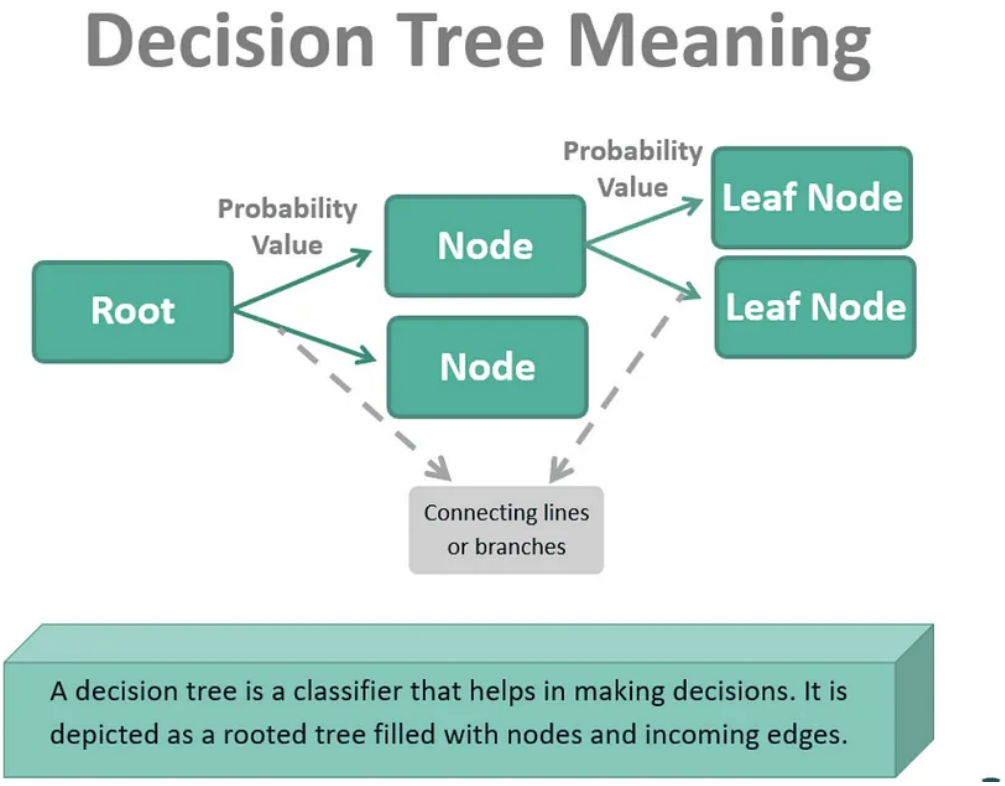
Interview Questions for Decision Tree



Q: What is a decision tree?
A: A decision tree is a machine learning algorithm used for both classification and regression tasks. It is a tree-like model in which each internal node represents a test on a feature, each branch represents the outcome of the test, and each leaf node represents a class label or a numerical value.

Q: What are some advantages of using a decision tree?
A: Decision trees are easy to understand and interpret, they can handle both categorical and numerical data, they can handle missing values and outliers, and they can be used for both classification and regression tasks.

Q: What is entropy in the context of decision trees?
A: Entropy is a measure of the impurity or randomness of a set of examples. In the context of decision trees, entropy is used to measure the impurity of a

set of training examples with respect to their class labels. The goal of a decision tree algorithm is to minimize the entropy at each level of the tree, which corresponds to maximizing the information gain.

Q: What is information gain in the context of decision trees?
A: Information gain is a measure of the reduction in entropy achieved by partitioning a set of examples based on a certain attribute. The information gain of an attribute is calculated as the difference between the entropy of the original set of examples and the weighted average of the entropies of the partitions created by the attribute.

Q: What is pruning in the context of decision trees?
A: Pruning is a technique used to prevent overfitting in decision trees. It involves removing branches from the tree that do not contribute significantly to its accuracy. There are two main types of pruning: pre-pruning, which involves stopping the tree from growing when certain conditions are met, and post-pruning, which involves removing branches from the fully grown tree.

Q: What are some common criteria used for splitting nodes in a decision tree?
A: Some common criteria used for splitting nodes in a decision tree include entropy, information gain, Gini impurity, and chi-squared test. The choice of criterion depends on the specific task and the nature of the data.

Q: How do you handle missing values in a decision tree?
A: There are several ways to handle missing values in a decision tree, including ignoring the examples with missing values, assigning a default value to missing values, imputing missing values with the mean or median of the corresponding feature, or using a separate branch to handle missing values.

Q: Can decision trees handle non-linear relationships between features and the target variable?
A: No, decision trees can only model linear relationships between features

and the target variable. However, by combining multiple decision trees in an ensemble method such as random forests or gradient boosting, non-linear relationships can be approximated.

Q: What is gain ratio in the context of decision tree splitting?
A: Gain ratio is a modification of information gain that takes into account the intrinsic information of the feature, which measures how much information is gained by knowing the feature itself. The gain ratio of a feature is calculated as the information gain of the feature divided by its intrinsic information.

Q: What is Gini impurity in the context of decision tree splitting?
A: Gini impurity is a measure of the impurity or randomness of a set of examples. In the context of decision tree splitting, Gini impurity is used to measure the impurity of a set of training examples with respect to their class labels. The goal of a decision tree algorithm is to minimize the Gini impurity at each level of the tree.

