Advanced Machine Learning – Practicum 1 – K-means Algorithm

[Tee Li Lin]

Abstract

This assignment's objective is to identify the lightning's clusters based on the location, according to the given dataset which contain the information where the lightning discharge events took place all over the world.

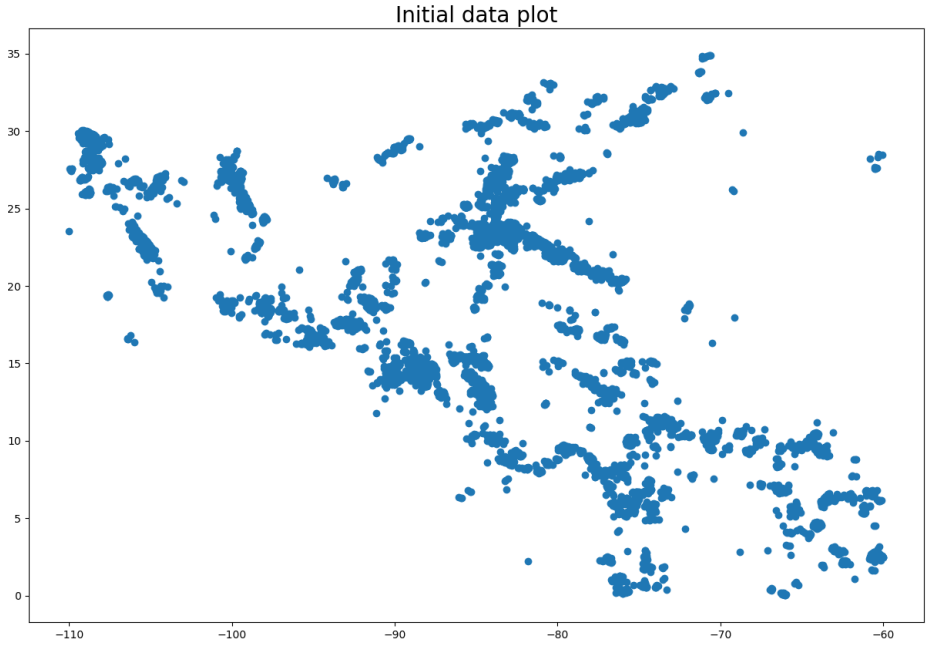
1. Introduction

1.1 Introduce the Problem

As the dataset only contains individual lightning discharge event information, it is very hard to identity where are the clusters located. Thus, by using K-means clustering method, this will help to solve this issue.

1.2 Dataset

Below is the initial data plot as per visualization, there is no information about the locations of the clusters took place.



2. Objective function

The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid. Below is the objective function for k-means where wik=1 for data point xi if it belongs to cluster k; otherwise, wik=0. Also, μk is the centroid of xi's cluster.

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \left\| x^i - \mu_k \right\|^2$$

3. Object assignment

Below are the initial steps for K-means:

1. Assume K Centroids (for K Clusters, for this assignment, K=5)
2. Compute Euclidean distance of each object with these Centroids.
3. Assign the objects to clusters with shortest distance

