Experiment 7

Objective:

To implement decision tree using CART algorithm.

Theory

The Classification and Regression Trees (CART) algorithm is a popular approach to building decision trees for both classification and regression tasks. Developed by Breiman et al., CART is highly versatile and forms the basis for more advanced tree-based algorithms like random forests and boosting methods. Unlike the ID3 algorithm, which is primarily used for classification, CART supports both classification and regression, making it widely applicable across a range of machine learning tasks.

Decision Tree Structure in CART:
A decision tree consists of internal nodes representing decisions based on feature values, branches representing the outcomes of those decisions, and leaf nodes where final predictions (either classes for classification or values for regression) are made. The CART algorithm builds binary trees, meaning each internal node splits the data into two subsets based on a chosen feature and threshold value.

The CART Algorithm Process:

Gini Impurity (Classification): For classification, CART uses the Gini impurity criterion to evaluate splits. Gini impurity measures the likelihood of incorrectly classifying a randomly chosen element if it were randomly labeled according to the distribution of labels in a subset. For a node m with classes c, the Gini impurity G is calculated as:

$$G(m) = 1 - \sum_{c=1}^{C} p_c^2$$

where p is the probability of selecting a data point from class c in node m. The lower the Gini impurity, the purer the node. CART chooses the feature and threshold that minimize the Gini impurity for each split.
Mean Squared Error (Regression): For regression tasks, CART uses mean squared error (MSE) to measure the quality of splits. It aims to minimize the variance of target values within each leaf node, ensuring that each region represents a homogeneous output. MSE at node
m is calculated as:

$$\text{MSE} = \frac{1}{N_m} \sum_{i=1}^{N_m} (y_i - \bar{y}_m)^2$$

Recursive Partitioning: The algorithm applies recursive binary splits to partition the dataset. At each step, CART evaluates all possible splits and selects the feature and threshold that minimize impurity (Gini for classification, MSE for regression).
Tree Pruning: Unlike ID3, CART includes a pruning step to prevent overfitting. Pruning removes branches that provide little predictive power, improving the model's generalization on new data.
Advantages and Limitations of CART: CART is highly interpretable, offering an intuitive way to model complex decision boundaries through simple rules. It performs well with both numeric and categorical data and handles multi-class classification naturally.

Result

As a result of this Experiment, we successfully wrote and executed the program to implement decision tree using CART algorithm.

Learning Outcomes

Understand and implement the CART algorithm for decision trees, using Gini impurity or MSE for splits and applying pruning to improve model interpretability and prevent overfitting.