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

Common hyperparameters of decision tree models and their effects on performance include:

**Maximum Depth (max\_depth):** This hyperparameter controls the maximum depth of the decision tree. A deeper tree can capture more complex relationships in the data, but it also increases the risk of overfitting, especially if the dataset is small.

**Minimum Samples Split (min\_samples\_split):** This hyperparameter sets the minimum number of samples required to split an internal node. It helps control the tree's growth by preventing the creation of nodes with too few samples, which can lead to overfitting.

**Minimum Samples Leaf (min\_samples\_leaf): This** hyperparameter sets the minimum number of samples required to be at a leaf node. It prevents the tree from creating leaf nodes with very few samples, which can also lead to overfitting.

**Maximum Features (max\_features):** This hyperparameter controls the number of features to consider when looking for the best split. Limiting the number of features can help prevent overfitting and speed up the training process, especially for datasets with a large number of features.

**Criterion:** Decision trees can use different criteria to measure the quality of a split, such as "gini" for Gini impurity or "entropy" for information gain. The choice of criterion can affect the tree's performance and the resulting splits.

**Tree Pruning:** Some decision tree algorithms support pruning techniques to reduce the size of the tree after it has been built. Pruning helps prevent overfitting by removing unnecessary branches that do not contribute significantly to improving the model's performance on unseen data.

Adjusting these hyperparameters can have a significant impact on the decision tree's performance, including its ability to generalize to unseen data, its computational efficiency, and its resistance to overfitting.

1. **What is the difference between the Label encoding and One-hot encoding?**

**Label Encoding:** In label encoding, each category or label of a categorical variable is assigned a unique integer. For example, if a variable has categories "red", "green", and "blue", they might be encoded as 0, 1, and 2, respectively. Label encoding preserves the ordinal relationship between categories, if any, but it may introduce unintended ordinality where none exists. It's typically used for ordinal categorical variables.

**One-Hot Encoding:** In one-hot encoding, each category of a categorical variable is represented as a binary vector (array) where only one element is 1 (indicating the presence of that category) and the rest are 0. For example, the categories "red", "green", and "blue" would be represented as [1, 0, 0], [0, 1, 0], and [0, 0, 1], respectively. One-hot encoding does not impose any ordinal relationship between categories and ensures that each category is treated as distinct. It's commonly used for nominal categorical variables.