Announcements

- HW1 due on Wed at 8 p.m. + HW 2 release the same day
- Quiz for last week due on Thu
- Recordings releases now running 1 week behind, follow the link on Ed post #1
- Recitations?
- Debugging during OHs:
  - Systematic debugging is an art worth learning! Lots of resources with tips. E.g.:
    - https://applab.unc.edu/posts/2021/02/17/debugging-tips/
  - 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.

CIS 4190/5190: Lec 09 Mon Sep 30, 2024. Part 1.

K-Nearest Neighbors

Optional Extra Readings: kNN and Decision Trees

- Bishop, Pattern Recognition and Machine Learning, Ch 2.5:
  - https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf
- Tom Mitchell, Machine Learning Textbook, Ch 3: http://www.cs.cmu.edu/~tom/files/MachineLearningTomMitchell.pdf
- R2D3's visualizations:
  - Intro to decision trees: http://www.r2d3.us/visual-intro-to-machine-learning-part-1/
  - Bias and variance in the context of decision trees: http://www.r2d3.us/visual-intro-to-machine-learning-part-2/

Optional Extra Readings: Logistic Regression

- Hastie and Tibshirani Ch 4.1-4
- Hardt and Recht Ch 3: Supervised Learning
  - Linear and logistic regression introduced as instances of a “perceptron”: https://mlstory.org/supervised.html
- d2l.ai interactive textbook chapter on logistic regression, taught as a simple instance of a neural network: https://d2l.ai/chapter_linear-classification/index.html (recommended to use in pytorch mode)

So far, we have seen:

- Machine learning methods are defined by:
  - A model family / hypothesis space
  - An objective function
  - An optimization approach

So far, we have seen:

- Machine learning methods are defined by:
  - A model family / hypothesis space
    - Defined in terms of some fixed-length parameter vector $\beta \in \mathbb{R}^D$
      - Linear regression: $\hat{y} = \beta^T x$
      - Logistic regression: $p(\hat{y} = 1) = \sigma(\beta^T x)$
  - An objective function
  - An optimization approach

So far, we have seen:

- Machine learning methods are defined by:
  - A model family / hypothesis space
    - Defined in terms of some fixed-length parameter vector $\beta \in \mathbb{R}^D$
      - Linear regression: $\hat{y} = \beta^T x$
      - Logistic regression: $p(\hat{y} = 1) = \sigma(\beta^T x)$
  - An objective function
    - $L(\beta; Z)$ defines what it means for parameters β to be good given training set Z,
      - e.g. MSE for linear regression, or maximum-likelihood logistic regression objective
  - An optimization approach

So far, we have seen:

- Machine learning methods are defined by:
  - A model family / hypothesis space
    - Defined in terms of some fixed-length parameter vector $\beta \in \mathbb{R}^D$
      - Linear regression: $\hat{y} = \beta^T x$
      - Logistic regression: $p(\hat{y} = 1) = \sigma(\beta^T x)$
  - An objective function
    - $L(\beta; Z)$ defines what it means for parameters β to be good given training set Z,
      - e.g. MSE for linear regression, or maximum-likelihood logistic regression objective
  - An optimization approach
    - Some process of searching for optimal parameter vector β

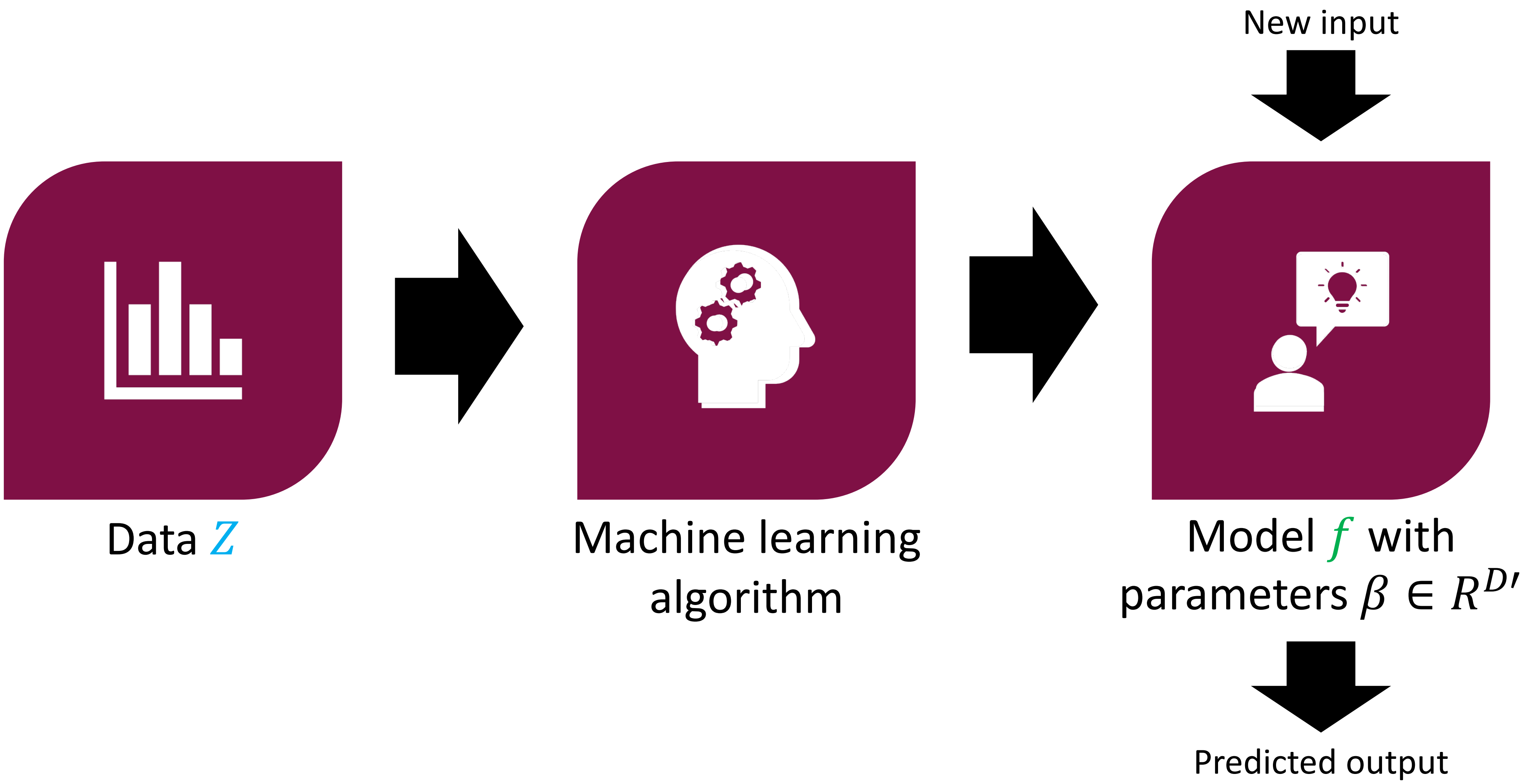
So far, we have seen:

