Help Guide

Mean-Shift

Exercise: Create a Python script file and perform the following tasks:

Pseudocode:

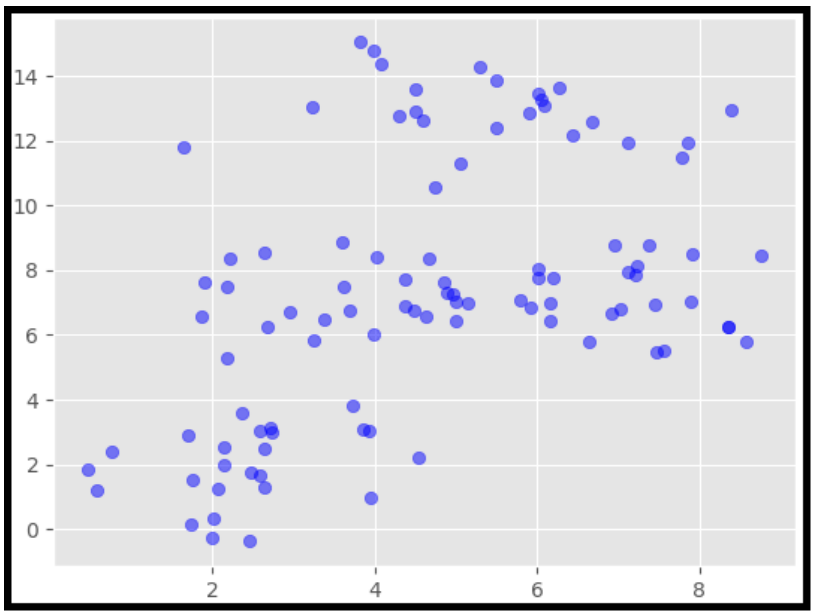
```
class Mean_Shift{

        constructor(radius){
                this.radius = radius
        }

        fit(data){
                initialize centroids with data points
                flag = True
                while(flag){
                        calculate new_centroids which is the result of the average of points
                        that are inside the circle with given radius
                        subtract the duplicate centroids
                        if(centroids == new_centroids){
                                flag = False
                        }
                 }
                 return centroids

        }

}
```

1. Import libraries:
   - numpy: For calculations of arrays.
   - cdist: To calculate Euclidean distance.
   - make_blobs: To create a dataset with points depending to input points and bandwidth value.
   - Matplotlib: To plot points in 2-dimension figure.

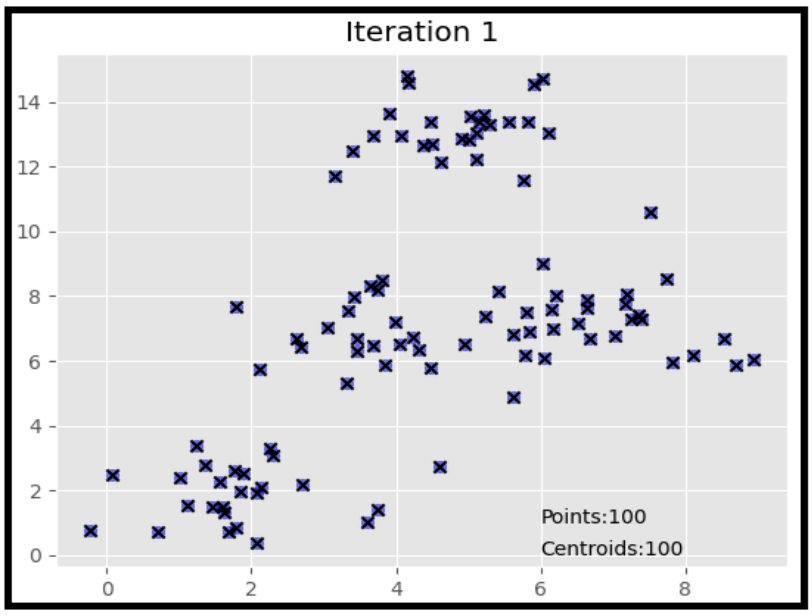
2. Create a class Mean Shift with the name Mean_Shift which contains a constructor _init_ and a function fit. The constructor takes radius as a parameter and has as default value 3, in case the user doesn't give a value. Function fit takes data as parameter. Radius is an int and data is a numpy.ndarray of 2-dimensions.

3. Load and use the dataset of sklearn make_blobs to create a dataset of 100 samples by giving input [[2, 2], [4, 7], [7, 7], [5, 13]] to centers parameter and

1 in cluster_std parameter. Load and use the library matplotlib to plot the points. You should do it like the picture below.



4. The algorithm starts by making all points centroids. Use the library matplotlib to plot the points. Plot the centroids with mark ‘x’. You should do it like the picture below.



5. Calculate new centroids by finding the points in the circle with center the centroid and radius (given value or 3) and find the mean point of them, which becomes the new centroid. Repeat this for each centroid.

Hint: Use method numpy.linalg.norm to calculate the distance between centroids and the rest of points.

6. You need to subtract duplicates centroids.

   Hint: Use method set() which transforms the list to set, because sets are no having duplicates and re transforms it to list again with list() method.

7. Repeat the last step until new centroids and previous ones are the same. Use a bool variable to end the while loop.

   Hint: Use a bool variable to end the while loop.

8. Plot the diagram of points and centroids with unique color for each cluster. Make a .gif file of the changes of centroids and points that change during the process.

   Hint: Plot a figure of each iteration and save them to a file. Use the site Animated Gif Maker to upload all figures and make the gif.