1. What is a Decision Tree, and how does it work?

Ans. A decision tree is a hierarchical, graphical tool used to represent decision-making processes. It begins with a root node, posing the initial question, and branches into internal nodes, each representing further decision points based on data attributes. These branches ultimately lead to leaf nodes, which signify final outcomes or predictions.

The tree functions by iteratively splitting data based on yes/no questions. Starting at the root node, each question divides the dataset into subsets, directing the flow through the tree. If a condition is met, one branch is followed; otherwise, another is chosen. This process continues through a sequence of internal nodes, refining the data until reaching the leaf nodes, which represent the ultimate conclusions. This step-by-step approach simplifies complex decisions by breaking them into manageable, sequential choices.

2. What are impurity measures in Decision Trees?

Ans. In decision tree construction, impurity measures serve as critical metrics for evaluating the homogeneity of data subsets. These measures quantify the degree of disorder or randomness within a node, guiding the algorithm in selecting optimal split points. The objective is to minimize impurity, thereby creating subsets that are as pure as possible, containing predominantly a single class.

Common impurity measures include Gini impurity and entropy. Gini impurity assesses the probability of misclassifying a randomly chosen element, with lower values indicating greater homogeneity. Entropy, on the other hand, measures the level of disorder or uncertainty, where lower entropy signifies reduced uncertainty and increased purity.

The decision tree algorithm employs these measures to calculate the impurity of potential splits, selecting the split that yields the greatest reduction in impurity, or information gain. This iterative process refines the data, progressively building the tree structure until reaching leaf nodes that represent relatively pure subsets. Thus, impurity measures are fundamental in optimizing the decision tree's predictive accuracy by ensuring effective data partitioning.

3. What is the mathematical formula for Gini Impurity?

Ans. The mathematical formula for Gini Impurity is as follows:

$\text{Gini}(D) = 1 - \Sigma (p_i)^2$

Where:

- D represents the dataset.
- p_i is the probability of an element belonging to class i.
- Σ represents the sum over all classes.

4. What is the mathematical formula for Entropy?

Ans. The mathematical formula for Entropy:

$\text{Entropy}(S) = - \Sigma\, p_i \log_2(p_i)$

Where:

- S represents the dataset.
- p_i is the proportion of data points belonging to class i.
- Σ represents the sum over all classes.
- $\log_2$ represents the base 2 logarithm.

5. What is Information Gain, and how is it used in Decision Trees?
   Ans. Information Gain measures how much a feature reduces uncertainty (entropy) in a dataset. Decision trees use it for feature selection, choosing the attribute that maximizes this reduction. It's calculated by subtracting the weighted entropy of child nodes from the parent node's entropy. Higher Information Gain indicates a more effective split. By repeatedly selecting features with the highest gain, the algorithm builds a tree that creates increasingly pure subsets, enhancing predictive accuracy. Essentially, it guides the tree in making the most informative splits, optimizing the model's efficiency and performance.

6. What is the difference between Gini Impurity and Entropy?

Ans. Gini impurity and entropy both measure data impurity in decision trees, guiding split decisions. However, they differ in calculation:

Gini impurity:

- Calculates the probability of misclassifying a random element.
- It is computationally faster.

Entropy:

- Measures the degree of disorder or uncertainty.
- It involves logarithmic calculations, making it more computationally intensive.
- Although they are different, they generally give similar results.

7. What is the mathematical explanation behind Decision Trees?

Ans. Decision trees mathematically partition data using recursive binary splits. At each node, a feature is selected to maximize information gain (entropy reduction) or minimize Gini impurity. This involves calculating impurity measures for potential splits and choosing the one with the optimal result. The process repeats, creating branches based on feature values. Leaf nodes represent final predictions. This optimization is based on minimizing the weighted average impurity of child nodes, compared to the parent node. Mathematically, it's about finding the feature and threshold that best separate classes, building a hierarchical structure for prediction.

8. What is Pre-Pruning in Decision Trees?

Ans. Pre-pruning in decision trees halts tree growth early to prevent overfitting. It sets limits on tree depth, minimum samples per leaf, or minimum information gain for a split. If a split violates these criteria, the node becomes a leaf, preventing further branching. This stops the tree from perfectly fitting noisy data, improving generalization to unseen data. It simplifies the model, reducing complexity and potential overfitting.

