IME672A: DATA MINING AND KNOWLEDGE DISCOVERY

Warehouse Location Problem

Submitted by: Ayushi Singh, 200258

An online retailer has collected data on the location (latitude and longitude) of its customers and their average monthly purchase (in kgs) for four product categories - Apparel, Books, Electronics and Grocery.

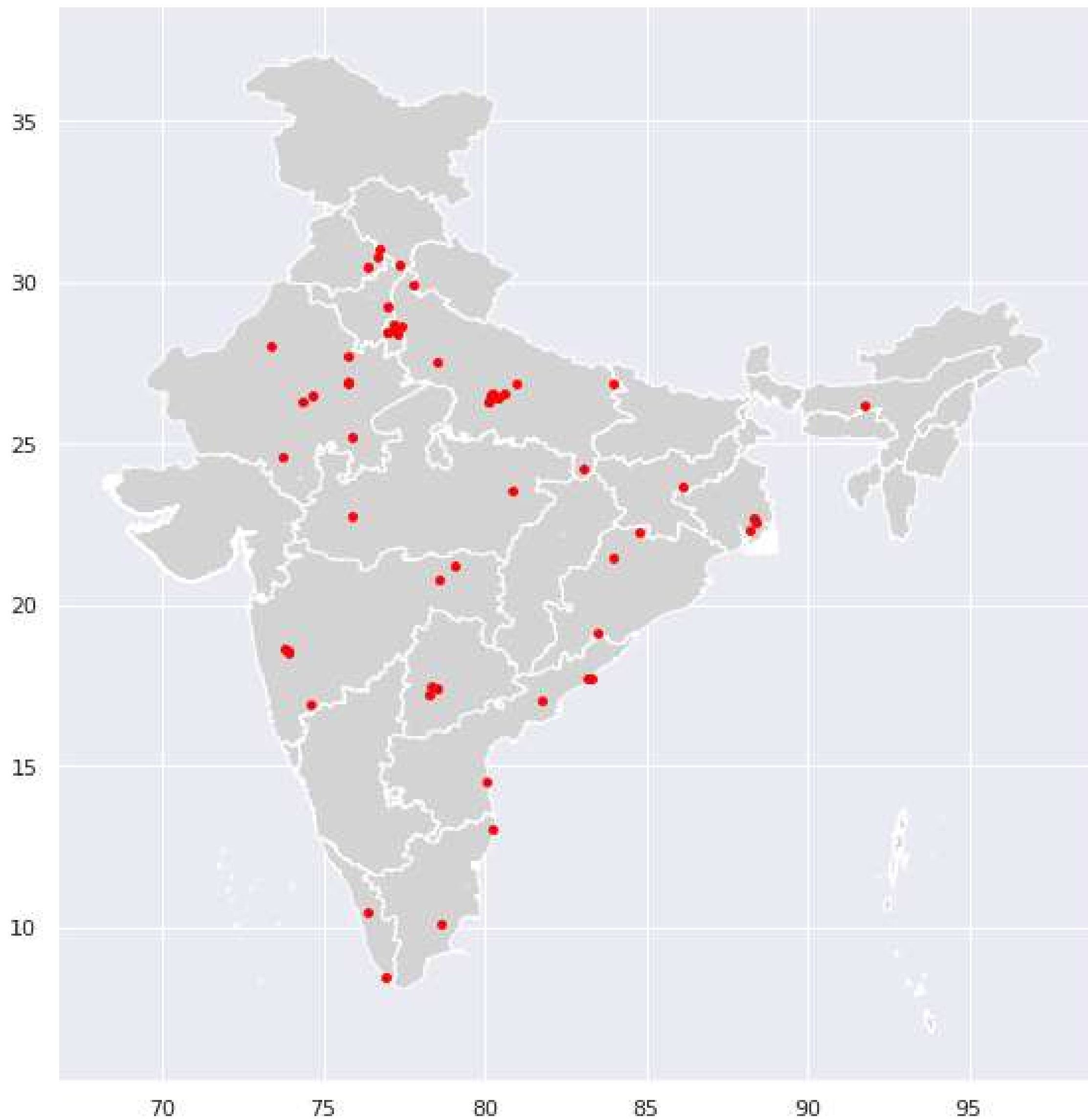
The provided dataset had 6 numerical columns with 0 null values. All data points lie in the Indian territory i.e., between latitudes 8°4'N and 37°6'N and longitudes 68°7'E and 97°25'E.

