

# Capston Project : The Battle of Neighbourhoods

July 12, 2020

Jogi Goghari, Ahmedabad , Gujarat , India

## 1. Introduction

Ahmedabad is the city of peace loving people, it is one of the metro city of India and more addition it is economic capital of Gujarat state. Ahmedabad is also name as one of the Heritage city of India.

The official language of Ahmedabad and the one that is most widely spoken is Gujarati. However, Hindi and English are also spoken as a formal language within businesses and government agencies. Over last decades it is continuously grow because of the city's important role in government and commercial business.

With its diverse culture, comes diverse food items. There are many restaurants in New Delhi City, each belonging to different categories like Gujarati, Panjabi, South Indian ,Chinese , Italian , French etc. So as part of this project , we will list and visualise all major parts of Ahmedabad City .

### 1.1 Problem Description:

The Objective of the project is to Explore and Analyse the neighbourhood for restaurants and to select best location in the Ahmedabad city related to open a new restaurant.

A Restaurant is a business which prepares and serves food and drink to customers in return for money, either paid before the meal, after the meal, or with an open account. The Ahmedabad City is famous for its excellent cuisine. It's food culture includes an array of individual states of India as well as some international cuisines.

### 1.3 Targeted Audience:

Ahmedabad is a city were 55. 7 Lakh people are living. Concern of this project will be going to target interested people form the city who are willing or wanted to start restaurant project in the Ahmedabad City area. This will also going to help current restaurant owners to how to perform Post COVID-19 condition. Like as home delivery pickup condition.

### 1.4 Data Requirement & Sources:

#### 1.4.1 Required data:

*List of neighbourhoods in the Ahmedabad City.* Witch will help us to give scope of the areas for restaurants.

Latitude and Longitudes of the extracted neighbourhoods data. This will act as the input parameters to the FOURSQUARE API to explore a neighbourhoods.

Cluster on neighbourhoods depending on the frequency of given radius of a particular data.

### 1.4.2 Sources and methods to get the data:

1. List of neighbourhoods in the Ahmedabad City.  
Source : <https://www.kaggle.com/rabhar/data>  
We are extracting Kaggle zomato Ahmedabad Restaurant data
2. We will then get the Latitude and Longitude of each neighborhood using geocoder library and attach these coordinates to our neighborhood data
3. Then with the help of Foursquare api calls we will send the coordinates of each neighbourhoods and get venues details of restaurants

This project will be completed with the help of below skills and methods:

- Data science skills,
- Data cleaning,
- Exploring,
- Analyzing,
- Visualizing using folium maps in particular.

More adding in this using Geocoder library to get the latitude and longitudes for neighborhoods. We will also take help from Foursquare API to get the Restaurants details which are near to each neighborhood.

And finally using a Machine learning technique 'K-means clustering' to cluster restaurants into different categories.

## 2. Data importing and Data Processing

### 2.1 Firstly we are importing related initial libraries

```
[28]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

### 2.2 Reading Relevant File

Now We will add our downloaded Kaggle File in to this notebook

```
[29]: df = pd.read_csv('zomato_restaurants_in_india.csv')
```

```
[30]: df.head(2)
```

	res_id	name	establishment	url	address	city	city_id	locality	latitude	longitude	price_range	currency
0	3400299	Bikanervala	[Quick Bites]	<a href="https://www.zomato.com/agra/bikanervala-likhanda">https://www.zomato.com/agra/bikanervala-likhanda...</a>	Kalyani Point, Near Tuls Cinema, Bypass Road, ...	Agra	34	Khandari	27.211450	76.002381	₹	Rs
1	3400005	Mama Chicken Mama Frankie House	[Quick Bites]	<a href="https://www.zomato.com/agra/mama-chicken-mama-...">https://www.zomato.com/agra/mama-chicken-mama-...</a>	Main Market, Sadar Bazaar, Agra Cantt, Agra	Agra	34	Agra Cantt	27.180569	78.011563	₹	Rs

2 rows × 13 columns

hape will is : (211944, 26)

Below will show us how many columns are there and how what are their types

```
[32]: df.dtypes

[32]: res_id      int64
      name       object
      establishment object
      url        object
      address    object
      city       object
      city_id    int64
      locality   object
      latitude   float64
      longitude  float64
      zipcode    object
      country_id int64
      locality_verbose object
      cuisines   object
      timings    object
      average_cost_for_two int64
      price_range int64
      currency   object
      highlights object
      aggregate_rating float64
      rating_text object
      votes      int64
      photo_count int64
      opentable_support float64
      delivery   int64
      takeaway   int64
      dtype: object
```

Now we will only filtered data which is related to Ahmedabad City

### 3. Filtering only Ahmedabad City from column 'City'

```
[33]: # Taking only the Ahmedabad in city column and dropping the rest of the cities
df.drop(df.index[df['city'] != 'Ahmedabad'], inplace = True)
df.head(2)
```

```
[33]:
```

	res_id	name	establishment	url	address	city	city_id	locality	latitude	longitude	...	price_range	currency	high
2622	18757684	Ph Se Food	[Casual Dining]	https://www.zomato.com/ahmedabad/ph-se-food-bo...	205-206, Silve Radance, Sindhu Bhawan Road, P...	Ahmedabad	11	Bodakdev	23.036755	72.510887	...	3	Rs.	Tak Ava
2623	18658177	Millhouse	[Casual Dining]	https://www.zomato.com/ahmedabad/millhouse-1-v...	3rd Floor, Alpha One Mall, Vastrepur, Ahmedabad	Ahmedabad	11	Ahmedabad One Mall, Vastrepur	23.039770	72.531508	...	3	Rs.	A

2 rows × 26 columns

