AI Academy: Introduction to Data Mining
Week 3 Workshop

Workshop 3 contains 3 questions. Please read and follow the instructions.

---

1. K-Means Clustering. Use the K-means clustering algorithm with *Euclidean Distance* to cluster the 10 data points in Figure 1 into 3 clusters. Suppose that the initial seed centroids are at points: C, I and H. The data are also given in tabular format in Table 1.

    1. After each iteration of k-means, report the coordinates of the new centroids and which cluster each data point belongs to. **Stop when the algorithm converges and clearly label on the graph where the algorithm converges.** To report your work, give your answer in tabular format with the following attributes: **Round** (e.g. Round 1, 2, etc), **Points** (e.g. {A, B, C}), and **Cluster_ID** (order does not matter). Also report the **centroids** for each cluster after each round. Please followed the example table format in Table 2

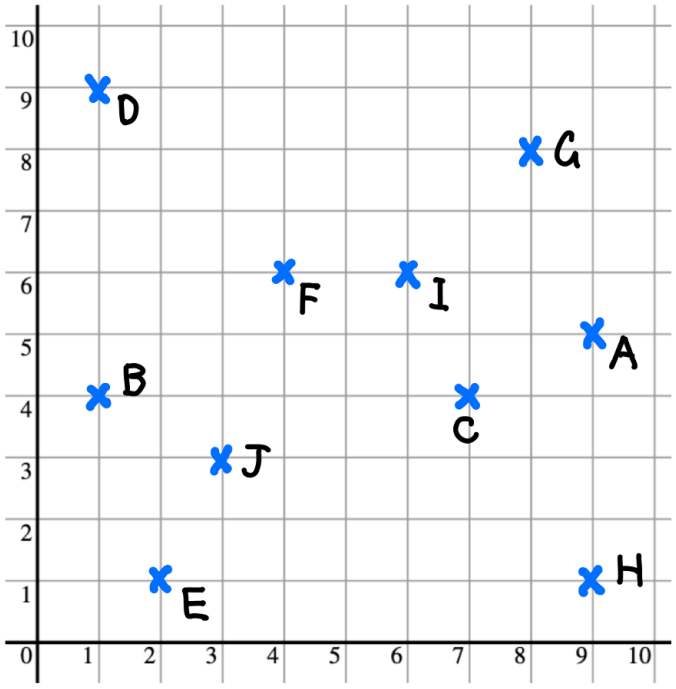

Figure 1: K-means Clustering (a)

Point	**x**	**y**
A	9	5
B	1	4
C	7	4
D	1	9
E	2	1
F	4	6
G	8	8
H	9	1
I	6	6
J	3	3

Table 1: K-means Clustering (b)

2. How many rounds are needed for the K-means clustering algorithm to converge?

2. Hierarchical Clustering (17 points) We will use the same dataset as in Question 1 (shown in Figure 1) for Hierarchical Clustering. The *Euclidean Distance* matrix between each pair of the datapoints is given in Figure 2 below:

    1. Perform *complete* link hierarchical clustering on the dataset. As above, show your calculations and report the corresponding dendrogram. Show your work at each iteration by giving the inter-cluster distances. Report your results by drawing a corresponding dendrogram. The dendrogram should clearly show the order and the height in which the clusters are merged. If possible, use a program to construct your dendrogram (e.g. PowerPoint, LucidChart[1], or VisualParadigm[2]). Scanned hand drawings will also be accepted if they are very clear. **NOTE**: There may be ties (i.e. two clusters have the same distance). In this case, you can choose any order to merge in, and ensure that this is reflected in your dendrogram.
    2. If we assume there are *two* clusters, will the *single* or *complete* link approach give a better clustering? Justify your answer.
    3. Consider the single-link hierarchical clustering with **3 clusters**. Compare the quality of this single link clustering with your final K-means clustering in Question 2. To evaluate the quality of each clustering, calculate its corresponding Sum of Squared Error (SSE). Based on this measure, which clustering (k-means, single link), is better? **Please show the SSE value for both clustering.** Note: you may want to write some code to help speed up these calculations, which you can include in lieu of showing your work.

