Classification and Regression Tree (CART) Implementation

ABUBAKAR

HCIA-AI TRAINER

Agenda

Datasets

CART Algorithm

Standard Deviation

Trees, Regression and Classifications

Data Science Libraries

CART Applications

Implemented CART Interpretation

Coding Marathon

Dataset and Subset

Feature	Dataset	Subset
Definition	The **entire collection** of data used for analysis/modeling.	A **smaller portion** of the dataset, selected based on conditions (e.g., filtering rows/columns).
Purpose	Serves as the **complete input** for training/testing models.	Used for **specific analyses** (e.g., splits in decision trees, train/test sets).
Size	Larger (contains all records).	Smaller (sampled or filtered from the dataset).
Example	A CSV file with 10,000 customer records.	Rows where Age > 30 or a 70% random sample for training.
In Decision Trees	Original data (df).	Temporary splits (subset1, subset2) during tree construction.

Types of Datasets

Dataset Type	Description	Common Uses
Structured Data	Tabular data with rows/columns (e.g., CSV, SQL tables).	Regression, classification, business analytics.
Unstructured Data	No predefined format (e.g., text, images, audio).	NLP, computer vision, speech recognition.
Time-Series Data	Data points indexed by time (e.g., stock prices, sensor logs).	Forecasting (ARIMA, LSTM), anomaly detection.
Geospatial Data	Data with geographic attributes (e.g., GPS coordinates, maps).	Route optimization, climate modeling, GIS applications.
Graph Data	Nodes and edges (e.g., social networks, recommendation systems).	Fraud detection, social network analysis (PageRank, GNNs).
Synthetic Data	Artificially generated data (e.g., GANs, simulations).	Privacy preservation, augmenting small datasets.
Labeled Data	Includes target/output values (e.g., "Spam" or "Not Spam").	Supervised learning (classification/regression).
Unlabeled Data	No target values (e.g., raw text, images).	Clustering (k-means), dimensionality reduction (PCA), self-supervised learning.
Imbalanced Data	Uneven class distribution (e.g., 95% "Normal" vs. 5% "Fraud").	Fraud detection, medical diagnosis (requires resampling/SMOTE).
Streaming Data	Continuously generated (e.g., live tweets, IoT sensor feeds).	Real-time analytics (Apache Kafka, Spark Streaming).

Common Dataset File Extensions

Extension	Format	Description	Common Uses
.csv	Comma-Separated Values	Plain text with data separated by commas (or other delimiters).	Tabular data analysis (Excel, Pandas).
.xlsx	Excel Workbook	Spreadsheet with multiple sheets, formulas, and formatting.	Business reports, financial data.
.json	JavaScript Object Notation	Lightweight key-value pair format, human-readable.	Web APIs, nested/hierarchical data.
.sqlite	SQLite Database	Lightweight relational database in a single file.	Mobile apps, local storage.
.xml	XML	Markup language for structured data.	Web data, document storage.
.db	Database File	Generic extension for databases (SQLite, Oracle, etc.).	Application data storage.
.txt	Plain Text	Unformatted text, often with line breaks or custom delimiters.	Log files, raw text data, simple datasets.

What is CART?

The CART algorithm is a foundational tool in machine learning due to its simplicity, interpretability, and versatility. By recursively splitting data based on optimized criteria (Gini impurity for classification, variance reduction for regression), CART builds decision trees that are effective for both classification and regression tasks. While it has limitations like overfitting and instability, techniques like pruning and ensemble methods mitigate these issues. Understanding CART provides a strong foundation for exploring more advanced tree-based models.

CART in Machine Learning

The term CART serves as a generic term for the following categories of decision trees:

Classification Trees: The tree is used to determine which "class" the target variable is most likely to fall into when it is continuous.

Regression trees: These are used to predict a continuous variable's value.

CART Algorithm (Cont.)

Classification and Regression Trees (CART) is a decision tree algorithm that is used for both classification and regression tasks. It is a supervised learning algorithm that learns from labelled data to predict unseen data.

Tree structure: CART builds a tree-like structure consisting of nodes and branches. The nodes represent different decision points, and the branches represent the possible outcomes of those decisions. The leaf nodes in the tree contain a predicted class label or value for the target variable.