The following statistical measures were calculated for the dataset. From the below table the range of all four products are similar and hence does not require separate pre-processing. We can proceed further.

	Latitude	Longitude	Apparel	Books	Electronics	Grocery
count	79.000000	79.000000	79.000000	79.000000	79.000000	79.000000
mean	24.229996	79.481421	5.129620	7.467215	3.911646	10.206835
std	4.952235	3.545489	4.944455	6.412815	4.778758	5.929100
min	8.469167	73.353185	0.160000	0.110000	0.110000	0.140000
25%	22.276201	77.014239	0.935000	0.770000	0.670000	6.145000
50%	26.504384	80.196540	3.220000	7.510000	2.100000	9.230000
75%	26.529297	80.234894	8.310000	10.740000	5.255000	15.655000
max	31.022162	91.778177	18.410000	19.560000	18.260000	19.610000

Plotting the data points on a map of India gives a better visualization of the distribution.

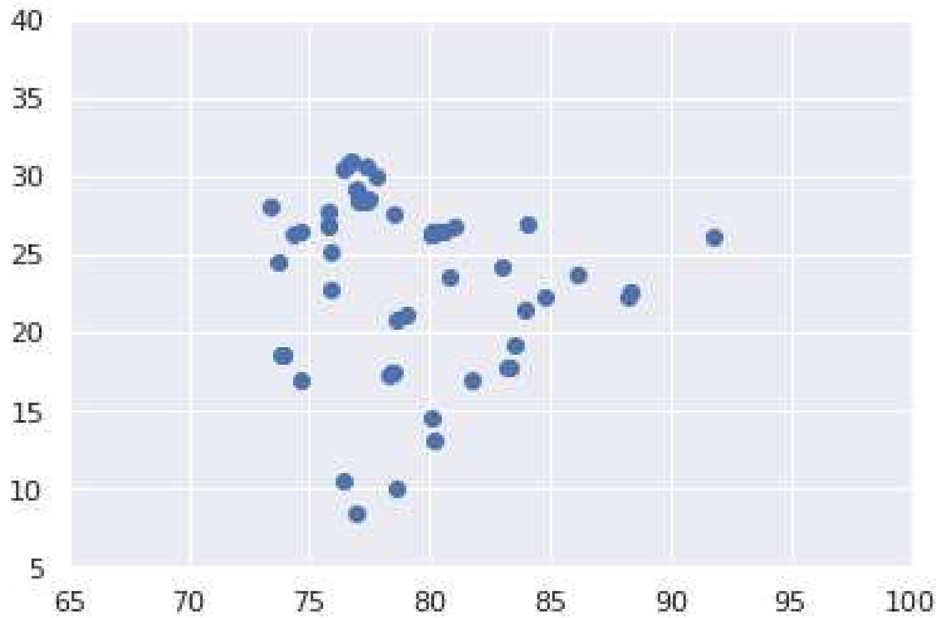
```
from shapely.geometry import Point
import geopandas as gpd
from geopandas import GeoDataFrame

geometry = [Point(xy) for xy in zip(dataset['Longitude'], dataset['Latitude'])]
gdf = GeoDataFrame(dataset, geometry=geometry)
#this is a simple map that goes with geopandas
# world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
world = gpd.read_file('/content/Indian_states.shp')
gdf.plot(ax=world.plot(figsize=(20, 10),color='lightgrey'), marker='o', color='red', markersize=15);
# ax.set_title('Distribution of Customers')
```



From this it can be observed that:

- Very few customers are from the North East and Southern regions of India.
- High concentration of customers near Delhi-NCR & Kanpur region.

