**Interview Questions**

**1.What are some common hyperparameters of decision tree models, and how do they affect the model's performance?**

 **Max Depth**:

* **Description**: Limits the maximum depth of the tree.
* **Effect**: A deeper tree can capture more complex patterns but may lead to overfitting. A shallower tree may underfit the data.

 **Min Samples Split**:

* **Description**: The minimum number of samples required to split an internal node.
* **Effect**: A higher value can prevent the model from learning overly specific patterns (reducing overfitting), while a lower value allows for more splits and potentially better learning of the training data.

 **Min Samples Leaf**:

* **Description**: The minimum number of samples required to be at a leaf node.
* **Effect**: Increasing this value can smooth the model by ensuring that leaf nodes have more samples, helping to generalize better.

 **Max Features**:

* **Description**: The maximum number of features to consider when looking for the best split.
* **Effect**: Limiting features can help reduce overfitting and improve training time. Using fewer features can lead to a more generalized model.

 **Criterion**:

* **Description**: The function to measure the quality of a split (e.g., "gini" for Gini impurity or "entropy" for information gain).
* **Effect**: Different criteria can lead to different splits and hence influence the structure of the tree and its performance.

 **Max Leaf Nodes**:

* **Description**: Limits the number of leaf nodes in the tree.
* **Effect**: Reducing the number of leaf nodes can help in controlling the complexity of the model and mitigate overfitting.

 **Class Weight**:

* **Description**: Weights associated with classes for handling imbalanced datasets.
* **Effect**: Adjusting class weights can improve the model's performance on underrepresented classes, helping to create a more balanced decision-making process.
  1. **What is the difference between the Label encoding and One-hot encoding?**

In an interview, you can explain the differences between label encoding and one-hot encoding clearly and concisely like this:

**Label Encoding vs. One-Hot Encoding**

1. **Definition**:
   * **Label Encoding**: Converts categorical variables into numerical values by assigning a unique integer to each category. For example, categories like "red," "green," and "blue" might be encoded as 0, 1, and 2, respectively.
   * **One-Hot Encoding**: Creates binary columns for each category. For the same colors, you would create three columns: one for "red," one for "green," and one for "blue." Each column has a value of 1 or 0, indicating the presence of that category.
2. **Usage**:
   * **Label Encoding**: Typically used for ordinal categorical variables where the order matters (e.g., "low," "medium," "high"). The numerical representation maintains the rank.
   * **One-Hot Encoding**: Best suited for nominal categorical variables where there is no inherent order (e.g., colors, city names). It avoids introducing ordinal relationships that don’t exist.
3. **Impact on Models**:
   * **Label Encoding**: Can lead to misleading results in models that interpret the numerical values as ordered (like linear regression), potentially inferring a false relationship.
   * **One-Hot Encoding**: Prevents this issue by ensuring each category is treated independently, although it can increase the dimensionality of the dataset, which might impact performance with large categorical variables.
4. **Dimensionality**:
   * **Label Encoding**: Does not increase the dimensionality of the dataset; the number of features remains the same.
   * **One-Hot Encoding**: Increases dimensionality, especially with high cardinality categories, which can lead to sparse data.