Clustering - Theoretical Questions

1. ***What is unsupervised learning in the context of machine learning***

   Unsupervised learning is a type of machine learning that analyses and clusters unlabelled datasets to discover hidden patterns or groupings without explicit human guidance. Unlike supervised learning, which uses pre-labelled data to train a model for specific outcomes, unsupervised learning algorithms explore raw, unclassified data on their own.

---

2. ***How does K-Means clustering algorithm work***

   K-Means is an unsupervised machine learning algorithm that groups unlabelled data into a specified number of clusters, denoted by *K*. The algorithm is iterative, and its main goal is to minimize the total variance within each cluster, making the data points within a cluster as similar as possible.

   Here is a step-by-step breakdown of how the K-Means algorithm works:

   - ✓ Choose the number of clusters, *K*.
   - ✓ Initialize *K* centroids. These are typically randomly selected from the dataset, though more advanced methods like K-Means++ can improve results.
   - ✓ Assign data points to the nearest centroid. Each point is assigned to the cluster whose centroid is closest, often measured using squared Euclidean distance.
   - ✓ Recalculate the centroids. The new centroid for each cluster is the mean of all assigned data points.
   - ✓ Repeat until convergence. Steps 3 and 4 are repeated until centroids stabilize, assignments don't change, or a maximum iteration limit is reached.

---

3. ***Explain the concept of a dendrogram in hierarchical clustering***

   A dendrogram is a tree-like diagram that visualizes the output of hierarchical clustering, showing how clusters are formed and merged step-by-step in a hierarchy. The data points are at the bottom (leaves), and as you move up, they merge into clusters. The height of the vertical lines (or the y-axis) represents the dissimilarity between clusters, with lower merges indicating greater similarity. A horizontal line drawn across the dendrogram reveals the clusters at that specific dissimilarity level.

   How a Dendrogram Works
   - ✓ **Bottom-Up (Agglomerative) Approach**:
     Most dendrograms are created using a bottom-up approach, starting with each data point as its own cluster.
   - ✓ **Merging Clusters**:
     At each step, the algorithm merges the two most similar clusters.
   - ✓ **Visualizing the Hierarchy**:
     The dendrogram shows this process:
     - **Leaves**: The individual data points are shown as the leaves of the tree.
     - **Branches**: Each branch represents a cluster, and the point where two branches join signifies that those two clusters have merged.
     - **Y-axis (Height)**: The height of the merge point (the vertical axis) indicates the distance or dissimilarity between the two merged clusters.
   - ✓ **Determining Clusters**:
     You can decide on the number of clusters by drawing a horizontal line across the dendrogram at a chosen height.

   Key Components and Interpretation

- ✓ **Leaves**: The data points or individual observations at the bottom.
- ✓ **Nodes**: The points where branches join, representing merged clusters.
- ✓ **Height**: The dissimilarity (or distance) between the clusters that merged to form that node.
- ✓ **Cut-off Line**: A horizontal line that you draw to group the data. The number of vertical lines your cut-off line crosses gives you the number of clusters at that specific level of similarity.

Uses of a Dendrogram

- ✓ **Exploratory Data Analysis**: It helps reveal natural groupings and patterns within the data.
- ✓ **Determining the Number of Clusters**: You can visually identify the optimal number of clusters by finding a suitable height for the cut-off line.
- ✓ **Identifying Relationships**: It provides a visual representation of the hierarchical relationships between data points and clusters.

---

4. *What is the main difference between K-Means and Hierarchical Clustering*

The main difference is that K-Means requires a predefined number of clusters (K) and partitions data into flat, non-overlapping groups, while Hierarchical Clustering does not require a predefined K and creates a tree-like hierarchy (dendrogram) of nested clusters. K-Means is generally faster but works best with spherical data and known cluster counts, whereas Hierarchical Clustering is more flexible, can reveal nested relationships, and allows for cluster selection after analysis via the dendrogram.

---

5. *What are the advantages of DBSCAN over K-Means*

DBSCAN excels over K-Means by handling clusters of arbitrary shapes, not just spherical ones, and by automatically identifying and excluding outliers from the dataset without requiring the number of clusters to be pre-specified. In contrast, K-Means requires a set number of clusters (K), assumes clusters are equally sized and spherical, and is sensitive to noise and outliers.