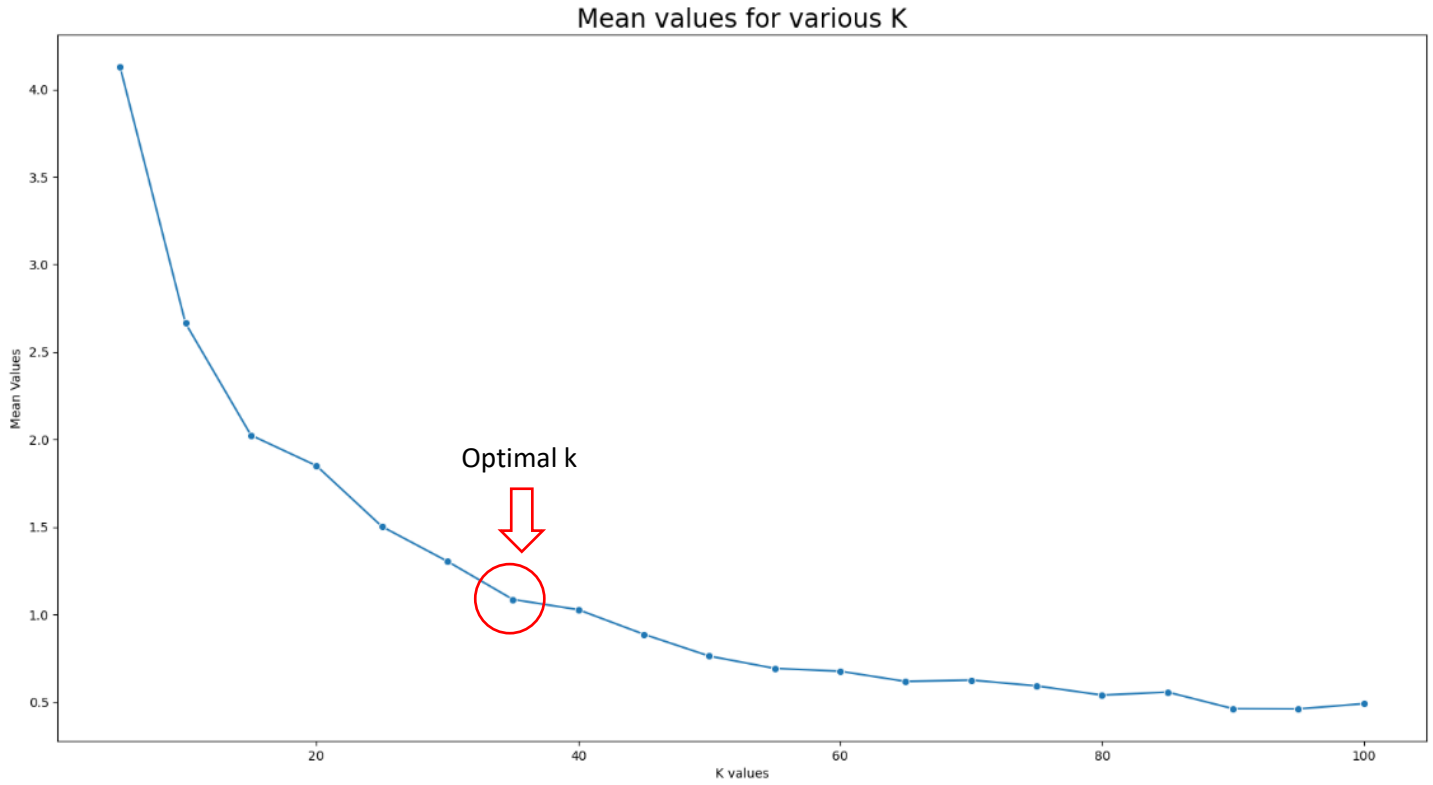
4. Updating means

Compute the new centroid (mean) of each cluster based on the objects assigned to each cluster. The K number of means obtained will become the new centroids for each cluster

From computing the Euclidean distance of each object with these centroids, assigning the objects to clusters with shortest distance and then compute the new centroid (mean) of each cluster based on the objects assigned to each cluster. These steps will continue till there is convergence, i.e., there is no movement of objects from one cluster to another (meaning there is no further changes in errCompute value or means)

5. Choosing *k*

Optimal k can be achieved by using the elbow method. Below is the plot mean values (from final_cal function) vs various values of k (ranging from 1 to 100). The sharp point of bend or a point (looking like an elbow joint) of the plot like an arm, will be considered as the best/optimal value of K. For this case, the optimal value for this case is k=35.



