

# Lecture 8: Decision Trees and Overfitting

<https://tinyurl.com/cis5190-9-28-2022>

Osbert Bastani and Zachary G. Ives

CIS 4190/5190 – Fall 2022

# Tasks

- Homework 2 due October 3 8pm
- Project team member submission: due October 4, 8pm

# Recall from Last Time

**Two kinds of nonparametric learning:** k-Nearest Neighbor and Decision Trees

Decision tree algorithm (C4.5):

Greedy recursive algorithm: successively splits the training data into “hyperrectangles”

Assume Boolean functions as the basis of intermediate nodes

Intermediate node *splits* are chosen based on *information gain*

Basic scheme: use **entropy** as a measure of information gain

$$H(\mathcal{D}) = - \sum_c P(Y = c) \log_2 P(Y = c), \text{ for each class } c$$

$$IG(\mathcal{D}, X_j) = H(\mathcal{D}) - \sum_v H(\mathcal{D}[X_j = v])P(X_j = v) \text{ for each value } v \text{ of } X_j$$

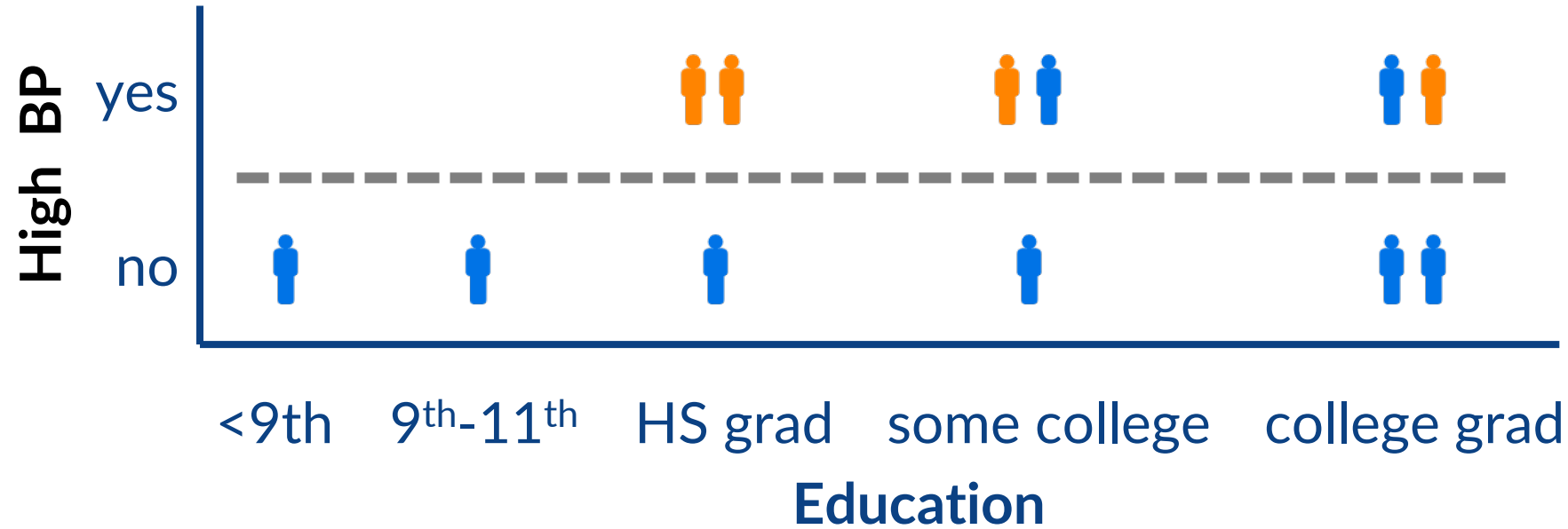
# Revisiting Diabetes Data

ID	AGE	WAIST	HE.	UPPER LEG LENGTH		BMI	RACE	HIGH BP	EDUCATION	FAMILY INCOME RATIO		DIABETIC			
				CHOLESTEROL	WEIGHT					GENDER	GLYCOHAEM				
SEQN	RIDAGEYR	BMXWAIST	BMXHT	LBXTC	BMXLEG	BMXWT	BMXBMI	RIDRETH1	BPQ020	ALQ120Q	DMDDEDUC2	RIAGENDR	INDFMPIR	LBXGH	DIABETIC
73557	69.0	100.0	171.3	167.0	39.2	78.3	26.7	Non-Hispanic Black	yes	1.0	high school graduate / GED	male	0.84	13.9	yes
73558	54.0	107.6	176.8	170.0	40.0	89.5	28.6	Non-Hispanic White	yes	7.0	high school graduate / GED	male	1.78	9.1	yes
73559	72.0	109.2	175.3	126.0	40.0	88.9	28.9	Non-Hispanic White	yes	0.0	some college or AA degree	male	4.51	8.9	yes
73562	56.0	123.1	158.7	226.0	34.2	105.0	41.7	Mexican American	yes	5.0	some college or AA degree	male	4.79	5.5	no
73564	61.0	110.8	161.8	168.0	37.1	93.4	35.7	Non-Hispanic White	yes	2.0	college graduate or above	female	5.0	5.5	no
73566	56.0	85.5	152.8	278.0	32.4	61.8	26.5	Non-Hispanic White	no	1.0	high school graduate / GED	female	0.48	5.4	no
73567	65.0	93.7	172.4	173.0	40.0	65.3	22.0	Non-Hispanic White	no	4.0	9th-11th grade	male	1.2	5.2	no
73568	26.0	73.7	152.5	168.0	34.4	47.1	20.3	Non-Hispanic White	no	2.0	college graduate or above	female	5.0	5.2	no
73571	76.0	122.1	172.5	167.0	35.5	102.4	34.4	Non-Hispanic White	yes	2.0	college graduate or above	male	5.0	6.9	yes
73577	32.0	100.0	166.2	182.0	36.5	79.7	28.9	Mexican American	no	20.0	Less than 9th grade	male	0.29	5.3	no
73581	50.0	99.3	185.0	202.0	42.8	80.9	23.6	Other or Multi-Racial	no	0.0	college graduate or above	male	5.0	5.0	no
73585	28.0	90.3	175.1	198.0	40.5	92.2	30.1	Other or Multi-Racial	no	4.0	some college or AA degree	male	2.26	5.0	no
73589	35.0	94.6	172.9	192.0	39.1	78.3	26.2	Non-Hispanic White	no	2.0	high school graduate / GED	male	1.74	5.5	no
73595	58.0	114.8	175.3	165.0	40.1	96.0	31.2	Other Hispanic	no	1.0	some college or AA degree	male	3.09	7.7	no
73596	57.0	117.8	164.7	151.0	35.3	104.0	38.3	Other or Multi-Racial	yes	1.0	college graduate or above	female	5.0	5.9	no
73600	37.0	122.9	185.1	189.0	48.1	126.2	36.8	Non-Hispanic Black	yes	2.0	high school graduate / GED	male	0.63	6.2	yes
73604	69.0	96.6	156.9	203.0	37.0	59.5	24.2	Non-Hispanic White	no	1.0	some college or AA degree	female	2.44	5.4	no
73607	75.0	130.5	169.6	161.0	36.5	111.9	38.9	Non-Hispanic White	yes	0.0	high school graduate / GED	male	1.08	5.0	no
73610	43.0	102.6	176.8	200.0	38.8	90.2	28.9	Non-Hispanic White	no	5.0	college graduate or above	male	2.03	4.9	no
73613	60.0	113.6	163.8	203.0	41.6	104.9	39.1	Non-Hispanic Black	yes	2.0	9th-11th grade	female	5.0	6.1	no
73614	55.0	90.9	167.9	256.0	43.5	60.9	21.6	Non-Hispanic White	no	0.0	high school graduate / GED	female	1.29	5.0	no
73615	65.0	100.3	145.9	166.0	30.0	55.4	26.0	Other Hispanic	yes	1.0	Less than 9th grade	female	1.22	6.3	yes

# Information Gain Example for Diabetes

## First Split

ID (SEQN)	HIGH_BP (BPQ020)	EDUCATION (DMDEDUC2)	DIABETIC
73557	yes	high school graduate / GED	yes
73558	yes	high school graduate / GED	yes
73559	yes	some college or AA degree	yes
73562	yes	some college or AA degree	no
73564	yes	college graduate or above	no
73566	no	high school graduate / GED	no
73567	no	9th-11th grade	no
73568	no	college graduate or above	no
73571	yes	college graduate or above	yes
73577	no	Less than 9th grade	no
73581	no	college graduate or above	no
73585	no	some college or AA degree	no



We compared two candidates:

$$IG(\mathcal{D}, High\ BP) = H(\mathcal{D}) - H(\mathcal{D} | High\ BP) = 0.918 - 0.459 = 0.459$$

0.459 ★

$$IG(\mathcal{D}, Education) = H(\mathcal{D}) - H(\mathcal{D} | Education) = 0.918 - 0.730 = 0.188$$

# Discounting for Many-Valued Attributes

IG tends toward selecting features that have many values

- e.g., unique identifiers, **dates**, etc.
- unique partitions → minimal impurity



Gain Ratio scales by entropy of the sub-dataset proportions:

$$\text{GainRatio}(\mathcal{D}, X_j) = \frac{IG(\mathcal{D}, X_j)}{\text{SplitInfo}(\mathcal{D}, X_j)}$$

