https://ttpoll.com/p/643115

COSC 325: Introduction to Machine Learning

Dr. Hector Santos-Villalobos

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Lecture 11: Regularization and Decision Trees

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Class Announcements

Homework:

Homework #3 DD 09/29

Start early. Don’t expect TA support during weekends.

Course Project:

Teaming issues.

Course grade distribution change:

- **Exams:** ~~45%~~ **35%**
- **Homework:** ~~20%~~ **30%**

Lectures:

On October 1st, no attendance record due to the Engineering Expo

Exams:

Exam #1: Thursday, 10/03

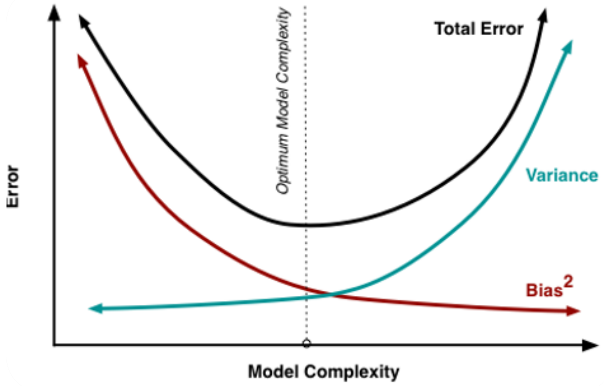
- Online
- Window 11 am to 1 pm
- 75 mins

Review

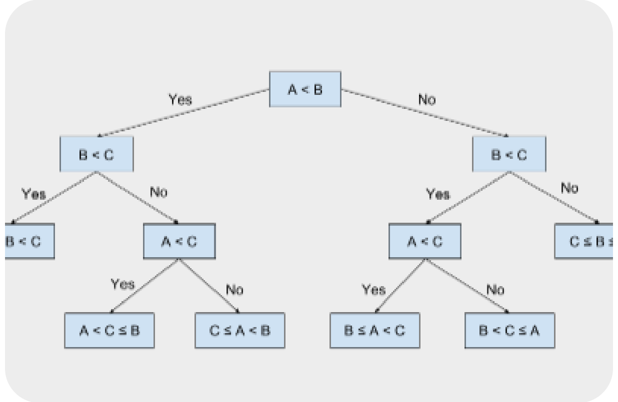
- Capacity
- Overfitting/Underfitting
- Bias-Variance Tradeoff
- Loss = Bias^2 + Variance + Irreducible Error
- Regularization techniques

Today's Topics

Decision Trees



Regularization



Regularization

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

L2 Regularization (Ridge)

Regularization

Parameter to control how much to regularize.

Logistic Regression

$$\min_{w,b} J(w) = -\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) + \overbrace{\frac{\lambda}{2m}\|w\|_2^2}$$

Removes sqrt of L2 norm.

L2 Regularization: $\|w\|_2^2 = \sum_{j=1}^{m} w_j^2 = w^T w$

Popular approach.

L2 Regularization (Ridge)

Logistic Regression

$$\min_{w,b} J(w) = -\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) \quad + \frac{\lambda}{2m}\|w\|_2^2$$

L2 Regularization: $\|w\|_2^2 = \sum_{j=1}^{m} w_j^2 = w^T w$

Heavily penalizes larger weights

L2 Regularization (Ridge)

Logistic Regression

$$\min_{w,b} J(w) = -\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) \quad + \frac{\lambda}{2m}\|w\|_2^2$$

L2 Regularization: $\|w\|_2^2 = \sum_{j=1}^{m} w_j^2 = w^T w$

Note: $\frac{\partial\left(\frac{\lambda}{2m}\|w\|_2^2\right)}{\partial w_i} = \frac{\partial\left(\frac{\lambda}{2m}\sum_{j=1}^{m} w_j^2\right)}{\partial w_i} = \frac{\lambda}{m} w_i$

Gradient Descent with Regularization

$$dW = \frac{dJ}{dW} = dW = \frac{1}{n} A \quad dZ + \frac{\lambda}{m} W$$

$$W := W - \alpha dW$$

$$= W - \alpha \left[\frac{1}{n} A \quad dZ + \frac{\lambda}{m} W \right]$$

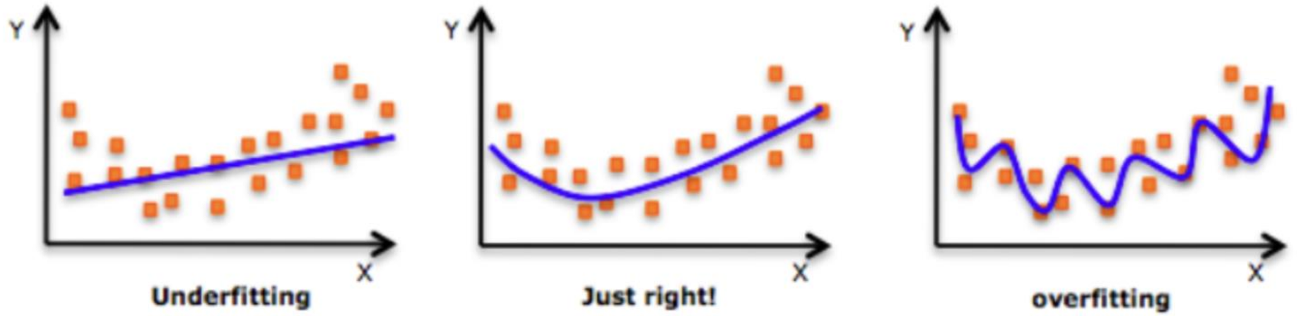
$$= \left(1 - \frac{\alpha\lambda}{m}\right) W - \alpha \left[\frac{1}{n} A \quad dZ \right]$$

We are penalizing weight magnitude.

Also called "Weight Decay" for this reason.

Intuition on Regularization

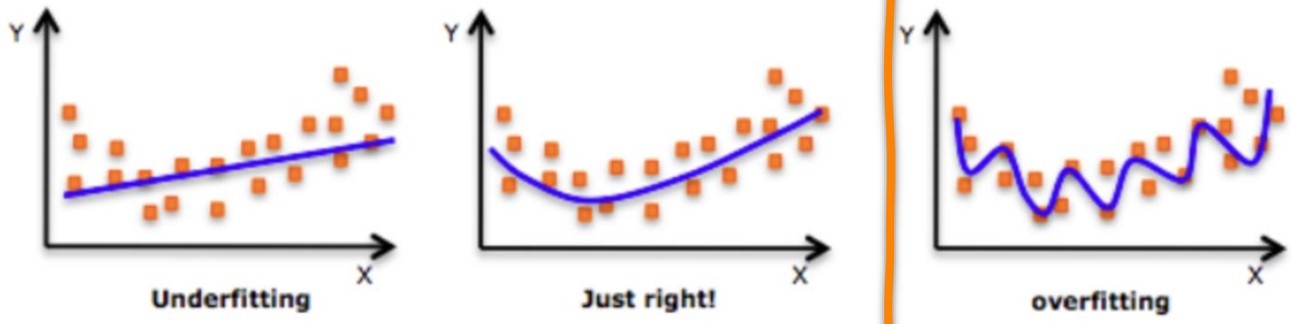
High Bias | Just Right | High Variance



$$\min_{W} J(W) = -\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) + \frac{\lambda}{2m}\|W\|_2^2$$

Intuition on Regularization

High Bias | Just Right | High Variance



$$\min_{W} J(W) = -\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) + \frac{\lambda}{2m}\|W\|_2^2$$

If $\lambda \gg 0$, then $W \to 0$

Capacity

Intuition on Regularization

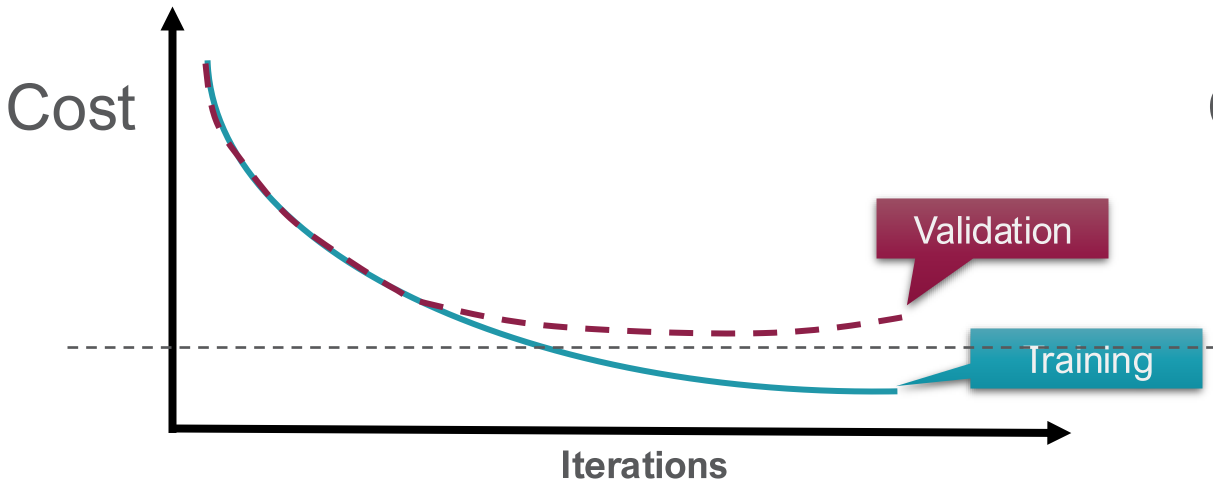
$$\min_{W} J(W) = -\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) + \frac{\lambda}{2m}\|W\|_2^2$$

If $\lambda \gg 0$, then $W \to 0$

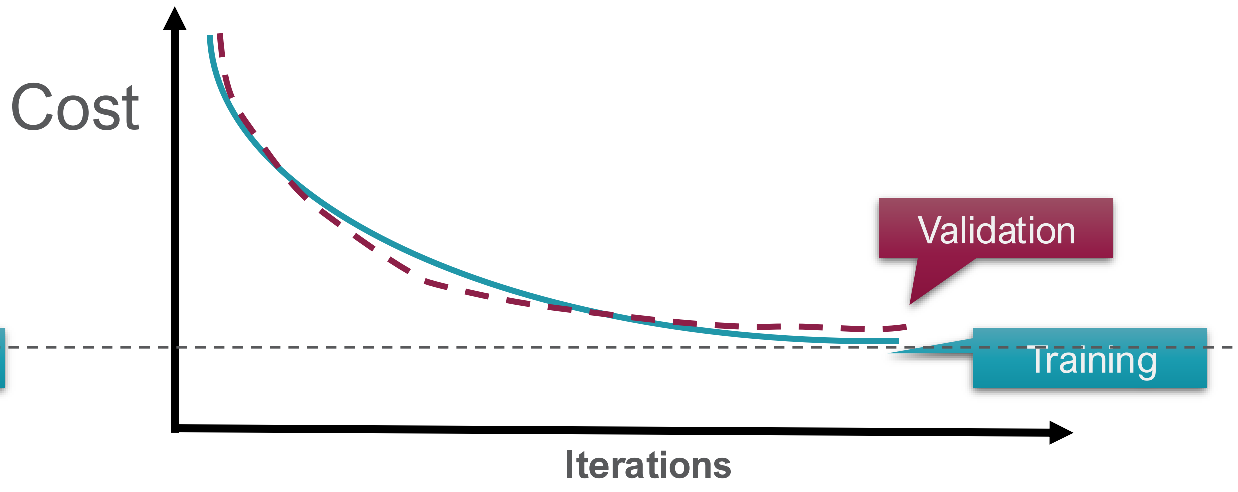
https://ttpoll.com/p/643115

Overfitting

Low Bias-High Variance



Higher Bias-Low Variance



$$\min_{W} J(W) = -\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) + \frac{\lambda}{2m}\|W\|_2^2$$

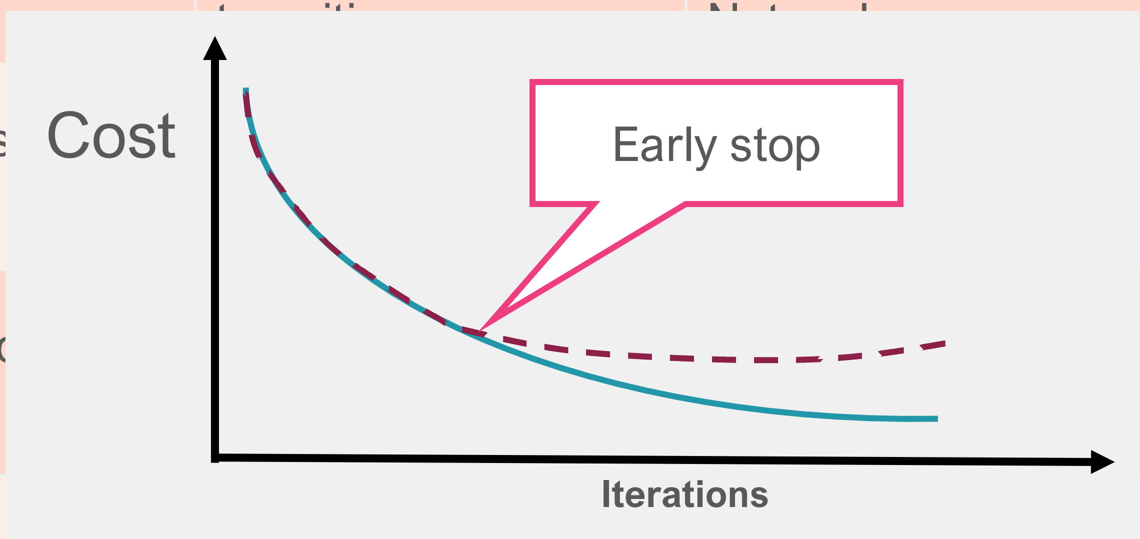
If $\lambda > 0$, then $W \rightarrow 0$

Popular Regularization/Penalty Terms

Technique	Formula	Type	Effect	Common use cases
Ridge (L2)	$\frac{\lambda}{2}\sum_{i=1}^{m} w_j^2 = w^T w$	Penalizes squared weights	Rewards smaller weights, smoother transitions.	Linear/Logistic Regression, Neural Networks
Lasso (L1)	$\lambda \sum_{j=1}^{m} \lvert w_j \rvert$	Penalizes absolute weights	Rewards sparsity (feature space reduction)	High-dimensional data
ElasticNet	$\frac{\lambda_1}{2} \lVert w \rVert_2^2 + \lambda_2 \lVert w \rVert_1$	Combines Ridge and Lasso	Balances sparsity (L1) and smoothness (L2)	High-dimensional data with correlated features
Early Stopping	N/A	Stops training after specified cost event.	Prevents overfitting by using an earlier checkpoint.	Neural Networks

Popular Regularization/Penalty Terms

Technique	Formula	Type	Effect	Common use cases
Ridge (L2)	$\frac{\lambda}{2}\sum_{i=1}^{m} w_j^2 = w^T w$	Penalizes squared weights	Rewards smaller weights, smoother	Linear/Logistic Regression, Neural
Lasso (L1)	$\lambda\sum_{j=1}^{m}\lvert w_j\rvert$	Penalizes abs weights		
ElasticNet	$\frac{\lambda_1}{2}\lVert w\rVert_2^2 + \lambda_2\lVert w\rVert_1$	Combines Ri Lasso		
Early Stopping	N/A	Stops training after specified cost event.	by using an earlier checkpoint.	Neural Networks



Pop Quiz

The purpose of regularization is to ______________.

A. compute the average parameter/weight value.

B. decrease the likelihood of overfitting.

C. decrease the capacity of the model.

D. filter the outliers in the dataset.

Pop Quiz

The purpose of regularization is to _______________.

A. compute the average parameter/weight value.

B. decrease the likelihood of overfitting.

C. decrease the capacity of the model.

D. filter the outliers in the dataset.

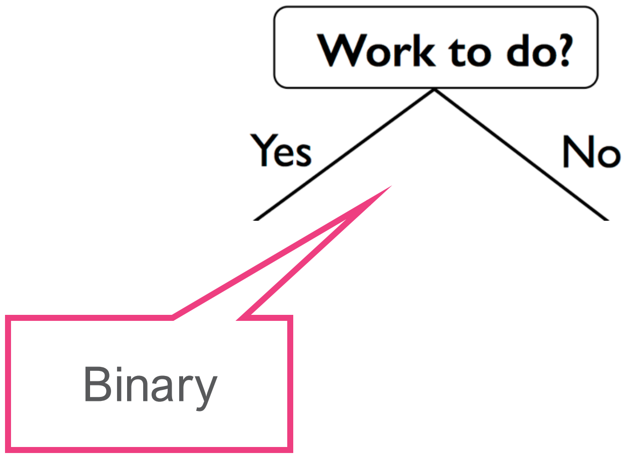
Decision Trees

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Decision Trees

Stay home or go to the movies

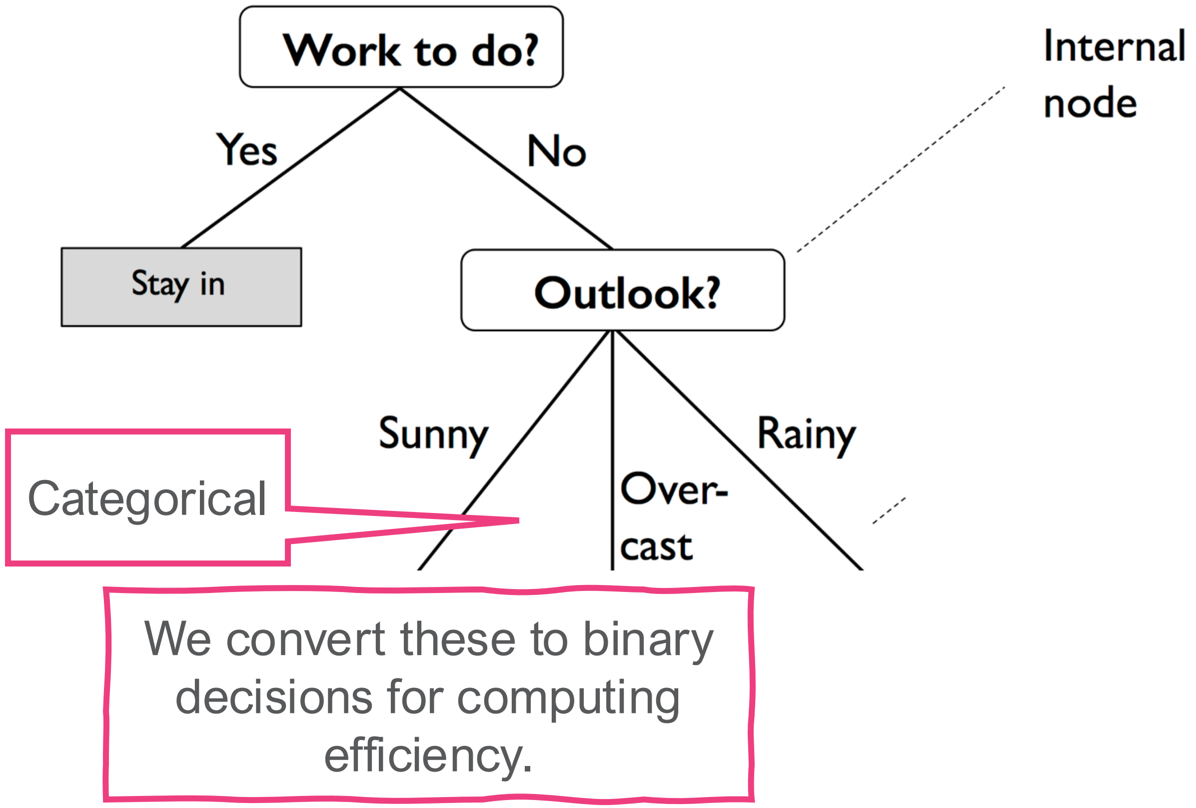
- Iterative top-down creatio of hypothesis (Classifier)
- Hierarchy of decisions
  – We ask questions to split the dataset.



