Decision Trees

↳ supervised Machine Learning

A decision Tree is a flowchart like structure in which each internal node represents a test on a feature.

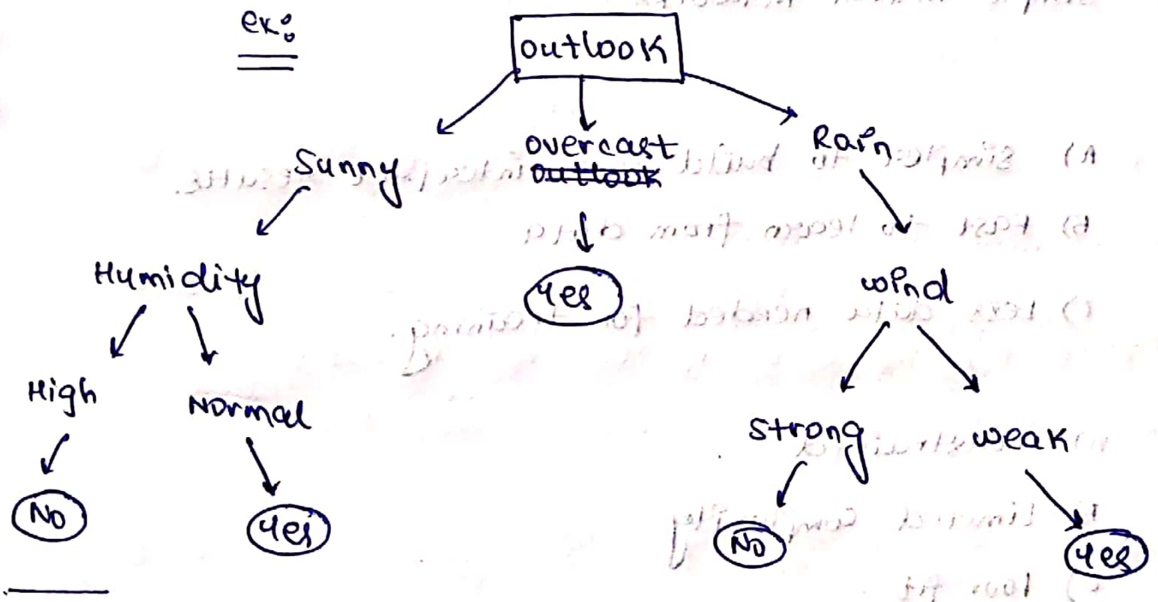
Each leaf node represents a class outcome label.

Branches represent conjunctions of features that lead to those class labels.

(Λ) final decision taken after computing all features.

The paths from root to leaf represent classification rules.

ex:



uses:

1. Data mining
2. ML
3. Statistics

Ⓐ Parametric Machine Learning Algo:

Assumptions can greatly simplify the learning process but can also limit what can be learned.

Algo. that simplify function to a known form → Parametric ML algo.

There Algo. involve 2 steps:

① select a form for the function.

② Learn the coeff. for the function from the training data.

ex:

A) Logistic regression

B) Linear regression

C) Naive Bayes

D) LDA

E) Perceptron (DL)

F) Simple Neural Networks

Benefits:

A) Simpler to build and interpret results.

B) Fast to learn from data

C) Less data needed for training.

Limitations:

A) Constrained

B) Limited Complexity

C) Poor fit.

———— x ———— x ————

Ⓑ Non-parametric ML Algo:

Algo. that don't make strong assumptions about the form of the mapping function are called non-parametric.

By not making assumptions, they are free to learn any functional form.

Non-parametric methods are good when you have a lot of data and no prior knowledge and you don't worry too much about choosing the right features.

Non-parametric methods seek to best fit the training data in constructing a mapping function whilst maintaing some ~~about~~ ability to generalize as well.

ex:

1) KNN

2) Decision Trees like CART

3) Support vector Machines

Benefits:

A) Flexibility: capability of fitting a large no. of functional forms.