Appendix – figure

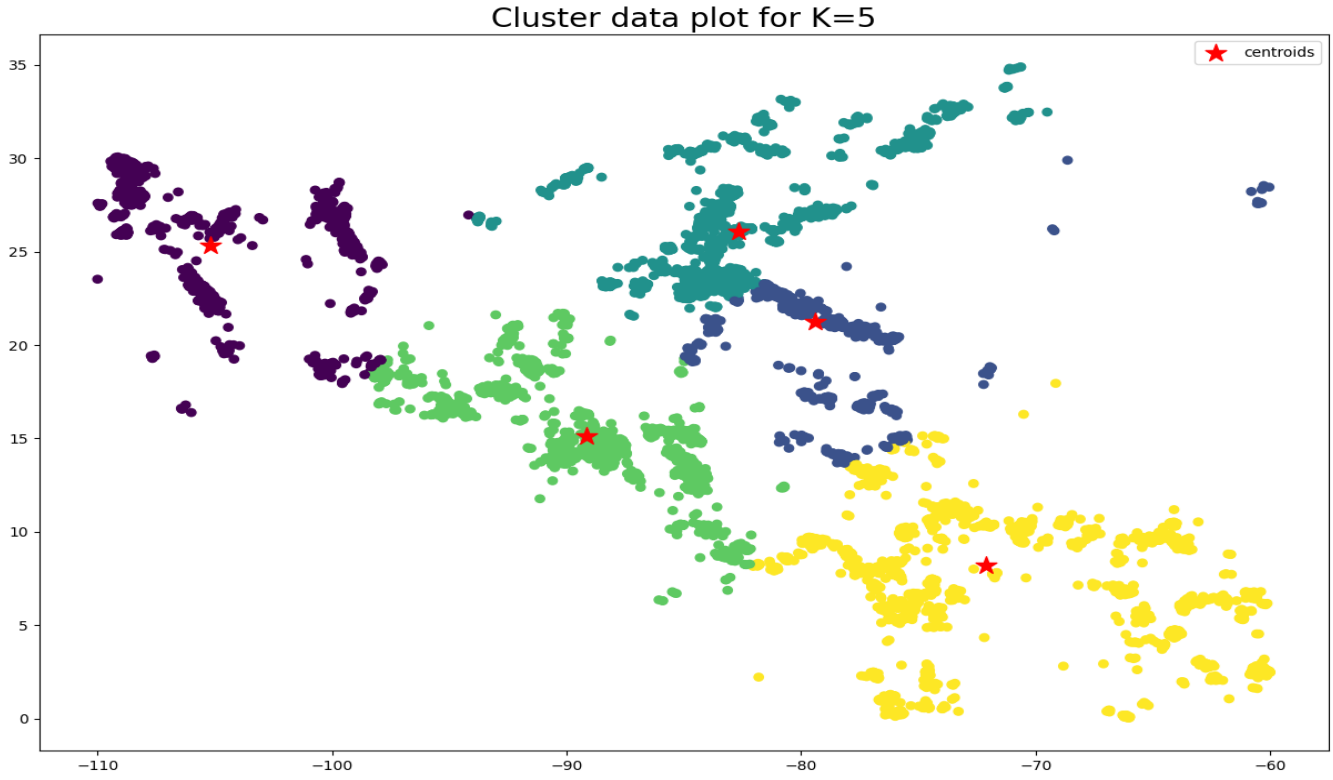


Figure 1. Cluster data plot (K=5)

