Practical No.9

Title: Write a program to demonstrate the working of the decision tree based CART algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

Objective

The objective of this practical is to implement a **Decision Tree Classifier** using the **CART (Classification and Regression Trees) algorithm**. The model will be trained using a dataset, and the trained tree will be used to classify a new sample.

Introduction to Decision Trees and the CART Algorithm

A **Decision Tree** is a supervised learning algorithm used for **classification and regression tasks**. It consists of decision nodes, branches, and leaf nodes, where:

- **Decision nodes** represent feature-based conditions.
- **Branches** represent decision outcomes.
- **Leaf nodes** contain the predicted class labels.

CART Algorithm Overview

The **CART (Classification and Regression Trees) algorithm** builds a **binary decision tree** by **recursively splitting** the dataset into two subsets at each node. The splits are made to **minimize impurity**, which is measured using the **Gini index** for classification problems.

Key Concepts in the CART Algorithm

1. **Gini Index (GGG)**
    - Measures the impurity of a dataset.
    - Formula:

Gini(S)=1−∑pi2Gini(S) = 1 - \sum p_i^2Gini(S)=1−∑pi2

where pip_ipi is the proportion of class iii in the dataset.

    - A **lower Gini index** means purer subsets, leading to better classification.

2. **Recursive Binary Splitting**
    - The dataset is repeatedly split into two parts at each node.
    - The split is chosen to minimize the weighted sum of the Gini index of the resulting subsets.
3. **Stopping Criteria**
    - The algorithm stops when:
        - All instances in a subset belong to the same class.
        - A predefined depth limit is reached.
        - The number of instances in a subset falls below a threshold.

Difference Between ID3 and CART

Feature	ID3 Algorithm	CART Algorithm
Splitting Criterion	Information Gain	Gini Index
Tree Structure	Multi-way Splits	Binary Splits
Handles Numerical Data	No (requires discretization)	Yes

Dataset Description

An appropriate dataset is chosen for constructing the decision tree. The dataset consists of:

- **Independent variables (features)** used for classification.
- **A categorical dependent variable (class labels)**.
- The dataset can contain both **categorical and numerical features**, as CART can handle both types.

Implementation Steps

Step 1: Load and Explore the Dataset

- Load the dataset from a **CSV file**.
- Display dataset details, including feature names, class distribution, and missing values.

Step 2: Data Preprocessing

- Convert categorical values into numerical format if needed.
- Handle missing values using **imputation techniques**.
- Split the dataset into **training data (80%)** and **testing data (20%)**.

Step 3: Build the Decision Tree Using CART Algorithm

- Compute the **Gini index** for each feature.
- Perform **recursive binary splitting** on the dataset.
- Construct a binary decision tree based on the best splits.
- Define **stopping conditions** to prevent overfitting.

Step 4: Visualize the Decision Tree

- Display the tree structure in **text format** or **graphical format**.

Step 5: Classify a New Sample

- Provide a new sample with feature values.
- Traverse the decision tree from the root node to a leaf node based on decision rules.
- Predict the class label for the new sample.

Step 6: Compute Model Accuracy

The accuracy of the model is computed using:

Accuracy=Correct PredictionsTotal Predictions×100Accuracy = \frac{Correct\ Predictions}{Total\ Predictions} \times 100Accuracy=Total PredictionsCorrect Predictions×100

Additionally, a **confusion matrix** is generated to analyze the performance of the classifier.

Expected Output

- **Decision Tree Structure** (text or graphical format).
- **Predicted vs. Actual Labels** for test samples.
- **Classification result for a new sample**.
- **Accuracy score and confusion matrix** for model evaluation.

Conclusion

This practical demonstrates the implementation of the **CART Decision Tree Algorithm** for classification. The model was trained, tested, and applied to classify a new sample.