Name: Gaurav Misra

ID: 011449815

NMI: 0.1609

Rank : 35

Note: The program developed is only for k = 7. While developing the program, I didn't consider the fact that the number of clusters generated can be variable. I just noticed this at the time of writing the report, and I cannot update the code as there is no time left.

Its my sincere request that you do not deduct my marks for this reason.

Choice of internal evaluation metric: Mean deviation

Find the mean of all values ... use it to work out distances ... then find the mean of those distances.

In three steps:

- 1. Find the mean of all values
- 2. Find the **distance** of each value from that mean (subtract the mean from each value, ignore minus signs)
- 3. Then find the **mean of those distances.**

Approach:

Distance measure : Eucledian distance

The idea is iteratively splitting your cloud of points in 2 parts. In other words, you build a random binary tree where each splitting (a node with two children) corresponds to splitting the points of your cloud in 2.

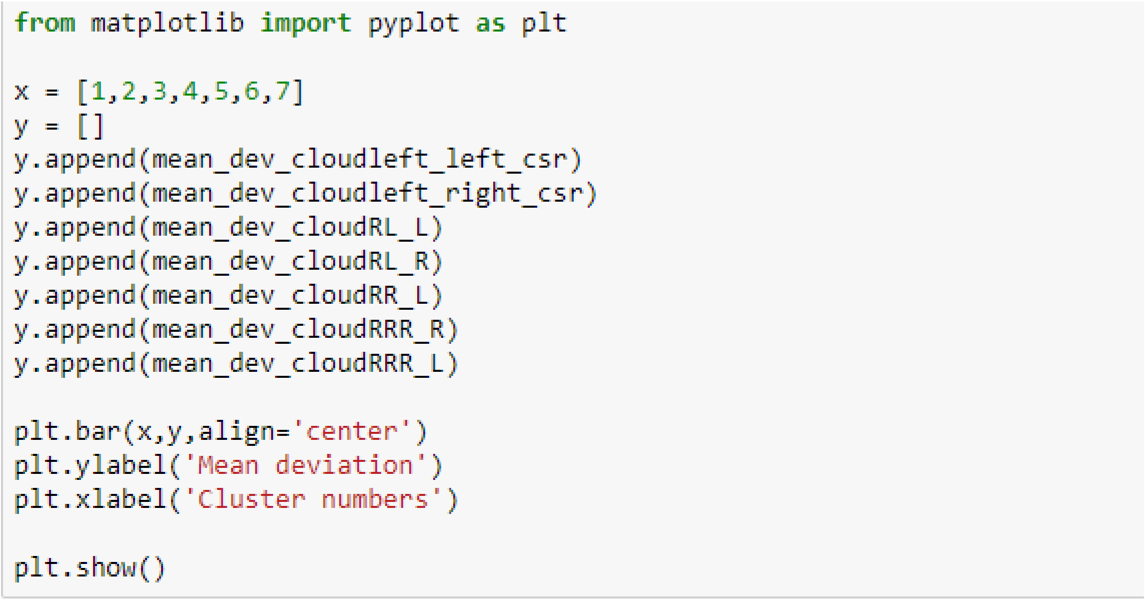
You begin with a cloud of points.

- Compute its centroid (barycenter) w
- Select a point at random cL among the points of the cloud(The points selected have mean deviation = mean+0.5/mean-0.5, for a given cloud of points).

- Construct the point cR as the symmetric point of cL when compared to w (the segment cL->w is the same as w->cR)(cR is found by searching for a point with mean deviation = mean(+0.5 or -0.5))
- Separate the points of your cloud in two, the ones closest to cR belong to a subcloud R, and the ones closest to cL belongs to the subcloud L
- Reiterate for the subclouds R and L

Note: Since the number of clouds = 7, I followed this approach until I created 4 clusters, after which I splitted clusters with more elements.

Mean deviation across clusters 1 to 7:

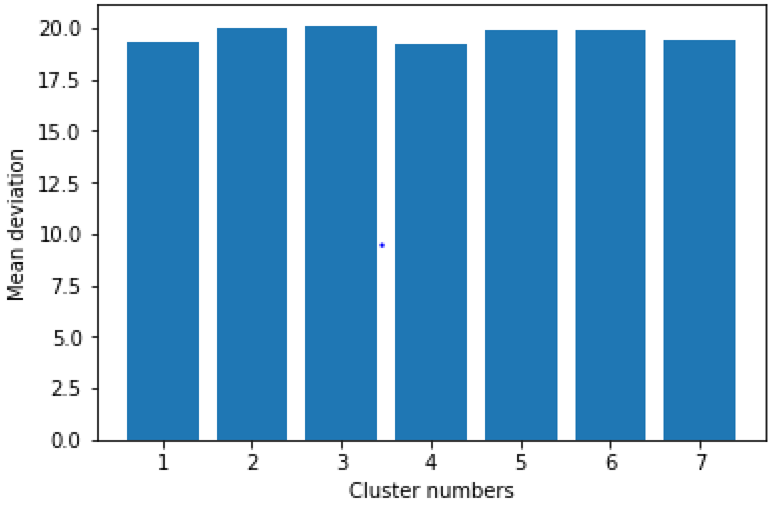


```
from matplotlib import pyplot as plt

x = [1,2,3,4,5,6,7]
y = []
y.append(mean_dev_cloudleft_left_csr)
y.append(mean_dev_cloudleft_right_csr)
y.append(mean_dev_cloudRL_L)
y.append(mean_dev_cloudRL_R)
y.append(mean_dev_cloudRR_L)
y.append(mean_dev_cloudRRR_R)
y.append(mean_dev_cloudRRR_L)

plt.bar(x,y,align='center')
plt.ylabel('Mean deviation')
plt.xlabel('Cluster numbers')

plt.show()
```


Mean deviation
Cluster numbers