Decision Tree Algorithm

A decision tree is a flowchart-like tree structure where an internal node represents a feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome.

The Decision Tree Algorithm: Entropy | ML-DAWN

Which Attribute to Choose? (Part1) In our last post,&nbsp; we introduced the idea of the decision trees (DTs) and you understood the big picture. Now

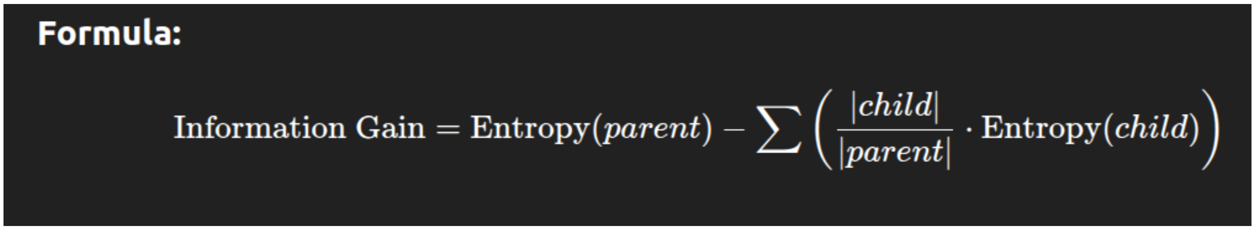
https://www.mldawn.com/decision-trees-entropy/

Choosing the Best Feature at each step

✅ 1. Information Gain (ID3 Algorithm)

- Based on **entropy**, which measures the impurity or disorder of a dataset.
- The feature that **reduces entropy the most** after the split is chosen.

🎯 Goal: Pick the feature that gives **maximum information gain**.



Formula:

$$\text{Information Gain} = \text{Entropy}(parent) - \sum \left(\frac{|child|}{|parent|} \cdot \text{Entropy}(child) \right)$$

✅ 2. Gain Ratio (C4.5 Algorithm)

- Improves on information gain by adjusting for features with **many unique values** (which can bias IG).

🎯 Goal: Pick feature with **highest gain ratio**, where:

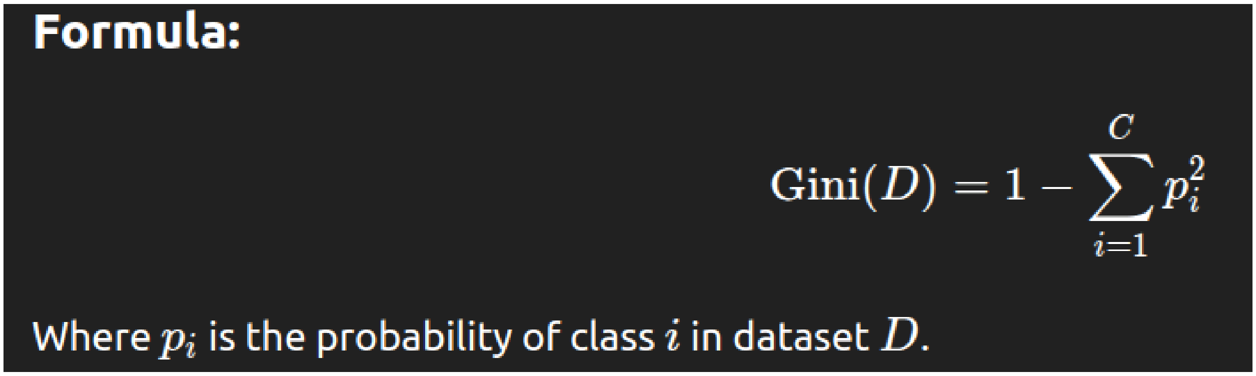
Gain Ratio=Information Gain / Split Information

This penalizes splits that just break the data into tiny unique chunks.

✅ 3. Gini Index (CART Algorithm)

- Measures impurity like entropy but is **computationally simpler**.
- Used in **CART** (Classification And Regression Trees).

🎯 Goal: Choose feature that **minimizes the Gini impurity** after the split.



Formula:

$$\text{Gini}(D) = 1 - \sum_{i=1}^{C} p_i^2$$

Where p_i is the probability of class i in dataset D.

and some other methods...

Decision Tree split for Numerical Features

To split continuous features in decision trees, you need to identify the optimal threshold (value) that separates the data into two groups. The goal is to find the split that minimizes impurity (for classification) or minimizes error (for regression).

Here's how you can split a continuous feature:

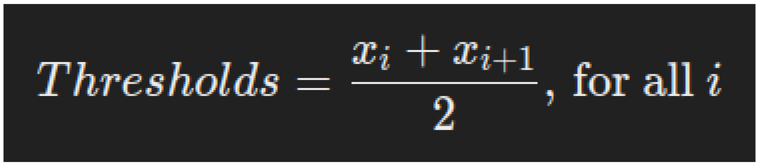
Steps to Split a Continuous Feature

1. Sort the Data:

- Arrange the data values of the continuous feature in ascending order.