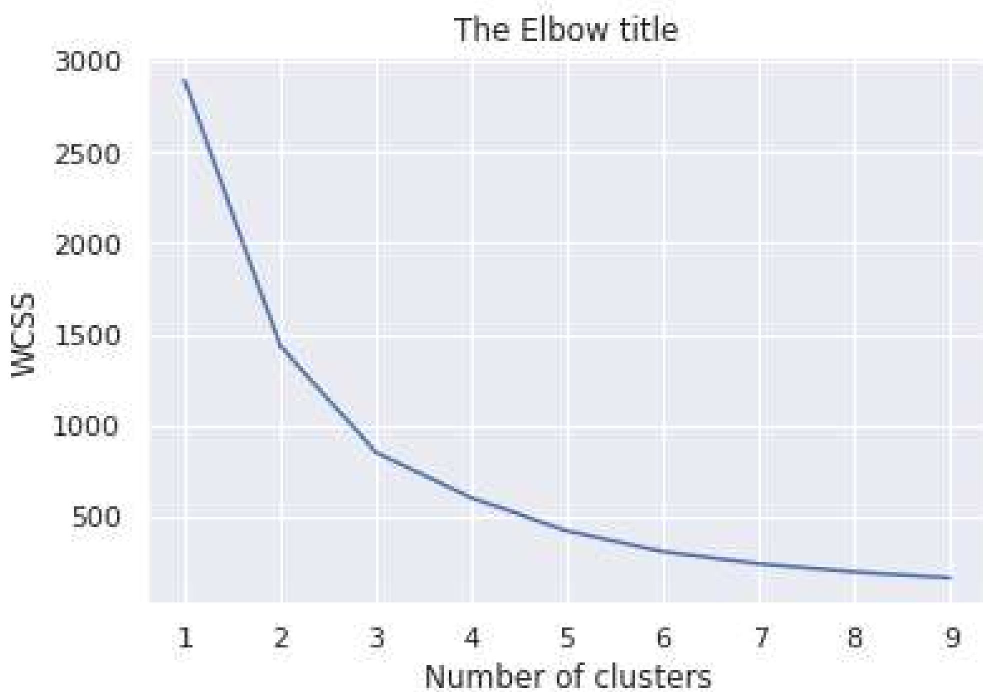
Plot of ‘Longitude’ vs ‘Latitude’



Ans 1) For determining the positions and number of warehouses given that each warehouse has sufficient capacity to serve all the customers for all the product types, the problem reduces to cluster customers based on region so that the distance between warehouse and customers reduces.

For this problem I chose k-means clustering which considers **Euclidean distance** between data points to form clusters. To determine the K value, the **elbow method** was used, for each K value **WCSS (Within-Cluster Sum of Square)** was calculated as a loss function.

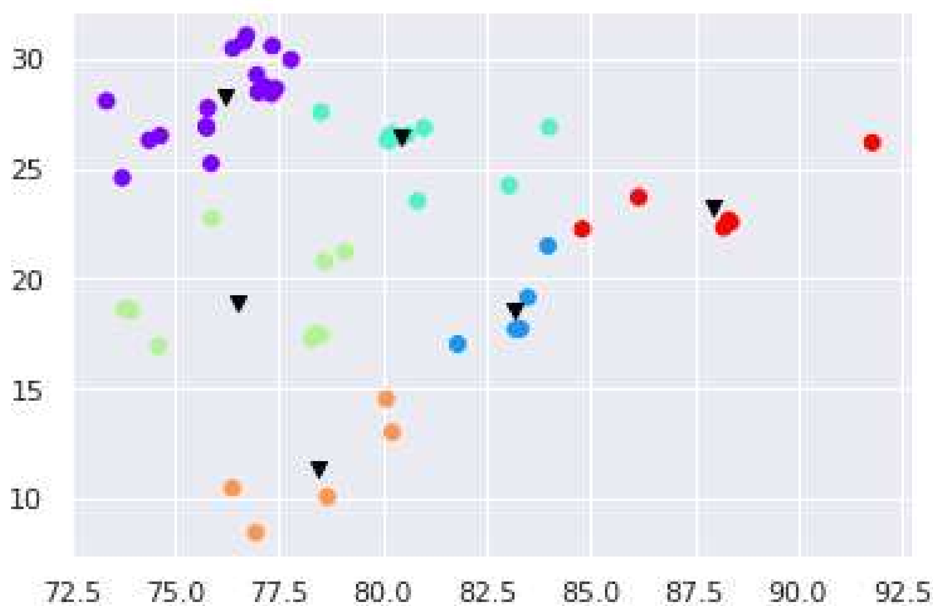
```
wcss=[]
for i in range(1,10):
  kmeans = KMeans(i)
  kmeans.fit(x)
  wcss_iter = kmeans.inertia_
  wcss.append(wcss_iter)

number_clusters = range(1,10)
plt.plot(number_clusters,wcss)
plt.title('The Elbow title')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
```



