

# All about decision trees



Abhishek Jain

Follow

5 min read · Sep 13, 2024



116



1



In machine learning, **decision trees** are one of the most intuitive and widely-used algorithms for classification and regression tasks. They model decision-making processes in a tree-like structure where each internal node represents a feature (attribute), each branch represents a decision rule, and each leaf node represents the outcome. Decision trees are easy to visualize and interpret, making them an excellent tool for explaining the logic behind predictions.

In this blog, we will explore the basics and the mathematics of decision trees, the ID3 algorithm (a popular method for building decision trees), and CART (Classification and Regression Trees).

## What is a Decision Tree?

A **decision tree** is a supervised learning algorithm that can be used for both classification and regression tasks. It works by recursively partitioning the

data into subsets based on feature values, making decisions at each node to maximize a specific criterion (e.g., information gain or Gini index).

### Key Components:

- **Root Node:** The top node in the tree that represents the best feature to split the data.
- **Internal Nodes:** Represent the features used for splitting the data based on specific decision rules.
- **Leaf Nodes:** Terminal nodes that represent the predicted outcome (class label or numerical value).
- **Branches:** Connections between nodes representing the possible values of the features.

### How Decision Trees Work?

The process of creating a decision tree involves:

1. **Selecting the Best Attribute:** Using a metric like Gini impurity, entropy, or information gain, the best attribute to split the data is selected.
2. **Splitting the Dataset:** The dataset is split into subsets based on the selected attribute.
3. **Repeating the Process:** The process is repeated recursively for each subset, creating a new internal node or leaf node until a stopping criterion is met (e.g., all instances in a node belong to the same class or a predefined depth is reached).

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Dataset where play tennis is the output feature and rest all are input features



Decision tree example where the root node chosen is outlook (random)

A **leaf node** in a decision tree is the terminal node at the bottom of the tree, where no further splits are made. Leaf nodes represent the final output or prediction of the decision tree. Once a data point reaches a leaf node, a

decision or prediction is made based on the majority class (for classification) or the average value (for regression) of the data points that reach that leaf.

To check **mathematically** if any split is **pure split** or not we use **entropy or gini impurity**. **Information Gain** helps us to determine which features need to be selected

## Entropy

Entropy is a concept borrowed from information theory and is commonly used as a measure of uncertainty or disorder in a set of data. In the context of decision trees, entropy is often employed as a criterion to decide how to split data points at each node, aiming to create subsets that are more homogeneous with respect to the target variable.

**Entropy is a measure of uncertainty or disorder. A low entropy indicates a more ordered or homogeneous set, while a high entropy signifies greater disorder or diversity.** In the context of a decision tree, the goal is to reduce entropy by selecting features and split points that result in more ordered subsets.

---

Get Abhishek Jain's stories in your inbox

Join Medium for free to get updates from this writer.

Enter your email

Subscribe

---

Entropy values range from 0 to 1. The minimum entropy (0) occurs when all instances belong to a single class, making the set perfectly ordered. The maximum entropy (1) occurs when instances are evenly distributed across all classes, creating a state of maximum disorder.

At each node of a decision tree, the algorithm evaluates the entropy for each feature and split point. **The feature and split point that result in the largest reduction in entropy are chosen for the split.** The reduction in entropy is often referred to as Information Gain and is calculated as the difference between the entropy before and after the split.

## **Gini impurity**

Gini impurity is a measure of the impurity or disorder in a set of elements, commonly used in decision tree algorithms, especially for classification tasks. It quantifies the likelihood of misclassification of a randomly chosen element in the dataset.

**A lower Gini impurity suggests a more homogeneous set of elements within the node, making it an attractive split in a decision tree. Decision tree algorithms aim to minimize the Gini impurity at each node, selecting the feature and split point that results in the lowest impurity.**

## **Entropy and Gini impurity formulas**

$$\text{Entropy}(S) = - \sum_{i=1}^c p_i \log_2(p_i)$$

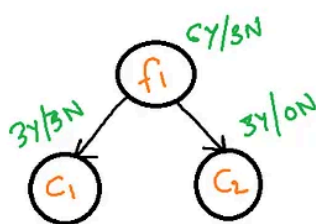
$$\text{Gini}(S) = 1 - \sum_{i=1}^c p_i^2$$

- The minimum value of entropy is 0.

- Thus, for different numbers of classes:
  - For 2 classes (binary classification): maximum entropy is 1.
  - For 3 classes: maximum entropy is  $\log_2(3) \approx 1.585$
  - For 4 classes: maximum entropy is  $\log_2(4) = 2$  and so on
- 
- $G = 0$  indicates a perfectly pure node (all elements belong to the same class).
  - $G = 0.5$  indicates maximum impurity (elements are evenly distributed among all classes)

## Example

Lets take an example of a feature  $f_1$  split into nodes  $c_1$  and  $c_2$ . The feature  $f_1$  has 6 yes and 3 no, and when it is split  $c_1$  it has 3 yes and 3 no and  $c_2$  has 3 yes and 0 no.



