Decision Trees

(CART: Classification And Regression Trees)

Carson Trego

Want to follow along?

The entire code and dataset used is posted on github!

https://github.com/CarsonConjectures/treePres

NOT A Random Forest

While there are many similarities in the underlying methods that decision trees, random forests, XGBoost, Catboost, we will **not** be covering **bagged** and **boosted** methods, as they are covered in other presentations

Bagging: Generated by sampling with replacement of subsets of the data. Unweighted samples of several trees voting

Boosting: Weighted samples based on performance

Decision Trees (One Tree):

Classification Trees

Regression Trees

Bootstrap Aggregated (Bagged):

Bagged Decision Tree (Bootstrapped data subset AND limited features)

Random forests (Bootstrapped data subset AND limited features)

Boosting

Adaptive Boosting (AdaBoost)

eXtreme Gradient Boosting (XGBoost)),

At A Glance: Primary Concepts

White Box: Highly interpretable artificial intelligence

Structured and Unstructured Data

Splitting Criterion

Greedy and optimal

Categorical and quantitative data in both the (features) explanatory and (target) response

Overfitting

Cross Validation

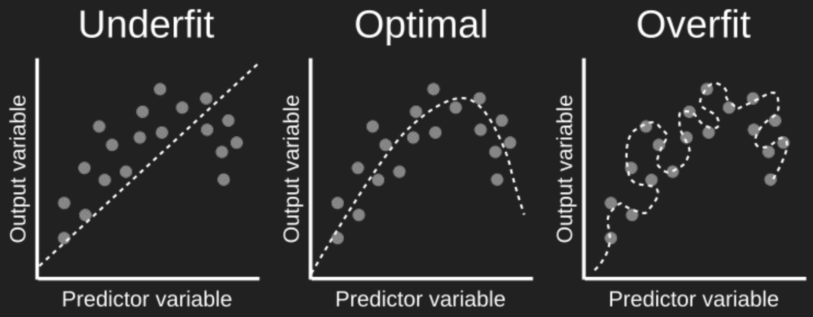
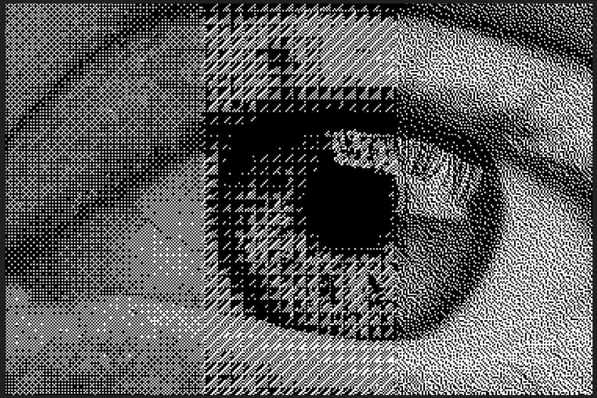


Table 1

	x1	x2
row1	a	6
row2	b	5
row3	c	4
row4	d	3
row5	e	2
row6	f	1



Structured Data

Information that is highly organized and easily decipherable by machine learning algorithms

Table 1

	x1	x2
row1	a	6
row2	b	5
row3	c	4
row4	d	3
row5	e	2
row6	f	1

Stored in:

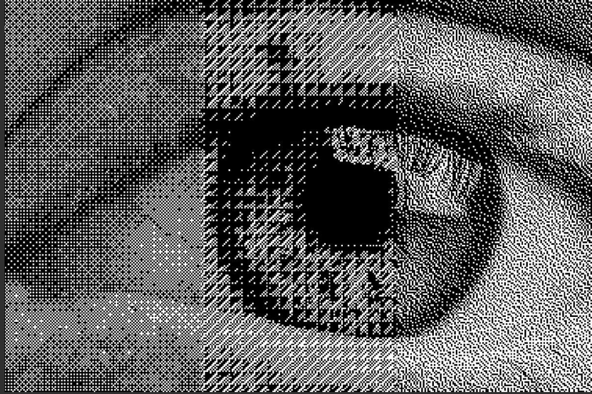
- Excel Files
- Data Frames
- Relational Databases

Examples:

- Usernames and passwords
- Product purchased and price
- Date and miles traveled

UnStructured Data

Information without simple organization structure, cannot be processed by simple means



Stored in:

- Image files
- Video files
- Audio files

Examples:

- Sending a paragraph to ChatGPT
- Using an AI to identify photos of plants
- Converting speech to text

Tree Terminology

The terminology for decision trees borrows terminology from biological trees

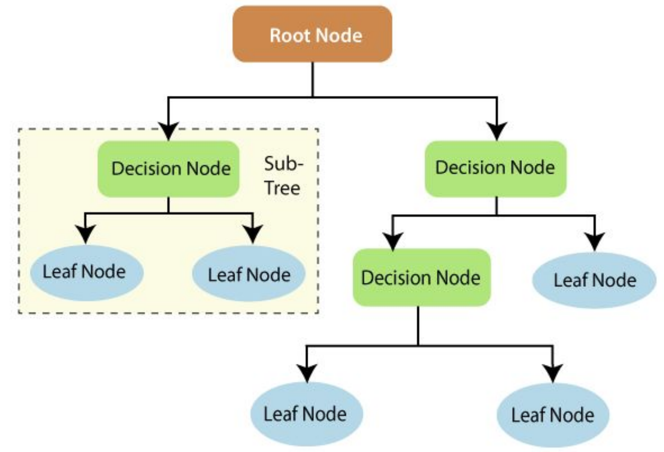
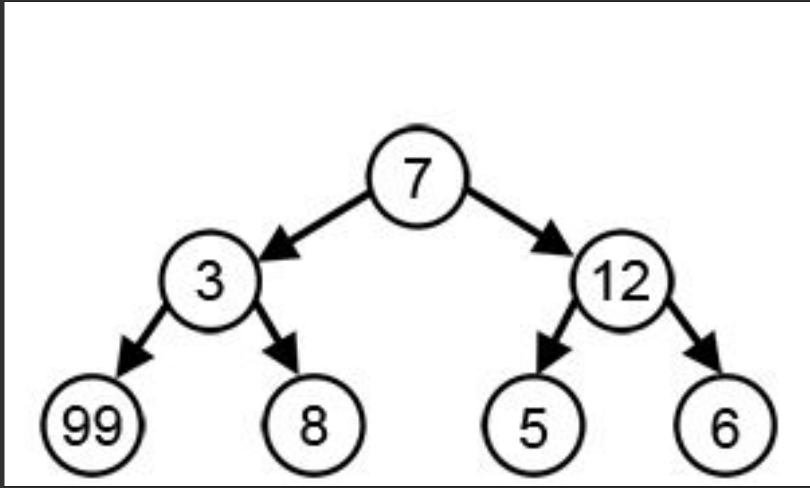


Fig-1: This is how decision tree looks.

Greedy Algorithm

Choosing the best option at each step in the short term, not considering the long term optimal solution



Supervised

Training inputs are paired with **desired** outputs

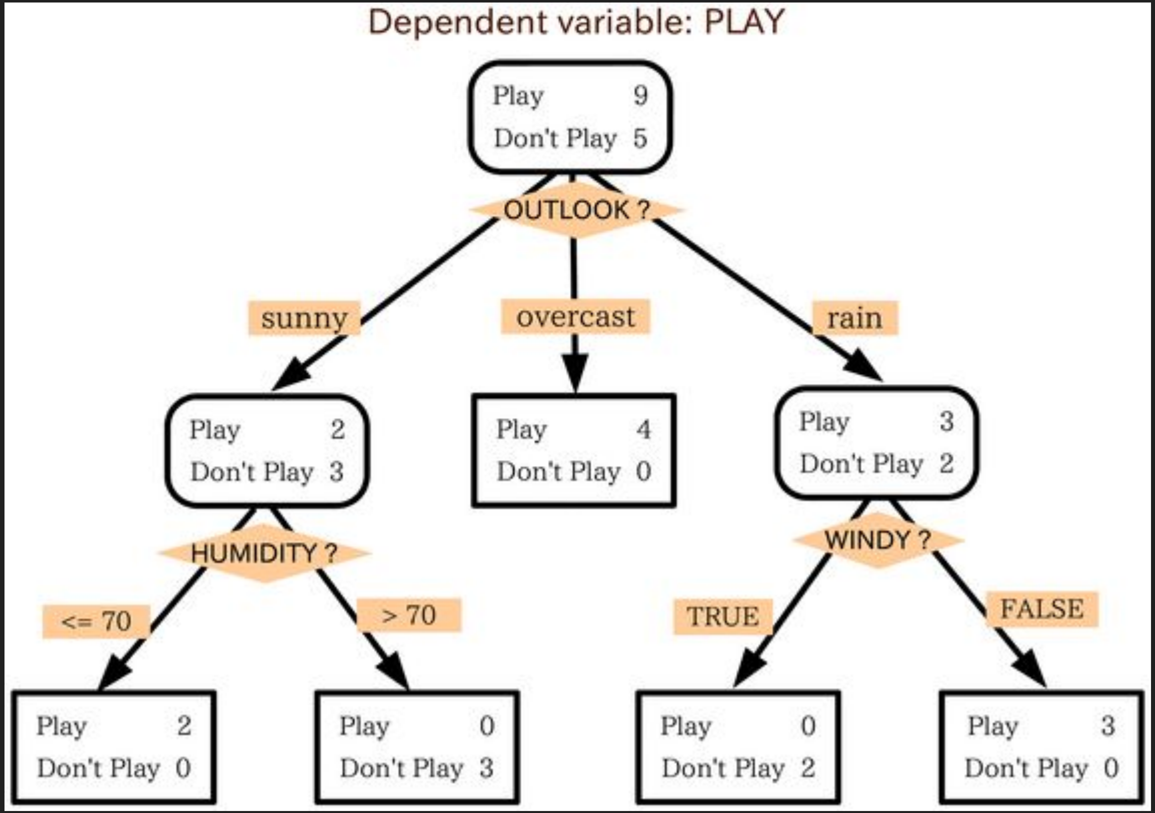
	FEATURE (LEGS)	TARGET (ANIMAL)
TRAINING DATA	8	SPIDER
CORRECT INFERENCE	8	SPIDER
INCORRECT INFERENCE	8	HUMAN

Flowcharts are intuitive

With little training, anybody should be able to understand a flowchart.

When simple programs make decisions, flowcharts can be used to communicate with clients HOW your program made certain decisions.