From this trend, the number of clusters can be taken as 6 since after that there is not much considerable reduction in WCSS. Further if we increase the number of warehouses, the cost of maintaining them is not compensated.

Hence number of warehouses needed=6

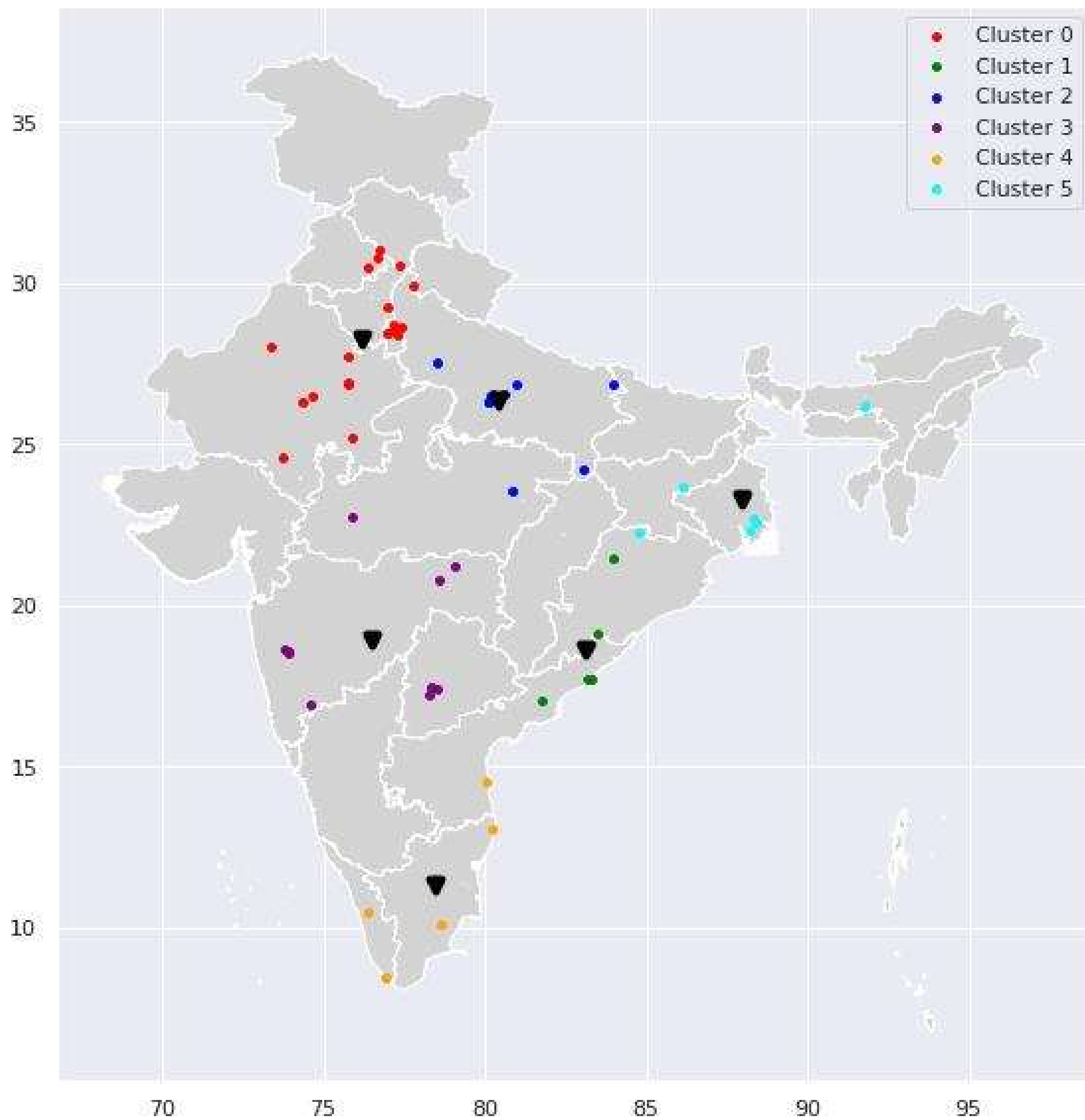
```
data_with_clusters = dataset.copy()
data_with_clusters['Clusters'] = identified_clusters
plt.scatter(data_with_clusters['Longitude'],data_with_clusters['Latitude'],c=data_with_clusters['Clusters'],cmap='rainbow')
plt.scatter(centroids[:,1],centroids[:,0], color='black',marker='v')
```