```
[34]: df.shape
```

```
[34]: (3162, 26)
```

After filtering Ahmedabad City Area in dataframe

```
[35]: # Reset the index and dropping the previous index
df = df.reset_index(drop=True)
df.head(2)
```

Now we will remove non related data form the dataframe

```
[36]: df1 = df.drop(['url', 'city', 'zipcode', 'city_id', 'country_id', 'timings', 'average_cost_for_two', 'price_range', 'currency', 'highlights', 'rating_text', 'p
#df1 = df.drop(['delivery'], axis=1)
df1.head(2)
```

	res_id	name	establishment	address	locality	latitude	longitude	locality_verbose	cuisines	aggregate_rating	votes
0	18757684	Ph Se Food	[Casual Dining]	205-206, Shree Radhance, Sindhu Bhawan Road, P...	Bodakdev	23.038755	72.510687	Bodakdev, Ahmedabad	North Indian, Chinese, Fast Food	4.5	1434
1	18659777	Mithouse	[Casual Dining]	3rd Floor, Alpha One Mall, Vastapur, Ahmedabad	Ahmedabad One Mall, Vastapur	23.039770	72.531508	Ahmedabad One Mall, Vastapur, Ahmedabad	Continental, Italian	4.5	1330

```
[37]: df1.shape
[38]: (3162, 11)
```

After removal of non-related columns, we will remove duplicate values from the dataframe

```
[38]: df2 = df1.drop_duplicates()
df2
```

Shape will be (1247,11) which means 1247 rows and 11 columns

Now we will also remove Zero rated rows

```
[40]: df2.drop(df2[df2['aggregate_rating'] <= 0].index, inplace = True)
df2
#df.drop(df[df['Age'] < 25].index, inplace = True)
```

Shape will be (1078, 11)

Chaging the title and placing reset for index condition

```
[42]: df2.columns=['Restaurant_ID', 'Name', 'Type', 'Address', 'Locality', 'Latitude', 'Longitude', 'Locality_verbose', 'Cuisines', 'Aggregate Rating', 'Votes']
df2.head(2)
#df1.dtypes
```

```
[41]: # Reset the index and dropping the previous index
df2 = df2.reset_index(drop=True)
df2.head()
```

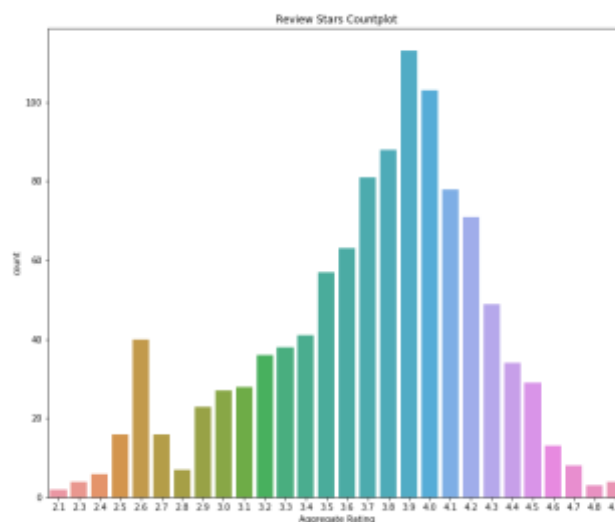
```
[43]:
```

	Restaurant_ID	Name	Type	Address	Locality	Latitude	Longitude	Locality_verbose	Cuisines	Aggregate Rating	Votes
0	18757684	Ph Se Food	['Casual Dining']	205-206, Silver Radiance, Sindhu Bhawan Road, P...	Bodakdev	23.038755	72.510687	Bodakdev, Ahmedabad	North Indian, Chinese, Fast Food	4.5	1434
1	18658177	Milhouse	['Casual Dining']	3rd Floor, Alpha One Mall, Vastrapur, Ahmedabad	Ahmedabad One Mall, Vastrapur	23.039770	72.531508	Ahmedabad One Mall, Vastrapur, Ahmedabad	Continental, Italian	4.5	1530
2	18663856	The Red Bistro	['Casual Dining']	Ameida, Sindhu Bhawan Road, Off SG Road, Boda...	Bodakdev	23.039652	72.509181	Bodakdev, Ahmedabad	Mexican, Italian, North Indian	4.6	695
3	18818827	Kadak Bhagat	['Casual Dining']	1, First Floor, Dev Aurum Complex, Deer Circle,...	Prahlad Nagar	23.012411	72.514392	Prahlad Nagar, Ahmedabad	North Indian, Chinese, Fast Food	4.4	977
4	18718880	Urban Krichdi	['Casual Dining']	A-108, Shivalik Plaza, Opposite AMA, IIM Road,...	Vastrapur	23.028315	72.542458	Vastrapur, Ahmedabad	North Indian, Continental	4.7	619

### 3. Analysing And Clustering

#### 3. 1 Analysing Aggregate Rating for Restaurants :

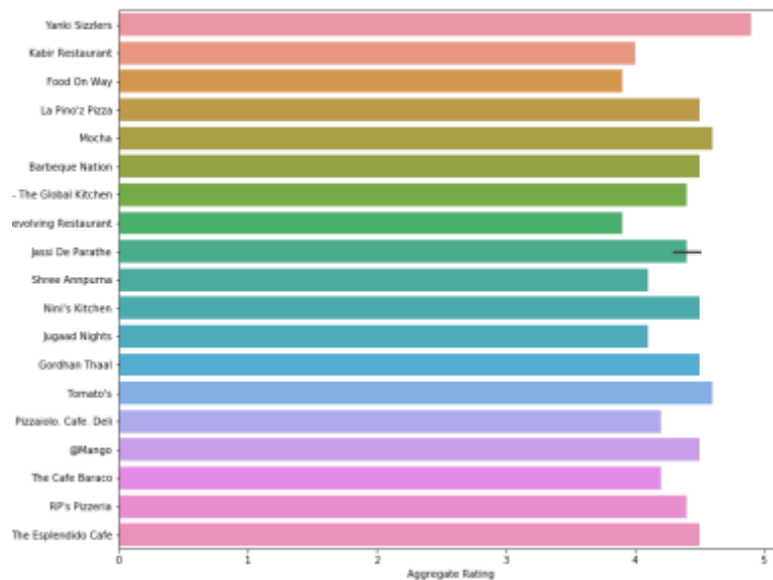
```
[151]: import seaborn as sns
fig, ax = plt.subplots(figsize=(12,10))
sns.countplot(df2['Aggregate Rating'], ax=ax)
plt.title('Review Stars Countplot')
plt.savefig('stars.png')
plt.show()
```



#### 3.2 Analysing Top Voted and Aggregate Rating for Restaurants :

