DATA2060 FINAL PROJECT

Classification And Regression Tree(CART) for regression on airfield dataset

PRESENTED BY:
JUNHE CHEN, HAN HUANG, JUNWEI SU,XUANYAO QIAN,

CART REGRESSION

SUMMARY

CART (Classification and Regression Trees) is a non-parametric, recursive partitioning algorithm that predicts continuous outcomes by partitioning the feature space into regions and assigning a constant value to each region.
Unlike linear models, CART does not assume linearity, independence, or normality; instead, it directly minimizes the sum of squared errors (SSE) within each region.

ADVANTAGES

- Interpretable: predictions can be explained by simple rules ("if feature j ≤ t").
- Handles nonlinear relationships and feature interactions.
- Works with unscaled, mixed-type features (after encoding categories).
- Forms the base learner for powerful ensembles (Random Forests, Gradient Boosting).

DISADVANTAGES

- High variance: small changes in data can lead to very different trees.
- Piecewise-constant predictions are discontinuous and can be unstable.
- Greedy local optimization does not guarantee a globally optimal tree.
- Without depth limits or pruning, trees can severely overfit.

CART REGRESSION

REPRESENTATION

A CART regression tree represents a piecewise-constant function. It partitions the input space $\mathbb{R}^d$ into a finite set of axis-aligned regions:

$$R_1, R_2, \ldots, R_M, \qquad R_m \subset \mathbb{R}^d,\ R_i \cap R_j = \varnothing.$$

Each region corresponds to a leaf node in the tree. A prediction is constant within that region.
A root-to-leaf path consists of binary splits of the form:

$$x_{jk} \leq t_k,$$

Each region corresponds to a leaf node in the tree. A prediction is constant within that region.
A root-to-leaf path consists of binary splits of the form:

$$\hat{f}(x) = \sum_{m=1}^{M} c_m \, 1\{x \in R_m\},$$

where the constant value assigned to region R_m is simply the mean of the training labels in that region:

$$c_m = \frac{1}{|R_m|} \sum_{i: x_i \in R_m} y_i.$$

THUS, THE REPRESENTATION OF A CART MODEL CONSISTS OF:

A partition of feature space into regions R_m defined by axis-aligned threshold splits.

A constant prediction c_m for each region, equal to the average target value within that region.

This yields an interpretable model:
each prediction is obtained by following a path of feature-threshold decisions from the root to a leaf, where the leaf stores the constant prediction.

CART REGRESSION

LOSS

At each node, CART evaluates the quality of a set of targets $S = \{y_i\}_{i=1}^{n}$ using Mean Squared Error (MSE) impurity:

$$I(S) = \frac{1}{|S|} \sum_{i \in S} (y_i - \bar{y})^2, \qquad \bar{y} = \frac{1}{|S|} \sum_{i \in S} y_i.$$

For a candidate split on feature j at threshold t, the data are partitioned into:

$$S_L = \{i : x_{ij} \le t\}, \qquad S_R = \{i : x_{ij} > t\}.$$

The post-split impurity is the weighted average impurity of the two child nodes:

$$I_{\text{child}}(j,t) = \frac{|S_L|}{|S|} I(S_L) + \frac{|S_R|}{|S|} I(S_R).$$

The impurity reduction (or "gain") from the split is:

$$\Delta I(j,t) = I(S) - I_{\text{child}}(j,t).$$

CART chooses the split (j,t) that maximizes this reduction—i.e., the split that most decreases MSE.

LOSS

Usual MSE between the predicted piecewise-constant function $\hat{f}$ and the true targets:

$$\mathcal{L}(\hat{f}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{f}(x_i))^2.$$

CART REGRESSION

OPTIMIZER

Unlike linear or neural network models, CART does not optimize a parameter vector with gradient descent. Instead, it performs a combinatorial search to determine:

1. Which feature jjj to split on.
2. Which threshold ttt to use for that feature.
3. How deep the tree should grow.

At each non-leaf node, CART solves the discrete optimization problem:

$$(j^*, t^*) = \arg\max_{j,t} \Delta I(j, t),$$

where $\Delta I(j,t)$ is the impurity reduction obtained from splitting on feature j at threshold t.

The tree is built using a top-down recursive binary splitting algorithm. For each node, CART scans over all features and all possible thresholds, computes the impurity reduction for each, and picks the split yielding the largest gain. If no split improves impurity sufficiently, the node becomes a leaf predicting the mean of its targets.

STOPPING CRITERIA

- Maximum depth: depth > max_depth
- Too few samples to split : |S| < min_sample_split
- Too few samples to create leaves: |S_L|, |S_R| < min_samples_leaf
- Insufficient impurity reduction: $\Delta I(j,t)$ < min_impurity_decrease

PSEUDOCODE

```
BuildTree(S, depth):
    if stopping criterion is satisfied:
        return Leaf(mean(targets in S))

    best_gain = 0
    best_feature, best_threshold = None, None

    for each feature j:
        for each threshold t:
            compute impurity reduction ΔI(j, t)
            if ΔI(j, t) > best_gain:
                update best feature/threshold

    if best_feature is None:
        return Leaf(mean(targets in S))

    split S into S_L, S_R using best_feature and best_threshold
    left_child = BuildTree(S_L, depth + 1)
    right_child = BuildTree(S_R, depth + 1)

    return Node(best_feature, best_threshold, left_child, right_child)
```

MODEL RESULTS

ABOUT DATASET

This dataset contains daily weather and runway-condition measurements collected at an airport, along with the corresponding Aircraft Total Movements, which serves as the target variable for prediction.

