Detecting the patterns in New Delhi-Zomato Restaurants:

P.V.Seetharam

1.Introduction:

Restaurants in New Delhi which offer variety of cuisine types are attracting various Indian and foreign guests across the globe.The quality of the food, affordability and proximity are of most important to any city dweller or for a new visitor. It will be helpful to have a general idea or bird's eye view of various restaurants in New Delhi.For this clustering is performed in order to gain insights.

1.2 Problem

The variety of restaurants in India in general, New Delhi in particular is humongonous .Number of cuisine types like North Indian, South Indian, Chinese.. etc often perplexes new guests, so it will be useful to have a cohesive framework to accommodate all cuisine types, locations and ratings which in turn make any guest to make appropriate choice to select a restaurant based on above factors.

2.Data acquisition and cleaning:

2.1Data sources:

To solve the problem, following data was used:

First using Foursquare API, list of restaurants in New Delhi are obtained . This data gives restaurant's Name, Address, Longitude, Latitude, Cuisine. This data is supplemented by Zomato data set which is published in Kaggle Competitions. This data set has extensive coverage of Restaurants in New Delhi with many details.

2.2 Data cleaning:

The data downloaded from Kaggle data set contained a lot of null values and longitude and latitude column had few zero values. These rows were deleted and appropriate changes were made into dataset.

Some rating values were not available so, these values were replaced with average ratings of remaining rows

There were more than 50 locations in original Kaggle data set, to make it manageable, only 18 large locations were chosen. The criteria for selecting locality was it should have atleast 65 restaurants registered with Zomato. Appropriate changes were made to Foursquare dataset to make it compatible with Kaggle-zomato data set.

All duplicate values are deleted

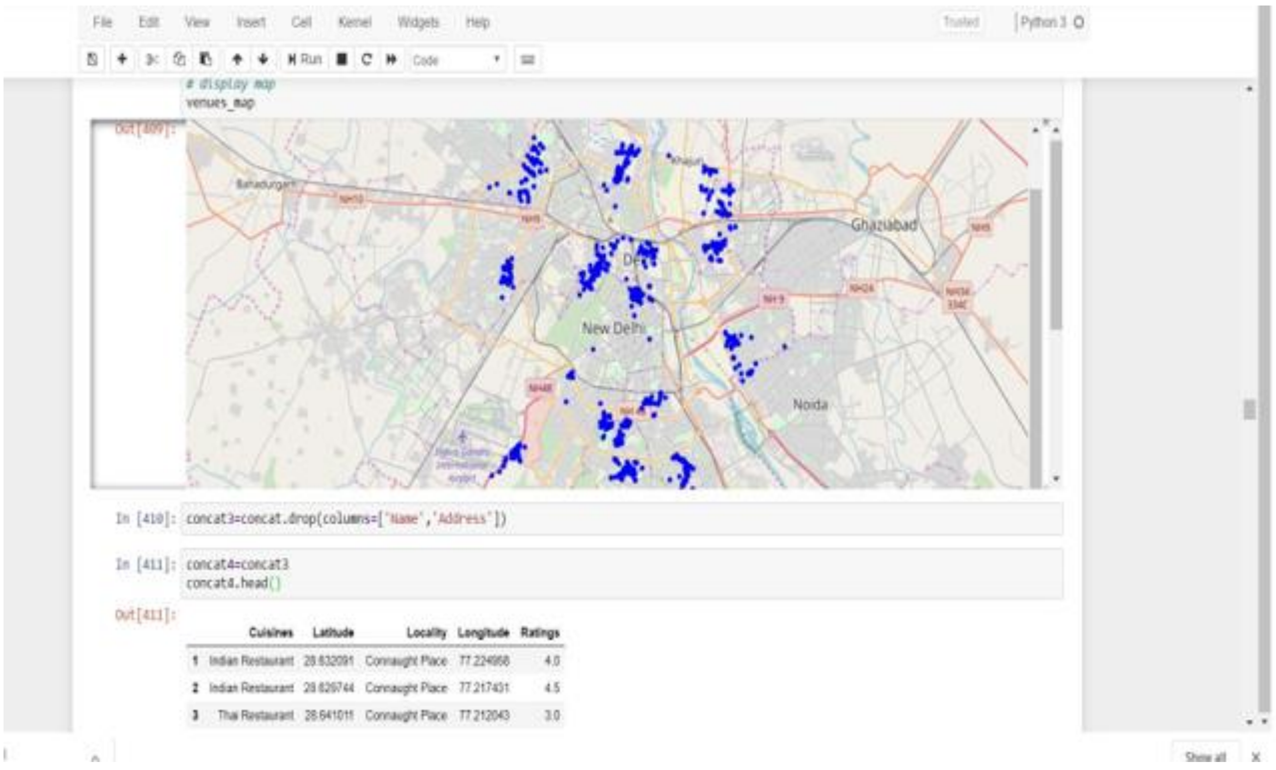
2.3 Feature selection:

Kaggle zomato dataset offers many features such as Locality Verbose, Longitude, Latitude, Cuisines, Average Cost for two, Currency, Has Table booking, Has Online delivery, Is delivering now, Switch to order menu, Price range, Aggregate rating, Rating color, Votes. But for this particular purpose,ie for cluster formation relevant features are Cuisines, Locality, Latitude,Longitude,Ratings.

3. Exploratory Data Analysis

The features selected for the unsupervised learning are locality,cuisine type(which are non numeric and hence not correlated),longitude,latitude,ratings.There is no multi-collinearity among features which is an obvious fact. Since no target variable is needed(we are not predicting any value,we need to have a grouping or clustering of restaurants),Unsupervised learning is performed.Normalisation is not required for numeric features.

Visualising the spread of restauarants in prominent locations in New Delhi:



```
# display map
venues_map
```

Out[409]:

```
In [410]: concat3=concat.drop(columns=['Name','Address'])
```

```
In [411]: concat4=concat3
          concat4.head()
```

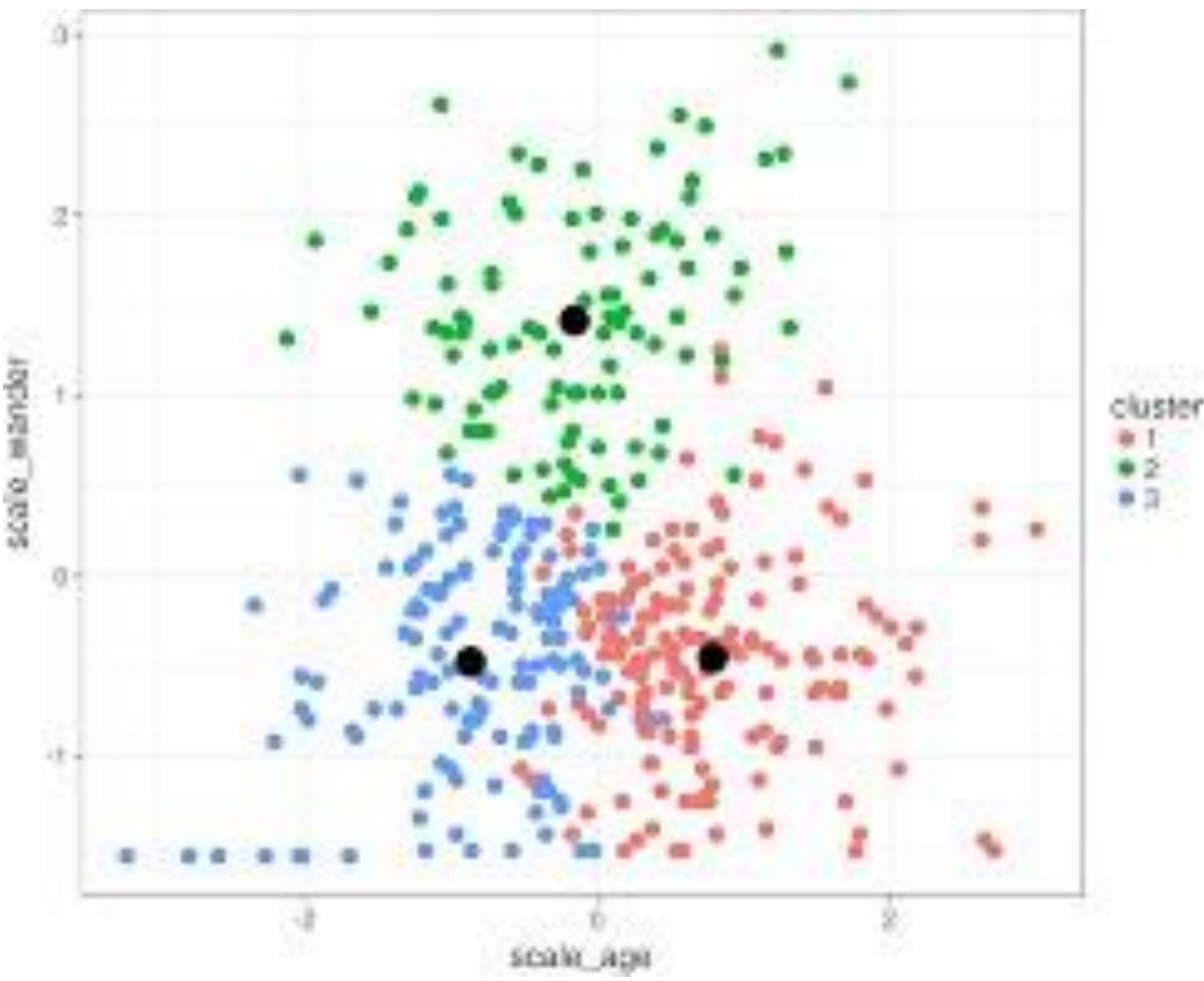
Out[411]:

	Cuisines	Latitude	Locality	Longitude	Ratings
1	Indian Restaurant	28.632091	Connaught Place	77.224958	4.0
2	Indian Restaurant	28.626744	Connaught Place	77.217431	4.5
3	Thai Restaurant	28.641011	Connaught Place	77.212043	3.0

4. Modeling:

This is an un supervised learning case and the model which would be deployed to get insight about restaurants in New Delhi is K means clustering 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 inter-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.



Courtesy:Google images

As shown in above figure the algorithm segments entire data into clusters of required numbers.

At first it chooses random centroid and it iteratively choose the best centroid and segment data into clusters.

4.1 Iterations of clustering:

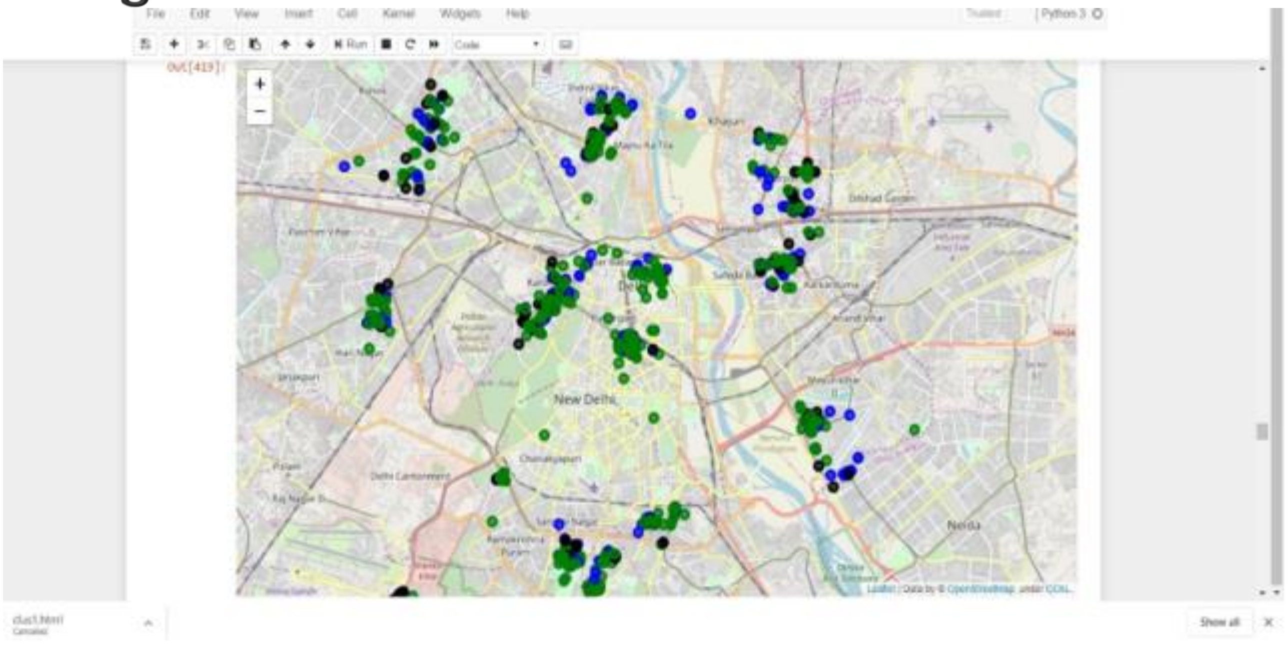
In order to give an optimum and stable solution, best centroid for each cluster will be selected after many iterations.