CART Algorithm

Splitting criteria: CART uses a greedy approach to split the data at each node. It evaluates all possible splits and selects the one that best reduces the impurity of the resulting subsets. For classification tasks, CART uses Gini impurity as the splitting criterion. The lower the Gini impurity, the more-pure the subset is. For regression tasks, CART uses residual reduction as the splitting criterion. The lower the residual reduction, the better the fit of the model to the data.

Pruning: To prevent overfitting of the data, pruning is a technique used to remove the nodes that contribute little to the model accuracy. Cost complexity pruning and information gain pruning are two popular pruning techniques. Cost complexity pruning involves calculating the cost of each node and removing nodes that have a negative cost. Information gain pruning involves calculating the information gain of each node and removing nodes that have a low information gain.

How does CART algorithm work?

The CART algorithm works via the following process:

The best-split point of each input is obtained.

Based on the best-split points of each input in Step 1, the new “best” split point is identified.

Split the chosen input according to the “best” split point.

Continue splitting until a stopping rule is satisfied or no further desirable splitting is available.

Advantages of CART

Interpretability: Decision trees are easy to understand and visualize.

Handles Mixed Data: Works with both numerical and categorical features.

Non-parametric: No assumptions about data distribution.

Feature Importance: Provides insights into which features are most influential.

Versatile: Suitable for both classification and regression tasks.

Standard Deviation

Standard Deviation (σ) is a statistical measure that quantifies how spread out or dispersed a set of data points is around their mean (average).

Measures Variability

- A low standard deviation means data points are close to the mean.
- A high standard deviation means data points are widely scattered.

Formula:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

N = Number of data points

$\mathbf{x_i}$ = Each individual value

μ = Mean of all values

Why is it Used in Decision Trees (Regression)?

In **regression trees**, standard deviation helps decide splits by measuring how much variance exists in the target variable.

The algorithm tries to **reduce standard deviation** in child nodes compared to the parent node (similar to how Gini impurity works in classification).

A split is considered good if it **maximizes standard deviation reduction** (i.e., subsets become more homogeneous).

Trees

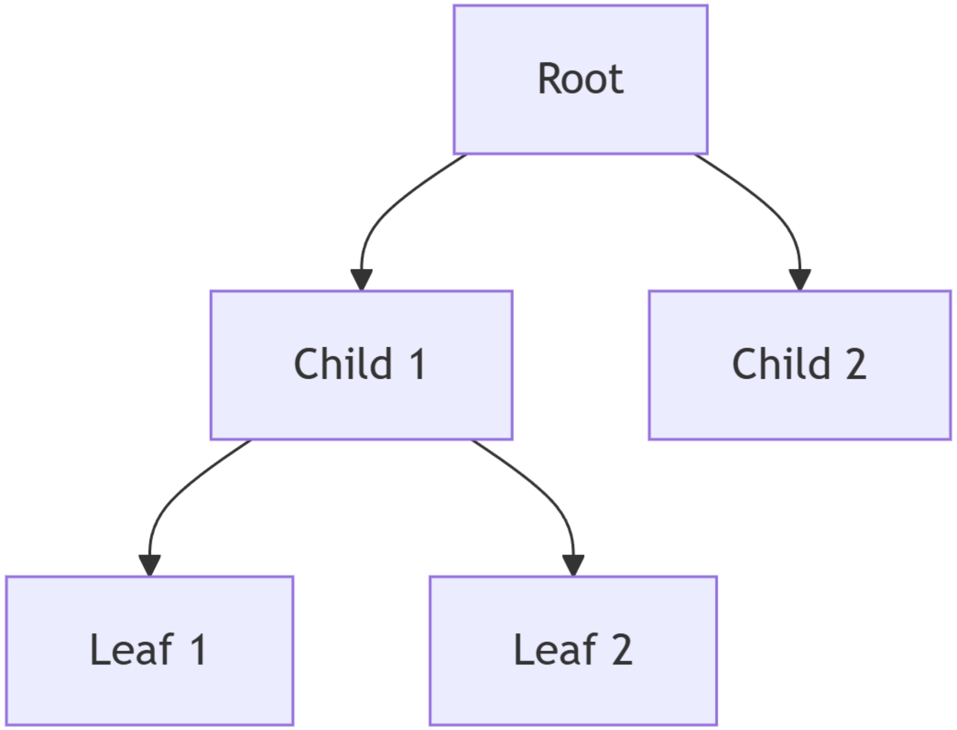
In computer science and mathematics, a **tree** is a hierarchical data structure consisting of **nodes** connected by **edges**. Key properties:

Root Node: The topmost node (starting point).

