Battle of the Neighborhoods

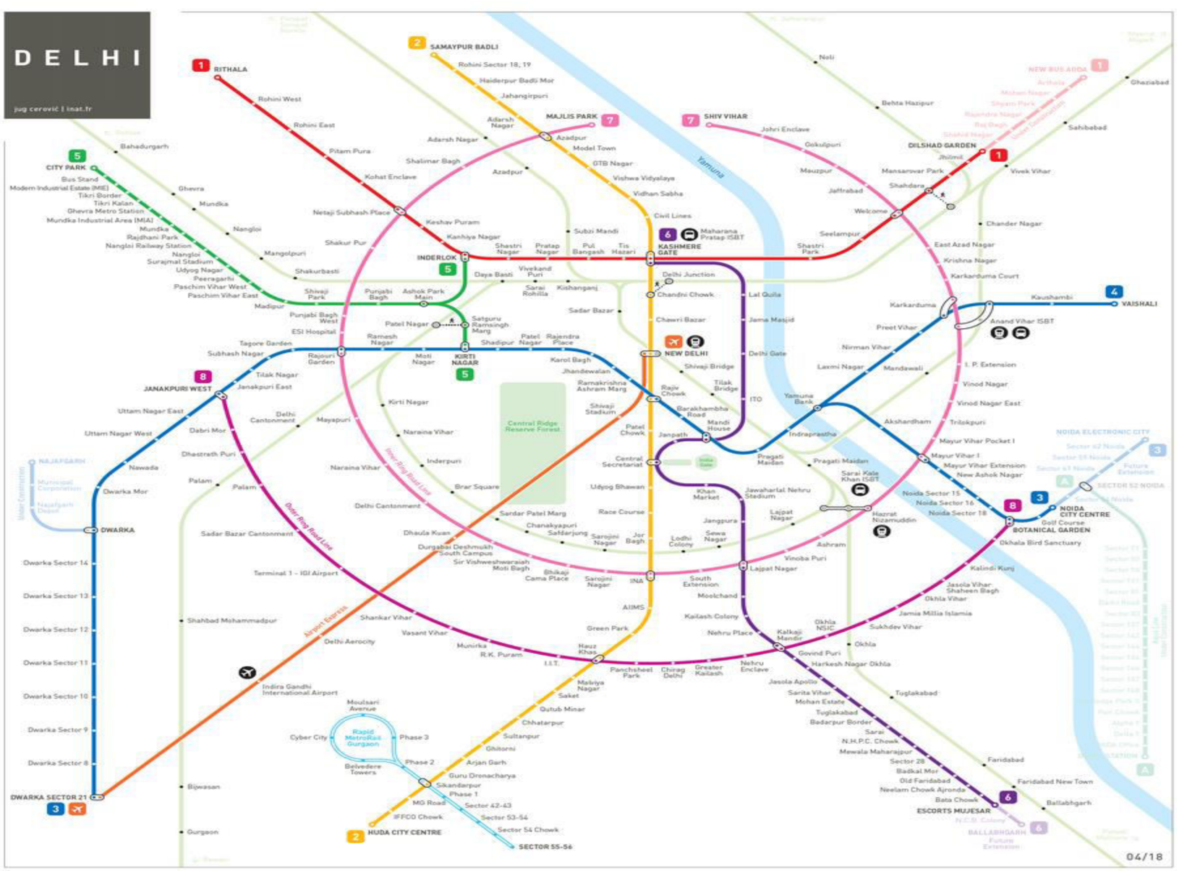
(Delhi Metro Rail Corporation)

Introduction: Business Problem

In this project we will try to find an optimal location for a Cafeteria. Specifically, this report will be targeted to stakeholders interested in opening an **Cafeteria** in **Delhi**, India.

Since there are lots of Cafe's in Delhi we will try to detect **locations that are not already crowded with self serving Cafeteria's**. We are also particularly interested in **all the areas within 500m radius of Metro Stations along Delhi Mtero Rail Corp. network**. We would also prefer locations **as close to Metro Station as possible**, assuming that first two conditions are met.

We will use our data science powers to generate a few most promising neighborhoods 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

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

- number of metro stations in DMRC network (included Rapid Metro Stations)
- number of existing Cafe/restaurants in the 500m radius of Metro Station (any type of Self Serving Restaurant/Cafe)
- number of and distance to nearby Cafeteria in the neighborhood, if any
- distance of neighborhood from Metro Stations

We decided to use regularly spaced grid of locations, centered around each Metro Station, to define our neighborhoods.