```
[46]: top_restaurants = df2.sort_values(by=['Votes', 'Aggregate Rating'], ascending=False)[:20]
top_restaurants.head(20)
```

```
[21]: fig, ax = plt.subplots(figsize=(12,10))
sns.barplot(x = 'Aggregate Rating', y = 'Name', data=top_restaurants, ax= ax);
plt.savefig('top20_restaurants.png')
plt.show()
```



### 3.3. Analysing Best and Aggregate Rating for Restaurants :

```
[23]: plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('The highest rated resturant in top 10 locality of Ahmedabad City')
#On x-axis

#giving a bar plot
df2.groupby('Locality')['Aggregate Rating'].mean().nlargest(10).plot(kind='bar')

Locality_verbose

plt.xlabel('Resturant Locality in Ahmedabad')
#On y-axis
plt.ylabel('Aggregate Rating')
#displays the plot
plt.savefig('top10_restaurants by locality.png')
plt.show()
```

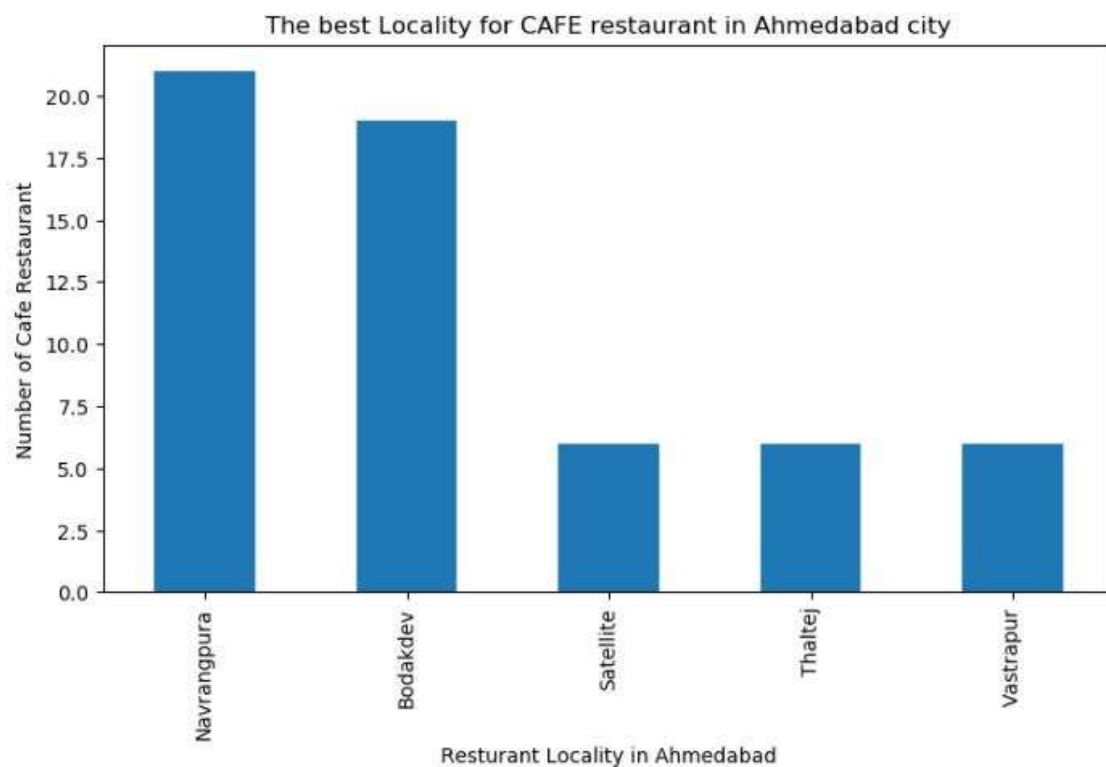
### 3.4. Analysing Best Locality for CAFÉ :

```
[25]: import matplotlib.pyplot as plt
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('The best Locality for CAFE restaurant in Ahmedabad city')
#On x-axis