Decision Trees

- Iterative top-down creatio of hypothesis (Classifier)
- Hierarchy of decisions
  – We ask questions to split th dataset.

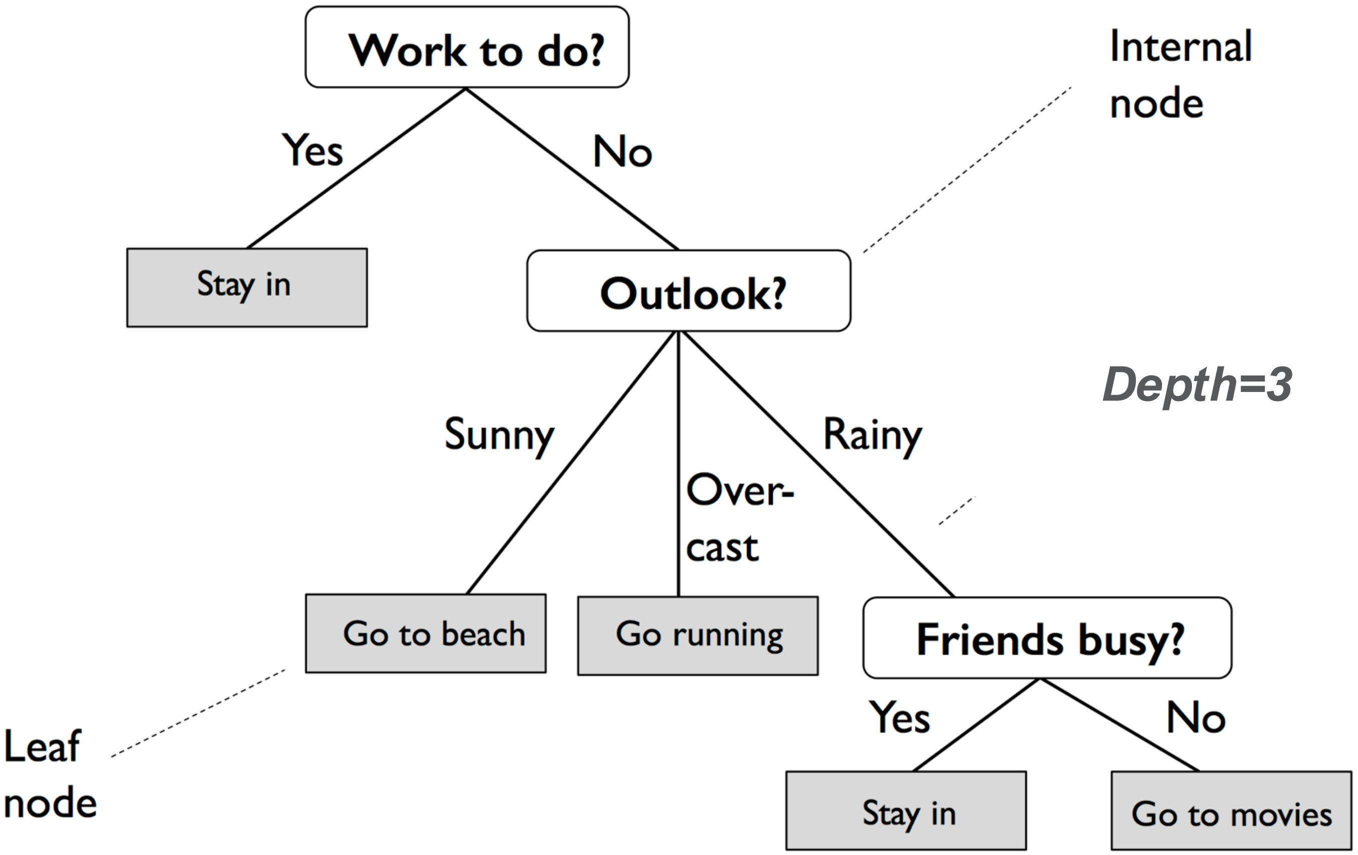
Stay home or go to the movies



Decision Trees

- Iterative top-down creatio of hypothesis (Classifier)
- Hierarchy of decisions
  - We ask questions to split th dataset.
- Highly explainable

Stay home or go to the movies



Source: Dr. Raschka, Machine Learning Book

Top-Down Induction of Decision Trees

- If you could only ask one question, what question would you ask?
- Natural greedy approach to growing a decision tree in a top-down way
- Algorithm:
  - Pick "best" attribute to split at the root based on training data
  - Recurse on children that are "impure" (e.g., have both yes and no)

Top-Down Induction of Decision Trees

- Natural greedy approaches where ***we grow the tree*** from the root to the leaves by repeatedly ***replacing*** an existing ***leaf*** with an ***internal node***

Slide Credit: Dr. Schumman

Data Example (Jedi/Sith)

Clothing	Human	Voice Pitch	Affiliation
Brown	Yes	Medium	Jedi
Black	No	Medium	Jedi
Black	Yes	High	Sith
Brown	No	Low	Jedi
Black	No	Low	Sith
Brown	Yes	Low	Sith
Brown	Yes	Medium	Jedi
Black	Yes	Low	Sith

X y

Team Exercise

- Create teams of three (Ideally neighbors)
- Assume ***Voice*** feature categories: {Low, Medium/High}
- You have ***3-5 minutes*** to design this data's ***"best"*** tree.

Pop Quiz

What is the depth of your tree, and how many internal and leaf nodes do you have?

A. depth=2, internal=2, leaf=4

B. depth=4, internal=5, leaf=6

C. depth=3, internal=5, leaf=6

D. depth=3, internal=3, leaf=4

E. depth=4, internal=3, leaf=4

F. Other

Data Example (Jedi/Sith)

Clothing	Human	Voice Pitch	Affiliation
Brown	Yes	Medium/High	Jedi
Black	No	Medium/High	Jedi
Black	Yes	Medium/High	Sith
Brown	No	Low	Jedi
Black	No	Low	Sith
Brown	Yes	Low	Sith
Brown	Yes	Medium/High	Jedi
Black	Yes	Low	Sith

X y

Data Example (Jedi/Sith): Clothing

Clothing (x_1)	Human (x_2)	Voice Pitch (x_3)	Affiliation (y)
Brown	Yes	Medium	Jedi
Black	No	[illegible]	Jedi
Black	Yes	[illegible]	Sith
Brown	No	Low	Jedi
Black	No	Low	Sith
Brown	Yes	[illegible]	Sith
Brown	Yes	[illegible]	Jedi
Black	Yes	[illegible]	Sith

$\underbrace{\hspace{6em}}_{X}$ $\underbrace{\hspace{3em}}_{y}$

Jedi

Sith

Black Clothing:
- ***Jedis – 1***
- ***Siths - 3***

Brown Clothing:
- ***Jedis – 3***
- ***Siths - 1***

Data Example (Jedi/Sith): Human

Clothing (x_1)	Human (x_2)	Voice Pitch (x_3)	Affiliation (y)
Brown	Yes	Medium	Jedi
Black	No	[illegible]	Jedi
Black	Yes	[illegible]	Sith
Brown	No	Low	Jedi
Black	No	Low	Sith
Brown	Yes	[illegible]	Sith
Brown	Yes	[illegible]	Jedi
Black	Yes	Low	Sith

X y

Sith

No Human:

- ***Jedis – 2***
- ***Siths - 1***

Yes Human:

- ***Jedis – 2***
- ***Siths - 3***

Data Example (Jedi/Sith): Voice

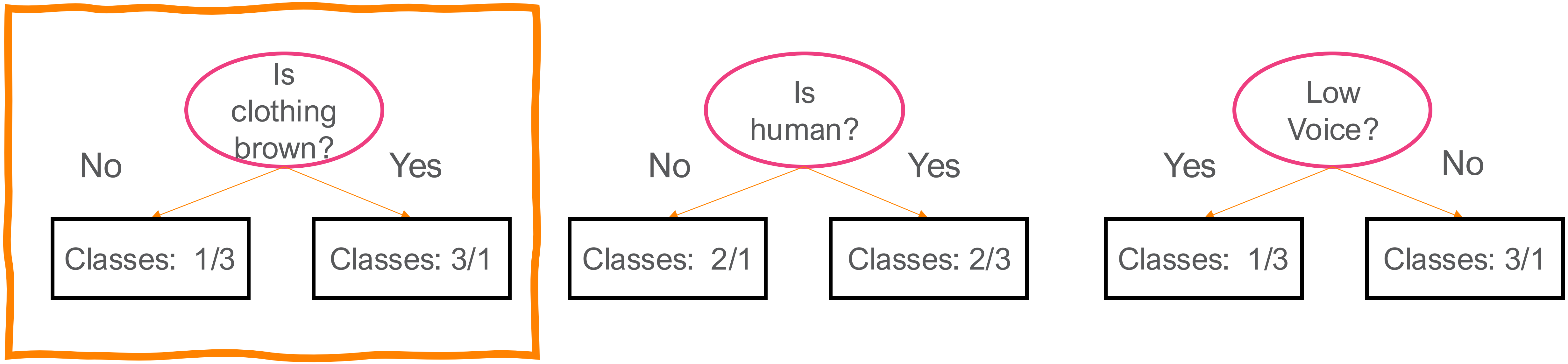
Clothing (x_1)	Human (x_2)	Voice Pitch (x_3)	Affiliation (y)
Brown	Yes	Medium	Jedi
Black		Medium	Jedi
Black		High	Sith
Brown	No	Low	Jedi
Black		Low	Sith
Brown		Low	Sith
Brown		Medium	Jedi
Black	Yes	Low	Sith

X y

Low Voice:
- ***Jedis – 1***
- ***Siths - 3***

Mid-High Voice:
- ***Jedis – 3***
- ***Siths - 1***

Totals: Jedis=5, Siths=4



No single feature enables sample classification on its own.

Data Example (Jedi/Sith): Clothing

Clothing (x_1)	Human (x_2)	Voice Pitch (x_3)	Affiliation (y)
Brown	No	Low	Jedi
Brown	Yes	Low	Sith
Brown	Yes	Medium	Jedi
Brown	Yes	Medium	Jedi
Black	No	Low	Sith
Black	No	Medium	Jedi
Black	Yes	Low	Sith
Black	Yes	High	Sith

$\underbrace{\text{Clothing, Human, Voice Pitch}}_{X}$ $\underbrace{\text{Affiliation}}_{y}$

Black Clothing:

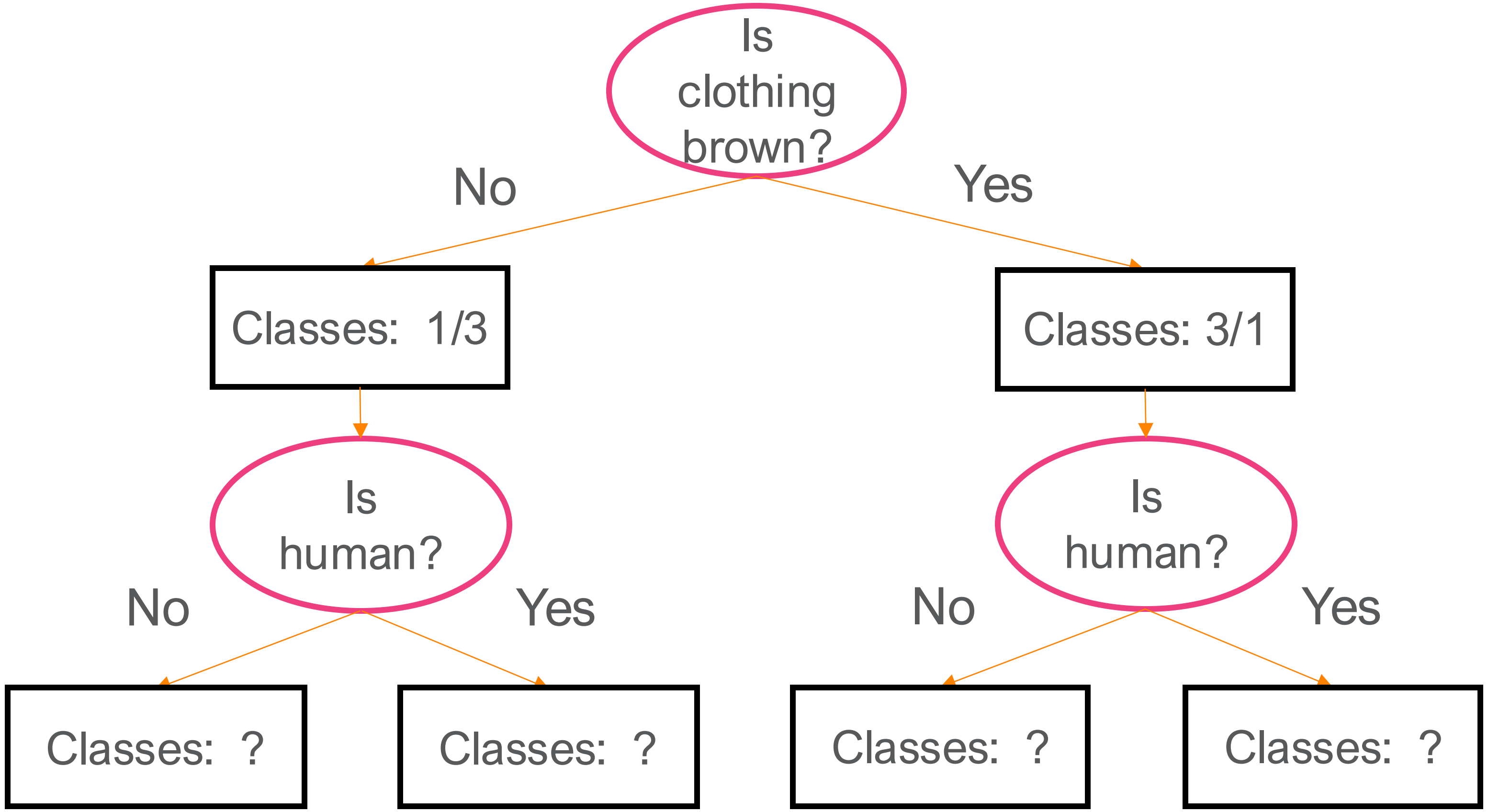
- ***Jedis – 1***
- ***Siths - 3***

Brown Clothing:

- ***Jedis – 3***
- ***Siths - 1***

Totals: Jedis=4, Siths=4

https://ttpoll.com/p/643115



Data Example (Jedi/Sith): Clothing

Clothing (x_1)	Human (x_2)	Voice Pitch (x_3)	Affiliation (y)
Brown	No	Low	Jedi
Brown	Yes	Low	Sith
Brown	Yes	Medium	Jedi
Brown	Yes	Medium	Jedi
Black	No	Low	Sith
Black	No	Medium	Jedi
Black	Yes	Low	Sith
Black	Yes	High	Sith