- Machine learning methods are defined by:
  - A model family / hypothesis space
    - Defined in terms of some fixed-length parameter vector $\beta \in \mathbb{R}^D$
      - Linear regression: $\hat{y} = \beta^T x$

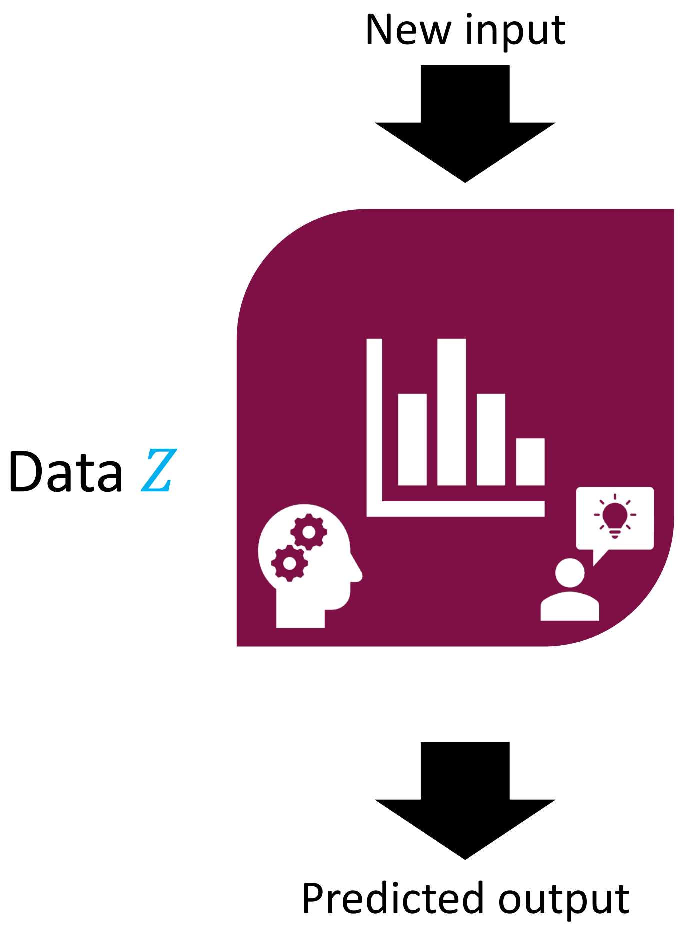
But not all machine learning approaches fit into this framework!

    - $L(\beta; Z)$ defines what it means for parameters β to be good given training set Z,
      - e.g. MSE for linear regression, or maximum-likelihood logistic regression objective
  - An optimization approach
    - Some process of searching for optimal parameter vector β

