Selection of best location to open a new restaurant in India

Minhaj Ahmed Ansari

1. Business Problem

1.1. Introduction

The India business landscape is prepared to grow in multiple areas due to several factors such as international trade, government stimulus, and an overall strong developing country. With a young population that is rising to leadership and technology driving growth and innovation, there are several business opportunities in multiple sectors that we are going to explore today.

Many small business start-ups, in order to reduce expenditure usually settle for a cheap location. Others believe that location doesn't matter provided the product is right. However, getting a good location is very critical to the success of your business but this can prove quite difficult because one of the challenges of starting a business successfully is getting a good business site.

Now, for a food joint or restaurant, the location in itself becomes one of the major contributing factor towards success. Because for people to like your food, delicacies and environment you offer, they first need to reach it and distinguish it among the competition. Assuming that a Multi-national fast food chain wants to open its new chain of restaurant in India, which location should they choose? Which type of cuisine should they serve? And most of all how to ensure people are easily able to reach the location.

1.2. Problem

Now India is a large country, with a population of more than 1.3 billion people. It will become impossible to simply zero in on perfect location by starting randomly choosing a place or city. So, it's better to first identify most viable target locations and then probably decide among them based on various virtues and features. For this reason alone, I restricted my analysis to 5 major cities of India, which in a way are Economic, Technological and demographic centres of India. They are,

- Delhi
- Mumbai
- Chennai
- Kolkata
- Bengaluru

Now, we need to just compare these cities and identify the best of them based on some common parameters. These parameters will eventually help us to select the best features to recommend best location.

- Optimal distance between city centre and the proposed location.
- Selection of cuisine.
- Density of local businesses at the proposed location.

- Traffic Conditions (also instrumental in case of online deliveries)

However, it is important to note that, these are not the only parameters that contribute in the selection of parameters other factors such as rent, condition of building etc. which also impact the success of business.

However, this analysis can give a head start to businesses which they can utilise to do further deeper analysis of other features.

1.3. Interests

This analysis would be really helpful for the small businesses and start-ups who are looking to locate their stores/ restaurants at optimal location to get a jump start.

2. Data Collection and wrangling

2.1. Data Sources

Now there are many location services which provide various details about and around a particular location in a city. However, Foursquare is one of the freely available services which return a list of the recommended venues around a location based on various factors which include distance, type of business and likes or dislikes.

An easy way to get this data is through Foursquare Places API's **explore** endpoint. The documentation on how to create foursquare developer account, rate limits and calling methods is available on the link here.

Another data source that we will be using in this project will be **Traffic API** provided by Bing Maps. This API can be used to get traffic information on a particular set of location. A complete detailed documentation on this REST API is available here.

2.2. Data cleaning

For calling Foursquare API's explore endpoint, first we need is the spatial details of the location around which you want to explore the venues. We can choose this location as the city centre as explained in section 1.2. Another important parameter that you need to specify is 'radius'. This will determine the radius within which the venues are returned. For the purpose of this analysis let us fix it to 10 KM as beyond that, a. customers might not travel in peak times b. online delivery might not be possible.

After initial cleaning of the returned json data file, we populate all the responses in separate data frames (one for each city) using Pandas library. Each data frame consists of almost 100 rows and 22 columns.

Next we will rename some of the columns so that it is easier to manipulate and understand them. For example - 'venue.categories' can be renamed as only category.

Another cleaning exercise that we will do would be to get the exact category name, since the column returned contains a list.

Next we will drop unnecessary columns which do not contribute in any meaningful way to our analysis. These include - **['venue.photos.count', 'venue.photos.groups', 'venue.venuePage.id', 'venue.location.neighborhood', 'referralId', 'reasons.count', 'reasons.items', 'venue.location.cc', 'venue.location.postalCode', 'venue.location.labeledLatLngs']**

After above all operations now we are left with following columns in our data frame - **['id', 'name', 'address', 'crossStreet', 'lat', 'lng', 'distance', 'city', 'state', 'country', 'formattedAddress', 'categories', 'category']**

2.3. Data Binning & One Hot Encoding

This step will especially help us in performing some Exploratory data analysis and Data Visualisation in upcoming Sections.

First, one hot encoding is possible for 'Category' column, this will help us in identifying the most popular category and their distribution across the cities.

Next, the 'distance' feature is actually a continuous value which will be difficult to plot for every venue. To achieve more appropriate analysis, let us bin them in four categories.

- Very Near = 0 to 2 KM
- Near = 2 to 4 KM
- Far = 4 to 6 KM
- Very Far = 6 to 8 KM
- Outlier = 8 to 10 KM

One hot encoding for these binned distances is also possible and it can be done for further EDA during our project.

