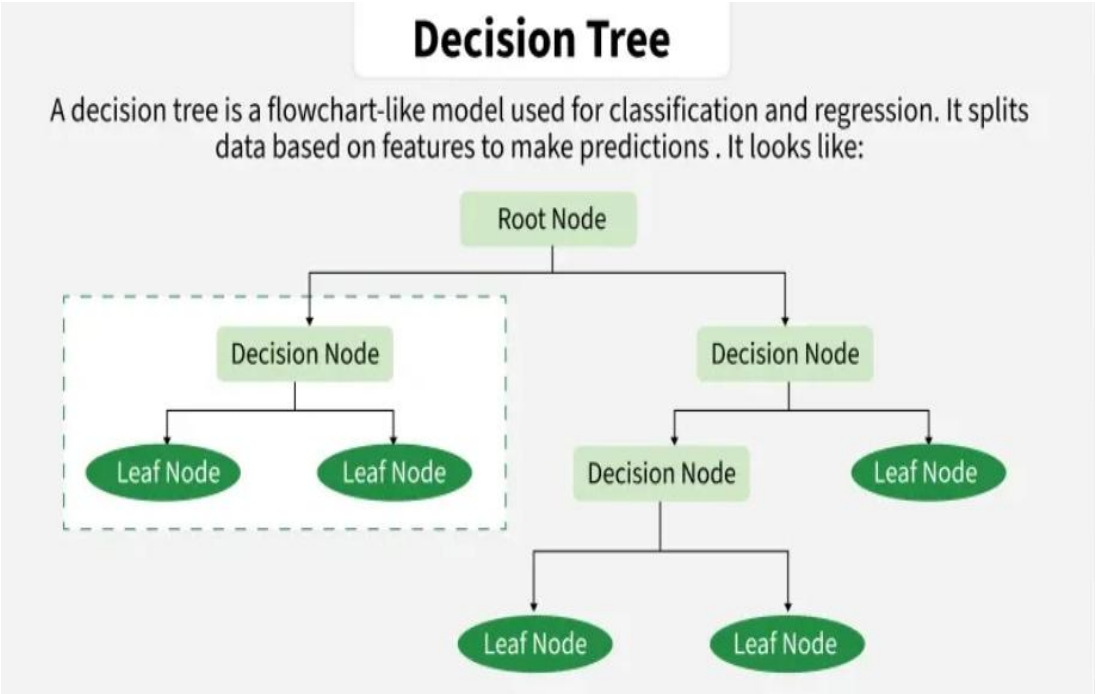
DECISION TREES IN MACHINE LEARNING

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.
-



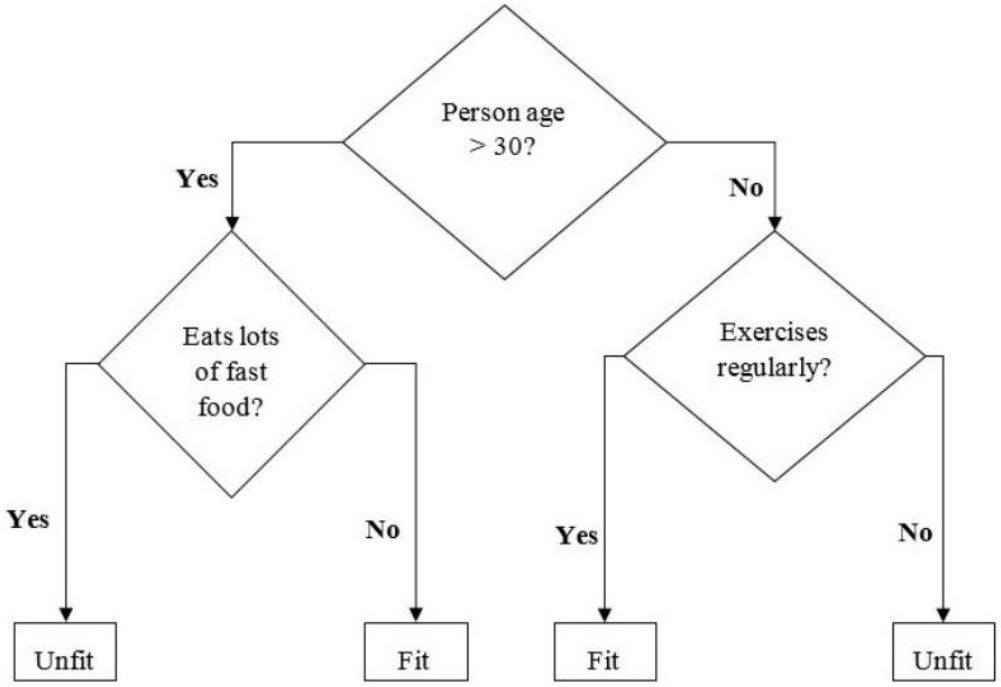
How Decision Trees Work?