Following data sources will be needed to extract/generate the required information:

- centers of candidate areas will be generated algorithmically and approximate addresses of centers of those areas will be obtained using **Google Maps API reverse geocoding**
- number of restaurants and their type and location in every neighborhood will be obtained using **Foursquare API**
- coordinate of Berlin center will be obtained using **Google Maps API geocoding** of well known Berlin location (Alexanderplatz)

Web Scraping using BeautifulSoup for Data Collection

Web Scraping (also termed Screen Scraping, Web Data Extraction, Web Harvesting etc.) is a technique employed to extract large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format.

Data displayed by most websites can only be viewed using a web browser. They do not offer the functionality to save a copy of this data for personal use. The only option then is to manually copy and paste the data - a very tedious job which can take many hours or sometimes days to complete. Web Scraping is the technique of automating this process, so that instead of manually copying the data from websites, the Web Scraping software will perform the same task within a fraction of the time.

Setting up Target URL

“ https://delhimetrorail.info/delhi-metro-stations “ - DelhiMetroRail website gonna be our Target URL from which we will fetch all the Metro Stations related Data and details of each metro Station.

Neighborhood Candidates

Let's create latitude & longitude coordinates for centroids of our candidate neighborhoods. We will create a grid of cells covering our area of interest which is aprox. 500 meters centered around each Metro Station of Delhi Metro.

Let's first find the Unique name of each Metro Station by Web Scraping the Metro Network Data from DMRC website using BeautifulSoup package and further Geo-Coding each using well known address and Google Maps geocoding API.

Geo-Coding

Let's create latitude & longitude coordinates for centroids of our Metro neighborhoods using grid of cells covering our area of interest which is aprox. 500 meters centered around each Metro Station of Delhi Metro.

Let's first find the latitude & longitude of each Metro Station , using specific, well known address and Google Maps geocoding API.

In [19]:
```
locator = Nominatim(user_agent = "myGeocoder")
location = locator.geocode("New Delhi, India")
```

In [20]:
```
print("Latitude = {}, Longitude = {}".format(location.latitude, location.longitude))
```

```
Latitude = 28.6138954, Longitude = 77.2090057
```

In [21]:
```
station_mark = []
station_seq = []
Longitude = []
Latitude = []
for i in Stations.values:
    try: # because some links are broken
        location = locator.geocode( i[0] + ", India" )

        print("Station : ", i[0] , "Sequence Number : ",  i[1] , "Latitude = {}, Longitude = {}".format(location.latitude, loc
ation.longitude))
        station_mark.append(i[0])
        station_seq.append(i[1])
        Longitude.append(location.longitude)
        Latitude.append(location.latitude)
    except:
        continue
```

```
Station :  Shaheed Sthal(New Bus Adda) Sequence Number :  13 Latitude = 28.67052875, Longitude = 77.41580947285303
Station :  Hindon River Sequence Number :  14 Latitude = 28.6734288, Longitude = 77.4065374
Station :  Arthala Sequence Number :  15 Latitude = 28.676999, Longitude = 77.3918919
Station :  Mohan Nagar Sequence Number :  16 Latitude = 28.60631905, Longitude = 77.10608184860985
Station :  Shyam park Sequence Number :  17 Latitude = 28.698807199999997, Longitude = 77.26846412516488
Station :  Kashmere Gate Sequence Number :  28 Latitude = 28.666814100000003, Longitude = 77.22905486082311
Station :  Tis Hazari Sequence Number :  29 Latitude = 28.6671626, Longitude = 77.2166306
Station :  Pul Bangash Sequence Number :  30 Latitude = 28.6664068, Longitude = 77.2074156
Station :  Pratap Nagar Sequence Number :  31 Latitude = 28.6667177, Longitude = 77.1988974
Station :  Shastri Nagar Sequence Number :  32 Latitude = 28.6700885, Longitude = 77.1818589
Station :  Inderlok Sequence Number :  33 Latitude = 28.67276855, Longitude = 77.16568970929804
Station :  Kanhaiya Nagar Sequence Number :  34 Latitude = 28.6824082, Longitude = 77.1647754
Station :  Keshav Puram Sequence Number :  35 Latitude = 28.6889264, Longitude = 77.1616833
Station :  Netaji Subash Place Sequence Number :  36 Latitude = 12.990637249999999, Longitude = 77.54423798682119
Station :  Kohat Enclave Sequence Number :  37 Latitude = 28.6980415, Longitude = 77.1405393
Station :  Pitam Pura Sequence Number :  38 Latitude = 28.7032676, Longitude = 77.1322497
```