- 570 Training Set Size
- 245 Testing Set Size
- 9 Features

TARGET VARIABLE

Aircraft total movements:
The total number of aircraft operations recorded on a given day (including takeoffs and landings).

INPUT FEATURES

Meteorological Features(numerical):
Days of Air Frost, Precipitation, Sunshine Hours, Humidity, Wind Speed

Weather Category Feature:
Ordinal feature indication weather and flight condition from 0-3

Runway Temperature Features:
Min&Max Temperature of Runway Surface

OUR CART VS. SKLEARN

SYNTHETIC DATA

$$y = 4x_1 - 2x_2 + \epsilon$$

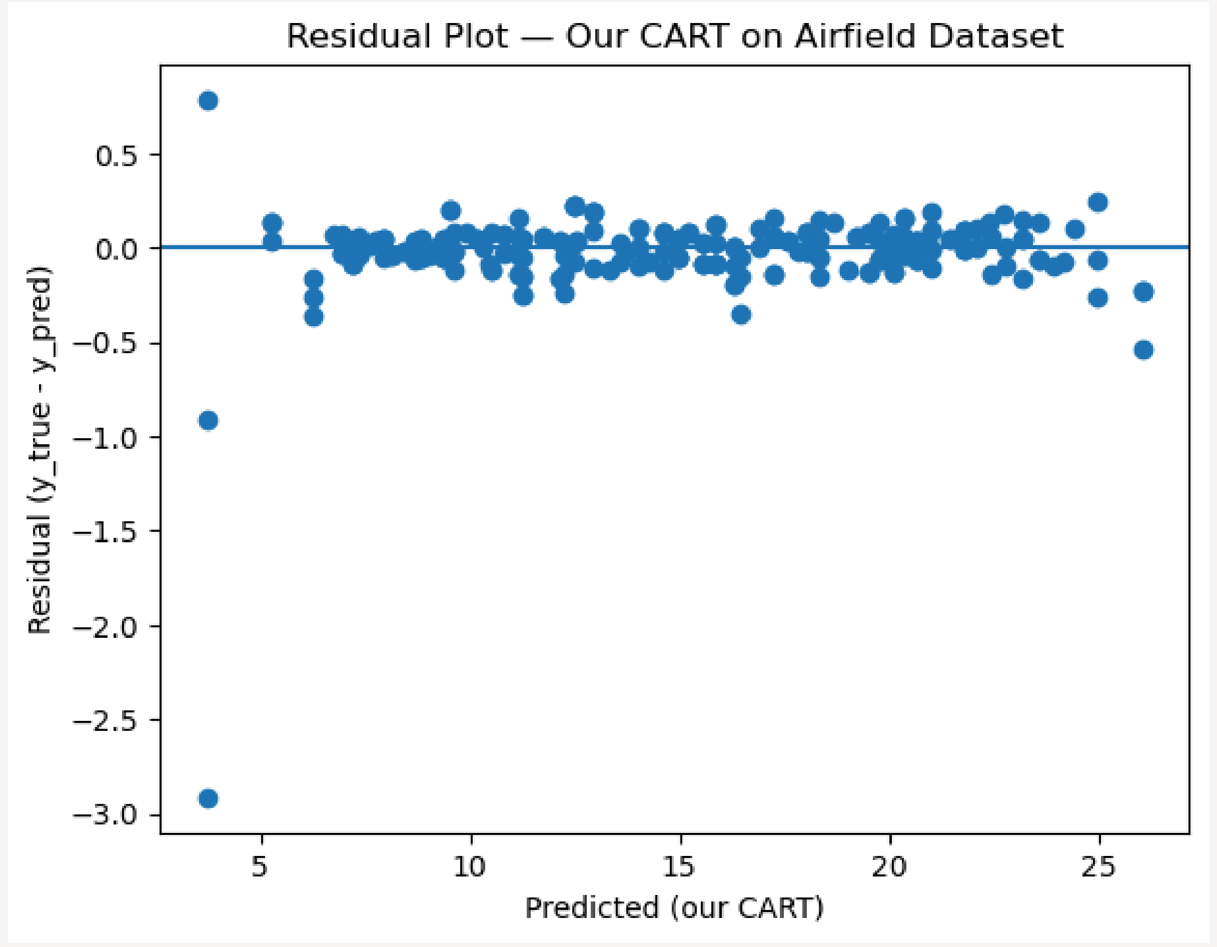
MODEL	MSE
OUR CART	2.869332471994586
SKLEARN CART	2.8693324719945856
DIFFERENCE	2.0872192862952942e-16

