Decision Trees

Mon, 23 May 2022 at 17:19

#entropy #information #gain #classification

Overview

DEFINITION

> *"A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements" wikipedia*

Used for classification and when dealing with continuous output regression trees are used instead.

INPUT

`dataset` with attributes and class output. Input variables can be both **discrete** or **continuous**.

OUTPUT

`probability scores` of class membership
`a tree` where its leaves are the probability score or each class value

Use Cases

Mostly used when a series of yes/no questions most be answered to arrive at the classification. Examples:

- Biological species classification
- Patient symptoms evaluation

When if/then conditions are needed. Examples:

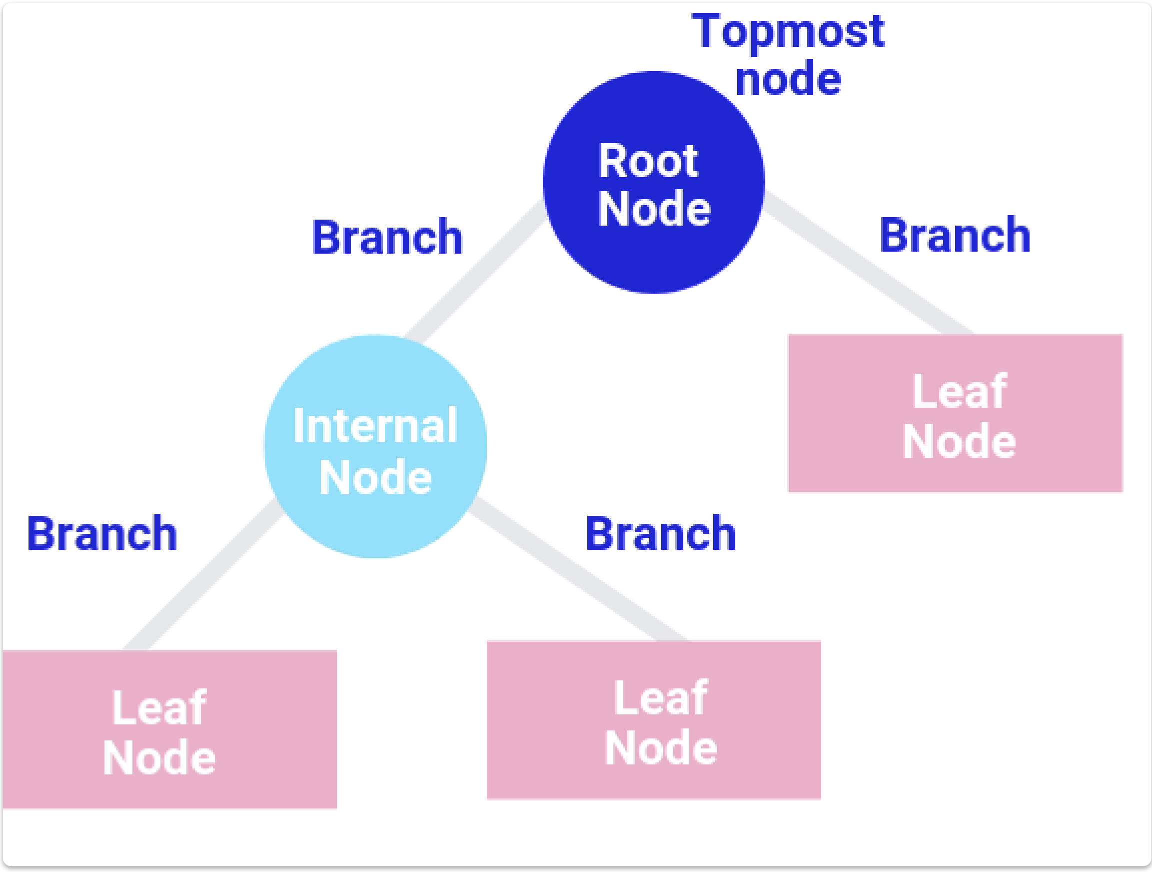
- Predicting customers segments
- Loan/mortgage approvals
- Fraud detection

Decision Trees Terminology

STRUCTURE

Decision trees are a flow-chart representation of if/then statements.

1. *Root Node* → top node
2. *Internal Nodes* → decision nodes
3. *Leaf Nodes* → class labels
4. *Branches* → decision output
5. *Depth* → no. of branches from current node to root
    1. a `stump` is a tree of depth 1



ALGORITHM

1. *Purity* → probability of the corresponding class based on the node's decision
    1. 100% **purity** when all of the node's data belongs to `one` class
    2. 100% **impurity** when the node's data is evenly split *same records of each class*
2. *Splitting rules* → defines how a decision tree is split

Tree Induction Algorithm

First: choose the most informative feature

Two ways to choose whether an attribute is informative or not:

1. *Gini Index* - used in CART Algorithm

$$Gini_x = 1 - \sum_{\forall x \in X} P(x)^2$$

2. *Entropy* - measures the `impurity` of an attribute. You first find the **base entropy** of the current dataset then find each attribute's **conditional entropy**. Finally, you calculate the **information gain** and `choose the most informative feature`

$$\textit{Base Entropy} \quad H_X = -\sum_{x \in X} P(x) \log_2 P(x)$$

$$\textit{Conditional Entropy} \quad H_{Y|X} = -\sum_{x \in X} P(x) \sum_{y \in Y} P(y|x) \log_2 P(y|x)$$

$$InfoGain = H_X - H_{Y|X}$$

- The minimum value of entropy is 0 which means the node is pure
- The maximum value of entropy is $\log_2 k \;\; ; \;\; k = no.\ of\ classes$

> - when the maximum value of entropy is reached; all classes are equally probable
>
> | | |
> |---|---|
> | impurity ↑ | entropy ↑ |
> | purity ↓ | entropy ↑ |

Second: split according to the feature

Third: repeat until split is pure

OTHER ALGORITHMS

ID3 Algorithm

C4.5 Algorithm

CART Algorithm

Pros & Cons

Reasons to Choose	Cautions
Takes any input type	Axis-aligned
Robust with redundant/correlated variables	Structure is sensitive to small changes in training data
Can handle non-linear variables	Can easily over-fit the data as depth increases
Computationally efficient	Does not handle missing values well
Easy to score and understand	Decision rules can be very complex