**1. Support Vector Machines (SVM)**

**Key Points**

* **Purpose**: SVM is used for classification and regression tasks.
* **Objective**: Find the best boundary (hyperplane) that separates different classes.

**How SVM Works**

* **Hyperplane**: A decision boundary that separates classes.
* **Support Vectors**: Points closest to the hyperplane. They help define the position and orientation of the hyperplane.
* **Margin**: Distance between the hyperplane and support vectors. SVM maximizes this margin for better separation.

**Types of SVM**

* **Linear SVM**: Works with linearly separable data.
* **Non-Linear SVM**: Uses kernel functions to handle data that is not linearly separable.

**Common Kernels**

* **Linear Kernel**: For linear separation.
* **Polynomial Kernel**: Uses polynomial functions to transform data.
* **RBF Kernel**: Uses radial basis functions for complex data transformations.

**2. K-Nearest Neighbors (KNN)**

**Key Points**

* **Purpose**: KNN is used for classification and regression tasks.
* **Objective**: Classify a point based on the majority class of its neighbors.

**How KNN Works**

* **K**: Number of nearest neighbors to consider.
* **Distance Metric**: Measure distance between points (e.g., Euclidean distance).
* **Classification**: Assign the class most common among the K nearest neighbors.

**Steps in KNN**

1. **Choose K**: Select the number of neighbors.
2. **Compute Distance**: Calculate the distance between the point and all other points.
3. **Find Neighbors**: Identify the K nearest neighbors.
4. **Majority Vote**: Assign the class that is most frequent among the neighbors.

**Advantages and Disadvantages**

* **Advantages**:
  + Simple to understand.
  + No training phase (instance-based learning).
* **Disadvantages**:
  + Can be slow with large datasets.
  + Performance depends on the choice of K and distance metric.

**3. K-Means Clustering**

**Key Points**

* **Purpose**: K-Means is used for unsupervised learning to group data into clusters.
* **Objective**: Partition data into K clusters where each point belongs to the cluster with the nearest mean.

**How K-Means Works**

* **K**: Number of clusters.
* **Centroids**: Central point of each cluster.

**Steps in K-Means**

1. **Choose K**: Select the number of clusters.
2. **Initialize Centroids**: Randomly pick K points as initial centroids.
3. **Assign Points**: Assign each point to the nearest centroid.
4. **Update Centroids**: Recalculate the centroids as the mean of all points in each cluster.
5. **Repeat**: Repeat the assign and update steps until centroids no longer change significantly.

**Advantages and Disadvantages**

* **Advantages**:
  + Simple and easy to implement.
  + Works well with large datasets.
* **Disadvantages**:
  + Requires specifying the number of clusters (K) in advance.
  + Can converge to local minima (results can vary with different initial centroids).
  + Assumes clusters are spherical and of similar size.