#giving a bar plot
df2[df2['Cuisines'].str.startswith('Cafe')].groupby('Locality')['Name'].count().nlargest(5).plot(kind='bar')

plt.xlabel('Resturant Locality in Ahmedabad')
#On y-axis
plt.ylabel('Number of Cafe Restaurant')

plt.savefig('Best_locality_CAFE.png')
#displays the plot
plt.show()
```



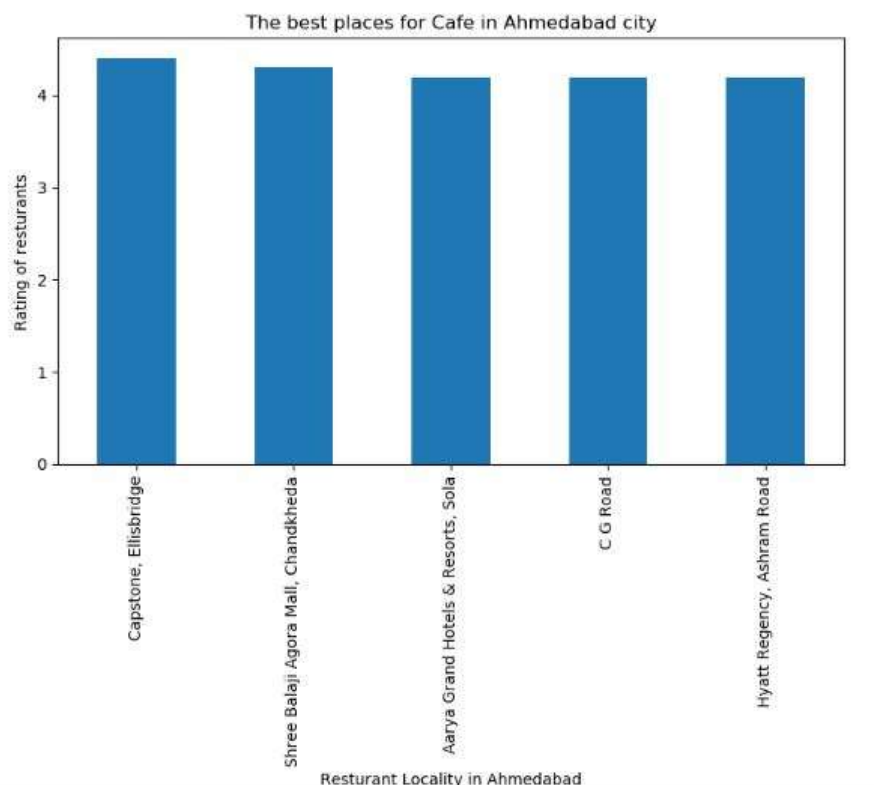
### 5. Analysing Best Location for CAFÉ :

```
[26]: import matplotlib.pyplot as plt
plt.figure(figsize=(9,5), dpi = 100)
# title
plt.title('The best places for Cafe in Ahmedabad city')
#On x-axis

#giving a bar plot
df2[df2['Cuisines'].str.startswith('Cafe')].groupby('Locality')['Aggregate Rating'].mean().nlargest(5).plot(kind='bar')

plt.xlabel('Resturant Locality in Ahmedabad')
#On y-axis
plt.ylabel('Rating of resturants')

plt.savefig('Best_place_CAFE.png')
#displays the plot
plt.show()
```



#### 4. KMean and Clustering :

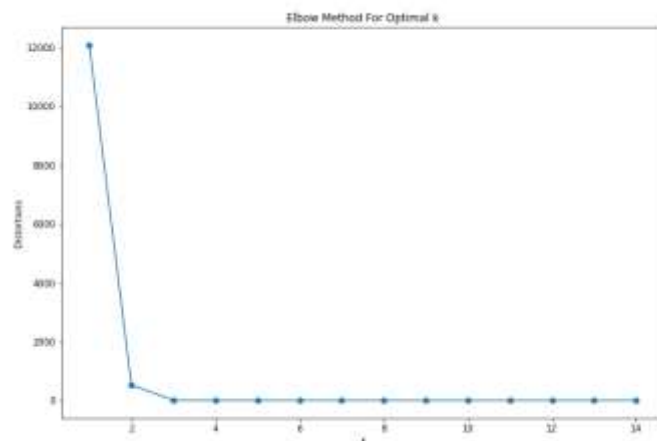
```
[56]: from sklearn.cluster import KMeans
      from sklearn.metrics import silhouette_score

      df3=df2
      # Elbow method to determine the number of K in Kmeans Clustering
      coords = df3[['Longitude','Latitude']]

      distortions = []
      K = range(1,15)
      for k in K:
          kmeansModel = KMeans(n_clusters=k)
          kmeansModel = kmeansModel.fit(coords)
          distortions.append(kmeansModel.inertia_)

[57]: fig, ax = plt.subplots(figsize=(12, 8))
      plt.plot(K, distortions, marker='o')
      plt.xlabel('k')
      plt.ylabel('Distortions')
      plt.title('Elbow Method For Optimal k')
      #plt.savefig('elbow.png')
      plt.show()
```





We can see that KMean will remain constant after its valuation is 3.

So our cluster value will be 3.

Now we will identify all the related restaurants on Ahmedabad City Map

```
[*]: import folium

Ahmedabad_Rest = folium.Map(location=[23.05, 72.52], zoom_start=12)

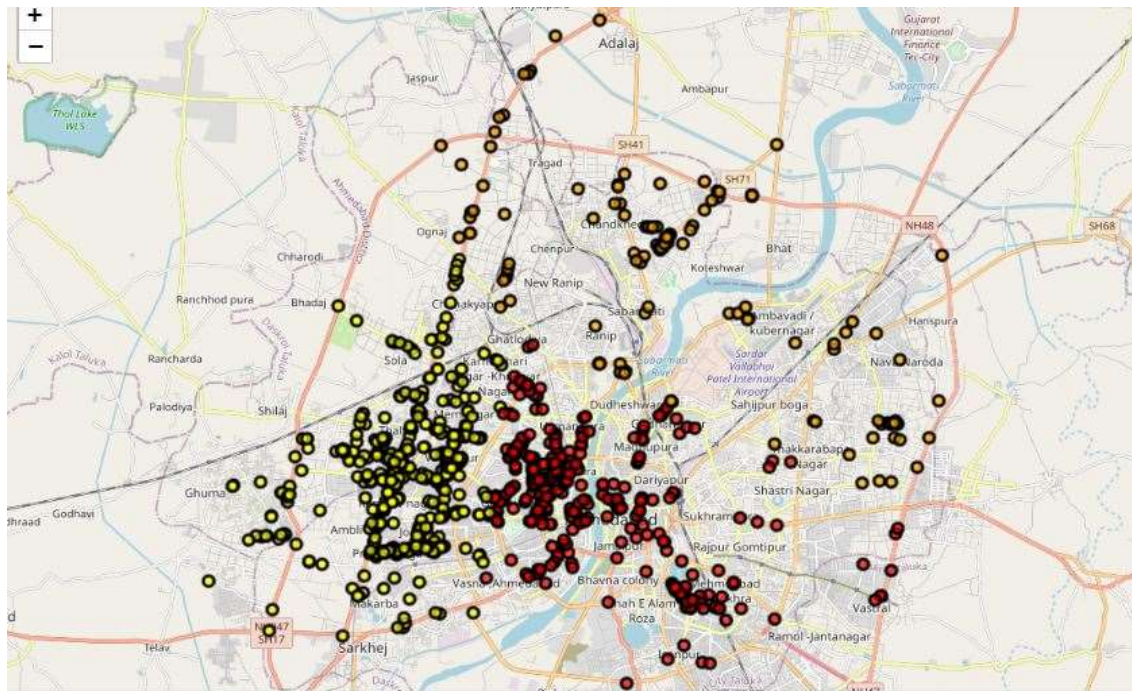
X = df3['Latitude']
Y = df3['Longitude']
Z = np.stack((X, Y), axis=1)

kmeans = KMeans(n_clusters=5, random_state=0).fit(Z)

clusters = kmeans.labels_
colors = ['red', 'green', 'blue', 'yellow', 'orange']
df3['Cluster'] = clusters

for latitude, longitude, Locality, cluster in zip(df3['Latitude'], df3['Longitude'], df3['Locality'], df3['Cluster']):
    label = folium.Popup(Locality, parse_html=True)
    folium.CircleMarker(
        [latitude, longitude],
        radius=5,
        popup=label,
        color='black',
        fill=True,
        fill_color=colors[cluster],
        fill_opacity=0.7).add_to(Ahmedabad_Rest)

Ahmedabad_Rest
```



```
[71]: df3.head()
```

	Restaurant_ID	Name	Type	Address	Locality	Latitude	Longitude	Locality_verbose	Cuisines	Aggregate Rating	Votes	Cluster
0	18757684	Pt Se Food	[Casual Dining]	205-206, Silver Radiance, Sindhu Bhawan Road, P...	Bodakdev	23.036755	72.510887	Bodakdev, Ahmedabad	North Indian, Chinese, Fast Food	4.5	1434	5
1	18658177	Mithouse	[Casual Dining]	3rd Floor, Alpha One Mall, Vastrapur, Ahmedabad	Ahmedabad One Mall, Vastrapur	23.039770	72.531508	Ahmedabad One Mall, Vastrapur, Ahmedabad	Continental, Italian	4.5	1530	3
2	18663856	The Red Bistro	[Casual Dining]	Ameida, Sindhu Bhawan Road, Off SG Road, Boda...	Bodakdev	23.039852	72.509181	Bodakdev, Ahmedabad	Mexican, Italian, North Indian	4.6	695	5
3	18819827	Kadak Bhagat	[Casual Dining]	1, First Floor, Dev Aurum Complex, Deer Circle...	Prahlad Nagar	23.012411	72.514392	Prahlad Nagar, Ahmedabad	North Indian, Chinese, Fast Food	4.4	977	3
4	18718880	Urban Khichdi	[Casual Dining]	A-103, Shivaji Plaza, Opposite AMA, IIM Road...	Vastrapur	23.028315	72.542458	Vastrapur, Ahmedabad	North Indian, Continental	4.7	619	0

Here we can see that one extra column is been added which is of 'Cluster'. This value of cluster is indicating that our data is lying in which number of cluster.

Data transforming :

In which we are doing locality base on groupby.

