Lecture 9: Non-Parametric Methods Part 3 (KNN and Decision Trees)

Feb 13, 2023

CIS 4190/5190

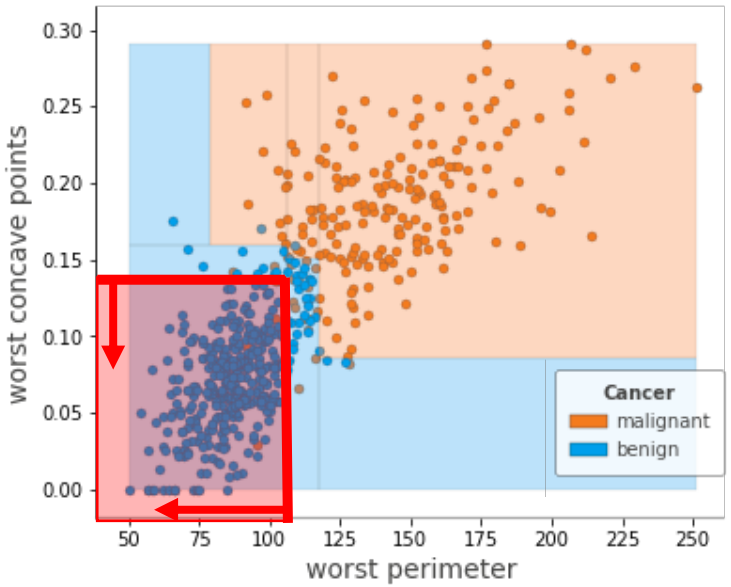
Spring 2023

Administrivia

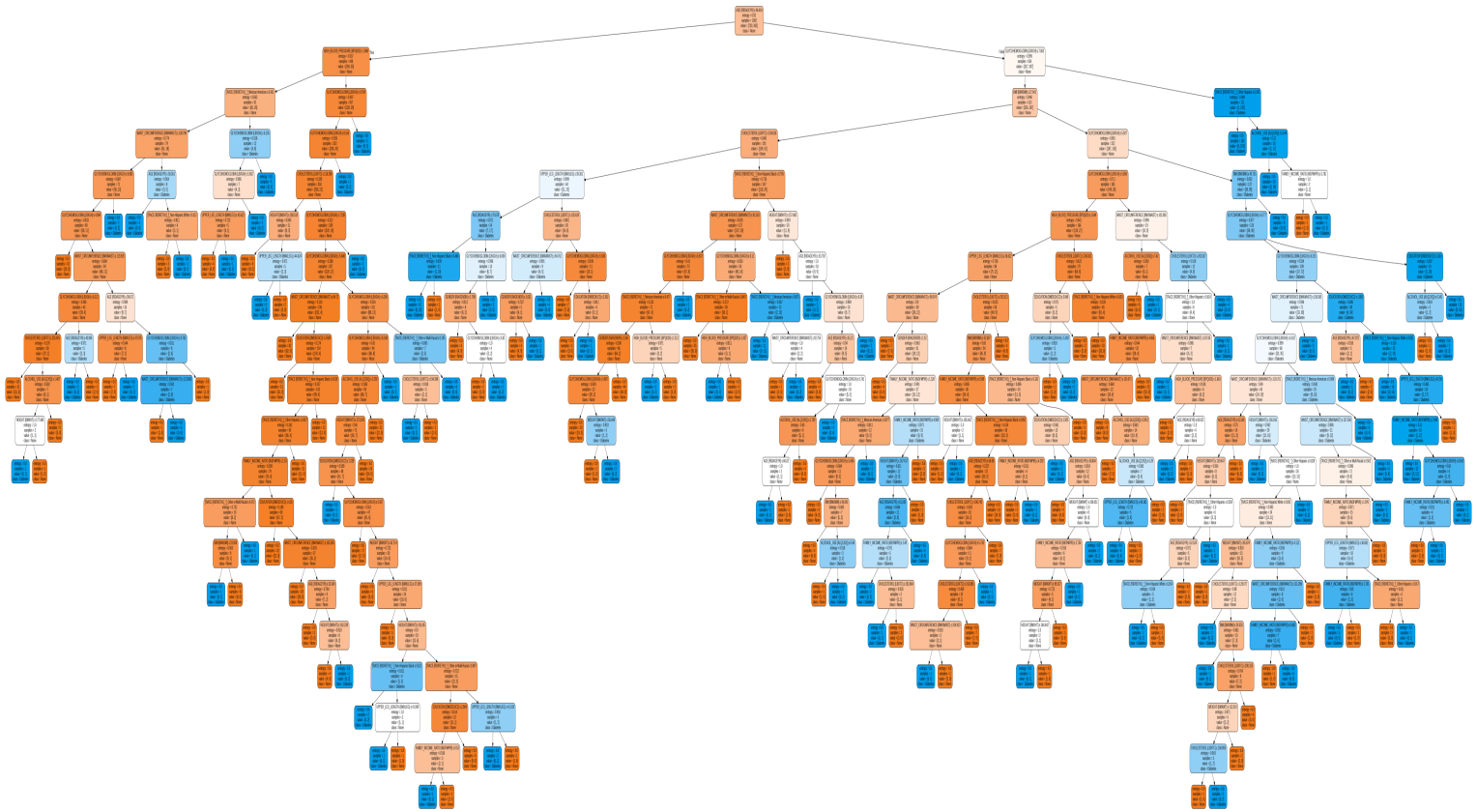
- HW3 released, due Feb 22.
  - Requires some DT materials we will wrap up today.
- Recordings of recitations online. Moved to in-person (+Zoom) recitations last week.
- Fall 2022 slides are at: https://www.seas.upenn.edu/~cis5190/fall2022/.
  - May be useful if you find that you absolutely require some starting slides to be able to make notes on. But there will almost always be changes in each semester.
- Debugging during OHs:
  - Debugging your code is not the TAs' responsibility. TAs can take a look, but are instructed to not debug for >5 minutes with any student.
  - If seeking help, remember:
    - Show evidence of your own systematic effort. **Thumb rule:** Before asking for 5 mins of OH time, spend minimum 1 hour debugging by yourself. Print statements, breakpoints, assert statements, unit tests, googling error messages etc.
  - Systematic debugging is an art worth learning! Lots of resources with tips. E.g.:
    - https://applab.unc.edu/posts/2021/02/17/debugging-tips/

Recap: Decision Trees and Training

```
|--- worst perimeter <= 105.95
|  |--- worst concave points <= 0.135
|  |  |--- class: benign
|  |--- worst concave points > 0.135
|  |  |--- worst concave points < 0.16
|  |  |  |--- class: benign
|  |  |--- worst concave points > 0.16
|  |  |  |  --- worst perimeter > 80
|  |  |  |  |  --- class: malignant
|  |  |  |  --- worst perimeter < 80
|  |  |  |  |  --- class: benign
                        …
                        …
```

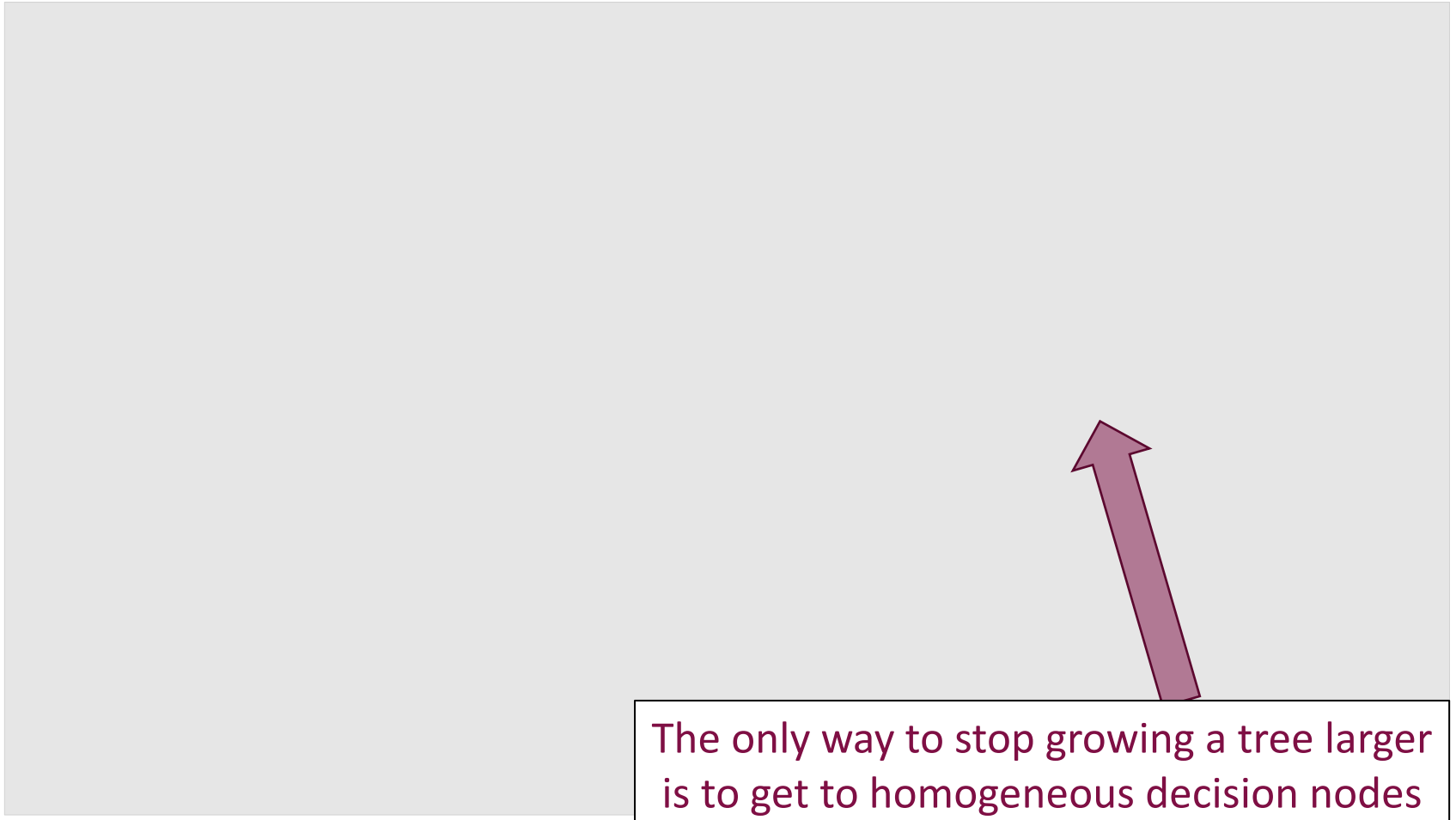


Our first attempt on diabetes data, choosing random features to split the data on