Methodology

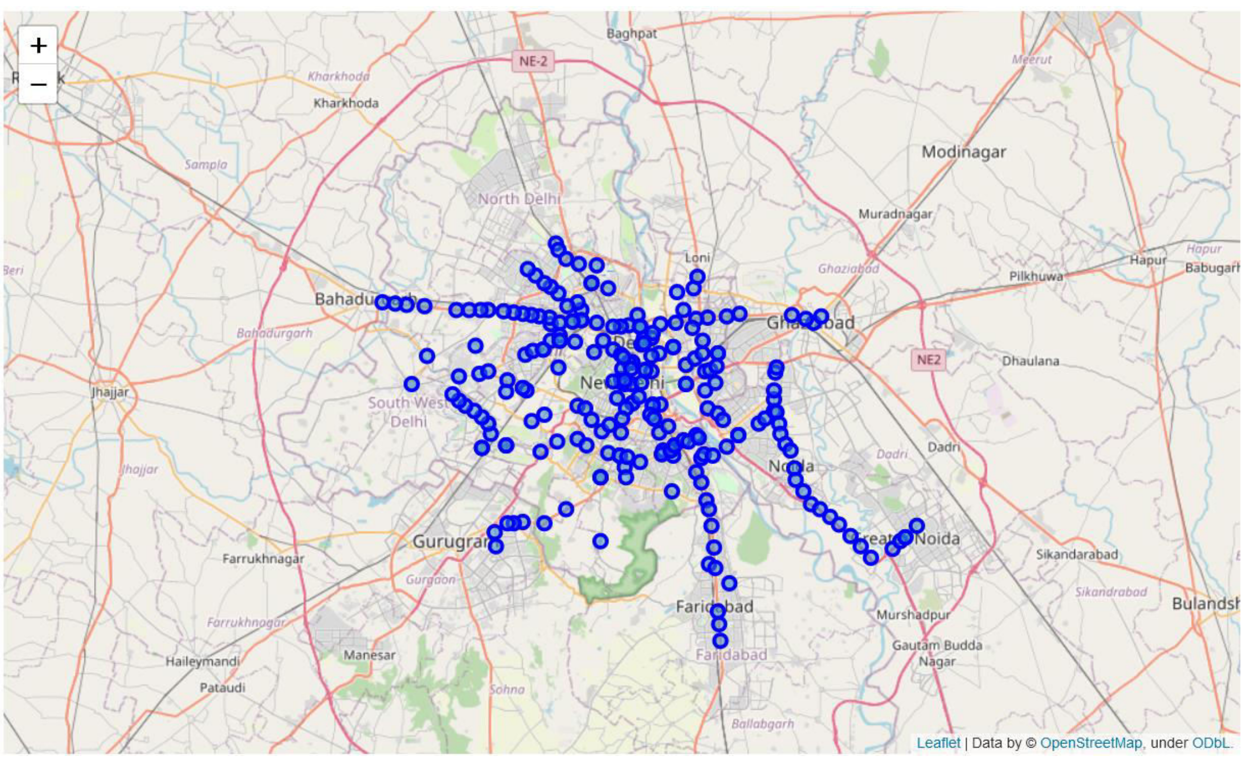
In this project we will direct our efforts on detecting areas along Delhi Metro Network that have low restaurant density, particularly those with low number of self serving Cafeteria's. We will limit our analysis to area ~500m around each Metro Station in DMRC network.

In first step we have collected the required **data: location and type (category) of every venue within 500m from each Metro Station center** . We have also **identified all the existing Cafeteria's** (according to Foursquare categorization).

