FLATIRON SCHOOL

Constructing Hierarchies with Decision Tree Algorithms.

hosted by KASH

CRITICAL QUESTION

How can we construct and apply algorithms that leverage decision-making criteria and information reduction in order to learn patterns in data?

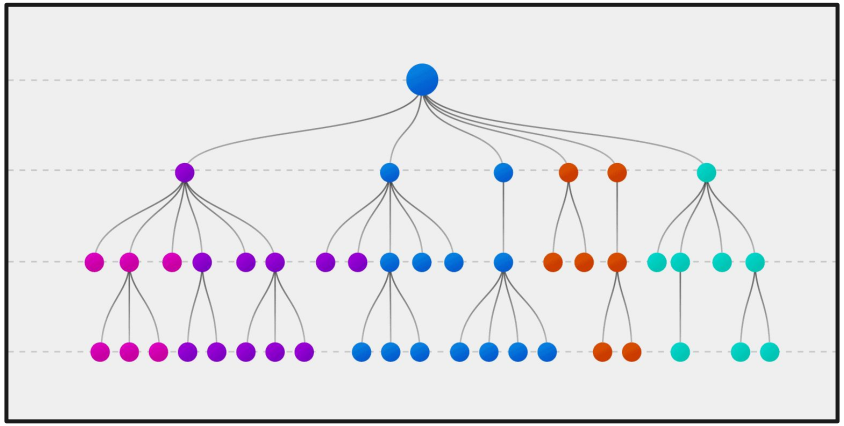
CRITICAL QUESTION

How can we utilize CART and ID3 algorithms to classify new data?

WRITTEN IDEATION

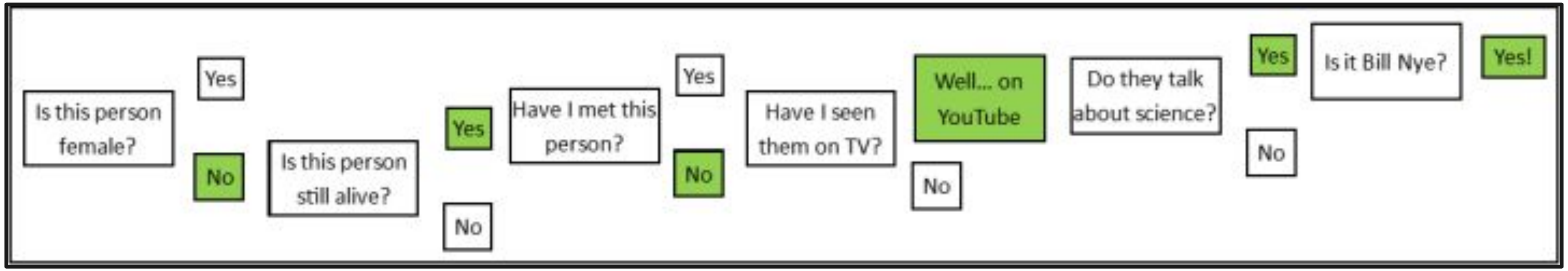
HOW COULD THE CONCEPT OF **BINARY TREES** BE APPLIED INTO THE **MACHINE LEARNING SPACE**?

AN INTRODUCTION TO DECISION TREE ALGORITHMS



Decision tree algorithms are among some of the simplest yet most powerful of standard classification algorithms available in our machine learning toolkit – they are capable of leveraging **tree-like data structure representations** in order to **“capture” information for each feature** in a dataset.