Q: What is chi-squared test in the context of decision tree splitting?
A: Chi-squared test is a statistical test used to measure the dependence between two variables. In the context of decision tree splitting, chi-squared test is used to measure the dependence between a feature and the target variable. The goal of the test is to select the feature that has the highest association with the target variable.

Q: How do you handle continuous features in decision tree splitting?
A: There are several ways to handle continuous features in decision tree splitting, including discretizing the features into categorical bins, using regression trees instead of classification trees, or using algorithms that can handle continuous features directly, such as CART or C4.5.

Q: What is pruning in the context of decision tree splitting?
A: Pruning is a technique used to prevent overfitting in decision trees by removing branches that do not contribute significantly to its accuracy. Pruning can be done either before or after the tree is fully grown. Pre-

pruning involves stopping the tree from growing when certain conditions are met, while post-pruning involves removing branches from the fully grown tree.

Q: What are the parameters of Decison Tree in scikit-learn?
A: Here are some commonly used parameters for decision trees in scikit-learn:

For Classification:

- *criterion*: This parameter specifies the function used to measure the quality of a split. The default value is “gini”, but “entropy” can also be used.
- *max_depth*: This parameter specifies the maximum depth of the decision tree. Setting a smaller value for max_depth can help prevent overfitting.
- *min_samples_split*: This parameter specifies the minimum number of samples required to split an internal node. Setting a larger value for min_samples_split can help prevent overfitting.
- *min_samples_leaf*: This parameter specifies the minimum number of samples required to be at a leaf node. Setting a larger value for min_samples_leaf can help prevent overfitting.
- *max_features*: This parameter specifies the maximum number of features that are considered when looking for the best split. Setting a smaller value for max_features can help prevent overfitting.

For Regression:

- *criterion*: This parameter specifies the function used to measure the quality of a split. The default value is “mse” (mean squared error), but “mae” (mean absolute error) can also be used.
- *max_depth*: This parameter specifies the maximum depth of the decision tree. Setting a smaller value for max_depth can help prevent overfitting.

- *min_samples_split*: This parameter specifies the minimum number of samples required to split an internal node. Setting a larger value for min_samples_split can help prevent overfitting.
- *min_samples_leaf*: This parameter specifies the minimum number of samples required to be at a leaf node. Setting a larger value for min_samples_leaf can help prevent overfitting.
- *max_features*: This parameter specifies the maximum number of features that are considered when looking for the best split. Setting a smaller value for max_features can help prevent overfitting.

These are just a few examples of parameters that can be used in scikit-learn's decision tree models. Depending on the specific problem, there may be other parameters that are more important to adjust.

Q: What is the default value of parameters for Decision Tree ?
A: Here are the default parameter values for scikit-learn's DecisionTreeClassifier and DecisionTreeRegressor:

For DecisionTreeClassifier:

- criterion: "gini"
- splitter: "best"
- max_depth: None
- min_samples_split: 2
- min_samples_leaf: 1
- min_weight_fraction_leaf: 0
- max_features: None
- random_state: None
- max_leaf_nodes: None
- min_impurity_decrease: 0

- min_impurity_split: None
- class_weight: None
- ccp_alpha: 0.0

For DecisionTreeRegressor:

- criterion: “mse”
- splitter: “best”
- max_depth: None
- min_samples_split: 2
- min_samples_leaf: 1
- min_weight_fraction_leaf: 0
- max_features: None
- random_state: None
- max_leaf_nodes: None
- min_impurity_decrease: 0
- min_impurity_split: None
- ccp_alpha: 0.0

Note that some of these parameters, such as max_depth and max_features, have default values of None, which means they are not constrained by default. Other parameters, such as min_samples_split and min_samples_leaf, have default values of 2 and 1, respectively. These default values may not be optimal for every problem, so it’s important to tune the parameters to your specific dataset and problem.

Q: What are the evaluation metrics for Decision Tree ?
A: Here are some commonly used evaluation metrics for decision tree models:

For Classification:

- Accuracy: This measures the proportion of correct predictions out of all predictions made. It's the most commonly used metric for classification problems.
- Precision: This measures the proportion of true positives out of all positive predictions. It's useful when the cost of false positives is high.
- Recall: This measures the proportion of true positives out of all actual positives. It's useful when the cost of false negatives is high.
- F1 Score: This is the harmonic mean of precision and recall, and is a good overall measure of a model's performance.
- Confusion Matrix: This is a table that shows the number of true positives, false positives, true negatives, and false negatives, and can be used to calculate other evaluation metrics.

For Regression:

- Mean Squared Error (MSE): This measures the average of the squared differences between the predicted and actual values.
- Mean Absolute Error (MAE): This measures the average of the absolute differences between the predicted and actual values.
- R-squared (R2): This measures the proportion of variance in the target variable that can be explained by the model. It ranges from 0 to 1, with higher values indicating a better fit.

Note that the specific metric(s) you use to evaluate a decision tree model will depend on the problem you're trying to solve and the specific goals of your analysis. It's often a good idea to use a combination of metrics to get a more complete picture of the model's performance.

Follow :: https://medium.com/@thedatabeast