Recall: The Typical Machine Learning Pipeline



k-Nearest Neighbors.
A Simple Approach, Connected Directly to The Data

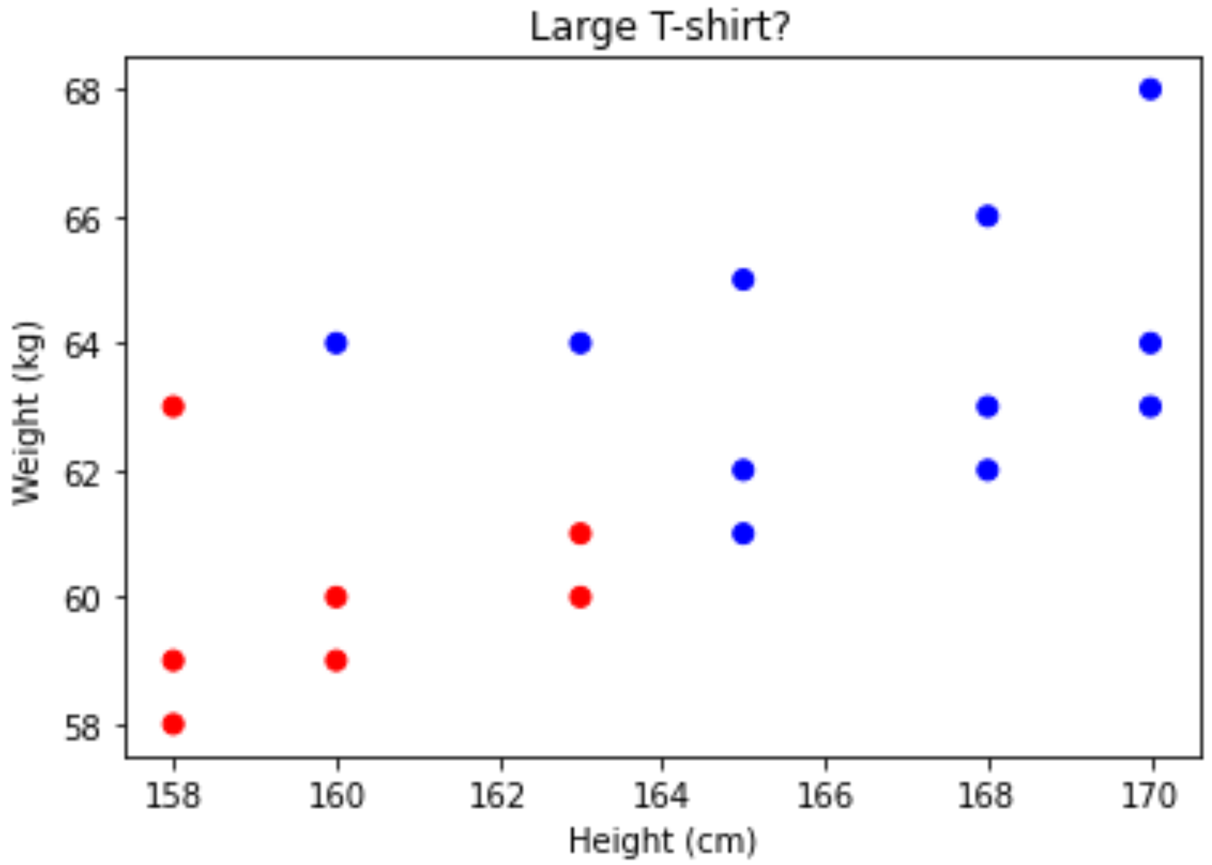


Note: this schematic seems to skip any explicit “model training” on data.

The data *is* the model.

How might this work?

Setup: Binary Classification (Training Data)



