K-Nearest Neighbours

**Interview Questions:**

1. What are the key hyper parameters in KNN?

2. What distance metrics can be used in KNN?

### 1. **What are the key hyper parameters in KNN?**

The key hyper parameters in K-Nearest Neighbors (KNN) include:

* **Number of Neighbors (k):** Specifies how many closest neighbors will contribute to making a prediction. Choosing kkk affects the balance between bias and variance in the model.
* **Distance Metric:** Defines how distance is calculated between data points. Common metrics are Euclidean, Manhattan, Minkowski, and Hamming.
* **Weights:** Determines whether each neighbor's vote is weighted equally or if closer neighbors have more influence. Options include “uniform” (equal weights) or “distance-based” (closer neighbors have more influence).
* **Algorithm for Finding Nearest Neighbors:** Choices include brute force, KD Tree, Ball Tree, and auto (to select the best option based on the data).
* **Leaf Size:** Relevant for tree-based algorithms (KD Tree and Ball Tree), it controls the number of points in the leaves, impacting speed and memory use.
* **p (Minkowski Distance Power Parameter):** When using Minkowski distance, ppp determines the order of the distance (e.g., p=1p=1p=1 for Manhattan, p=2p=2p=2 for Euclidean).

### 2. **What distance metrics can be used in KNN?**

In K-Nearest Neighbors, different distance metrics are used depending on the data type and problem requirements:

* **Euclidean Distance:** The default metric, it calculates the straight-line distance between two points and is commonly used for continuous numerical data.
* **Manhattan Distance:** Calculates the sum of the absolute differences, ideal for cases where data points are on a grid-like structure or have high-dimensional data.
* **Minkowski Distance:** A generalized form that can adapt to other distances using a power parameter ppp, with Euclidean (when p=2p=2p=2) and Manhattan (when p=1p=1p=1) as special cases.
* **Hamming Distance:** Counts the number of differing attributes, often used for categorical or binary data.
* **Cosine Similarity:** Although not typically available as a direct option in KNN, it is sometimes used by normalizing data vectors; it measures the cosine of the angle between two vectors, making it useful for text data or cases where direction, not magnitude, is important.

Each metric has strengths for different data types, and choosing the right one can enhance KNN performance for specific tasks.