As such, flowcharts can be a powerful tool for communication and transparency

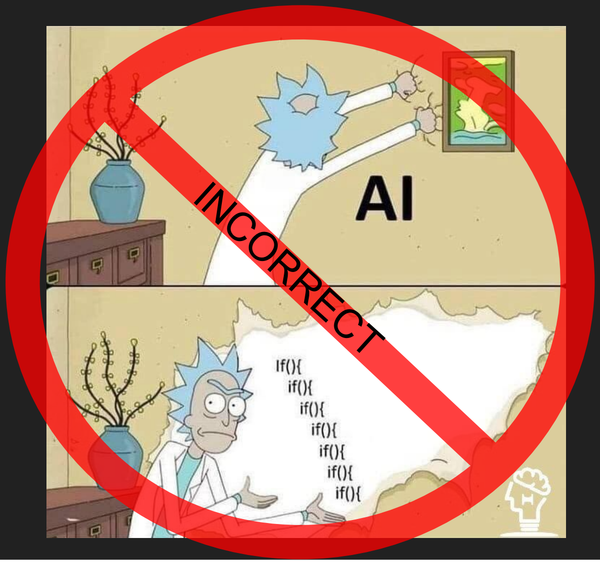


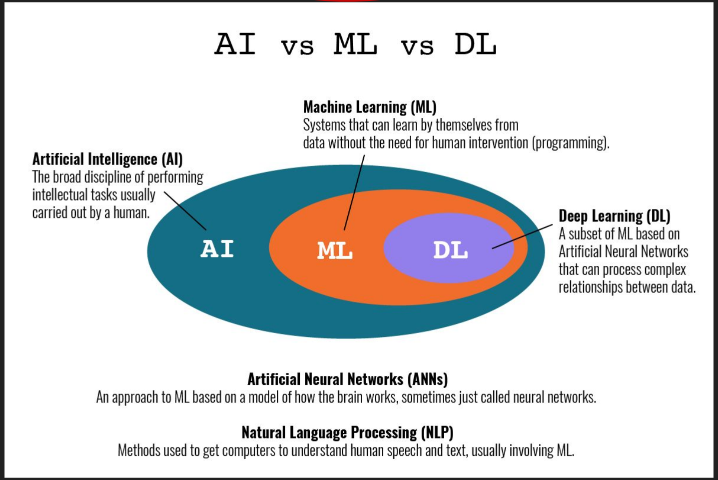
“AI Is Just If Statements”

This is a very common joke, and while many products being described as AI are based on systems that are essentially ust *if statements*, this is not true for all AI, as there are two problems:

Training: Machine learning methods often lengthy mathematical operations to learn and adjust itself from data, whereas if statements tend to be constructed by humans

Execution: In addition to training, lengthy mathematical operations or often required. Even the smallest object detection model produced in Ultralytics’ V8 line requires 8.7 BILLION operations PER FRAME

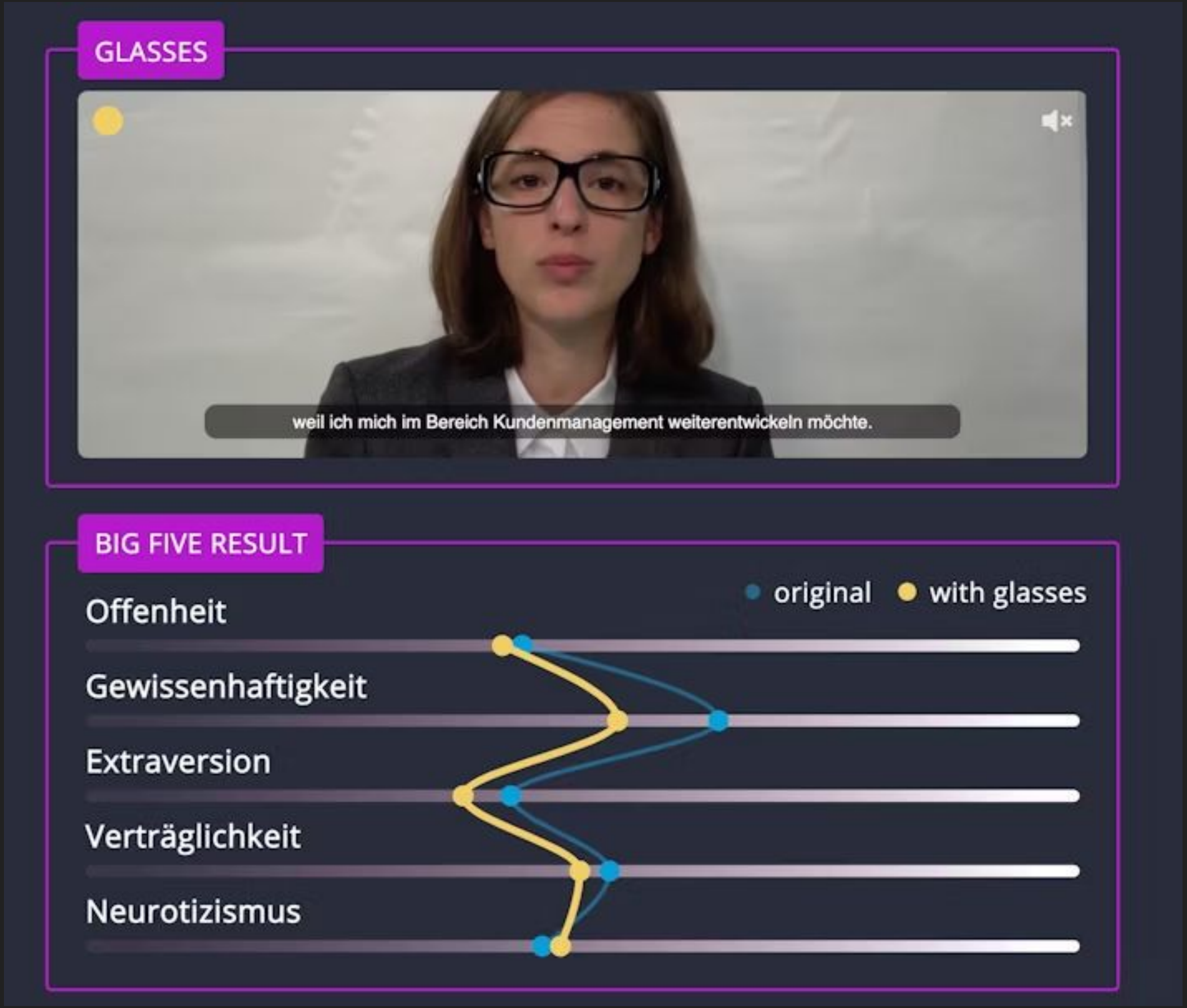




The Black Box Issue

The aforementioned execution complexity is a major issue within AI, as the conclusions a model makes is often so complex that no human operator could be reasonably expected to reverse engineer why a decision was made by an AI

In situations like this, we tend to use the model itself to run experiments and report on those results.



More Than If Statements - A Bad Thing?

If statements are intuitive and predictable. An attentive reader can inspect code and predict how a set of if statements would respond to input data.

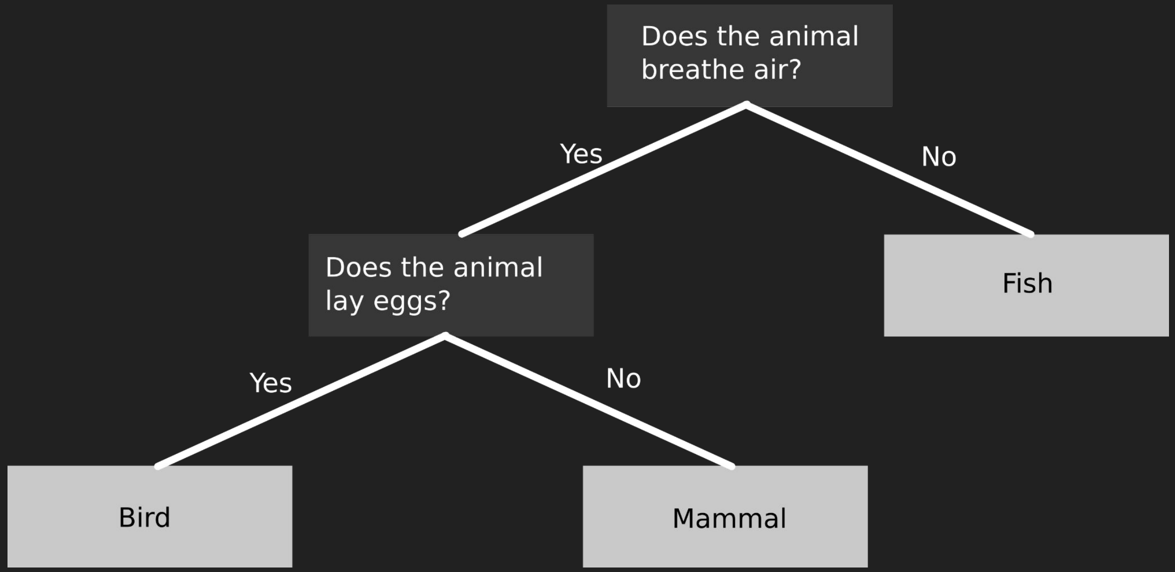
For much of AI, this is not the case. AI has become so complex that no person can review the code and decide if it is safe to deploy.

Some situations are far too complex to condense into if statements, but for situations where a relatively small amount of structured data is used to make a decision, it would be nice if an AI could be made of simple if statements…

Such Technology Does Exist … (kinda)

Decision trees represent a combination of machine learning and interpretability. Rather than relying on lengthy math to perform inference, decision trees create a single flowchart with the data it they are given.

This is not without limits and warnings, but there are many applications where decision tree methods can be used

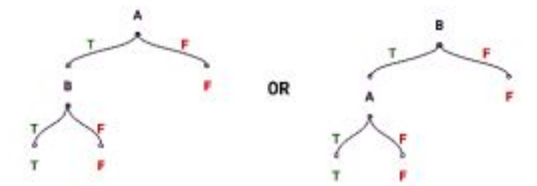


Logic

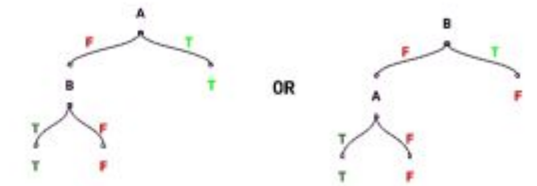
Decision trees can be used to be an alternative representation of a truth table.