This scales by the entropy of the split itself, ignoring the classes

$$\text{SplitInfo}(\mathcal{D}, X_j) = - \sum_v P(X_j = v) \log_2 P(X_j = v)$$

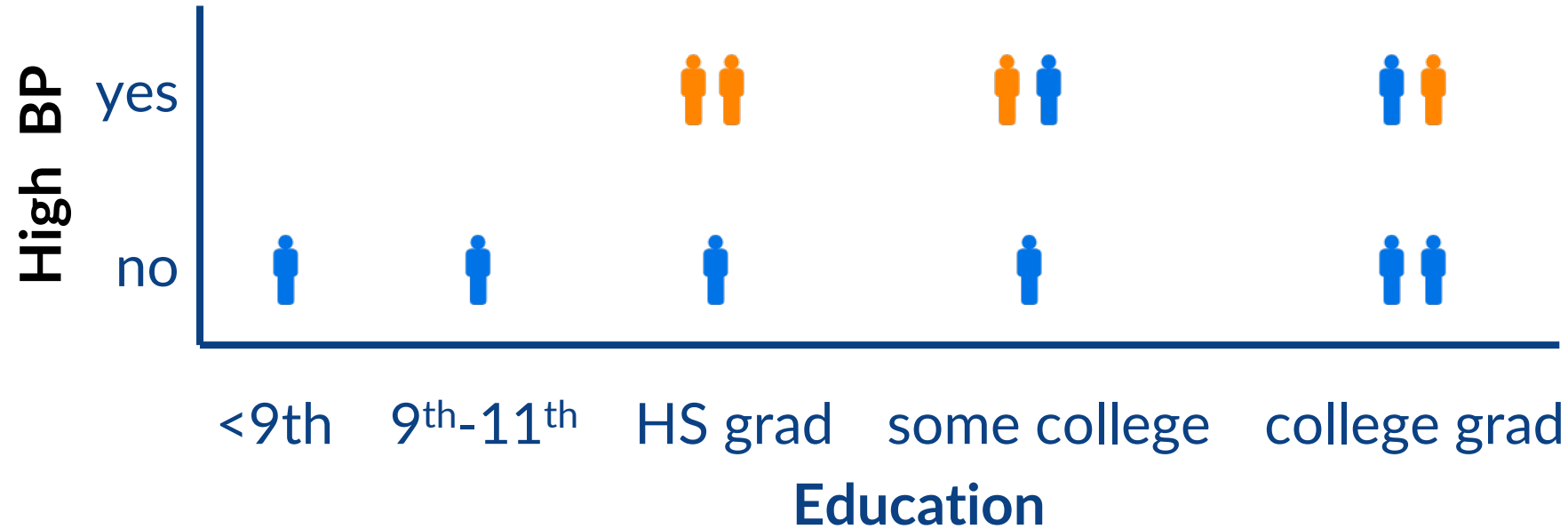
aka Intrinsic Information

$$\frac{|\mathcal{D}[X_j = v]|}{|\mathcal{D}|}$$

# Gain Ratio Example

Already Computed:

- $H(\mathcal{D}) = 0.918$
- $H(\mathcal{D} \mid \text{High BP}) = 0.459$
- $H(\mathcal{D} \mid \text{Education}) = 0.730$
- $IG(\mathcal{D} \mid \text{High BP}) = 0.459$
- $IG(\mathcal{D}, \text{Education}) = 0.188$



Need to compute:

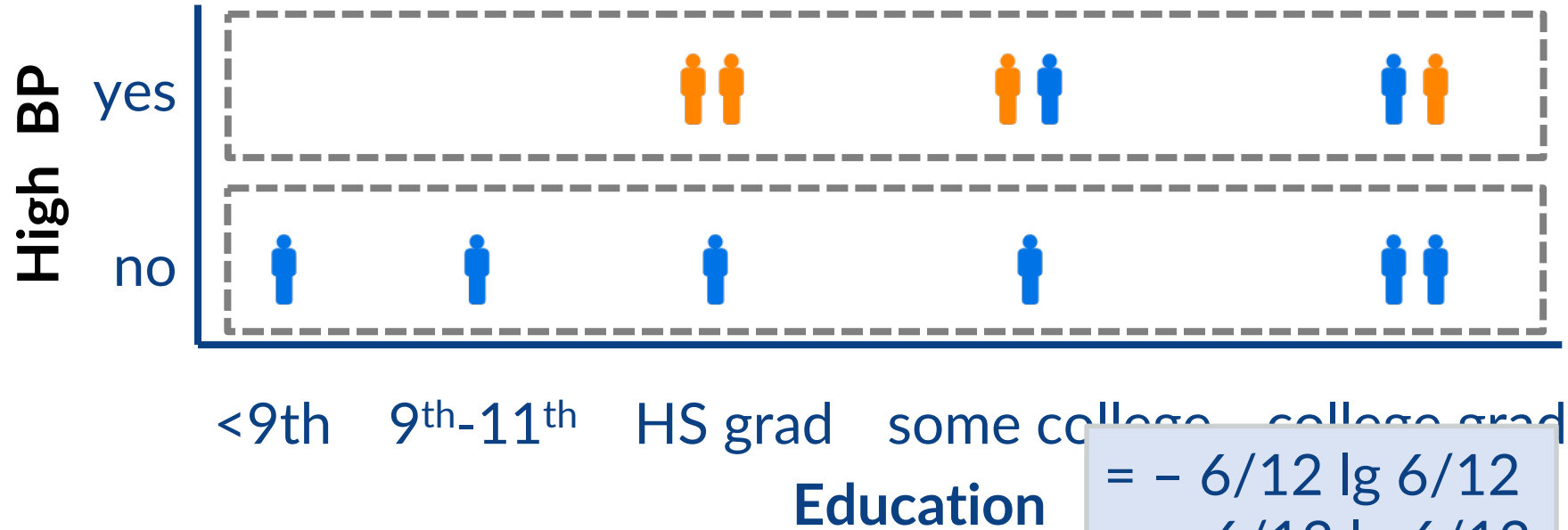
$$\text{GainRatio}(\mathcal{D}, \text{High BP}) = IG(\mathcal{D}, \text{High BP}) / \text{SplitInfo}(\mathcal{D}, \text{High BP})$$

$$\text{GainRatio}(\mathcal{D}, \text{Education}) = IG(\mathcal{D}, \text{Education}) / \text{SplitInfo}(\mathcal{D}, \text{Education})$$

# Gain Ratio Example

Already Computed:

- $H(\mathcal{D}) = 0.918$
- $H(\mathcal{D} \mid \text{High BP}) = 0.459$
- $H(\mathcal{D} \mid \text{Education}) = 0.730$
- $IG(\mathcal{D} \mid \text{High BP}) = 0.459$
- $IG(\mathcal{D}, \text{Education}) = 0.188$



$$\begin{aligned}
 &= -6/12 \lg 6/12 \\
 &\quad - 6/12 \lg 6/12 \\
 &= 1
 \end{aligned}$$

Need to compute:

$$\text{GainRatio}(\mathcal{D} \mid \text{High BP}) = \frac{IG(\mathcal{D}, \text{High BP})}{\text{SplitInfo}(\mathcal{D}, \text{High BP})}$$

$$\text{GainRatio}(\mathcal{D}, \text{Education}) = \frac{IG(\mathcal{D}, \text{Education})}{\text{SplitInfo}(\mathcal{D}, \text{Education})}$$

0.459

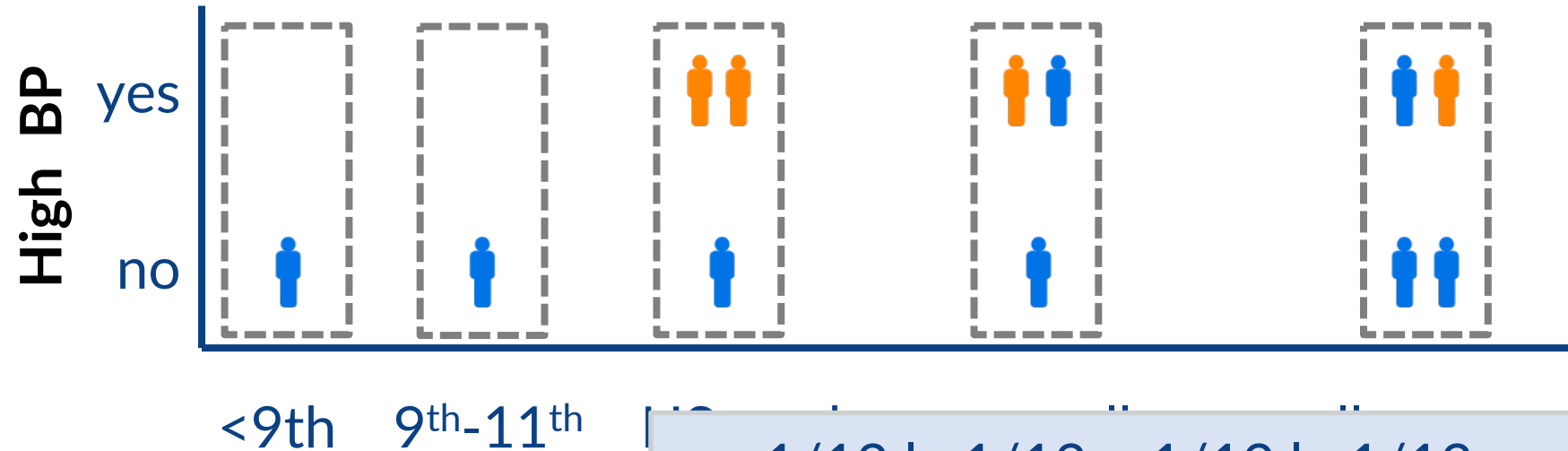
0.188



# Gain Ratio Example

Already Computed:

- $H(\mathcal{D}) = 0.918$
- $H(\mathcal{D} \mid \text{High BP}) = 0.459$
- $H(\mathcal{D} \mid \text{Education}) = 0.730$
- $IG(\mathcal{D} \mid \text{High BP}) = 0.459$
- $IG(\mathcal{D}, \text{Education}) = 0.188$



Need to compute:

$\text{GainRatio}(\mathcal{D} \mid \text{High BP}) = IG(\mathcal{D}, \text{High BP}) / \text{SplitInfo}(\mathcal{D}, \text{High BP})$

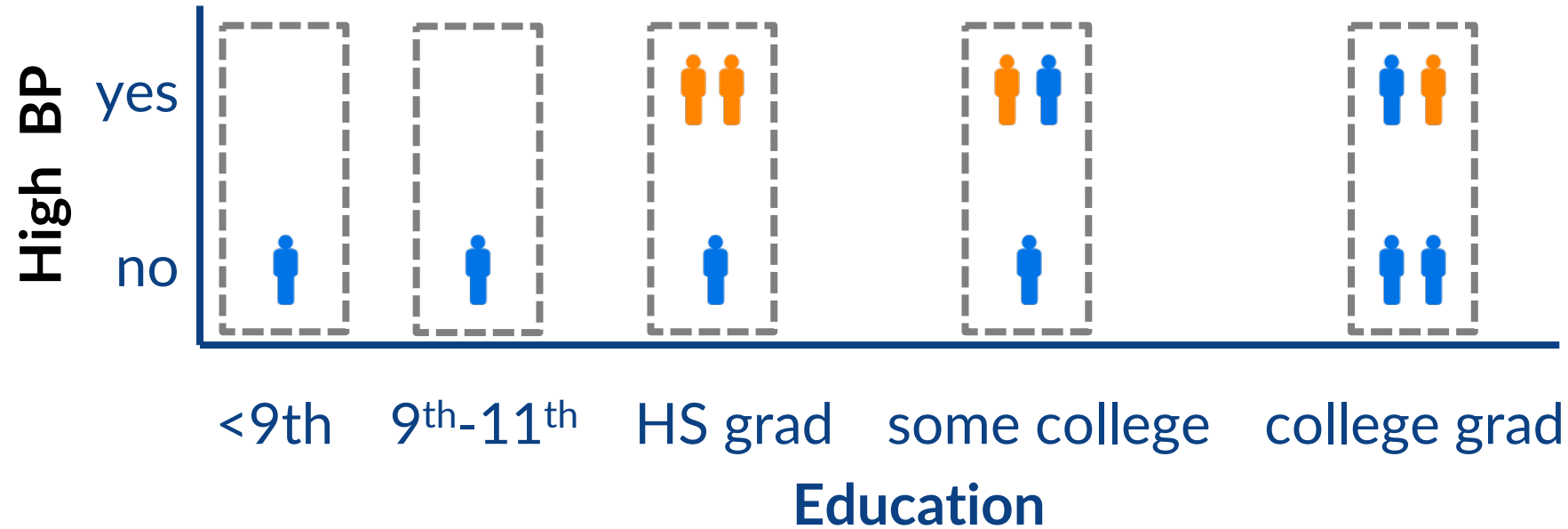
$\text{GainRatio}(\mathcal{D}, \text{Education}) = IG(\mathcal{D}, \text{Education}) / \text{SplitInfo}(\mathcal{D}, \text{Education})$

$$\begin{aligned}
 &= -1/12 \lg 1/12 - 1/12 \lg 1/12 \\
 &\quad - 3/12 \lg 3/12 - 3/12 \lg 3/12 \\
 &\quad - 4/12 \lg 4/12 \\
 &= 2.1258
 \end{aligned}$$

