

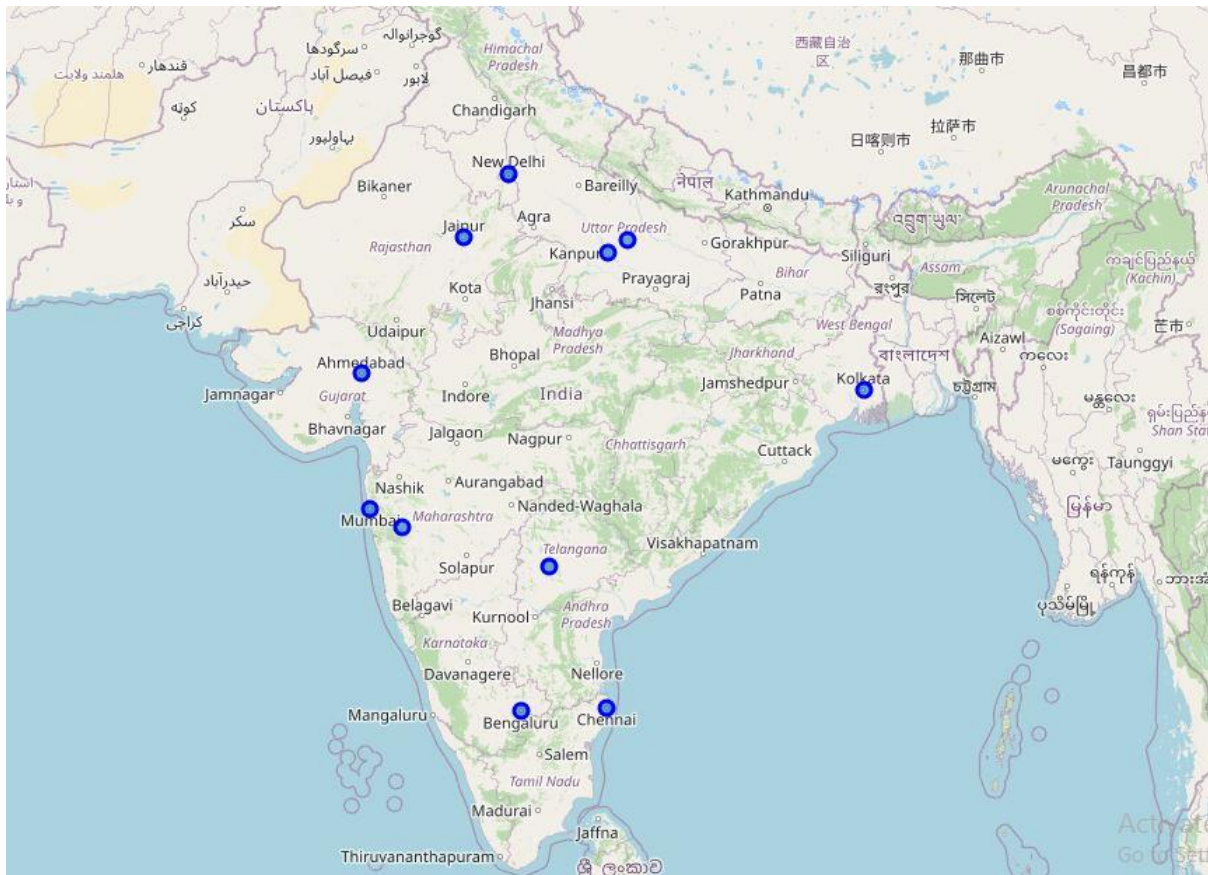
MEASURE & ANALYZE

SELECTING THE MOST POPULOUS CITIES IN INDIA

Here it is assumed to take the following 11 cities

CITIES	APPROX. POPULATION (IN MILLIONS)
MUMBAI	18
KOLKATA	15
NEW DELHI	22
AHMEDABAD	5.6
BENGALURU	8.4
CHENNAI	7.1
HYDERABAD	6.8
PUNE	3.1
JAIPUR	3.1
KANPUR	2.9
LUCKNOW	2.8

MARKING THEM ON THE MAP OF INDIA USING FOLIUM IN PYTHON



NOW APPLYING K-MEANS

K-MEANS

The *k*-means algorithm searches for a pre-determined number of clusters within an unlabelled multidimensional dataset. It accomplishes this using a simple conception of what the optimal clustering looks like:

- The "cluster centre" is the arithmetic mean of all the points belonging to the cluster.
- Each point is closer to its own cluster centre than to other cluster centres.