9. What is Post-Pruning in Decision Trees?

Ans. Post-pruning in decision trees involves growing the tree to its full potential and then selectively removing branches to improve generalization. It starts with a complex, potentially overfitted tree. Then, it evaluates the impact of removing subtrees on a validation dataset. If removing a subtree improves performance (e.g., reduces error), the subtree is pruned, and the node becomes a leaf. This process continues until no further pruning enhances performance. Post-pruning helps to simplify the tree, reduce overfitting, and enhance its ability to generalize to unseen data.

10. What is the difference between Pre-Pruning and Post-Pruning?

Ans. Pre-pruning and post-pruning are both techniques to prevent overfitting in decision trees, but they differ in their approach:

Pre-pruning:

- Stops tree growth during construction.
- Sets limits (e.g., max depth, min samples) to prevent further splits.
- Simpler and faster, but may underfit if limits are too strict.

Post-pruning:

- Grows the tree fully, then removes branches after construction.
- Evaluates subtrees on a validation set and prunes those that don't improve performance.
- More computationally intensive, but often yields better results by allowing the tree to capture more complex patterns before simplification.
- Post pruning is generally considered to be more effective.

11. What is a Decision Tree Regressor?

Ans. A Decision Tree Regressor is a supervised learning algorithm that predicts continuous values. Unlike classification trees, it uses a tree-like structure to model relationships between features and a target variable. At each node, it splits the data based on feature values to minimize variance or mean squared error. Leaf nodes contain the average or predicted value for the region. This approach allows for non-linear relationships and is useful for regression tasks where the target variable is continuous.

12. What are the advantages and disadvantages of Decision Trees?

Ans. Advantages of Decision Trees:

- Interpretability: Easy to understand and visualize, making them transparent.
- Handles both categorical and numerical data: Versatile and adaptable to various data types.
- Non-parametric: No assumptions about data distribution.

- Feature importance: Provides insights into which features are most significant.
- Relatively fast: Can be quick to train and predict.

Disadvantages of Decision Trees:

- Overfitting: Prone to creating overly complex trees that fit noise.
- Instability: Small changes in data can lead to significant changes in the tree.
- Bias towards features with more levels: Can favor features with many categories.
- Doesn't perform well with complex relationships: Can struggle to capture smooth, continuous relationships as well as other models.

13. How does a Decision Tree handle missing values?

Ans. Decision trees handle missing values in a few ways. During training, some algorithms can learn split rules that account for missing values, directing data to appropriate branches. Alternatively, missing values might be imputed with the most frequent value or the mean/median before training. Some implementations create surrogate splits, using other features to mimic the split where the missing value occurs. Ultimately, the approach depends on the specific algorithm's implementation.

14. How does a Decision Tree handle categorical features?

Ans. Decision trees handle categorical features by splitting the data based on following categories. Such as:

- Binary Splits:
  - For binary categorical features (e.g., "yes/no"), the tree creates a simple binary split.
- Multi-way Splits:
  - For multi-category features (e.g., "red," "blue," "green"), the tree can create multiple branches, one for each category.
  - Alternatively, it can create binary splits by grouping categories.
- Impurity Measures:
  - The algorithm uses impurity measures (Gini impurity or entropy) to determine the best categorical split, aiming to create subsets with the purest class distribution.
- Encoding:
  - Some implementations will encode categorical data into numerical data, before processing it.

15. What are some real-world applications of Decision Trees?

Ans. Decision trees are used across various fields due to their interpretability and versatility. Some real-world applications are stated below:

- Medical Diagnosis:
  Predicting patient risk for diseases based on symptoms and medical history.
- Financial Risk Assessment:
  Evaluating creditworthiness and detecting fraudulent transactions.

- Customer Relationship Management (CRM):
  Predicting customer churn and targeting marketing campaigns.
- Manufacturing Quality Control:
  Identifying defects in products based on production parameters.
- Environmental Science:
  Classifying land cover types and predicting natural disasters.
- E-commerce:
  Recommending products to customers based on their purchase history.
- Business decision making:
  Analysing various factors to decide on strategies to implement.