Task 5 - Decision Tree & Random Forest Q&A

1. How does a decision tree work?

A decision tree splits the data into branches based on feature values. It asks a question at each node (e.g., 'Is age > 50?') and continues branching until it reaches a final prediction at a leaf node.

2. What is entropy and information gain?

Entropy measures disorder in data. Information gain is the reduction in entropy after a split. A good split has high information gain because it makes the data more organized.

3. How is random forest better than a single tree?

Random Forest builds many decision trees using random samples and averages their results. It is more accurate, stable, and less likely to overfit compared to a single decision tree.

4. What is overfitting and how do you prevent it?

Overfitting happens when a model memorizes training data. You can prevent it by limiting tree depth, setting minimum samples per split, using cross-validation, or using random forests.

5. What is bagging?

Bagging stands for Bootstrap Aggregating. It creates multiple datasets by sampling with replacement, trains a model on each, and averages the results. Random Forest uses bagging.

6. How do you visualize a decision tree?

Use the plot_tree() function in sklearn to display splits, conditions, and leaf nodes. It shows how decisions are made step-by-step.

7. How do you interpret feature importance?

Feature importance shows how much each feature helps in prediction. In random forests, it is based on how much each feature reduces impurity across all trees.

8. What are the pros/cons of random forests?

Pros: High accuracy, less overfitting, handles noise and missing data.

Cons: Slower than single trees, harder to interpret, more memory usage.