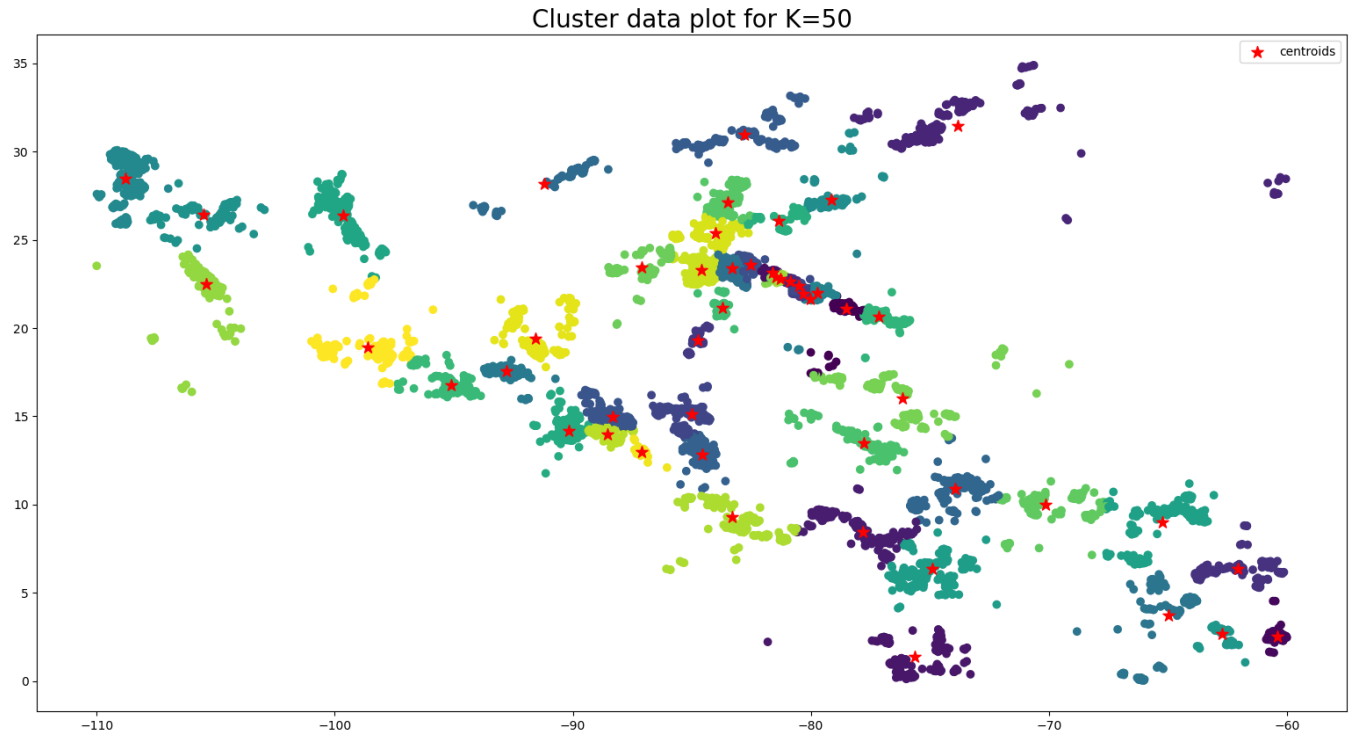


Figure 2. Cluster data plot (K=50)

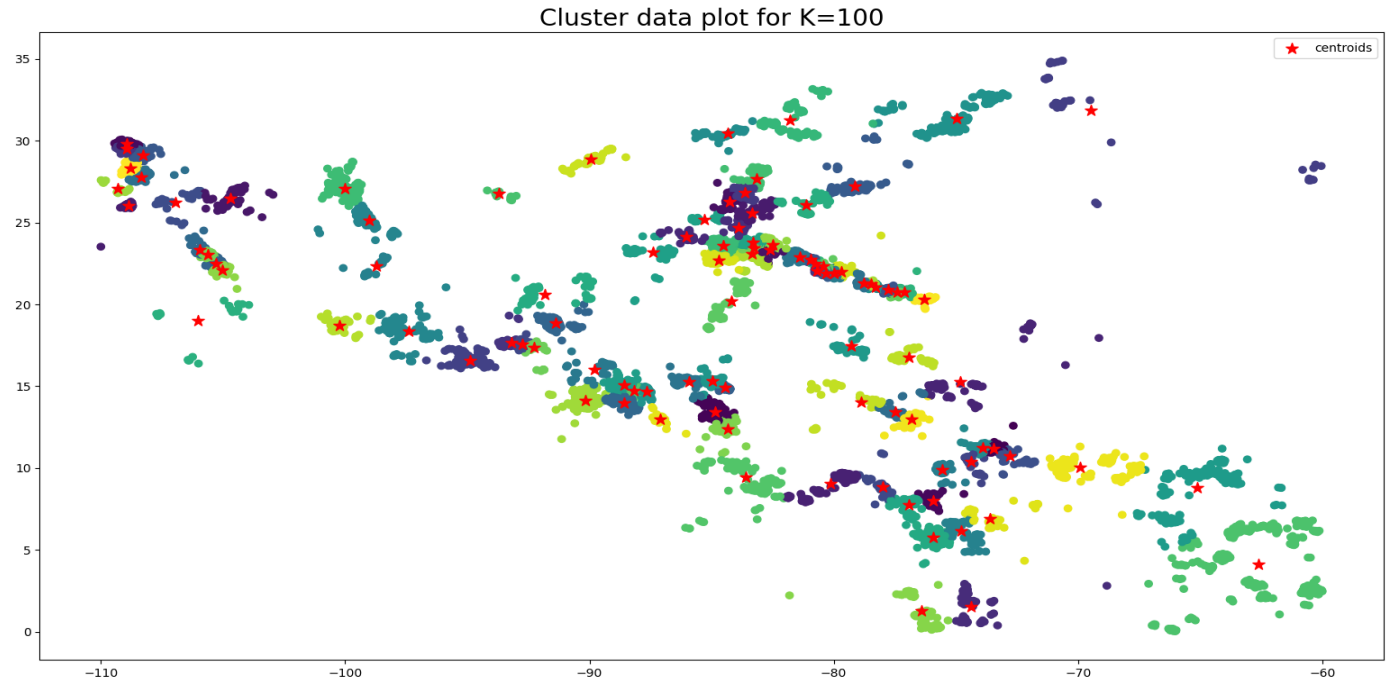


Figure 3. Cluster data plot (K=100)