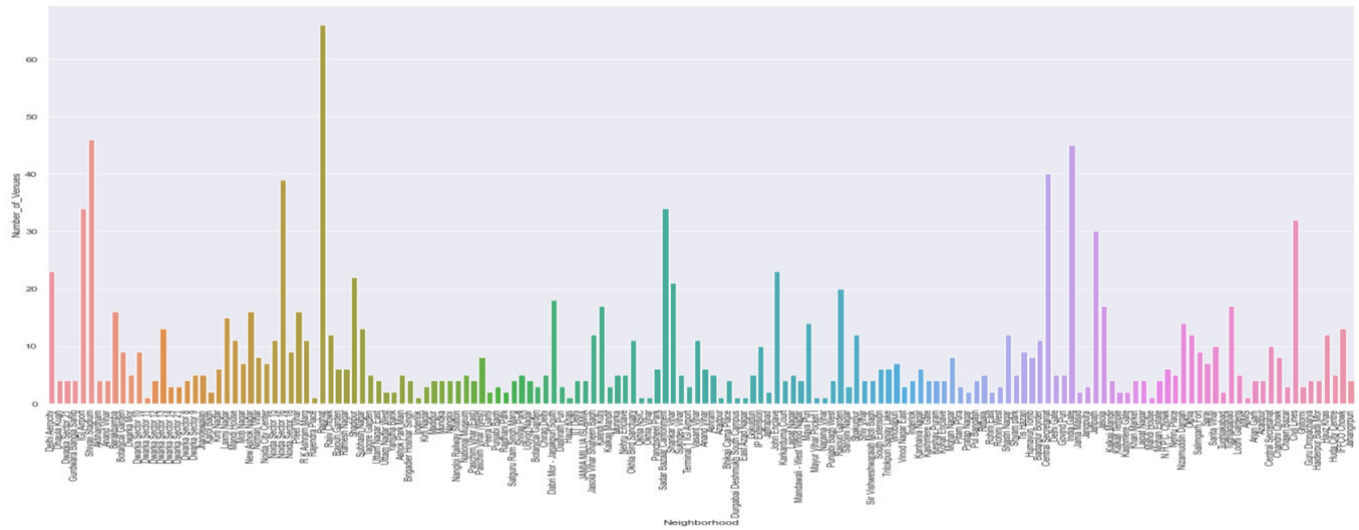
Second step in our analysis will be calculation and exploration of '**Cafeteria density**' across different Metro Stations of DMRC network - we will use **folium map visualizations** to identify a few promising areas close to Metro Stations with low number of Cafeteria's in general (*and* no Self Serving Cafeteria in vicinity) and focus our attention on those areas.

In third and final step we will focus on most promising areas and within those create **clusters of locations that meet some basic requirements** established in discussion with stakeholders: we will take into consideration locations with **all the Venues/Restaurants in radius of 500 meters**, and we want locations **without much Cafe's in radius of 500 meters**. We will present map of all such locations but also create clusters (using **k-means clustering**) of those locations to identify general zones / neighborhoods / addresses which should be a starting point for final 'street level' exploration and search for optimal venue location by stakeholders.

Map Visualization using Folium



Foursquare Venues Collection



Neighbourhood Venues Data Collection

	Neighborhood	Metro_Line	Neighborhood_Latitude	Neighborhood_Longitude	Venue	Venue_Latitude	Venue_Longitude	Venue_Category
0	Shaheed Sthal(New Bus Adda)	Red Line	28.670529	77.415809	Ghaziabad New Bus Stand	28.668896	77.413315	Bus Station
1	Shaheed Sthal(New Bus Adda)	Red Line	28.670529	77.415809	Floralbay	28.670859	77.412479	Gift Shop
2	Shaheed Sthal(New Bus Adda)	Red Line	28.670529	77.415809	Mohan makins	28.668894	77.418897	Brewery
3	Hindon River	Red Line	28.673429	77.406537	Axis Bank ATM	28.673000	77.407220	ATM
4	Hindon River	Red Line	28.673429	77.406537	Agresen Chowk	28.674032	77.402234	Moving Target
...	...	...	...	...	...	...	...	...
1611	Najafgarh	Gray Line	28.612304	76.982391	HDFC Bank	28.610185	76.981458	Bank
1612	Najafgarh	Gray Line	28.612304	76.982391	Axis Bank ATM	28.609630	76.981030	ATM
1613	Najafgarh	Gray Line	28.612304	76.982391	Axis Bank ATM	28.611666	76.978676	ATM
1614	Najafgarh	Gray Line	28.612304	76.982391	Axis Bank ATM	28.608890	76.982340	ATM
1615	Najafgarh	Gray Line	28.612304	76.982391	First Choice Food	28.614778	76.985149	Food & Drink Shop

1616 rows × 8 columns

Clustering

Clustering is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data. It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different. In other words, we try to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure such as euclidean-based distance or correlation-based distance. The decision of which similarity measure to use is application-specific.