2.4. Feature Selection

Above data wrangling and cleaning leaves us with all the features, which are required for our further Exploratory data analysis and Modelling. Further features may be dropped once we have completed our EDA.

3. Exploratory Data Analysis

3.1. Understanding the Categories

Although it seems quite intuitive that in a particular country the most restaurants will serve the local cuisine only. However, the relationship between the cuisine and the restaurant where it is served and where it is located might not have been explored earlier. During this project, when trying to probe this relation, I found out that it may not always be the case.

To understand better, I binned, then one hot coded the distances from the city centre and tried to find the distribution of top 3 categories in all the distance zones. To my surprise at least for one city i.e. Kolkata, the most recommended venues were Chinese restaurants.

On the other hand, in Bengaluru, hotels were most preferred location but, and Indian restaurants were second followed by dessert shops. I know, it may seem that in terms of 'restaurants' Indian cuisine was still on the top but more no. of hotels suggests that more no. of visitors or travellers (tourists or business travellers) are staying near the city centre. The graph for each city is shown below for reference.

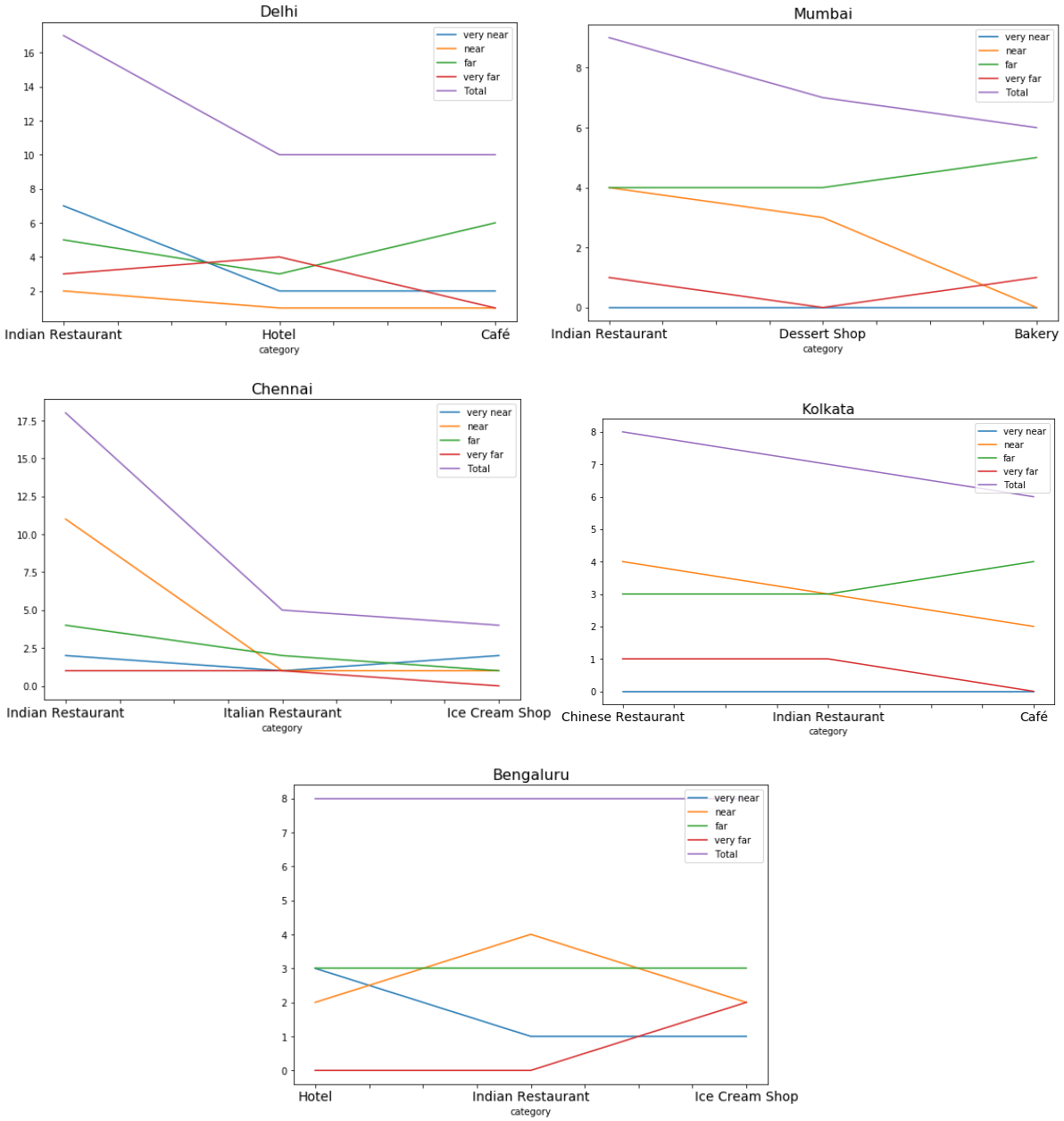

Fig 1: Graphs representing the spatial distribution of most recommended categories.