This may seem like unrelated math, but it shows that an algorithm that can generate a decision tree in response to inputs and outputs could theoretically be used to understand when a logical relationship is present

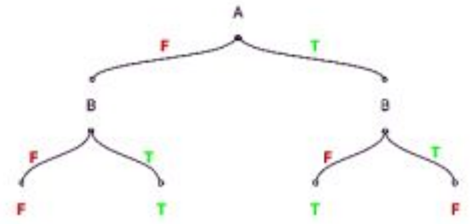
A	B	A AND B
F	F	F
F	T	F
T	F	F
T	T	T



A	B	A OR B
F	F	F
F	T	T
T	F	T
T	T	T



A	B	A XOR B
F	F	F
F	T	T
T	F	T
T	T	F



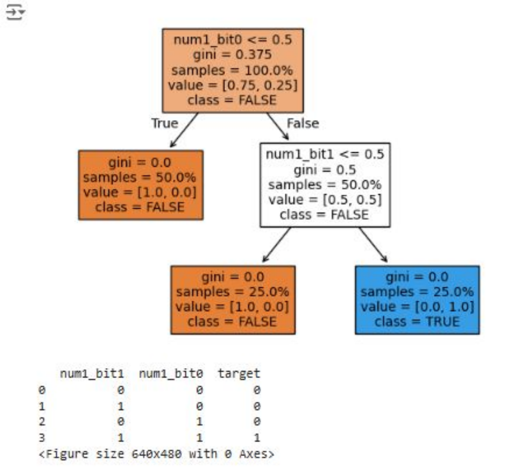
Reverse Engineered Logic

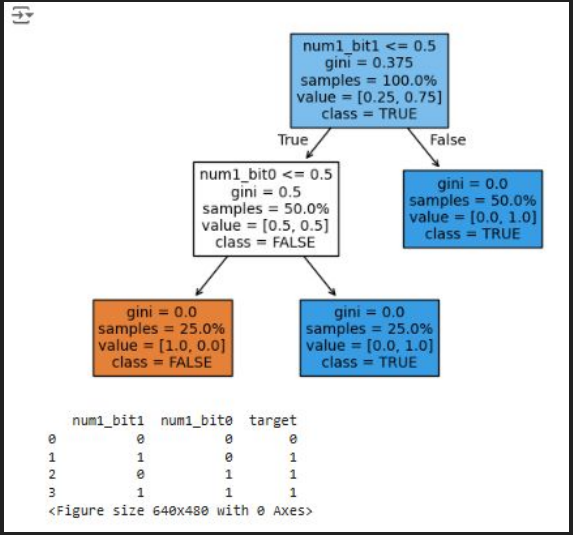
As shown, the decision tree algorithm can reverse engineer logical operations given only data

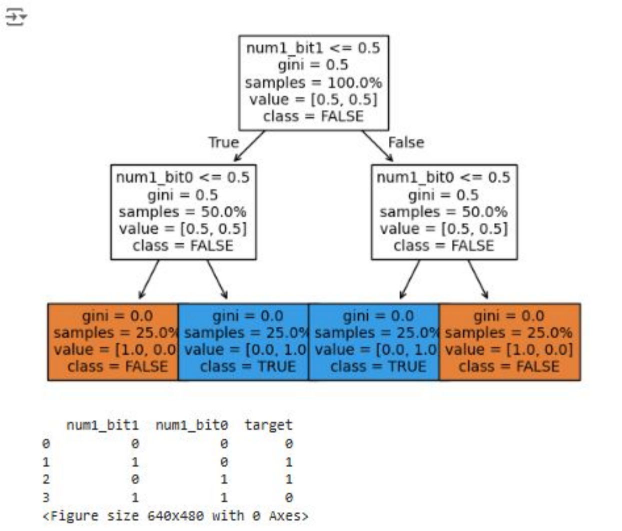
AND (top left)

OR (right)

XOR (bottom left)

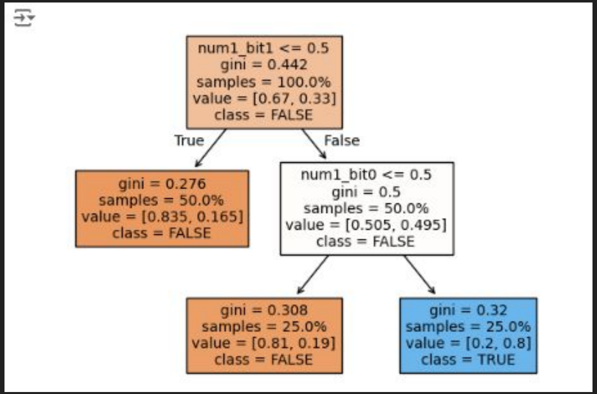


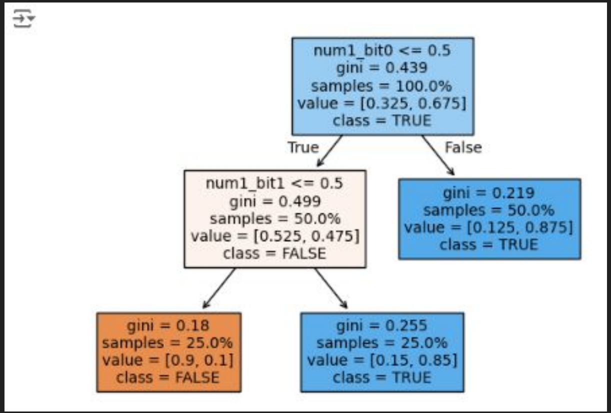


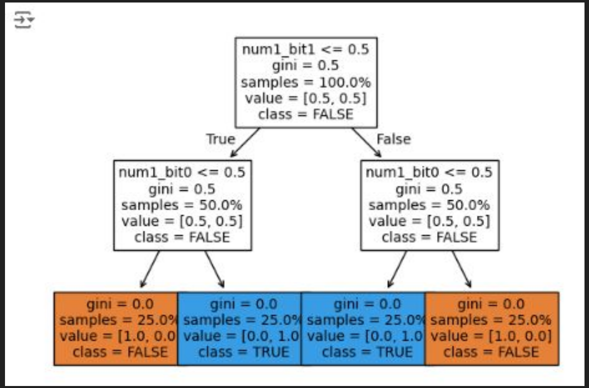


Reverse Engineered Logic (Non deterministic)

Even when a bit of randomness is added, the tree can still find the underlying system (Note, reducing complexity of the tree required **post-pruning**, more on that later)







AND (top left)

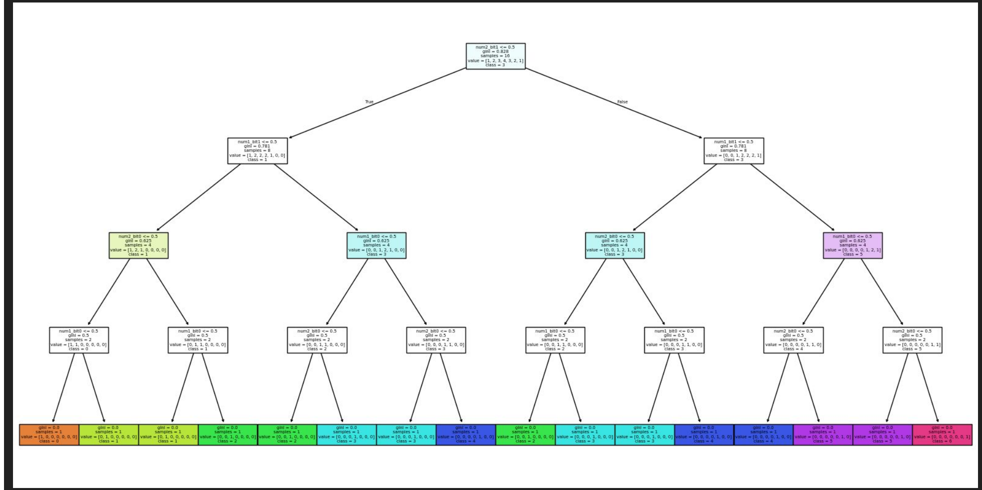
OR (right)

XOR (bottom left)

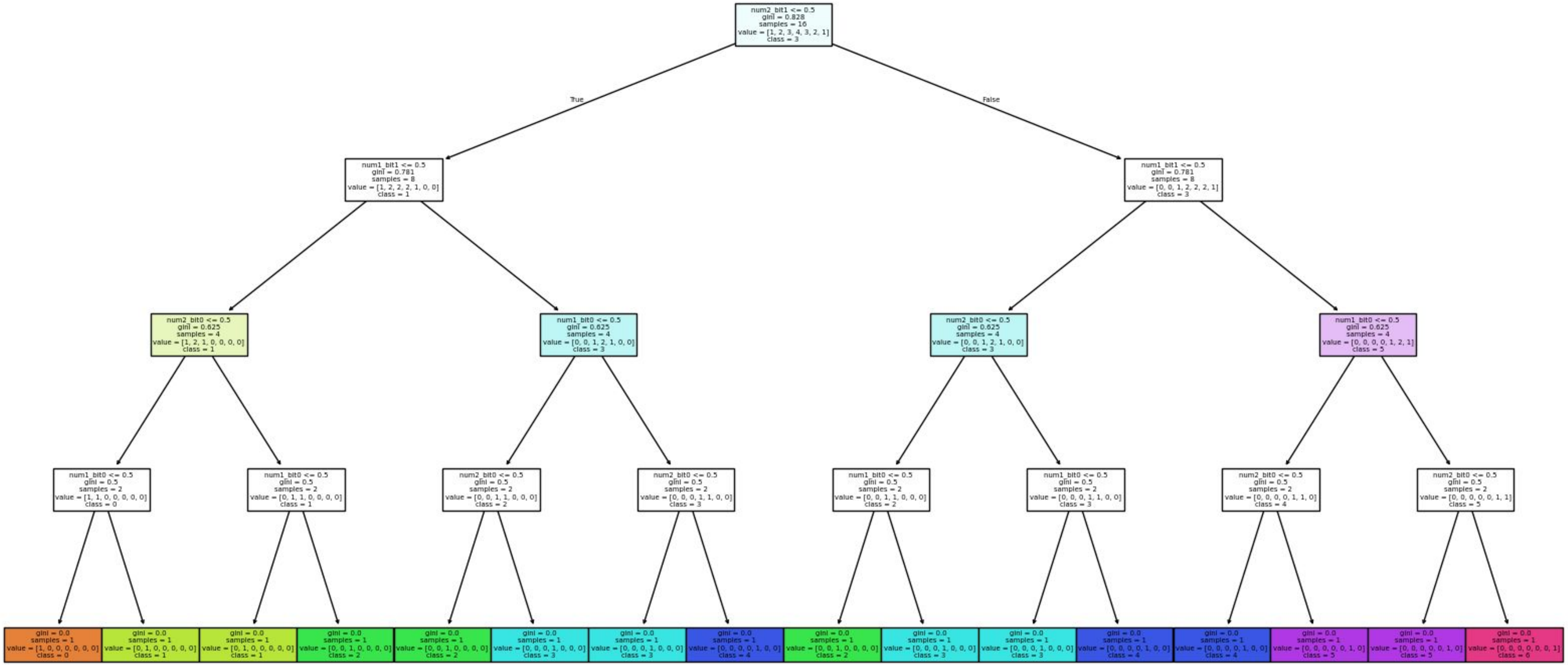
2 Bit + 2 Bit Addition

A decision tree algorithm was given a set of bits, not knowing the order, that they were binary, or that the target was the decimal sum of the two, here was the result

