.559443	77.256739
11	0	0	0	1	0	0	0	0	0	0	2000	4.0	4.1	28.549471	77.240234
12	0	0	0	1	0	0	0	0	0	0	500	2.0	3.2	28.656441	77.124744
13	0	0	0	1	0	0	0	0	0	0	550	2.0	4.2	28.536727	77.253087
14	0	0	0	0	0	0	1	0	0	0	1500	3.0	3.6	28.573691	77.228528
15	0	0	0	1	0	0	0	0	0	0	1300	3.0	3.9	28.567631	77.220950

Data Analysis:

Data is grouped to check the spread of categories.

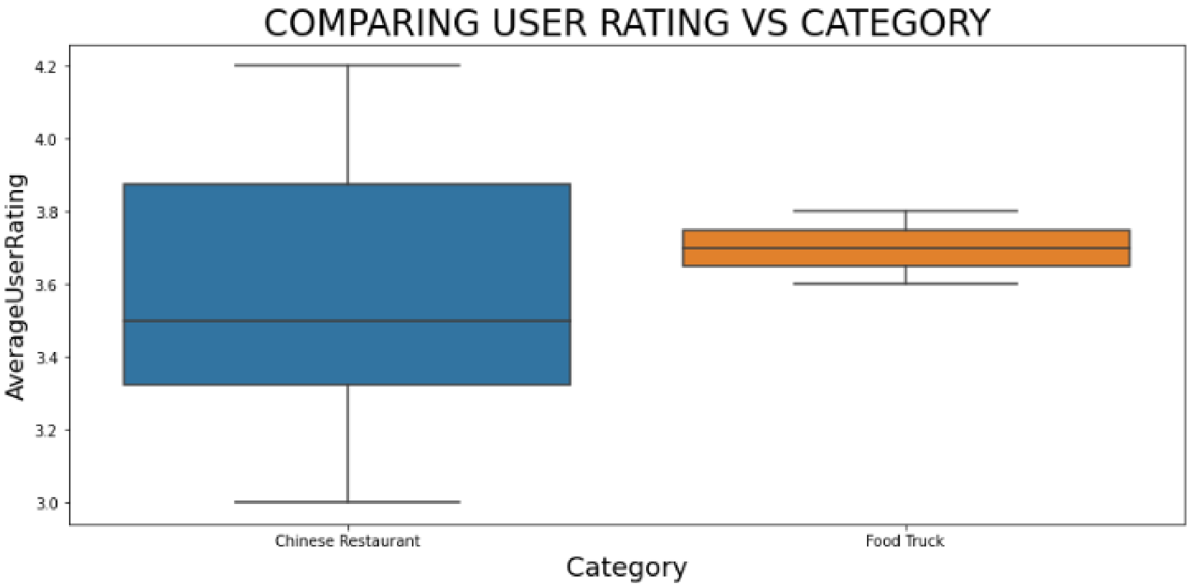


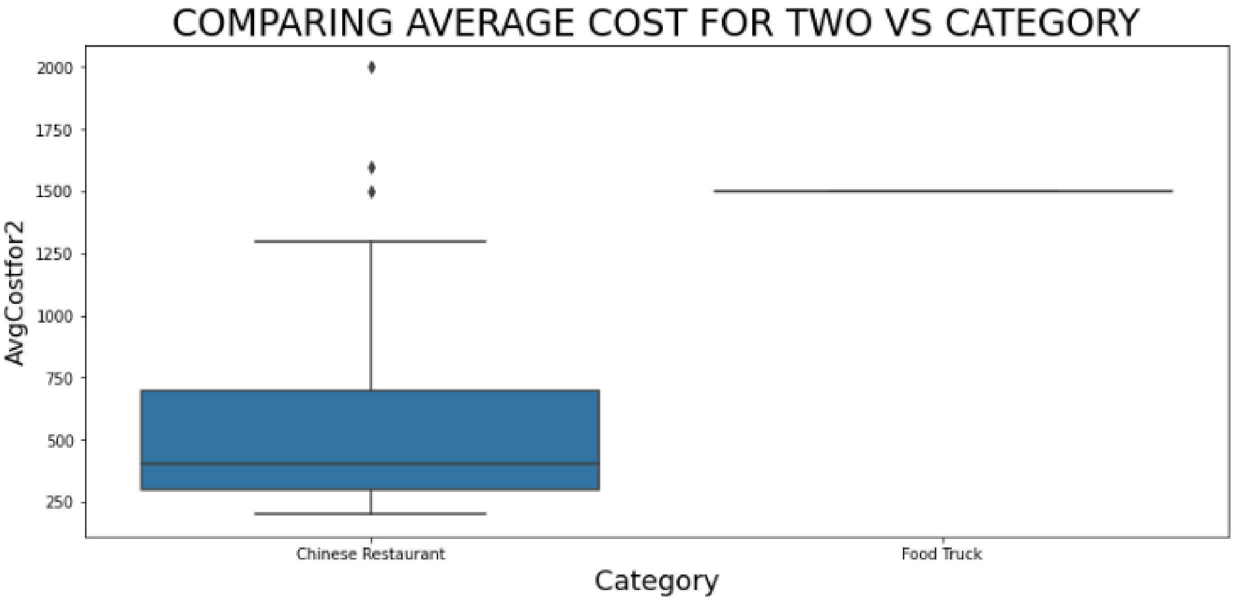
```
a=DataMerged.groupby('Category')['Category'].count()

a
```