AIRFIELD DATA

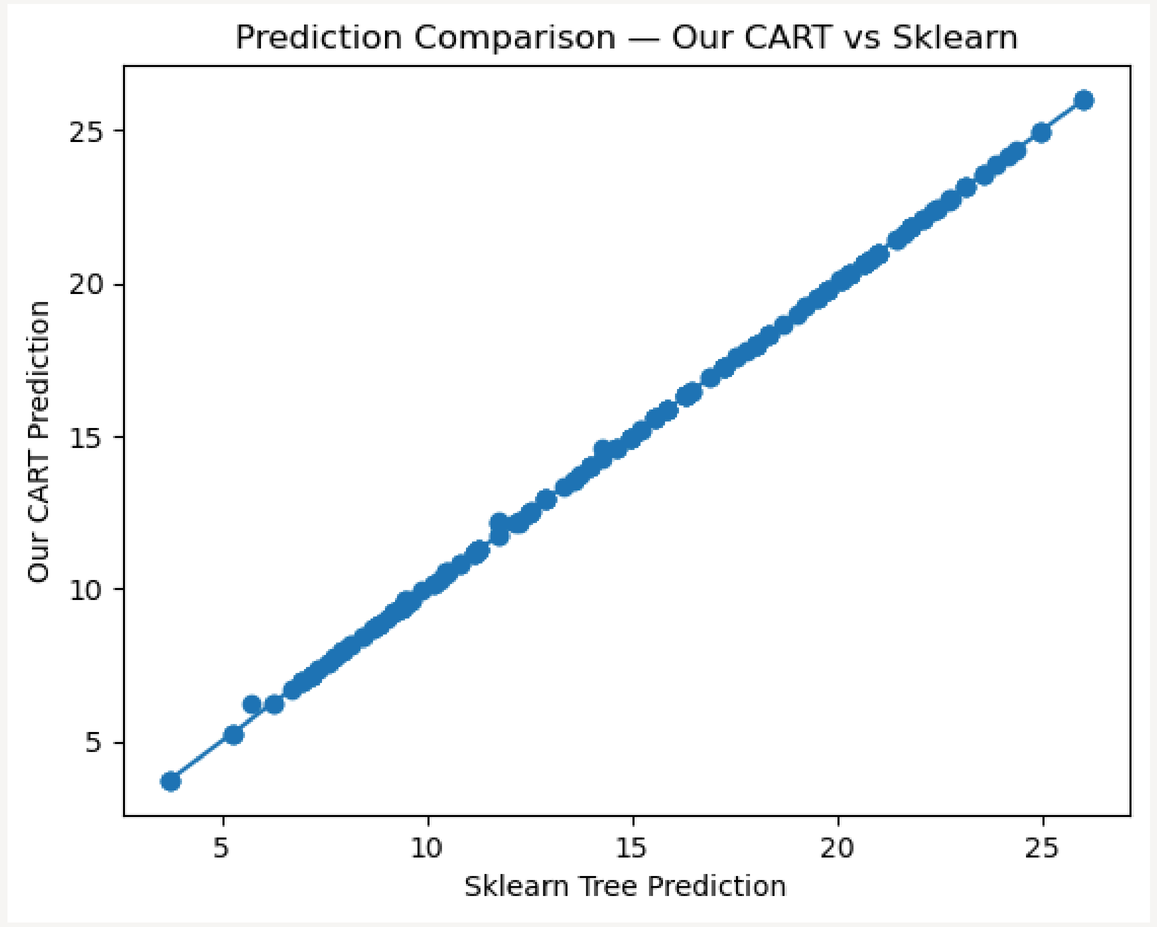
MODEL	MSE
OUR CART	0.05238581738704548
SKLEARN CART	0.05235679244373483
DIFFERENCE	0.0008344671201823231

MODEL RESULTS cont.

RESIDUAL PLOT



PREDICTION COMPARISON



- The tree model captures the general relationship in the dataset.
- Errors appear random—not systematically biased.
- Occasional outliers likely reflect unobserved factors

- Numerical differences between the two models are negligible.
- Our CART implementation successfully reproduces the behavior of scikit-learn’s regression tree.

SUMMARY

WHAT WAS INTERESTING ABOUT CART

- Implementing the split search made me realize how sensitive CART is to threshold selection
- Using unique feature values as thresholds works logically, but produces noticeably different trees from sklearn.
- Switching to midpoint thresholds between sorted unique values (as sklearn does) significantly reduces prediction discrepancies..

Unique thresholds	MSE
OUR CART	0.052189383185246845
SKLEARN CART	0.05235679244373483
DIFFERENCE	0.07358476774690736

Midpoint thresholds	MSE
OUR CART	0.05238581738704548
SKLEARN CART	0.05235679244373483
DIFFERENCE	0.0008344671201823231

WHAT WAS CHALLENGING

- Ensuring the split logic exactly matches the theoretical CART algorithm and sklearn’s behavior.
- Handling edge cases: Empty splits, Minimum samples per leaf, Impurity reduction too small

THANKYOU