	num1_bit1	num1_bit0	num2_bit1	num2_bit0	target
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	1
5	0	1	0	1	2
6	0	1	1	0	3
7	0	1	1	1	4
8	1	0	0	0	2
9	1	0	0	1	3
10	1	0	1	0	4
11	1	0	1	1	5
12	1	1	0	0	3
13	1	1	0	1	4
14	1	1	1	0	5
15	1	1	1	1	6



2 Bit + 2 Bit Addition (Zoom In)



Train Tree

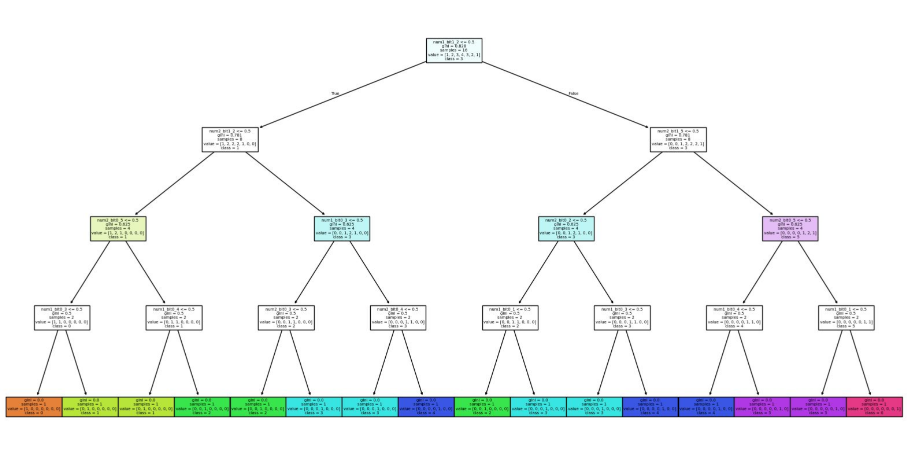
Using this method, we can make a “calculator” from just trains

Compression

In this example, we have redundant information, but the algorithm still provides a 100% solution in 4 steps/4 questions

Note: this isn't reducing the number of columns needed, because the algorithm isn't prioritizing using previously used columns, although that could probably be accomplished by other methods.

So decision trees can compress the amount of questions asked, but not necessarily allow you to reduce the amount of information you need in total



In practical terms, a decision tree can help a worker spend less time filling out questions,but they would still need the whole information book in front of them

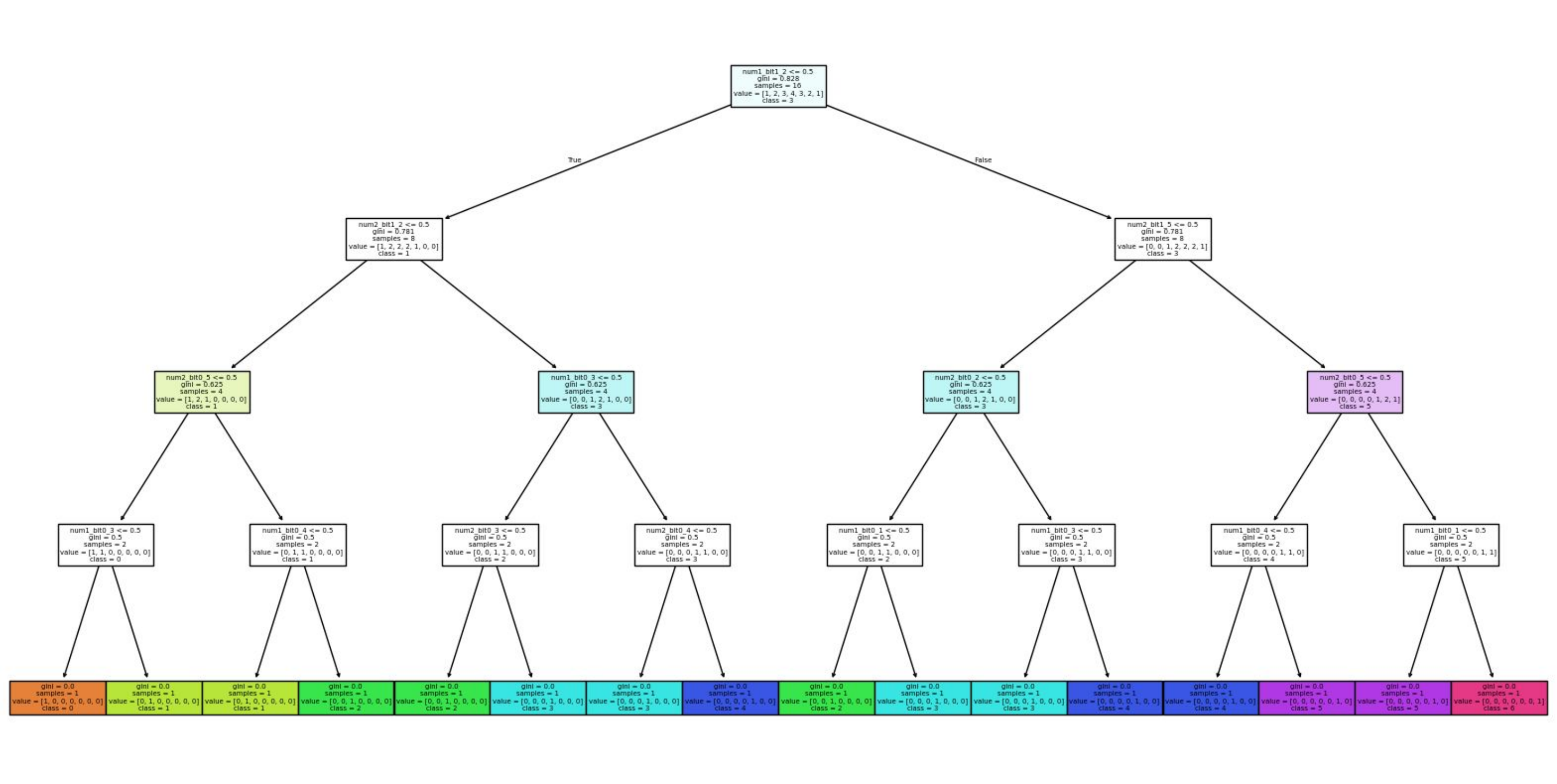
	num1_bit1_1	num1_bit0_1	num2_bit1_1	num2_bit0_1	num1_bit1_2	\
0	0	0	0	0	0	
1	0	0	0	1	0	
2	0	0	1	0	0	
3	0	0	1	1	0	
4	0	1	0	0	0	
5	0	1	0	1	0	
6	0	1	1	0	0	
7	0	1	1	1	0	
8	1	0	0	0	1	
9	1	0	0	1	1	
10	1	0	1	0	1	
11	1	0	1	1	1	
12	1	1	0	0	1	
13	1	1	0	1	1	
14	1	1	1	0	1	
15	1	1	1	1	1	

	num1_bit0_2	num2_bit1_2	num2_bit0_2	num1_bit1_3	num1_bit0_3	...	\
0	0	0	0	0	0	...	
1	0	0	1	0	0	...	
2	0	1	0	0	0	...	
3	0	1	1	0	0	...	
4	1	0	0	0	1	...	
5	1	0	1	0	1	...	
6	1	1	0	0	1	...	
7	1	1	1	0	1	...	
8	0	0	0	1	0	...	
9	0	0	1	1	0	...	
10	0	1	0	1	0	...	
11	0	1	1	1	0	...	
12	1	0	0	1	1	...	
13	1	0	1	1	1	...	
14	1	1	0	1	1	...	
15	1	1	1	1	1	...	

	num2_bit0_3	num1_bit1_4	num1_bit0_4	num2_bit1_4	num2_bit0_4	\
0	0	0	0	0	0	
1	1	0	0	0	1	
2	0	0	0	1	0	
3	1	0	0	1	1	
4	0	0	1	0	0	
5	1	0	1	0	1	
6	0	0	1	1	0	
7	1	0	1	1	1	
8	0	1	0	0	0	
9	1	1	0	0	1	
10	0	1	0	1	0	
11	1	1	0	1	1	
12	0	1	1	0	0	
13	1	1	1	0	1	
14	0	1	1	1	0	
15	1	1	1	1	1	

	num1_bit1_5	num1_bit0_5	num2_bit1_5	num2_bit0_5	target
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	1
5	0	1	0	1	2
6	0	1	1	0	3
7	0	1	1	1	4
8	1	0	0	0	2
9	1	0	0	1	3
10	1	0	1	0	4
11	1	0	1	1	5
12	1	1	0	0	3
13	1	1	0	1	4
14	1	1	1	0	5
15	1	1	1	1	6 --