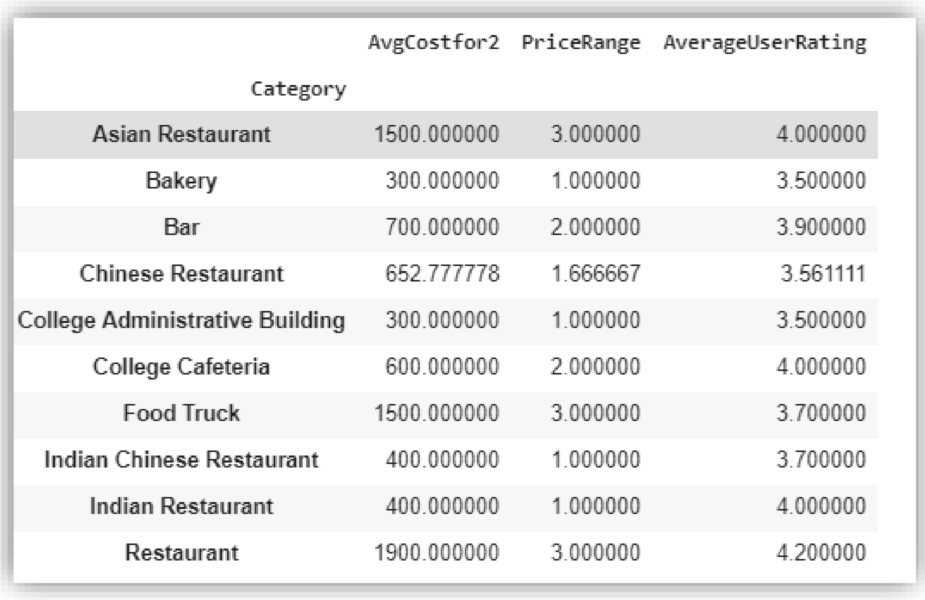
```
Category
Asian Restaurant                   1
Bakery                             1
Bar                                1
Chinese Restaurant                18
College Administrative Building    1
College Cafeteria                  1
Food Truck                         2
Indian Chinese Restaurant          1
Indian Restaurant                  1
Restaurant                         1
Name: Category, dtype: int64
```

As we can see most number of occurrences are for Chinese Restaurant and Food Truck. So plotting the boxplot to see variations.



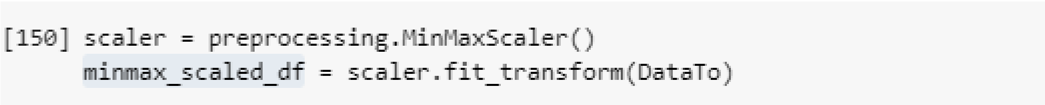


By initial glance we see that Food trucks are better as they have higher cost and better ratings that Chinese Restaurants.

