Interview Questions:

1. What are the key hyperparameters in KNN?

ANS –

In K-Nearest Neighbors (KNN), there are several key hyperparameters that influence the performance of the model. Below are the main ones:

1. n\_neighbors (k)

Description: This is the number of neighbors to consider when making a prediction for a new data point.

Effect:

A small value (e.g., k = 1) can lead to an overfitting model, where the model is too sensitive to noise in the data.

A large value (e.g., k = 100) might lead to underfitting, where the model is too general and doesn't capture local patterns well.

Typical Range: 1 to the total number of training samples.

2. weights

Description: This determines the weight of each neighbor in the decision-making process.

'uniform': All neighbors are weighted equally.

'distance': Neighbors are weighted by their distance, meaning closer neighbors have a higher influence.

Custom weighting can also be specified using a callable function.

Effect:

'uniform' is generally simpler and can work well in many cases.

'distance' can improve performance when the distances between samples are more indicative of their class.

3. algorithm

Description: This defines the algorithm used to compute the nearest neighbors.

'auto': The algorithm will choose the best approach based on the input data.

'ball\_tree': Uses a Ball Tree data structure to find neighbors.

'kd\_tree': Uses a KD Tree data structure.

'brute': Uses a brute-force search to find nearest neighbors.

Effect:

For smaller datasets, brute-force search might be fine, but for larger datasets, ball\_tree or kd\_tree can speed up the nearest neighbor search.

4. metric

Description: The distance metric used to calculate the proximity between data points.

'euclidean': The most common distance measure, used for continuous features.

'manhattan', 'chebyshev', 'minkowski', etc., are other options.

Effect:

Different distance metrics can have a significant impact on performance depending on the data.

Euclidean is most commonly used for numeric data.

5. p

Description: The power parameter for the Minkowski distance.

p = 1 corresponds to the Manhattan distance (L1 norm).

p = 2 corresponds to the Euclidean distance (L2 norm).

Effect:

The choice of p affects the distance calculation and can influence classification accuracy.

6. leaf\_size

Description: The leaf size parameter controls the size of the leaf in the underlying tree structure (when using ball\_tree or kd\_tree).

Effect:

Smaller leaf sizes may improve accuracy but can slow down the model.

Larger leaf sizes speed up the model but can reduce accuracy.

7. n\_jobs

Description: The number of CPU cores to use during the computation of nearest neighbors.

n\_jobs = -1 will use all available cores.

Positive integers indicate the number of cores to use.

Effect:

Useful for parallelizing the model and speeding up computation, especially on large datasets.

8. metric\_params

Description: Additional parameters for the distance metric (used when metric is not one of the predefined options like 'euclidean', 'manhattan', etc.).

Effect:

These parameters can modify the behavior of the distance metric to suit specialized use cases.

9. leaf\_size

Description: The size of the leaf for the BallTree or KDTree algorithms.

Effect:

Affects performance in high-dimensional data (trade-off between speed and memory usage).

Summary:

n\_neighbors: Number of neighbors to consider.

weights: How to weight neighbors.

algorithm: Which algorithm to use for nearest neighbor search.

metric: Distance metric to use (e.g., Euclidean, Manhattan).

p: Affects the Minkowski distance (default is Euclidean).

leaf\_size: Affects the speed of tree-based algorithms.

n\_jobs: Number of CPU cores to use.

metric\_params: Additional parameters for custom metrics.

2. What distance metrics can be used in KNN?

ANS:-

In K-Nearest Neighbors (KNN), the distance metric is used to calculate the similarity or dissimilarity between points (i.e., how far two points are from each other). The choice of distance metric can significantly affect the model's performance, as it determines how the "neighbors" are identified.

Common Distance Metrics in KNN:

1. Euclidean Distance (L2 norm)

o Formula: d(x,y)=∑i=1n(xi−yi)2d(x, y) = \sqrt{\sum\_{i=1}^{n}(x\_i - y\_i)^2}d(x,y)=i=1∑n(xi−yi)2 Where xxx and yyy are two points in an nnn-dimensional space, and xi,yix\_i, y\_ixi,yi are the coordinates of the points in each dimension.

o Description: The most commonly used distance metric in KNN. It measures the straight-line distance between two points in Euclidean space.

o Use case: Suitable for continuous features and typically works well when the data points are geometrically distributed.

2. Manhattan Distance (L1 norm)

o Formula: d(x,y)=∑i=1n∣xi−yi∣d(x, y) = \sum\_{i=1}^{n} |x\_i - y\_i|d(x,y)=i=1∑n∣xi−yi∣

o Description: Also known as city block distance or taxicab distance, it sums the absolute differences of the coordinates. This is equivalent to the path a taxi would take on a grid.

o Use case: Useful when features are not continuous or when the data has outliers. It is often preferred when data has a grid-like structure, such as in urban planning or games with grid maps.