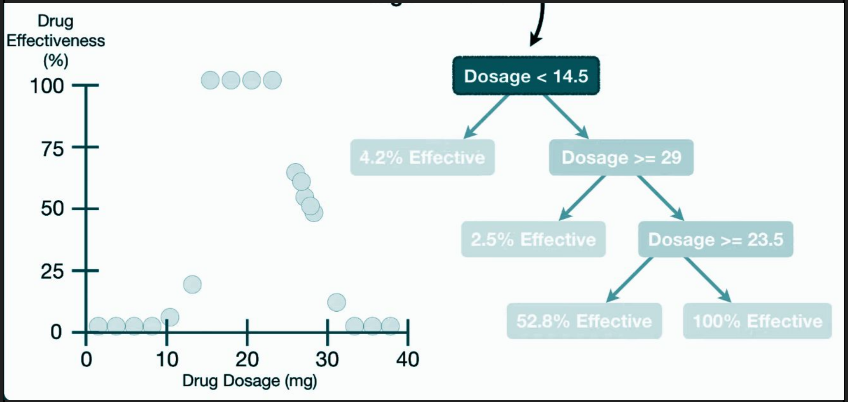
Redundant Tree



Regression vs Classification

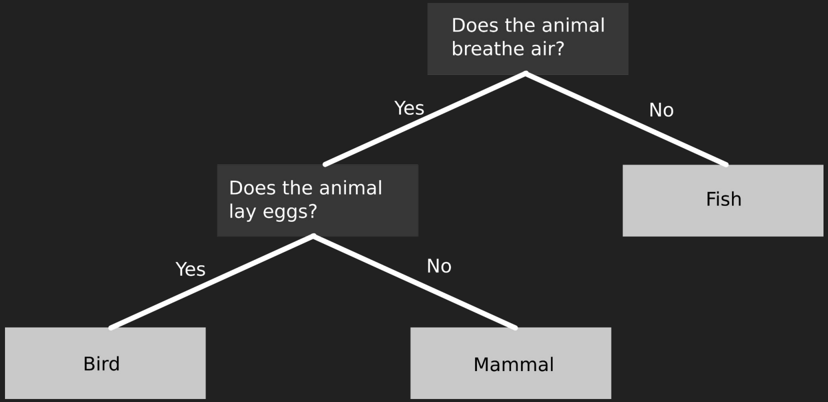
Regression Tree

- Uses a quantitative target (response)
- Produces discrete prediction groups



Classification Tree

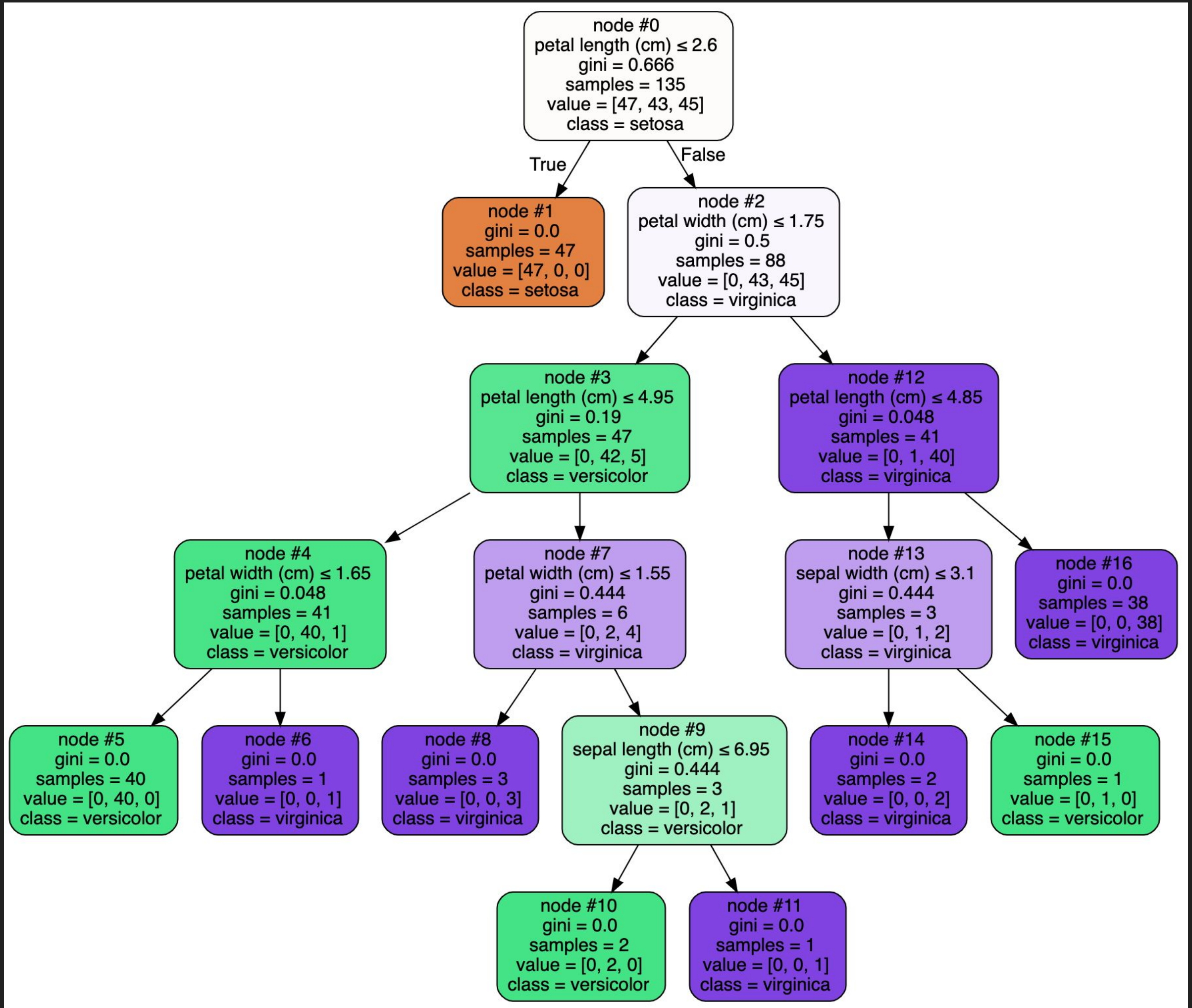
- Uses a categorical target (response)
- Produces discrete prediction groups



How The Machine Learning Works (1)

The decision tree needs to take every group produced and split it somehow.

The computer has no context of what the data is or how it “should” be organized, so an algorithm is needed to decide where and how to make splits.



How The Machine Learning Works (2)

There are many ways to build a decision tree, but we will talk about the *greedy case* first. There are two primary methods to create a decision tree in the *greedy case*: Gini index and Entropy. Both measure impurity, non-homogeneity.

Information gain describes the expected entropy difference after a split in the tree is made

$$Gini = 1 - \sum_{i=1}^{n} p^2(c_i)$$

$$Entropy = \sum_{i=1}^{n} -p(c_i)log_2(p(c_i))$$

Entropy

Entropy measures the impurity or uncertainty present in the data.

$$H(S) = -\sum_{i=1}^{N} p_i \log_2 p_i$$

where:

- *S – set of all instances in the dataset*
- *N – number of distinct class values*
- *pi – event probability*

Information Gain (IG)

IG indicates how much "information" a particular feature/ variable gives us about the final outcome.

$$Gain(A,S) = H(S) - \sum_{j=1}^{v} \frac{|S_j|}{|S|} \cdot H(S_j) = H(S) - H(A,S)$$

where:

H(S) – entropy of the whole dataset S

- ***|Sj| – number of instance with j value of an attribute A***
- ***|S| – total number of instances in dataset S***
- ***v – set of distinct values of an attribute A***
- ***H(Sj) – entropy of subset of instances for attribute A***
- ***H(A, S) – entropy of an attribute A***

Intuition of Gini Impurity

In this simple example from *StatQuest* , we can see two criteria with TRUE and FALSE answers: “Loves Popcorn” and “Loves Soda”

A split is tried with both criteria, and the individual gini impurities are calculated. The total gini impurities of the splitting criteria are then calculated by adding both impurities weighted by their relative size. We use the gini impunity to select how we split the data (lower is better)

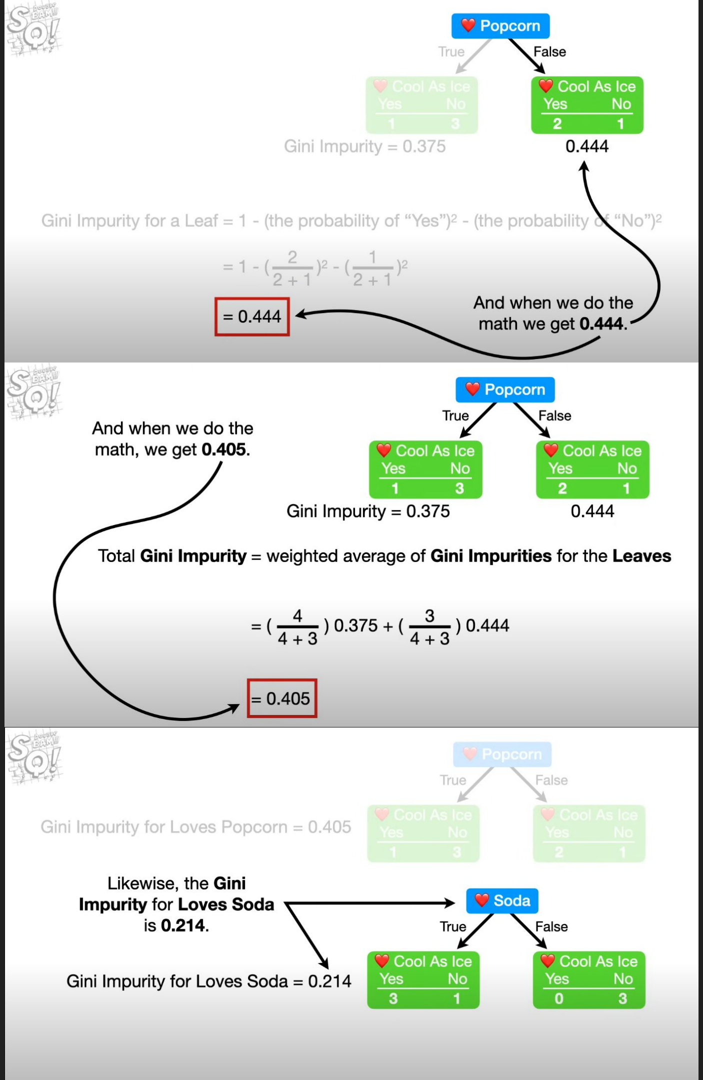
In the best possible case, the data would be split into two groups, with each group being homogenous

Leaf_1 = 1-1.0**2-0.0**2 = 0.0

Leaf_2 = 1-1.0**2-0.0**2 = 0.0

Total   = 0.5*0.0+0.5*0.0=0

We “like” to cut the data into big homogenous slices. Logically, that helps you make definitive classifications with few steps.