B) no assumptions.

C) better performance.

Limitations:

A) Requires a lot of data.

B) Slower

C) Overfitting. (higher risk)

Decision Trees are a "non-parametric" supervised learning method used for both regression as well as classification.

Approach: while making a decision tree, at each node of the tree we ask different types of questions.

Based on the asked question, we will calculate the information corresponding to it.

Information gain is used to decide which feature to split on at each step in building the tree.

"Simplicity is best, so we want to keep our tree small. To do so, at each step we choose the split that results in the purest daughter nodes.

A commonly used measure of purity is "information".

This process keeps on repeating until the information gain is 0.

Gini Impurity:

→ Pure: In a selected sample of dataset, all data belongs to same class.

→ Impure: Data is a mixture of various classes.

Gini impurity is a measurement of the likelihood of an incorrect classification of a new instance, if that new instance were randomly classified acc. to distribution of class labels.

Note: If our dataset is pure then likelihood of incorrect class. = 0 else it will be high.

Steps for making a decision tree:

(1) Get list of rows (dataset) which are taken into consideration for constructing decision tree at each node.

(2) Calculate uncertainty or gini impurity of our dataset or how much our data is mixed up.

(3) Generate list of all questions that need to be asked.

(4) Partition rows into true rows and false rows based on the question asked.

(5) calculate info. gain based on gini impurity and partition of data from the previous step.

(6) update highest info gain based on each question asked.

(7) update best question based on info. gain.

(8) Divide the node on best question.

Repeat again!

Advantages:

A) Easy to use and understand

B) can handle both categorical and numerical data.

C) resistant to outliers, hence requires less data preprocessing.

Disadvantages:

A) Prone to overfitting.

B) req. performance measure.

C) careful parameter tuning.

D) can create biased trees.

Post-pruning

Pre-pruning → Early stopping.

Entropy: measures the purity of split.

H(S)

→ we want leaf node to be encountered very quickly.

f_1, f_2, f_3

i/p features

o/p

yes No

9 yes / 5 No

f_1

not pure: → f_2 3 yes / 2 No

pure → 3 yes 2 No

$H(S) = -P_{(+)} \cdot \log_2(P_+) - P_{(-)} \cdot \log_2(P_-)$

for one node.

$\Rightarrow P_+ = \%$ of +ve class

$\Rightarrow P_- = \%$ of -ve class

$\Rightarrow S =$ subset of training example.

f1

f2 — 3Y | 2No

f3 — 3Y | 3No

$H(S) = -\frac{3}{5} \times \log_2\left(\frac{3}{5}\right) - \frac{2}{5} \times \log_2\left(\frac{2}{5}\right)$

$= +0.78$ bits

3Yes | 3No → completely impure.

→ worst split

$H(S) = 1$

[0,1]

$H(S) =$ lesser $\Rightarrow$ better.

information gain → go from top to leaf (combines all the entropy together)

when $H(S) = 0 \Rightarrow$ stop splitting

pure split

we have reached the leaf node.

we need overall gain:

$$\boxed{Gain(S, A) = H(S) - \sum_{v \in Val} \frac{|S_v|}{|S|} \cdot H(S_v)}$$

subset after splitting

Ex

9Y | 5N

f1 — $H(f1) = 0.94 \rightarrow H(S)$

6Y | 2N — f2 → $H(S_v)$

3Y | 3N — f3 → $H(S_v)$

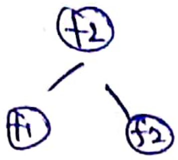
$H(f2) = 0.81$

$H(f3) = 1$

$Gain(S, f1) = H(S) - \frac{8}{14} \times H(f2) - \frac{6}{14} \times H(f3)$

$= 0.91 - \frac{8}{14} \times 0.81 - \frac{6}{14} \times 1 = 0.049$

Scanned by CamScanner

now go for 

→ The one having higher info. gain that way of splitting up decision tree will be considered.