We can conclude from above that, Indian Cuisine is the most recommended and favoured category among all at all the cities.

3.2. Distribution of venues

Once again, it is always assumed that most recommended and visited venues are usually nearer to the city square or centre. However, this may be case if all cities are planned, but in

real life most cities especially in India are not planned. When trying to establish this distribution, I identified that in almost all cities most of the venues were located either in 2 to 4 KM radius (near) or 4 to 6KM radius (far) region. Now this is an interesting finding and may result because of various factors which could be traffic, lack of commercial space or any other local issue. Table demonstrating this phenomenon are shown below.

City	*Very Near*	*Near*	*Far*	*Very Far*	*Outlier*
Delhi	0.29	0.21	0.32	0.17	0.01
Mumbai	0.05	0.24	0.55	0.16	0.00
Chennai	0.12	0.56	0.21	0.08	0.03
Kolkata	0.05	0.31	0.39	0.22	0.03
Bengaluru	0.14	0.33	0.38	0.15	0.00

Hence, just ensuring the nearest location to the heart of the city might not be the best idea.

3.3. Small Note on Distances and Traffic

Up till now we have done various analysis based on the distance of the venues from the city centre, but one important thing to understand here is these distance are not legal distances or in other words navigational distance from one point to another. The Foursquare API returns the great circle distance or the orthodromic distance. This distance is the shortest distance between two points on a sphere. For more details, you may refer this interesting article.

Having known about above detail we cannot use this distance as an indication of traffic density in any area or on between two locations. Here, the Traffic API which we discussed in Data section will come handy.

3.4. Final feature Selection for Modelling

We have seen during EDA, that not all feature labels including category, distance as well as density represent a true picture of the data available. And, even if end up using anyone of them we will be making huge assumption based on limited amount of spatial data. Hence, the best option here will be to cluster the venues based on their spatial distance with respect to each other and then probably build a recommender system to recommend best cluster based on distance and density of each cluster

4. Model Selection and Evaluation

4.1. Introduction

Cluster analysis or clustering is the most commonly used technique of unsupervised learning. It is used to find data clusters such that each cluster has the most closely matched data. Grouping similar entities together help profile the attributes of different groups. In other words, this will give us insight into underlying patterns of different groups. There are many applications of grouping unlabelled data, for example, you can identify different groups/segments of customers and market each group in a different way to maximize the

revenue. Another example is grouping documents together which belong to the similar topics etc.

4.2. Selecting Modelling algorithm

KMeans and DBScan are most popular clustering algorithms when it comes to clustering and the another one is hierarchal clustering. Density clustering (for example DBSCAN) seems to correspond more to human intuitions of clustering, rather than distance from a central clustering point (for example KMeans). Usually in case of large geospatial data, DBScan is the most trusted algorithm however, for the purpose of our project where the spatial data is limited between only few kilometres we can consider using KMeans algorithm which considers Euclidean distance between the points by default.

K-mean Clustering

1. It starts with K as the input which is how many clusters you want to find. Place K centroids in random locations in your space.
2. Now, using the Euclidean distance between data points and centroids, assign each data point to the cluster which is close to it.
3. Recalculate the cluster centres as a mean of data points assigned to it.
4. Repeat 2 and 3 until no further changes occur.

4.3. Selecting Best K

A cluster centre is the representative of its cluster. The squared distance between each point and its cluster centre is the required variation. The aim of k-means clustering is to find these k clusters and their centres while reducing the total error.

The k can be determined using two methods,

1. **Elbow Method:** This is probably the most well-known method for determining the optimal number of clusters. It is also a bit naive in its approach. Calculate the Within-Cluster-Sum of Squared Errors (WSS) for different values of k, and choose the k for which WSS becomes first starts to diminish. In the plot of WSS-versus-k, this is visible as an elbow.
2. **Silhouette Method:** The silhouette value measures how similar a point is to its own cluster (cohesion) compared to other clusters (separation). The range of the Silhouette value is between +1 and -1. A high value is desirable and indicates that the point is placed in the correct cluster. If many points have a negative Silhouette value, it may indicate that we have created too many or too few clusters. The Silhouette Score reaches its global maximum at the optimal k. This should ideally appear as a peak in the Silhouette Value-versus-k plot.

Silhouette coefficient exhibits a peak characteristic as compared to the gentle bend in the elbow method. This is easier to visualize and reason with.

Another problem with elbow method is it usually fails to find the global maximum and zeroes in on local optima.