Plotting the same on the map of India we get:

(The warehouse locations are shown with black inverted triangle markers)

```
[67] geometry = [Point(xy) for xy in zip(dataset['Longitude'], dataset['Latitude'])]
     gdf = GeoDataFrame(data_with_clusters, geometry=geometry)
     #this is a simple map that goes with geopandas
     # world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
     world = gpd.read_file('/content/Indian_states.shp')
     fig,ax=plt.subplots(figsize=(20,10))
     world.plot(ax=ax,color='lightgrey')
     gdf[gdf['Clusters']==0].plot(ax=ax, marker='o', color='red', markersize=15,label='Cluster 0');
     gdf[gdf['Clusters']==1].plot(ax=ax, marker='o', color='green', markersize=15,label='Cluster 1');
     gdf[gdf['Clusters']==2].plot(ax=ax, marker='o', color='blue', markersize=15,label='Cluster 2');
     gdf[gdf['Clusters']==3].plot(ax=ax, marker='o', color='purple', markersize=15,label='Cluster 3');
     gdf[gdf['Clusters']==4].plot(ax=ax, marker='o', color='orange', markersize=15,label='Cluster 4');
     gdf[gdf['Clusters']==5].plot(ax=ax, marker='o', color='cyan', markersize=15,label='Cluster 5');
     plt.scatter(centroids[:,1],centroids[:,0],marker='v',color='black',linewidths=4)
     ax.legend()
     # ax.set_title('Distribution of Customers')
```



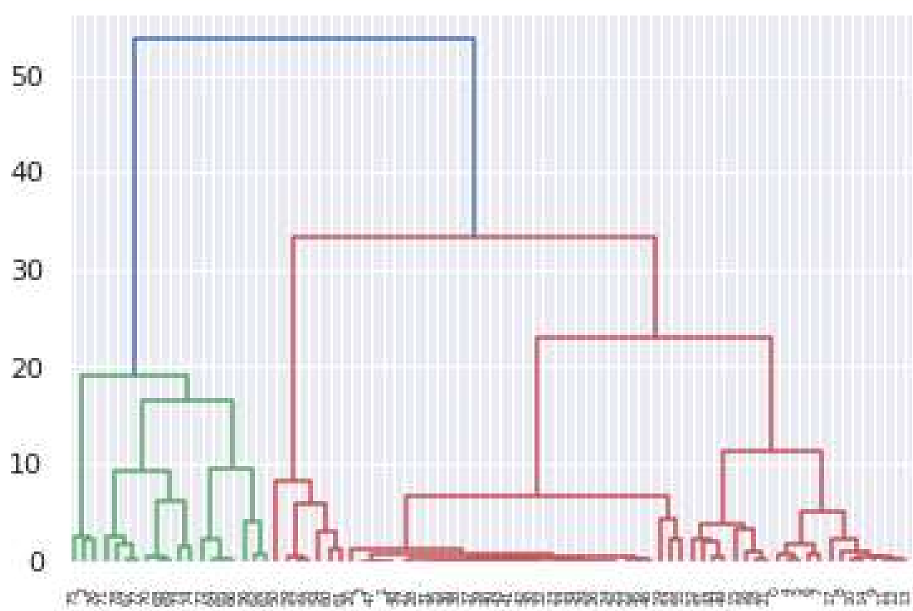
Ans 2) For this question we are assuming that each warehouse can store a maximum 100 kgs of any product type. From the previous part, the clusters we got had the following distribution:

Clusters	Latitude	Longitude	Apparel	Books	Electronics	Grocery
0	565.214693	1524.253425	93.66	128.19	87.69	223.01
1	93.034155	415.779002	19.11	46.91	40.49	38.01
2	870.284850	2654.026849	191.17	250.74	93.49	333.59
3	189.445456	765.002200	57.77	50.89	53.09	107.62
4	56.579036	392.312754	11.95	42.61	13.75	61.09
5	139.611525	527.658058	31.58	70.57	20.51	43.02

Here in cluster 0 & cluster 2 due to high density of customers, the required condition of 100 kg is not fulfilled. Therefore, we must form new clusters. For this I chose an **Agglomerative clustering algorithm.**

```
[75] import pandas as pd
     import numpy as np
     from matplotlib import pyplot as plt
     from sklearn.cluster import AgglomerativeClustering
     import scipy.cluster.hierarchy as sch

[76] X = dataset.iloc[:, [1, 0]].values

dendrogram = sch.dendrogram(sch.linkage(X, method='ward'))
```

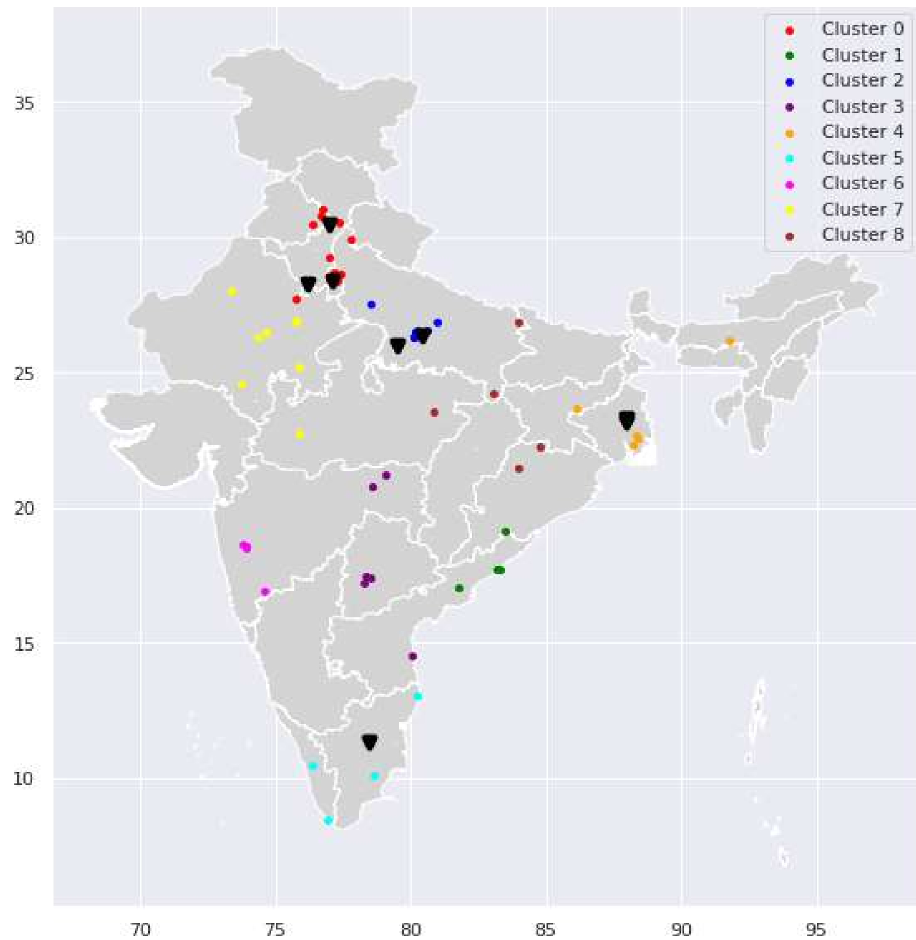
Dendrogram using Ward method

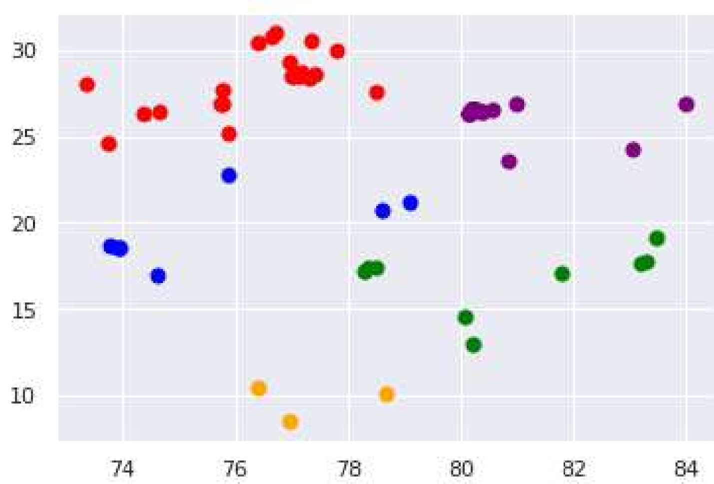


Clusters	Latitude	Longitude	Apparel	Books	Electronics	Grocery
0	380.863119	1000.771499	62.06	79.30	71.30	127.39
1	71.560633	331.801949	6.83	46.46	35.51	28.95
2	795.661915	2406.137495	175.17	235.03	66.17	317.98
3	108.585445	472.964907	17.80	49.28	35.78	66.01
4	117.370126	442.851062	15.22	68.38	19.80	35.79
5	42.051668	312.225216	7.79	24.47	13.48	51.86
6	72.653490	296.237887	34.25	19.60	14.92	31.88
7	207.085463	599.368870	41.48	49.04	19.05	114.58
8	118.337856	416.673403	44.64	18.35	33.01	31.90

In the below map the black inverted triangles depict the Warehouse locations:

```
[73] geometry = [Point(xy) for xy in zip(dataset['Longitude'], dataset['Latitude'])]