```
[96]: df3_Loc = df3.groupby('Locality').count()['Name'].to_frame()
df3_Res_rating = df3.groupby('Locality')['Aggregate Rating'].mean().to_frame()
df3_Cuisines = df3.groupby(['Locality'])['Cuisines'].agg(', '.join).reset_index()
#df2_R = df2.groupby(['Locality'])['Rating text'].unique().agg(', '.join).reset_index()
df3_V = df3.groupby(['Locality'])['Votes'].sum().to_frame()
df3_Lat = df3.groupby('Locality').mean()['Latitude'].to_frame()
df3_Lng = df3.groupby('Locality').mean()['Longitude'].to_frame()
df3_Final = pd.merge(df3_Lat, df3_Lng, on='Locality').merge(df3_Loc, on='Locality').merge(df3_Cuisines, on='Locality').merge(df3_Res_rating, on='Locality').merge(df3_V, on='Locality')

[97]: print("size :", df3_Final.shape)
df3_Final

size : (100, 7)
```

	Locality	Latitude	Longitude	Name	Cuisines	Aggregate Rating	Votes
0	Sigma Legacy Building, Vastrapur	23.027271	72.544209	2	North Indian, BBQ, Desserts, Bakery	3.900000	979
1	10 Acres Mall, Kankaria	23.014879	72.596397	2	Beverages, Cafe, Sandwich, Fast Food, Burger, ...	3.150000	432
2	4D Square Mall, Chandkheda	23.102908	72.595708	9	Pizza, Fast Food, Pizza, Fast Food, North Indi...	3.611111	4026
3	Aanya Grand Hotels & Resorts, Soli	23.075418	72.522681	2	North Indian, Cafe	3.950000	226
4	Acropolis Mall, Thaltej	23.048584	72.516107	4	North Indian, Mughlai, Continental, Desserts, ...	3.675000	633

After adding groupby number we will implement our Foursquare API for clustering process in which first we will implement my personal CLIENT\_ID ,CLIENT\_SECRET.

```

## create a function to repeat the same process to all the Locality in Ahmedabad City
def getNearbyVenues(names, latitudes, longitudes, radius=500,LIMIT = 100):

    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):
        print(name)

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            LIMIT)

        # make the GET request
        results = requests.get(url).json()["response"]["groups"][0]["items"]

        # return only relevant information for each nearby venue
        venues_list.append([
            name,
            lat,
            lng,
            v['venue']['name'],
            v['venue']['location']['lat'],
            v['venue']['location']['lng'],
            v['venue']['categories'][0]['name'] for v in results])

    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Locality',
                            'Locality Latitude',
                            'Locality Longitude',
                            'Venue',
                            'Venue Latitude',
                            'Venue Longitude',
                            'Venue Category']

    return(nearby_venues)

```

```

[107]: import requests # library to handle requests

# find the venues in all New Delhi Locality
new_Ahmedabad_venues = getNearbyVenues(names=df3_final['Locality'],
                                         latitudes=df3_final['Latitude'],
                                         longitudes=df3_final['Longitude']
                                         )

```

```

[113]: print("Size of New Venues :",new_Ahmedabad_venues.shape)
new_Ahmedabad_venues.head()

```

Size of New Venues : (942, 7)

```

[113]:

```

	Locality	Locality Latitude	Locality Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Sigma Legacy Building, Vastrapur	23.027271	72.544209	Mint Route	23.027645	72.544113	Vegetarian / Vegan Restaurant
1	Sigma Legacy Building, Vastrapur	23.027271	72.544209	Birmies	23.027362	72.544465	Indian Restaurant
2	Sigma Legacy Building, Vastrapur	23.027271	72.544209	Dangee Dums	23.027597	72.544235	Dessert Shop
3	Sigma Legacy Building, Vastrapur	23.027271	72.544209	Subway	23.028550	72.542598	Sandwich Place
4	Sigma Legacy Building, Vastrapur	23.027271	72.544209	SandwichworkZ	23.028640	72.542761	Café

```

[114]: new_Ahmedabad_venues.groupby('Locality').count()

```

```

[114]:

```

	Locality	Locality Latitude	Locality Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
	Sigma Legacy Building, Vastrapur	19	19	19	19	19	19
	10 Acres Mall, Kankaria	7	7	7	7	7	7
	4D Square Mall, Chandkheda	8	8	8	8	8	8
	Aarya Grand Hotels & Resorts, Sola	5	5	5	5	5	5
	Acropolis Mall, Thaltej	11	11	11	11	11	11
	...	...	...	...	...	...	...
	Vastrapur	25	25	25	25	25	25

```
[115]: print('There are {} uniques categories.'.format(len(new_Ahmedabad_venues['Venue Category'].unique())))
There are 97 uniques categories.

[116]: ## Analyze Each Locality

# one hot encoding
new_Ahmedabad_onehot = pd.get_dummies(new_Ahmedabad_venues[['Venue Category']], prefix="", prefix_sep="")

# add locality column back to dataframe
new_Ahmedabad_onehot['Locality'] = new_Ahmedabad_venues['Locality']

# move locality column to the first column
column_list = new_Ahmedabad_onehot.columns.tolist()
column_number = int(column_list.index('Locality'))
column_list = [column_list[column_number]] + column_list[column_number+1:]
new_Ahmedabad_onehot = new_Ahmedabad_onehot[column_list]

new_Ahmedabad_onehot.head()

[123]: ## print each Locality along with the top 5 most common venues

num_top_venues = 5

for hood in Ahmedabad_grouped['Locality']:
    print("-----"+hood+"-----")
    temp = Ahmedabad_grouped[Ahmedabad_grouped['Locality'] == hood].T.reset_index()
    temp.columns = ['venue','freq']
    temp = temp.iloc[1:]
    temp['freq'] = temp['freq'].astype(float)
    temp = temp.round({'freq': 2})
    print(temp.sort_values('freq', ascending=False).reset_index(drop=True).head(num_top_venues))
    print('\n')
```

Above process will help us to identifying for our most common data.