Silhouette Coefficient = (x - y)/ max(x, y)

where, y is the mean intra cluster distance: mean distance to the other instances in the same cluster. x depicts mean nearest cluster distance i.e. mean distance to the instances of the next closest cluster.

I used elbow method to first determine the best k for each city and tried validating those with Silhouette Coefficient for each k. Pl. note I used **StandardScaler()** method from **Sklearn** library to first scale the data. Also, to avoid issues with selection of initial centroid, I initiated **init = "k-means++"**. k-means++ selects initial cluster centres for k-mean clustering in a smart way to speed up convergence. Following table present the findings of the above analysis.

CITY	**ELBOW METHOD** K	**SILHOUETTE COEFFICIENT** K
DELHI	4	14
MUMBAI	5	9
CHENNAI	4	8
KOLKATA	4	19
BENGALURU	5	8

The difference shows that; the global optimum widely varies from local optimum. Thus it would be prudent to proceed with global optimums.

4.4. Visualisation of Clusters

I used folium library to plot the clusters on the map. I used following markers for indicating clusters,

- Folium.Circle for plotting circles with radius equal to binned distances from the centre of city.
- Folium.Marker to plot Indian restaurants.
- Folium.CircleMarker to plot venues within a cluster.

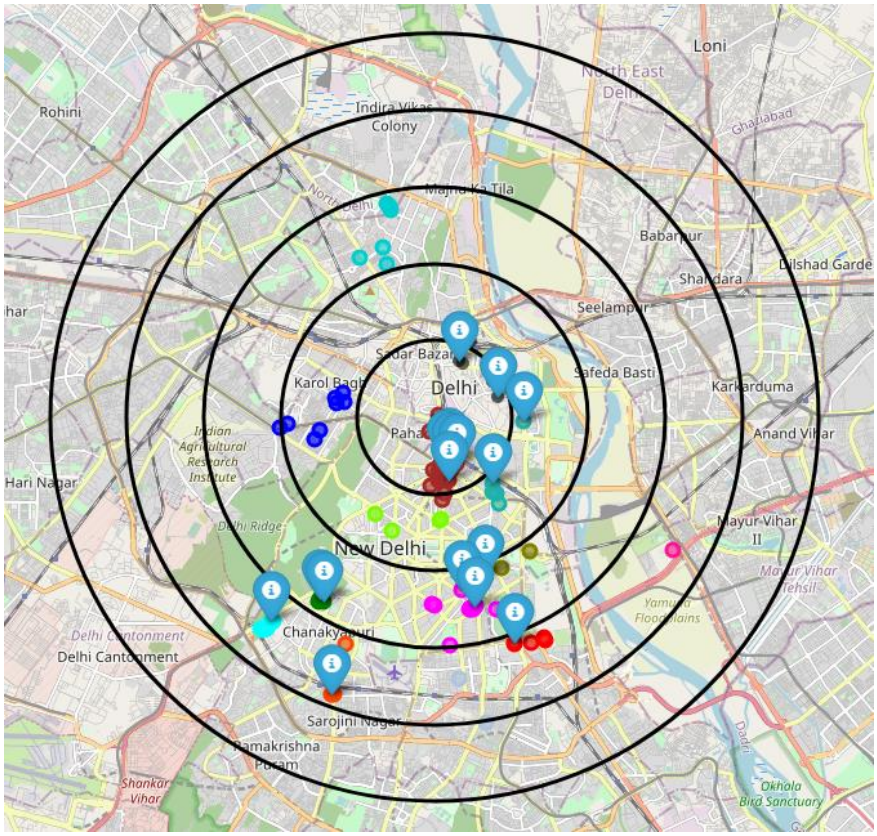

Fig 2: Delhi Cluster Map

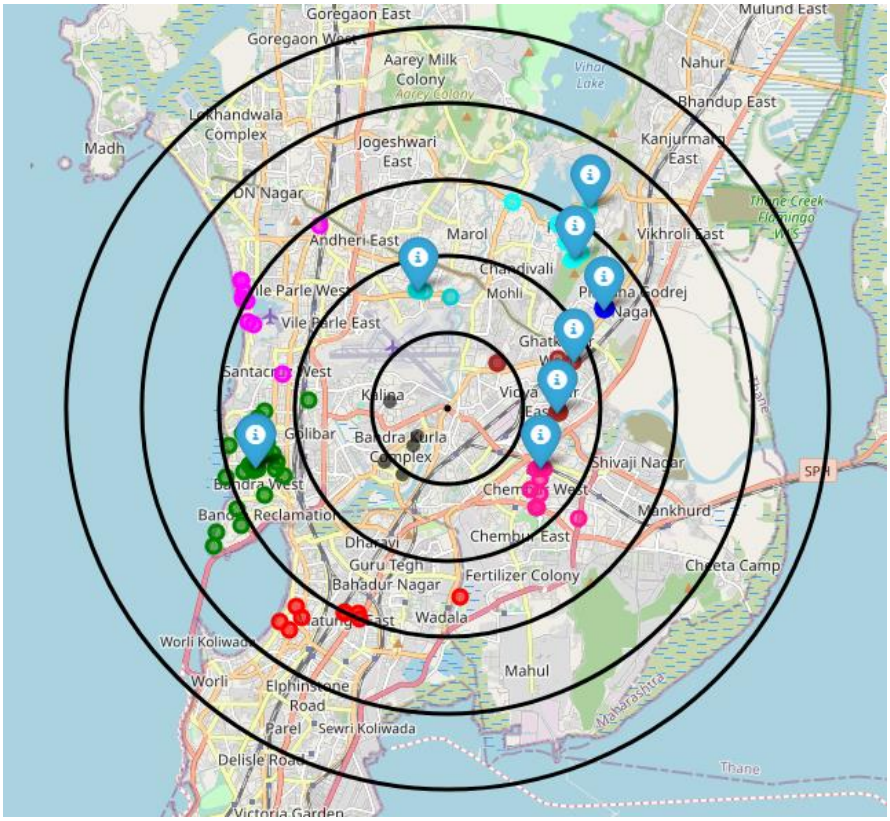

Fig 3: Mumbai Cluster Map

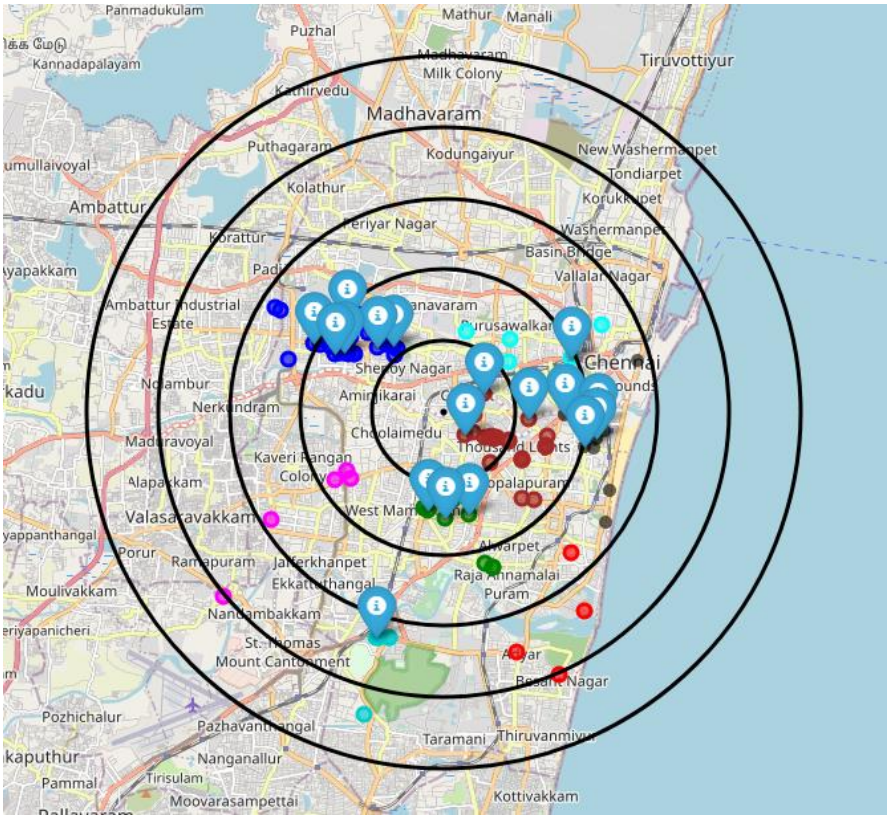

Fig 4: Chennai Cluster Map

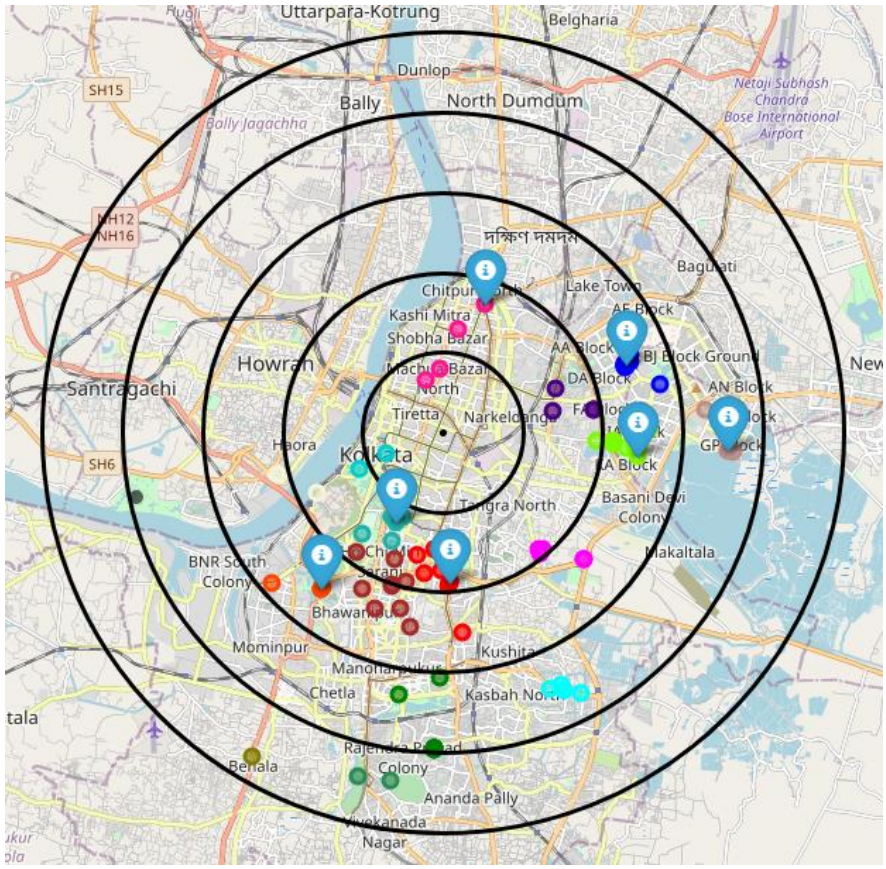

Fig 5: Kolkata Cluster Map

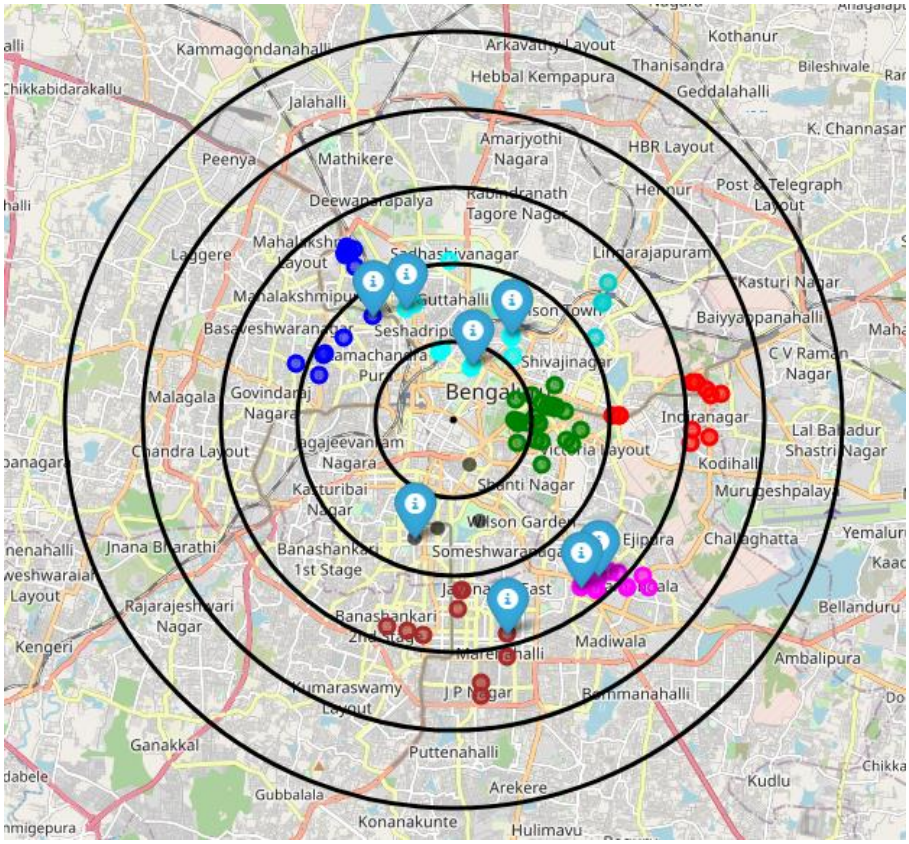

Fig 6: Bengaluru Cluster Map

5. Recommender System for Selecting best cluster

5.1. Setting Criteria for Best Cluster

We will set up following criteria for our recommender system.