3. Minkowski Distance

o Formula: d(x,y)=(∑i=1n∣xi−yi∣p)1/pd(x, y) = \left(\sum\_{i=1}^{n} |x\_i - y\_i|^p\right)^{1/p}d(x,y)=(i=1∑n∣xi−yi∣p)1/p Where ppp is a parameter that controls the type of distance metric:

 If p=1p = 1p=1, Minkowski distance is equivalent to Manhattan distance.

 If p=2p = 2p=2, it is equivalent to Euclidean distance.

o Description: A generalized version of both Euclidean and Manhattan distances. It allows you to control the distance behavior by adjusting ppp.

o Use case: Useful when you want flexibility between the Manhattan and Euclidean metrics. The choice of ppp depends on the nature of the data.

4. Chebyshev Distance

o Formula: d(x,y)=max⁡i∣xi−yi∣d(x, y) = \max\_i |x\_i - y\_i|d(x,y)=imax∣xi−yi∣

o Description: Also known as the L∞ norm. This distance metric considers the largest difference along any coordinate axis between two points.

o Use case: Useful in situations where the data is highly asymmetric, and you want to emphasize the largest difference between the coordinates.

5. Cosine Similarity (Angle-based distance)

o Formula: Cosine Similarity(x,y)=x⋅y∣∣x∣∣∣∣y∣∣\text{Cosine Similarity}(x, y) = \frac{x \cdot y}{||x|| ||y||}Cosine Similarity(x,y)=∣∣x∣∣∣∣y∣∣x⋅y Where x⋅yx \cdot yx⋅y is the dot product of the vectors, and ∣∣x∣∣||x||∣∣x∣∣ and ∣∣y∣∣||y||∣∣y∣∣ are the Euclidean norms (magnitudes) of the vectors.

o Description: Measures the cosine of the angle between two vectors in an nnn-dimensional space. It ranges from -1 (opposite directions) to 1 (same direction).

o Use case: Typically used in text classification and natural language processing (NLP) when dealing with document-term matrices, where the magnitude of the vector isn't as important as the direction.

6. Hamming Distance

o Formula: d(x,y)=∑i=1n1(xi≠yi)d(x, y) = \sum\_{i=1}^{n} \mathbf{1}\_{(x\_i \neq y\_i)}d(x,y)=i=1∑n1(xi=yi) Where 1(xi≠yi)\mathbf{1}\_{(x\_i \neq y\_i)}1(xi=yi) is an indicator function that is 1 when the elements differ and 0 otherwise.

o Description: Measures the number of positions at which the corresponding elements in two strings or vectors are different.

o Use case: Useful for categorical data (e.g., binary or categorical features) or comparing strings.

7. Mahalanobis Distance

o Formula: d(x,y)=(x−y)TS−1(x−y)d(x, y) = \sqrt{(x - y)^{T} S^{-1} (x - y)}d(x,y)=(x−y)TS−1(x−y) Where SSS is the covariance matrix of the dataset.

o Description: Measures the distance between a point and a distribution. Unlike other distance metrics, it accounts for correlations between variables and scales the distance based on the covariance matrix.

o Use case: Useful when the data has correlated features and when you want a distance metric that takes the data distribution into account. It is often used in multivariate statistical analysis.

Custom Distance Metrics

• Description: In KNN, you can also define your own custom distance metric by passing a callable function to the metric parameter. This function should take two input arrays (representing two points) and return a distance value.

Summary of Common Metrics:

Distance Metric Formula Use Case

Euclidean ∑(xi−yi)2\sqrt{\sum (x\_i - y\_i)^2}∑(xi−yi)2

Most common, for continuous numerical data.

Manhattan (\sum x\_i - y\_i

Minkowski (\left(\sum x\_i - y\_i

Chebyshev (\max x\_i - y\_i

Cosine Similarity (\frac{x \cdot y}{

Hamming ∑1(xi≠yi)\sum \mathbf{1}\_{(x\_i \neq y\_i)}∑1(xi=yi)

For categorical or binary data.

Mahalanobis (x−y)TS−1(x−y)\sqrt{(x - y)^{T} S^{-1} (x - y)}(x−y)TS−1(x−y)

For correlated features and multivariate data.

Conclusion:

The choice of distance metric in KNN depends on the nature of the data. Euclidean and Manhattan distances are most commonly used for continuous data, while metrics like Hamming or Cosine similarity are more suitable for categorical or text data. Mahalanobis distance is ideal for cases where feature correlations are important.