**1. How does K-Means clustering work?**

**K-Means** is an **unsupervised learning algorithm** used to partition data into **K distinct clusters**. It works as follows:

1. **Initialize**: Choose K initial centroids (randomly or using smarter methods like k-means++).
2. **Assign**: Assign each data point to the nearest centroid based on a distance metric (usually Euclidean distance).
3. **Update**: Compute new centroids by taking the mean of all points assigned to each cluster.
4. **Repeat**: Repeat the assign and update steps until centroids stabilize (no/little change) or a maximum number of iterations is reached.

**2. What is the Elbow method?**

The **Elbow method** helps determine the optimal number of clusters (K) for K-Means:

* Run K-Means with different values of K.
* Plot **inertia** (sum of squared distances from points to their cluster center) vs. K.
* Look for the "elbow" point—where the rate of decrease sharply slows—this is a good balance between underfitting and overfitting.

**3. What are the limitations of K-Means?**

* **Assumes spherical clusters**: Performs poorly on non-globular shapes.
* **Sensitive to initialization**: Can converge to local minima.
* **Fixed number of clusters**: You must specify K beforehand.
* **Sensitive to outliers**: Outliers can distort centroids.
* **Uneven cluster sizes**: Struggles with clusters of different densities or sizes.

**4. How does initialization affect results?**

Poor **initialization** can lead to:

* **Suboptimal clustering** (bad local minima).
* **Inconsistent results** on different runs.

Using **k-means++** initialization improves the chances of better clustering by spreading initial centroids apart.

**5. What is inertia in K-Means?**

**Inertia** measures **how tightly data points are grouped** around their centroids:

Inertia=∑i=1k∑x∈Ci∥x−μi∥2\text{Inertia} = \sum\_{i=1}^{k} \sum\_{x \in C\_i} \|x - \mu\_i\|^2Inertia=i=1∑k​x∈Ci​∑​∥x−μi​∥2

Where CiC\_iCi​ is the set of points in cluster iii and μi\mu\_iμi​ is the centroid.

* Lower inertia = tighter clusters.
* However, inertia **always decreases** as K increases, so it can't be used alone to pick K.

**6. What is Silhouette Score?**

**Silhouette Score** evaluates **how well each data point fits within its cluster**:

Score=b−amax⁡(a,b)\text{Score} = \frac{b - a}{\max(a, b)}Score=max(a,b)b−a​

* aaa = average distance to other points in the same cluster.
* bbb = average distance to points in the nearest neighboring cluster.
* Score ranges from **-1 to 1**:
  + **Close to 1**: well-clustered.
  + **Close to 0**: overlapping clusters.
  + **Negative**: misclassified points.

**7. How do you choose the right number of clusters?**

Common methods:

* **Elbow Method** (based on inertia).
* **Silhouette Score** (higher is better).
* **Gap Statistic**.
* **Domain knowledge**.
* **Cross-validation** (for clustering + downstream supervised tasks).

**8. What’s the difference between clustering and classification?**

| **Feature** | **Clustering** | **Classification** |
| --- | --- | --- |
| Type | Unsupervised learning | Supervised learning |
| Labels | No prior labels | Uses labeled data |
| Goal | Find structure/patterns in data | Predict predefined labels |
| Example | Grouping customers by behavior | Identifying spam emails |