Here we will use the latitudes and longitudes of the cities as two dimensional datasets.

Depending on the number of clusters we set we'll get the clusters' centres from which we can shortlist the areas for our warehouses. Here's the code for k-means in python after creating data frames containing location and cities.

```
In [11]: k_means = KMeans(init = "k-means++", n_clusters = 4, n_init = 12)
         k_means.fit(df_loc)

Out[11]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
               n_clusters=4, n_init=12, n_jobs=None, precompute_distances='auto',
               random_state=None, tol=0.0001, verbose=0)

In [12]: k_means_cluster_centers = k_means.cluster_centers_
         k_means_cluster_centers

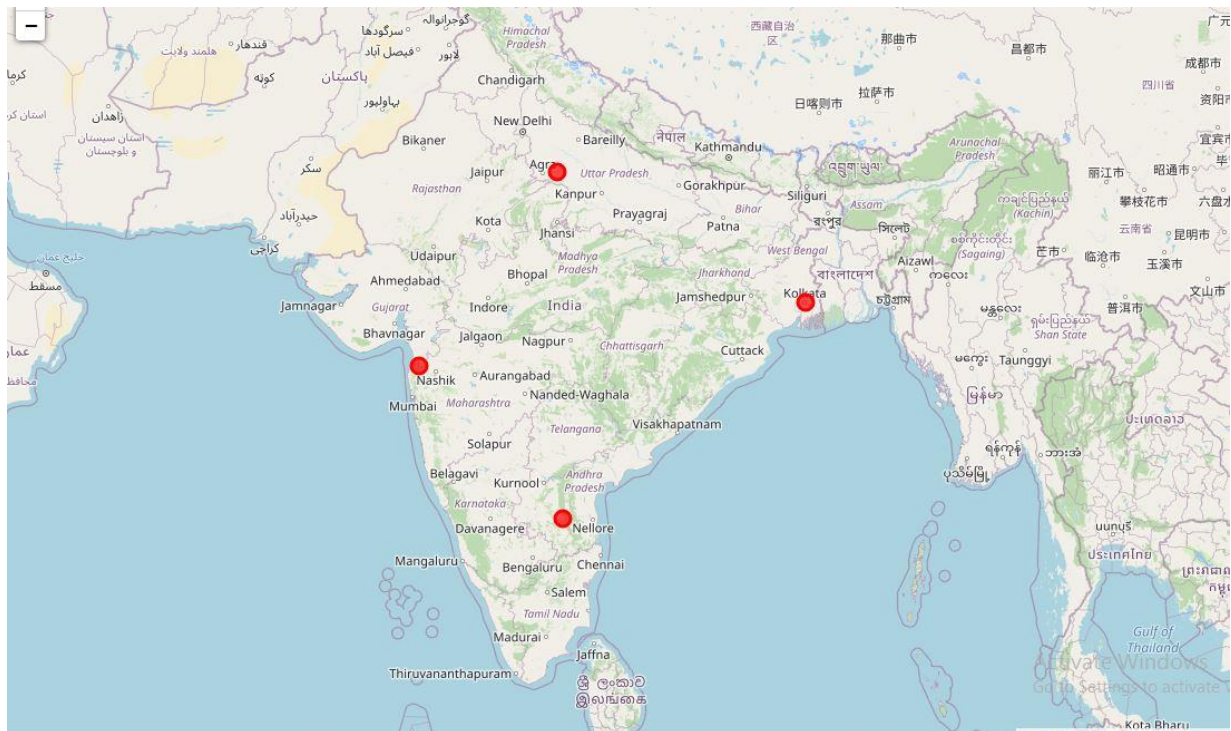
Out[12]: array([[27.21673133, 78.57466168],
               [14.47363327, 78.7832538 ],
               [22.54541245, 88.35677516],
               [20.20634723, 73.10385123]])

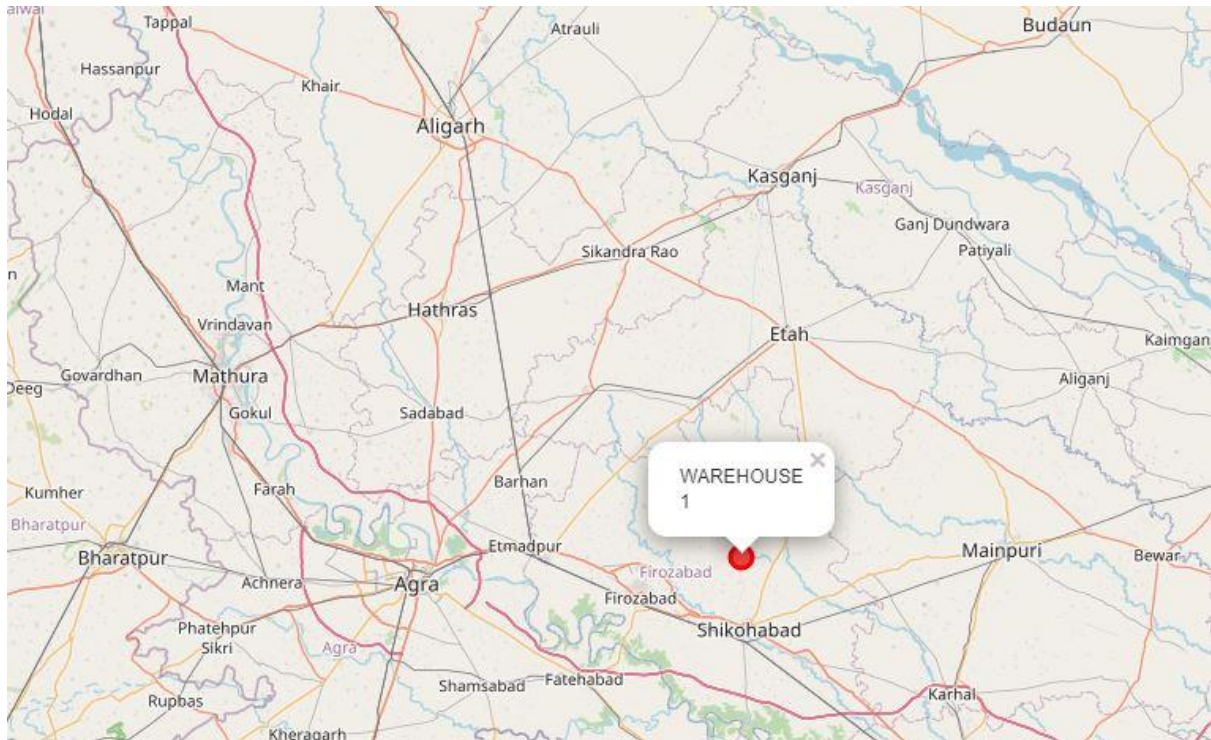
In [13]: numpy_data = k_means_cluster_centers
         df2 = pd.DataFrame(data=numpy_data, columns=["latitude", "longitude"])
         #df2
```