Blue: “T” i.e. Large t-shirt

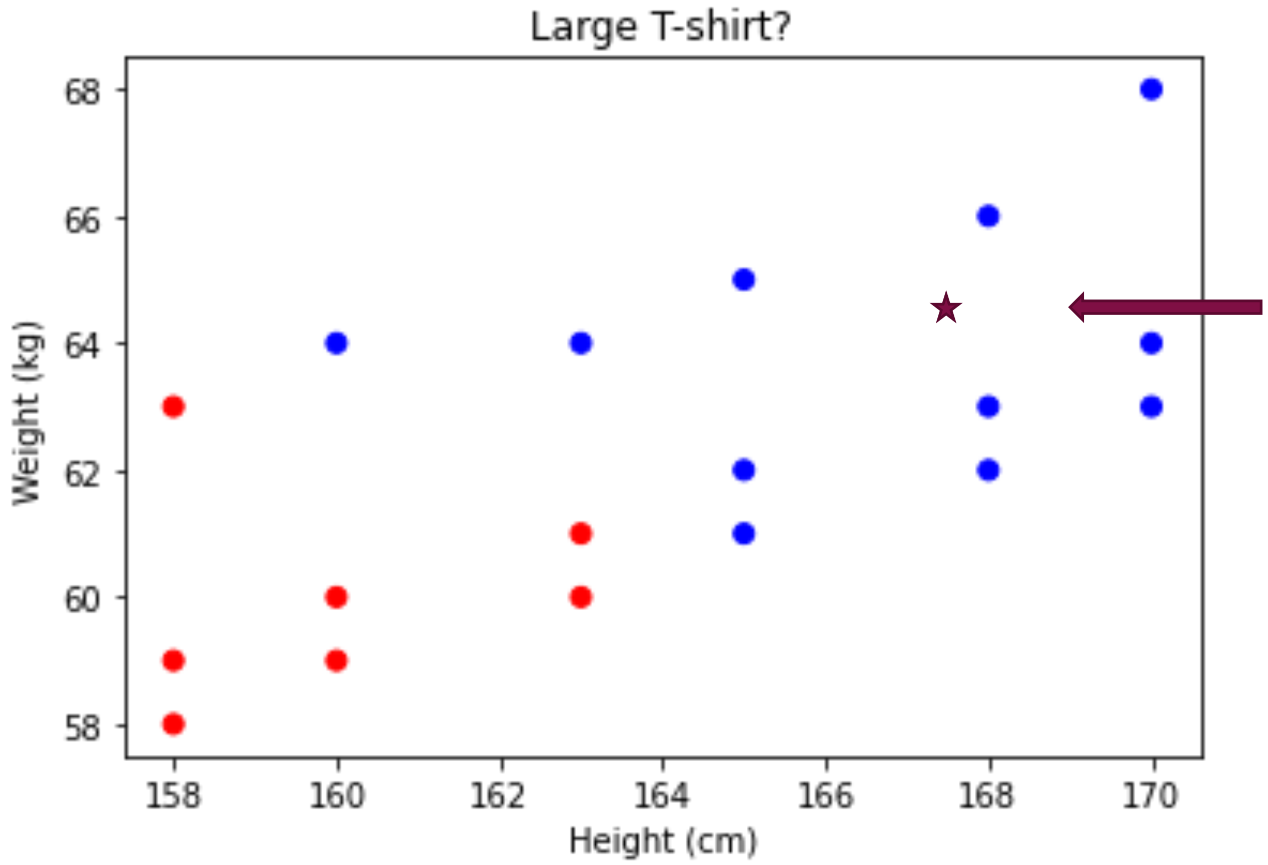
Red: “F” i.e. Medium t-shirt

Height (cm)	Weight (kg)	Large (vs Medium) t-shirt?
158	58	F
158	59	F
158	63	F
160	59	F
160	60	F
163	60	F
163	61	F
160	64	T
163	64	T
165	61	T
165	62	T
165	65	T
168	62	T
168	63	T
168	66	T
170	63	T
170	64	T
170	68	T

x_i y_i

Based on data from https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html

Test Time! Guess the Label For A New Sample?



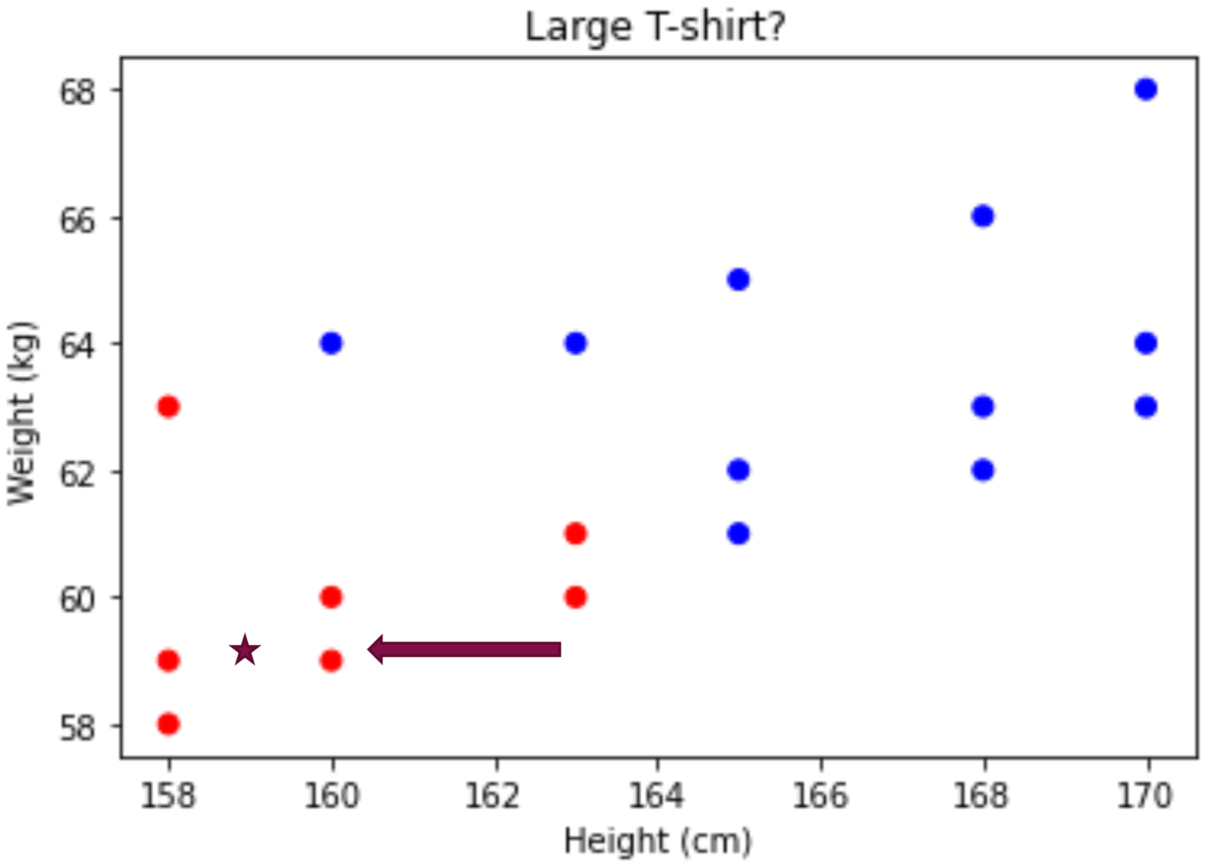
Blue: “T” i.e. Large t-shirt

Red: “F” i.e. Medium t-shirt

	Height (cm)	Weight (kg)	Large (vs Medium) t-shirt?	
	158	58	F	
x_i	158	59	F	y_i
	158	63	F	
	160	59	F	
	160	60	F	
	163	60	F	
	163	61	F	
	160	64	T	
	163	64	T	
	165	61	T	
	165	62	T	
	165	65	T	
	168	62	T	
	168	63	T	
	168	66	T	
	170	63	T	
	170	64	T	
	170	68	T	

Based on data from https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html

Test Time! Guess the Label For A New Sample?