Neighbourhood's Most Common Venues

	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue
0	Humayun Tomb	Venue_Category_Monument / Landmark	Venue_Category_Historic Site	Venue_Category_Garden	Venue_Category_Café
1	Lodhi Gardens	Venue_Category_Boarding House	Venue_Category_History Museum	Venue_Category_Park	Venue_Category_Spa
2	AIIMS	Venue_Category_Jewelry Store	Venue_Category_Airport Food Court	Venue_Category_Snack Place	Venue_Category_Light Rail Station
3	Akshardham	Venue_Category_Hindu Temple	Venue_Category_Athletics & Sports	Venue_Category_Yoga Studio	Venue_Category_Electronics Store
4	Anand Vihar	Venue_Category_Pizza Place	Venue_Category_Shoe Store	Venue_Category_Movie Theater	Venue_Category_Hotel
5	Arjan Garh	Venue_Category_Hotel	Venue_Category_Furniture / Home Store	Venue_Category_Light Rail Station	Venue_Category_Yoga Studio
6	Ashok Park Main	Venue_Category_Train Station	Venue_Category_Yoga Studio	Venue_Category_Eastern European Restaurant	Venue_Category_Food Court
7	Ashram	Venue_Category_Hotel	Venue_Category_Sculpture Garden	Venue_Category_Indian Restaurant	Venue_Category_Bakery
8	Azadpur	Venue_Category_Restaurant	Venue_Category_Pool Hall	Venue_Category_Bus Station	Venue_Category_Yoga Studio
9	Badarpur Border	Venue_Category_IT Services	Venue_Category_Train Station	Venue_Category_Eastern European Restaurant	Venue_Category_Food Court
10	Barakhamba	Venue_Category_Indian Restaurant	Venue_Category_Hotel	Venue_Category_Cocktail Bar	Venue_Category_Monument / Landmark
11	Bhikaji Cama Place	Venue_Category_Lounge	Venue_Category_Market	Venue_Category_Bakery	Venue_Category_Asian Restaurant

Clustering using K-Means

Clustering analysis can be done on the basis of features where we try to find subgroups of samples based on features or on the basis of samples where we try to find subgroups of features based on samples. We'll cover here clustering based on features. Clustering is used in market segmentation; where we try to find customers that are similar to each other whether in terms of behaviors or attributes, image segmentation/compression; where we try to group similar regions together, document clustering based on topics, etc.

Unlike supervised learning, clustering is considered an unsupervised learning method since we don't have the ground truth to compare the output of the clustering algorithm to the true labels to evaluate its performance. We only want to try to investigate the structure of the data by grouping the data points into distinct subgroups.

K-means Algorithm

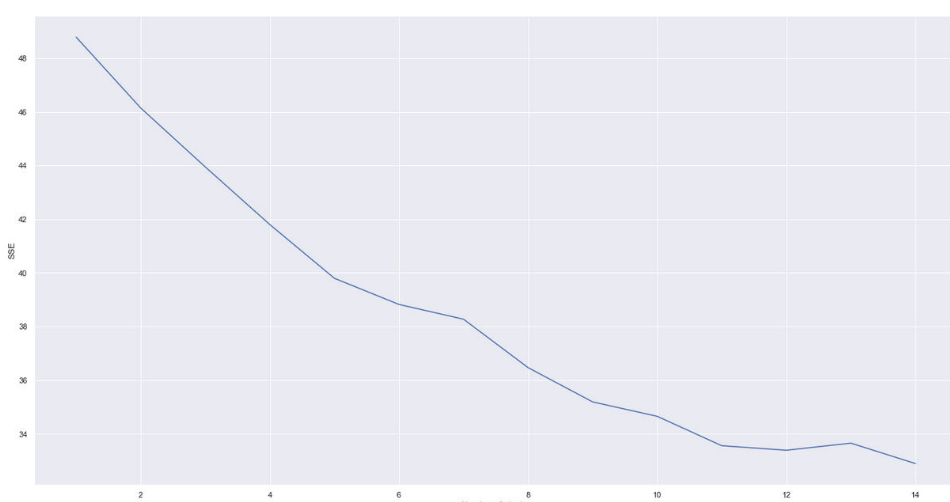
Kmeans algorithm is an iterative algorithm that tries to partition the dataset into *K*pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