Key Advantages of DBSCAN Over K-Means:

- ✓ **Arbitrary Cluster Shapes:**
  DBSCAN can discover clusters of any shape because it works based on data density, whereas K-Means assumes clusters are compact and spherical.
- ✓ **Noise and Outlier Handling:**
  DBSCAN effectively identifies and separates noisy data points from clusters, treating them as outliers, a capability K-Means lacks.
- ✓ **No Need to Specify Cluster Count (K):**
  DBSCAN does not require the number of clusters to be defined in advance; it determines the number of clusters organically based on the data's density.
- ✓ **Robustness to Noise:**
  Due to its density-based nature, DBSCAN is more resistant to noise in the data compared to K-Means.

When K-Means is Not Ideal:

- ✓ When clusters are not spherical.
- ✓ When there is significant noise or many outliers in the data.
- ✓ When you don't know the optimal number of clusters beforehand.
- ✓ DBSCAN vs. K-Means: A Guide in Python - New Horizons
- ✓ Difference between K-Means and DBScan Clustering

---

6. *When would you use Silhouette Score in clustering*

You would use Silhouette Score in clustering to evaluate the quality of a clustering result by measuring the cohesion (how well each point fits its cluster) and separation (how distinct clusters are from one another). Key uses include determining the optimal number of clusters (k) for an algorithm, validating a chosen k, and comparing the performance of different clustering algorithms or parameter settings.

When to Use Silhouette Score

- ✓ **To determine the optimal number of clusters (k):**
  You can run a clustering algorithm for various values of k and calculate the silhouette score for each k. The k value that yields the highest average silhouette score is often the most appropriate.
- ✓ **To validate a chosen k:**
  After a clustering algorithm is run with a specific k, the silhouette score can provide a quantitative measure of how well-defined and separated the resulting clusters are.
- ✓ **To compare clustering algorithms:**
  You can use the silhouette score to assess and compare the performance of different clustering algorithms on the same dataset to see which one produces more robust and well-defined clusters.
- ✓ **To assess the overall quality of the clustering result:**
  A higher silhouette score (closer to +1) indicates that the data points are well-clustered, with strong cohesion and separation, while a lower score (closer to 0 or negative) suggests poor clustering.

----

7. *What are the limitations of Hierarchical Clustering*

Hierarchical clustering is limited by its high computational cost and difficulty scaling to large datasets, its sensitivity to outliers and noise, and the irreversible nature of its merge or split decisions, which can lead to suboptimal results. The interpretation of the dendrogram can be complex, and the algorithm can perform poorly with high-dimensional data or datasets with missing values.

----

8. *Why is feature scaling important in clustering algorithms like K-Means*

Feature scaling is critical for K-Means because it's a distance-based algorithm, and features with larger numerical ranges would dominate the distance calculations, leading to biased clusters. Scaling ensures all features contribute equally, preventing any single feature from being disproportionately influential in determining cluster assignments and improving the overall performance and speed of the algorithm.

----

9. *How does DBSCAN identify noise points*

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) identifies noise points based on their density relative to other data points, using two key parameters:

- ✓ **Epsilon (ε):** This defines the radius of the neighbourhood around a data point.
- ✓ **Minimum Points (MinPts):** This specifies the minimum number of points required within a point's ε-neighbourhood for it to be considered a "core point."

Here's how DBSCAN categorizes points, including noise:

- ✓ **Core Points:**
  A point is a core point if its ε-neighbourhood contains at least MinPts number of points (including itself). Core points are the fundamental building blocks of clusters, as they represent dense regions.
- ✓ **Border Points:**
  A point is a border point if it is not a core point itself, but it lies within the ε-neighbourhood of a core point. Border points are essentially on the "edge" of a cluster, extending its reach but not contributing to its core density.
- ✓ **Noise Points:**
  A point is classified as a noise point if it is neither a core point nor a border point. This means that its ε-neighbourhood does not contain MinPts points, and it is not within the ε-neighbourhood of any core point. Noise points are isolated or lie in low-density regions, far from any significant cluster.

DBSCAN identifies noise points as those that are not sufficiently "dense" or "connected" to any cluster based on the defined ε and MinPts parameters. They are the outliers that do not fit into any of the identified dense regions.

---

10. Define inertia in the context of K-Means

It is calculated by measuring the distance between each data point and its centroid, squaring this distance, and summing these squares across one cluster. A good model is one with low inertia AND a low number of clusters (K). However, this is a trade-off because as K increases, inertia decreases.

---

11. What is the elbow method in K-Means clustering

The elbow method is a heuristic method used in k-means clustering to find the optimal number of clusters (k) for a dataset. It works by plotting the within-cluster sum of squares (WCSS) for different values of k and identifying the "elbow" point, which indicates the optimal number of clusters.