Blue: “T” i.e. Large t-shirt

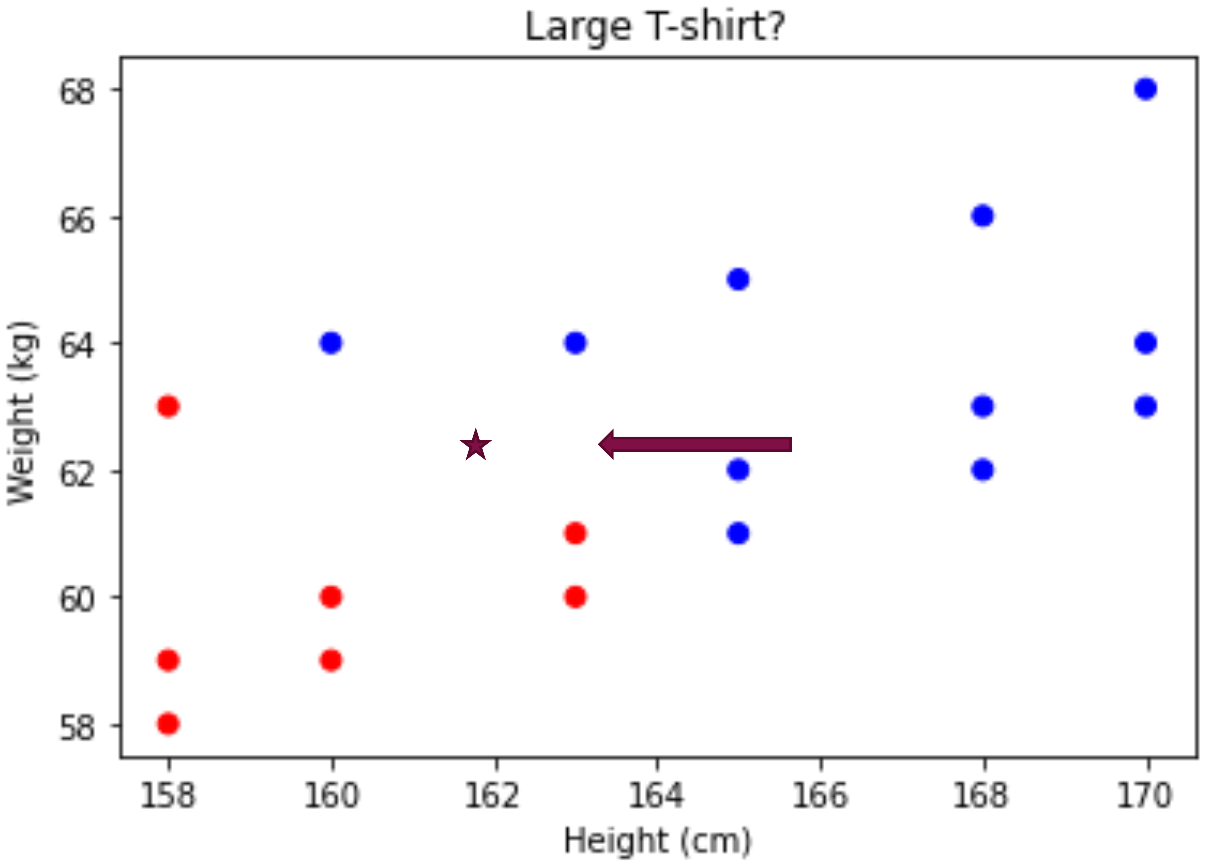
Red: “F” i.e. Medium t-shirt

Height (cm)	Weight (kg)	Large (vs Medium) t-shirt?
158	58	F
158	59	F
158	63	F
160	59	F
160	60	F
163	60	F
163	61	F
160	64	T
163	64	T
165	61	T
165	62	T
165	65	T
168	62	T
168	63	T
168	66	T
170	63	T
170	64	T
170	68	T

x_i y_i

Based on data from https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html

Test Time! Guess the Label For A New Sample?



Blue: “T” i.e. Large t-shirt

Red: “F” i.e. Medium t-shirt

	Height (cm)	Weight (kg)	Large (vs Medium) t-shirt?	
	158	58	F	
x_i	158	59	F	y_i
	158	63	F	
	160	59	F	
	160	60	F	
	163	60	F	
	163	61	F	
	160	64	T	
	163	64	T	
	165	61	T	
	165	62	T	
	165	65	T	
	168	62	T	
	168	63	T	
	168	66	T	
	170	63	T	
	170	64	T	
	170	68	T	

Based on data from https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html