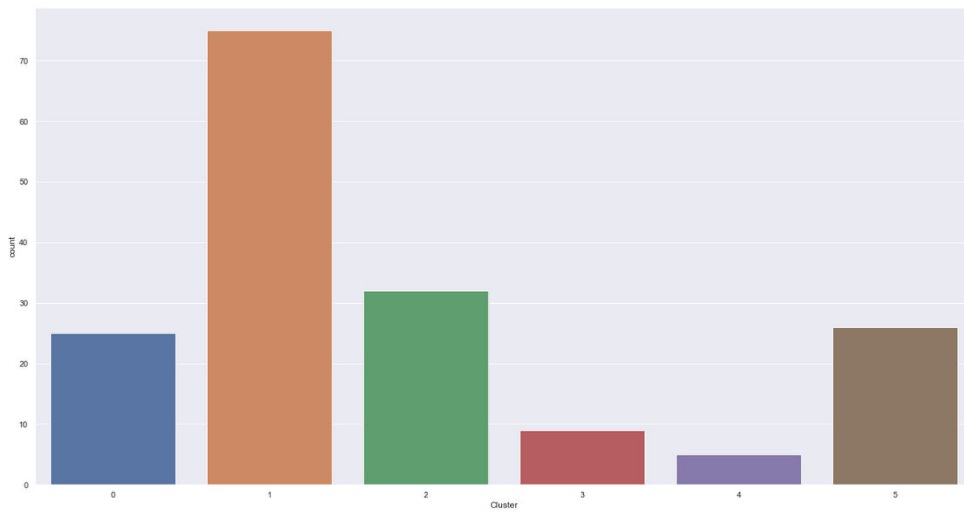
The way kmeans algorithm works is as follows:

1. Specify number of clusters *K*.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.

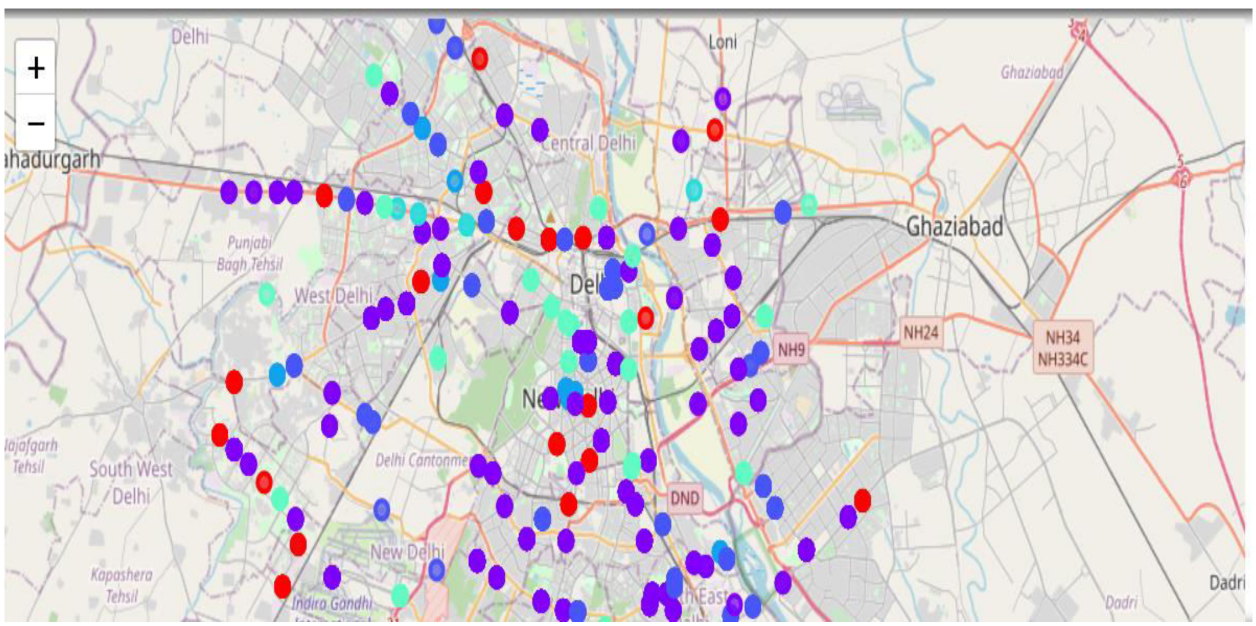
- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

The approach kmeans follows to solve the problem is called **Expectation-Maximization**. The E-step is assigning the data points to the closest cluster.