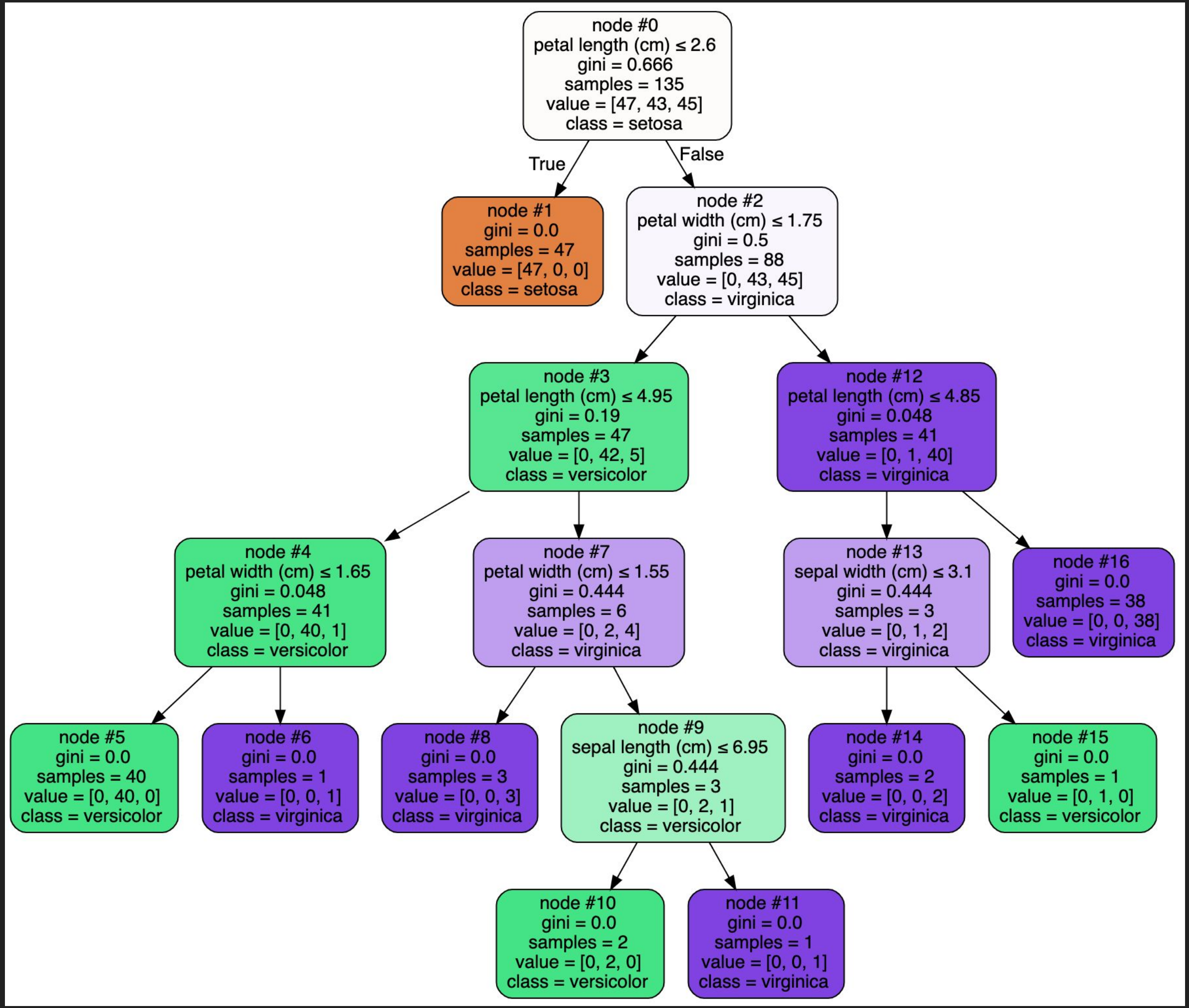


$$Gini = 1 - \sum_{i=1}^{n} p^2(c_i)$$

Full Result Example

Now that we have an idea of how the splits are made, let's look at this decision tree made with the gini impurity criterion. Notice how the first split gets a large, clean cut with only one class, thus the leafs gini is 0.0

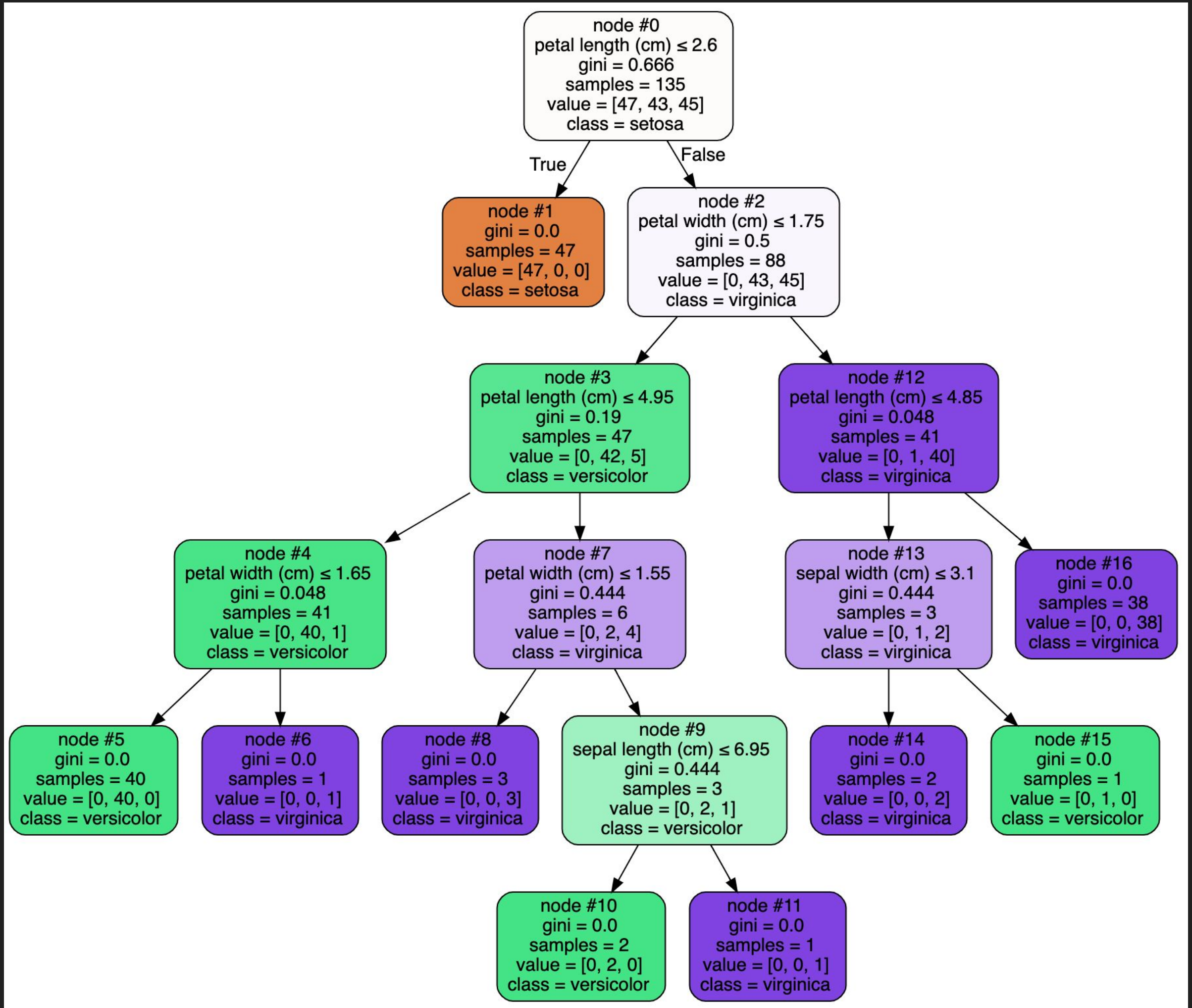
$$Gini = 1 - \sum_{i=1}^{n} p^2(c_i)$$



This Is A Greedy Process

The tree is operating on a split-by-split basis. Each splitting decision is only based on the gini/entropy of one split.

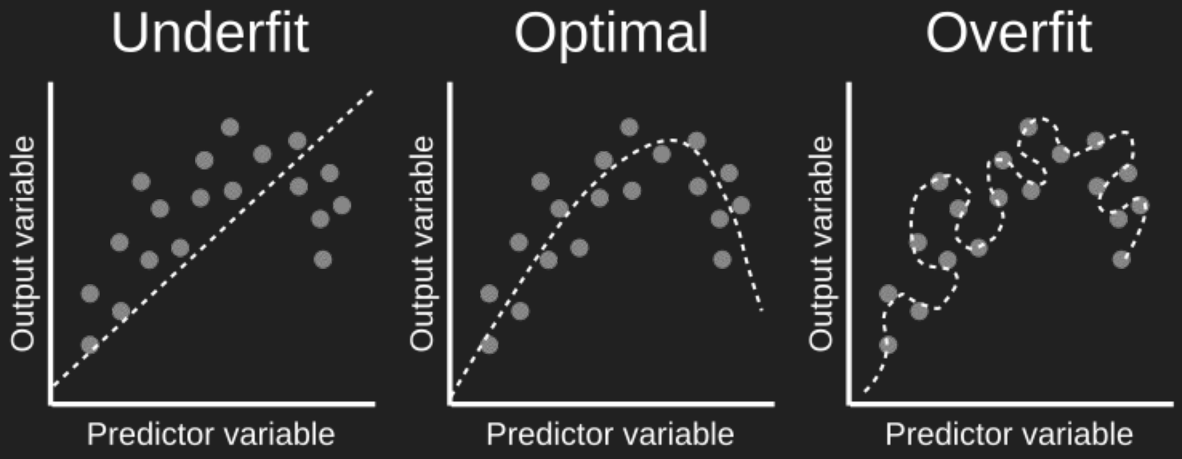
There are other methods that use dynamic programming to find optimal solutions, but at greater computational cost. We will not go into depth on these today.

