Assignment 3

1 Problem Statement

You have been provided with 6 datasets. You can download these datasets using this link. You need to identify clusters in these datasets using K-means clustering algorithm which we had seen in Week-3.

1. Before implementing K-means clustering algorithm, you must select an appropriate value for `k` (number of clusters). For this, you need to first visualize the dataset. You can do this with the help of a scatter plot provided in `matplotlib` library in Python. Each dataset consists of 3 columns `x`, `y` and `class`. Using the `x` and `y` values, plot a 2D scatter plot and the color of each data point is given according to the `class` value. Also add a legend to your scatter plot for reference. For example :-

	x	y	class
0	0.229506	0.823132	1
1	0.210182	0.771073	1
2	0.171068	0.857589	1
3	0.218243	0.913649	1
4	0.250728	0.807309	1
...	...	...	...
95	0.698913	0.276833	2
96	0.697731	0.296100	2
97	0.668168	0.345077	2
98	0.604743	0.329147	2
99	0.683261	0.346385	2

Figure 1: Dataset

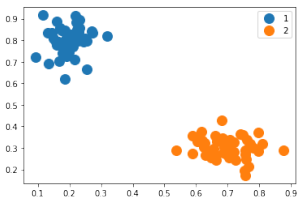

Figure 2: Scatter plot

2. Based on the legend of the scatter plot that you plotted in the first part, you can determine `k` (number of clusters). For the above example, there are 2 classes each represented by orange and blue respectively in the scatter plot and so the value of `k`, which will be used in K-means algorithm, will be 2. Using this value of `k`, perform K-means clustering on the given dataset. Avoid using `Kmeans` function provided by sklearn library in Python. You may use it to verify whether your program is implementing the K-means clustering algorithm properly or not but your final code shouldn't use this function. While implementing K-means algorithm, your function must not use the `class` values. The `class` values are meant to check whether your clustering algorithm has successfully grouped the dataset into different clusters or not.

3. Finally visualise these clusters using `matplotlib` library in Python. You must obtain a similar scatter plot as the one shown in the above example. Essentially, the K-means algorithm assigns a `class` value to each data point and using this `class` value as label, you plot the scatter plot similar to the one shown in the above example.

4. For some datasets, K-means algorithm successfully groups the dataset into different clusters which matches the clustering as per the class labels. In some cases, the algorithm is not able to group dataset into appropriate clusters. Report such datasets and provide a reasonable explanation on why the K-means algorithm failed for those datasets. For example :-

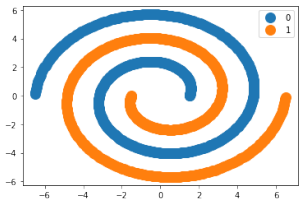

Figure 3: Original Dataset

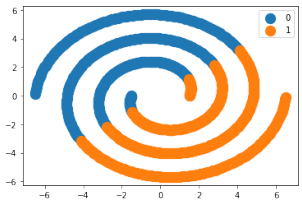

Figure 4: Scatter plot obtained after K-means

5. Finally, you need to submit a Python file/Juptyer Notebook which contains your code. Also provide 2 images for each dataset as shown in the above example (one scatter plot with labels as per the `class` values and other scatter plot obtained after K-means). If the K-means algorithm fails for a dataset then write the reason for failure in a single .txt file.

6. There are some datasets that I have given for fun. You will know it when you see the scatter plots of those datasets. You can try K-means with arbitrary `k` values and play around with it. You don't need to submit anything related to it and are just meant for fun.