Decision Tree Algorithm

A Decision Tree is a supervised machine learning algorithm used for both classification and regression tasks. It splits the data into branches based on feature values to create a tree-like structure that helps make predictions.

1. Working Principle

Decision Trees use a top-down approach called recursive partitioning. At each node, the dataset is split based on a feature that results in the highest information gain or lowest impurity. This process continues until a stopping criterion (like maximum depth or minimum samples) is reached.

2. Key Components

Root Node: Represents the entire dataset, chosen based on the best feature to split.

Decision Nodes: Points where the dataset is split further based on feature conditions.

Leaf Nodes: Represent the final output or decision.

3. Splitting Criteria

The decision of where to split is based on impurity measures such as Entropy or Gini Index.

4. Entropy

Entropy measures the disorder or impurity in a dataset. It is calculated as:

Entropy = - Σ p(i) * log■(p(i))

Where p(i) is the probability of class i. A node is pure if entropy = 0 (i.e., all samples belong to one class).

5. Information Gain

Information Gain measures the reduction in entropy after a dataset split. It helps select the best feature.

Information Gain = Entropy(Parent) - Weighted Average * Entropy(Children)

6. Gini Index

Gini Index measures impurity or purity used in CART (Classification and Regression Trees).

Gini = 1 - Σ $p(i)^2$

Lower Gini value indicates higher purity.

7. Difference Between Entropy and Gini Index

Entropy: Uses logarithmic function; more computationally intensive. It is more sensitive to class imbalance.

Gini Index: Uses squared probabilities; computationally simpler and preferred in most implementations.

Practical Note: Both metrics usually give similar results; Gini tends to isolate the most frequent class faster.

8. Pruning

Pruning helps prevent overfitting by removing branches that add little predictive power. It can be pre-pruning (early stopping) or post-pruning (after tree construction).

9. Advantages

• Easy to interpret and visualize. • Handles both numerical and categorical data. • Requires little data preprocessing. • Captures nonlinear relationships.

10. Disadvantages

• Prone to overfitting if not pruned. • Small data changes can drastically change the tree (high variance). • Biased toward features with more levels.

11. Assumptions of Decision Tree

1. The training data accurately represents the population. 2. Instances are assumed to be independent of each other. 3. Attributes are assumed to have meaningful splits. 4. The relationships between variables are hierarchical. 5. Decision boundaries are axis-aligned (each decision is based on one feature at a time).

12. Decision Tree for Regression

For regression, Decision Trees use metrics like Mean Squared Error (MSE) or Mean Absolute Error (MAE) instead of Entropy or Gini Index to decide splits.

13. Summary Table: Entropy vs Gini Index

Entropy: Measures information gain using log function; slower but more precise. **Gini Index:** Measures impurity using squared probabilities; faster and commonly used in CART. **Similarity:** Both result in similar splits in most real-world cases.

14. Implementation Hint

In Python, Decision Trees can be implemented using scikit-learn:

from sklearn.tree import DecisionTreeClassifier model = DecisionTreeClassifier(criterion='gini', max_depth=4) model.fit(X_train, y_train) y_pred = model.predict(X_test)