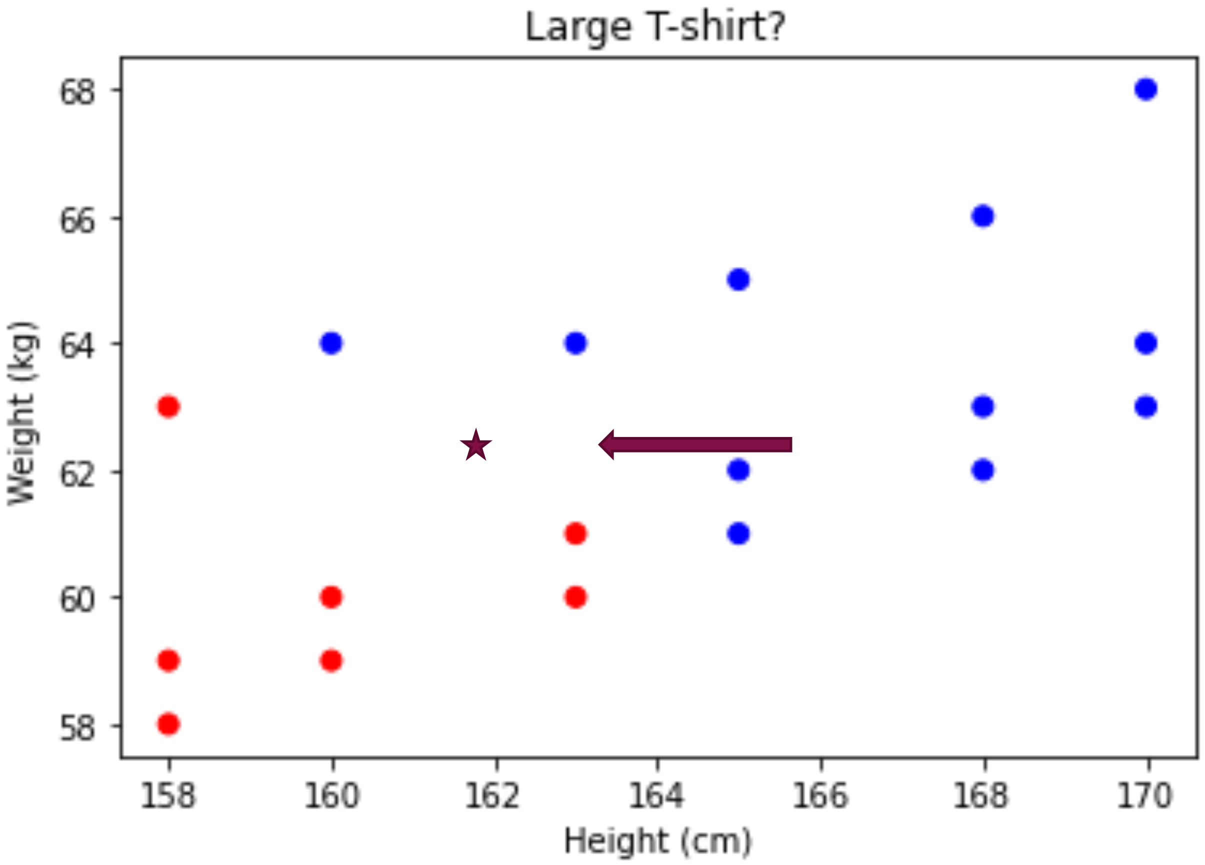
k-Nearest Neighbors (kNN)

- **kNN Classification:** To predict category label y of a new point $\boldsymbol{x}$:
  - Find k "nearest neighbors"
  - Assign the majority label

- **kNN regression:** To predict numeric value y of a new point $\boldsymbol{x}$:
  - Find k "nearest neighbors"
  - Average the values associated with the neighbors

In each case, varying k could change the predictions

kNN Prediction: What Label?