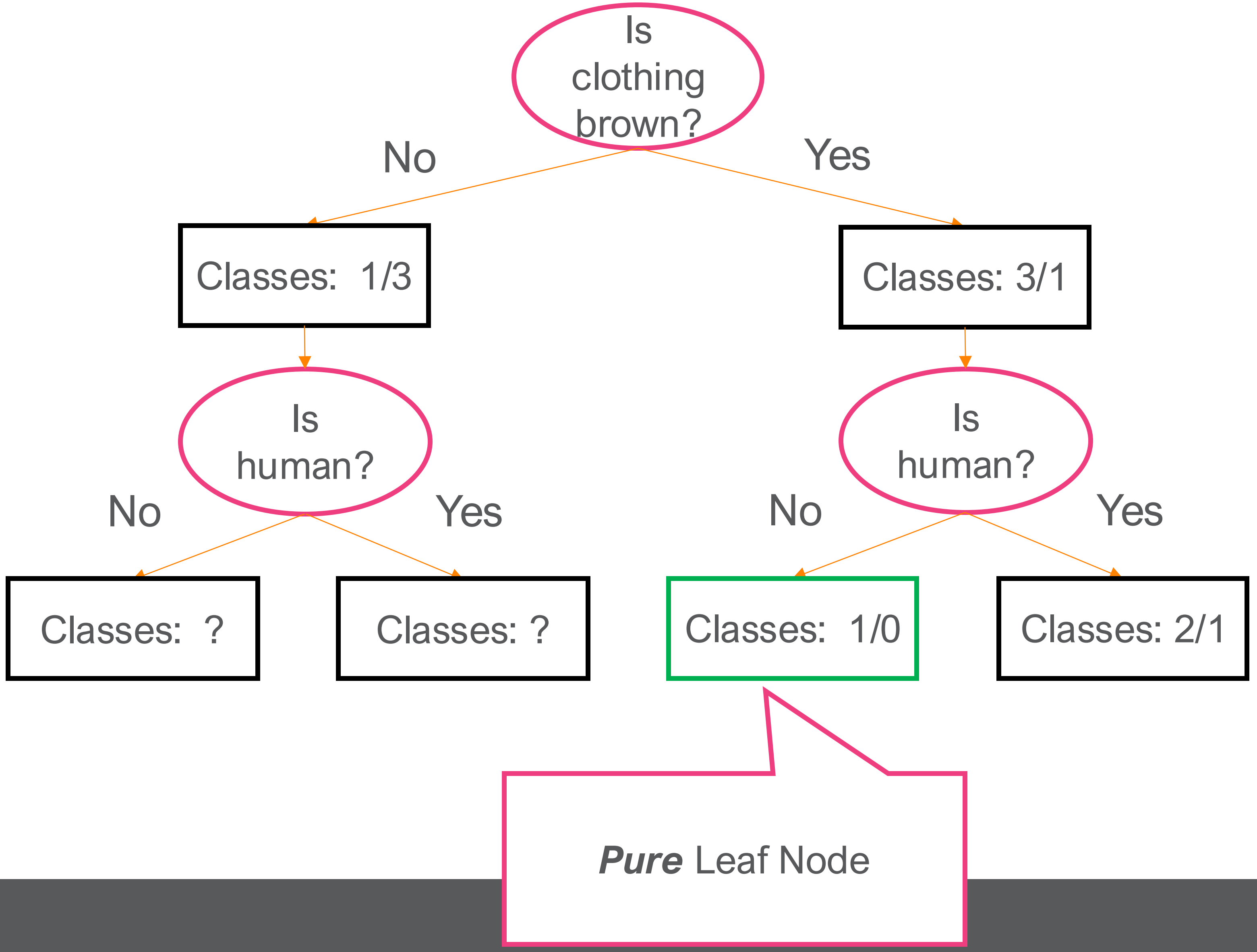
X y

Brown-Human:
- ***Jedis – 2***
- ***Siths - 1***

Brown No-Human:
- ***Jedis – 1***
- ***Siths - 0***

Totals: Jedis=4, Siths=4

https://ttpoll.com/p/643115



THE UNIVERSITY OF TENNESSEE KNOXVILLE

Data Example (Jedi/Sith): Clothing

Clothing (x_1)	Human (x_2)	Voice Pitch (x_3)	Affiliation (y)
Brown	No	Low	Jedi
Brown	Yes	Low	Sith
Brown	Yes	Medium	Jedi
Brown	Yes	Medium	Jedi
Black	No	Low	Sith
Black	No	Medium	Jedi
Black	Yes	Low	Sith
Black	Yes	High	Sith

X y

Black-Human:

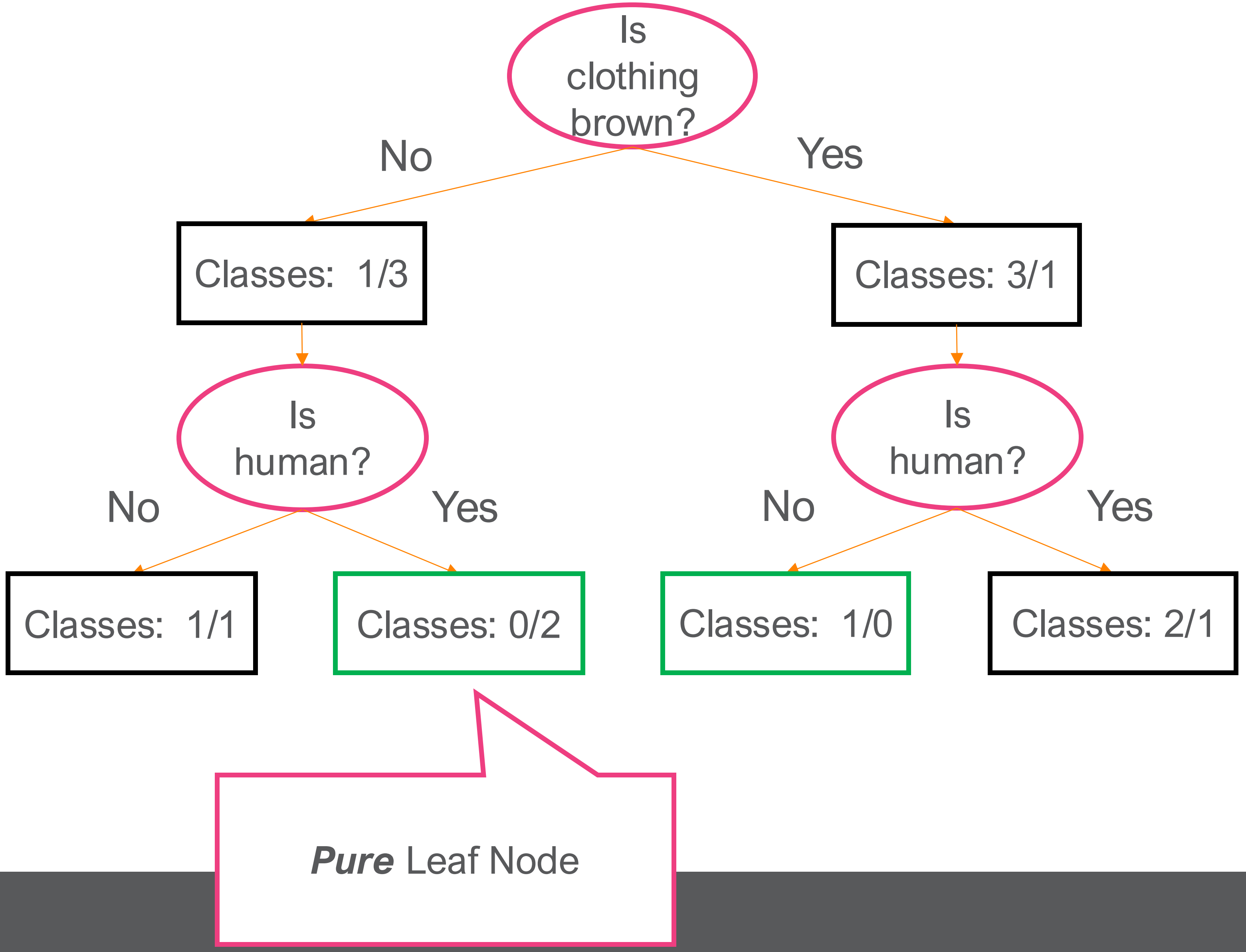
- ***Jedis – 0***
- ***Siths - 2***

Black No-Human:

- ***Jedis – 1***
- ***Siths - 1***

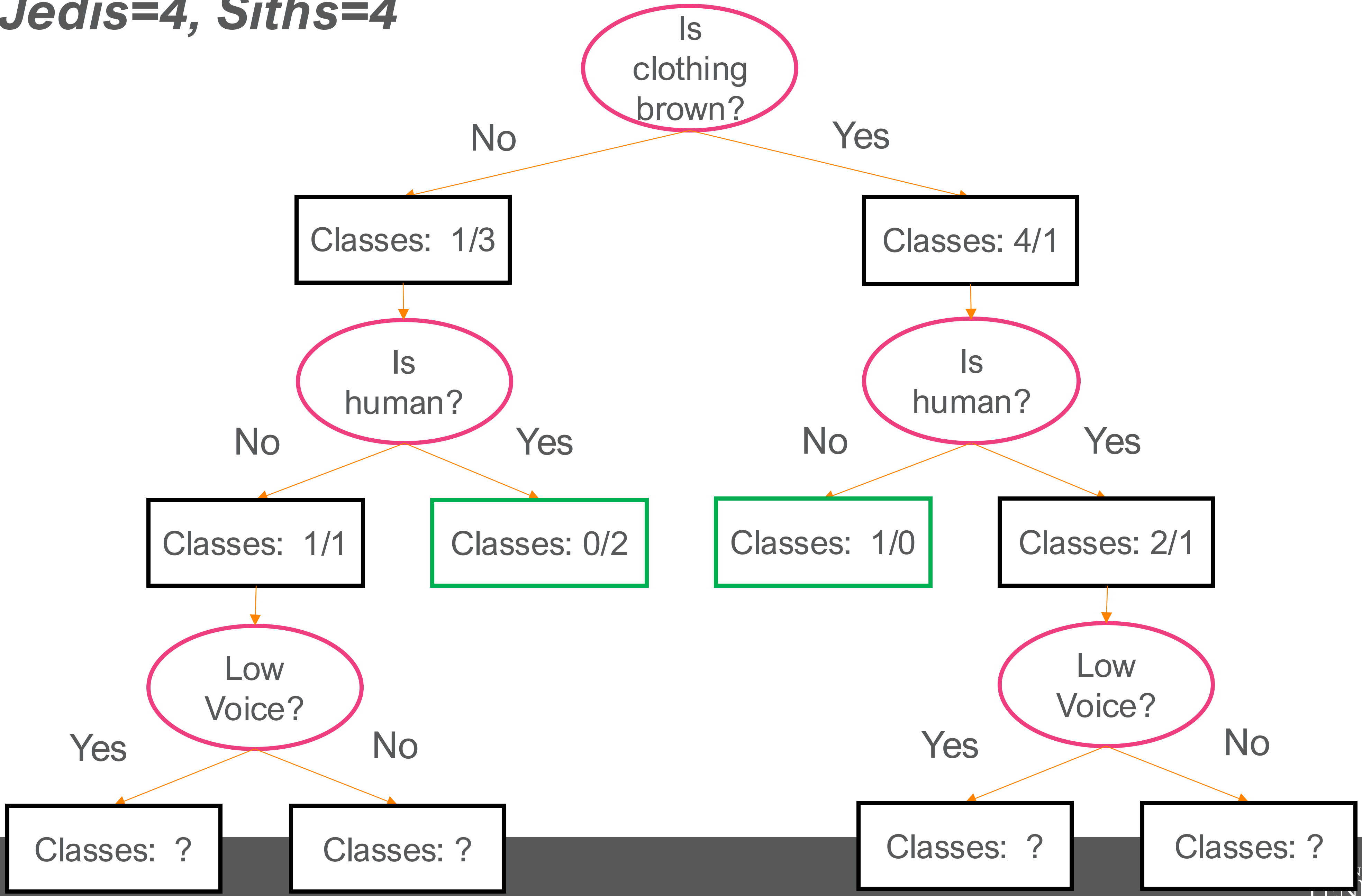
https://ttpoll.com/p/643115

Totals: Jedis=4, Siths=4



https://ttpoll.com/p/643115

Totals: Jedis=4, Siths=4



Data Example (Jedi/Sith): Clothing

Clothing (x_1)	Human (x_2)	Voice Pitch (x_3)	Affiliation (y)
Brown	No	Low	Jedi
Brown	Yes	Low	Sith
Brown	Yes	Medium	Jedi
Brown	Yes	Medium	Jedi
Black	No	Low	Sith
Black	No	Medium	Jedi
Black	Yes	Low	Sith
Black	Yes	High	Sith