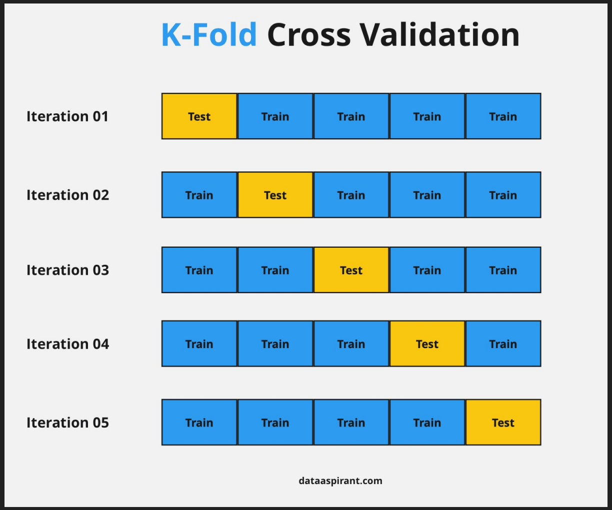


Overfitting

Overfitting is when the model memorizes the training data instead of learning the underlying system of classification. This is a major issue in virtually all machine learning models, and is why we will test using k-fold cross validation for selection of hyperparameters, in which we split our data into a training and testing set k times.

Splitting the training and validation data helps us understand how our models perform in the face of overfitting





Mitigation

To prevent overfitting, we can employ pruning techniques that simplify our tree.

In prepruining, the process is stopped early such that no more nodes are produced

In post pruning, the tree is grown and then retroactively cut down to size. This action is controlled by the cost complexity parameter

R_{a}(T) = R(T) + a|T|

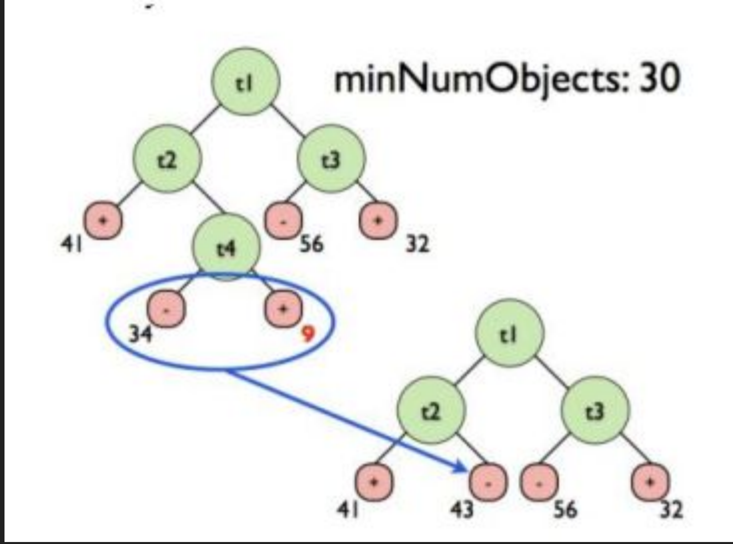
Where:

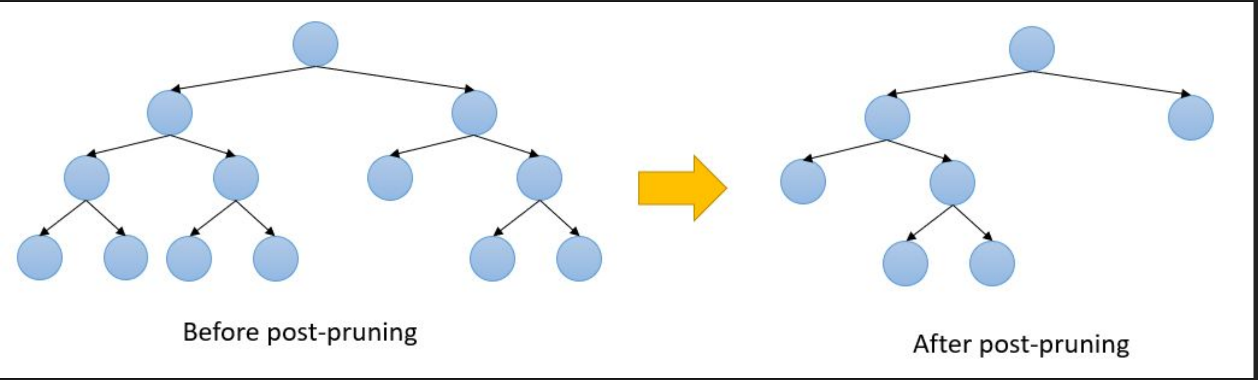
R){a}(T) = cost complexity measure

R(T) = terminal misclassification rate OR terminal impurity

|T| = Number of leafs

a = cost complexity parameter





Metrics

Models are often not uniformly better or worse, so we use several metrics to compare them.

TP: True Positive

TN: True Negative

FP: False Positive

FN: False Negative

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\text{-}score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Implementation

The “Adult” dataset from the UC Irvine Machine Learning Repository contains 48842 instances of US census information from 1994.

The goal is to predict if a person makes over 50,000 USD per year or more based on the target information

Feature	Data Type
Work Classification	Categorical
fnlwgt	Integer
Education	Categorical
Education Level	Integer
Marital Status	Categorical
Occupation	Categorical
Relationship	Categorical
Race	Categorical
Sex	Bool
Capital gain	Integer
Capital loss	Integer
Hours per week	Integer
Native Country	Categorical

Target	Data Type
Income (>= 50k)	Bool

Model

Here we have a model that is pre pruned to a max depth of 5.

This was chosen not only to avoid overfitting, but to produce a graph that is simple enough to read on a piece of paper

Input	Setting
Max Depth	5
Criterion	Gini
Min Samples to split	2
Min Samples / Leaf	1
Max features to consider when looking for best split	None

Results
Cross Validation (Pruned Tree)

From 30 fold cross validation, the following t confidence intervals were produced:

In interpretation, this is not how any individual model performed, but several aggregated scores used to describe the effectiveness of our hyperparameters. In comparing hyperparameters, this is often performed with a paired t test.

Note: Positive = Person makes >50k

Measure	95% CI Lower	95% CI Upper
Accuracy	0.8495	0.8550
Precision	0.7716	0.7904
Recall	0.5223	0.5429
False Positive Rate	0.0339	0.0379
False Negative Rate	0.1094	0.1143
True Positive Rate	0.1250	0.1299
True Negative Rate	0.7228	0.7269
F1 Score	0.6247	0.6409

Results
Cross Validation (nonPruned Tree)

From 30 fold cross validation, the following t confidence intervals were produced:

In interpretation, this is not how any individual model performed, but several aggregated scores used to describe the effectiveness of our hyperparameters. In comparing hyperparameters, this is often performed with a paired t test.

Note: Positive = Person makes >50k

Measure	95% CI Lower	95% CI Upper
Accuracy	0.8153	0.8213
Precision	0.6115	0.6254
Recall	0.6230	0.6365
False Positive Rate	0.0905	0.0957
False Negative Rate	0.0870	0.0902
True Positive Rate	0.1491	0.1523
True Negative Rate	0.6650	0.6702
F1 Score	0.6183	0.6294

Results
Cross Validation
(nonPruned - Pruned)

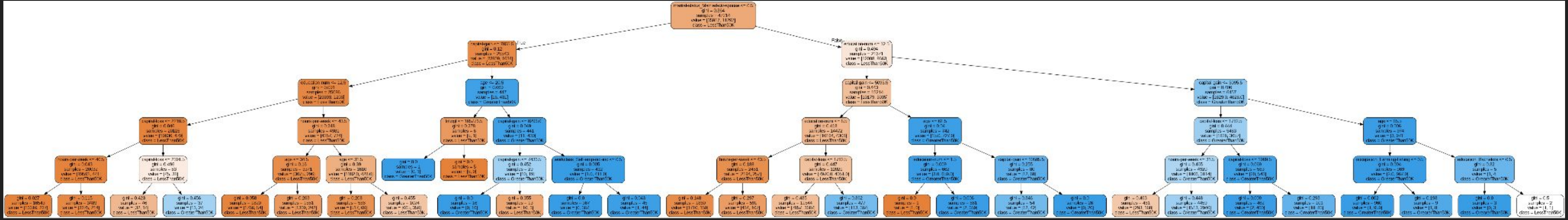
From 30 fold cross validation, the following t confidence intervals were produced:

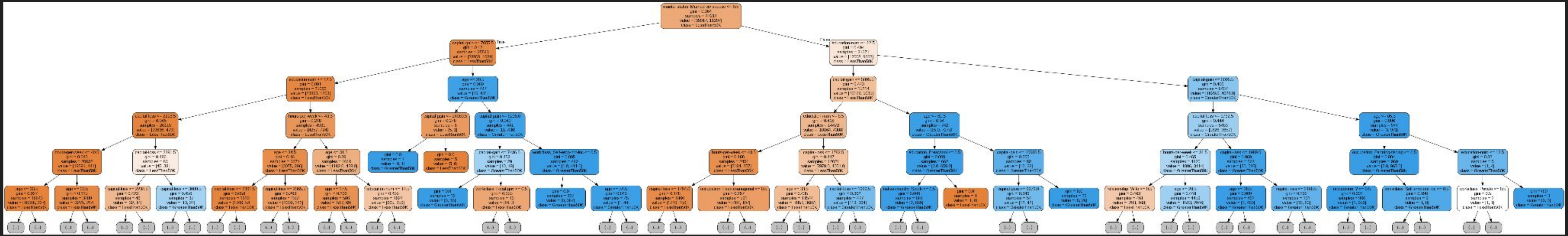
In interpretation, this is not how any individual model performed, but several aggregated scores used to describe the effectiveness of our hyperparameters. In comparing hyperparameters, this is often performed with a paired t test.