AN ILLUSTRATIVE APPROACH TO DECISION TREES



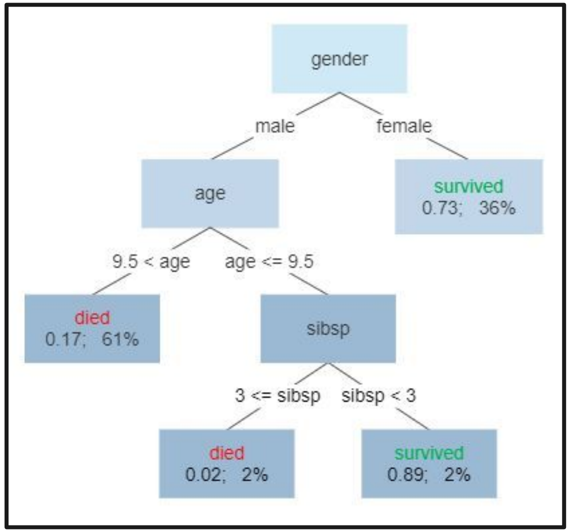
Consider playing a game of **Twenty Questions**, where you're allowed to ask **binary decision questions** in an attempt to discern the **"identity" of some unknown person, object, or thing**.

Decision trees work very similarly: they examine a dataset and create **"decision junctions" or "splits"** that serve as **computational boolean "question-and-answers"** in an attempt to discern the **identity of some datum**.

A CONCEPTUAL APPROACH TO DECISION TREES

Decision trees start at the **“root”**, which comprises our entire training dataset – from there, they **iterate through specific features and create decision splits** that are designed to be as optimal as possible in order to group data into two-or-more categories.

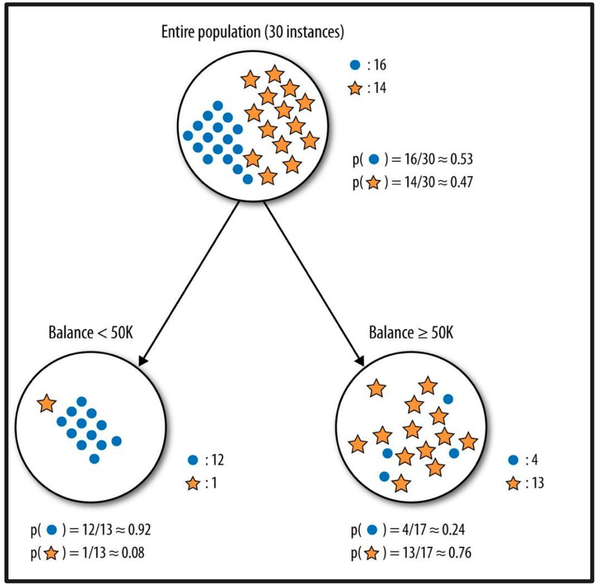
The goal with each of these splits is to **reduce error and minimize disorder** as much as possible, such that by the type a full tree is produced, **all data is most effectively segmented into appropriate nodes on the tree**.



ENTROPY: STATISTICAL SORCERY FOR DECISION TREES

Rather than literally asking questions, decision trees leverage **probabilistic processes** in order to produce the most effective decision junctions per feature – these processes are heavily reliant on a computational measure of information called **“entropy”**, which effectively captures the **uncertainty and disorder of our current data**.

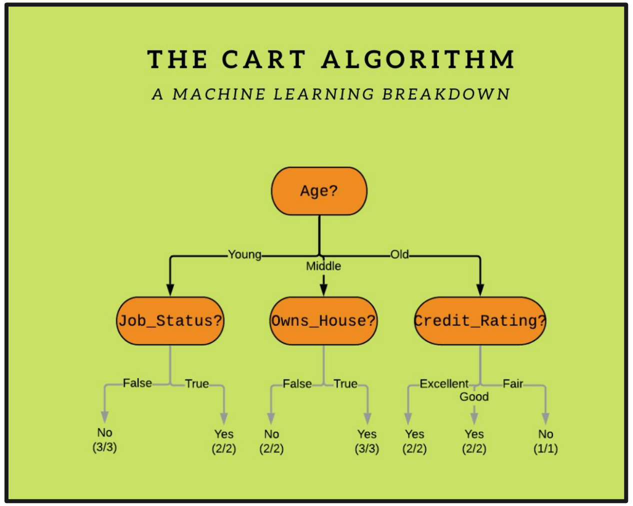
All decision trees attempt to **minimize entropy** at every step, which in turn **reduces error and improves accuracy** with each branch created across our tree.