CASE 1: K=4 (4 CLUSTERS AND THEIR CENTRES)

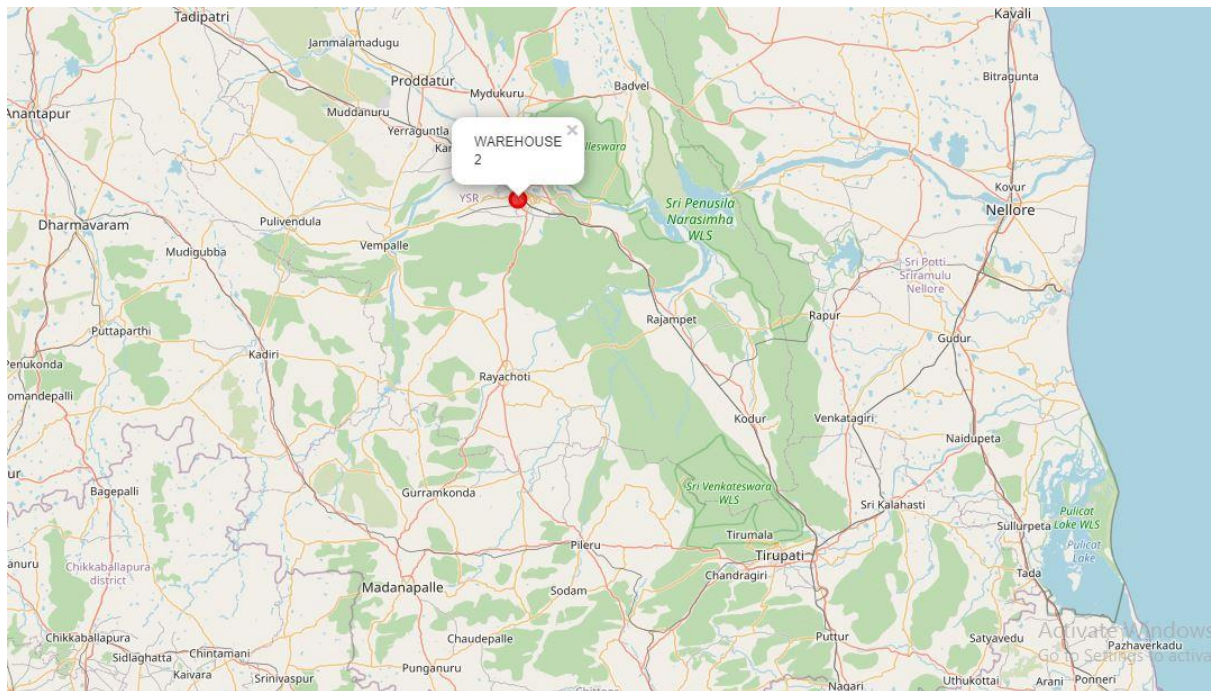
THE AREAS WE GOT HAD THE CENTRES IN FORM OF LATITUDES AND LONGITUDES WERE:

	Warehouses	latitude	longitude
0	WAREHOUSE 1	27.216731	78.574662
1	WAREHOUSE 2	14.473633	78.783254
2	WAREHOUSE 3	22.545412	88.356775
3	WAREHOUSE 4	20.206347	73.103851



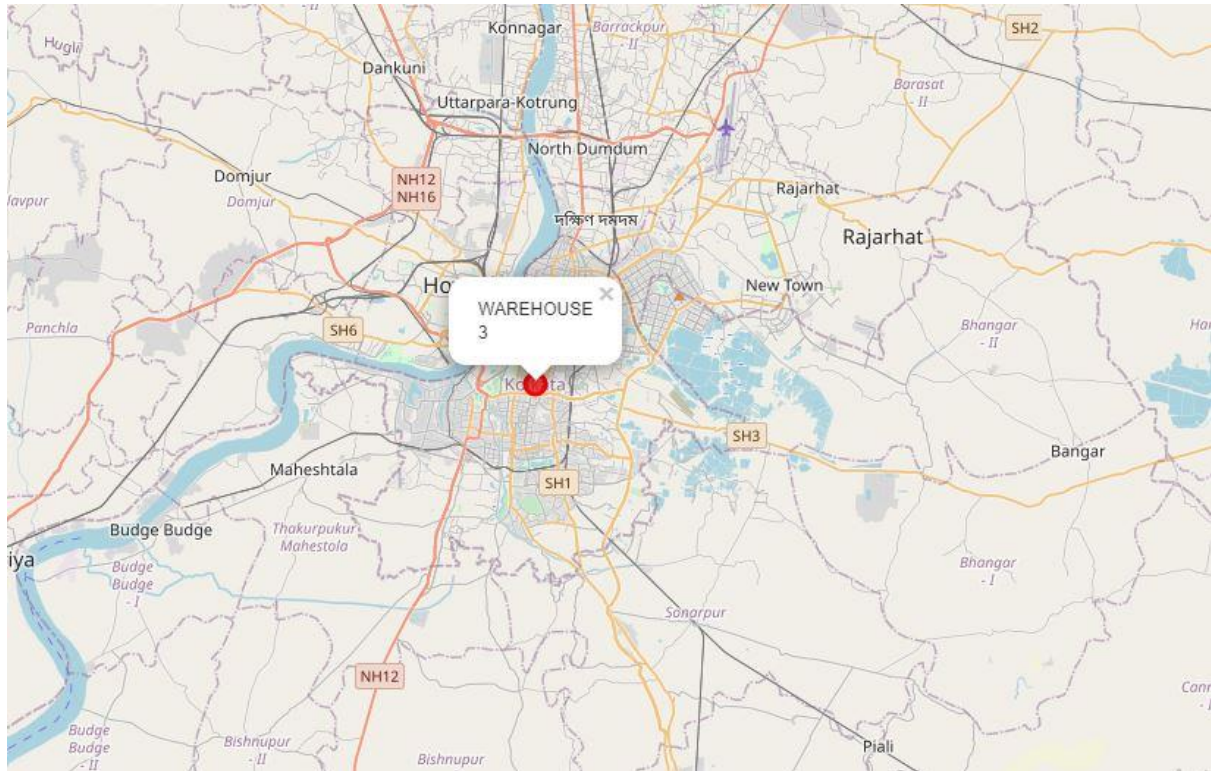


WAREHOUSE 1: was near Agra, Aligarh

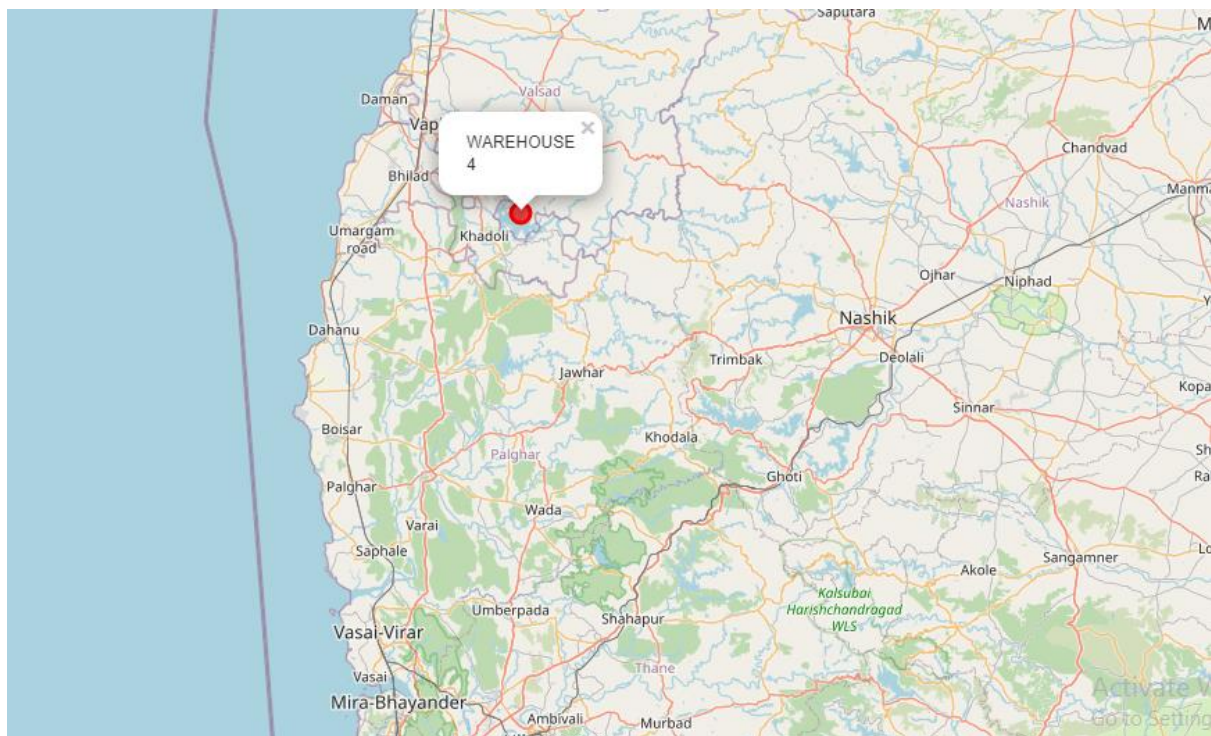


WAREHOUSE 2: was near places like Nellore, Tirupati, Anantapur & Poddatur.

WAREHOUSE 3: in Kolkata



WAREHOUSE 4: near Nashik, Vapi, Vasai-Virar (Mumbai)