	A	B	C	D	E	F2	G	H	I	J
A	0	8.06	2.24	8.94	8.06	5.1	3.16	4	3.16	6.32
B	8.06	0	6	5	3.16	3.61	8.06	8.54	5.39	2.24
C	2.24	6	0	7.81	5.83	3.61	4.12	3.61	2.24	4.12
D	8.94	5	7.81	0	8.06	4.24	7.07	11.31	5.83	6.32
E	8.06	3.16	5.83	8.06	0	5.39	9.22	7	6.4	2.24
F	5.1	3.61	3.61	4.24	5.39	0	4.47	7.07	2	3.16
G	3.16	8.06	4.12	7.07	9.22	4.47	0	7.07	2.83	7.07
H	4	8.54	3.61	11.31	7	7.07	7.07	0	5.83	6.32
I	3.16	5.39	2.24	5.83	6.4	2	2.83	5.83	0	4.24
J	6.32	2.24	4.12	6.32	2.24	3.16	7.07	6.32	4.24	0

Figure 2: Hierarchical Clustering Dataset

[1] https://www.lucidchart.com/
[2] https://online.visual-paradigm.com/features/dendrogram-software/

3. DBSCAN (10 Points). Now we're going to use a new dataset, shown in Figure 3. While you most likely not need it, the distance matrix for this dataset is also provided.

   For this problem, let **EPS=2**, and a **MinPoints=4**.

   1. Classify each point as either a core, border, or noise point.
   2. Give the final clusters (there may be more than one clustering).

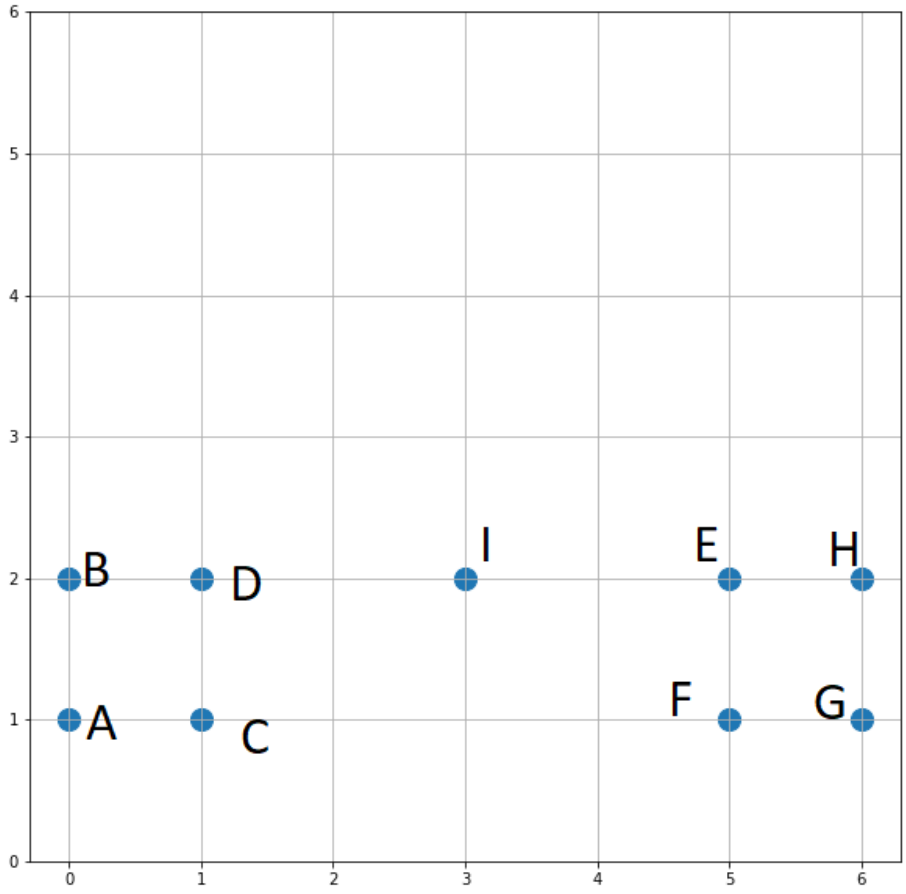

Figure 3: DBSCAN Plot

—	**A**	**B**	**C**	**D**	**E**	**F**	**J**	**H**	**I**
A	0.00	1.00	1.00	1.41	5.10	5.00	6.00	6.08	3.16
B	1.00	0.00	1.41	1.00	5.00	5.10	6.08	6.00	3.00
C	1.00	1.41	0.00	1.00	4.12	4.00	5.00	5.10	2.24
D	1.41	1.00	1.00	0.00	4.00	4.12	5.10	5.00	2.00
E	5.10	5.00	4.12	4.00	0.00	1.00	1.41	1.00	2.00
F	5.00	5.10	4.00	4.12	1.00	0.00	1.00	1.41	2.24
G	6.00	6.08	5.00	5.10	1.41	1.00	0.00	1.00	3.16
H	6.08	6.00	5.10	5.00	1.00	1.41	1.00	0.00	3.00
I	3.16	3.00	2.24	2.00	2.00	2.24	3.16	3.00	0.00

Table 3: Distance Matrix for DBSCAN