1/1

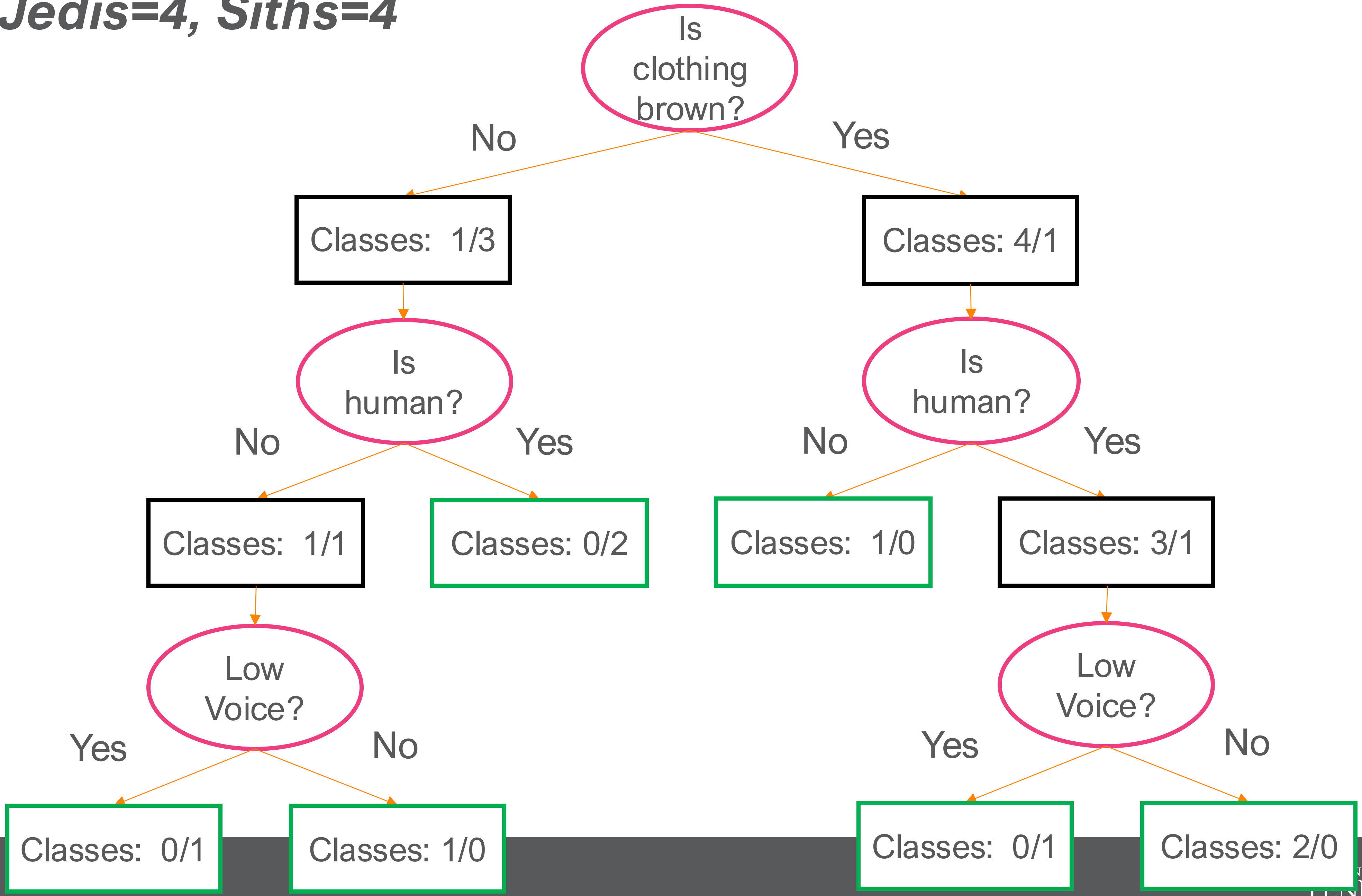
2/0

0/1

1/0

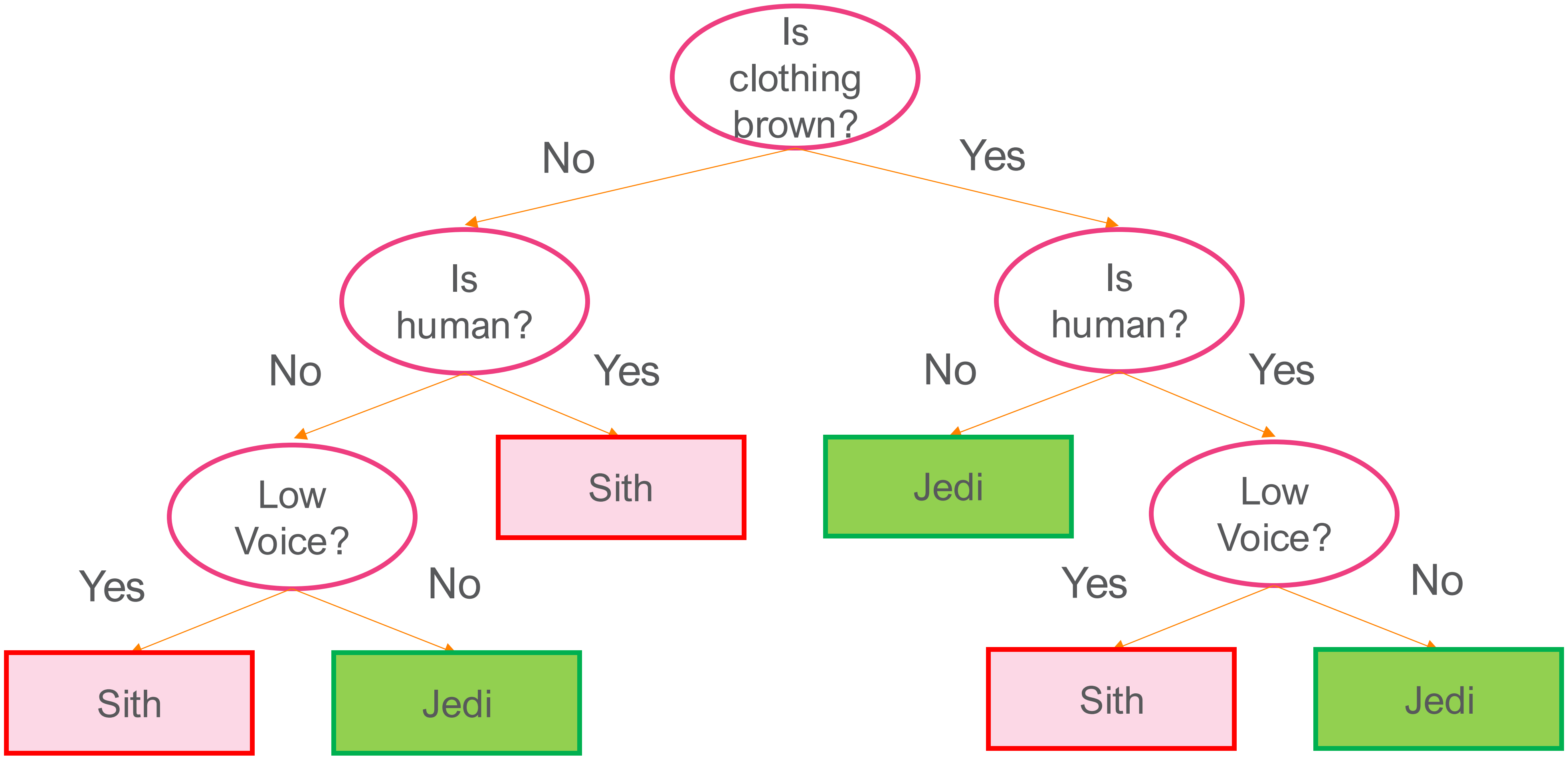
X y

Totals: Jedis=4, Siths=4



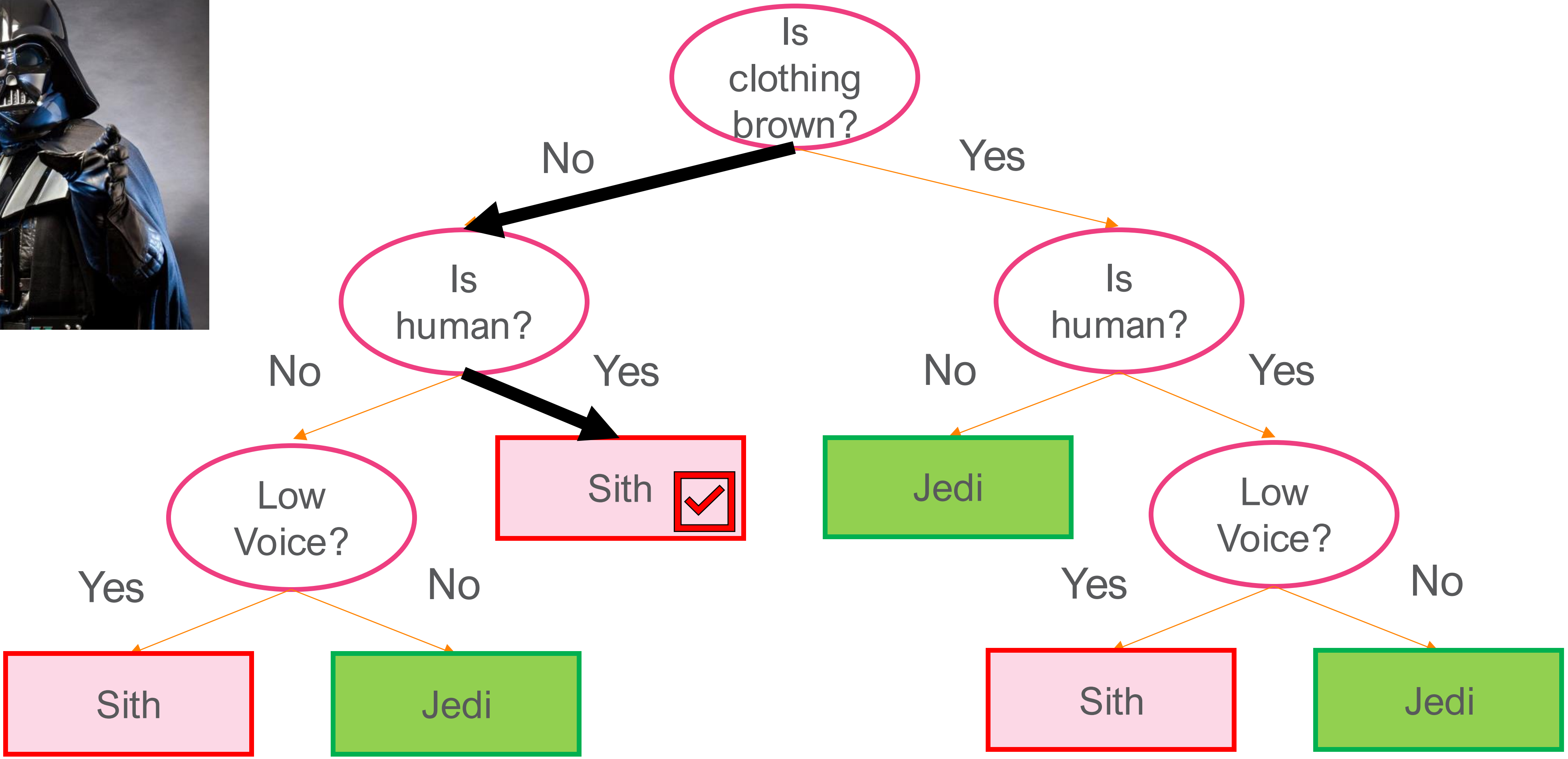
Final Tree

https://ttpoll.com/p/643115

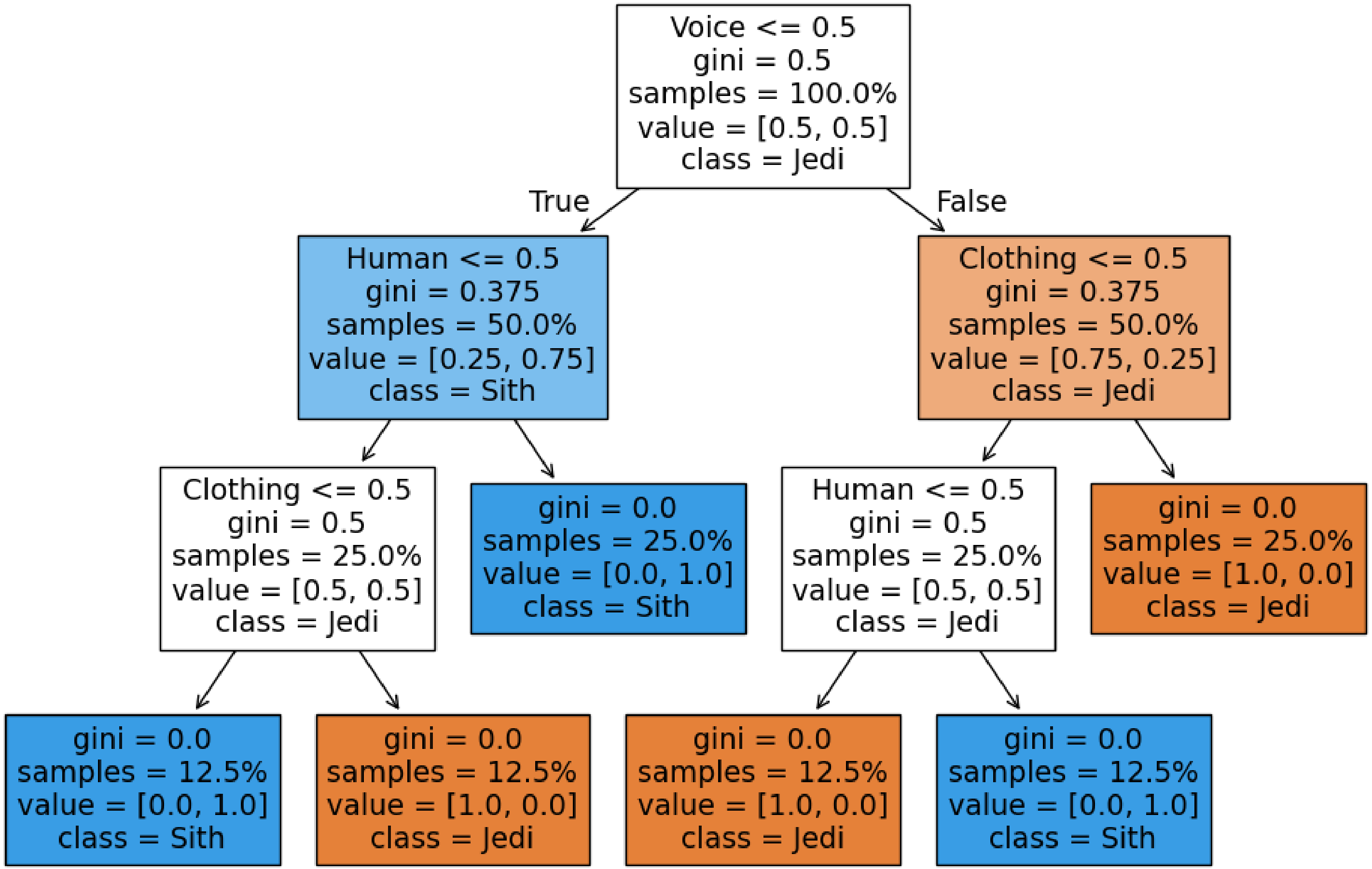


https://ttpoll.com/p/643115

Querying the Tree



SKLearn Fitting



Decision Tree Pseudocode

CART algorithm
(Classification and Regression Trees)

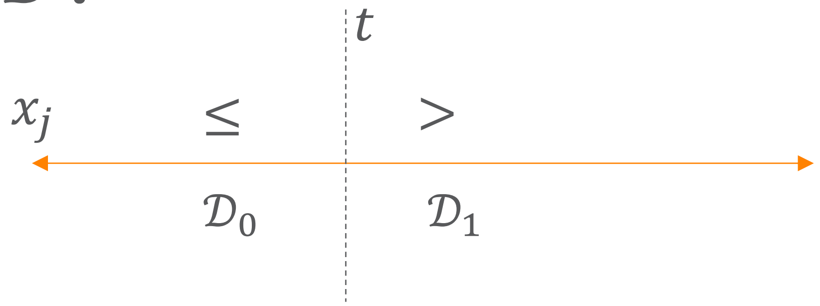
$GenerateTree(\mathcal{D})$:

    if $y = 1\ \forall \langle x, y\rangle \in \mathcal{D}$ or $y = 0\ \forall \langle x, y\rangle \in \mathcal{D}$:

        return Tree

    else:

        Pick "best" feature x_j:

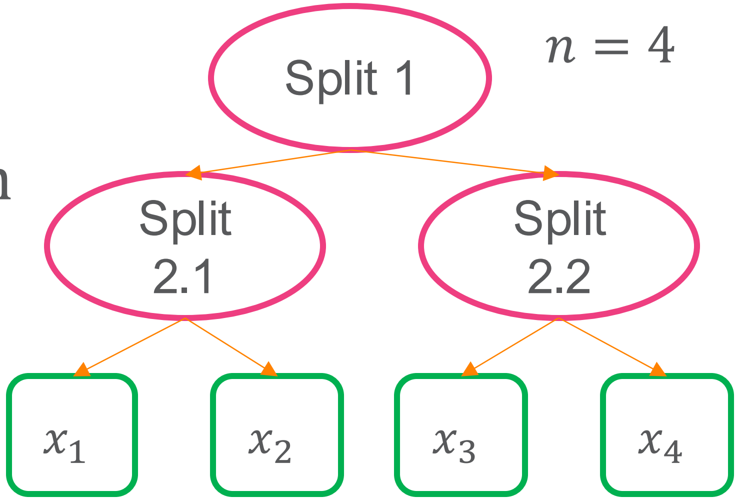


$$\left.\begin{array}{l} \mathcal{D}_0 \text{ at } Child_0 : x_j = 0\ \forall \langle x, y\rangle \in \mathcal{D} \\ \mathcal{D}_1 \text{ at } Child_1 : x_j = 1\ \forall \langle x, y\rangle \in \mathcal{D} \end{array}\right\} x_j < 0.5$$

        return $Node\left(x_j, GenerateTree(\mathcal{D}_0), GenerateTree(\mathcal{D}_1)\right)$

Time Complexity of Growing Tree (Learning)

- Computing split nodes for a perfectly balanced binary tree:
  - The total number of nodes in a binary tree is $2n-1$
  - The depth of a binary tree is $\log_2 n$
  - The number of leaf nodes at the bottom is $2^{(\log_2 n)} = \mathrm{n}$
    - Each training example is a leaf node
  - The number of split nodes is $2n-1-n=n-1$



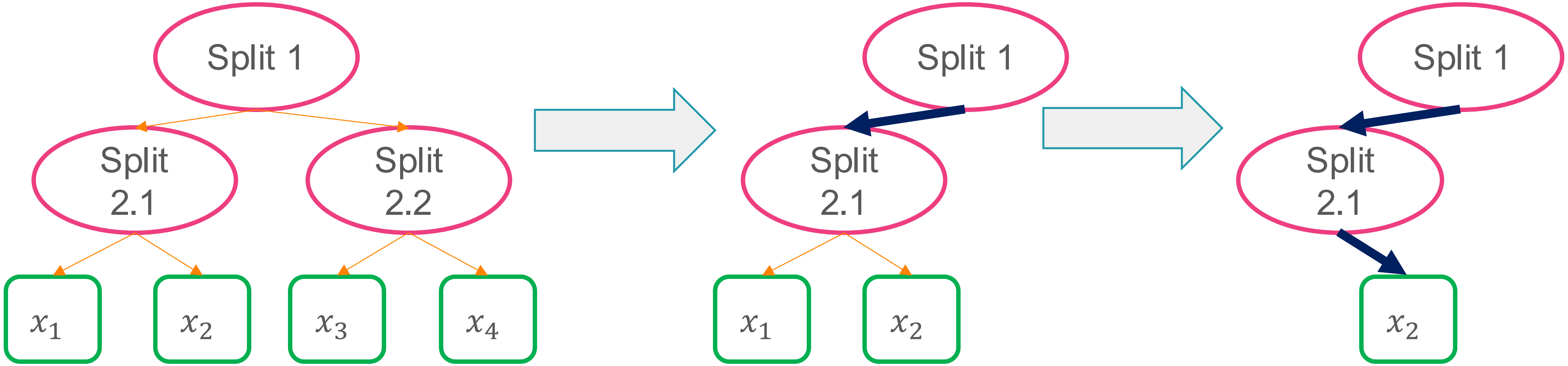
- Complexity computation
  - Sorting a feature $\mathcal{O}(n\log(n))$
  - We have m features, then, sorting all features takes $\mathcal{O}(mn\log(n))$
  - Perform operations 1 and 2 for $n-1$ split nodes, then, growing tree takes $\mathcal{O}(mn^2\log(n))$

Time Complexity for Predictions

Depth of tree

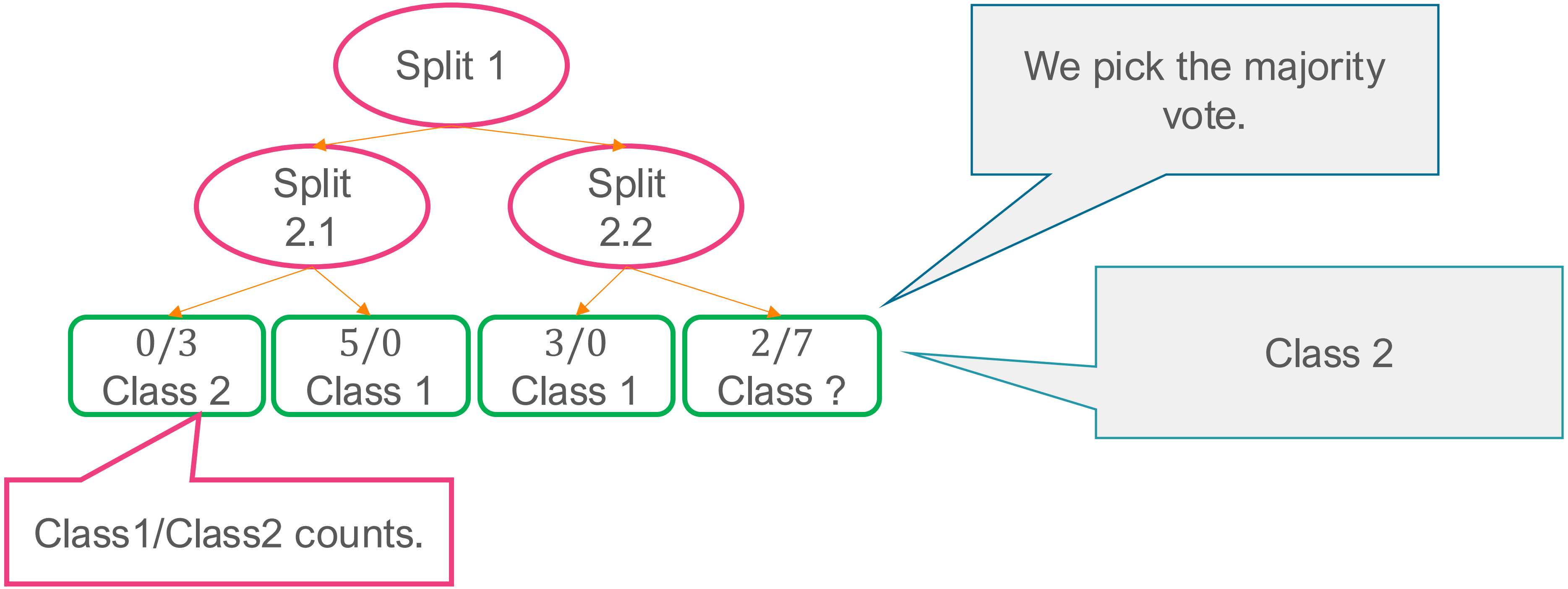
$$Steps = \log_2 n$$

$n = 4$



A prediction takes $O(\log(n))$

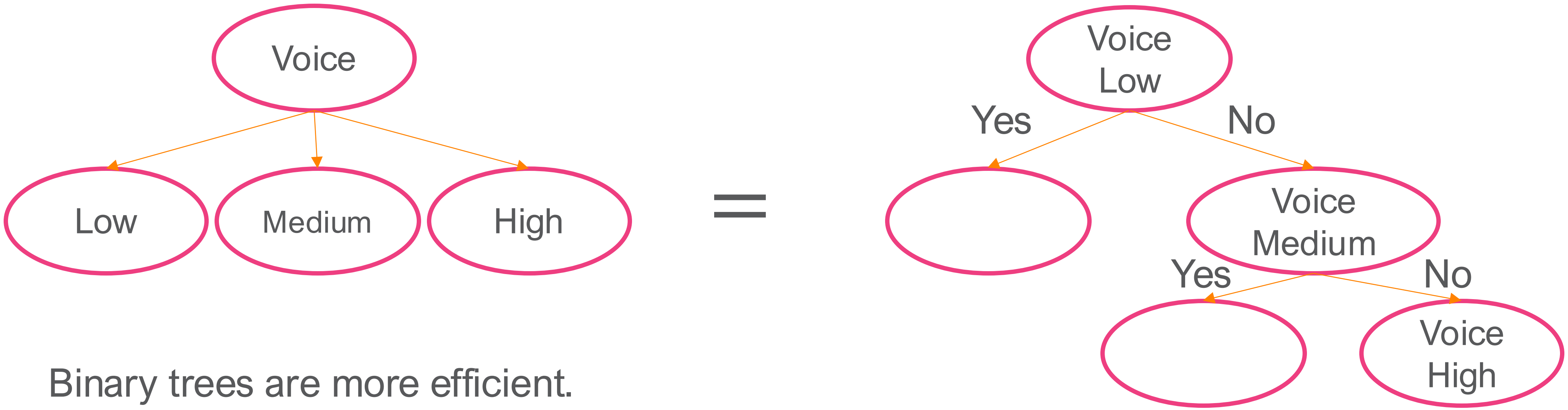
How to handle decisions at non-pure leaf nodes?



Binary vs Categorical

Voice Pitch: {Low, Medium, High}

One-Hot Encoding

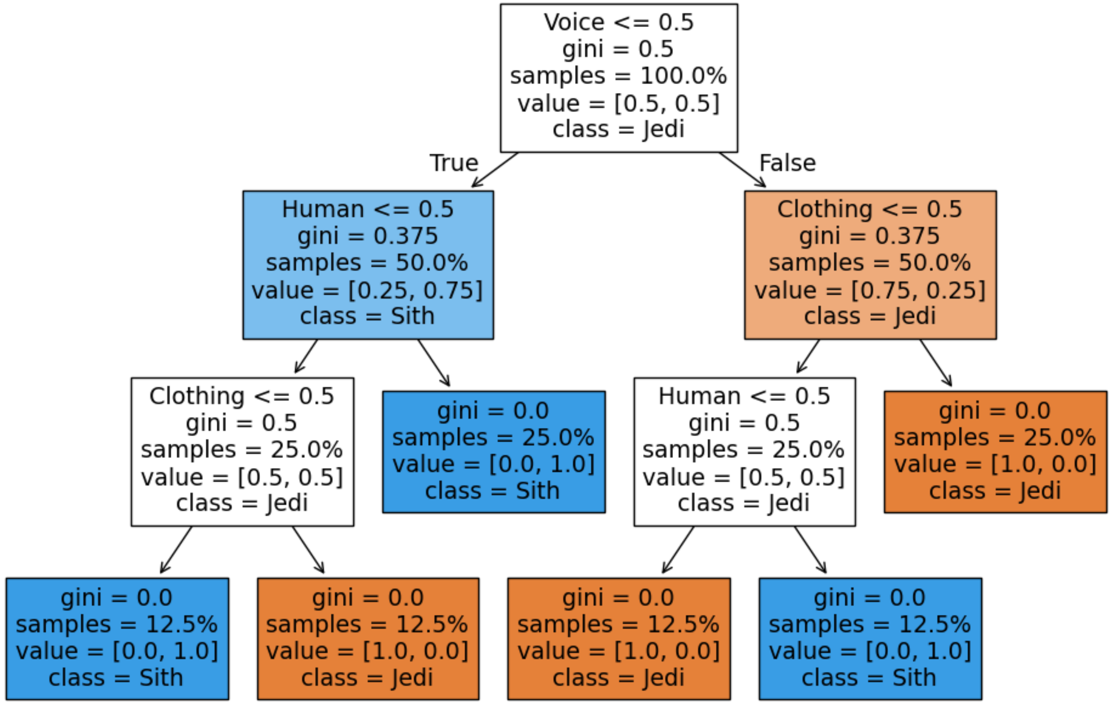


Binary trees are more efficient.

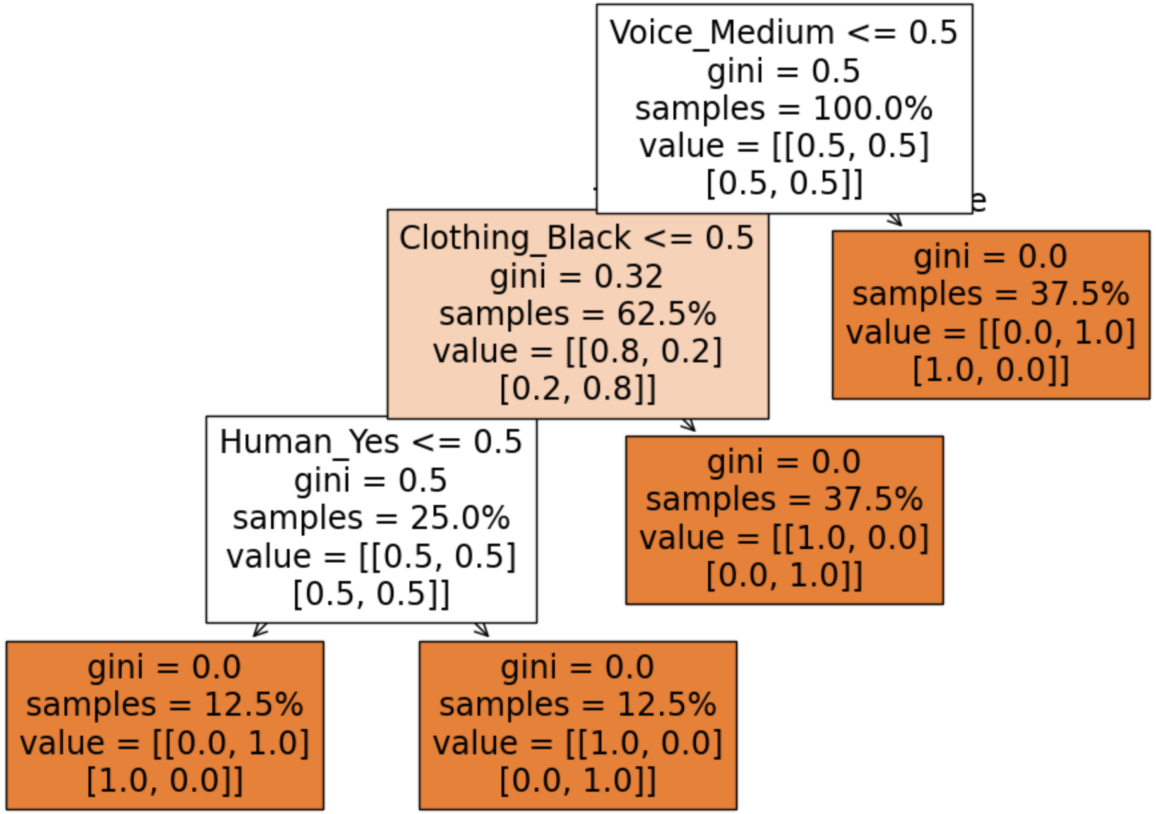
Similar to linear regression: Add a new feature per category (i.e., new columns in X).

Including Categorical Values

Binary Voice



Categorical Voice



When to stop growing the tree?

- Node is pure
  - Leaf node contains only examples of the same class
- x_j feature values are the same for all examples
- Statistical significance test
  - E.g., Chi-Square: Are parent and child class distributions significantly different?

So far…

- General Steps:
  1. Find a feature that offers the **“best”** split
  2. Stop when a split is ***pure*** (i.e., elements for the same class)
  3. Otherwise, go to Step 1 for each split subset

- Can we exhaustively search for these splits?
  – No, too many combinations
  – E.g., potential thresholds, existing features, # of samples, categories.

- Is there a most efficient way to find the splits?
  – Divide and Conquer Algorithms (e.g., quicksort, timsort)
  – Information gain maximization

Splitting Criteria

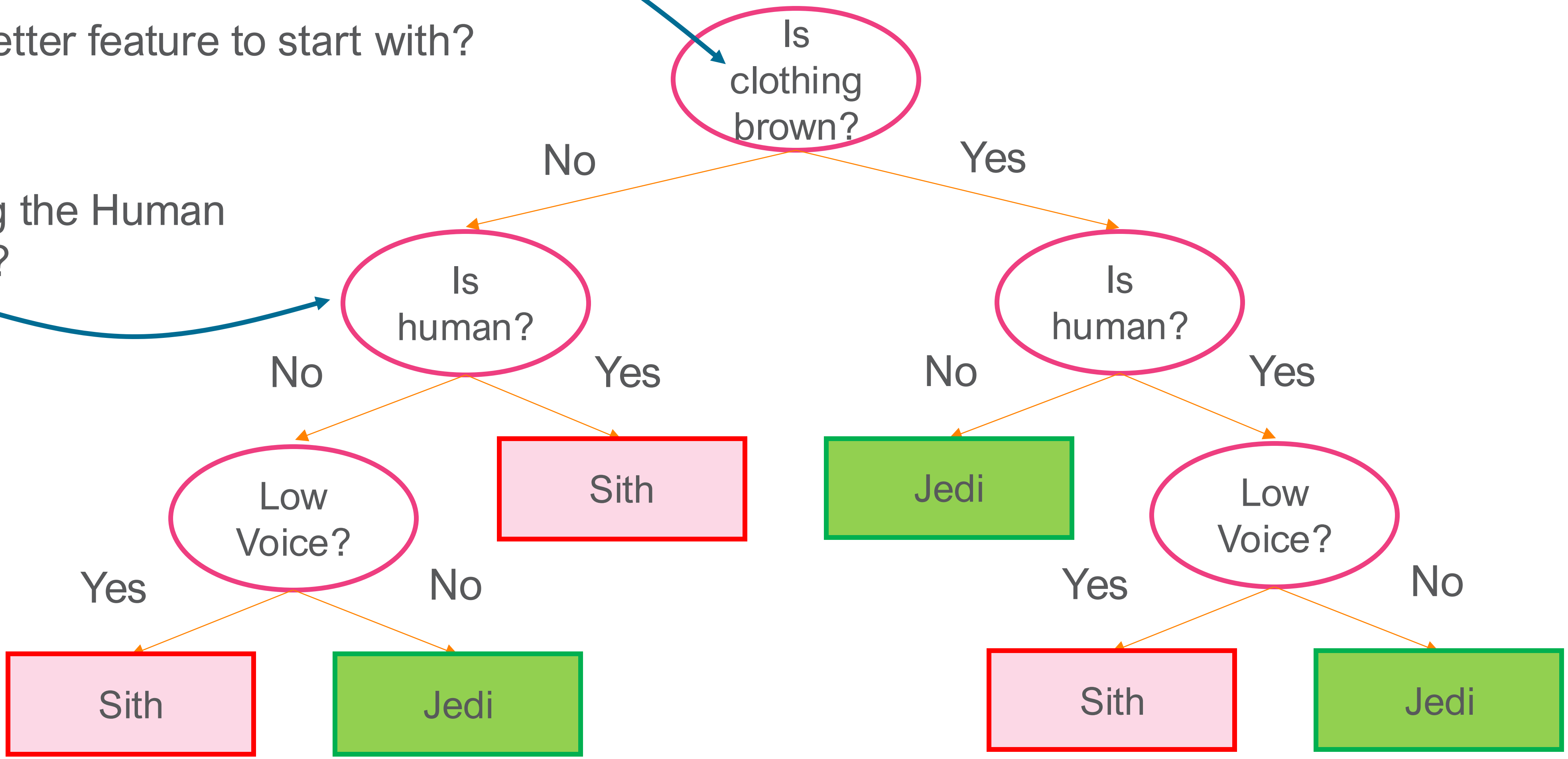
THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

https://ttpoll.com/p/643115

Jedi/Sith Tree

Was this the better feature to start with?

What about selecting the Human feature at this depth?



Choosing the “*best*” attribute

- **Key problem:** choosing which attribute to split a given set of examples
- Some possibilities are:
  - Random: Select any attribute at random
  - Least-Values: Choose the attribute with the smallest number of possible values
  - Most-Values: Choose the attribute with the largest number of possible values
  - Max-Gain: Choose the attribute that has the largest expected information gain
    - i.e., the attribute that results in the smallest expected size of the subtrees rooted at its children

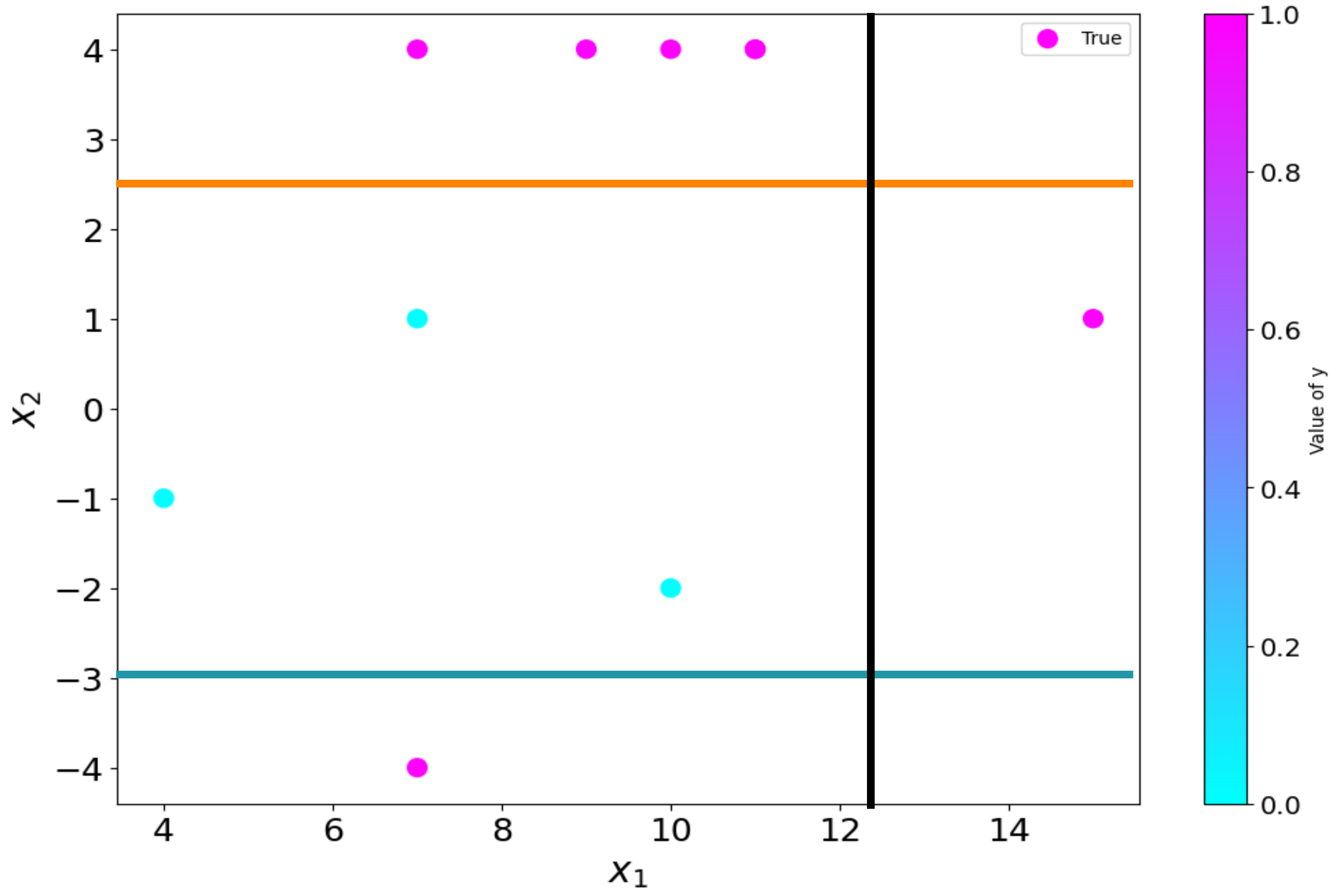
Slide Credit: Dr. Schumman

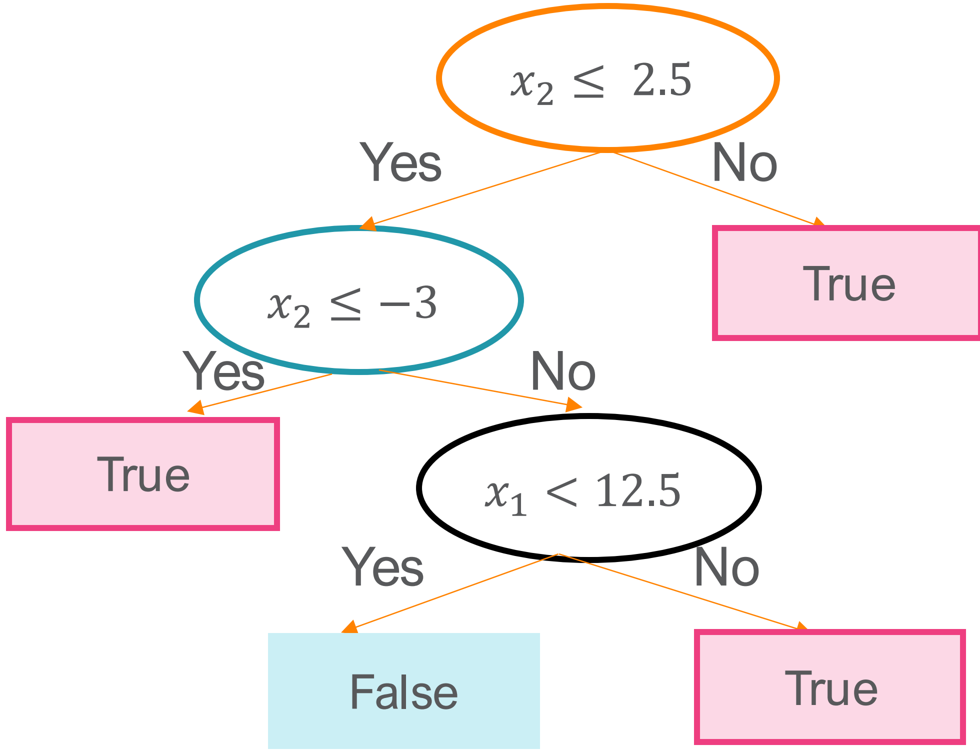
Splits Example

x1	x2	y
6	0	FALSE
13	2	TRUE
15	0	TRUE
9	0	FALSE
4	4	TRUE
14	-1	TRUE
12	0	FALSE
12	-1	FALSE
5	1	FALSE
7	-3	TRUE

https://ttpoll.com/p/643115

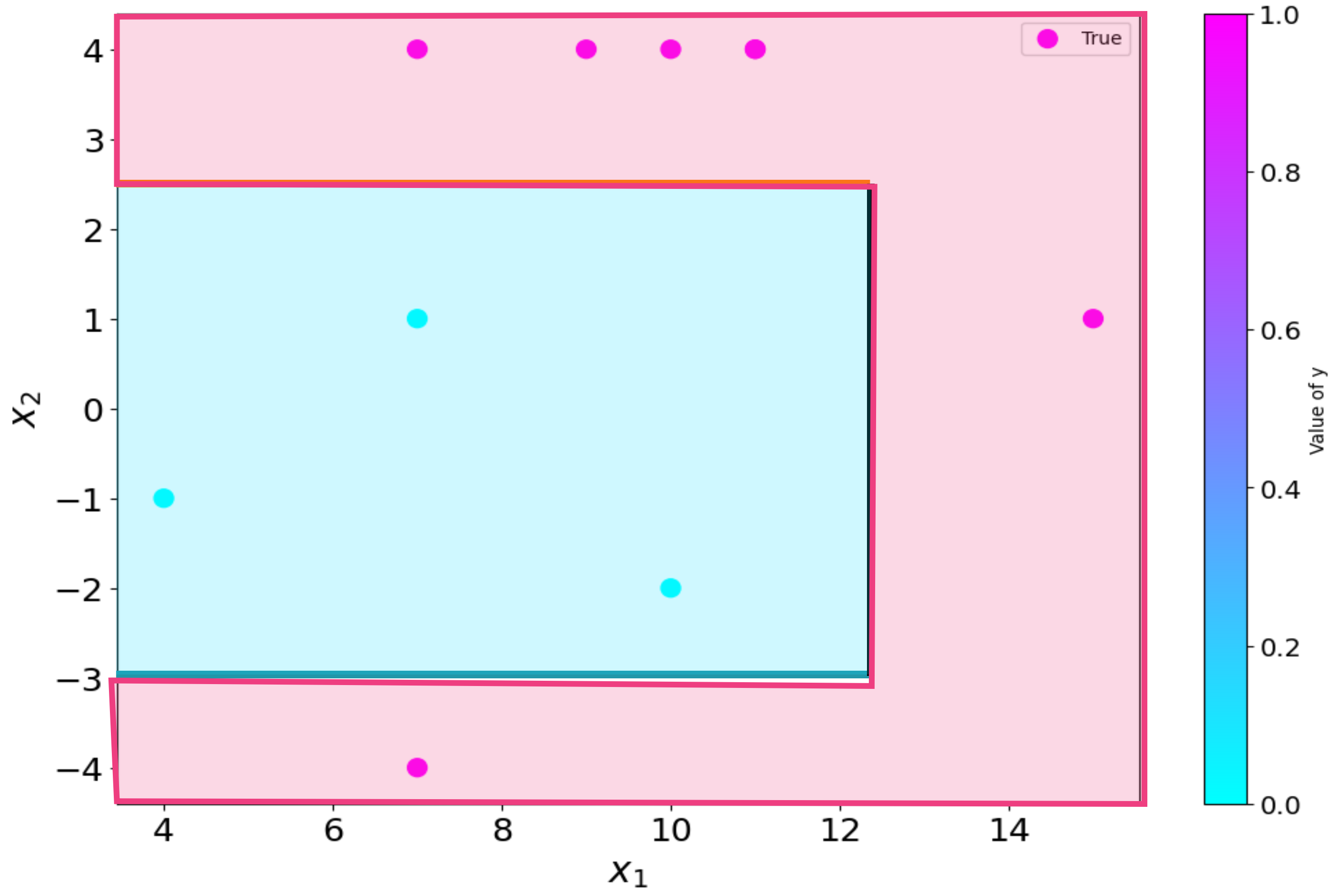
Step 1: Plot Samples

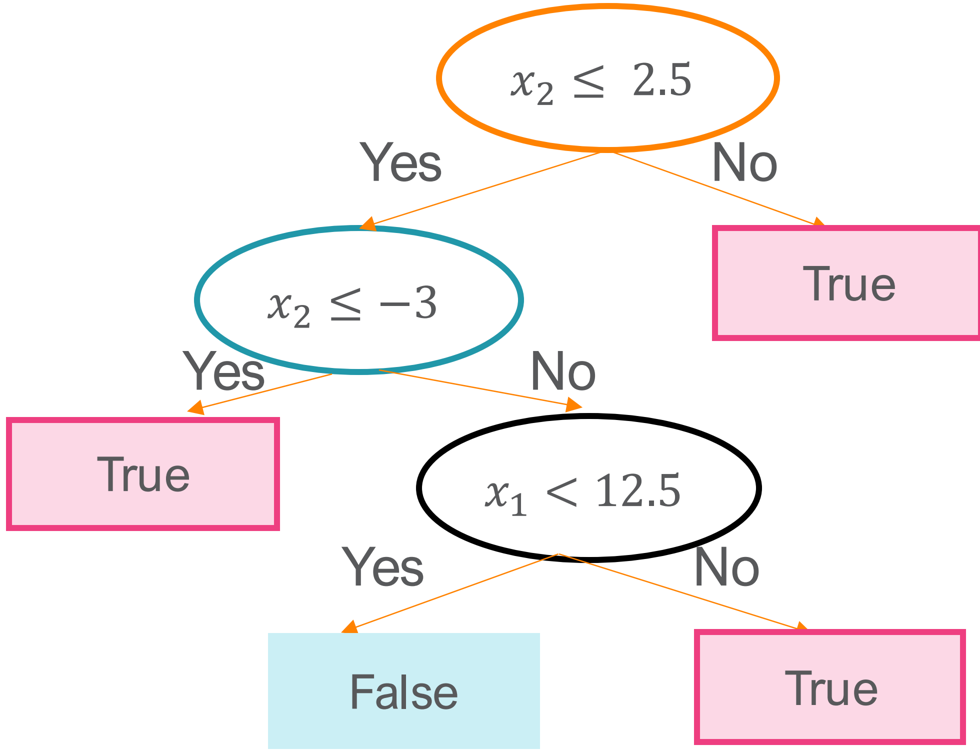




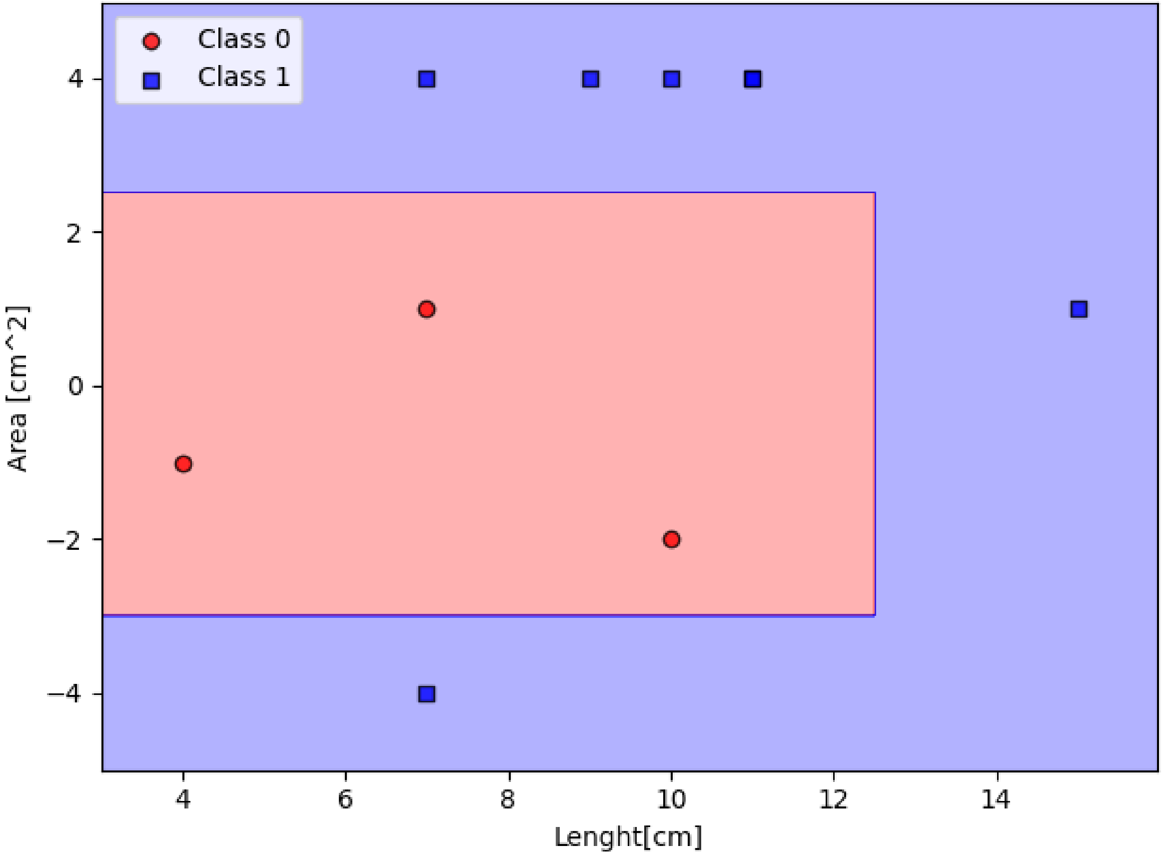
THE UNIVERSITY OF TENNESSEE KNOXVILLE

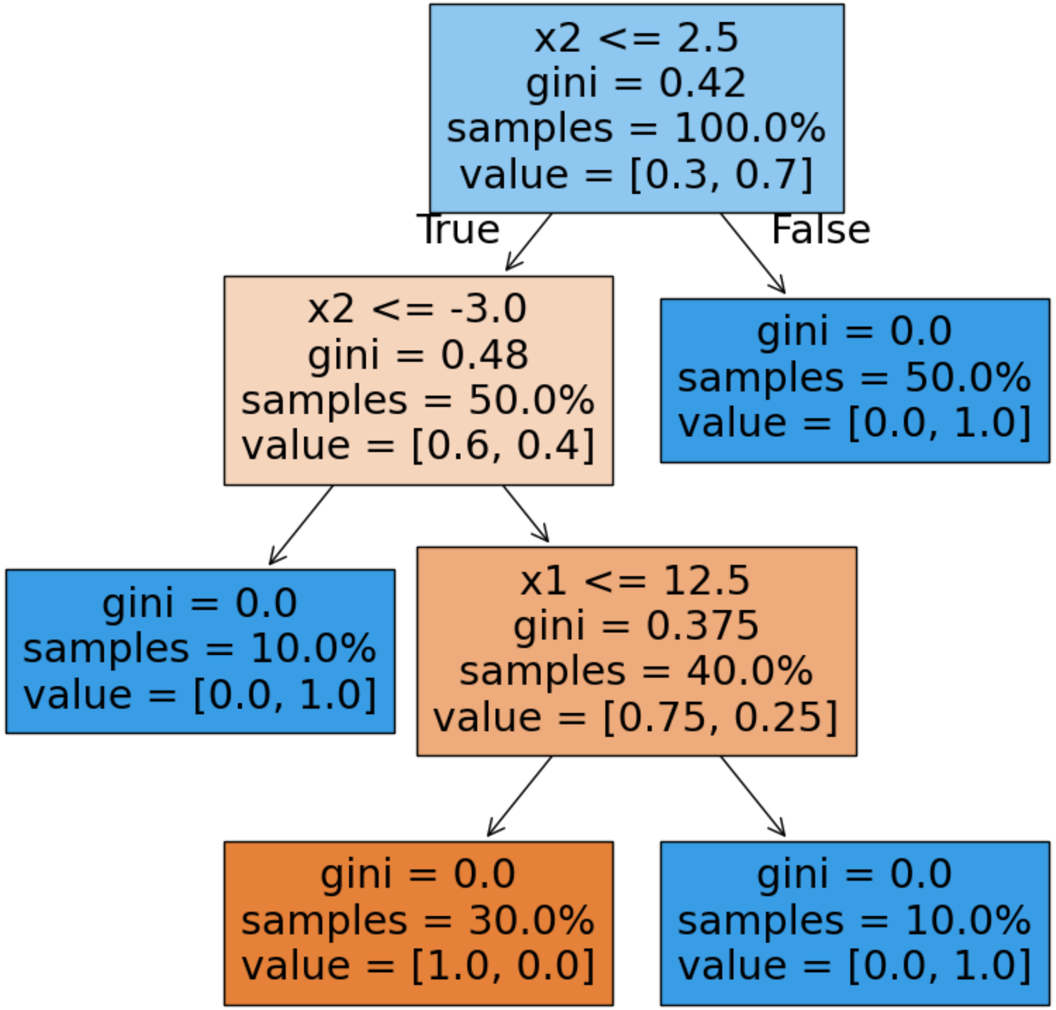
Step 1: Plot Samples





Sklearn Output





Information Gain

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

f: Feature to split

D_p: dataset of parent node

D_j: dataset of child node j

I: Impurity measurement

N_p: Number of training examples for parent node

N_j: Number of training examples for children node j

Information Gain: Binary Tree

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j) \quad = I(D_p) - \frac{N_{Left}}{N_p} I(D_{Left}) - \frac{N_{Right}}{N_p} I(D_{Right})$$

Intuition (Assume $0 \leq I(D) \leq 1$ ***):***

If there is no information gain, then, $IG(D_p, f) = 0$:

$$I(D_p) = \frac{N_{Left}}{N_p} I(D_{Left}) + \frac{N_{Right}}{N_p} I(D_{Right})$$

If there is information gain, then, $IG(D_p, f) > 0$:

$$I(D_p) > \frac{N_{Left}}{N_p} I(D_{Left}) + \frac{N_{Right}}{N_p} I(D_{Right})$$

Impurity Metrics

- Entropy (I_H):
  - Attempts to maximize mutual information.
  - How much knowledge about y we gain from knowing split D_j?
- Gini (I_G):
  - Minimizes the probability of misclassification
  - Produces very similar results to Entropy.
- Classification Error (I_E):
  - Less sensitive to changes in the node class distribution
  - Useful when pruning the tree

Impurity Metrics

- **Entropy (I_H) - Shannon**
- Gini (I_G)
- Classification Error (I_E)

$$I_H(t) = -\sum_{i=1}^{c} p(i|t)\log_2\big(p(i|t)\big)$$

$p(i|t)$: Proportion of the samples in node t that belong to class i.

Binary tree:

$$\begin{aligned} I_H(t) &= -p(1|t)\log_2\big(p(1|t)\big) - p(0|t)\log_2\big(p(0|t)\big) \\ &= -p(1|t)\log_2\big(p(1|t)\big) - \big(1-p(1|t)\big)\log_2\big(1-p(1|t)\big) \\ &= -p\log_2(p) - (1-p)\log_2(1-p) \end{aligned}$$

Note: $p(1|t) = 1 - p(0|t)$

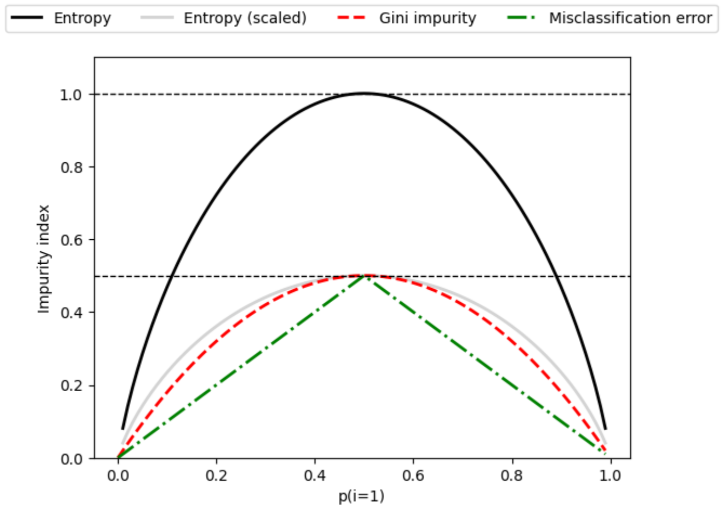
Impurity Metrics

- **Entropy (I_H)**
- Gini (I_G)
- Classification Error (I_E)

Binary tree:

$$I_H(t) = -p\log_2(p) - (1-p)\log_2(1-p)$$

p: Proportion of the samples in node ***t*** that belong to the ***True*** class.