$$\text{Entropy of } c_1 = - \left( \frac{3}{6} \log_2 \left( \frac{3}{6} \right) + \frac{3}{6} \log_2 \left( \frac{3}{6} \right) \right) = 1$$

$$\text{Entropy of } c_2 = - \left( \frac{3}{3} \log_2 \left( \frac{3}{3} \right) \right) = 0$$

If entropy of a node is 0, it is pure split

$$\text{Gini impurity of } c_1 = 1 - \left( \left( \frac{3}{6} \right)^2 + \left( \frac{3}{6} \right)^2 \right) = 0.5$$

$$\text{Gini impurity of } c_2 = 1 - \left( \left( \frac{3}{3} \right)^2 + \left( \frac{0}{3} \right)^2 \right) = 0$$

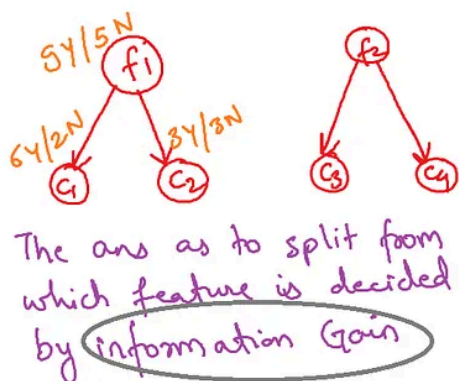
If our dataset is huge we should choose Gini impurity as its calculation is much simpler compared to entropy

## Information Gain

### Information Gain

$$\text{Gain}(S, f_1) = \underbrace{\text{Entropy}(S)}_{\substack{\text{Entropy of node} \\ \text{from which} \\ \text{bifurcation} \\ \text{needs to be} \\ \text{done}}} - \sum_{v \in \text{values}} \underbrace{\frac{|S_v|}{|S|}}_{\substack{\text{Total no of samples}}} \underbrace{\text{Entropy}(S_v)}_{\substack{\text{Entropy of category}}}$$

Information Gain formula



$$\text{Entropy}(f_1) = -\left(\frac{9}{14} \log_2\left(\frac{9}{14}\right) + \frac{5}{14} \log_2\left(\frac{5}{14}\right)\right) \approx 0.94$$

$$\text{Entropy}(C_1) = -\left(\frac{6}{8} \log_2\left(\frac{6}{8}\right) + \frac{2}{8} \log_2\left(\frac{2}{8}\right)\right) \approx 0.81$$

$$\text{Entropy}(C_2) = -\left(\frac{3}{6} \log_2\left(\frac{3}{6}\right) + \frac{3}{6} \log_2\left(\frac{3}{6}\right)\right) = 1$$

$$\text{Gain}(S, f_1) = 0.94 - \left(\frac{8}{14} \times (0.81) + \frac{6}{14} \times (1)\right) \approx 0.049$$

$$\text{Assume } \text{Gain}(S, f_2) = 0.052$$

Here in information gain formula -> 8 comes from (6Yes and 2No) and 14 comes from (9Yes and 5No). Similarly 6 comes from (3Yes and 3No) and 14 comes from (9Yes and 5No)

Since the information gain of  $f_2$  is greater than that of  $f_1$ , the split will be done with respect to that of  $f_2$

Decision Tree

Entropy

Gini Impurity

Information Gain

Machine Learning



## Written by Abhishek Jain

Follow

894 followers · 34 following

Data Science Enthusiast || AI-ML Engineer Linked in :  
<https://www.linkedin.com/in/abhishek-jain-pict-aiml-enthusiast/>

## Responses (1)



Write a response

Open in app ↗

Sign up

Sign in

# Medium



Search



Write



Kaustubhdeshmukh

Jun 30



great blog!!



1

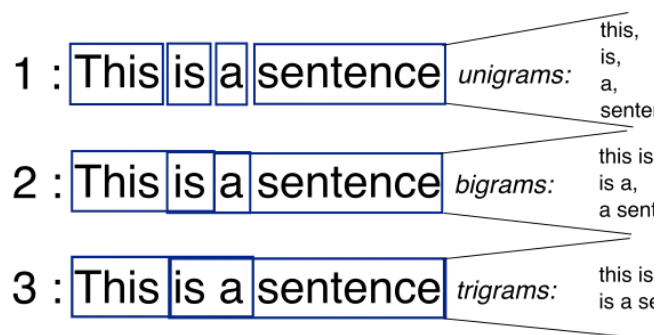


1 reply

[Reply](#)



# More from Abhishek Jain

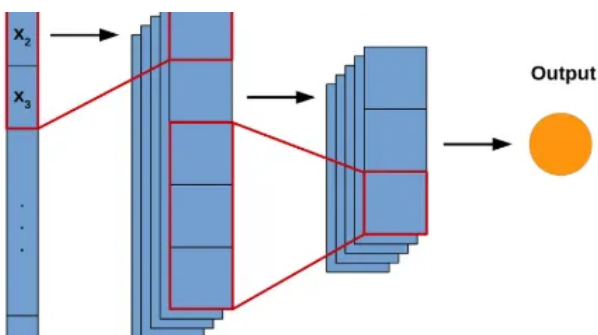


Abhishek Jain

## N-grams in NLP

N-grams, a fundamental concept in NLP, play a pivotal role in capturing patterns and...