# Gain Ratio Example

Already Computed:

- $H(\mathcal{D}) = 0.918$
- $H(\mathcal{D} \mid \text{High BP}) = 0.459$
- $H(\mathcal{D} \mid \text{Education}) = 0.730$
- $IG(\mathcal{D} \mid \text{High BP}) = 0.459$
- $IG(\mathcal{D}, \text{Education}) = 0.188$



Need to compute:

$$\text{GainRatio}(\mathcal{D} \mid \text{High BP}) = IG(\mathcal{D}, \text{High BP}) / \text{SplitInfo}(\mathcal{D}, \text{High BP}) = 0.459 / 1 = \mathbf{0.459}$$

$$\text{GainRatio}(\mathcal{D}, \text{Education}) = IG(\mathcal{D}, \text{Education}) / \text{SplitInfo}(\mathcal{D}, \text{Education}) = 0.188 / 2.12 = 0.089$$

*Same as before.... But much stronger preference for BP!*

# Gain Ratio vs “Standard” Information Gain

Gain ratio is **Information Gain** scaled discounted by the **Intrinsic Information** of the split itself

→ Biases against many-valued splits, which otherwise may have less impurity simply due to size

Adds a bit of extra computational overhead, so it is not *always* used – but it can be helpful in many real-world use cases!

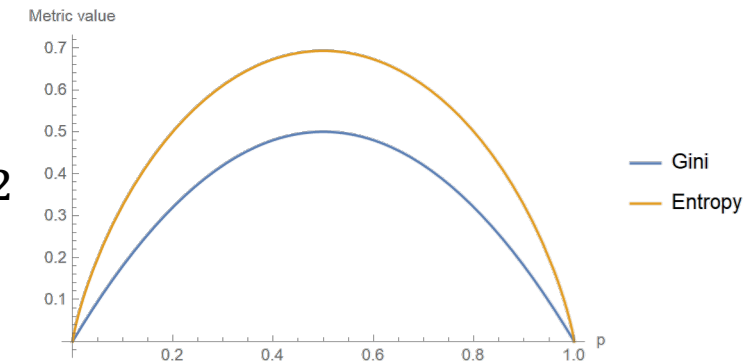
# Another Alternative: The Gini Index

Issue: choosing a split point by IG is a bit expensive – logarithm is an expensive float operation

Popular alternative to entropy: **Gini index**, produces similar results and is less expensive to compute

- Measures how often a randomly chosen element from a set would be incorrectly labeled, if it was **randomly labeled according to the distribution of labels in the subset**
- Like entropy, ranges from 0 – 1 with max value at 50%

$$Gini(p) = \sum_{i=1}^K p_i(1 - p_i) = 1 - \sum_{i=1}^K p_i^2$$



used in one common decision-tree algorithm (CaRT) and in SciKit-Learn

# Feature Scaling in Decision Trees

Decision trees are generally univariate -- split **one feature (dimension) at a time**

While this limits

They are **scale invariant**, i.e., we don't need to standardize the scale!

# **DT Training for Diabetes**

# We are Ready to Train the DT for Diabetes!

SEQN	RIDAGEYR	BMXWAIST	BMXHT	LBXTC	BMXLEG	BMXWT	BMXBMI	RIDRETH1	BPQ020	ALQ120Q	DMDDEDUC2	RIAGENDR	INDFMPIR	LBXGH	DIABETIC
73557	69.0	100.0	171.3	167.0	39.2	78.3	26.7	Non-Hispanic Black	yes	1.0	high school graduate / GED	male	0.84	13.9	yes
73558	54.0	107.6	176.8	170.0	40.0	89.5	28.6	Non-Hispanic White	yes	7.0	high school graduate / GED	male	1.78	9.1	yes
73559	72.0	109.2	175.3	126.0	40.0	88.9	28.9	Non-Hispanic White	yes	0.0	some college or AA degree	male	4.51	8.9	yes
73562	56.0	123.1	158.7	226.0	34.2	105.0	41.7	Mexican American	yes	5.0	some college or AA degree	male	4.79	5.5	no
73564	61.0	110.8	161.8	168.0	37.1	93.4	35.7	Non-Hispanic White	yes	2.0	college graduate or above	female	5.0	5.5	no
73566	56.0	85.5	152.8	278.0	32.4	61.8	26.5	Non-Hispanic White	no	1.0	high school graduate / GED	female	0.48	5.4	no
73567	65.0	93.7	172.4	173.0	40.0	65.3	22.0	Non-Hispanic White	no	4.0	9th-11th grade	male	1.2	5.2	no
73568	26.0	73.7	152.5	168.0	34.4	47.1	20.3	Non-Hispanic White	no	2.0	college graduate or above	female	5.0	5.2	no
73571	76.0	122.1	172.5	167.0	35.5	102.4	34.4	Non-Hispanic White	yes	2.0	college graduate or above	male	5.0	6.9	yes
73577	32.0	100.0	166.2	182.0	36.5	79.7	28.9	Mexican American	no	20.0	Less than 9th grade	male	0.29	5.3	no
73581	50.0	99.3	185.0	202.0	42.8	80.9	23.6	Other or Multi-Racial	no	0.0	college graduate or above	male	5.0	5.0	no
73585	28.0	90.3	175.1	198.0	40.5	92.2	30.1	Other or Multi-Racial	no	4.0	some college or AA degree	male	2.26	5.0	no
73589	35.0	94.6	172.9	192.0	39.1	78.3	26.2	Non-Hispanic White	no	2.0	high school graduate / GED	male	1.74	5.5	no
73595	58.0	114.8	175.3	165.0	40.1	96.0	31.2	Other Hispanic	no	1.0	some college or AA degree	male	3.09	7.7	no
73596	57.0	117.8	164.7	151.0	35.3	104.0	38.3	Other or Multi-Racial	yes	1.0	college graduate or above	female	5.0	5.9	no
73600	37.0	122.9	185.1	189.0	48.1	126.2	36.8	Non-Hispanic Black	yes	2.0	high school graduate / GED	male	0.63	6.2	yes
73604	69.0	96.6	156.9	203.0	37.0	59.5	24.2	Non-Hispanic White	no	1.0	some college or AA degree	female	2.44	5.4	no
73607	75.0	130.5	169.6	161.0	36.5	111.9	38.9	Non-Hispanic White	yes	0.0	high school graduate / GED	male	1.08	5.0	no
73610	43.0	102.6	176.8	200.0	38.8	90.2	28.9	Non-Hispanic White	no	5.0	college graduate or above	male	2.03	4.9	no
73613	60.0	113.6	163.8	203.0	41.6	104.9	39.1	Non-Hispanic Black	yes	2.0	9th-11th grade	female	5.0	6.1	no
73614	55.0	90.9	167.9	256.0	43.5	60.9	21.6	Non-Hispanic White	no	0.0	high school graduate / GED	female	1.29	5.0	no
73615	65.0	100.3	145.9	166.0	30.0	55.4	26.0	Other Hispanic	yes	1.0	Less than 9th grade	female	1.22	6.3	yes
73616	62.0	95.5	172.8	171.0	38.4	71.8	24.0	Non-Hispanic White	no	2.0	some college or AA degree	female	5.0	5.5	no
73619	36.0	91.1	173.1	162.0	38.9	81.7	27.3	Mexican American	no	2.0	high school graduate / GED	female	0.84	5.0	no
73621	80.0	98.2	176.2	161.0	40.4	76.4	24.6	Non-Hispanic White	no	5.0	college graduate or above	male	5.0	5.6	no
73622	72.0	115.6	185.4	186.0	39.7	99.5	28.9	Non-Hispanic White	no	4.0	college graduate or above	male	5.0	6.0	no

# Recall the Basic Algorithm

`function train_tree( $\mathcal{D}$ )`

1. If data  $\mathcal{D}$  **all have the same label**  $y$ , return new `leaf_node( $y$ )`, else:
2. Pick the feature  $X_j$  to partition  $\mathcal{D}$  that maximizes **Information Gain**
3. Set `node = new decision_node( $X_j$ )`
4. For each value  $v$  that  $X_j$  can take  
    Recursively create a new child `train_tree( $\mathcal{D}[X_j = v]$ )` of node
5. Return node



# Entropy-Based Greedy DT Construction

SEQN	RIDAGEYR	BMXWAIST	BMXHT	LBXTC	BMXLEG	BMXWT	BMXBMI	RIDRETH1	BPQ020	ALQ120Q	DMDEDUC2	RIAGENDR	INDFMPPIR	LBXGH	DIABETIC
73557	69.0	100.0	171.3	167.0	39.2	78.3	26.7	Non-Hispanic Black	yes	1.0	high school graduate / GED	male	0.84	13.9	yes
73558	54.0	107.6	176.8	170.0	40.0	89.5	28.6	Non-Hispanic White	yes	7.0	high school graduate / GED	male	1.78	9.1	yes
73559	72.0	109.2	175.3	126.0	40.0	88.9	28.9	Non-Hispanic White	yes	0.0	some college or AA degree	male	4.51	8.9	yes
73562	56.0	123.1	158.7	226.0	34.2	105.0	41.7	Mexican American	yes	5.0	some college or AA degree	male	4.79	5.5	no
73564	61.0	110.8	161.8	168.0	37.1	93.4	35.7	Non-Hispanic White	yes	2.0	college graduate or above	female	5.0	5.5	no
73566	56.0	85.5	152.8	278.0	32.4	61.8	26.5	Non-Hispanic White	no	1.0	high school graduate / GED	female	0.48	5.4	no
73567	65.0	93.7	172.4	173.0	40.0	65.3	22.0	Non-Hispanic White	no	4.0	9th-11th grade	male	1.2	5.2	no
73568	26.0	73.7	152.5	168.0	34.4	47.1	20.3	Non-Hispanic White	no	2.0	college graduate or above	female	5.0	5.2	no
73571	76.0	122.1	172.5	167.0	35.5	102.4	34.4	Non-Hispanic White	yes	2.0	college graduate or above	male	5.0	6.9	yes
73577	32.0	100.0	166.2	182.0	36.5	79.7	28.9	Mexican American	no	20.0	Less than 9th grade	male	0.29	5.3	no
73581	50.0	99.3	185.0	202.0	42.8	80.9	23.6	Other or Multi-Racial	no	0.0	college graduate or above	male	5.0	5.0	no
73585	28.0	90.3	175.1	198.0	40.5	92.2	30.1	Other or Multi-Racial	no	4.0	some college or AA degree	male	2.26	5.0	no
73589	35.0	94.6	172.9	192.0	39.1	78.3	26.2	Non-Hispanic White	no	2.0	high school graduate / GED	male	1.74	5.5	no
73595	58.0	114.8	175.3	165.0	40.1	96.0	31.2	Other Hispanic	no	1.0	some college or AA degree	male	3.09	7.7	no
73596	57.0	117.8	164.7	151.0	35.3	104.0	38.3	Other or Multi-Racial	yes	1.0	college graduate or above	female	5.0	5.9	no
73600	37.0	122.9	185.1	189.0	48.1	126.2	36.8	Non-Hispanic Black	yes	2.0	high school graduate / GED	male	0.63	6.2	yes
73604	69.0	96.6	156.9	203.0	37.0	59.5	24.2	Non-Hispanic White	no	1.0	some college or AA degree	female	2.44	5.4	no
73607	75.0	130.5	169.6	161.0	36.5	111.9	38.9	Non-Hispanic White	yes	0.0	high school graduate / GED	male	1.08	5.0	no
73610	43.0	102.6	176.8	200.0	38.8	90.2	28.9	Non-Hispanic White	no	5.0	college graduate or above	male	2.03	4.9	no
73613	60.0	113.6	163.8	203.0	41.6	104.9	39.1	Non-Hispanic Black	yes	2.0	9th-11th grade	female	5.0	6.1	no
73614	55.0	90.9	167.9	256.0	43.5	60.9	21.6	Non-Hispanic White	no	0.0	high school graduate / GED	female	1.29	5.0	no
73615	65.0	100.3	145.9	166.0	30.0	55.4	26.0	Other Hispanic	yes	1.0	Less than 9th grade	female	1.22	6.3	yes

$X_1 \ X_2 \ \dots$

$X_{14}$

$X_{14}$  (LBXGH)  $\leq 6.15$  has the highest IG

GLYCOHEMOGLOBIN (LBXGH)  $\leq 6.15$   
entropy = 0.92  
samples = 1082  
value = [720, 362]  
class = None

True

False

entropy = 0.533  
samples = 792  
value = [696, 96]  
class = None

entropy = 0.412  
samples = 290  
value = [24, 266]  
class = Diabetes

Dataset partition  $\mathcal{D}[\text{LBXGH} \leq 6.15]$

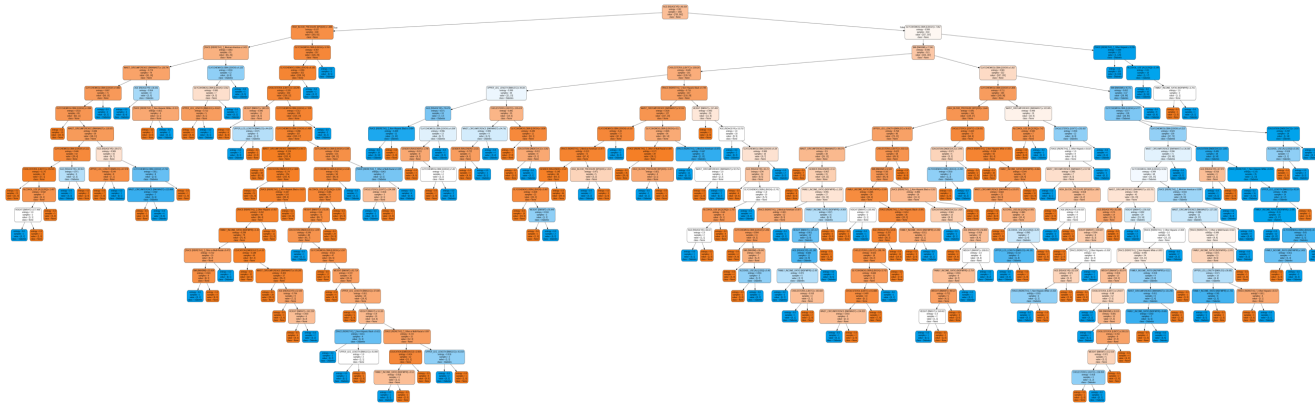
SEQN	RIDAGEYR	BMXWAIST	BMXHT	LBXTC	BMXLEG	BMXWT	BMXBMI	RIDRETH1	BPQ020	ALQ120Q	DMDEDUC2	RIAGENDR	INDFMPPIR	LBXGH	DIABETIC
73562	56.0	123.1	158.7	226.0	34.2	105.0	41.7	Mexican American	yes	5.0	some college or AA degree	male	4.79	5.5	no
73564	61.0	110.8	161.8	168.0	37.1	93.4	35.7	Non-Hispanic White	yes	2.0	college graduate or above	female	5.0	5.5	no
73566	56.0	85.5	152.8	278.0	32.4	61.8	26.5	Non-Hispanic White	no	1.0	high school graduate / GED	female	0.48	5.4	no
73567	65.0	93.7	172.4	173.0	40.0	65.3	22.0	Non-Hispanic White	no	4.0	9th-11th grade	male	1.2	5.2	no
73568	26.0	73.7	152.5	168.0	34.4	47.1	20.3	Non-Hispanic White	no	2.0	college graduate or above	female	5.0	5.2	no
73577	32.0	100.0	166.2	182.0	36.5	79.7	28.9	Mexican American	no	20.0	Less than 9th grade	male	0.29	5.3	no
73581	50.0	99.3	185.0	202.0	42.8	80.9	23.6	Other or Multi-Racial	no	0.0	college graduate or above	male	5.0	5.0	no
73585	28.0	90.3	175.1	198.0	40.5	92.2	30.1	Other or Multi-Racial	no	4.0	some college or AA degree	male	2.26	5.0	no
73589	35.0	94.6	172.9	192.0	39.1	78.3	26.2	Non-Hispanic White	no	2.0	high school graduate / GED	male	1.74	5.5	no
73596	57.0	117.8	164.7	151.0	35.3	104.0	38.3	Other or Multi-Racial	yes	1.0	college graduate or above	female	5.0	5.9	no
73604	69.0	96.6	156.9	203.0	37.0	59.5	24.2	Non-Hispanic White	no	1.0	some college or AA degree	female	2.44	5.4	no
73607	75.0	130.5	169.6	161.0	36.5	111.9	38.9	Non-Hispanic White	yes	0.0	high school graduate / GED	male	1.08	5.0	no
73610	43.0	102.6	176.8	200.0	38.8	90.2	28.9	Non-Hispanic White	no	5.0	college graduate or above	male	2.03	4.9	no
73613	60.0	113.6	163.8	203.0	41.6	104.9	39.1	Non-Hispanic Black	yes	2.0	9th-11th grade	female	5.0	6.1	no
73614	55.0	90.9	167.9	256.0	43.5	60.9	21.6	Non-Hispanic White	no	0.0	high school graduate / GED	female	1.29	5.0	no

Dataset partition  $\mathcal{D}[\text{LBXGH} > 6.15]$

SEQN	RIDAGEYR	BMXWAIST	BMXHT	LBXTC	BMXLEG	BMXWT	BMXBMI	RIDRETH1	BPQ020	ALQ120Q	DMDEDUC2	RIAGENDR	INDFMPPIR	LBXGH	DIABETIC
73557	69.0	100.0	171.3	167.0	39.2	78.3	26.7	Non-Hispanic Black	yes	1.0	high school graduate / GED	male	0.84	13.9	yes
73558	54.0	107.6	176.8	170.0	40.0	89.5	28.6	Non-Hispanic White	yes	7.0	high school graduate / GED	male	1.78	9.1	yes
73559	72.0	109.2	175.3	126.0	40.0	88.9	28.9	Non-Hispanic White	yes	0.0	some college or AA degree	male	4.51	8.9	yes
73571	76.0	122.1	172.5	167.0	35.5	102.4	34.4	Non-Hispanic White	yes	2.0	college graduate or above	male	5.0	6.9	yes
73595	58.0	114.8	175.3	165.0	40.1	96.0	31.2	Other Hispanic	no	1.0	some college or AA degree	male	3.09	7.7	no
73600	37.0	122.9	185.1	189.0	48.1	126.2	36.8	Non-Hispanic Black	yes	2.0	high school graduate / GED	male	0.63	6.2	yes
73615	65.0	100.3	145.9	166.0	30.0	55.4	26.0	Other Hispanic	yes	1.0	Less than 9th grade	female	1.22	6.3	yes

# Diabetes DT – Random vs IG Features

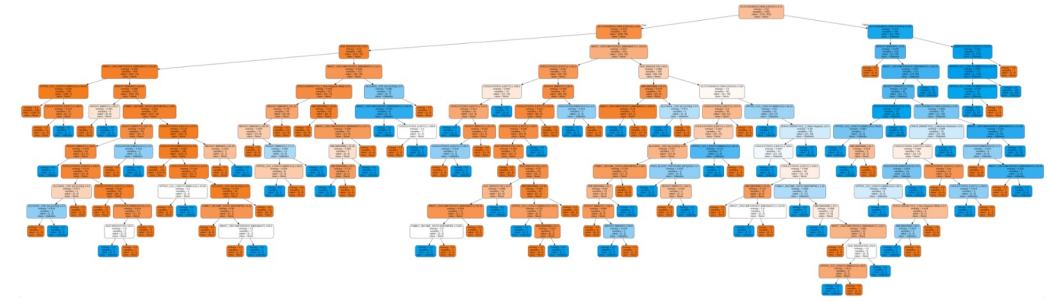
DT with random feature splits



Accuracy on diabetes data = 100%

- It is smaller while retaining 100 % accuracy on our training data
- Still rather complex, though, and vulnerable to *overfitting* (we'll see in a bit)...
- But first: let's see a sketch of building Decision Trees in Scikit-Learn

DT via IG



Accuracy on diabetes data = 100%

# **An Example Using Pandas, Numpy, Sklearn**

# Classifying Mammals

Organism	Temp reg.?	Live birth?	Four legs?	Hibernates?	Fuzz/hair?	Mammal?
bear	Y	Y	Y	Y	Y	Y
dog	Y	Y	Y	N	Y	Y
dolphin	Y	Y	N	N	N	Y
bat	Y	Y	N	Y	Y	Y
platypus	Y	N	Y	N	Y	Y
newt	N	N	Y	Y	N	N
skink	N	N	Y	N	N	N
rat snake	N	N	N	Y	N	N
lobster	N	N	N	N	N	N
kiwi	Y	N	N	N	Y	N
blue shark	N	Y	N	N	N	N

*Inspired by an example by Mohsen Afsharchi*

	organism	endothermic	live_birth	four_legs	hibernates	fuzz	mammal
0	bear	True	True	True	True	True	True
1	dog	True	True	True	False	True	True
2	dolphin	True	True	False	False	False	True
3	bat	True	True	False	True	True	True
4	platypus	True	True	True	False	True	True
5	newt	False	True	True	True	True	False
6	skink	False	True	True	False	True	False
7	rat_snake	False	False	False	True	False	False
8	lobster	False	False	False	False	False	False
9	kiwi	True	False	False	False	True	False
10	blue_shark	False	True	True	True	False	False

```
'fuzz': True,
'fuzz': True,
False, 'fuzz': False,
'fuzz': True,
False, 'fuzz': True,
e, 'fuzz': True,
lse, 'fuzz': True,
s': True, 'fuzz': False,
: False, 'fuzz': False,
lse, 'fuzz': True,
': True, 'fuzz': False,
```

# Some Basics:

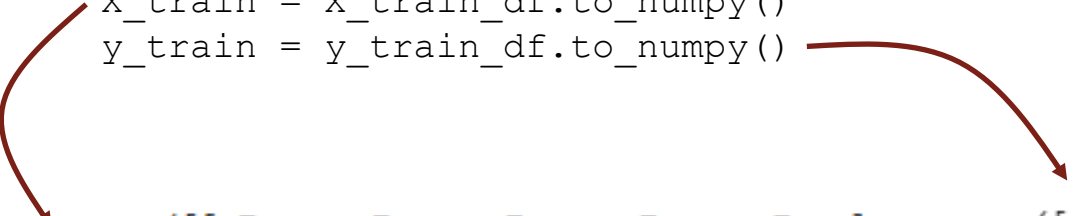
## Identifying Features

	endothermic	live_birth	four_legs	hibernates	fuzz	mammal
0	True	True	True	True	True	True
1	True	True	True	False	True	True
2	True	True	False	False	False	True
3	True	True	False	True	True	True
4	True	True	True	False	True	True
5	False	True	True	True	True	False
6	False	True	True	False	True	False
7	False	False	False	True	False	False
8	False	False	False	False	False	False
9	True	False	False	False	True	False
10	False	True	True	True	False	False

# Getting Training Data into Form

```
X_train_df = animals_train_df[['endothermic', 'live_birth', 'four_legs', 'hibernates', 'fuzz']]
y_train_df = animals_train_df['mammal']
```

```
X_train = X_train_df.to_numpy()
y_train = y_train_df.to_numpy()
```



```
array([[ True,  True,  True,  True,  True],
       [ True,  True,  True, False,  True],
       [ True,  True, False, False, False],
       [ True,  True, False,  True,  True],
       [ True,  True,  True, False,  True],
       [False,  True,  True,  True,  True],
       [False,  True,  True, False,  True],
       [False, False, False,  True, False],
       [False, False, False, False, False],
       [ True, False, False, False,  True],
       [False,  True,  True,  True, False]])
```

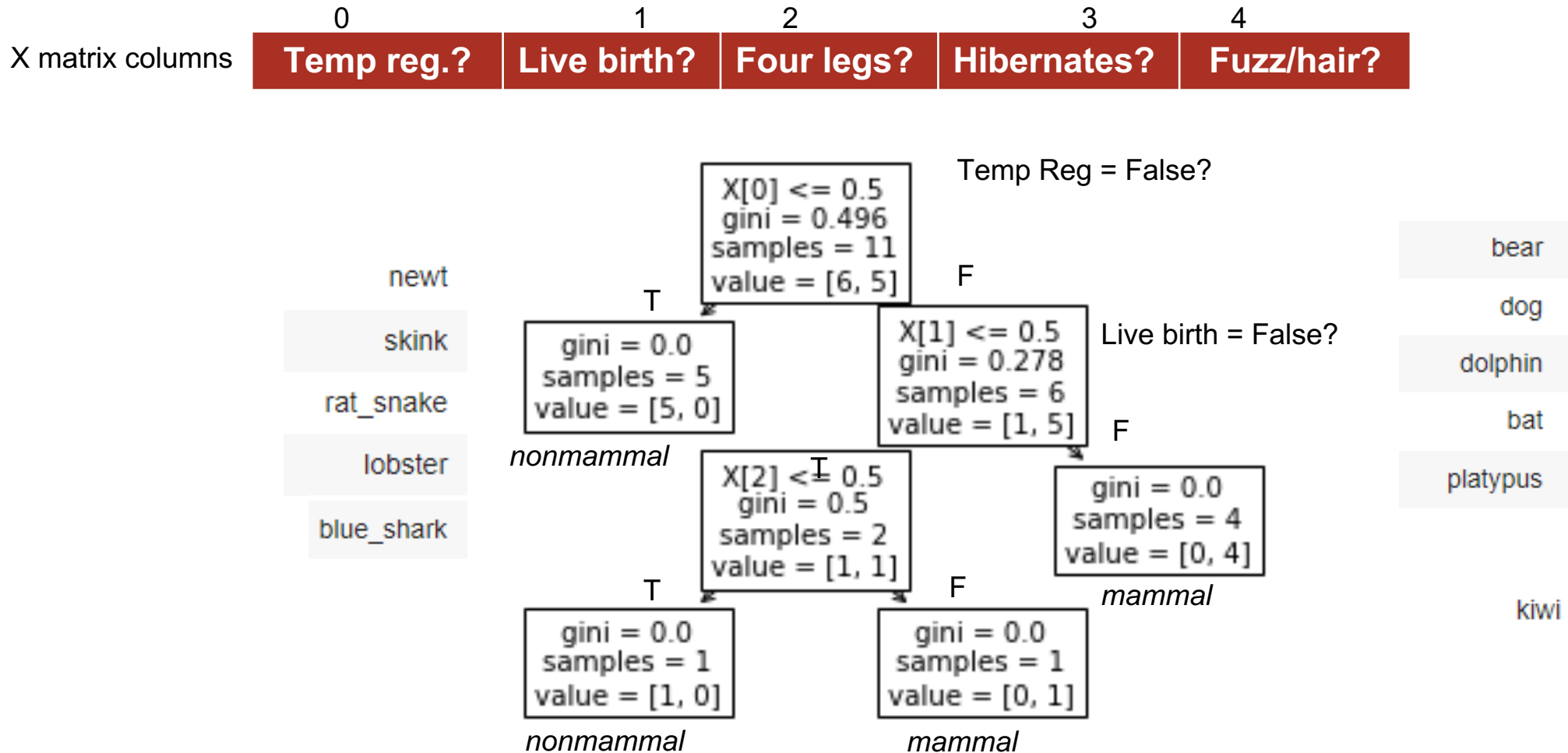
```
array([ True,  True,  True,  True,  True, False, False, False, False,
        False, False])
```

```
clf = tree.DecisionTreeClassifier()

trained = clf.fit(X_train,y_train)

tree.plot_tree(trained)
```

# The Trained DT Model





# **Overfitting and Decision Trees**

# Looks Perfect, Until We Test...

X\_test\_df:

	organism	endothermic	live_birth	four_legs	hibernates	fuzz	mammal
0	dolphin	True	True	False	False	False	True
1	human	True	True	False	False	True	True
2	hairless_cat	True	True	True	False	False	True
3	whale	True	True	False	False	False	True
4	echidna	True	False	True	False	False	True
5	therizinosaurus	True	False	True	False	True	False
6	bald_eagle	True	False	False	False	False	False

```
trained.predict(X_test_df.to_numpy())
```

```
array([ True,  True,  True,  True, False, False, False])
```

# Similarly: with Diabetes Decision Tree

Original Patient Data: 100.00 % (n = 1082)

New Patient Data: 82.796 % (n = 465)

# The Overfitting Problem

What is happening?

- Our algorithm has chosen some model representing hypothesis  $h$
- But there (likely) exists another hypothesis  $h'$  such that:

$$\text{error}(h(D_{\text{train}})) < \text{error}(h'(D_{\text{train}}))$$

$$\text{error}(h'(D)) < \text{error}(h(D))$$

*(or else our heuristics are in fact preventing us from finding  $h'$ : our greedy algorithm doesn't consider all possible trees)*

# What Causes Overfitting?

- Noisy training data: noise/errors can cause **contradictory labels** for data with the same features
- Training data is non-representative, or does not include unusual cases (e.g., egg-laying mammals, non-endothermic mammals)

# Avoiding Overfitting

## How can we avoid overfitting?

1. Acquire more training data (might be very hard)
2. Remove irrelevant attributes (manual process, not always possible)
3. Keep our model from getting too complex

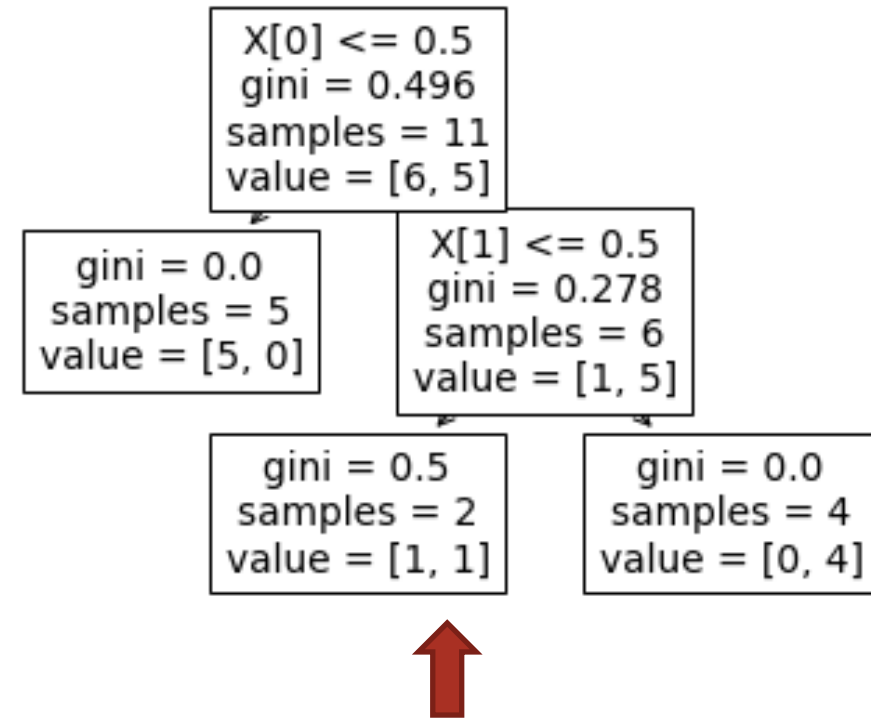
# Recall Occam's Razor



**Key Idea:** The simplest consistent explanation is the best

# What this Entails

- Have a *less complex* decision tree!
- This might actually look worse on training data but may generalize better!





# Avoiding Overfitting

How can we avoid overfitting?

1. Acquire more training data
2. Remove irrelevant attributes (manual process, not always possible)
3. **Stop growing, e.g., when data split is not statistically significant**
4. **Grow full tree, then post-prune**

Try various tree hyperparameters (e.g., tree depth, splitting criterion, termination criterion) and pick the one with the **best estimated generalization performance**. How to estimate?

- Cross-validation
- Add a complexity penalty to performance measure e.g., training accuracy – average depth of leaf node

# Stopping Growth

- Set a maximum **depth** to the decision tree (`max_depth` in Scikit-Learn)
- Set a minimum number of samples in a node, for us to split (e.g., 2) (`min_samples_split` in skl)
- Set a minimum number of samples in a leaf (`min_samples_leaf`)

(Again: we might use k-fold cross-validation to compare)

But alternatively, we can build “the perfect tree” and then **prune back**, based on validation set

# Reduced-Error Pruning

Split the original training data into training and **validation sets**

## Training Stage

Grow the decision tree based on the training set

## Pruning Stage

Loop until further pruning hurts validation performance:

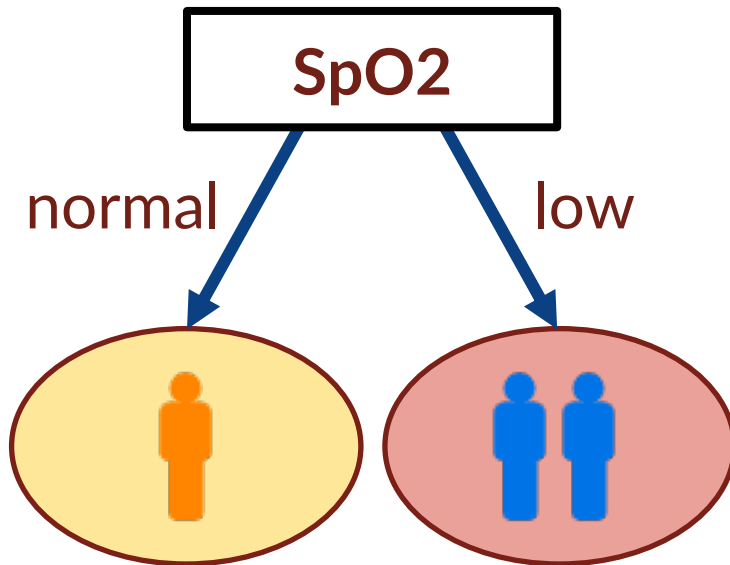
- Measure the validation performance of pruning each node (and its children)
- Greedily remove the node that most improves validation performance

This is very helpful for our Diabetes data

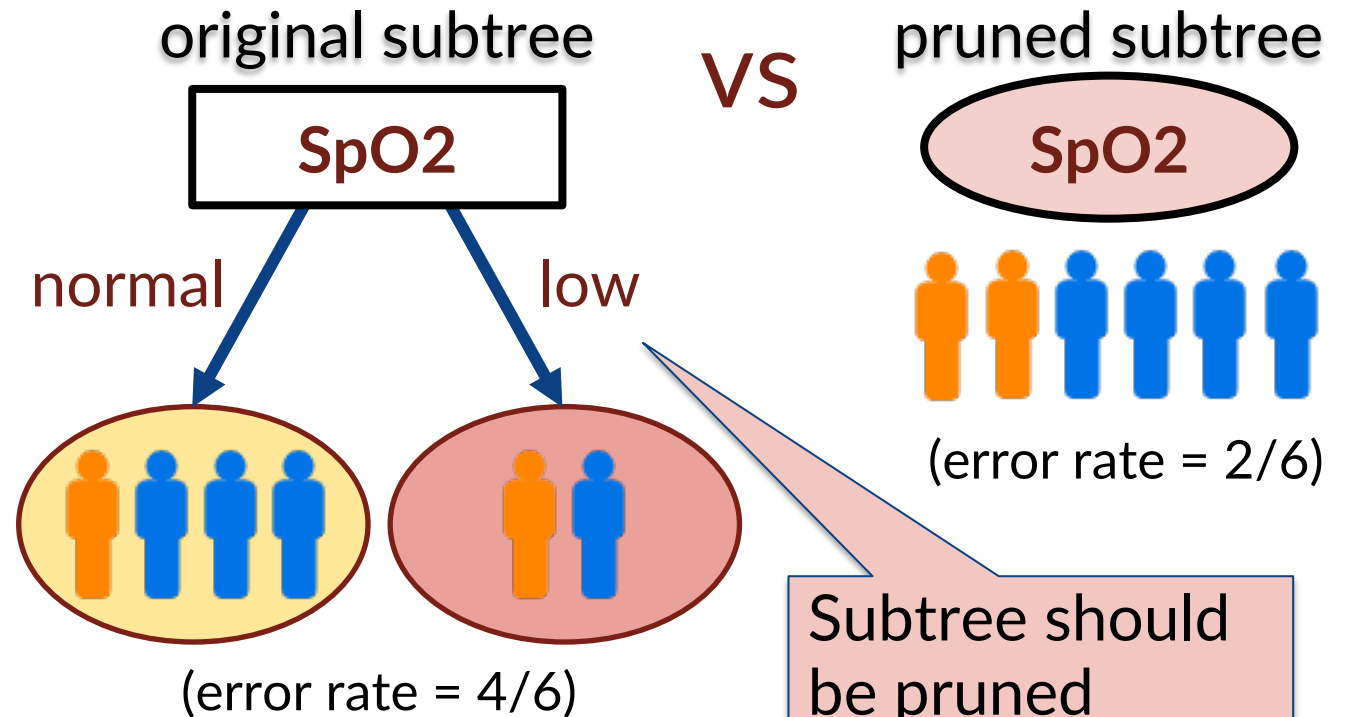
# Reduced-Error Pruning

- Pruning replaces a **whole subtree with a leaf node**
- Replacement occurs if the error rate of the subtree is greater than that of the leaf

## Training



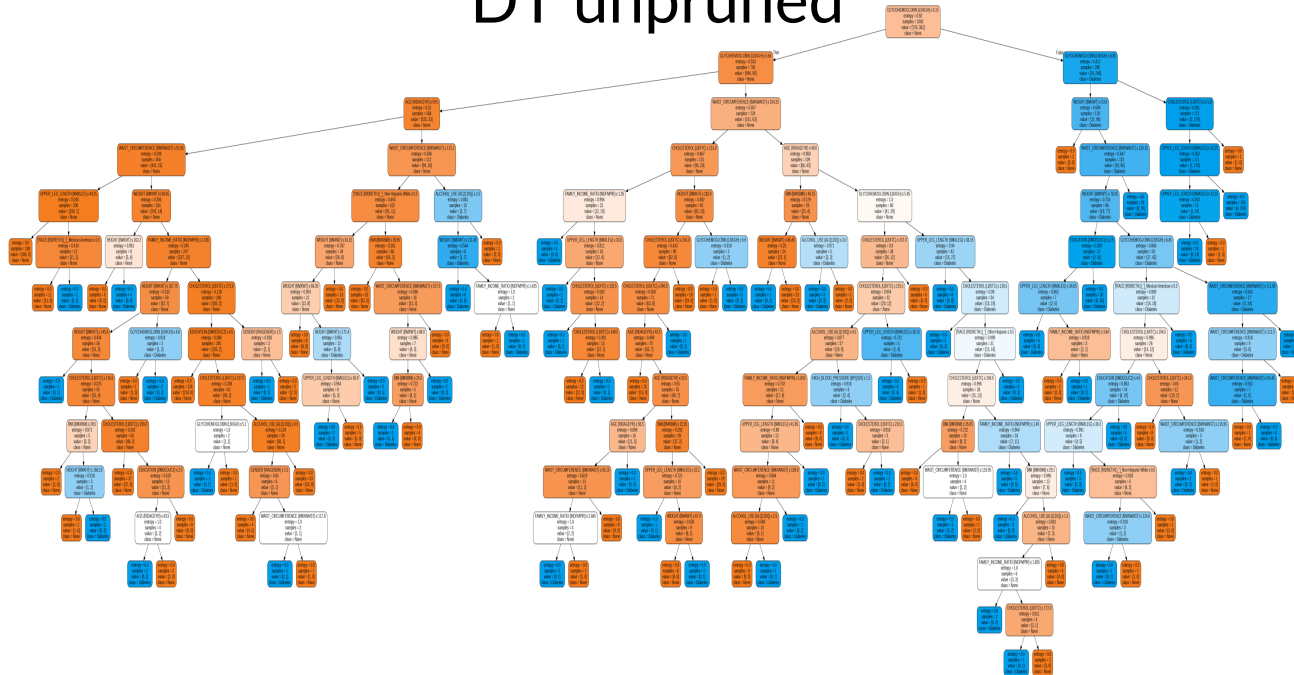
## Validation



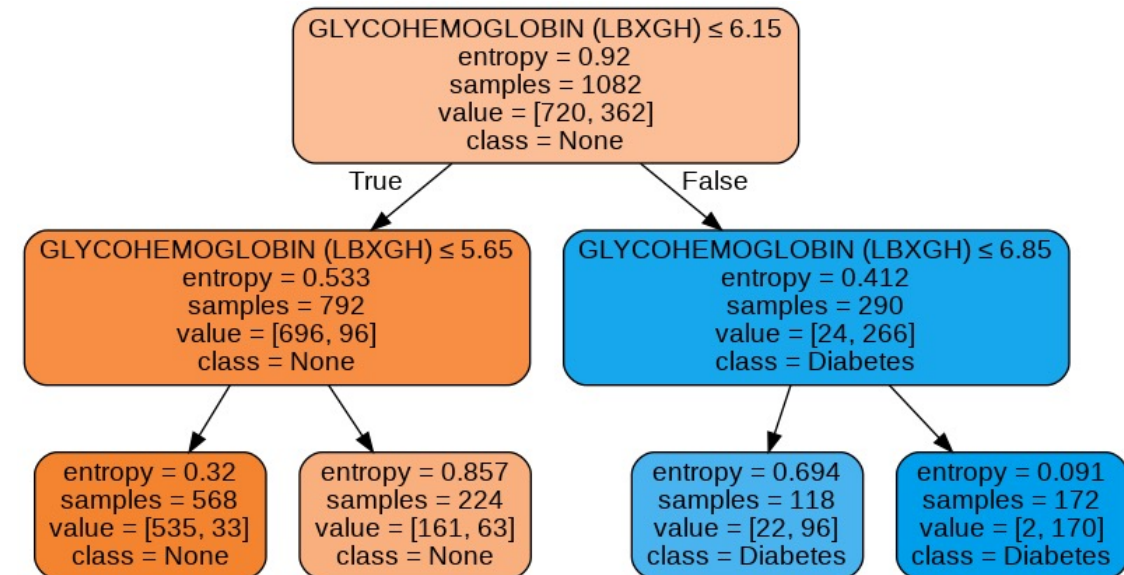
# Accuracy – Decision Trees

Original Patient Data:	DT unpruned	DT pruned	
	100.000 %	88.909 %	(n = 1082)
New Patient Data:	82.796 %	85.591 %	(n = 465)

DT unpruned

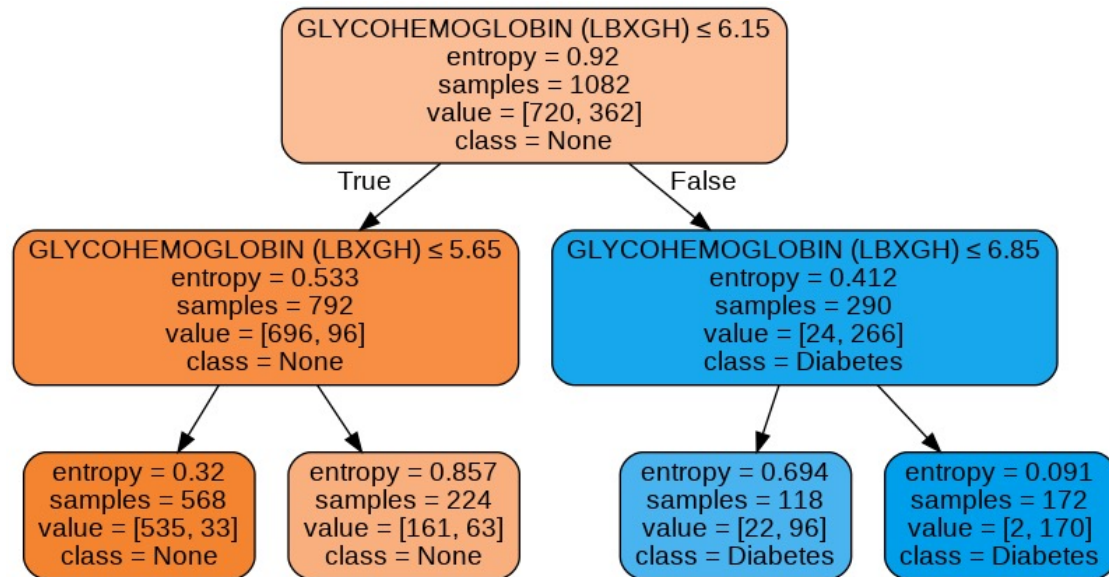


DT pruned



# The Final Diabetes DT

## Our Pruned Decision Tree



## How Diabetes is Actually Diagnosed



- If your A1C level is between 5.7 and less than 6.5%, your levels have been in the prediabetes range.
- If you have an A1C level of 6.5% or higher, your levels were in the diabetes range.

(screenshot from diabetes.org)

Strong similarity to how diabetes is *actually* diagnosed!

# Decision Tree Algorithms

## ID3

- Information gain on nominal features

## C4.5

- Can use info gain or gain ratio
- Nominal or numeric features
- Missing values
- Post-pruning
- Rule generation

## CART (Classification and Regression Tree)

- Similar to C4.5
- Can handle continuous target prediction (regression)
- No rule sets
- Sklearn's `DecisionTreeClassifier` is based on CART, but can't handle nominal features (as of version 0.22.1)

## Many Other Algorithms ...

# Strengths and Weaknesses of DTs

## Strengths

- 👍 Widely used in practice
- 👍 Fast and simple to implement
- 👍 Small trees are easily interpretable
- 👍 Handles a variety of feature types
- 👍 Can convert to rules
- 👍 Handles noisy / missing data
- 👍 Insensitive to feature scaling
- 👍 Handles irrelevant features
- 👍 Handles large datasets

## Weaknesses

- 👎 Univariate partitions limit potential trees
- 👎 Heuristic-Based greedy training



# Another Idea to Prevent Overfitting

A single decision tree can be prone to **overfitting** to the training data

What if we use **randomization** to create multiple decision trees, each a bit different:

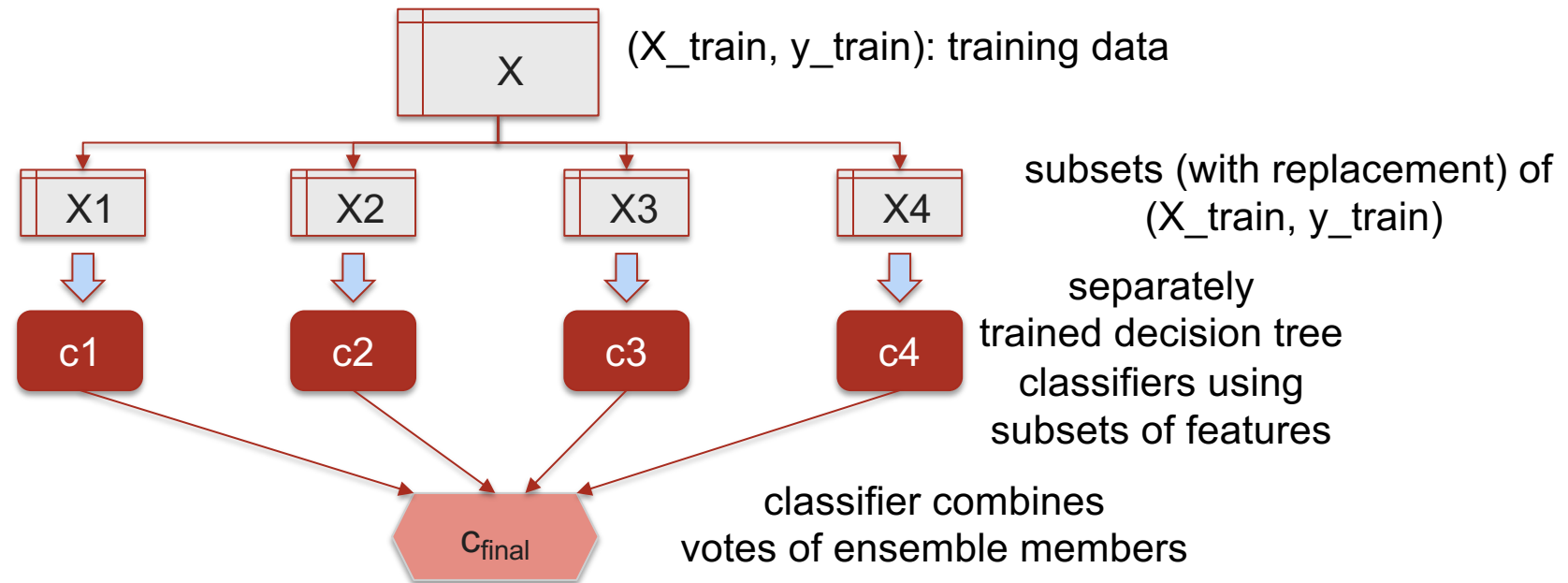
- Each is trained on a **sample** of the training data
- Each splits along a **subset** of the possible features
- Each is a small decision tree (“stump”)

Then we rely on **voting** to make this work!

- Intuition: **the most predictive features** will be selected in many decision stumps!

(Note we now give up the “explainability” property)

# Random Forests



# Training a Decision Tree in a Random Forest

1. Draw a random **bootstrap** data sample of size  $n$ , with replacement
2. Build (“grow”) a small decision tree (often just a “stump”)
  - At each split point node, randomly select  $d$  candidate features (w/o replacement)
  - Split the node using the feature with best split according to objective function (e.g. information gain)
3. Repeat to produce  $k$  decision trees (a forest!)
4. For prediction, use **majority vote** to predict a class for new data

# Benefits of Random Forests

One of the most popular and accurate classifiers for big data (more in a moment)

- Scale-invariant
- Much less susceptible to overfitting than “plain” decision trees
- Can be generalized to continuous data (random forests of CaRT trees)

Also: training is highly parallelizable!

- Take a data set, draw samples of size  $n$  with replacement
- Train a separate decision tree on this, at each split point selecting from a subset of the features (without replacement **for this tree**)

Let's see a case study...

# Case Study: Seizure Prediction on Kaggle

Research Prediction Competition

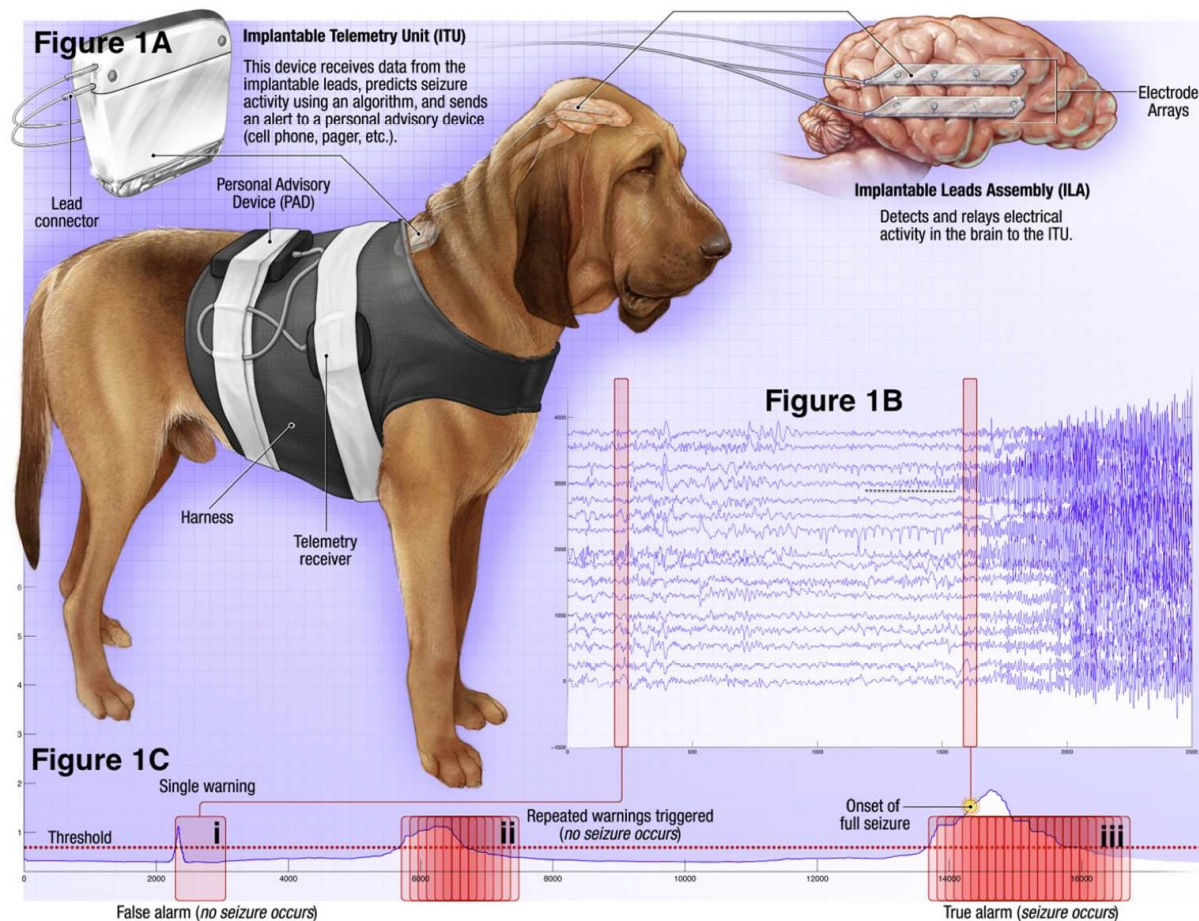
## American Epilepsy Society Seizure Prediction Challenge