Here's a more detailed explanation:

- ✓ **Calculate WCSS for different k:**
  For a range of potential k values (e.g., 1 to 10), run k-means clustering with each k. For each clustering, calculate the WCSS, which is the sum of squared distances between each data point and its assigned cluster centroid.
- ✓ **Plot the results:**
  Create a plot with the number of clusters (k) on the x-axis and the corresponding WCSS on the y-axis.
- ✓ **Identify the elbow:**
  The plot will typically show a decrease in WCSS as k increases. The "elbow" point is the location where the decrease in WCSS starts to slow down, forming a shape similar to an elbow on the curve. This point suggests that adding more clusters beyond this point does not significantly improve the clustering.
- ✓ **Optimal k:**
  The k value at the elbow point is considered the optimal number of clusters for your data. Some sources explain the elbow point as the point where the rate of decrease in WCSS starts to slow down significantly.

---

12. Describe the concept of "density" in DBSCAN

In DBSCAN, "density" refers to the presence of many data points clustered closely together, defining a high-density region. The algorithm identifies clusters as areas of high point density separated by low-density areas. DBSCAN uses two main parameters, epsilon (the radius for neighbourhood search) and minPts (the minimum number of points within that radius), to determine if a region is dense enough to form a cluster.

---

13. Can hierarchical clustering be used on categorical data

Yes, hierarchical clustering can be used with categorical data by employing specialized distance metrics, such as Gower's distance, which are designed to quantify dissimilarity between non-numerical attributes. Traditional distance measures like Euclidean distance don't apply to categorical data, so alternative dissimilarity measures are needed to group data points into a hierarchical structure represented by a dendrogram.

How it works:

- ✓ **Choose a dissimilarity metric:**
  Instead of Euclidean distance, you select a measure that works for categorical or mixed data types.
  - **Gower's Distance:** A popular choice for mixed data types, including categorical ones, it calculates the difference for each variable and then averages them.
  - **Hamming Distance or Jaccard Distance:** These are other options for calculating similarities between categorical data points.
- ✓ **Calculate the distance matrix:**

A matrix is created that shows the dissimilarity between every pair of data points based on your chosen metric.

- ✓ **Build the dendrogram:**
  The algorithm uses the dissimilarity matrix to group data points into a hierarchy.
  - **Agglomerative (bottom-up):** Starts with each data point as its own cluster and merges them based on similarity.
  - **Divisive (top-down):** Starts with one large cluster and repeatedly splits it into smaller ones.
- ✓ **Visualize the hierarchy:**
  The result is a dendrogram, a tree-like diagram that illustrates the nested structure of the clusters and helps in choosing an appropriate number of clusters.

Key considerations:

- ✓ **Data preprocessing:**
  Categorical variables often need to be properly encoded or transformed before clustering can be applied effectively.
- ✓ **Choice of metric:**
  The selection of the dissimilarity measure is crucial and significantly impacts the clustering results.
- ✓ **Variable weighting:**
  Some methods, like Gower's distance via the daisy() function in R, allow for weighing features to control their impact on the dissimilarity scores.

---

14. What does a negative Silhouette Score indicate

A negative Silhouette Score for a data point indicates that it has likely been assigned to the wrong cluster. It is closer, on average, to the data points in a neighbouring cluster than it is to the points in its own assigned cluster.
The Silhouette Score, which ranges from -1 to +1, measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation).

---

15. Explain the term "linkage criteria" in hierarchical clustering

In hierarchical clustering, linkage criteria define how to measure the distance between two clusters, influencing the shape and structure of the resulting clusters by determining which clusters are merged at each step. Common linkage methods include single linkage (minimum distance between any two points), complete linkage (maximum distance), average linkage (average of all pairwise distances), and Ward's method (minimizing within-cluster variance).

---

16. Why might K-Means clustering perform poorly on data with varying cluster sizes or densities

K-Means clustering may perform poorly on data with varying cluster sizes or densities due to its inherent assumptions about cluster characteristics:

- ✓ **Assumption of Spherical and Evenly Sized Clusters:**
  K-Means implicitly assumes that clusters are roughly spherical in shape and have similar sizes. This is because it defines clusters based on the mean (centroid) of data points and assigns points to the closest centroid. When clusters have significantly different sizes, the larger clusters can exert a stronger pull on the centroids, potentially causing smaller clusters to be absorbed or misrepresented.
- ✓ **Sensitivity to Density Variations:**
  K-Means does not inherently account for varying densities within clusters. It treats all data points equally in its distance calculations, regardless of how densely packed they are. In data with varying densities, a sparse cluster might be incorrectly merged with a denser, larger cluster, or a dense, small cluster might be overlooked if its centroid is not well-positioned initially.
- ✓ **Impact of Outliers:**

K-Means is sensitive to outliers. Outliers, especially in sparse regions, can significantly shift the position of a cluster centroid, distorting the cluster boundaries and potentially causing misclassification of data points, particularly in clusters with lower densities or smaller sizes.