Recap: Selecting “Good” Features While Training DTs

We would like to select splits that lead as quickly as possible to homogeneous children nodes



The only way to stop growing a tree larger is to get to homogeneous decision nodes where all samples have the same label

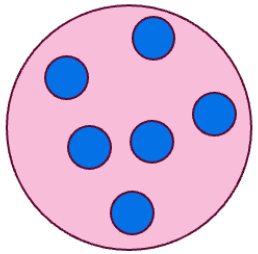
Recap: Entropy and the Information Gain Criterion

Entropy $H(\mathcal{D}) = -\sum_c P(Y = c) \log_2 P(Y = c)$,

where different $c's$ correspond to different class labels

Min Impurity

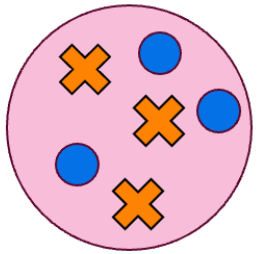
All instances in same class



$H(\mathcal{D}) = -1 \log 1 = 0$

Max Impurity

Instances split evenly among classes



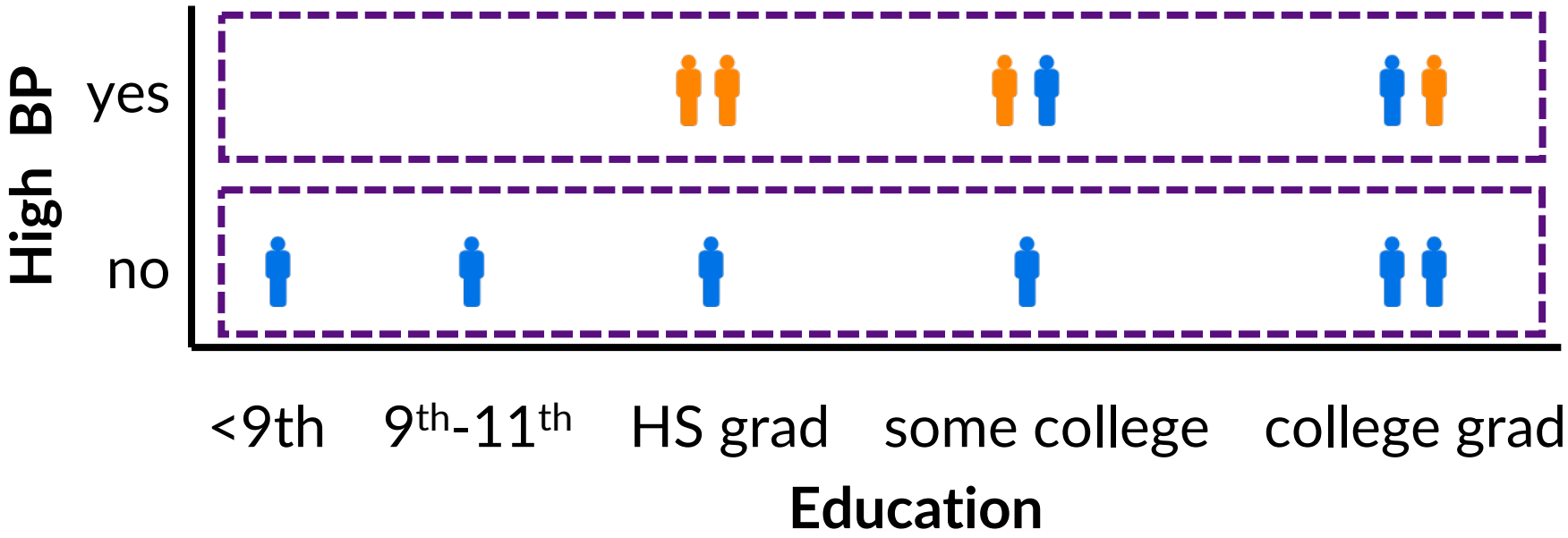
$H(\mathcal{D}) = -0.5 \log 0.5 - 0.5 \log 0.5 = 1$

Information Gain Criterion:

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

Recap: Information Gain For Diabetes Example

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



Need to compute:

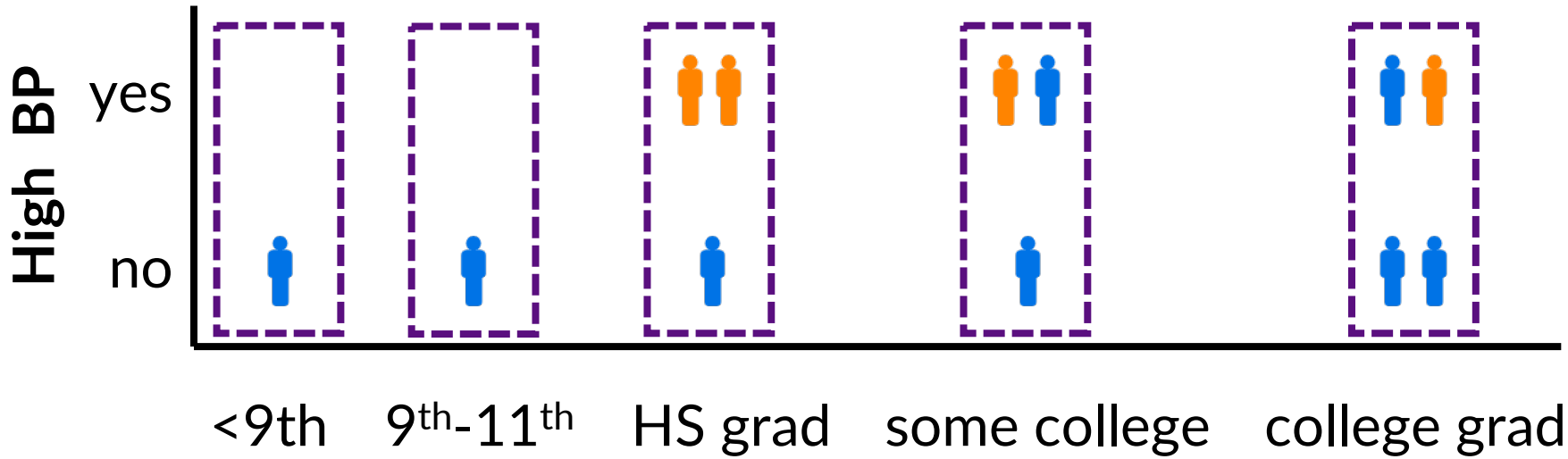
$$IG(\mathcal{D}, High\ BP) = H(\mathcal{D}) - H(\mathcal{D}\,|High\ BP)$$

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

= (6/12) * (-2/6 lg 2/6
- 4/6 lg 4/6)
+ (6/12) * (0)
= 0.459

Recap: Information Gain For Diabetes Example

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



Need to compute:

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

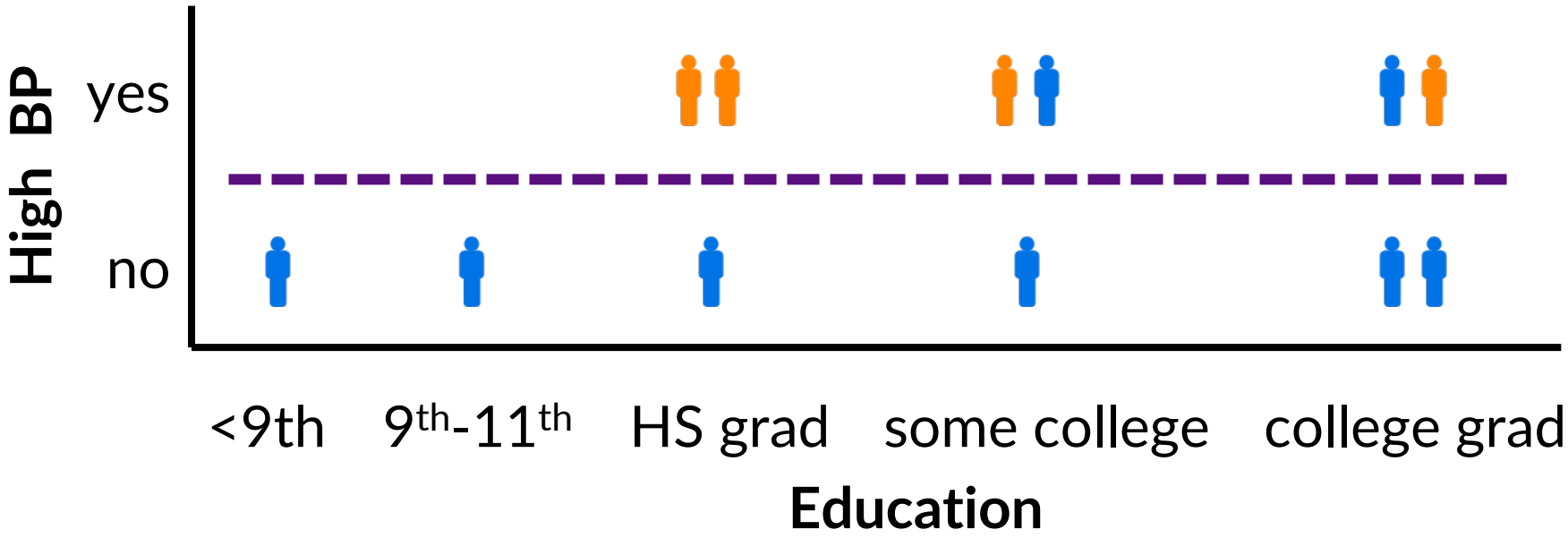
$$IG(\mathcal{D}, Education) = H(\mathcal{D}) - H(\mathcal{D}|\ Education)$$

= (1/12) * 0 + (1/12) * 0
+ (3/12) * (–1/3 lg 1/3
– 2/3 lg 2/3)
+ (3/12) * (–2/3 lg 2/3
– 1/3 lg 1/3)
+ (4/12) * (–3/4 lg 3/4
– 1/4 lg 1/4)

= 0.730

Recap: Information Gain For Diabetes Example

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



Need to compute:

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

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

So is IG always the right criterion?

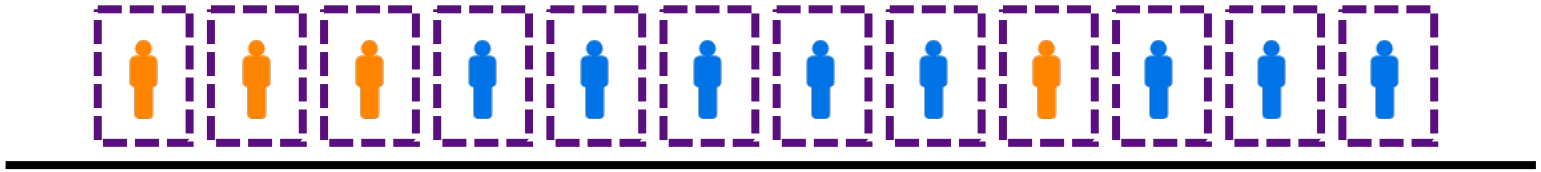
A Problem with Information Gain

- IG does indeed identify features that lead to more homogeneous child nodes.

- But note that it is easier for child nodes to be more homogeneous, when there are more children.
  - For example, what if each child has just one sample? E.g. unique IDs, dates, phone number etc.

What If Every Child Node Holds 1 Training Sample?

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



Patient ID

Need to compute:

$$IG(\mathcal{D}, ID) = H(\mathcal{D}) - H(\mathcal{D}|ID)$$

=1/12*0+1/12*0+….
= 0

IG = 0.918 … highest possible!

A Problem with Information Gain

- IG does indeed identify features that lead to more homogeneous child nodes.

- But note that it is easier for child nodes to be more homogeneous, when there are more children.
  - For example, what if each child has just one sample? e.g. unique IDs, dates, phone number etc.

  - More broadly, more child nodes ⇒ fewer data at each node ⇒ less reliable estimates of statistical properties such as entropy and more likely to overfit.

So we would like to combat IG's preference for creating many child nodes

Compensating for Features with Many Values

Gain Ratio can compensate for this:

$$GR\left(\mathcal{D}, X_j\right) = \frac{IG\left(\mathcal{D}, X_j\right)}{SplitInfo\left(\mathcal{D}, X_j\right)}$$

Ratio of *task-relevant* information to task-non-specific *intrinsic* information

$$SplitInfo\left(\mathcal{D}, X_j\right) = -\sum_{v} P\left(X_j = v\right) \log_2 P\left(X_j = v\right)$$

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

This scales by the entropy of the split, ignoring classes

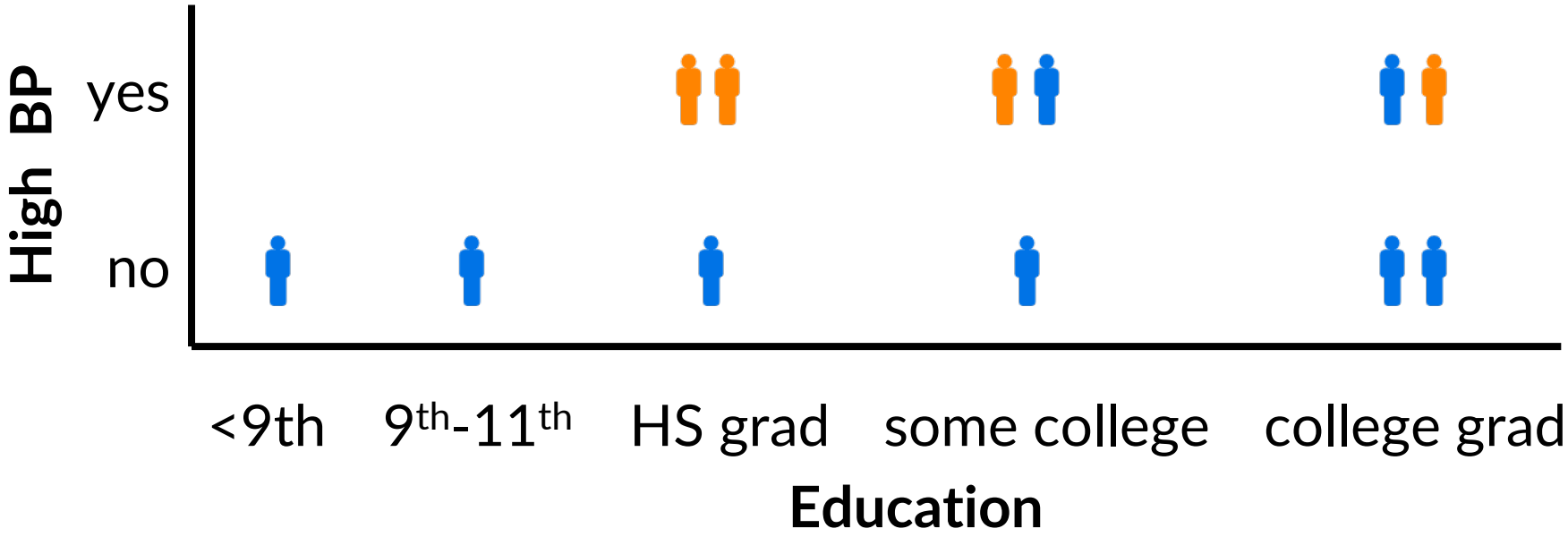
Split entropy measures the intrinsic information in the feature, not specific to the task --- it doesn't account for the class labels in any way.

Higher "split entropy" =>

- more child nodes (splits), and/or
- more even distribution of parent samples amongst the children.

Gain Ratio Example

Already Computed:
- H($\mathcal{D}$) = 0.918
- H ($\mathcal{D}$ | High BP) = 0.459
- H ($\mathcal{D}$ | Education) = 0.730
- IG($\mathcal{D}$ High BP) = 0.459
- IG($\mathcal{D}$, Education) = 0.188



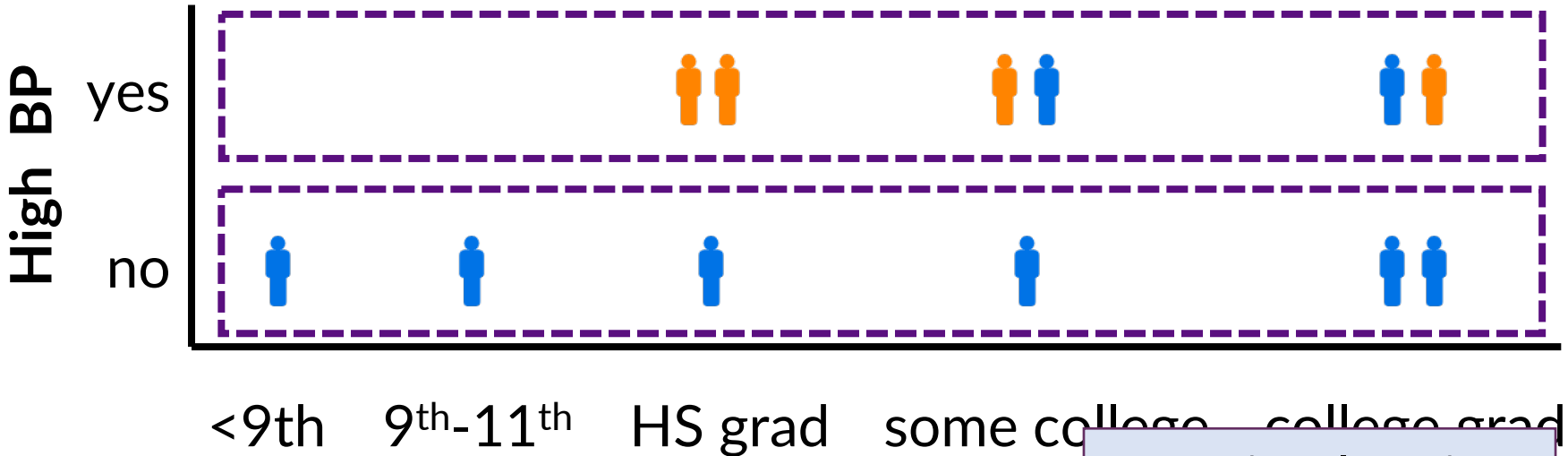
Need to compute:

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

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

Gain Ratio Example

Already Computed:
- H($\mathcal{D}$) = 0.918
- H ($\mathcal{D}$ | High BP) = 0.459
- H ($\mathcal{D}$ | Education) = 0.730
- IG($\mathcal{D}$ High BP) = 0.459
- IG($\mathcal{D}$, Education) = 0.188