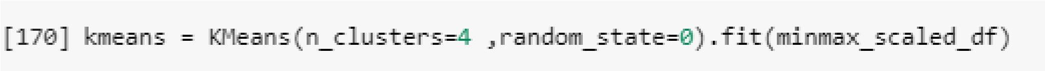


	AvgCostfor2	PriceRange	AverageUserRating
Category			
Asian Restaurant	1500.000000	3.000000	4.000000
Bakery	300.000000	1.000000	3.500000
Bar	700.000000	2.000000	3.900000
Chinese Restaurant	652.777778	1.666667	3.561111
College Administrative Building	300.000000	1.000000	3.500000
College Cafeteria	600.000000	2.000000	4.000000
Food Truck	1500.000000	3.000000	3.700000
Indian Chinese Restaurant	400.000000	1.000000	3.700000
Indian Restaurant	400.000000	1.000000	4.000000
Restaurant	1900.000000	3.000000	4.200000

Now all that’s left is that we min-max scale the data (since K-means is based on Euclidean Distance , we don’t want any features contributing more than they should!) and feed them into the Kmeans from sklearn.



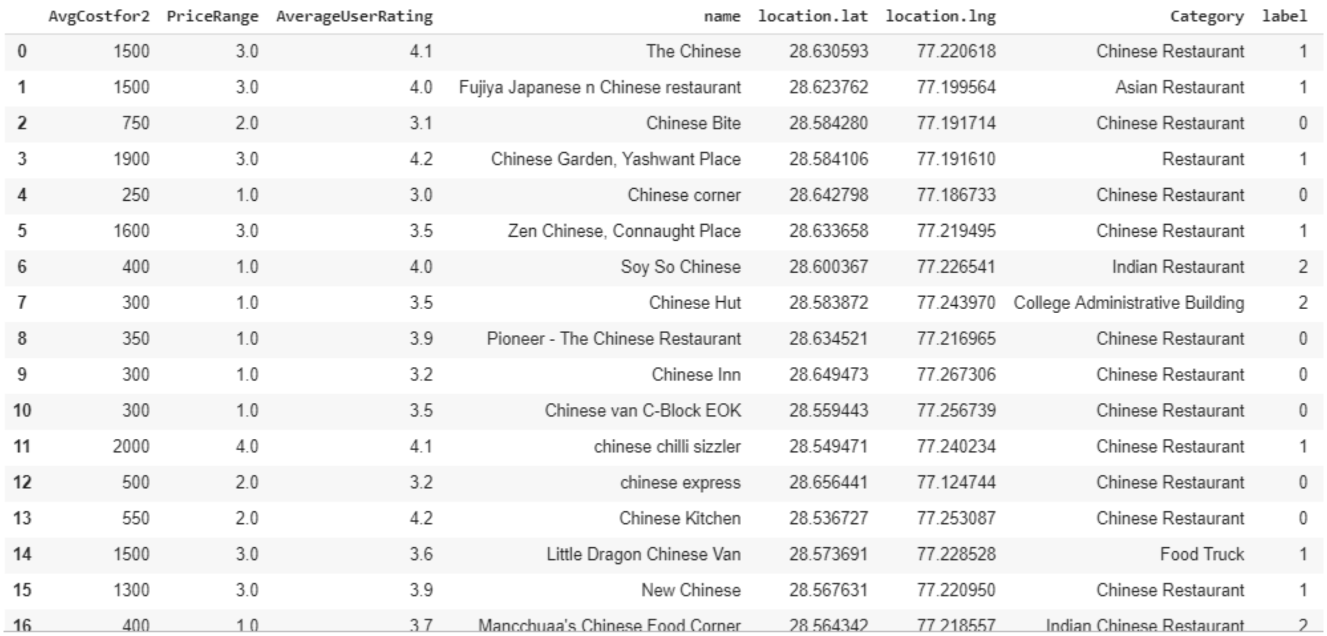
```
[150] scaler = preprocessing.MinMaxScaler()
      minmax_scaled_df = scaler.fit_transform(DataTo)
```



```
[170] kmeans = KMeans(n_clusters=4 ,random_state=0).fit(minmax_scaled_df)
```

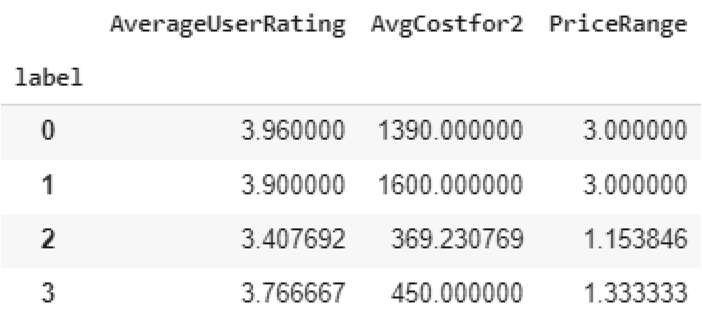
Results:

As we have used K means for clustering the DF with labels looks like



	AvgCostfor2	PriceRange	AverageUserRating	name	location.lat	location.lng	Category	label
0	1500	3.0	4.1	The Chinese	28.630593	77.220618	Chinese Restaurant	1
1	1500	3.0	4.0	Fujiya Japanese n Chinese restaurant	28.623762	77.199564	Asian Restaurant	1
2	750	2.0	3.1	Chinese Bite	28.584280	77.191714	Chinese Restaurant	0
3	1900	3.0	4.2	Chinese Garden, Yashwant Place	28.584106	77.191610	Restaurant	1
4	250	1.0	3.0	Chinese corner	28.642798	77.186733	Chinese Restaurant	0
5	1600	3.0	3.5	Zen Chinese, Connaught Place	28.633658	77.219495	Chinese Restaurant	1
6	400	1.0	4.0	Soy So Chinese	28.600367	77.226541	Indian Restaurant	2
7	300	1.0	3.5	Chinese Hut	28.583872	77.243970	College Administrative Building	2
8	350	1.0	3.9	Pioneer - The Chinese Restaurant	28.634521	77.216965	Chinese Restaurant	0
9	300	1.0	3.2	Chinese Inn	28.649473	77.267306	Chinese Restaurant	0
10	300	1.0	3.5	Chinese van C-Block EOK	28.559443	77.256739	Chinese Restaurant	0
11	2000	4.0	4.1	chinese chilli sizzler	28.549471	77.240234	Chinese Restaurant	1
12	500	2.0	3.2	chinese express	28.656441	77.124744	Chinese Restaurant	0
13	550	2.0	4.2	Chinese Kitchen	28.536727	77.253087	Chinese Restaurant	0
14	1500	3.0	3.6	Little Dragon Chinese Van	28.573691	77.228528	Food Truck	1
15	1300	3.0	3.9	New Chinese	28.567631	77.220950	Chinese Restaurant	1
16	400	1.0	3.7	Mancchuaa's Chinese Food Corner	28.564342	77.218557	Indian Chinese Restaurant	2

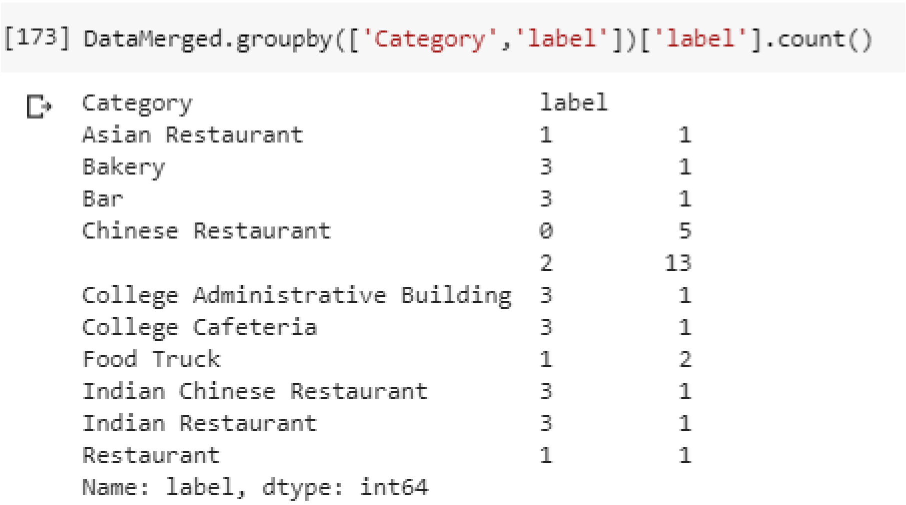
Grouping By Labels to see some pattern:



	AverageUserRating	AvgCostfor2	PriceRange
label			
0	3.960000	1390.000000	3.000000
1	3.900000	1600.000000	3.000000
2	3.407692	369.230769	1.153846
3	3.766667	450.000000	1.333333

1) 0--> HIGH USER RATING AND MEDIUM PRICE RANGE
2) 1--> HIGH PRICE RANGE AND HIGH USER RATING
3) 2--> LOW COST + LOW RATING
4) 3--> LOW COST + MEDIUM RATING

