**1. What is clustering? List two common clustering methods.**

Clustering is unsupervised learning where you want to discover the inheritance of grouping data. So that we can find the similarity between the data and populate the results depending on those similarities we give better search results.

1. k-means clustering:

k-means clustering is model that aims to partition of n observations into k clusters in which observations belong to clusters with nearest mean (cluster centroid and cluster centers), serving as a prototype to the cluster.

1. Mean-Shift clustering:

Mean-shift clustering algorithm is like unsupervised clustering where groups of data are formed directly without being trained or labeled. The nature of the Mean Shift clustering algorithm is hierarchical in nature, which means it builds on a hierarchy of clusters, step by step.

**2. What are the steps of the k-means clustering algorithm?**

As k-means clustering algorithm contains n observations and k clusters

Step-1: First, we need to specify the number of clusters, K, that need to be generated by this algorithm.

Step-2: Next, randomly select K data points and assign each data point to a cluster. In simple words, classify the data based on the number of data points.

Step-3: Now it will compute the cluster centroids.

Step-4: Next, keep iterating the following until we find optimal centroid which is the assignment of data points to the clusters that are not changing any more −

4.1. First, the sum of squared distance between data points and centroids would be computed.

4.2. Now, we must assign each data point to the cluster that is closer than other cluster (centroid).

4.3. Finally compute the centroids for the clusters by taking the average of all data points of that cluster.

**3. The following dataset has example A – E, the single attribute is X.**

**Example**  **A**  **B**  **C**  **D**  **E**

**Attribute Value (X)** **0.1**  **0.6**  **0.8**  **2.0**  **3.0**

**Apply k-Means Clustering to this data set to produce two data clusters (k=2). Assuming initially, we have chosen example A to initialize cluster #1 and example B to initialize cluster #2.**

**1) Apply k-Means Clustering algorithm, write down the cluster assignment and their centroids (means) on each iteration:**

**Iteration0:**  **Cluster #1: A**  **Cluster #2: B**

**Centroid #1: \_\_0.1\_\_\_\_\_\_\_\_\_**  **Centroid #2: \_\_\_0.6\_\_\_\_\_\_\_\_**

**Iteration1:**  **Cluster #1:\_\_\_A\_\_\_\_\_\_\_\_\_**  **Cluster #2: \_\_\_B\_\_\_\_\_\_\_\_**

**Centroid #1: \_\_\_0.1\_\_\_\_\_\_\_\_**  **Centroid #2: \_\_\_1.6\_\_\_\_\_\_\_\_**

**Iteration2:**  **Cluster #1:\_\_\_A\_\_\_\_\_\_\_\_\_**  **Cluster #2: \_\_\_B\_\_\_\_\_\_\_\_**

**Centroid #1: \_\_\_0.5\_\_\_\_\_\_\_\_**  **Centroid #2: \_\_\_2.5\_\_\_\_\_\_\_\_**

**Iteration3:**  **Cluster #1:\_\_\_A\_\_\_\_\_\_\_\_\_**  **Cluster #2: \_\_\_B\_\_\_\_\_\_\_\_**

**Centroid #1: \_\_\_0.5\_\_\_\_\_\_\_\_**  **Centroid #2: \_\_\_2.5\_\_\_\_\_\_\_\_**

**2) In what condition will the iteration stop?**

When the centroids are A = 0.5 and B = 2.5.

**3) What are the final cluster assignments and their centroids?**

Clusters assigned:

A = {0.1, 0.6, 0.8}

B = {2.0, 3.0}

Centroids

A= 0.5

B= 2.5

Please find image files for the detailed solution.

**4. Can you represent the following function with a perceptron (i.e., a single unit from a neural network)? If yes, draw the diagram and show the weights and threshold values. If not, explain why not in 1-2 sentences.**

**1) Logical NOT gate**

**2) Logical AND gate**

**3) Logical OR gate**

**4) Logical XOR gate.**

A single-layer perceptron is the basic unit of a neural network. A perceptron consists of input values, weights and a bias, a weighted sum and activation function.

Single-layer perceptron function looks like

Y = X1W1+X2W2+X3W3

Activation functions are:

1. Hyperbolic Functions which give output as –1 or 1

2. Logistic Functions which give output as 0 or 1 => g(x) = 1 / (1+exp(-x)); {0,1}

For logical gates to perform single-layer perceptron we should not have any hidden layers in between.

For NOT, AND, OR Gate we can perform single-layer perceptron for supervised learning.

XOR uses multi-layer perceptron supervised learning. XOR Gate needs hidden layers to perform activation functions like sigmoid Activation function.

The diagrams look like:

Function: Σ (Xi \* Wi) + bias {Activation function gives us the output depending on the functions which we are going to see using Logical gates.}

So, for each gate the supervised learning happens with the activation function, which helps us in dividing the clusters.

Let us take an Example to demonstrate logical gates using perceptron and activation function concepts.

Let us consider 2 datasets (be X, Y) of same length with random numbers in it.

Like:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| X | 0.6 | 0.9 | 0.56 | 2 | 0.9 | 0.5 |
| Y | 0.7 | 5.6 | 1.5 | 0.5 | 0.7 | 0.54 |

We want to do some supervised learning on the above dataset, to do so we need to calculate the activation function. So, the activation function for the logical operators are as follows:

1. **Logical NOT gate:**

We use only one dataset for this operation, we check the number with NOT gate condition and populate the output accordingly.

g(x) = Σ (Xi \* Wi) + bias {if >=0 then 1 else if <0 then 0}

|  |  |
| --- | --- |
| X | NOT |
| 0 | 1 |
| 1 | 0 |

1. **Logical AND gate**

g(x) = Σ (Xi \* Wi) + bias {if >=0 then 1 else if <0 then 0}

g(Y) = Σ (Yi \* Wi) + bias {if >=0 then 1 else if <0 then 0}

|  |  |  |
| --- | --- | --- |
| X | Y | AND |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

1. **Logical OR gate**

g(x) = Σ (Xi \* Wi) + bias {if >=0 then 1 else if <0 then 0}

g(y) = Σ (Yi \* Wi) + bias {if >=0 then 1 else if <0 then 0}

|  |  |  |
| --- | --- | --- |
| X | Y | OR |
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |