Decision Trees & Random Forests – Interview Questions
1. How does a Decision Tree work?
A Decision Tree splits data into branches based on feature values.

At each node, it asks a yes/no or condition question.

It keeps splitting until it reaches leaves that represent the final decision (class/label).

Example:

pgsql
Copy
Edit
Is Age > 50?
 Yes Check Cholesterol
 No Check Blood Pressure
It chooses splits that best separate the classes at each step.

2. What is Entropy and Information Gain?
Entropy measures disorder or uncertainty in data.

If all samples are from the same class entropy = 0 (pure)
If classes are mixed entropy is higher (impure).
Information Gain is the reduction in entropy after a split.
A good feature split gives high information gain.
Decision Trees choose the feature with the highest information gain to split the data.

3. How is Random Forest better than a single tree?
A Random Forest is a collection of many decision trees trained on different subsets of the data.
It combines the outputs of all trees (via majority vote or average).

Advantages:
Less overfitting than a single tree

More accurate predictions
Handles noisy data better

4. What is overfitting and how do you prevent it?
Overfitting happens when a model learns both useful patterns and noise in training data.
It performs very well on training data but poorly on new, unseen data.
To prevent overfitting:

Limit the depth of the tree (max_depth)
Set a minimum number of samples per leaf (min_samples_leaf)
Prune unnecessary branches
Use ensemble methods like Random Forest
Apply cross-validation to test generalization

5. What is bagging?
Bagging stands for "Bootstrap Aggregating".

It means training multiple models on different random subsets of the data (with replacement).
The outputs are combined (voting or averaging).
Bagging reduces variance and improves generalization.

6. How do you visualize a decision tree?
You can use plot_tree from Scikit-learn:

python
Copy
Edit
from sklearn.tree import plot_tree
plot_tree(model, feature_names=X.columns, filled=True)
Or export the tree to a .dot file and visualize it using Graphviz.

7. How do you interpret feature importance?
Feature importance tells us how valuable each feature is for making predictions.

It is based on how much each feature reduces impurity (entropy) across all splits.
A feature with high importance is used more often and contributes more to predictions.
In Random Forest, use:
python
Copy
Edit
model.feature_importances_
Visualize it with a bar chart to understand which features matter most.

8. What are the pros and cons of Random Forests?
Pros:

High accuracy
Works for both classification and regression
Handles missing values and noise well
Reduces overfitting through averaging
Cons:

Slower to train than a single tree
Harder to interpret
Can consume more memory