```
[124]: ## put that into a pandas dataframe
## First, write a function to sort the venues in descending order.

def return_most_common_venues(row, num_top_venues):
    row_categories = row.iloc[1:]
    row_categories_sorted = row_categories.sort_values(ascending=False)

    return row_categories_sorted.index.values[0:num_top_venues]

[125]: ## create the new dataframe and display the top 10 venues for each Locality.

num_top_venues = 10

indicators = ['st', 'nd', 'rd']

# create columns according to number of top venues
columns = ['Locality']
for ind in np.arange(num_top_venues):
    try:
        columns.append('{}{} Most Common Venue'.format(ind+1, indicators[ind]))
    except:
        columns.append('{}th Most Common Venue'.format(ind+1))

# create a new dataframe
Locality_venues_sorted = pd.DataFrame(columns=columns)
Locality_venues_sorted['Locality'] = Ahmedabad_grouped['Locality']

for ind in np.arange(Ahmedabad_grouped.shape[0]):
    Locality_venues_sorted.iloc[ind, 1:] = return_most_common_venues(Ahmedabad_grouped.iloc[ind, :], num_top_venues)

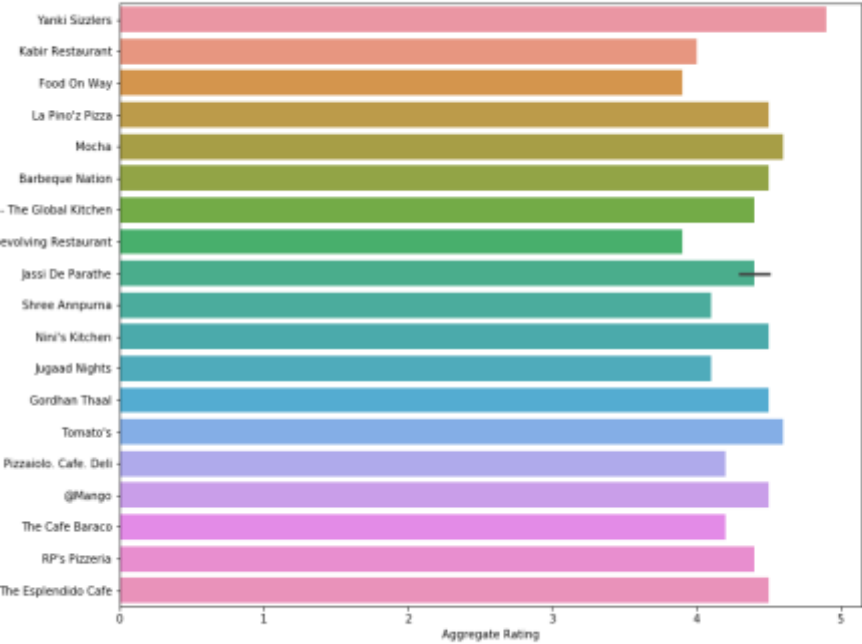
Locality_venues_sorted
```



Now we will convert our identified data in to dataframe.  
Below will be our result.

[[125]]

	Locality	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	Sigma Legacy Building, Vasthapur	Café	Indian Restaurant	Bakery	Pizza Place	Sandwich Place	Breakfast Spot	Restaurant	Tea Room	Mexican Restaurant	Vegetarian / Vegan Restaurant
1	10 Acres Mall, Kankaria	Multiplex	Shopping Mall	Fast Food Restaurant	Clothing Store	Bus Station	Zoo	Dessert Shop	Diner	Donut Shop	Electronics Store
2	4D Square Mall, Chandkheda	Snack Place	Pizza Place	Arcade	North Indian Restaurant	Coffee Shop	Multiplex	Sandwich Place	Café	Zoo	Diner
3	Aanya Grand Hotels & Resorts, Sola	Hotel	Snack Place	Pool	Fast Food Restaurant	Dance Studio	Department Store	Dessert Shop	Diner	Donut Shop	Electronics Store
4	Acropolis Mall, Thaltej	Indian Restaurant	Gym / Fitness Center	Multiplex	Sandwich Place	Café	Shopping Mall	Snack Place	Mediterranean Restaurant	Ice Cream Shop	Furniture / Home Store
...	...	...	...	...	...	...	...	...	...	...	...
96	Vasthapur	Clothing Store	Pizza Place	Indian Restaurant	Fast Food Restaurant	Café	Shopping Mall	Snack Place	Food Court	Donut Shop	Men's Store
97	Vejalpur	ATM	Pool Hall	American Restaurant	Food & Drink Shop	Dessert Shop	Diner	Donut Shop	Electronics Store	Event Space	Falafel Restaurant
98	Venus Atlantis, Prahlad Nagar	Café	Vegetarian / Vegan Restaurant	Indian Restaurant	BBQ Joint	Coffee Shop	Restaurant	Hotel	Gas Station	Yoga Studio	Ice Cream Shop
99	Vittal Mall, Chandkheda	Pizza Place	Coffee Shop	North Indian Restaurant	Multiplex	Sandwich Place	Café	Snack Place	Zoo	Dance Studio	Dessert Shop



```
[46]: ## Cluster Locality
      ## Run k-means to cluster the Locality into 3 clusters.