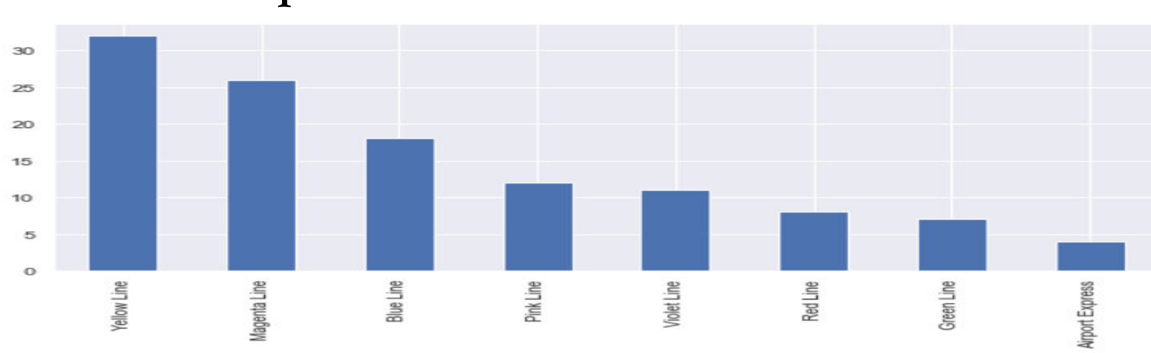


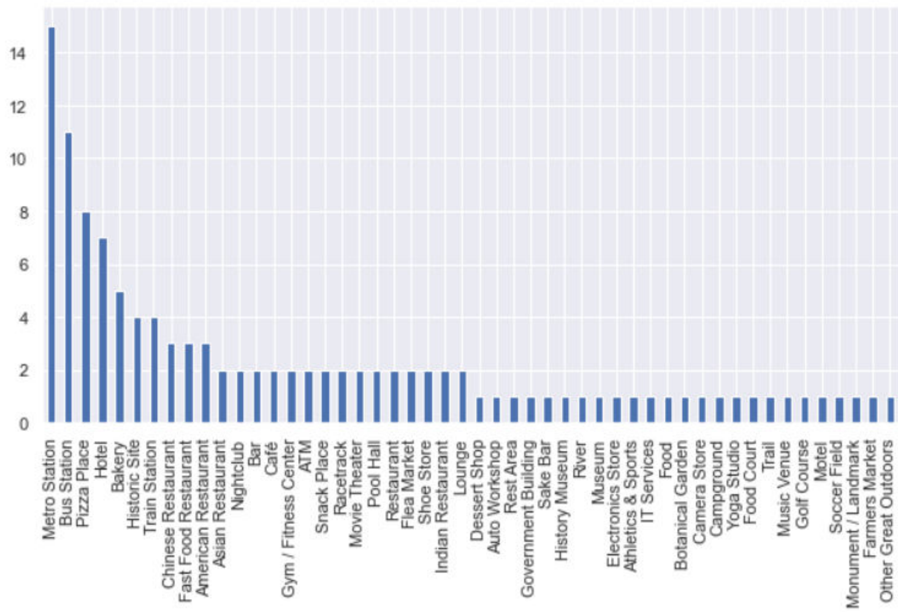


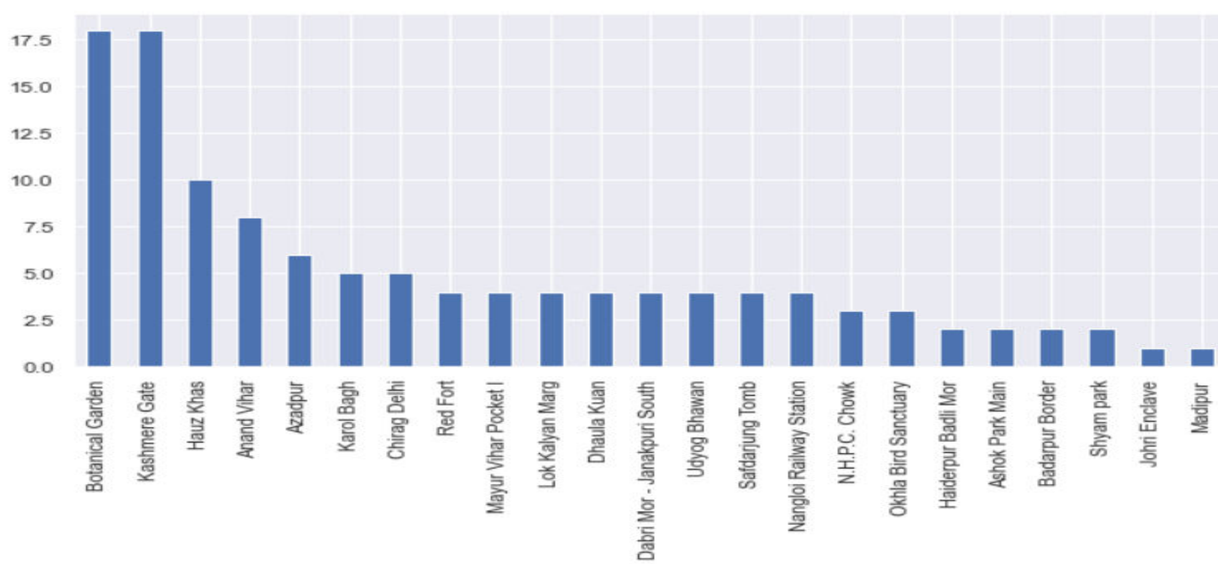
Cluster Visualization



Cluster-1 Interpretation







Locations Suggested from Cluster 1

(similar Analysis for each Cluster)

Results and Discussion

- Our analysis shows that although there is a great number of restaurants/Cafe's in Delhi (~1800 in our initial area of interest which was whole DMRC network), there are pockets of low restaurant density fairly close to some Metro Stations. Highest concentration of Cafeteria's was detec ted in Cluster 0 and 2 along Blue , Yellow ,Violet and Magenta Line of DMRC, so we focused our attention to areas where Cafe intensity is comparatively low. Another Metro Line was identified as potentially interesting (Magenta_Line and Green Line), but our attention was focused on Cluster 1 which offer a combination of popularity among travellers, closeness to city center, strong socio-economic dynamics *and* a number of pockets of low Cafe density.

- After directing our attention to this more narrow area of interest (covering cluster 1) we first created a dense grid of location candidates ; those locations are identified as Tagore Garden,India Gate,Kalkaji Mandir,AIIMS and Laxmi Nagar. These location candidates are our zones of interest which contain lowest number of existing Cafe's. Addresses of centers of those zones were also generated using reverse geocoding to be used as markers/starting points for more detailed local analysis based on other factors.

- Result of all this is 6 zones containing largest number of potential new Cafe locations based on number of and distance to existing venues - both Self Serving Restaurants in general and Cafeteria particularly. This, of course, does not imply that those zones are actually optimal locations for a new Cafeteria! Purpose