1. **The cluster must not have any Indian Restaurant:** As we decided the best restaurant cuisine should be Indian as per our Exploratory Data Analysis. And to reduce competition within the local cluster, it would be prudent to avoid any cluster which already has an Indian restaurant.
2. **The cluster should be 'nearest possible' to the city centre:** Again, the EDA showed that nearest may not be the best thus this parameter should be viewed in conjunction with point 3 mentioned below.
3. **The cluster should have 'highest possible density':** This might sound counter intuitive at first to place our business in the middle of densest cluster. But this could work in our favour as higher density indicates higher footfall, visits and popularity meaning higher visibility for our restaurant. Thus, as explained in above point, our recommender system should choose the optimal distanced cluster based on nearest possible distance and highest possible density.

5.2. Recommending best cluster

First thing first, I eliminated all cluster which already had Indian restaurants in them.

Next, I realised that finding recommender score should be a function of distance and density. But while my one parameter was to be selected for lowest value and other parameter for highest value. So, I needed to scale one of the parameter to ensure the direction of change of slope for each parameter is in one direction.

Thus, I inversed my direction data and multiplied it with cluster data. Providing me with a score which would be highest if the both distance and cluster density (count of venues) are optimised for their minimum and maximum value respectively. Mathematically we can represent it as,

Recommendation Score = f(distance, cluster density) = distance^{-1} *cluster venue counts

So, I grouped each city dataframe by its cluster label and calculated the mean of all parameters and finally joined it with count of venues in each cluster. Then, I implemented functions to calculate the score for each cluster.

After, implementing above solution, I finally was able to select the most suitable clusters of each city.

City	*Cluster label*	*Lat*	*Lng*	*Distance*	*Count*	*Score*
Delhi	**1**	28.645444	77.186265	3008.75	8	0.002659
Mumbai	**7**	19.067727	72.86588	1639.6	5	0.00305
Chennai	**5**	13.043017	80.201891	4742.2	5	0.001054
Kolkata	**3**	22.537695	88.350328	4166.555556	9	0.00216
Bengaluru	**2**	12.972181	77.601005	2251.384615	26	0.011548

6. Predicting Traffic for our selected Cluster

6.1. Background

The final result suggested by our models here unequivocally suggest that cluster 2 of Bengaluru is the most suitable cluster for the new restaurant. However, where exactly this restaurant shall be placed in the cluster depends upon various other factors, one of them might be traffic information. How far it is from city centre? How easily people can reach to the location? To which all places our business can deliver? Can we suggest at what time people should start in case of reservations?

Fortunately, we have Traffic API provided by Bing Maps which calculates for us Isochrone.

The Bing Maps Isochrone API provides time-specific, isoline polygons for the distance that is reachable from a given location and supports multiple modes of transportation (i.e., driving, walking, and public transit). Use this solution to plan the area that can be reached from a designated starting point within a set time period.

- **Store Locators** – Show me all locations that are within 10 minutes of a user.
- **Stolen Vehicle Recovery** – Where could a vehicle have travelled to since it was stolen.
- **Real Estate** – Limit search results such that only those that are within the users preferred commute preferences to work. For example, show me all homes that are within a 30-minute drive of work.

More details about this API is available here.

6.2. Calculating Isochrone

I chose the mean of latitude and longitude of best cluster as base location for calling this API. In real life scenario, businesses can choose the potential store locations to get results.

Few of the parameters that I selected for making this API call include,

- **maxTime:** The maximum travel time in the specified time units in which the isochrone polygon is generated. I have set it as 30 Minutes.
- **dateTime:** When a **maxTime** value is specified, predictive traffic data is used to calculate the best isochrone route for the specified date time. I have provided two values,
    - **25th Jan 2020 18:00 hrs** as it was a holiday weekend and presumably peak traffic time. (marked in yellow)
    - **22nd Jan 2020 18:00 hrs** as it was mid-week (marked in red)
- **Optimize:** Specifies what parameters to use to optimize the isochrone route. One of the following values.
    - **time [default]:** The route is calculated to minimize the time. Traffic information is not used. Use with maxTime.
    - **timeWithTraffic:** The route is calculated to minimize the time and uses current or predictive traffic information depending on if a dateTime value is specified. Use with maxTime.
    - I have chosen this as **timeWithTraffic**

Also,

- I have plotted **folium.Circle** for binned distance now choosing mean lat and long as base location.
- folim.Marker to denote mean lat and long location
- folium.CircleMarker to denote all cluster venues.
- And folium.PolyLine for plotting the isochrone diagram.

We can see that, the traffic density on a weekend is actually slightly less than that on week days as isochrone area for weekend is slightly larger.

The map indicates that the restaurant at mean location will be able to cater the needs to people in the radius of 4KM and in certain cases up to 6 KM.