Points to take note:

For K=50, K=100 as well as finding the optimal k, I used the random function to select my initial means. Below is the random function code:

```
def random_M(data,k):
    idx = np.random.choice(len(data), k, replace=False)
    centroids = data[idx, :2]
    return centroids
```

Appendix – table

Table 1. Result Table (K=5)

```
Results for K=5 using top 5 objects from X
------------------------------------------------------------

Total number of iterations for K=5: 16

Final errCompute for K=5: 4.01

Final M for K=5:
[[-105.19604652   25.33132344]
 [ -79.39863196   21.23281229]
 [ -82.67640192   26.06156976]
 [ -89.16882233   15.14357792]
 [ -72.13285047    8.21465361]]

Final Group for K=5:
[[-84.5152  19.1264   1.    ]
 [-76.6465  20.2474   1.    ]
 [-76.6615  20.342    1.    ]
 ...
 [-80.6016  22.0641   1.    ]
 [-80.3845  21.9461   1.    ]
 [-75.4638  10.13     4.    ]]

Time taken for K=5:0.297s
```

Table 2. Result Table (K=50)

```
Results for K=50 using random 50 objects from X
------------------------------------------------------------

Total number of iterations for K=50: 19

Final errCompute for K=50: 0.75

Final M for K=50:
[[ -78.5003897    21.08581697]
 [ -60.41728246    2.50459298]
 [ -81.61079884   23.12217907]
 [ -75.63513411    1.36864112]
 [ -77.80826653    8.44729733]
 [ -73.83000415   31.46666024]
 [ -80.88631304   22.64815435]
 [ -62.0550837     6.36019239]
 [ -84.7389125    19.3018125 ]
 [ -80.30733868   21.9289327 ]
 [ -85.01129297   15.12495078]
 [ -80.51943714   22.36634571]
 [ -82.52721125   23.59941447]
 [ -88.33561472   14.97152987]
 [ -82.80288355   30.93412511]
 [ -84.57380823   12.84411433]
 [ -73.95693995   10.9077339 ]
 [ -91.18610086   28.15762672]
 [ -83.30476873   23.41057887]
 [ -64.97962577    3.69720123]
 [ -92.79468117   17.54891413]
 [ -80.03904792   21.64823437]
 [ -79.71662338   22.00901948]
 [-108.78156611   28.48072798]
 [ -79.14428444   27.27676222]
 [-105.51562376   26.39871386]
 [ -62.72621591    2.67137614]
 [ -74.89778232    6.34645182]
 [ -65.23463818    8.97893527]
 [ -99.65871327   26.38742506]
 [ -90.16603903   14.17957474]
 [ -81.35401209   26.05407692]
 [ -77.15798883   20.65017496]
 [ -95.10153684   16.75145428]
 [ -83.70395278   21.13268611]
 [ -77.77240448   13.47882313]
 [ -83.47652134   27.09753254]
 [ -70.12733217    9.98495701]
 [ -87.09068952   23.42834454]
 [ -76.1483916    15.99846681]
 [ -81.45777537   22.91228079]
 [-105.39548072   22.47772249]
 [ -81.29489899   22.79382929]
 [ -83.31509535    9.30848346]
 [ -88.5483953    13.9985604 ]
 [ -84.58326911   23.30495494]
 [ -84.00278727   25.38874286]
 [ -91.55096184   19.41816132]
 [ -87.09392069   12.99242529]
 [ -98.61346528   18.9215966 ]]

Final Group for K=50:
[[-84.5152  19.1264   8.    ]
 [-76.6465  20.2474  32.    ]
 [-76.6615  20.342   32.    ]
 ...
 [-80.6016  22.0641  11.    ]
 [-80.3845  21.9461   9.    ]
 [-75.4638  10.13    16.    ]]

Time taken for K=50:1.828s
```