Predict seizures in intracranial EEG recordings

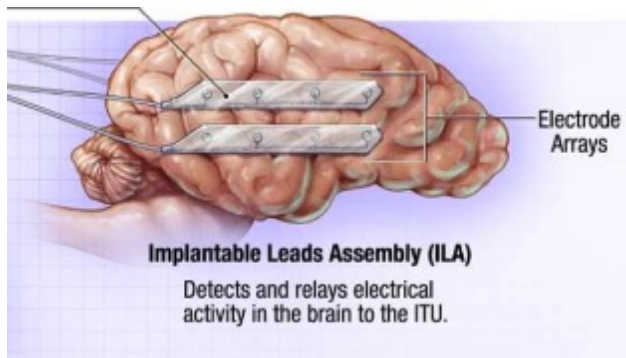
\$25,000

Prize Money

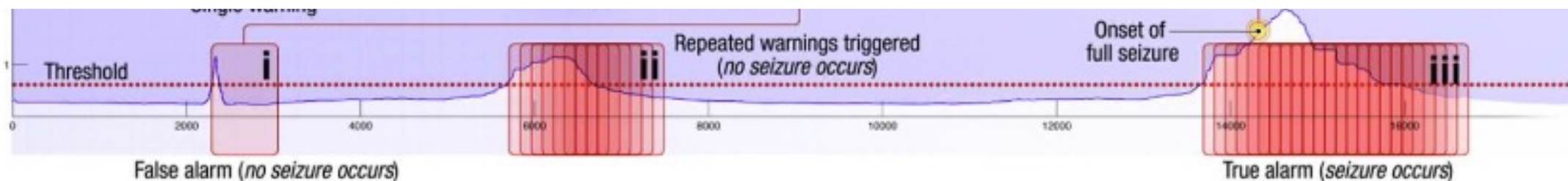
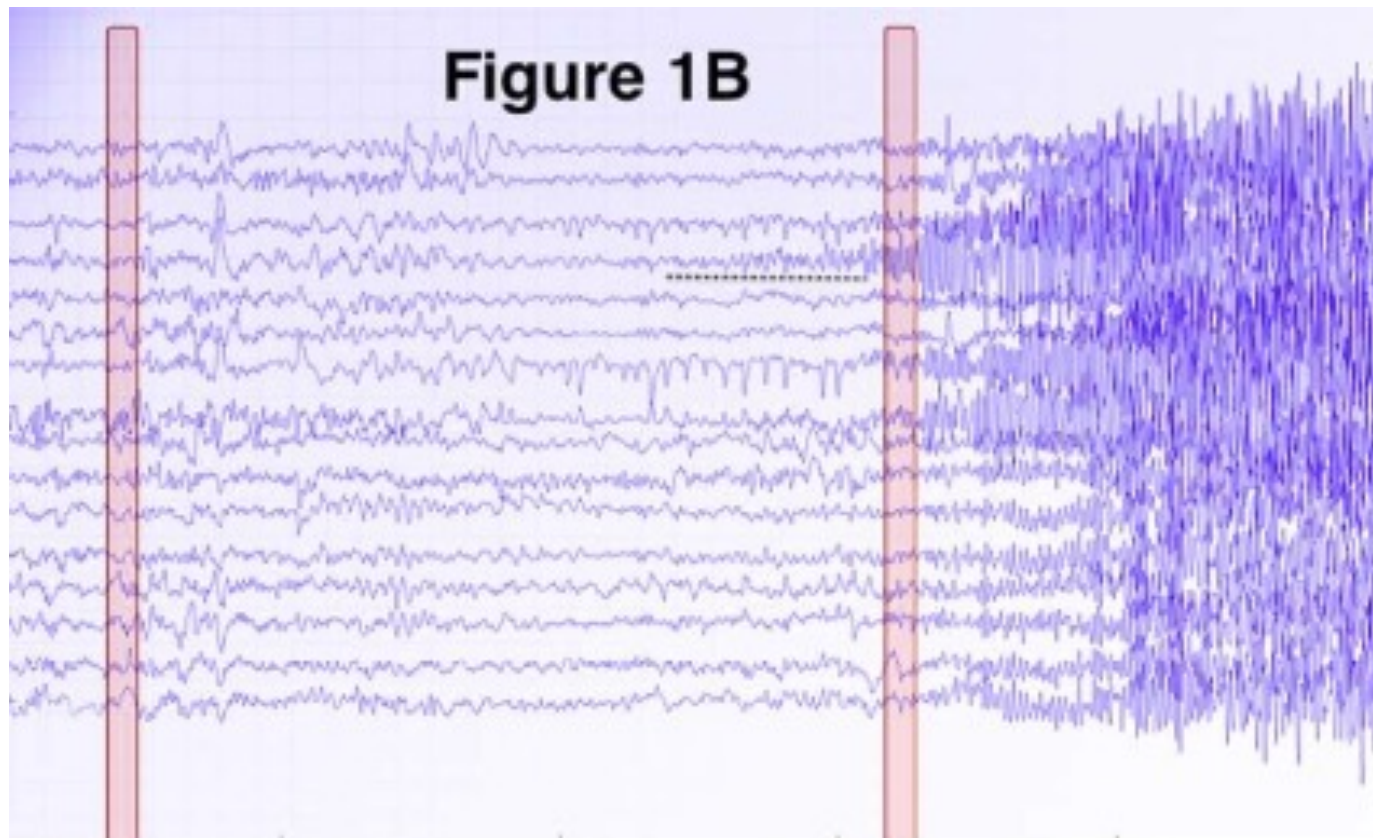
504 teams







# Multi-Channel Time Series Data



# The Data

Data was broken into small fixed-length segments (labeled by a human expert)

Essentially just a 2D matrix: voltage levels per channel vs time

Kaggle divided data into training, leaderboard, and *actual* test sets

**Table 1** Data characteristics for the Kaggle.com seizure forecasting contest and held-out data experiment











Subject	Sampling rate (Hz)	Recorded data (h)	Seizures	Lead seizures	Training clips (% interictal)	Testing clips (% interictal)	Held-out clips (% interictal)
Dog 1	400	1920	22	8	504 (95.2)	502 (95.2)	2000 (99.7)
Dog 2	400	8208	47	40	542 (92.3)	1000 (91.0)	1000 (100)
Dog 3	400	5112	104	18	1512 (95.2)	907 (95.4)	1000 (100)
Dog 4	400	7152	29	27	901 (89.2)	990 (94.2)	1000 (95.8)

1.7) 0

1.9) 0

1.7) 0

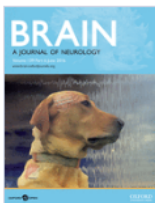
■ Prize Winners

#	Δ	Team	Members	Score	Entries	Last	Code
1	—	Medrr		0.83993	264	8Y	
2	▲ 2	QMSDP	    	0.81962	501	8Y	
3	▲ 5	Birchwood		0.80078	160	8Y	
4	▲ 11	ESAI CEU-UCH	  	0.79347	182	8Y	

# Features

- Just using the data samples wasn't really enough!
- A few examples of techniques for extracting features from EEG timeseries:
  - Spectral analysis: use Fourier transforms to re-express the signal as a composition of sine waves, identify the frequency bands with the most power
  - look at the area under the curve or the derivative of the curve
  - etc.





Volume 139, Issue 6  
June 2016

## Article Contents

## Abstract

## Introduction

## JOURNAL ARTICLE

# Crowdsourcing reproducible seizure forecasting in human and canine epilepsy

Benjamin H. Brinkmann, Joost Wagenaar, Drew Abbot, Phillip Adkins,  
Simone C. Bosshard, Min Chen, Quang M. Tieng, Jialune He, F. J. Muñoz-Almaraz,  
Paloma Botella-Rocamora, Juan Pardo, Francisco Zamora-Martinez, Michael Hills,  
Wei Wu, Iryna Korshunova, Will Cukierski, Charles Vite, Edward E. Patterson, Brian Litt,  
Gregory A. Worrell

## Author Notes

*Brain*, Volume 139, Issue 6, June 2016, Pages 1713–1722,

<https://doi.org/10.1093/brain/aww045>

Published: 31 March 2016 Article history ▼

**Table 3** AUC scores for the held-out data experiment compared to scores on the public and private leader boards

Task	Team performance						
	Team name	Window (overlap)	Features	Machine learning algorithm	Ensemble method	Public leader board	Private leader board
P1	QMSDP	60s (0%), 8 s (97%)	Spectral power, spectral entropy, correlation, fractal dimensions, Hjorth parameters, distribution statistics, signal variance	LassoGLM, Bagged SVM, Random Forest	Weighted average	0.86	0.82
	QMSDP	60 s (0%)	Spectral entropy, correlation, fractal dimensions, Hjorth parameters, distribution statistics	LassoGLM		0.84	0.81
	QMSDP	8 s (97%)	Spectral power, correlation, signal variance	Bagged SVM		0.79	0.76
	QMSDP	8 s (97%)	Spectral power, correlation, signal variance	Random Forest		0.79	0.72
	QMSDP	8 s (97%)	Spectral power, correlation, signal variance	Random Forest		0.79	0.72

# Two Notes Here

- We saw from the competition that Random Forests – which used randomization to reduce overfitting (variance) in decision trees – were useful
- But additionally they combined many other kinds of classifiers

Are there some basic principles here?

*Ensembles, next...*