- The process of creating a decision tree involves:

- Selecting the Best Attribute: Using a metric like Gini impurity, entropy, or information gain, the best attribute to split the data is selected.
- Splitting the Dataset: The dataset is split into subsets based on the selected attribute.
- 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).

Example

- The example of a binary tree for predicting whether a person is fit or unfit providing various information like age, eating habits and exercise habits, is given below



Types of Decision Tree Algorithm

- There are two main types of Decision Tree algorithm −
- **Classification Tree** − A classification tree is used to classify data into different classes or categories. It works by splitting the data into subsets based on the values of the input features and assigning each subset to a different class.
- **Regression Tree** − A regression tree is used to predict numerical values or continuous variables. It works by splitting the data into subsets based on the values of the input features and assigning each subset a numerical value.

Why Use Decision Trees?

- Simple to understand and interpret (visualizable)
- Works with both numerical and categorical data
- No feature scaling required
- Can handle non-linear relationships

How to choose the best attribute at each node

- Two methods,
  - information gain and
  - Gini impurity,
- act as popular splitting criterion for decision tree models
- They help to evaluate the quality of each test condition and how well it will be able to classify samples into a class.

Entropy and information gain

- **Entropy is a concept that stems from information theory, which measures the impurity of the sample values.**
- It is defined with by the following formula, where:

$$\text{Entropy}(S) = -\sum_{c \in C} p(c)\log_2 p(c)$$

- S represents the data set that entropy is calculated
- c represents the classes in set, S
- p(c) represents the proportion of data points that belong to class c to the number of total data points in set, S
- Entropy values can fall between 0 and 1.
- If all samples in data set, S, belong to one class, then entropy will equal zero.
- If half of the samples are classified as one class and the other half are in another class, entropy will be at its highest at 1.
- In order to select the best feature to split on and find the optimal decision tree, the attribute with the smallest amount of entropy should be used.

- **Information gain represents the difference in entropy before and after a split on a given attribute.**
- The attribute with the highest information gain will produce the best split as it's doing the best job at classifying the training data according to its target classification.
- Information gain is usually represented with the following formula,

$$\text{Information Gain}(S,a) = \text{Entropy}(S) - \sum_{v \in values(a)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

where

- a represents a specific attribute or class label
- Entropy(S) is the entropy of dataset, S
- |Sv|/|S| represents the proportion of the values in Sv to the number of values in dataset, S.

EXAMPLE

- Imagine that we have the following arbitrary dataset:

Day	Outlook	Temp	Humidity	Wind	Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Weak	Yes

Step 1: Parent Entropy

- From the dataset (Yes = 4, No = 3):
- H(S)=−(4/7)log$f_0$2(4/7)−(3/7)log$f_0$2(3/7)=0.985 bits
-

Split on Humidity

- High Humidity (4 samples: 2 Yes, 2 No)
- H(High)=−(2/4)log$f_0$2(2/4)−(2/4)log$f_0$2(2/4)
- =1.0

- Normal Humidity (3 samples: 2 Yes, 1 No)
- H(Normal)=−(2/3)log2(2/3)−(1/3)log2(1/3)
- =0.918

Weighted Average Entropy

- Hsplit=(4/7)(1.0)+(3/7)(0.918)
- Hsplit=0.571+0.393=0.964

Information Gain

- IG(S,Humidity)=H(S)−Hsplit
- IG(S,Humidity)=0.985−0.964=0.021 bits
- So, with High Humidity = (2 Yes, 2 No) and Normal Humidity = (2 Yes, 1 No), the Information Gain for Humidity = 0.021 bits. This is very low, meaning Humidity is not a good splitting attribute for this dataset compared to others like Outlook.

Split on Outlook

- Sunny (2 samples: 0 Yes, 2 No) → Entropy = 0
- Overcast (2 samples: 2 Yes, 0 No) → Entropy = 0
- Rain (3 samples: 2 Yes, 1 No) → Entropy = 0.918
- Weighted entropy:
- Hsplit=(2/7)(0)+(2/7)(0)+(3/7)(0.918)=0.393
- Information Gain:
- IG(S,Outlook)=0.985−0.393=0.592

Temperature

- Hot (3 samples: 1 Yes, 2 No) → Entropy = 0.918
- Mild (2 samples: 2 Yes, 0 No) → Entropy = 0
- Cool (2 samples: 1 Yes, 1 No) → Entropy = 1.0
- Weighted entropy:
- Hsplit=(3/7)(0.918)+(2/7)(0)+(2/7)(1.0)=0.678
- Information Gain:
- IG(S,Temp)=0.985−0.678=0.307

Wind

- Weak (5 samples: 4 Yes, 1 No) → Entropy = 0.722
- Strong (2 samples: 0 Yes, 2 No) → Entropy = 0
- Weighted entropy:
- Hsplit=(5/7)(0.722)+(2/7)(0)=0.516
- Information Gain:
- IG(S,Wind)=0.985−0.516=0.469

Best Splitting Attribute

- Outlook → 0.592 bits
- Wind → 0.469 bits
- Temperature → 0.307 bits
- Humidity → 0.021 bits

- ✅ The best splitting attribute is Outlook, since it gives the highest Information Gain (0.592 bits).