Parent/Child Nodes: Nodes branch out from parents to children.

Leaf Nodes: Terminal nodes with no children.

Non-linear Structure: Unlike arrays/lists, trees allow branching.



Use of Trees

Classification: Spam detection, medical diagnosis.

Regression: Sales forecasting, risk analysis.

Feature Selection: Identifies important features.

***Note:**

Decision trees form the basis for advanced algorithms like **Random Forests** and **Gradient Boosting Machines (GBM)**.

Regression

Definition:

A supervised learning method that predicts **continuous numerical values** (real numbers).

Examples:
- Predicting house prices ($450,000, $520,000, etc.)
- Forecasting temperature (28.5°C, 30.1°C)
- Estimating stock market trends

Algorithms:
- Linear Regression
- Decision Tree Regression
- Random Forest Regression

Classification

Definition: A supervised learning method that predicts **discrete categorical labels** (classes).
Examples:

- Spam detection ("Spam" or "Not Spam")
- Disease diagnosis ("Positive" or "Negative")
- Image recognition ("Cat", "Dog", "Bird")

Algorithms:

- Logistic Regression
- Decision Tree Classifier
- Support Vector Machines (SVM)

Purpose of Pandas, Math, and NumPy Libraries

Pandas is used for data loading, manipulation, and preprocessing (like handling CSV files and Data Frames).

NumPy enables fast numerical operations and array manipulations (like threshold comparisons and binning numerical features).

Math provides basic mathematical functions (like logarithms and exponents) for entropy and Gini impurity calculations.

Key Differences

Feature	Regression	Classification
Output Type	Continuous (numbers)	Discrete (categories)
Goal	Predict a quantity	Assign a class label
Examples	Price, temperature, age	Yes/No, labels (A/B/C), binary outcomes
Evaluation	MSE, RMSE, R^2	Accuracy, Precision, Recall

Limitations of CART

Overfitting: Deep trees can overfit noisy data unless pruned or constrained.

Instability: Small changes in the data can lead to different tree structures.

Bias Toward Dominant Classes: Can struggle with imbalanced datasets.

Greedy Algorithm: The splitting process is locally optimal, not globally optimal.

Limited Expressiveness: Single trees may not capture complex relationships as well as ensemble methods like Random Forests.

Practical Applications

CART is used in various domains, including:

Finance: Credit risk assessment, fraud detection.

Healthcare: Disease diagnosis, patient outcome prediction.

Marketing: Customer segmentation, churn prediction.

Environmental Science: Predicting weather patterns or species distribution.

Manufacturing: Quality control and fault detection.

Machine Learning Fundamentals

Term	Explanation
Supervised Learning	ML paradigm where models learn from labeled training data
Classification	Predicting discrete categories (e.g., "Yes/No")
Regression	Predicting continuous values (e.g., temperature)
Target Variable	The output variable being predicted (called Decision in this code)
Features	Input variables used for prediction (columns in the dataset)

Decision Tree Concepts

Term	Explanation
Node	A point in the tree that contains a decision rule
Root Node	Topmost decision node (first split)
Leaf Node	Terminal node that provides final prediction
Splitting	Dividing data based on feature values
Pruning	Removing unnecessary branches to prevent overfitting

Splitting Criteria

Term	Explanation
Gini Impurity	Measure of node purity (0 = perfectly pure) used in classification
Entropy	Alternative impurity measure (not used in calculations here)
Standard Deviation Reduction	Regression equivalent of impurity reduction
Threshold	Value used to split continuous features (e.g., "≤25.4")