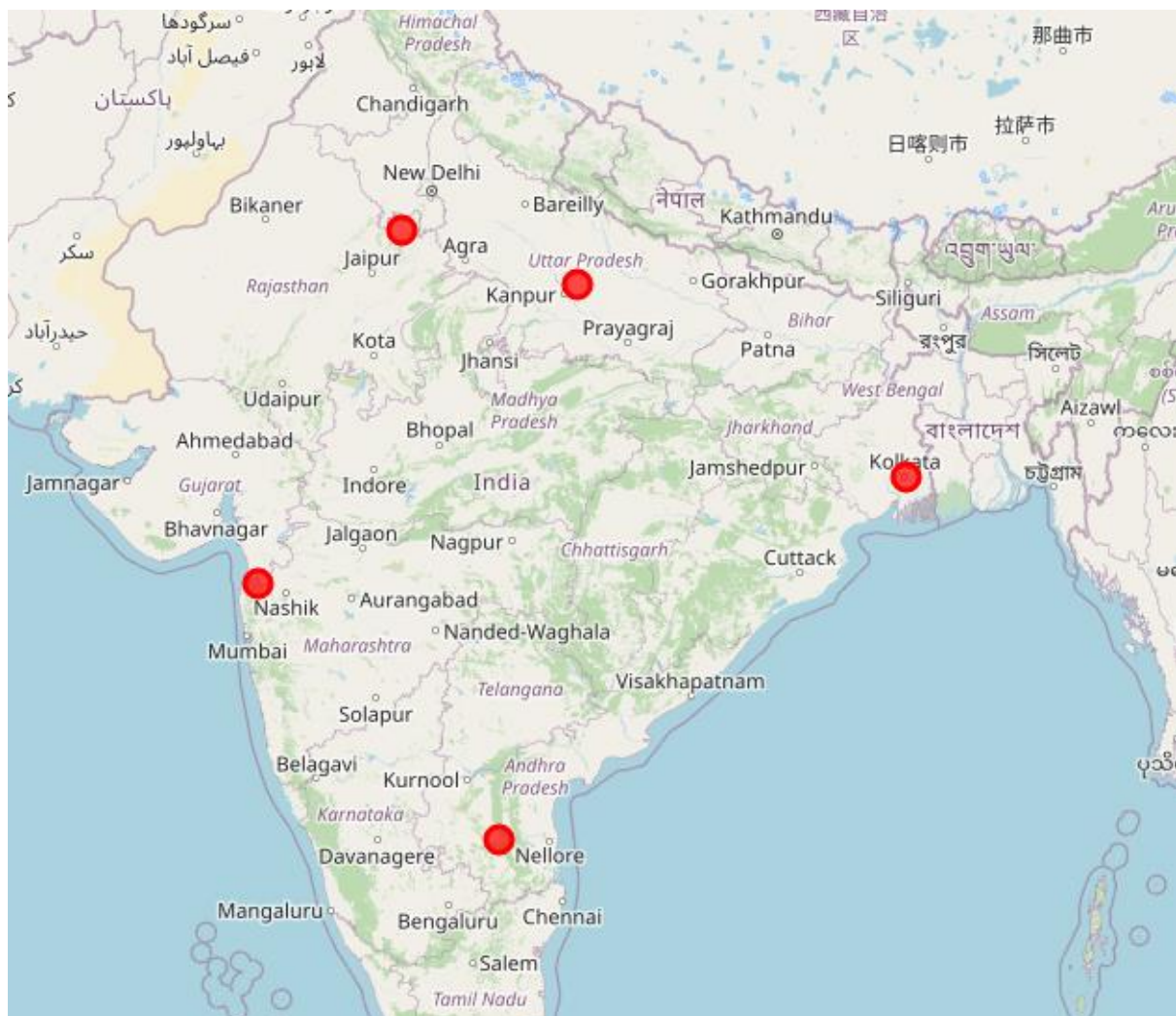
CASE 2: K=5 (5 CLUSTERS AND THEIR CENTRES)

	Warehouses	latitude	longitude
0	WAREHOUSE 1	14.473633	78.783254
1	WAREHOUSE 2	27.783956	76.521144
2	WAREHOUSE 3	22.545412	88.356775
3	WAREHOUSE 4	20.206347	73.103851
4	WAREHOUSE 5	26.649507	80.628179

WAREHOUSE 4,1, & 3 ARE SAME ASIN CASE 1'S 4,2 & 3 respectively.

Warehouse 1 is nearer to Jaipur and 5 is between Lucknow & Kanpur.

Not so feasible.



CASE 3: K=6 (6 CLUSTERS AND THEIR CENTRES)

Warehouses		latitude	longitude
0	WAREHOUSE 1	13.029646	78.937566
1	WAREHOUSE 2	26.649507	80.628179
2	WAREHOUSE 3	20.206347	73.103851
3	WAREHOUSE 4	22.545412	88.356775
4	WAREHOUSE 5	27.783956	76.521144
5	WAREHOUSE 6	17.361608	78.474629

The recent change is

Warehouse 1 is near Vellore & Warehouse 6 is in Hyderabad.

TOO MANY WAREHOUSES. WOULD BE COSTLIER.