Blue: “T” i.e. Large t-shirt

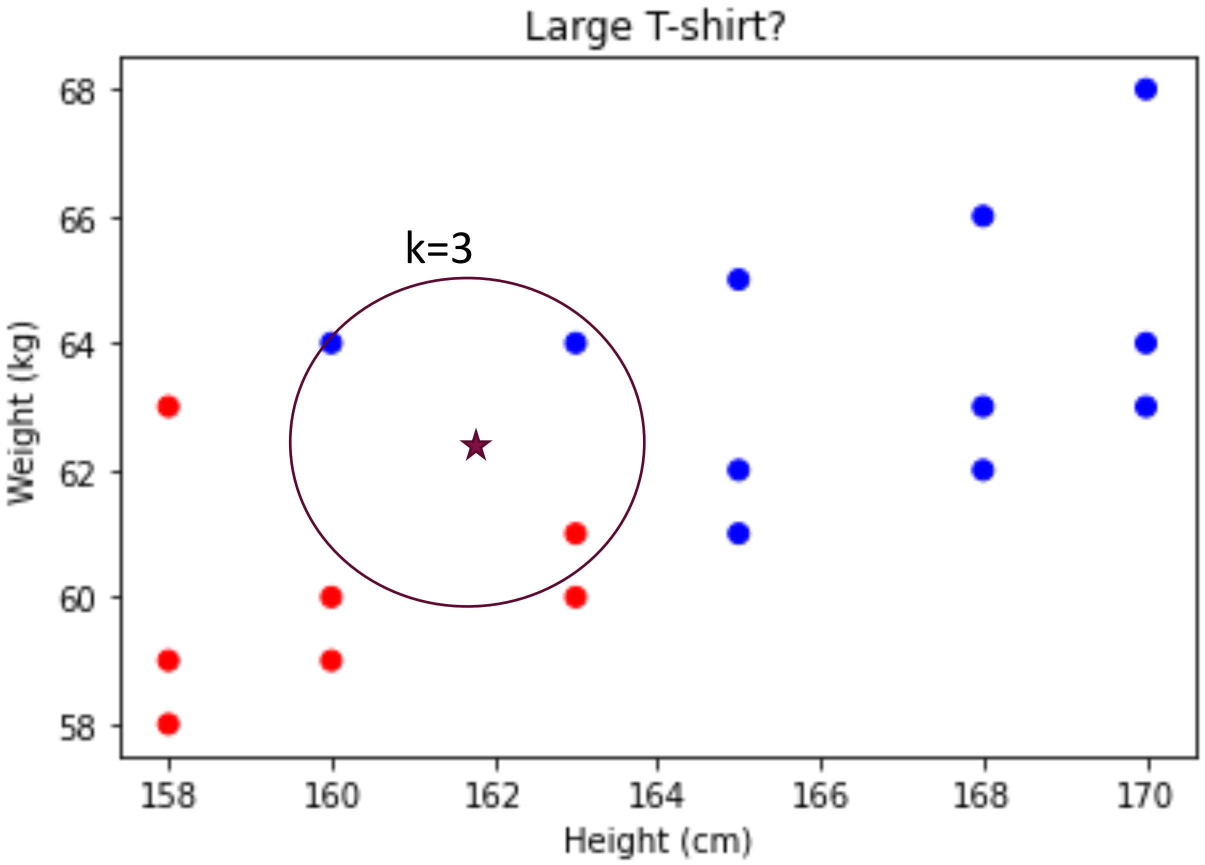
Red: “F” i.e. Medium t-shirt

Height (cm)	Weight (kg)	Large (vs Medium) t-shirt?
158	58	F
158	59	F
158	63	F
160	59	F
160	60	F
163	60	F
163	61	F
160	64	T
163	64	T
165	61	T
165	62	T
165	65	T
168	62	T
168	63	T
168	66	T
170	63	T
170	64	T
170	68	T

x_i y_i

Based on data from https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html

kNN Prediction: What Label?



Blue: “T” i.e. Large t-shirt

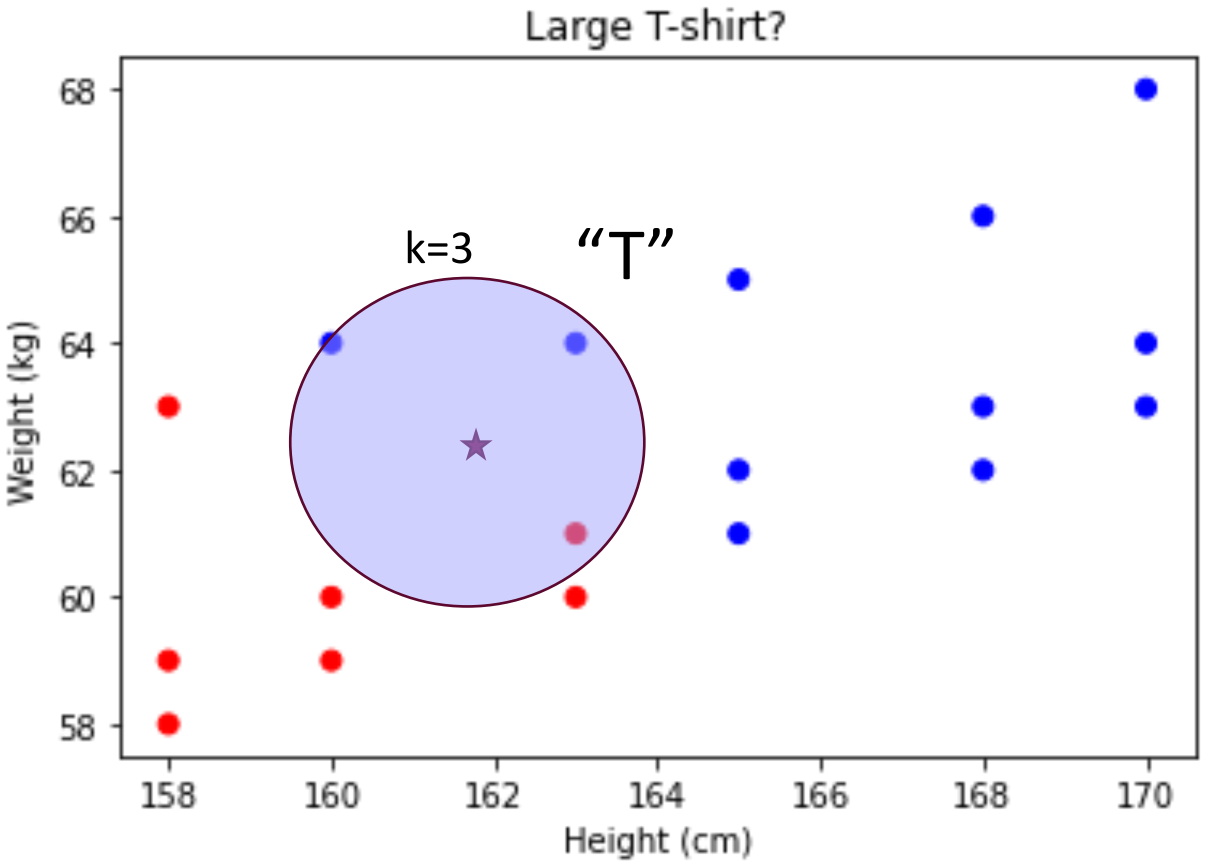
Red: “F” i.e. Medium t-shirt

Height (cm)	Weight (kg)	Large (vs Medium) t-shirt?
158	58	F
158	59	F
158	63	F
160	59	F
160	60	F
163	60	F
163	61	F
160	64	T
163	64	T
165	61	T
165	62	T
165	65	T
168	62	T
168	63	T
168	66	T
170	63	T
170	64	T
170	68	T

x_i y_i

Based on data from https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html

kNN Prediction: What Label?