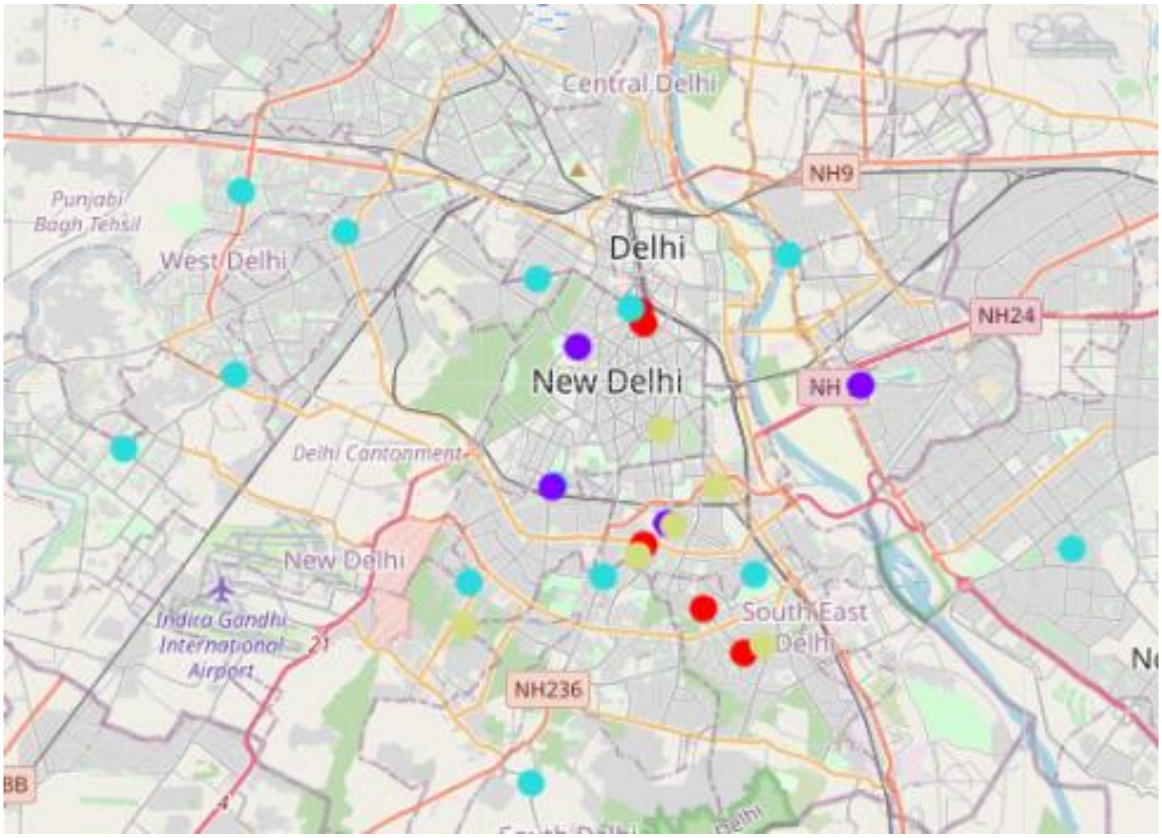
Grouping Labels with category:



```
[173] DataMerged.groupby(['Category','label'])['label'].count()
```

```
Category                          label
Asian Restaurant                  1          1
Bakery                            3          1
Bar                               3          1
Chinese Restaurant                0          5
                                  2         13
College Administrative Building   3          1
College Cafeteria                 3          1
Food Truck                        1          2
Indian Chinese Restaurant         3          1
Indian Restaurant                 3          1
Restaurant                        1          1
Name: label, dtype: int64
```

Then we can plot cluster on the Map again to visualize it better



Red:	Cluster 0	– 5 Nos
Purple:	Cluster 1	- 4 Nos
Cyan:	Cluster 3	- 13 Nos
Yellow:	Cluster 4	- 6 Nos

Discussion:

- Label 0 seems like the best option if you have a medium budget. (Fried chicken spots)
- We can see clusters of Purple- Label 1 [Food truck, BBQ joint etc.] around south and east Delhi– being posh places, where people like to spend more. So if you have budget you should go for these locations.
- Label 2 has the lowest rating. So no point being in this category even though it is most crowded category.
- Label 3 is for Low Price Medium rating cab be a good choice if you are running low on budget.

CONCLUSION

Several parameters of the problem were analyzed and discussed successfully, and the question where and what type of Chinese Restaurant should you open was answered.