Intuition:

- All samples in node t belong to the ***True*** class

$$I_H(t) = -1\cdot(0) - (1-1)\cdot(\infty) = 0$$

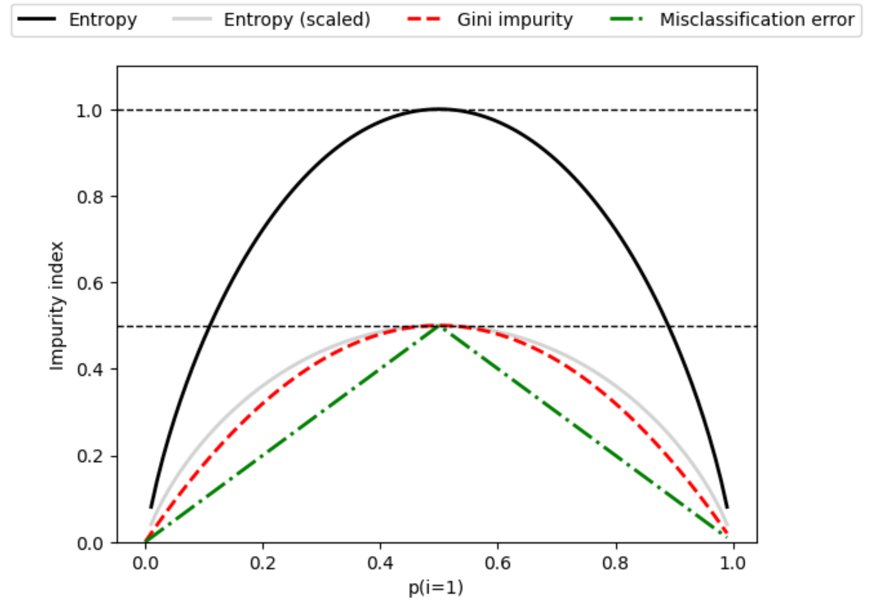
- Equal number of samples per class

$$I_H(t) = -0.5\cdot(-1) - (1-0.5)\cdot(-1) = 1$$

Impurity Metrics

- Entropy (I_H)
- **Gini (I_G)**
- Classification Error (I_E)

$$I_G(t) = \sum_{i=1}^{c} p(i|t)\big(1 - p(i|t)\big) = 1 - \sum_{i=1}^{c} p(i|t)^2$$



Binary tree:

$$\begin{aligned} I_G(t) &= 1 - p(1|t)^2 - p(0|t)^2 \\ &= 1 - p(1|t)^2 - \big(1 - p(1|t)\big)^2 \\ &= -2(p^2 - p) \end{aligned}$$

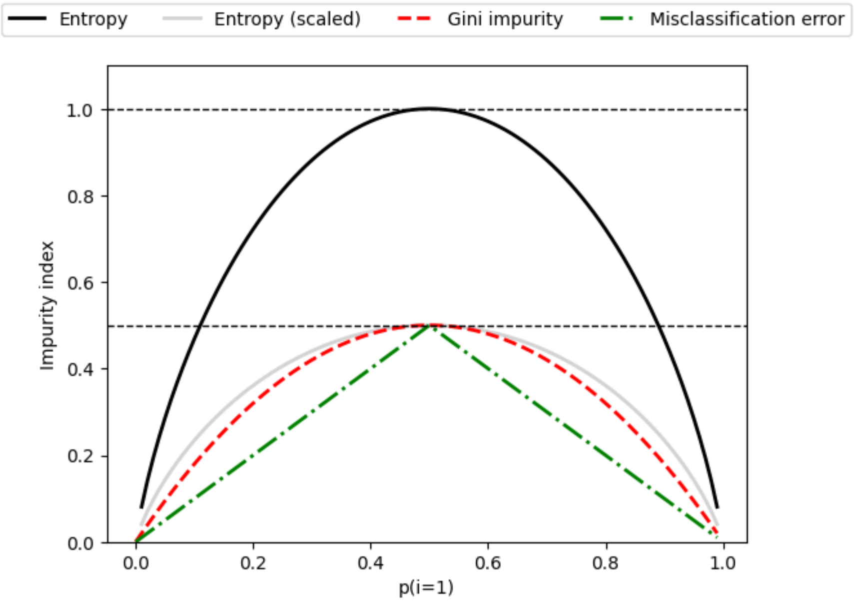
Impurity Metrics

- Entropy (I_H)
- Gini (I_G)
- **Classification Error (I_E)**

$$I_E = 1 - \max_{i \in c}\{p(i|t)\}$$

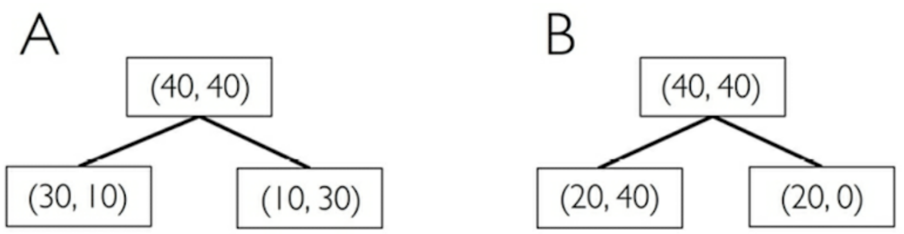
Binary tree:

$$I_E = 1 - \max\{p, 1 - p\}$$



Less sensitive to class differences.

Error vs Entropy/Gini



Error:

$$I_E(D_p) = 1 - 0.5 = 0.5$$

$$A: \quad I_E(D_{left}) = 1 - \frac{3}{4} = 0.25$$

$$A: \quad I_E(D_{right}) = 1 - \frac{3}{4} = 0.25$$

$$\boxed{A: \quad IG_E = 0.5 - \frac{4}{8}0.25 - \frac{4}{8}0.25 = 0.25}$$

$$B: \quad I_E(D_{left}) = 1 - \frac{4}{6} = \frac{1}{3}$$

$$B: \quad I_E(D_{right}) = 1 - 1 = 0$$

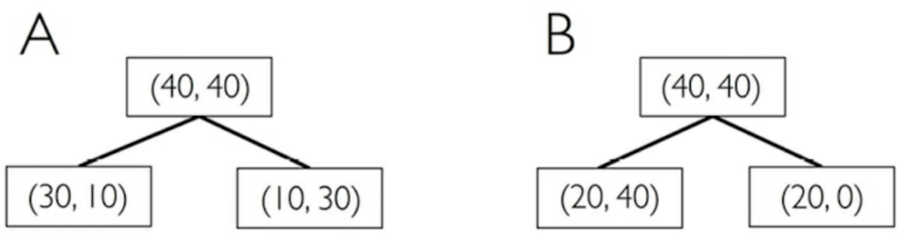
$$\boxed{B: \quad IG_E = 0.5 - \frac{6}{8} \times \frac{1}{3} - 0 = 0.25}$$

Equal Impurity

Image source: Raschka, “Machine Learning Book”

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Error vs Entropy/Gini



Error:

$$I_E(D_p) = 1 - 0.5 = 0.5$$

$$A: \quad I_E(D_{left}) = 1 - \frac{3}{4} = 0.25$$

$$A: \quad I_E(D_{right}) = 1 - \frac{3}{4} = 0.25$$

$$A: \quad IG_E = 0.5 - \frac{4}{8}0.25 - \frac{4}{8}0.25 = \boxed{0.25}$$

$$B: \quad I_E(D_{left}) = 1 - \frac{4}{6} = \frac{1}{3}$$

$$B: \quad I_E(D_{right}) = 1 - 1 = 0$$

$$B: \quad IG_E = 0.5 - \frac{6}{8} \times \frac{1}{3} - 0 = \boxed{0.25}$$

Gini:

$$I_G(D_p) = 1 - (0.5^2 + 0.5^2) = 0.5$$

$$A: \quad I_G(D_{left}) = 1 - \left(\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right) = \frac{3}{8} = 0.375$$

$$A: \quad I_G(D_{right}) = 1 - \left(\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2\right) = \frac{3}{8} = 0.375$$

$$A: \quad IG_G = 0.5 - \frac{4}{8}0.375 - \frac{4}{8}0.375 = \boxed{0.125}$$

$$B: \quad I_G(D_{left}) = 1 - \left(\left(\frac{2}{6}\right)^2 + \left(\frac{4}{6}\right)^2\right) = \frac{4}{9} = 0.\bar{4}$$

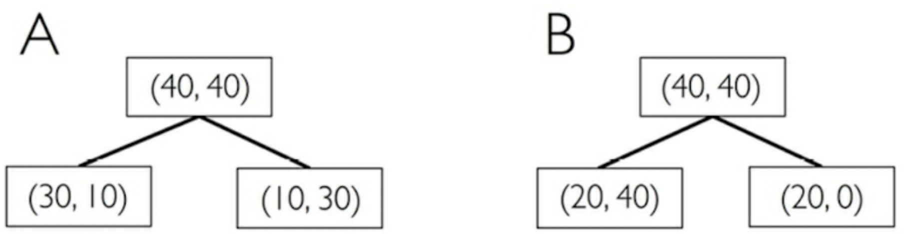
$$B: \quad I_G(D_{right}) = 1 - (1^2 + 0^2) = 0$$

$$B: \quad IG_G = 0.5 - \frac{6}{8}0.\bar{4} - 0 = \boxed{0.1\bar{6}}$$

Greater Purity

Image source: Raschka, “Machine Learning Book”

Error vs Entropy/Gini



Error:

$I_E(D_p) = 1 - 0.5 = 0.5$

A: $I_E(D_{left}) = 1 - \frac{3}{4} = 0.25$

A: $I_E(D_{right}) = 1 - \frac{3}{4} = 0.25$

A: $IG_E = 0.5 - \frac{4}{8}0.25 - \frac{4}{8}0.25 = \boxed{0.25}$

B: $I_E(D_{left}) = 1 - \frac{4}{6} = \frac{1}{3}$

B: $I_E(D_{right}) = 1 - 1 = 0$

B: $IG_E = 0.5 - \frac{6}{8} \times \frac{1}{3} - 0 = \boxed{0.25}$

Gini:

$I_G(D_p) = 1 - (0.5^2 + 0.5^2) = 0.5$

A: $I_G(D_{left}) = 1 - \left(\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right) = \frac{3}{8} = 0.375$

A: $I_G(D_{right}) = 1 - \left(\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2\right) = \frac{3}{8} = 0.375$

A: $IG_G = 0.5 - \frac{4}{8}0.375 - \frac{4}{8}0.375 = \boxed{0.125}$

B: $I_G(D_{left}) = 1 - \left(\left(\frac{2}{6}\right)^2 + \left(\frac{4}{6}\right)^2\right) = \frac{4}{9} = 0.\bar{4}$

B: $I_G(D_{right}) = 1 - (1^2 + 0^2) = 0$

B: $IG_G = 0.5 - \frac{6}{8}0.\bar{4} - 0 = \boxed{0.1\bar{6}}$

Entropy:

$I_H(D_p) = -(0.5\log_2(0.5) + 0.5\log_2(0.5)) = 1$

A: $I_H(D_{left}) = -\left(\frac{3}{4}\log_2\left(\frac{3}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right) = 0.81$

A: $I_H(D_{right}) = -\left(\frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{3}{4}\log_2\left(\frac{3}{4}\right)\right) = 0.81$

A: $IG_H = 1 - \frac{4}{8}0.81 - \frac{4}{8}0.81 = \boxed{0.19}$

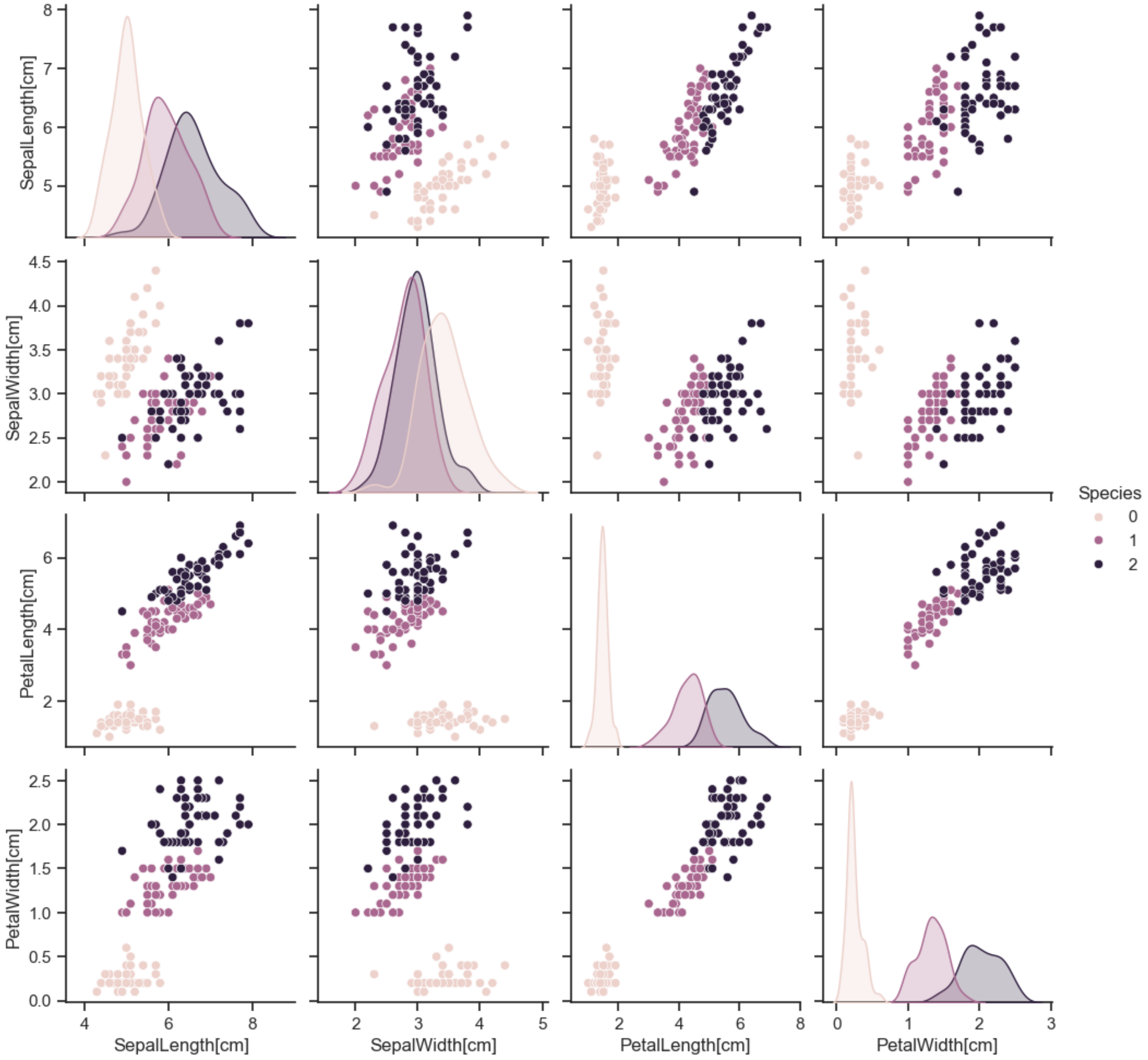
B: $I_H(D_{left}) = -\left(\frac{2}{6}\log_2\left(\frac{2}{6}\right) + \frac{4}{6}\log_2\left(\frac{4}{6}\right)\right) = 0.92$

B: $I_H(D_{right}) = 0$

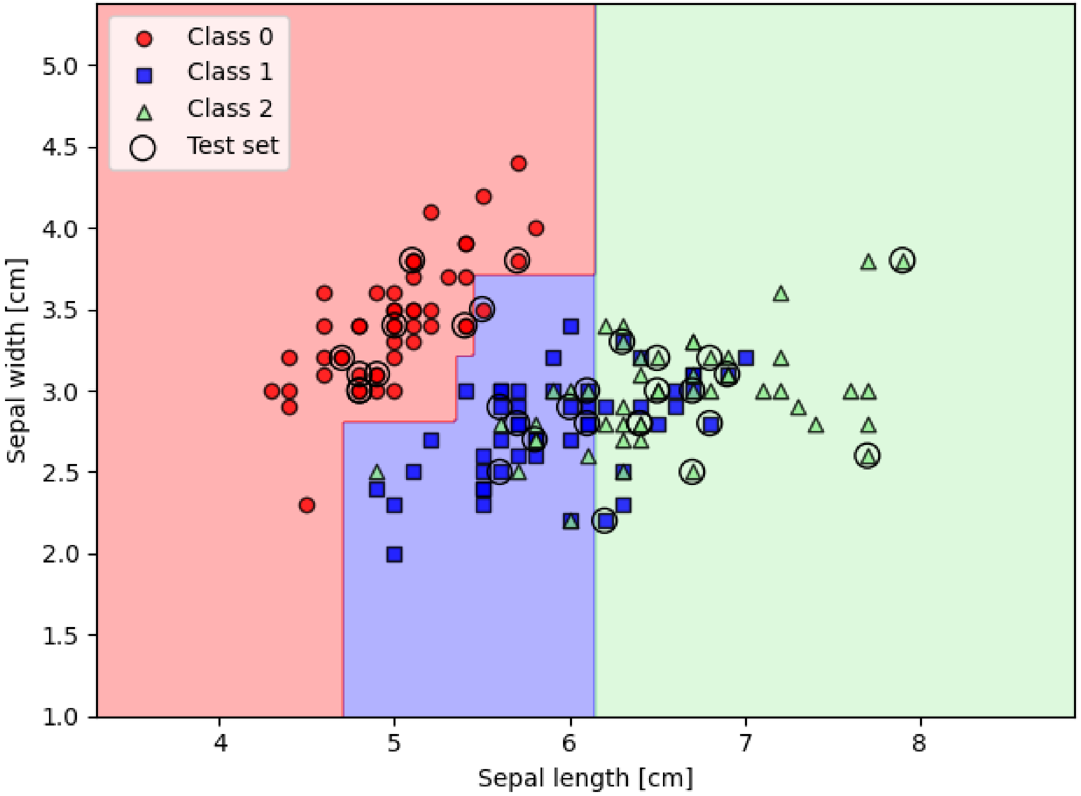
B: $IG_H = 1 - \frac{6}{8}0.92 - 0 = \boxed{0.31}$