Result Table (K=100):

```
Results for K=100 using random 100 objects from X
------------------------------------------------------------

Total number of iterations for K=100: 22

Final errCompute for K=100: 0.48

Final M for K=100:
[[ -73.46214583   11.16645167]
 [ -84.8633614    13.43961637]
 [ -75.91595714    7.97654643]
 [-108.93313131   29.79798687]
 [ -84.25729646   26.28866637]
 [ -82.60587167   23.34936   ]
 [-104.68061905   26.48613016]
 [ -83.34998554   25.59040248]
 [-108.88157812   26.02168516]
 [ -80.13619762    9.06584286]
 [ -74.80776016   15.25211951]
 [ -86.04050952   24.13001111]
 [ -74.38874353    1.53336118]
 [ -83.8811717    24.69214717]
 [ -80.67483      22.03741   ]
 [ -93.2015471    17.65791097]
 [-108.93931579   29.48935564]
 [ -83.63380052   26.80437827]
 [ -69.48265181   31.86369277]
 [ -94.88541466   16.58327368]
 [ -74.37751613   10.39289032]
 [ -92.74417231   17.58471385]
 [ -78.00125983    8.83832051]
 [-106.94041667   26.24824167]
 [ -72.78733879   10.71870948]
 [ -80.94731915   22.71448582]
 [ -77.7843013    20.87821818]
 [-108.27462651   29.12584096]
 [ -79.15563423   27.22578198]
 [ -80.3090523    21.90584028]
 [ -84.46464463   14.94184132]
 [-105.29888028   22.50459695]
 [ -91.39878677   18.86368521]
 [ -88.59534143   13.98043857]
 [-105.96292045   23.35742841]
 [ -77.46757685   13.43343796]
 [-108.36533673   27.76139286]
 [ -78.76326694   21.28001855]
 [ -85.91022564   15.25952308]
 [ -81.40185912   22.8839807 ]
 [ -89.786945     16.028165  ]
 [ -79.99154379   21.88466078]
 [ -75.55904485    9.90783382]
 [ -98.72467097   22.3590871 ]
 [ -74.79052899    6.13879822]
```

```
[ -99.02035556   25.12433407]
[ -97.38938441   18.36500376]
[ -73.91733123   11.24842943]
[ -84.34334066   30.45747473]
[ -78.28691413   21.06958696]
[ -74.95367629   31.363969  ]
[ -87.67651101   14.66015596]
[ -84.96526467   15.30527844]
[ -79.27609273   17.45531636]
[ -65.11035176    8.79243944]
[ -88.58963403   15.05567563]
[ -87.41174277   23.19447052]
[ -85.30226667   25.18071923]
[ -83.31431564   23.79601508]
[ -76.93254646    7.74475752]
[ -75.93935534    5.76333689]
[ -91.83976      20.61142167]
[ -81.14021688   26.05385455]
[-106.03123077   19.00438846]
[ -77.09339911   20.76222232]
[ -80.76833684   22.50188158]
[ -81.77742837   31.2425461 ]
[ -84.50614023   23.56646054]
[-100.00561348   27.09246779]
[ -83.16577432   27.66946027]
[ -93.70330526   26.75287368]
[ -62.62105566    4.09963491]
[ -83.60964467    9.4666145 ]
[ -88.15887364   14.72446909]
[ -84.18378403   20.2026521 ]
[ -80.41752418   22.34948462]
[ -77.4085557    20.72912437]
[ -92.28011939   17.35976224]
[ -83.28047544   23.45173579]
[-105.6057732    23.04042165]
[ -84.35709714   12.39136629]
[ -76.39766385    1.27203462]
[-105.04143262   22.10785106]
[ -82.50047888   23.63799442]
[-109.31925714   27.04836571]
[ -90.15311117   14.11539761]
[ -78.464965     21.2443675 ]
[ -83.33507227   23.06793594]
[-100.22938901   18.68901099]
[ -76.93098961   16.76588961]
[ -78.88095114   14.03386705]
[ -89.95977051   28.84199359]
[ -79.70089901   22.00810421]
[ -84.69434841   22.71010952]
[ -73.59689339    6.90961818]
[ -87.09392069   12.99242529]
[ -69.91291338   10.06193144]
[ -76.8331359    12.9958859 ]
[ -76.31484      20.27417565]
```

```
 [-108.77445283   28.30748019]]

Final Group for K=100:
[[-84.5152  19.1264  74.    ]
 [-76.6465  20.2474  98.    ]
 [-76.6615  20.342   98.    ]
 ...
 [-80.6016  22.0641  14.    ]
 [-80.3845  21.9461  29.    ]
 [-75.4638  10.13    42.    ]]

Time taken for K=100:4.141s
```