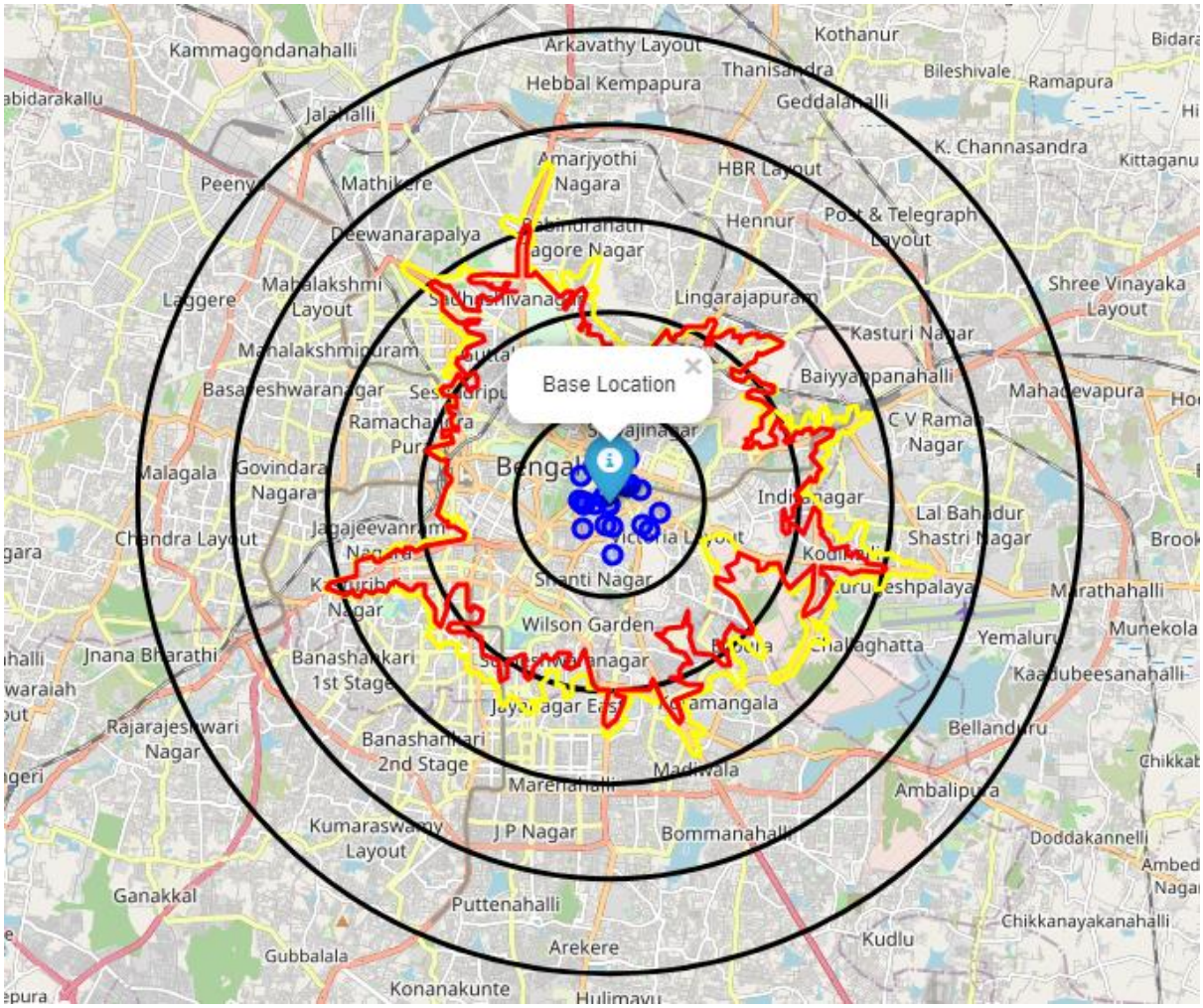

Fig 7: Isochrone diagram for selected cluster.

7. Conclusions

In my project, I have tried to predict the best location to open a new restaurant along with the recommendation on which cuisine they should offer. I found through exploratory data analysis that, regardless of the city, Indian cuisine is still the most popular among people.

Another revelation was that, proximity to the city centre or square doesn't really translates into the best location for business.

I used KMeans clustering algorithm on spatial data and silhouette coefficient to determine most distinct clusters with least intra-cluster distance and most inter-cluster distance.

Finally, I presented a recommender system to recommend best location based on count of venues in a particular cluster, distance of cluster from city centre and absence of Indian restaurants in the cluster.

Finally, the cluster 2 of Bengaluru city was suggested as best cluster to place our business based on our modelling and recommendation. I hope this model and analysis will help new business in identifying the best location within a city for their new businesses.

8. Future directions

The analysis produced here based on the recommendations suggested by the Foursquare API, spatial distance between venues and type of business that are already present. However, the actual location of the restaurant may vary based on the availability of the space, rent rates, proximity to roads or highways etc.

Another interesting study could be analysis of the crime data in the locality and use it as one of the modelling parameters.