Data Handling Terms

Term	Explanation
Continuous Feature	Numeric values requiring binning (e.g., temperature)
Categorical Feature	Discrete values (e.g., "Sunny/Rainy")
DataFrame	Pandas 2D data structure (table-like)
dtype	Data type (object=string, float64=numeric)

Algorithm-Specific Terms

Term	Explanation
Global Stdev	Overall standard deviation of regression target
Weighted Impurity	Average impurity across child nodes after split
Value Counts	Frequency distribution of categories
Early Stopping	Termination condition (e.g., stddev < 40% of global)

Programming Concepts

Term	Explanation
Recursion	Function calling itself (used in tree building)
Vectorization	Using NumPy/Pandas for batch operations
Docstring	Function documentation (triple-quoted strings)
Error Handling	try/except blocks for file loading

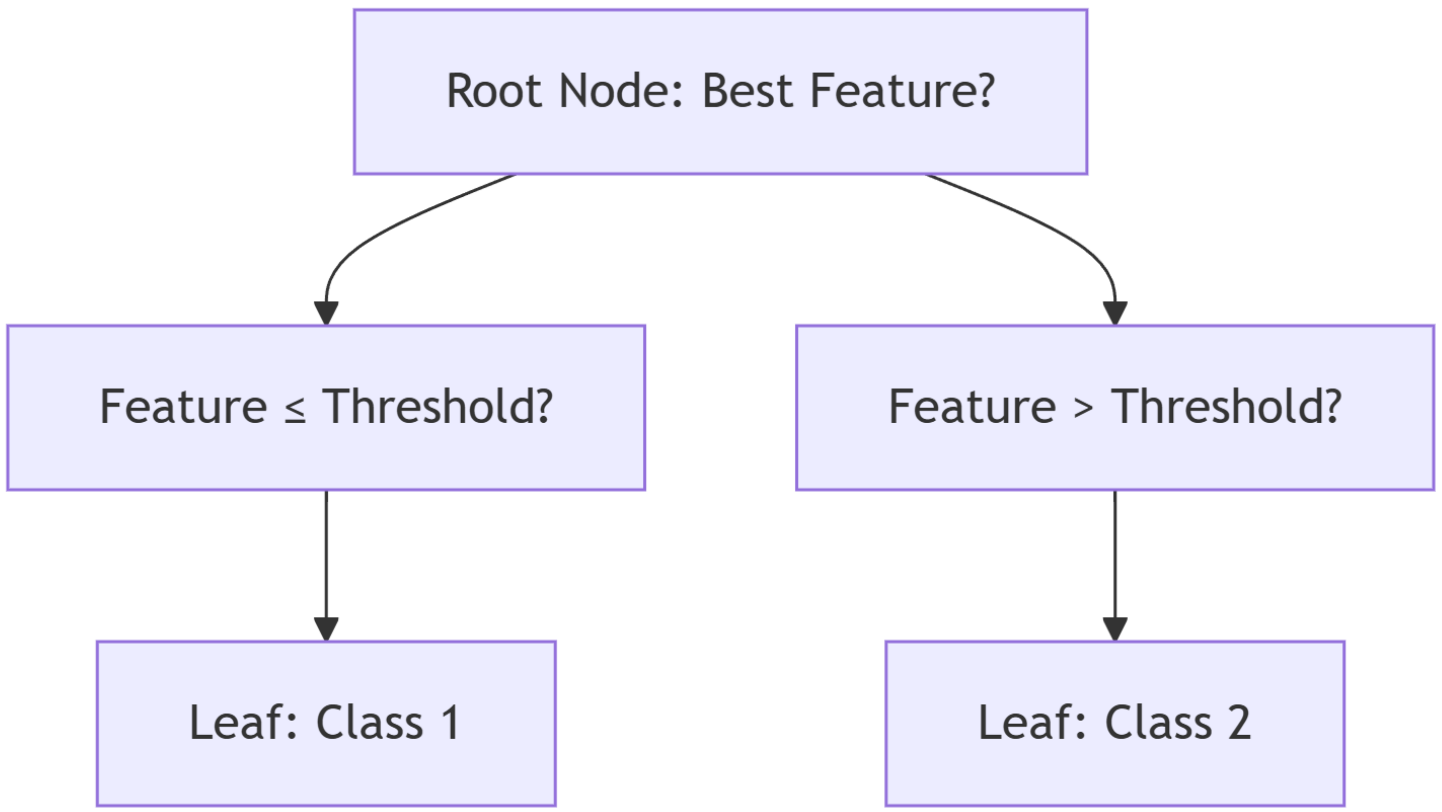
Key Functions Explained

Function	Purpose
processContinousFeatures()	Converts numeric features to categorical bins
calculateEntropy()	Computes entropy (unused in splits but implemented)
findDecision()	Identifies best feature to split on
buildDecisionTree()	Recursively constructs and prints the tree

Critical Variables

Variable	Role
algorithm	Switches between classification/regression
target_column	Dynamically detected output variable
dataset_features	Dictionary storing feature data types
global_stdev	Reference value for regression stopping

Visualization of Key Concepts



CART Dataflow:

Modularity in Machine Learning & Programming

Modularity is a design principle that breaks a system into independent, interchangeable components (*modules*), each responsible for a specific task. In machine learning (ML) and software development, it promotes **readability, reusability, and maintainability**.

Data Loading Module

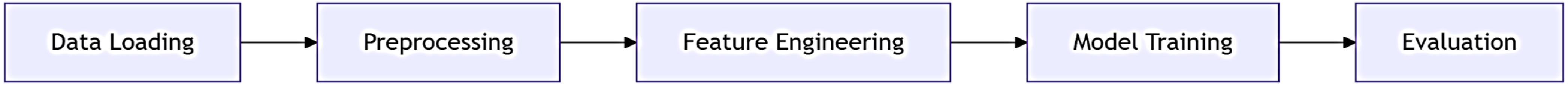
- Reads CSV/JSON files → Outputs a DataFrame.

Preprocessing Module

- Handles missing values, scaling → Outputs clean data.

Model Training Module

- Accepts cleaned data → Outputs a trained model.



What is a Threshold?

A threshold is a predefined cutoff value used to make decisions in ML models. It converts raw model outputs (probabilities, scores) into actionable predictions or classifications.

Applications of Thresholds in ML