Feb 5, 2024 210 5



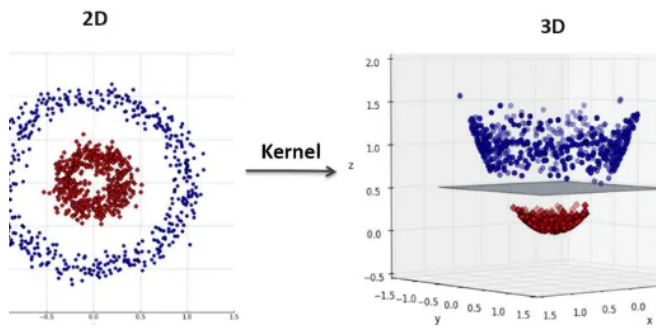
Abhishek Jain

## Understanding the 1D Convolutional Layer in Deep...

Convolutional layers are one of the cornerstones of deep learning, particularly i...

Jan 7 50 1





 Abhishek Jain

## SVM kernels and its type

Support Vector Machines (SVMs) are a popular and powerful class of machine...


Sep 11, 2024  117  2



Sent 2 : good girl  
Sent 3 : boy girl good

$TF = \frac{\text{No of rep of words in a sentence}}{\text{No of words in a sentence}}$

$IDF = \log / \text{No of sentences}$

 Abhishek Jain

## TF-IDF in NLP (Term Frequency Inverse Document Frequency)

In the realm of Natural Language Processing (NLP), TF-IDF (Term Frequency-Inverse...

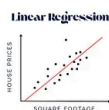
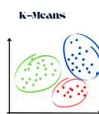
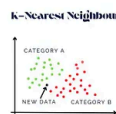
Feb 4, 2024  85  8



See all from Abhishek Jain

## Recommended from Medium

### TOP 10 MACHINE LEARNING ALGORITHMS EXPLAINED





In Learning Data by Rita Angelou

## 10 ML Algorithms Every Data Scientist Should Know—Part 1

I understand well that machine learning might sound intimidating. But once you break dow...



Jun 10



38



October

## How One Interview Changed Everything About How I See Hiring

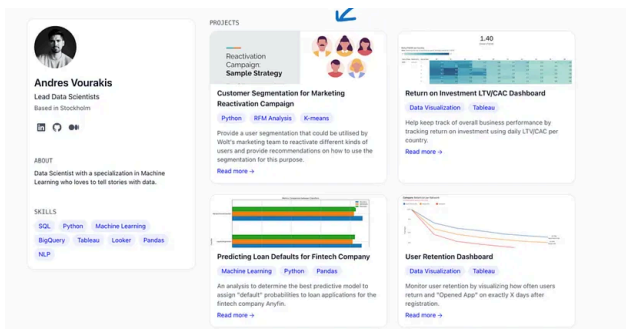
Another Monday, another interview story.



Jun 9



3



In Data Science Collective by Andres Vourakis

## Ultimate Guide to Building a Data Science Portfolio (That Lands...

The best way to bridge the experience gap in 2025–2026



In Level Up Coding by Jacob Ingle

## DIY AI & ML: Decision Trees for Classification

A step-by-step guide on how to build a Classification Decision Tree Object in Pytho...



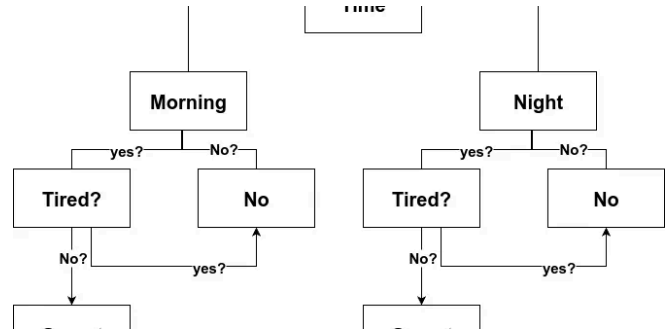
Apr 21



264



2



Hemasri

## Decision Tree Classifier

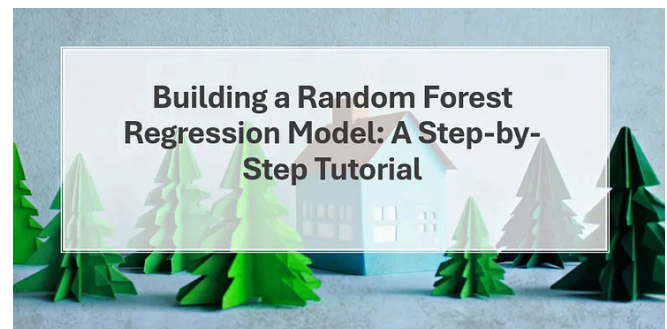
Part 4: Learning Machine Learning/Deep Learning



Sep 4



51



Vikash Singh

## Building a Random Forest Regression Model: A Step-by-Ste...

Learn how to build a powerful regression model using Random Forest—from data...

★ Sep 2 🖱 132



★ Jun 18 🖱 25



---

See more recommendations