2. Identify Possible Thresholds:

- Consider the midpoints between consecutive feature values as potential thresholds.
- For a sorted feature X=[x1,x2,x3,...,xn], potential thresholds are:



$$Thresholds = \frac{x_i + x_{i+1}}{2}, \text{ for all } i$$

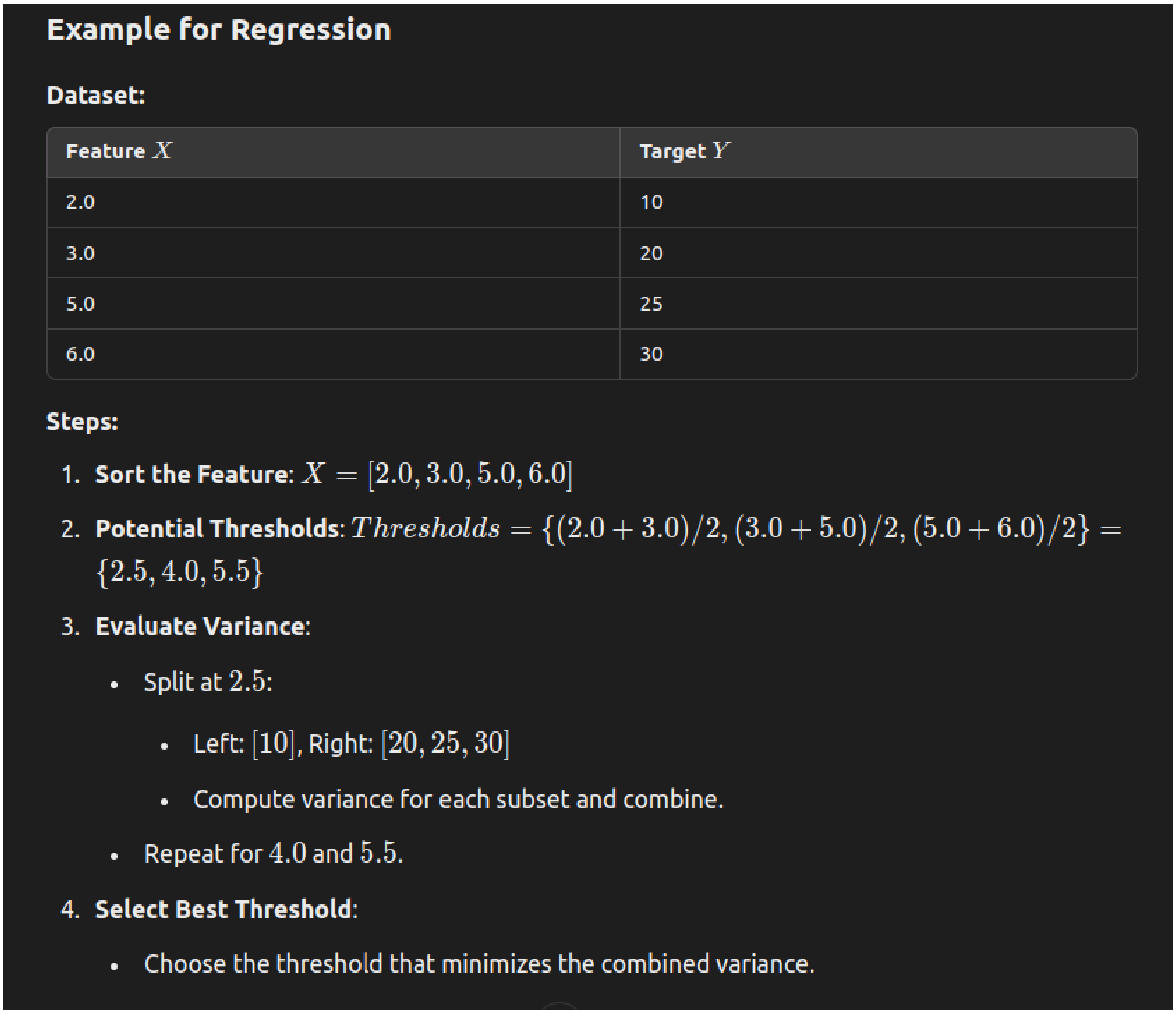
3. Evaluate Each Threshold:

- Split the dataset into two subsets:
  - Left subset: X≤Threshold
  - Right subset: X>Threshold
- Calculate the impurity or error for the split:
  - **For Classification**: Use metrics like Gini Impurity or Information Gain.
  - **For Regression**: Use metrics like Variance Reduction or Mean Squared Error (MSE).

4. Choose the Best Threshold:

- The best threshold is the one that minimizes the weighted impurity/error of the two subsets.

Example:



Example for Regression

Dataset:

Feature X	Target Y
2.0	10
3.0	20
5.0	25
6.0	30

Steps:

1. **Sort the Feature**: $X = [2.0, 3.0, 5.0, 6.0]$
2. **Potential Thresholds**: $Thresholds = \{(2.0 + 3.0)/2, (3.0 + 5.0)/2, (5.0 + 6.0)/2\} = \{2.5, 4.0, 5.5\}$
3. **Evaluate Variance**:
   - Split at 2.5:
     - Left: $[10]$, Right: $[20, 25, 30]$
     - Compute variance for each subset and combine.
   - Repeat for 4.0 and 5.5.
4. **Select Best Threshold**:
   - Choose the threshold that minimizes the combined variance.

Decision Tree Pruning

Refer this blog to explore further

Decision Tree Pruning: The Hows and Whys - KDnuggets

Decision trees are a machine learning algorithm that is susceptible to overfitting. One of the techniques you can use to reduce overfitting in decision trees is pruning.

https://www.kdnuggets.com/2022/09/decision-tree-pruning-hows-whys.html