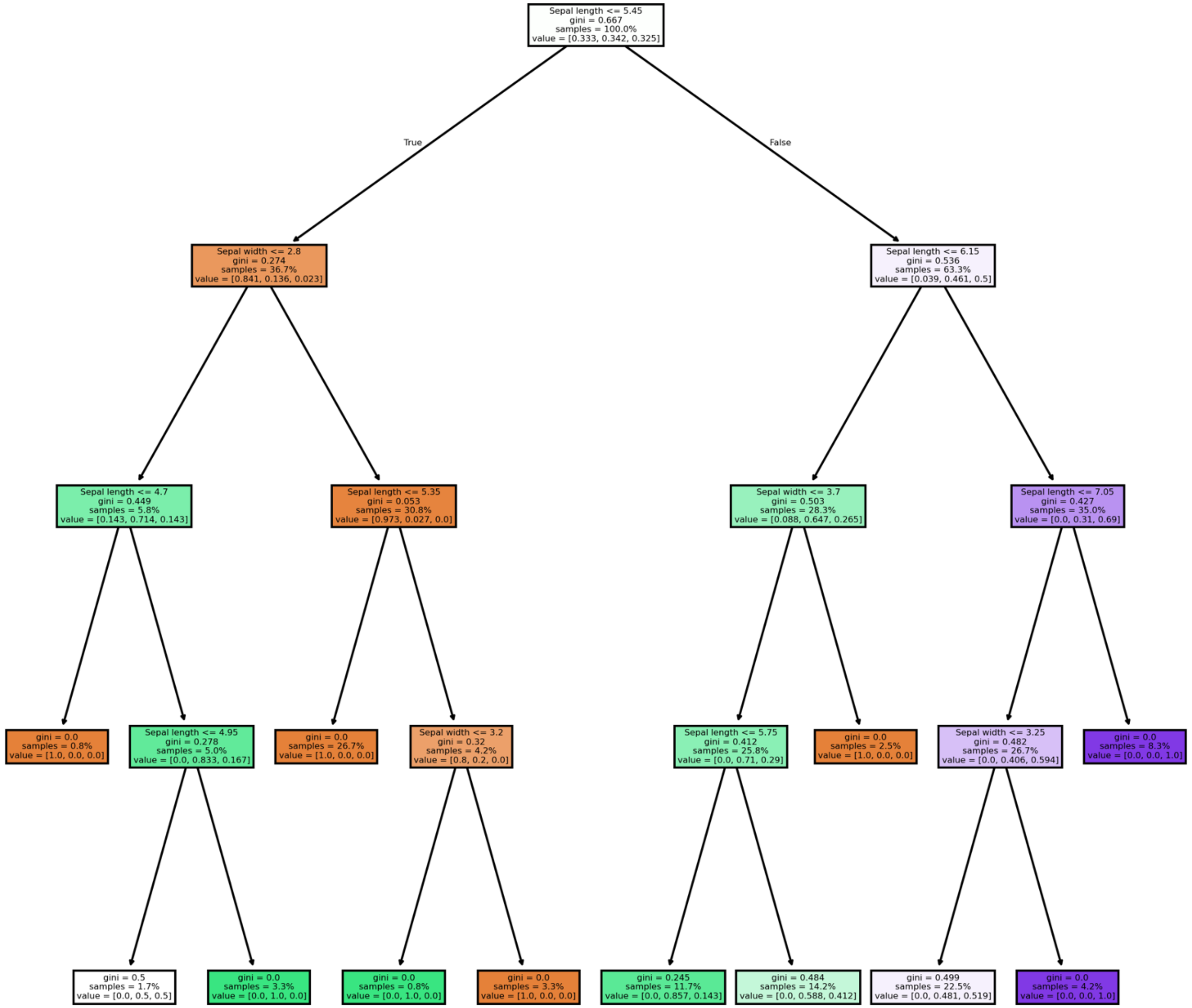
Image source: Raschka, “Machine Learning Book”

Demo with Iris dataset

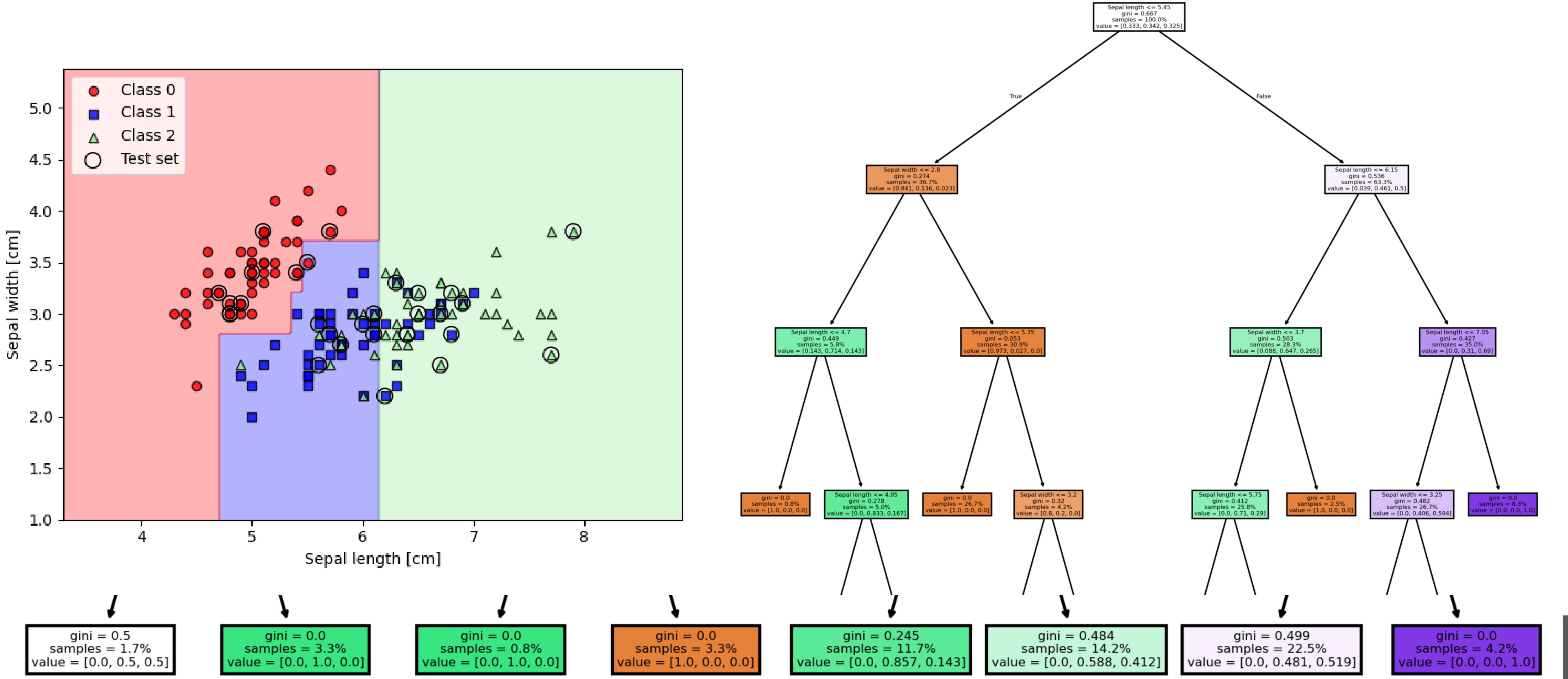


Demo with Iris dataset: Sepal Width and Length



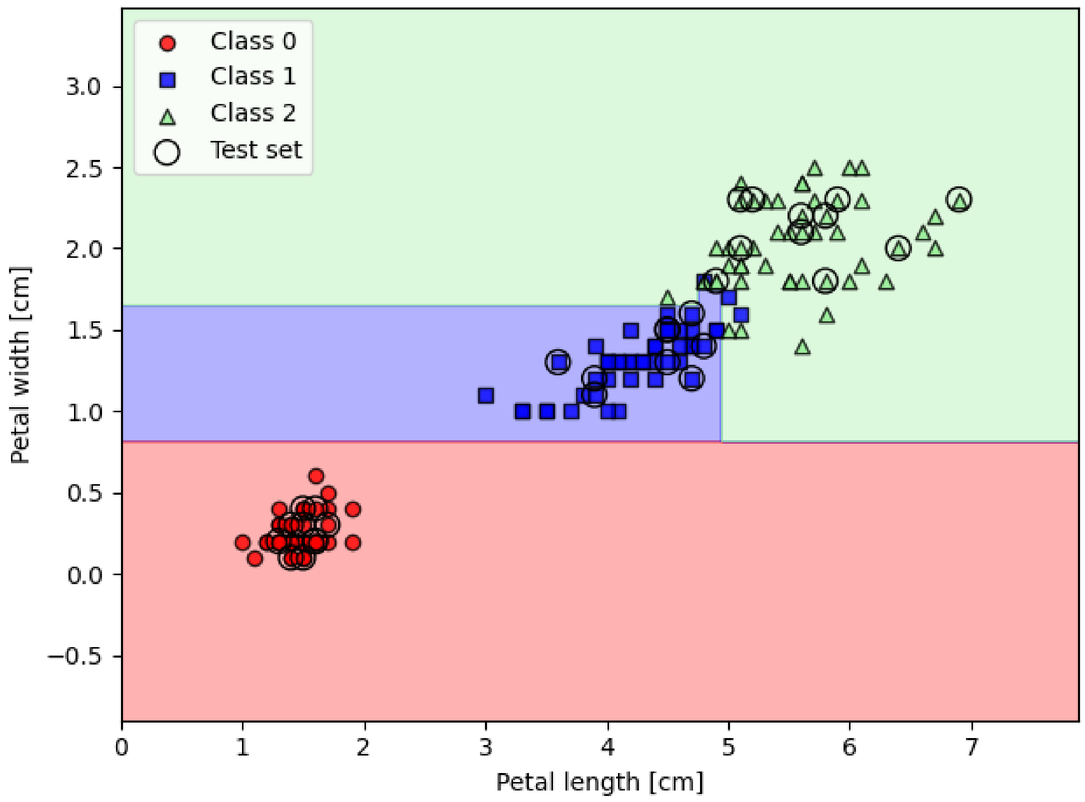


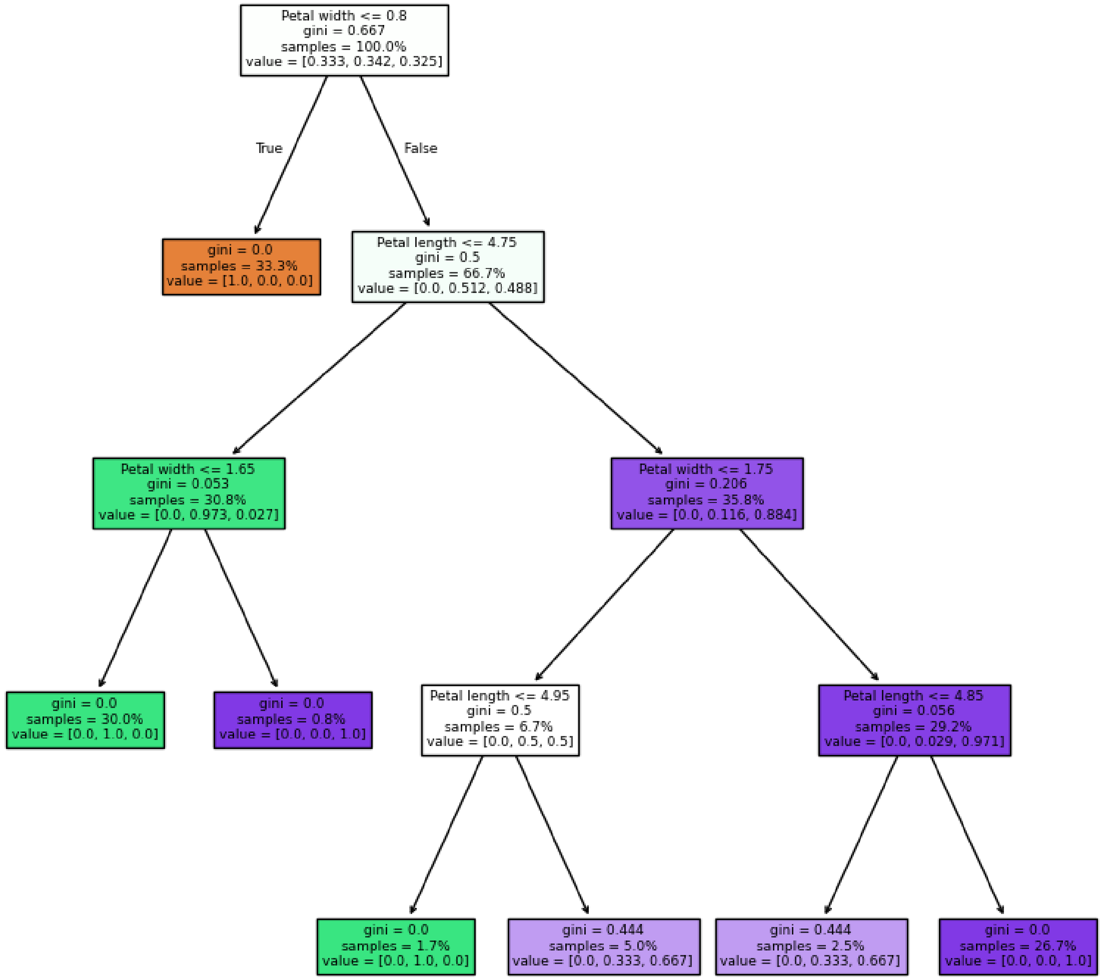
Demo with Iris dataset: Sepal Width and Length



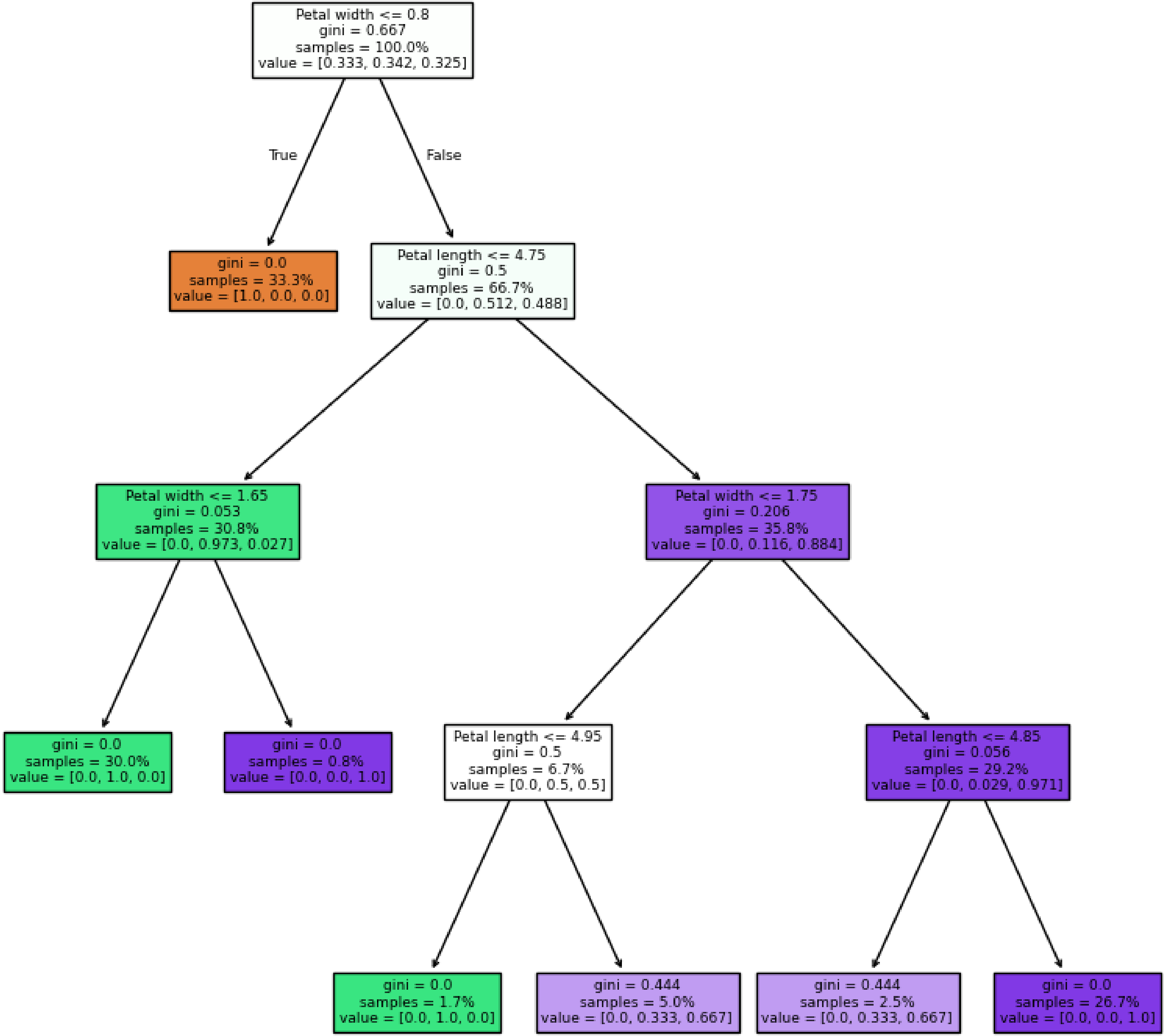
https://ttpoll.com/p/643115

Demo with Iris dataset: Petal Width and Length





Demo with Iris dataset: All Features



ID3 – Iterative Dichotomizer

- Early algorithm proposed by Quinlan, 1986.
- Cannot handle numeric values
- Prone to overfitting (no pruning)
- Produce short and wide trees
- Maximize information gain by minimizing entropy
- Support discrete features, binary and multi-category features

$$GainRatio(\mathcal{D}, f) = \frac{Gain(\mathcal{D}, f)}{SplitInfo(\mathcal{D}, f)}$$

$$SplitInfo(\mathcal{D}, f) = -\sum_{v \in f} \frac{|\mathcal{D}_v|}{|\mathcal{D}|} \log_2\left(\frac{|\mathcal{D}_v|}{|\mathcal{D}|}\right)$$

Measures entropy of the feature variable instead of the classes.

C4.5

- Continuous and discrete features, Quinlan 1993.
- Continuous is very expensive, because must consider all possible ranges
- Handles missing attributes (ignores them in gain compute)
- Post-pruning (bottom-up pruning)
- Gain Ratio stop criteria

CART

- **C**lassification **A**nd **R**egression **T**rees proposed by Breiman 1984.
- Handles continuous and discrete features
- Strictly uses binary splits (taller trees than ID3, C4.5)
- Trees produce better results that ID3 and C4.5 but are harder to interpret
- Tree growth
  - Variance reduction in regression trees
  - Gini impurity, also known as twoing.
- Cost complexity pruning