Gini Impurity

- Gini impurity is the probability of incorrectly classifying random data point in the dataset if it were labeled based on the class distribution of the dataset.
- Similar to entropy, if set, S, is pure—i.e. belonging to one class) then, its impurity is zero. This is denoted by the following formula:

$$\text{Gini Impurity} = 1 - \sum_i (p_i)^2$$

Gini Impurity

- Yes = 4
- No = 3
- Gini(S)=1−((4/7)^2+(3/7)^2)
- =1−(0.327+0.184)=0.489
- So the Gini impurity = 0.489.

Advantages

- Easy to interpret: The Boolean logic and visual representations of decision trees make them easier to understand and consume. The hierarchical nature of a decision tree also makes it easy to see which attributes are most important, which isn't always clear with other algorithms, like neural networks.
- Little to no data preparation required: Decision trees have a number of characteristics, which make it more flexible than other classifiers. It can handle various data types—i.e. discrete or continuous values, and continuous values can be converted into categorical values through the use of thresholds. Additionally, it can also handle values with missing values, which can be problematic for other classifiers, like Naïve Bayes.
- More flexible: Decision trees can be leveraged for both classification and regression tasks, making it more flexible than some other algorithms. It's also insensitive to underlying relationships between attributes; this means that if two variables are highly correlated, the algorithm will only choose one of the features to split on.

Disadvantages

- Prone to overfitting: Complex decision trees tend to overfit and do not generalize well to new data. This scenario can be avoided through the processes of pre-pruning or post-pruning. Pre-pruning halts tree growth when there is insufficient

data while post-pruning removes subtrees with inadequate data after tree construction.

- High variance estimators: Small variations within data can produce a very different decision tree. Bagging, or the averaging of estimates, can be a method of reducing variance of decision trees. However, this approach is limited as it can lead to highly correlated predictors.
- More costly: Given that decision trees take a greedy search approach during construction, they can be more expensive to train compared to other algorithms.

DECISION TREE EXAMPLE

Day	Outlook	Temp	Humidity	Wind	Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Weak	Yes

Step 1: Parent Entropy

- Yes = 4, No = 3
- $H(S) = -(4/7)\log_2(4/7) - (3/7)\log_2(3/7) = 0.985$

Step 2: Information Gain

- Outlook → IG = 0.592 (best split)
- Wind → IG = 0.469
- Temperature → IG = 0.307
- Humidity → IG = 0.021
- So the root node = Outlook.

Step 3: First-Level Splits

- Sunny → No (pure branch)
- Overcast → Yes (pure branch)
- Rain → needs further split (2 Yes, 1 No)

Step 4: Expand Rain Branch

- Subset (Rain):
- Mild, High, Weak → Yes
- Cool, Normal, Weak → Yes

- Cool, Normal, Strong → No
- Best split = Wind (IG = 0.918).

Step 1: Entropy of Rain Subset

- $H(Rain) = -(2/3)\log_2(2/3) - (1/3)\log_2(1/3)$
- $H(Rain) = 0.918$ bits

Step 2: Candidate Splits
- 1. Temperature
- Mild (1 Yes) → Entropy = 0
- Cool (1 Yes, 1 No) → Entropy = 1.0
- Weighted entropy:
- $H_{split} = (1/3)(0) + (2/3)(1.0) = 0.667$
- $IG = 0.918 - 0.667 = 0.251$

- 2. Humidity
- High (1 Yes) → Entropy = 0
- Normal (1 Yes, 1 No) → Entropy = 1.0
- Weighted entropy:
- $H_{split} = 1/3(0) + 2/3(1.0) = 0.667$
- $IG = 0.918 - 0.667 = 0.251$

- 3. Wind
- Weak (2 Yes) → Entropy = 0
- Strong (1 No) → Entropy = 0
- Weighted entropy:
- $H_{split} = 2/3(0) + 1/3(0) = 0$
- $IG = 0.918 - 0 = 0.918$

Best Split

- Temperature → IG = 0.251
- Humidity → IG = 0.251
- Wind → IG = 0.918 (best)
- ✓ Therefore, the best split for the Rain subset is Wind, because it perfectly separates the data into pure branches.

FINAL DECISION TREE

Outlook
/ | \
Sunny Overcast Rain
No Yes Wind
/ \
Weak Strong
Yes No