of this analysis was to only provide info on areas close to Metro Stations but not crowded with existing Self Serving Restaurants (particularly Cafe) - it is entirely possible that there is a very good reason for small number of restaurants in any of those areas, reasons which would make them unsuitable for a new restaurant regardless of lack of competition in the area. Recommended zones should therefore be considered only as a starting point for more detailed analysis which could eventually result in location which has not only no nearby competition but also other factors taken into account and all other relevant conditions met.

Conclusion

Purpose of this project was to identify Delhi areas close to DMRC Metro Stations with low number of Self Serving - Restaurants (particularly Cafeteria) in order to aid stakeholders in narrowing down the search for optimal location for a new Cafeteria. By calculating restaurant density distribution from Foursquare data we have first identified general boroughs that justify further analysis (Cluster 1 and 2), and then generated extensive collection of locations which satisfy some basic requirements regarding existing nearby restaurants. Clustering of those locations was then performed in order to create major zones of interest (containing greatest number of potential locations) and addresses of those zone centers were created to be used as starting points for final exploration by stakeholders.

Final decission on optimal Cafe location will be made by stakeholders based on specific characteristics of neighborhoods and locations in every recommended zone, taking into consideration additional factors like attractiveness of each location (proximity to park or water), levels of noise / proximity to major roads, real estate availability, prices, social and economic dynamics of every neighborhood etc.