Centroid iterations as shown in jupyter notebook

```
Centroids: [[ 1.78571429e-02  2.67857143e-02  6.47321429e-02  1.1160714
3e-03
   1.11607143e-03  1.11607143e-02  3.23660714e-02  7.14285714e-
02
   1.42857143e-01  1.11607143e-03  5.02232143e-02  2.12053571e
-02
   2.23214286e-01  1.11607143e-03  4.35267857e-02  8.92857143e
-03
   2.23214286e-02  1.11607143e-03  1.11607143e-03  1.11607143e
-03
   4.46428571e-03  1.11607143e-03 -3.41393580e-15  1.97064587e
-15
   1.38392857e-01  1.11607143e-03  5.35714286e-02  5.58035714e
-02
   1.11607143e-03  4.46428571e-02  1.01562500e-01  6.25000000e
-02
   5.02232143e-02  5.80357143e-02  5.02232143e-02  5.24553571e
-02
   4.01785714e-02  2.56696429e-02  6.25000000e-02  4.79910714e
-02
   3.57142857e-02  5.91517857e-02  5.80357143e-02  6.80803571e
-02
   5.35714286e-02  7.25446429e-02  5.69196429e-02  2.86199866e
+01
   7.72076374e+01  3.37700893e+00]
 [-7.28583860e-17 -5.89805982e-17 -1.24900090e-16 -4.55364912e
-18
  -4.55364912e-18  2.86229374e-17 -6.24500451e-17 -2.91433544e
-16
  -5.82867088e-16 -4.55364912e-18 -2.77555756e-17  4.16333634e
-17
```

```
  -4.44089210e-16 -4.55364912e-18  1.52655666e-16 -3.64291930e
-17
   5.72458747e-17 -4.55364912e-18 -4.55364912e-18 -4.55364912e
-18
  -1.82145965e-17 -4.55364912e-18  1.00000000e+00 -7.77156117e
-16
  -4.44089210e-16 -4.55364912e-18 -1.17961196e-16 -1.11022302e
-16
  -4.55364912e-18  7.51633987e-02  1.07843137e-01  5.55555556e
-02
   2.61437908e-02  3.92156863e-02  5.55555556e-02  3.92156863e
-02
   7.51633987e-02  7.51633987e-02  2.61437908e-02  5.55555556e
-02
   8.49673203e-02  5.88235294e-02  4.24836601e-02  7.18954248e
-02
   3.59477124e-02  9.80392157e-03  6.53594771e-02  2.86282860e
+01
   7.72114906e+01  3.27352941e+00]
 [-1.56125113e-17 -5.20417043e-17  6.93889390e-17 -9.75781955e
-19
  -9.75781955e-19  2.34187669e-17  3.46944695e-17 -6.24500451e
-17
  -1.24900090e-16 -9.75781955e-19 -2.77555756e-17  5.20417043e
-18
  -2.22044605e-16 -9.75781955e-19  1.28369537e-16 -7.80625564e
-18
   4.68375339e-17 -9.75781955e-19 -9.75781955e-19 -9.75781955e
-19
  -3.90312782e-18 -9.75781955e-19  0.00000000e+00  1.00000000e
+00
   1.38777878e-17 -9.75781955e-19 -1.04083409e-16 -5.55111512e
-17
  -9.75781955e-19  9.52380952e-03  7.14285714e-02  5.71428571e
-02
   5.71428571e-02  3.33333333e-02  4.28571429e-02  7.61904762e
-02
   3.80952381e-02  9.52380952e-02  9.52380952e-02  6.19047619e
-02
   3.33333333e-02  5.71428571e-02  6.66666667e-02  7.14285714e
-02
   5.23809524e-02  3.33333333e-02  4.76190476e-02  2.86147514e
+01
   7.72018240e+01  3.18380952e+00]]
```

Images of 3 clusters formed in our data:



Cluster1: green color points

Cluster 2: blue color points

Cluster 3: black color points

5.Conclusion:

K-means clustering algorithm in this particular case is giving more weightage to cuisine features as compared to locality and ratings features while forming clusters.

As shown above cluster 2 only picked up restaurants which offer North Indian and cluster 3 only picked up

Restaurants which offer North Indian,Chinese combined. The clusters formed above are not location specific nor ratings specific.