CODING WALKTHROUGH

Applying Decision Tree Classifiers Across Forest Cover Data

CARTs: CLASSIFICATION AND REGRESSION TREES



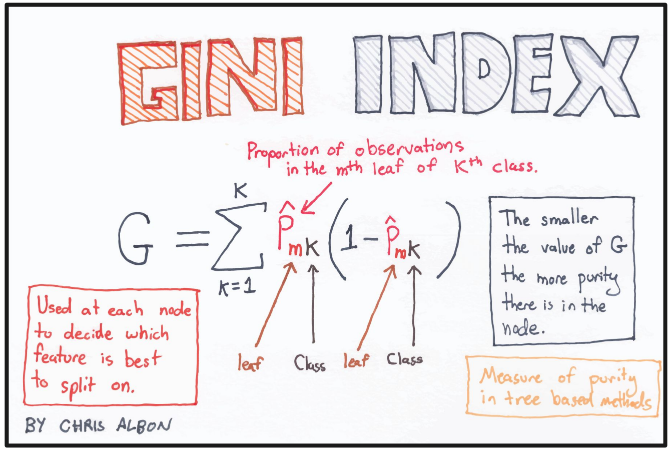
CART models are among the gold standard for decision tree construction and utilization in machine learning – they comprise a **family of decision tree models capable of classification and regression** purposes, allowing for a wide range of applicability for predictive analytics.

Most deployed and pre-designed implementations of decision tree classifiers make use of CART methodology due to its simplicity and relative effectiveness.

A TECHNICAL APPROACH TO UNDERSTANDING CARTs

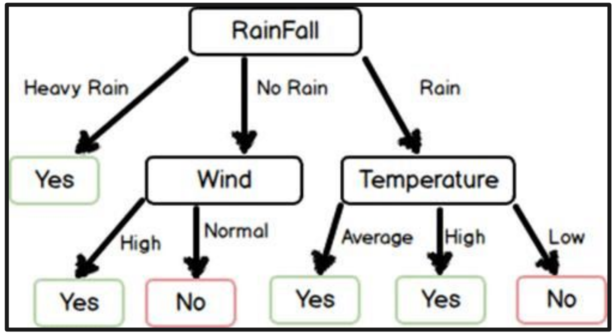
CART algorithms rely on achieving splits across data through the **minimization of Gini impurity** as its entropy measure – the precise mathematical construct is represented as **one minus the sum of probabilities of each class occurring over all events of the current slice/subset**.

The Gini index is a **highly simple and deterministic** method of reducing impurity across a dataset, which allows for the flexibility of CART algorithms in practice.



GINI INDEX
Proportion of observations in the mth leaf of Kth class.
$G = \sum_{k=1}^{K} \hat{p}_{mk}(1-\hat{p}_{mk})$
leaf
Class
leaf
Class
Used at each node to decide which feature is best to split on.
The smaller the value of G the more purity there is in the node.
Measure of purity in tree based methods
BY CHRIS ALBON

ID3s: THE ITERATIVE DICHOTOMISER 3 ALGORITHM



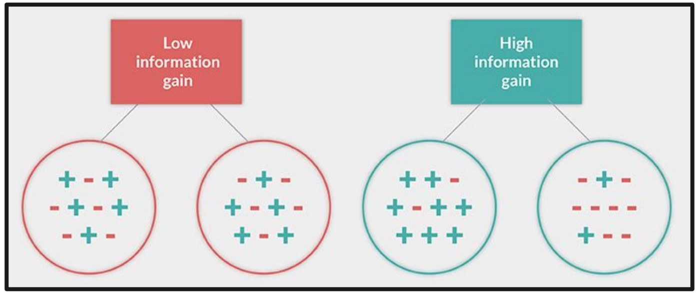
One major disadvantage of CART algorithms is that it’s heavily restricted to performing binary splits on data, but what if we want to **produce trees with multiple decision junctions and answers**?

Thus, the ID3 algorithm was created: a competing decision tree classifier model that utilizes a **different entropy minimization function** to allow handling of a **variable number of decision junctions per node**.

