Worksheet 4: K-Means

1. Worksheet Questions

a. **(1-dimensional clustering) Walk through Lloyd's algorithm step by step on the following dataset:**

`[0, .5, 1.5, 2, 6, 6.5, 7]` (note: each of these are 1-dimensional data points)

Given the initial centroids:

`[0, 2]`

- **Step 1: Initialize Centroids**
  - We start with initial centroids:
  - $C_1 = 0, C_2 = 2$
- **Step 2: Assign Each Data Point to the Nearest Centroid**
  - We compute the absolute distance from each data point to the centroids and assign them to the closest one:

Data Point	Distance to $C_1 = 0$	Distance to $C_2 = 2$	Assigned Cluster
0	**0**	2	C_1 (0)
0.5	**0.5**	1.5	C_1 (0)
1.5	**1.5**	0.5	C_2 (2)
2	2	**0**	C_2 (2)
6	6	4	C_2 (2)
6.5	6.5	4.5	C_2 (2)
7	7	5	C_2 (2)

  - Clusters after assignment:
    - **Cluster 1 (C_1 = 0):** {0, 0.5}
    - **Cluster 2 (C_2 = 2):** {1.5, 2, 6, 6.5, 7}
- **Step 3: Update Centroids**
  - We compute the new centroids as the mean of each cluster:
  - $C_1' = \frac{0+0.5}{2} = 0.25$

- $C_2' = \frac{1.5+2+6+6.5+7}{5} = \frac{23}{5} = 4.6$
- Updated centroids: $C_1' = 0.25$, $C_2' = 4.6$

- **Step 4: Reassign Data Points to the New Centroids**
  - We compute the new distances and reassign clusters:

Data Point	Distance to $C_1 = 0.25$	Distance to $C_2 = 4.6$	Assigned Cluster
0	**0.25**	4.6	C_1 (0.25)
0.5	**0.25**	4.1	C_1 (0.25)
1.5	**1.25**	3.1	C_1 (0.25)
2	**1.75**	2.6	C_1 (0.25)
6	5.75	**1.4**	C_2 (4.6)
6.5	6.25	**1.9**	C_2 (4.6)
7	6.75	**2.4**	C_2 (4.6)

Clusters after reassignment:

- **Cluster 1 (C_1 = 0.25):** {0, 0.5, 1.5, 2}
- **Cluster 2 (C_2 = 4.6):** {6, 6.5, 7}

- **Step 5: Update the centroids again**

Step 5: Update Centroids Again

$$C_1' = \frac{0 + 0.5 + 1.5 + 2}{4} = 1$$

$$C_2' = \frac{6 + 6.5 + 7}{3} = 6.5$$

Updated centroids:
$C_1 = 1, C_2 = 6.5$

Step 6: Reassign Data Points to the New Centroids

Data Point	Distance to $C_1 = 1$	Distance to $C_2 = 6.5$	Assigned Cluster
0	**1**	6.5	C_1 (1)
0.5	**0.5**	6	C_1 (1)
1.5	**0.5**	5	C_1 (1)
2	**1**	4.5	C_1 (1)
6	5	**0.5**	C_2 (6.5)
6.5	5.5	**0**	C_2 (6.5)
7	6	**0.5**	C_2 (6.5)

Since the clusters haven't changed, the algorithm **converges**.

Final clusters:

- **Cluster 1 ($C_1 = 1$):** {0, 0.5, 1.5, 2}
- **Cluster 2 ($C_2 = 6.5$):** {6, 6.5, 7}

b. Describe in plain english what the cost function for k means is.

In simple terms, the **cost function** for k-means (also called the **within-cluster sum of squares, WCSS**) measures how well the data points are clustered. It calculates the sum of squared distances between each point and its assigned centroid:

$$J = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2$$

where:

- K is the number of clusters
- C_i is a cluster
- μ_i is the centroid of cluster C_i
- $||x - \mu_i||^2$ is the squared Euclidean distance between each point and its centroid

Minimizing this function ensures that data points are as close as possible to their cluster center.

c. **For the same number of clusters K, why could there be very different solutions to the K means algorithm on a given dataset?**

- K-means can yield different solutions due to the random initialization of centroids. The algorithm can get stuck in local minima, meaning the final clustering depends on the initial centroid placement. Some scenarios where different solutions arise:
  - Poor initial centroids lead to suboptimal clustering.
  - If the dataset has multiple natural groupings, different runs may cluster them differently.
  - Running k-means multiple times with different initial centroids (e.g., using k-means++) can help find a more stable solution.

d. **Do you think Lloyd's Algorithm always converges?**

- Yes, Lloyd’s algorithm always converges, but not necessarily to the optimal solution. The reasons are:
  - Each step (reassignment and updating centroids) reduces or keeps the cost function the same.
  - Since the cost function has a lower bound (zero), the algorithm must eventually stop.
  - However, convergence may lead to a local minimum, not necessarily the global optimal solution.
- To improve convergence:
  - Use k-means++ initialization to select better starting centroids.
  - Run the algorithm multiple times and pick the best clustering based on the cost function.