- ✓ **Reliance on Euclidean Distance:**
  K-Means typically uses Euclidean distance to measure similarity and assign points to clusters. This metric works well for compact, spherical clusters but can be less effective for clusters with irregular shapes or varying densities, where other distance metrics or density-based approaches might be more appropriate.

---

17. What are the core parameters in DBSCAN, and how do they influence clustering

The two core parameters in DBSCAN are ε (epsilon), which defines the radius for a point's neighbourhood, and MinPts, the minimum number of points required to form a dense region. Epsilon influences cluster granularity: a smaller ε creates more, smaller clusters, while a larger ε groups more points together. MinPts influences noise tolerance and cluster density: higher MinPts values lead to fewer, more robust clusters, while lower values produce more, smaller, and potentially sparser clusters.

---

18. How does K-Means++ improve upon standard K-Means initialization

K-Means++ improves upon standard K-Means initialization by employing a more intelligent and deterministic strategy for selecting the initial cluster centroids, rather than relying solely on random selection. This approach aims to place initial centroids in a way that is more likely to lead to better and more consistent clustering results.

Here's how K-Means++ achieves this improvement:

- ✓ **First Centroid Selection:**
  K-Means++ begins by randomly selecting the first cluster centroid from the dataset, similar to standard K-Means.
- ✓ **Subsequent Centroid Selection based on Distance:**
  For each subsequent centroid, K-Means++ calculates the squared distance of every data point to its nearest already selected centroid. The next centroid is then chosen from the remaining data points with a probability proportional to this squared distance. This means that data points that are far away from all existing centroids have a higher chance of being selected as the next centroid.

Benefits of K-Means++ Initialization:

- ✓ **Better Initial Centroid Spread:**
  By prioritizing points that are farther from existing centroids, K-Means++ encourages a more dispersed and representative initial placement of centroids across the data space. This helps to avoid situations where all initial centroids are clustered together in a small region, leading to suboptimal clustering.
- ✓ **Improved Convergence:**
  The more strategic initialization in K-Means++ generally leads to faster convergence of the K-Means algorithm to a stable and often better local optimum.
- ✓ **Increased Robustness:**
  K-Means++ is more robust to the initial random seed compared to standard K-Means, as the selection process reduces the likelihood of encountering poor initializations that can trap the algorithm in undesirable local minima.

---

19. What is agglomerative clustering

Agglomerative clustering is a hierarchical clustering algorithm that works by iteratively merging the closest clusters until all data points are in a single cluster. It's a "bottom-up" approach, starting with each data point as its own cluster and progressively merging them based on similarity. The result is a dendrogram, a tree-like structure that visualizes the hierarchical relationships between clusters.

---

20. What makes Silhouette Score a better metric than just inertia for model evaluation

Silhouette Score is a better metric than inertia for evaluating clusters because it balances cohesion (how tight an object is to its own cluster) with separation (how far it is from other clusters), providing a more complete picture of cluster quality than inertia's focus solely on compactness. Inertia alone can be misleading, as it always decreases with more clusters, making it difficult to identify an optimal number of clusters.

Here's why Silhouette Score is superior:

- ✓ **Measures Both Cohesion and Separation:**
  - **Cohesion:** Measures how similar a data point is to other points within its own cluster.
  - **Separation:** Measures how different a data point is from points in other clusters.
  - **Inertia:** Primarily measures compactness (cohesion), but doesn't account for separation between clusters.
- ✓ **Provides a Balanced View of Cluster Quality:**
  - A high silhouette score indicates that clusters are both tight and well-separated.
  - A score near 0 suggests that data points are on the boundaries of clusters, while a negative score means a point may be misplaced in its cluster.
- ✓ **Helps Identify Optimal Number of Clusters:**
  - ▪ Unlike inertia, which always decreases as the number of clusters increases, the silhouette score increases and then decreases.
  - ▪ The number of clusters that yields the highest silhouette score is often the optimal number of clusters for the dataset.
- ✓ **Offers More Nuanced Insights:**
  - ▪ It offers more nuanced insights into cluster quality than inertia alone, giving a more comprehensive evaluation of clustering algorithms.

While inertia is a useful measure of cluster compactness, the silhouette score gives a more complete and meaningful evaluation of the overall quality of the clustering solution.