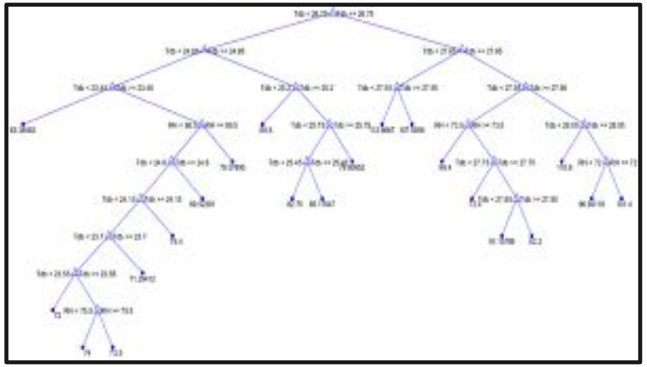
A TECHNICAL APPROACH TO UNDERSTANDING ID3s

Specifically, the precise entropy minimization function that ID3 algorithms use – instead of the classic Gini index – is referred to as **"information gain"**, which calculates **reduction of entropy and measures how well a specific feature separates/classifies target classes**.

The feature with the **highest information gain** (minimized entropy) is selected as the **best one for splitting and segmentation**.



A CONCEPTUAL DISTINCTION BETWEEN CARTs AND ID3s



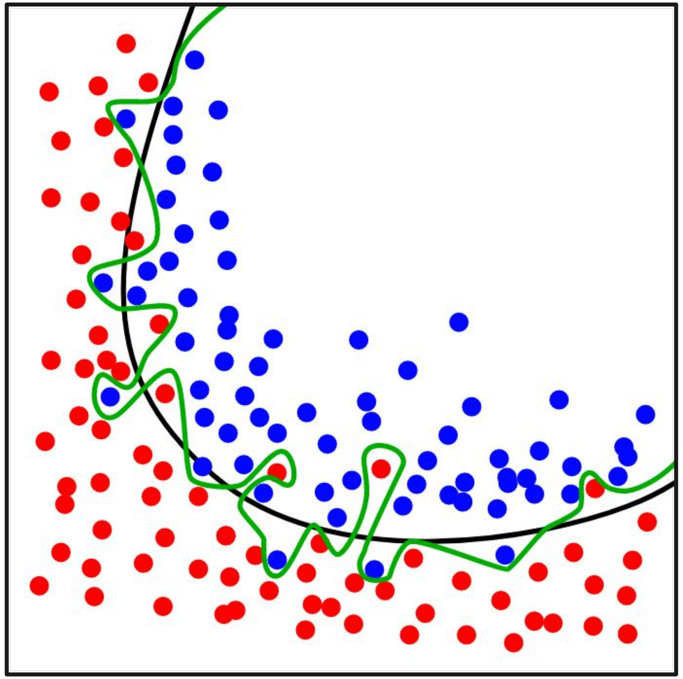
Rather than being distinct algorithms, CARTs and ID3s are heavily similar and often regarded as the **same type of model with different design constraints** in terms of **how the model creates decision junctions** and produces tree-like data structures across a target dataset.

Usually, **CARTs attempt to create deeper trees that track more features**, while **ID3s attempt to create more shallow trees with more decision junctions per feature.**

OVERFITNESS: EVERY DATA SCIENTIST'S WORST NIGHTMARE

Despite the advantages in simplicity and flexibility that decision trees provide, decision trees (along with most (if not all) predictive algorithms) are plagued with one major issue to mitigate: **the risk of overfitness**.

Overfitness especially occurs in more deterministic models: it indicates when **the model has learned variance and micropatterns across training data too well** and is now **unable to effectively extrapolate to patterns across new testing data.**



RECOMMENDED RESOURCE

Multiple Methods of Visualizing Decision Tree Fitness for Classification and Regression Tasks

THANK YOU FOR YOUR TIME!

ADVANCED LEARNING

Building an ID3 Algorithm from Scratch in Python