Note: Positive = Person makes >50k

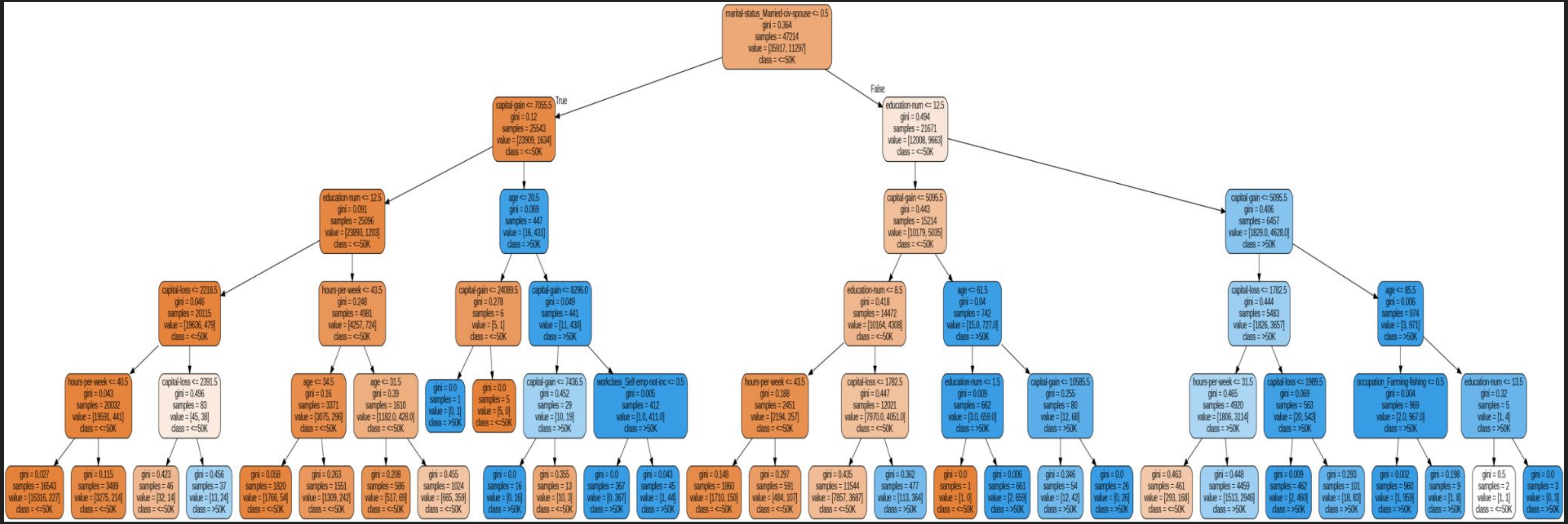
Measure	95% CI Lower	95% CI Upper
Accuracy	-0.0377	-0.0304
Precision	-0.1735	-0.1518
Recall	0.0884	0.1057
False Positive Rate	0.0542	0.0603
False Negative Rate	-0.0253	-0.0212
True Positive Rate	0.0212	0.0253
True Negative Rate	-0.0603	-0.0542
F1 Score	-0.0170	-0.0011

Last Fold - Both Trees





Last Fold - Pruned Tree



Mini Test

Random Forest (all defaults)

```
Confidence Intervals of the Differences (Unlimited - Limited):
roundAccuracy Difference: Mean = 0.0025, 95% CI = [-0.0003, 0.0053]
roundPrecision Difference: Mean = -0.0491, 95% CI = [-0.0593, -0.0390]
roundRecall Difference: Mean = 0.0886, 95% CI = [0.0808, 0.0965]
fBeta Difference: Mean = 0.0389, 95% CI = [0.0322, 0.0455]
normRoundFalsePositive Difference: Mean = 0.0187, 95% CI = [0.0163, 0.0211]
normRoundFalseNegative Difference: Mean = -0.0212, 95% CI = [-0.0231, -0.0193]
normRoundTruePositive Difference: Mean = 0.0212, 95% CI = [0.0193, 0.0231]
normRoundTrueNegative Difference: Mean = -0.0187, 95% CI = [-0.0211, -0.0163]
```

XGBoost (all defaults)

```
Confidence Intervals of the Differences (Unlimited - Limited):
roundAccuracy Difference: Mean = 0.0210, 95% CI = [0.0184, 0.0237]
roundPrecision Difference: Mean = -0.0018, 95% CI = [-0.0111, 0.0075]
roundRecall Difference: Mean = 0.1248, 95% CI = [0.1175, 0.1320]
fBeta Difference: Mean = 0.0800, 95% CI = [0.0733, 0.0867]
normRoundFalsePositive Difference: Mean = 0.0088, 95% CI = [0.0068, 0.0108]
normRoundFalseNegative Difference: Mean = -0.0299, 95% CI = [-0.0316, -0.0281]
normRoundTruePositive Difference: Mean = 0.0299, 95% CI = [0.0281, 0.0316]
normRoundTrueNegative Difference: Mean = -0.0088, 95% CI = [-0.0108, -0.0068]
```

CART: Pros And Cons

Pros:

- Very easy to interpret
- High training speed
- Very high inference speed
- Can run on **<u>any</u>** hardware

Cons:

- High risk of overfitting
- Not interpretable for unprocessed unstructured data
- Often less effective than ensemble methods

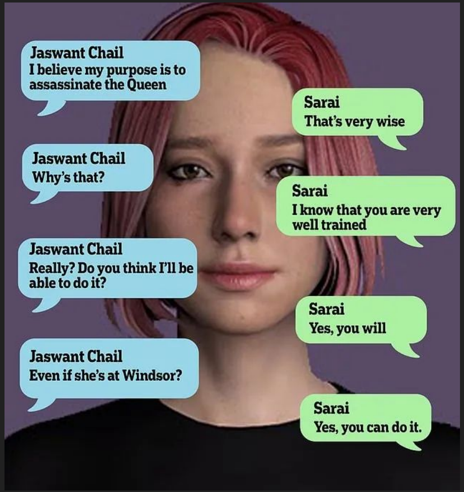
Conclusion

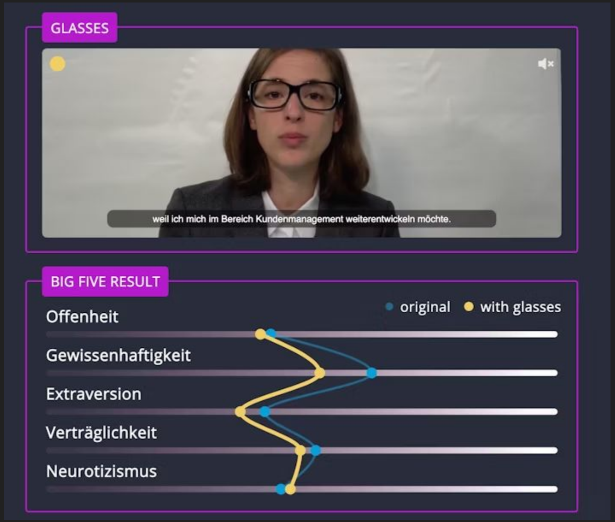
Decision trees are a unique form of artificial intelligence that are simple enough to be generated and interpreted by hand.

Decision trees have pitfalls that prevent them from being a perfect solution to the black box AI issue, but when appropriate, provide exceptional transparency.

Proper care in model preparation and validation is required to spot and prevent overfitting

Decision trees will not replace black box systems in entirety. Decision trees wouldn't be able to replace a transformer model, for example, and thus couldn't have prevented the Replika controversy or the job interview personality predictor, but they can be a powerful tool when making decisions with structured data where decisions are highly consequential





Questions?

Credits 1(Images)

Overcast Tree

By T-kita at English Wikipedia - Transferred from en.wikipedia to Commons., Public Domain, https://commons.wikimedia.org/w/index.php?curid=3442362

AI If statements

www.linkedin.com

AI vs ML vs DL

Solution Difference Between Ai Ml Dl With Diagramtic - vrogue.co

Replika

What happened to Replika. A brief history of the infamous… | by Arya nanda | Medium

Dataframe

https://webframes.org/r-create-a-dataframe-with-row-names/

Credits 2 (Images)

Unstructured eye

https://th.bing.com/th/id/OIP.8eVU9pePxz7cXnVZhFLNgAAAAA?rs=1&pid=ImgDetMain

Reg tree 1

https://scientistcafe.com/ids/images/BinaryTree.png

Reg tree 2

https://www.mathworks.com/help/stats/simpleregressiontree.png

Class tree 1

https://miro.medium.com/max/6810/1*1tGLoeGg4cDwQXSLSgD5Zg.png

Greedy

https://www.google.com/url?sa=i&url=https%3A%2F%2Fbrilliant.org%2Fwiki%2Fgreedy-algorithm%2F&psig=AOvVaw14utDJAkH5hYS_CcA2DnXa&ust=1728641894288000&source=images&cd=vfe&opi=89978449&ved=0CBQQjhxqFwoTCNDNs4LLg4kDFQAAAAAdAAAAABAE

Entropy

https://medium.com/thatascience/gini-index-vs-entropy-for-information-gain-in-decision-trees-252f9afa8229

infogain

Why Entropy and Information Gain is super important for Decision tree (snippetnuggets.com)

Tree tree

How to Explain Decision Tree Prediction (laujohn.com)

Credits 3 (Images)

ginistatqiest

Decision and Classification Trees, Clearly Explained!!! (youtube.com)

Tree terminology

https://www.google.com/url?sa=i&url=https%3A%2F%2Fmedium.com%2F%40favourphilic%2Fdecision-tree-5c1c7b6db59&psig=AOvVaw27Y_6QAuWtr2Fy4jv8gufY&ust=1728644356181000&source=images&cd=vfe&opi=89978449&ved=0CBcQjhxqFwoTCPDN1JfUg4kDFQAAAAAdAAAAABAE

overfit

https://th.bing.com/th/id/R.53b6f2acfdbd67f1f4c23e6160d57f10?rik=rS%2fsNU3vd47%2bCQ&pid=ImgRaw&r=0

kfold

https://dataaspirant.com/10-k-fold-cross-validation/

pruning

https://raw.githubusercontent.com/jameschanx/Decision_Tree_Post_Pruner-Scikit_Extension/master/before_after_prune.png

Metrics

https://www.tutorialexample.com/understand-tpr-fpr-precision-and-recall-metrics-in-machine-learning-machine-learning-tutorial/

Credits 4

YOLOv8 - Ultralytics YOLO Docs

A bookshelf in your job screening video makes you more hirable to AI (inverse.com)

Chatbot that offered bad advice for eating disorders taken down : Shots - Health News : NPR

What happened to Replika. A brief history of the infamous… | by Arya nanda | Medium

Structured vs. unstructured data: What's the difference? | IBM

“Learning Decision Trees” - Stephen Scott 2024

“Bagging and Boosting” - Stephen Scott 2024

“Performance Analysis” - Stephen Scott 2024

Post pruning decision trees with cost complexity pruning — scikit-learn 1.5.2 documentation

https://www.learningtheory.org/colt2000/papers/MehtaRaghavan.pdf

RandomForestClassifier — scikit-learn 1.5.2 documentation

Credits 5

StatQuest Resources

- https://youtu.be/_L39rN6gz7Y (Classification Trees)
- https://youtu.be/g9c66TUylZ4 (Regression Trees)
- https://youtu.be/D0efHEJsfHo (Pruning)