Application	How Threshold is Used	Example
Binary Classification	Converts probabilities (0–1) to class labels (0/1).	Spam detection (≥0.5 = "Spam").
Medical Diagnosis	Balances sensitivity/specificity (e.g., lower threshold to catch more diseases).	Cancer screening (≥0.3 = "Positive").
Recommendation Systems	Filters items by relevance score (e.g., recommend if predicted rating ≥ 4/5).	Netflix movie suggestions.
Autonomous Vehicles	Object detection confidence threshold (e.g., ignore detections with <80% confidence).	Tesla's pedestrian detection.
Natural Language Processing	Discards low-confidence intent classifications in chatbots.	Customer support bots.

CART Implementation Continue…

The data flow in this decision tree implementation follows a top-down recursive splitting approach, beginning with data loading and preprocessing before constructing the tree structure.

Initially, the script loads the dataset into a Pandas ***DataFrame (`df`)*** and identifies the target column through flexible name matching. It then analyzes feature types, storing them in ***dataset_features***, and validates the algorithm choice (classification/regression).

CART Implementation Continue…

For numeric features, ***processContinousFeatures()*** dynamically determines optimal thresholds, converting them into categorical bins (e.g., "≤25.4" or ">25.4") using Gini impurity (classification) or standard deviation reduction (regression).

The ***findDecision()*** function evaluates all features to select the best split based on these metrics, while ***buildDecisionTree()*** recursively partitions the data, printing rules at each node.

CART Implementation

The recursion terminates when meeting leaf conditions: pure nodes (classification), sufficiently small standard deviation (regression), or exhausted features. Throughout this flow, temporary DataFrames (`subset1`, `subset2`, `temp_df`) handle data splits without modifying the original dataset, ensuring clean separation of training logic from data storage.

The process emphasizes modularity, with impurity calculations, threshold selection, and tree construction isolated in dedicated functions, making the workflow both interpretable and extensible.

Coding Marathon

LET'S DIV DEEP INTO CODE