      # set number of clusters
      kclusters = 5

      Ahmedabad_clustering = Ahmedabad_grouped.drop('Locality', 1)

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

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

[46]: (101,)

[47]: # add clustering labels
      Ahmedabad_merged = df3_final.head(101)
      Ahmedabad_merged['Cluster Labels'] = kmeans.labels_

      # merge New_Delhi_grouped with df_Chinese to add Latitude/Longitude for each Locality
      Ahmedabad_merged = Ahmedabad_merged.join(Locality_venues_sorted.set_index('Locality'), on='Locality')

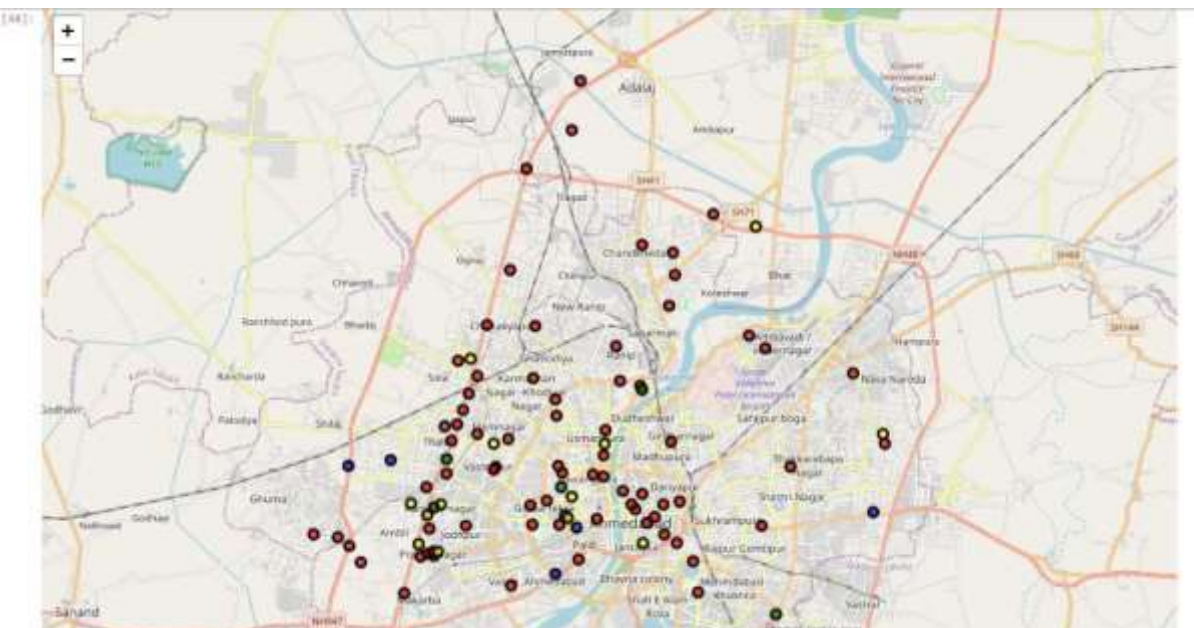
      Ahmedabad_merged.head()

[48]: # create final map
      map_clusters = folium.Map(location=[latitude, longitude], zoom_start=10)

      # 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]
      colors = ['red', 'green', 'blue', 'yellow', 'orange']

      # add markers to the map
      markers_colors = []
      for lat, lon, poi, cluster in zip(Ahmedabad_merged['Latitude'], Ahmedabad_merged['Longitude'], Ahmedabad_merged['Locality'],
      label = folium.Popup(str(poi) + ' Cluster ' + str(cluster), parse_html=True)
      folium.CircleMarker(
          [lat, lon],
          radius=5,
          popup=label,
          color='black',
          fill=True,
          fill_color=colors[cluster],
          fill_opacity=0.7).add_to(map_clusters)

      map_clusters
```



Above Map is showing clustered data from Top places in Ahmedabad

Now final Step will be to Identify 3 clusters.

```
[138]: ## Examine Clusters

      ## Cluster 1
      Ahmedabad_merged.loc[Ahmedabad_merged['Cluster Labels'] == 0, Ahmedabad_merged.columns[[1] + list(range(3, Ahmedabad_merged.shape[1]))]]

[140]: ## Examine Clusters

      ## Cluster 2
      Ahmedabad_merged.loc[Ahmedabad_merged['Cluster Labels'] == 1, Ahmedabad_merged.columns[[1] + list(range(3, Ahmedabad_merged.shape[1]))]]

[142]: ## Examine Clusters

      ## Cluster 3
      Ahmedabad_merged.loc[Ahmedabad_merged['Cluster Labels'] == 2, Ahmedabad_merged.columns[[1] + list(range(3, Ahmedabad_merged.shape[1]))]]
```

## 5. Conclusion

Best Neighborhoods for **CAFE** : Navrangpura , Bodakdev , Satellite , Thaltej , Vastrapur

Best **Cafe Restaurant** :Capstone , Shree Balaji Agrora Mall , Aarya Grand Hotel , Hyatt

Best **Restaurants** in Ahmedabad : Hyatt(Vastrapur) , The ferm(Sola) , Hyatt(Ashram Road)

**Cluster 1:** It is most recommended for *Cafe*.

**Cluster 2:** It is most recommended for *Indian Restaurants*.

**Cluster 3:** It is most recommended for *Fast food*.