     gdf = GeoDataFrame(data_with_clusters, geometry=geometry)
     #this is a simple map that goes with geopandas
     # world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
     world = gpd.read_file('/content/Indian_states.shp')
     fig,ax=plt.subplots(figsize=(20,10))
     world.plot(ax=ax,color='lightgrey')
     gdf[gdf['Clusters']==0].plot(ax=ax, marker='o', color='red', markersize=15,label='Cluster 0');
     gdf[gdf['Clusters']==1].plot(ax=ax, marker='o', color='green', markersize=15,label='Cluster 1');
     gdf[gdf['Clusters']==2].plot(ax=ax, marker='o', color='blue', markersize=15,label='Cluster 2');
     gdf[gdf['Clusters']==3].plot(ax=ax, marker='o', color='purple', markersize=15,label='Cluster 3');
     gdf[gdf['Clusters']==4].plot(ax=ax, marker='o', color='orange', markersize=15,label='Cluster 4');
     gdf[gdf['Clusters']==5].plot(ax=ax, marker='o', color='cyan', markersize=15,label='Cluster 5');
     gdf[gdf['Clusters']==6].plot(ax=ax, marker='o', color='magenta', markersize=15,label='Cluster 6');
     gdf[gdf['Clusters']==7].plot(ax=ax, marker='o', color='yellow', markersize=15,label='Cluster 7');
     gdf[gdf['Clusters']==8].plot(ax=ax, marker='o', color='brown', markersize=15,label='Cluster 8');
     plt.scatter(centroids_demo[:,1],centroids_demo[:,0],marker='v',color='black',linewidths=4)
     ax.legend()
     # ax.set_title('Distribution of Customers')
```





Hence although we divided the customers into **9 clusters, the number of warehouses needed will be 11** since in two regions (Near Delhi & Kanpur), the density of customers is high hence more than one warehouse is needed to cater the supplies.