= – 6/12 lg 6/12
– 6/12 lg 6/12
= 1

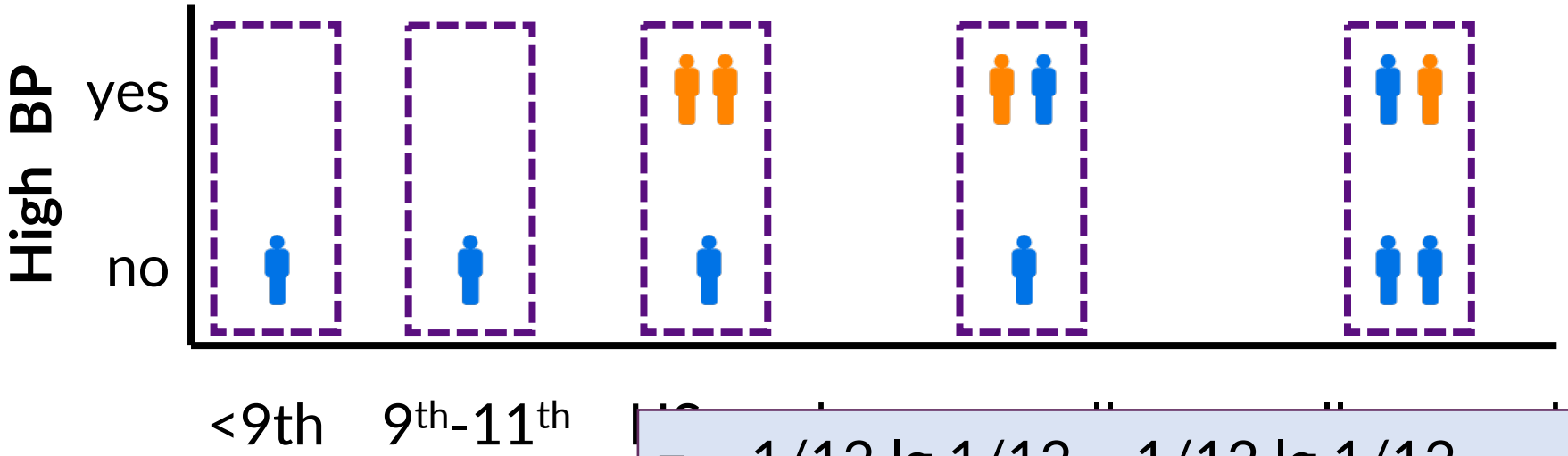
Need to compute:

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

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

Gain Ratio Example

Already Computed:
- H($\mathcal{D}$) = 0.918
- H ($\mathcal{D}$ | High BP) = 0.459
- H ($\mathcal{D}$ | Education) = 0.730
- IG($\mathcal{D}$ High BP) = 0.459
- IG($\mathcal{D}$, Education) = 0.188



= – 1/12 lg 1/12 – 1/12 lg 1/12
– 3/12 lg 3/12 – 3/12 lg 3/12
– 4/12 lg 4/12
= 2.1258

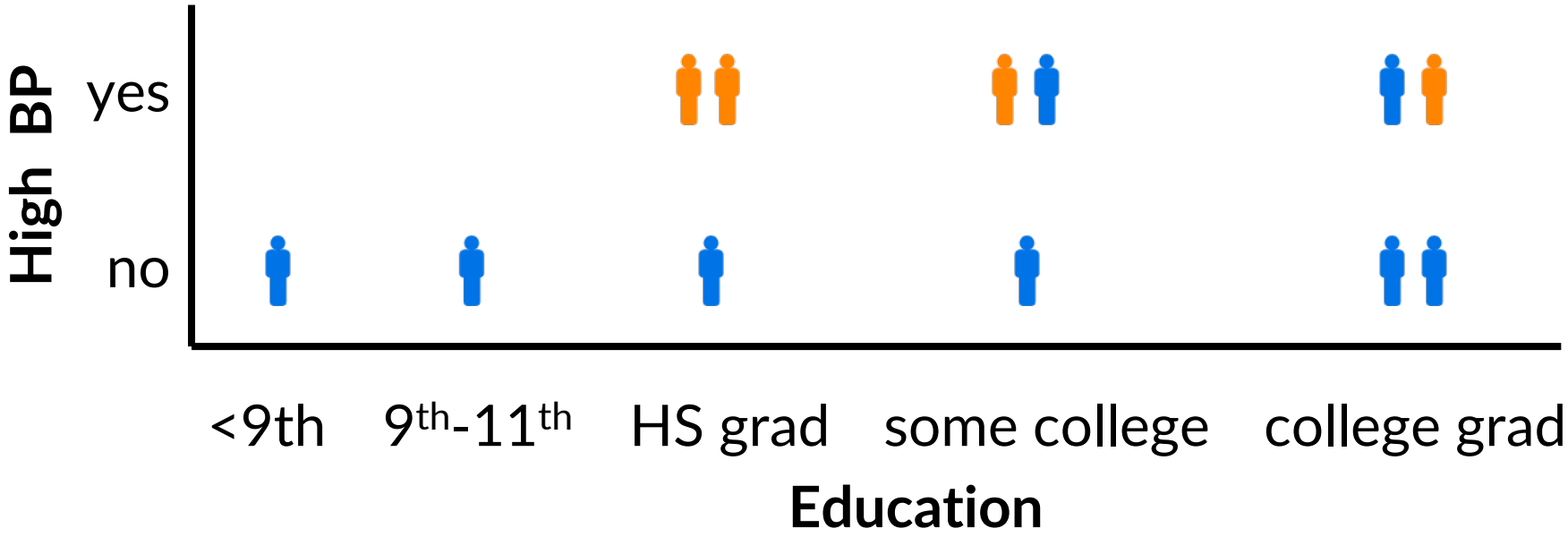
Need to compute:

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

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

Gain Ratio Example

Already Computed:
- H($\mathcal{D}$) = 0.918
- H ($\mathcal{D}$ | High BP) = 0.459
- H ($\mathcal{D}$ | Education) = 0.730
- IG($\mathcal{D}$ High BP) = 0.459
- IG($\mathcal{D}$, Education) = 0.188



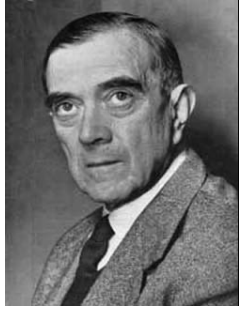
Need to compute:

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

GainRatio($\mathcal{D}$, Education) = IG($\mathcal{D}$, Education) / SplitInfo($\mathcal{D}$, Education)=0.188/2.126 = 0.088

Exercise: Try this with the patient ID feature.

Aside: Gini Index Reduction Criterion



(not a great guy)

There is another widely used criterion aside from IG and GR, the "Gini Index" for binary classification.

- Recall how we compute Information Gain = Entropy Reduction:
  - Entropy $H(\mathcal{D}) = \sum_c P(Y = c)(-\log_2 P(Y = c))$
  - Information Gain = Entropy of parent – Weighted Average Entropy of Children
- Analogously, Gini Index Reduction:
  - Gini index $\text{Gini}(\mathcal{D}) = \sum_c P(Y = c)\,(1 - P(Y = c))$
  - Gini gain = Gini of parent – Weighted Average Gini of Children

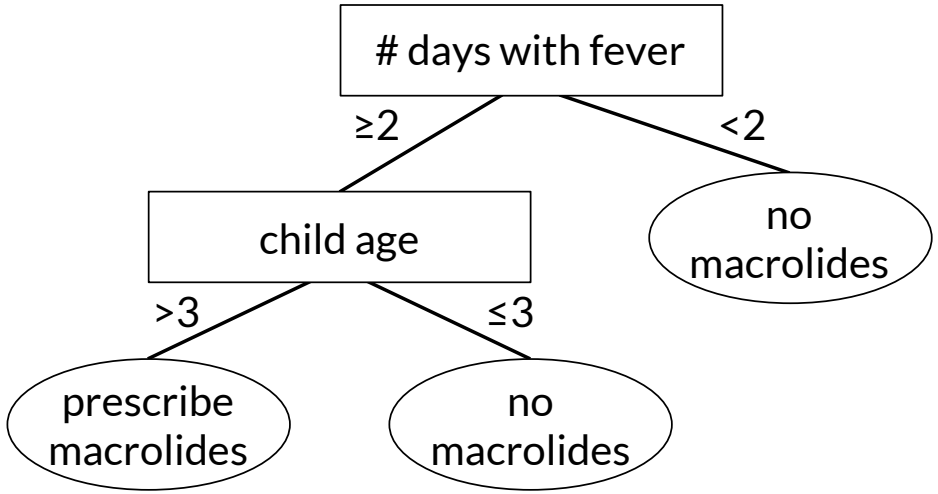
You will get to play with this in HW3.

Q: Does Gini index also prefer creating more children?

Yes. Discussion here: https://stats.stackexchange.com/questions/395278/the-reason-why-gini-index-is-in-favor-of-multivalued-attributes

Aside: Real-Valued Features

- Change to binary splits by choosing a threshold
- One method:
  - Sort instances by value, identify adjacencies with different classes



Days with Fever:	1	1	2	3	4	6
Prescribe macrolides?:	No	No	Yes	No	Yes	Yes

candidate splits

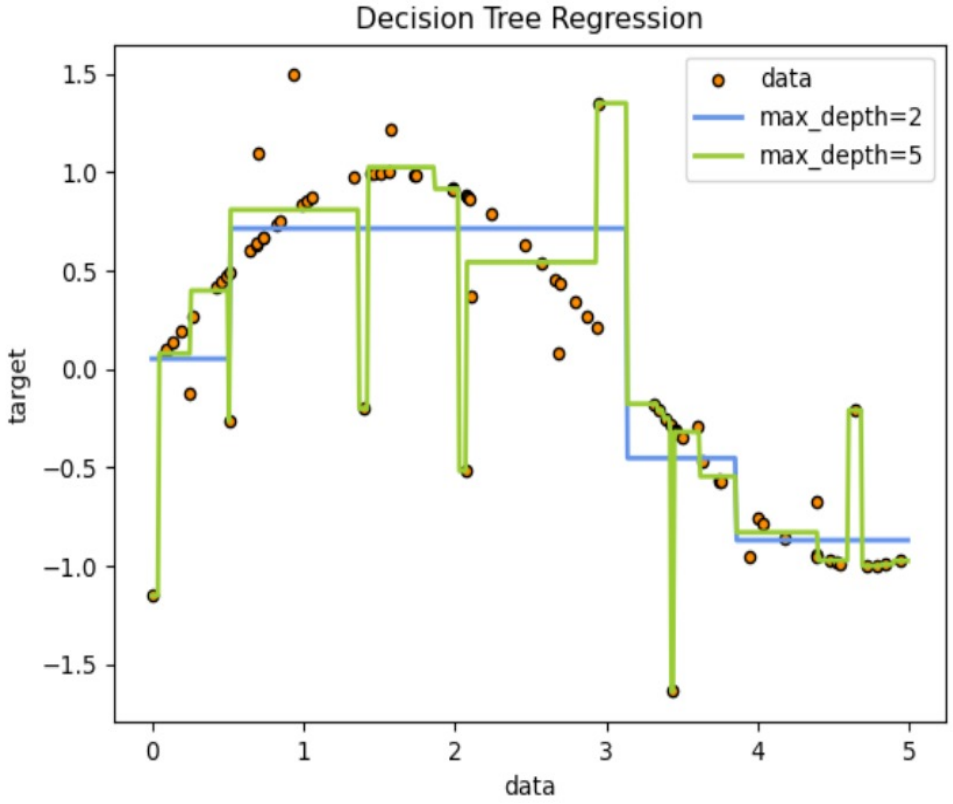
  - Then, choose among splits by IG or GR

This amounts to converting a continuous attribute X_j into a collection of binary attributes: $1[X_j > t_1], 1[X_j > t_2], 1[X_j > t_3]$, … before selecting highest IG / GR attrbutes

Aside: Decision Trees for Regression (Real-Valued Targets)

Everything remains the same except:

- Measure of impurity has to apply to continuous targets. E.g. standard deviation or entropy of continuous target
  - So, e.g., impurity reduction = Standard deviation of parent node – weighted average standard deviation of children nodes
- Making scalar label predictions at a leaf node:
  - Similar to KNNs for regression, simply take the average of the training target labels at the leaf node.



https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html

DT Training via Gain Ratio

Recap

A: Gain Ratio

We are Ready to Train the DT for Diabetes!

SEQN	RIDAGEYR	BMXWAIST	BMXHT	LBXTC	BMXLEG	BMXWT	BMXBMI	RIDRETH1	BPQ020	ALQ120Q	DMDEDUC2	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

Gain Ratio-Based Greedy DT Construction

SEQN	RIDAGEYR	BMXWAIST	BMXHT	LBXTC	BMXLEG	BMXWT	BMXBMI	RIDRETH1	BPQ020	ALQ120Q	DMDEDUC2	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-Racia	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-Racia	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-Racia	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

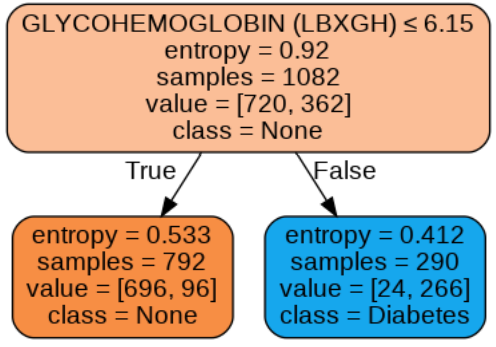
X_1 X_2 . . .

X_{14}

Given dataset $\mathcal{D} = [X, \boldsymbol{y}]$

- Pick feature X_j to split upon with the highest IG (or GainRatio)
- Partition $\mathcal{D}$ via X_j
- Recurse until nodes are homogenous

X_{14} (LBXGH) ≤ 6.15 has the highest IG



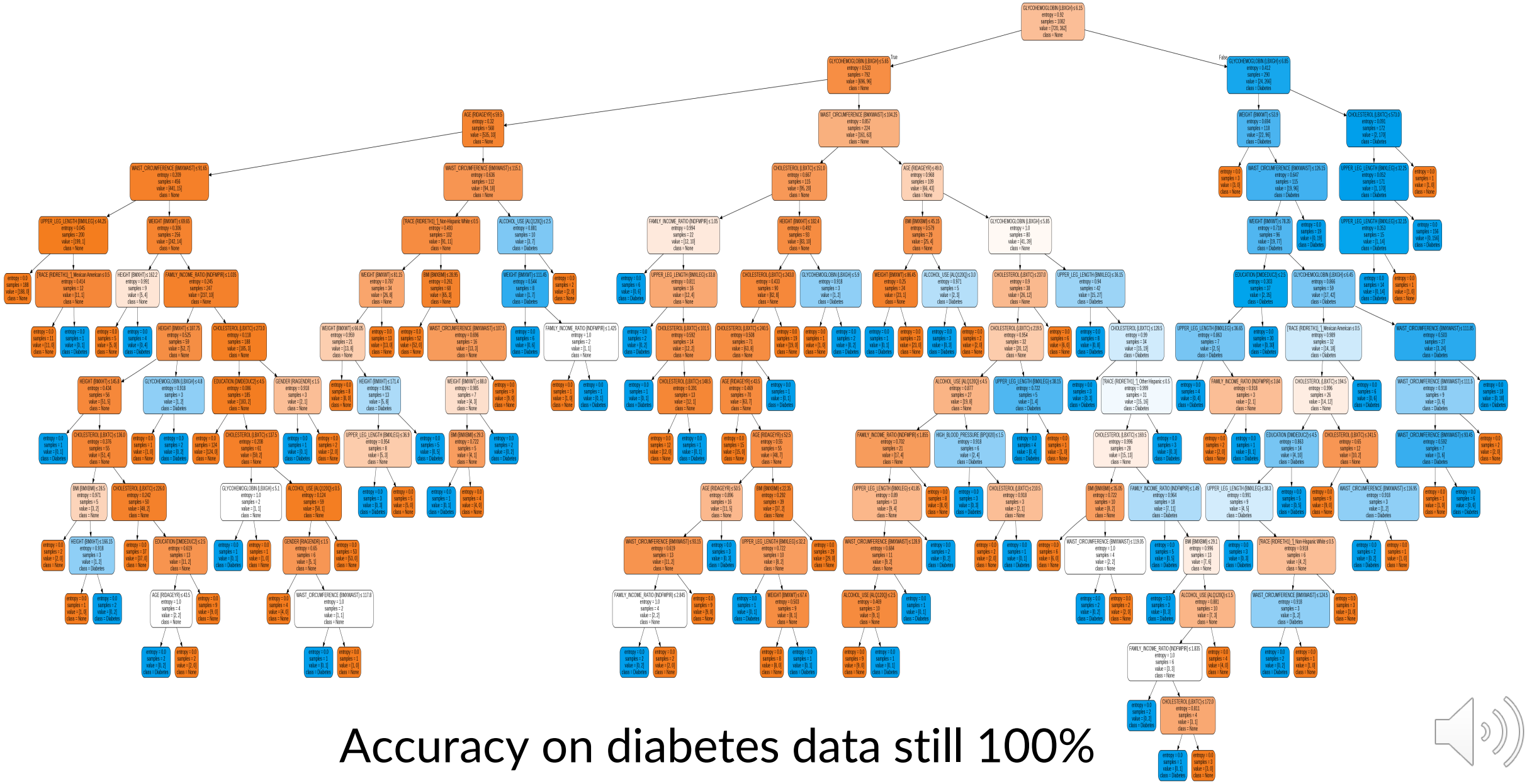
Dataset partition $\mathcal{D}$[LBXGH ≤ 6.15]

SEQN	RIDAGEYR	BMXWAIST	BMXHT	LBXTC	BMXLEG	BMXWT	BMXBMI	RIDRETH1	BPQ020	ALQ120Q	DMDEDUC2	RIAGENDR	INDFMPIR	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 $\mathcal{D}$[LBXGH > 6.15]

SEQN	RIDAGEYR	BMXWAIST	BMXHT	LBXTC	BMXLEG	BMXWT	BMXBMI	RIDRETH1	BPQ020	ALQ120Q	DMDEDUC2	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
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

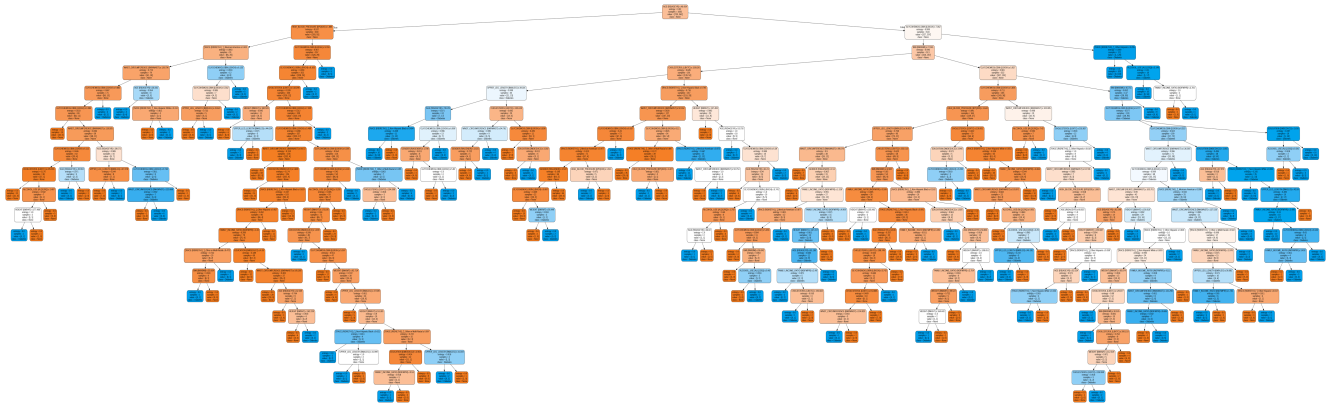
Decision Tree (Version 1)



Accuracy on diabetes data still 100%

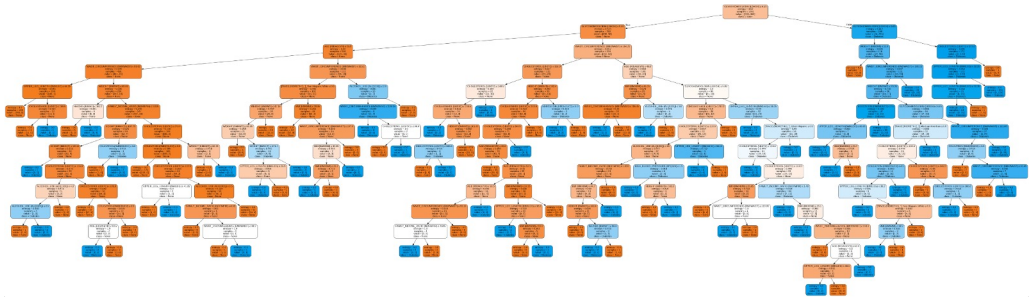
Diabetes DT – Random vs IG Features

DT with random feature splits



Accuracy on diabetes data = 100%

DT via IG



Accuracy on diabetes data = 100%

- Well, it is smaller while retaining 100 % accuracy on our training data
- Still rather complex, though …

Feedback From Our Physician Friend

Thanks for those models!

Dinesh Jayaraman (seas.upenn.edu)

Thanks for those models!

Hi Dinesh,
Thanks so much for sending those decision tree models along!

They worked really great on the dataset I had sent you before, but we're collecting some new data and noticing some weird issues. Could you take a look at these results and let us know if you have any thoughts?

Best,
Your fictional physician friend

Accuracy – Decision Tree (Version 1)

Original Patient Data: 100.000 % (n = 1082)

New Patient Data: 82.796 % (n = 465)

Recall: Overfitting

This is just classic “**overfitting**”

Larger, more complex models sometimes do poorly on new data, even if they perform on par or better than small models on the training data.

Combating Overfitting

Avoiding Overfitting

How can we avoid overfitting?

- Acquire more training data
- Remove irrelevant attributes (manual process – not always possible)
- **Stop growing when data split is not statistically significant**
  - E.g. a pre-selected maximum depth, minimum #samples, minimum #samples in each class
- **Grow full tree, then post-prune**

Try various tree hyperparameters (like tree depth and 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

Based on Slide by Pedro Domingos

Overview: Reduced-Error Pruning

- Split the original training data into training and validation sets

Training Stage

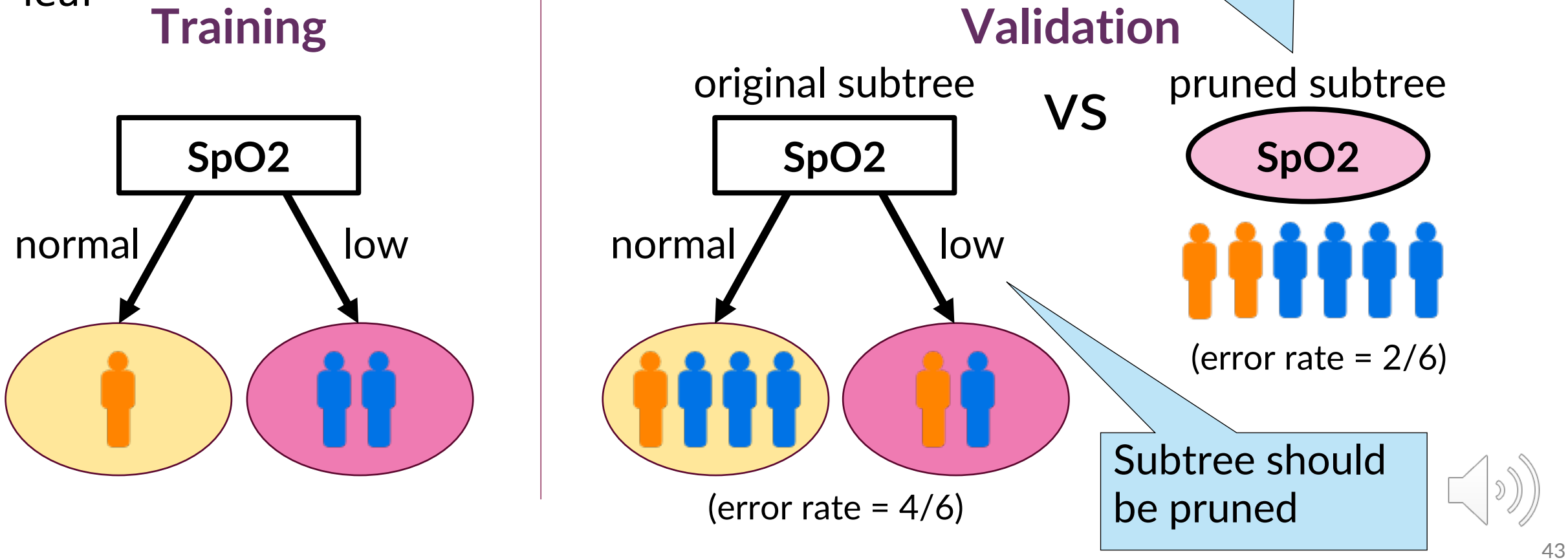
- Grow the decision tree based on the training set

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

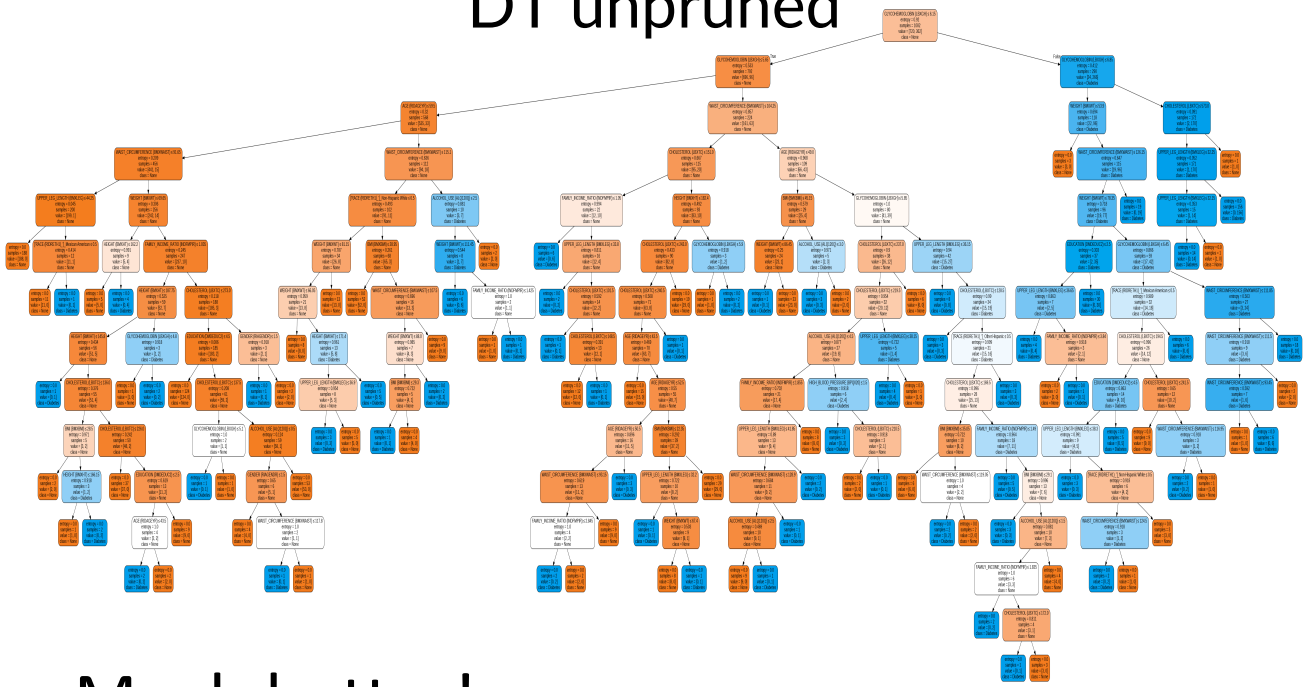
Overview: Reduced-Error Pruning

- Pruning replaces a whole subtree by a leaf node
- Replacement occurs if the expected error rate of the subtree on validation data is greater than that of the leaf

