Interview Questions:

Q1.How does a decision tree work?
Ans.A decision tree works by splitting data into branches based on decision rules derived from input features. At each node, the algorithm selects the feature that best separates the data using criteria like Gini impurity or information gain. This process continues recursively, forming a tree structure where each internal node represents a decision based on a feature, and each leaf node represents an outcome or class label. The tree enables clear, interpretable paths from input features to predictions or classifications.

Q 2. What is entropy and information gain?
Ans. Entropy is a measure of uncertainty or randomness in a dataset. In decision trees, it quantifies how mixed the class labels are within a node—higher entropy means more disorder. Information gain measures the reduction in entropy achieved by splitting the data based on a particular feature. It helps the decision tree select the feature that best separates the data; the feature with the highest information gain is chosen for the split, as it provides the most clarity about the target variable.

Q 3. How is random forest better than a single tree?
Ans. Random Forest is better than a single decision tree because it combines the predictions of multiple trees to improve accuracy and reduce overfitting. While a single tree may capture noise in the data and perform poorly on unseen data, Random Forest builds many trees on different subsets of the data and averages their results, leading to more robust and reliable predictions. This ensemble approach increases stability and generalization, making Random Forest a more powerful and accurate model overall.

Q4. What is overfitting and how do you prevent it?
Ans. Overfitting occurs when a model learns not only the underlying patterns in the training data but also the noise, causing it to perform well on training data but poorly on new, unseen data. It essentially means the model is too complex and too tailored to the training set. To prevent overfitting, techniques such as cross-validation, simplifying the model, pruning decision trees, regularization (like L1 or L2), and using more training data can be applied. These methods help the model generalize better to new data.

Q 5. What is bagging?
Ans. Bagging, short for Bootstrap Aggregating, is an ensemble learning technique that improves the accuracy and stability of machine learning models by combining multiple models trained on different random subsets of the training data. Each subset is created using bootstrapping (sampling with replacement), and a model (typically a decision tree) is trained on each subset. The final prediction is made by averaging the outputs (for regression) or taking a majority vote (for classification). Bagging helps reduce variance and prevents overfitting, especially with high-variance models like decision trees.

Q 6. How do you visualize a decision tree?
Ans. To visualize a decision tree, you typically use a tree diagram that shows the hierarchy of decisions based on input features. Each internal node represents a decision or test on a feature, each branch represents the outcome of that test, and each leaf node represents a final prediction or outcome. Tools like scikit-learn's plot_tree(), Graphviz, or visualization libraries in platforms like Tableau or Excel can be used to create clear, interpretable tree diagrams. These visuals help understand the model's logic and the paths leading to specific predictions.

Q7. How do you interpret feature importance?
Ans. Feature importance measures how much each input variable contributes to a model's predictions. Higher importance means the feature has a stronger influence on the target outcome. By

interpreting feature importance, you can identify which variables drive the model's decisions, helping prioritize key factors, improve model transparency, and guide feature selection or engineering for better performance.

Q8. What are the pros/cons of random forests?
Ans. Random forests offer strong predictive accuracy and handle large datasets with many features well by averaging multiple decision trees to reduce overfitting. They are robust to noise and can model complex relationships without requiring much parameter tuning. However, they can be slower to train and less interpretable compared to single decision trees, and they may struggle with very high-dimensional sparse data or extrapolation beyond the training range.