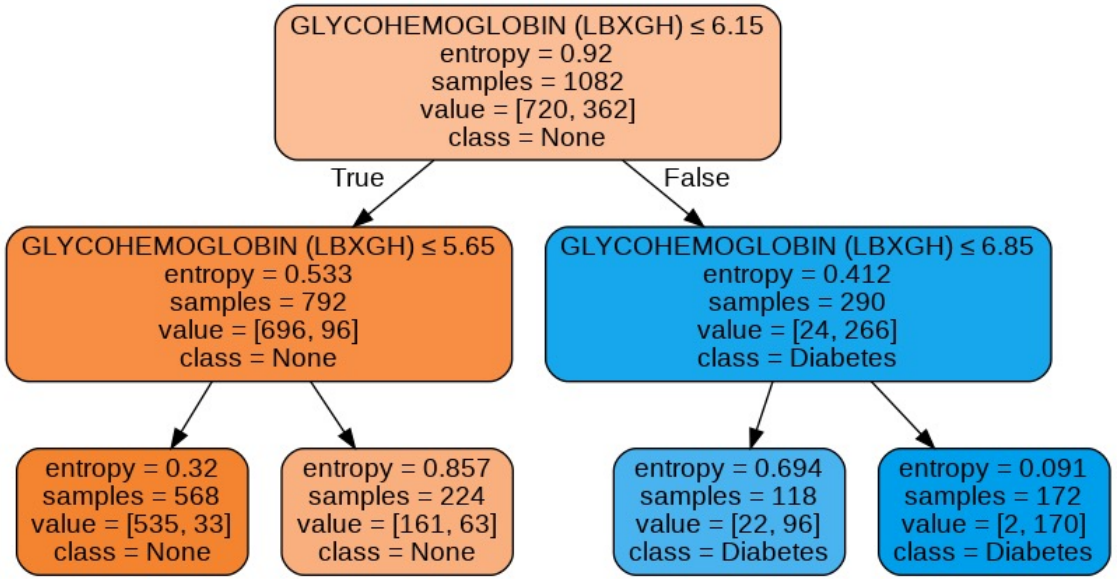


Reduced-Error Pruning on the Diabetes DT

DT unpruned



DT pruned

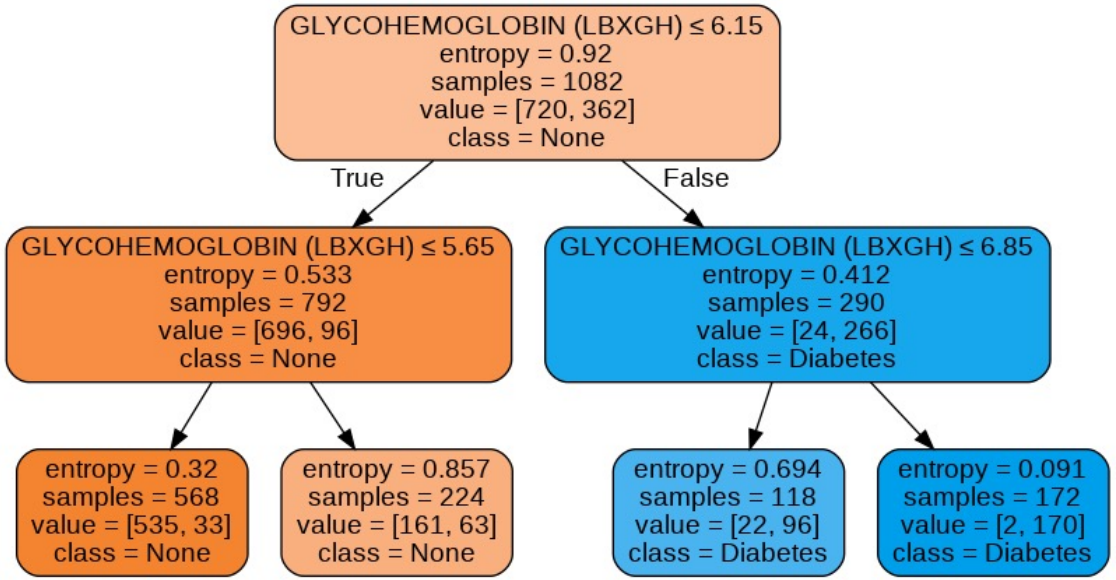


Much better!

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

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!

You'll get to play around with this data some more in HW3.

Are DTs feature scaling invariant?

- Yes, DTs are naturally feature-scaling invariant in most implementations.
  - Information Gain, Gain Ratio etc. don’t rely on the specific values of the features, so scaling a feature doesn’t affect the tree training, and it predicts identical outputs afterwards.
  - In fact, more general than even just “scaling”, DTs are usually invariant under arbitrary monotone transformations of the input.

Where are the parameters in Decision Trees?

- Parameters to select at each node:
  - Which attribute to select?
  - Sometimes, also how to create branches from it? E.g. which threshold to set on a continuous variable?

- For a fixed maximum depth d, a decision tree has a fixed number of parameters (or at least a fixed *maximum* number of parameters).

- In general, we don't know the number of nodes, and consequently, the number of parameters. Non-parametric! just like k-NN.

Are We Optimizing A Loss Function?

- Trivially, we are of course seeking high classification accuracy.

- But our optimizer is *greedy*.
  - Local optimization of a “heuristic function” such as the information gain.

- There is no notion of a specific loss function for which we can claim that our ID3 / C4.5 training approach will “finding the decision tree that incurs the lowest loss”.

Decision Tree Algorithm Variants Overview

ID3

- Information gain on nominal features

C4.5

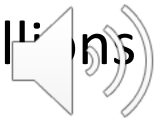
- Can use info gain or gain ratio
- Nominal or numeric features
- Missing values
- Post-pruning
- Rule generation

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)

Other Algorithms

- SPRINT, SLIQ: multiple sequential scans of data (1M instances)
- VFDT: at most one sequential scan (billions of instances)



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
- 👎 Limited predictive power
- 👎 Heuristic-Based Greedy Training

DTs are the basic component of what is arguably the single best “off-the-shelf” ML algorithm for arbitrary problems, particularly with tabular data, called XGBoost (more on this soon).

More Administrivia: Projects

- 3 % of course grade (20% for full project: 3 + 5 + 12)
- **Team information due Fri Feb 20.**
  - 3 members per team.
  - Submit information on google form (announcement soon).
- **Project proposal due Wed Mar 1.**
  - A proposal template document will be released in the coming days.
  - A project mentor will be assigned to you based on your proposal.

- Guidance on project topics: See next 2 slides.

“Standard” Projects

- The recommended option barring exceptional cases.
- Tied closely to any one from a pre-approved list of Kaggle projects (announcement soon).
- **Part 1: Implementation**
  - **Option 1:** Extensive evaluation of design decisions in pre-existing codebases.
    - Evaluate the design decisions in existing Kaggle submissions, e.g. current leading submissions, or other codebases on the web for this problem. Always cite and acknowledge.
    - Recommendations:
      - For tabular datasets, significant feature engineering and try several models
      - For image datasets, try different neural network architectures etc.
  - **Option 2:** New ML approach. A new ML approach, not directly building on top of current codebases
    - Typically a learning strategy (e.g., semi-supervised learning) or a neural network architecture
- **Part 2: Evaluation**
  - **Part 2a:** Systematic evaluation of hyperparameters (e.g., regularization, learning rate, etc.)
  - **Part 2b:** Evaluate on test data distributions different from training data
    - E.g. Add synthetic noise to test set, train-test split based on demographic features or time
    - Plot performance measures vs. degree of shift (e.g. for demographic features, include X% fraction of minority in the training set, where X is degree of shift)
    - Particularly interesting to identify “small” shifts that break the model.

- No collaboration outside your project team.
- Public submission to Kaggle leaderboard at end of project period together with code. **You will not be graded only on leaderboard position though.** More creative and ambitious projects will be held to lower final performance standards than more incremental projects.

“Non-Standard” Projects

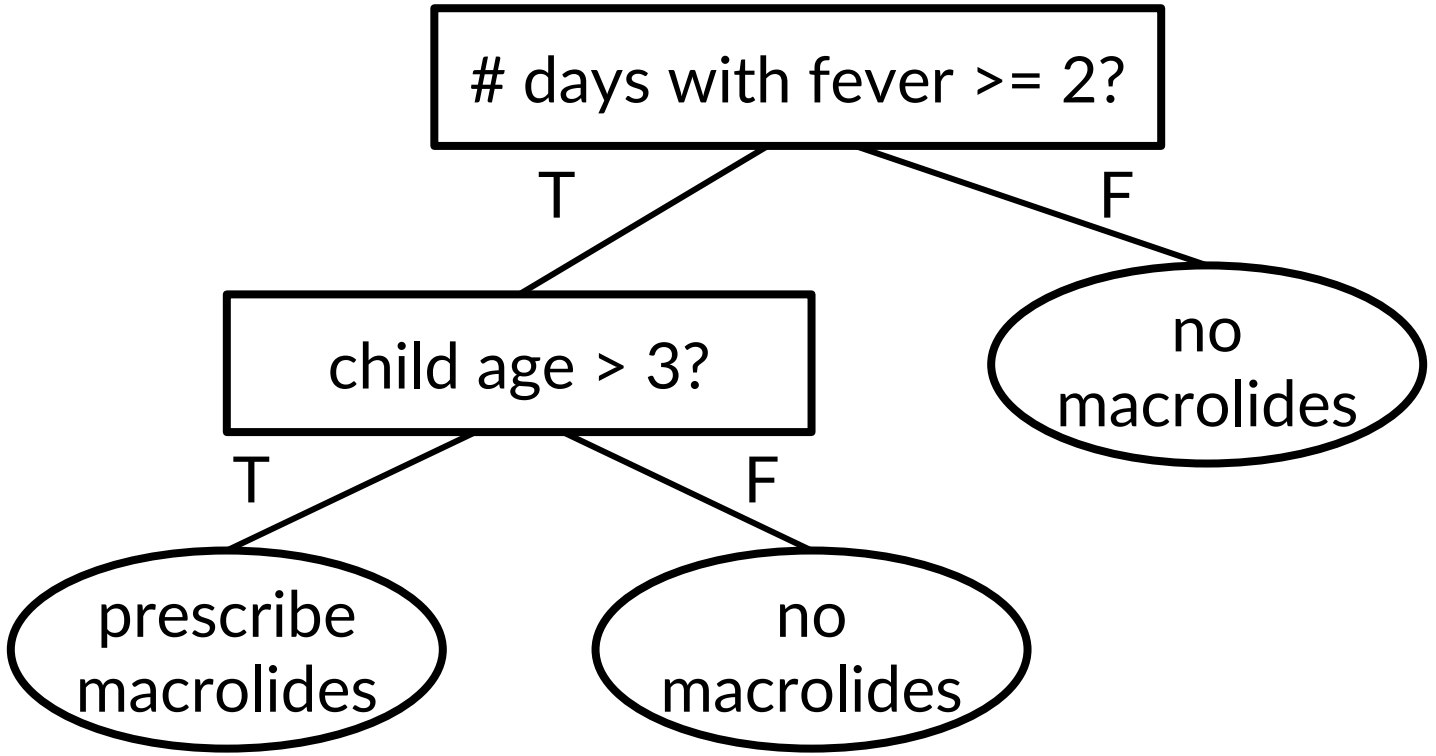
- Strongly recommended that you use the “standard” option from the last slide.
- If you have good reason to go beyond this, e.g., you would like to propose a project tied to your PhD research, you could do so, but these submissions will go through greater scrutiny for approval.

Lecture 10: Learning Ensembles

CIS 4190/5190

Spring 2023

Decision Tree Shortcomings

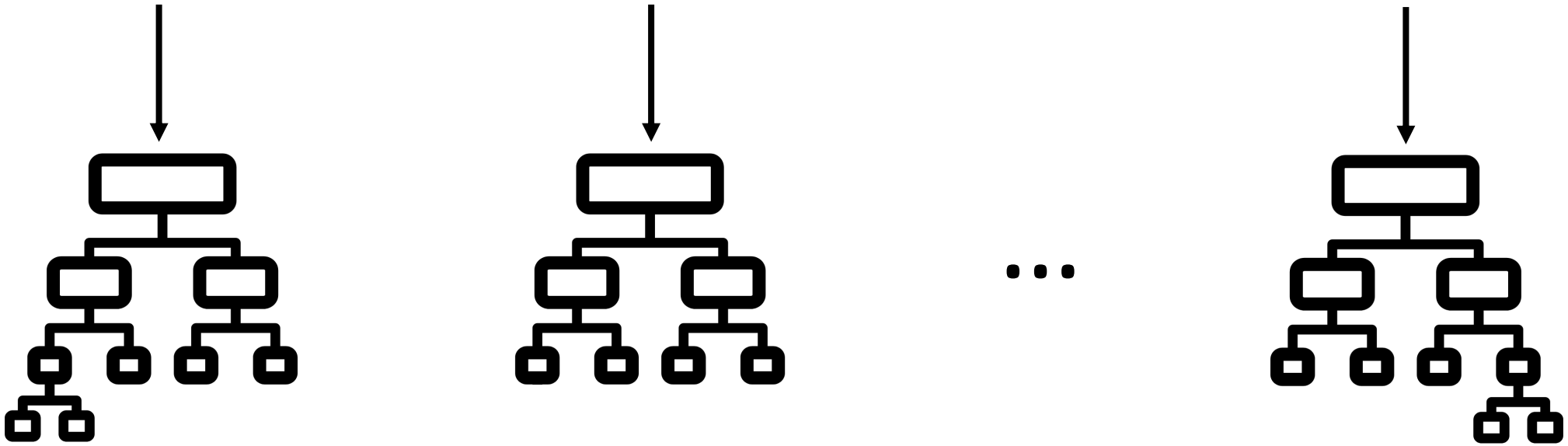


Decision tree example from: Martignon and Monti. (2010). Conditions for risk assessment as a topic for probabilistic education. *Proceedings of the Eighth International Conference on Teaching Statistics* (ICOTS8).

Decision Tree Shortcomings

- Hard to manage bias-variance tradeoff
  - Small depth → High bias, low variance
  - Large depth → Small bias, high variance
  - What if we need to grow deeper in some branches but not others?
- Can we manage this tradeoff in a principled way?
- **Idea:** Random forests
  - Grow large decision trees
  - Rather than prune, average many of them!

Random Forests



Random Forests

- Train many decision trees and average them!
  - Large depth → High variance, low bias
  - Averaging many decision trees → average away “irrelevant” variance
- Very powerful model family in practice

Ensembles

- More generally, **ensembles** are an effective strategy for mitigating the bias-variance tradeoff

- **Approaches so far:**
  - Different model family
  - Feature engineering

- **Ensembles:**
  - Combine models to reduce bias without increasing variance

Ensemble Learning

- **Step 1:** Learn a set of "base" models $f_1, \ldots, f_k$
- **Step 2:** Construct model $F(x)$ that combines predictions of $f_1, \ldots, f_k$

Example: Netflix Movie Recommendations

- **Goal:** Predict how a user will rate a movie based on:
  - The user's ratings for other movies
  - Other users' ratings for this movie (and others)
  - **No features!**

- **Netflix Prize (2007-2009):** $1 million for the first team to do 10% better than the existing Netflix recommendation system

- **Winner:** BellKor's Pragmatic Chaos
  - An ensemble of 800+ rating systems

Ensembles of Decision Trees

- **Strategy 1:** Random forests
- **Strategy 2:** Gradient boosted decision trees
- Among the most powerful and widely-used models for “tabular” data (i.e., not images, text, graphs, or other highly structured data)

Ensemble Design Decisions

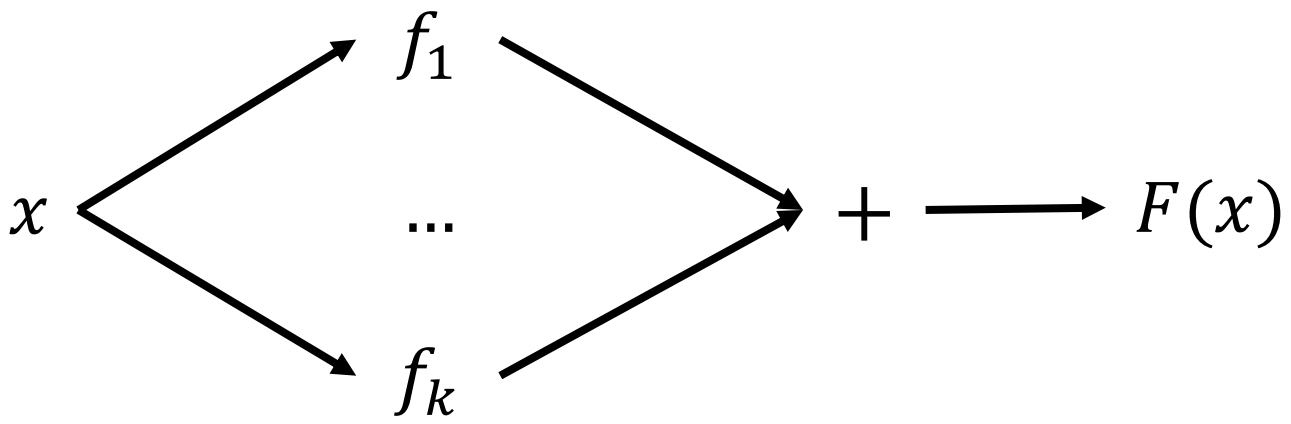
- How to learn the base models?
- How to combine the learned base models?

Ensemble Design Decisions

- How to learn the base models?
- **How to combine the learned base models?**

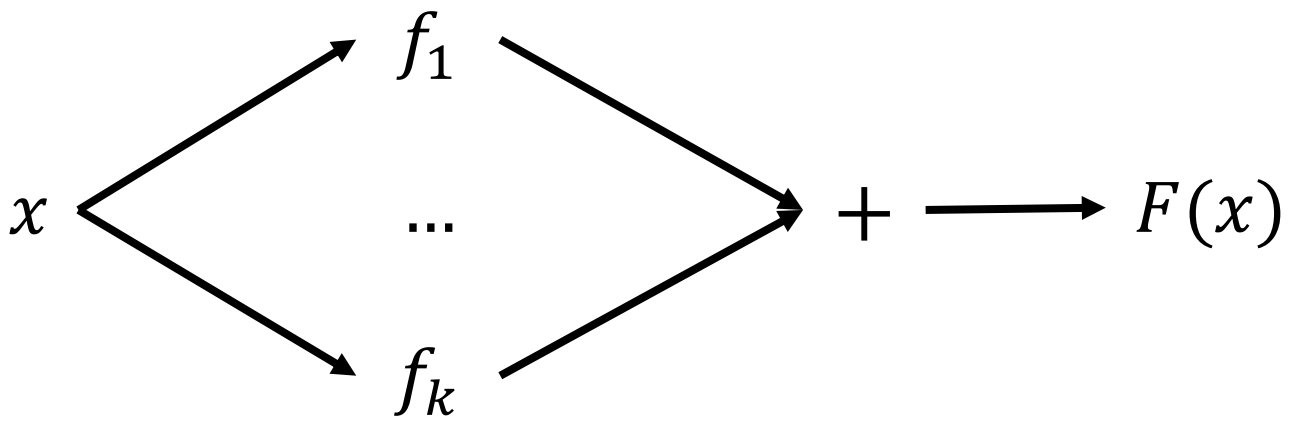
Combining Learned Base Models

- **Regression:** Average predictions $F(x) = \frac{1}{k}\sum_{i=1}^{k} f_i(x)$
  - Works well if the base models have similar performance



Combining Learned Base Models

- **Classification:** Majority vote $F(x) = 1\left(\sum_{i=1}^{k} f_i(x) \geq \frac{k}{2}\right)$ (for binary)
  - Can also average probabilities for classification



Combining Learned Base Models

- Can use weighted average:

$$F(x) = \sum_{i=1}^{k} \beta_i \cdot f_i(x)$$

- Can fit weights using linear regression on second training set

- More generally, can fit a second layer model

$$F(x) = g_\beta\big(f_1(x), \ldots, f_k(x)\big)$$

Combining Learned Base Models

- Second model as “mixture of experts”:

$$F(x) = \sum_{i=1}^{k} g(x)_i \cdot f_i(x)$$

- Second stage model predicts weights over “experts” $f_i(x)$

Combining Learned Base Models

- Second model as “mixture of experts”:
  - **Special case:** $g(x)$ is one-hot
  - **Advantage:** Only need to run $g(x)$ and $f_{g(x)}(x)$