x_i y_i

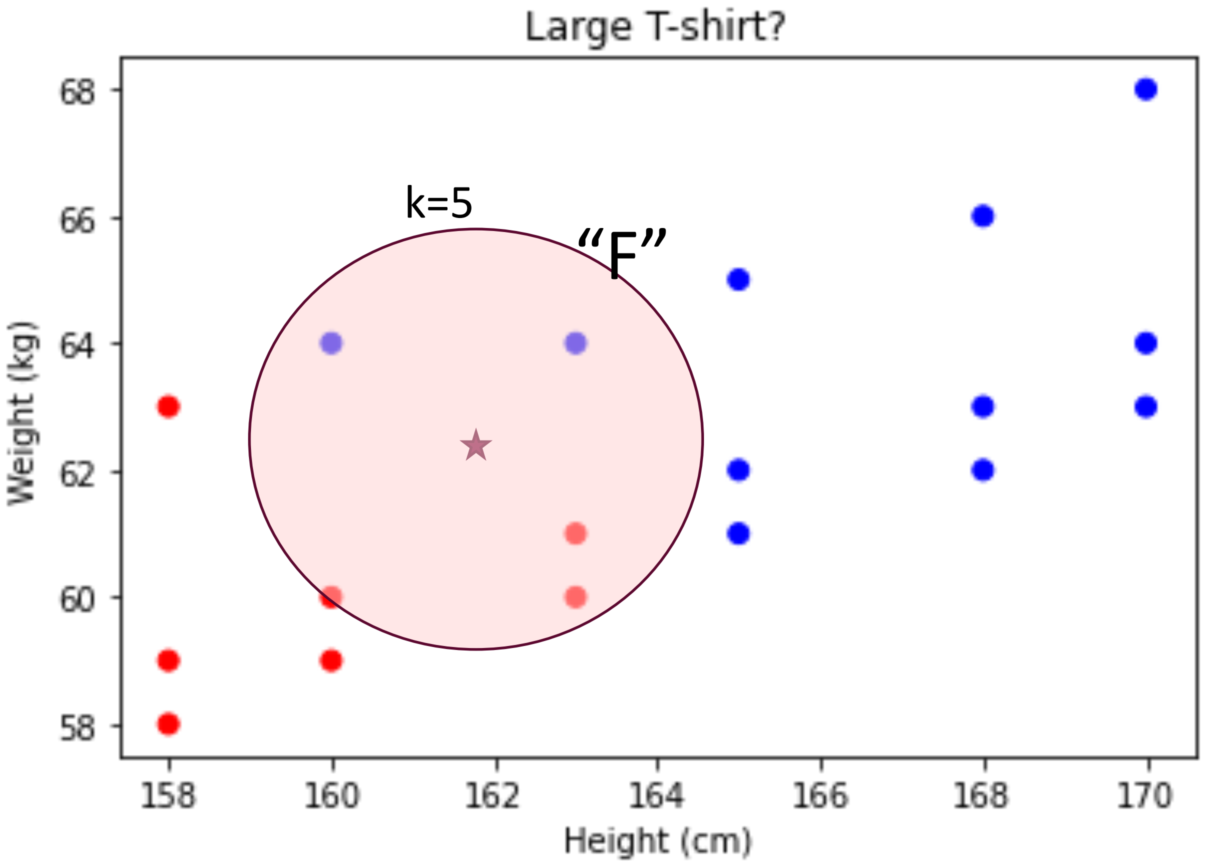
Height (cm)	Weight (kg)	Large (vs Medium) t-shirt?
158	58	F
158	59	F
158	63	F
160	59	F
160	60	F
163	60	F
163	61	F
160	64	T
163	64	T
165	61	T
165	62	T
165	65	T
168	62	T
168	63	T
168	66	T
170	63	T
170	64	T
170	68	T

Blue: "T" i.e. Large t-shirt

Red: "F" i.e. Medium t-shirt

Based on data from https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html

kNN Prediction: What Label?



Blue: "T" i.e. Large t-shirt

Red: "F" i.e. Medium t-shirt

Height (cm)	Weight (kg)	Large (vs Medium) t-shirt?
158	58	F
158	59	F
158	63	F
160	59	F
160	60	F
163	60	F
163	61	F
160	64	T
163	64	T
165	61	T
165	62	T
165	65	T
168	62	T
168	63	T
168	66	T
170	63	T
170	64	T
170	68	T

x_i $\boxed{158 \quad 59}$ $\boxed{F}$ y_i

Based on data from https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html

What Does "Nearest" Mean?

"Nearest neighbors" = training instances with the least "distance".

The choice of "distance function" is critical!

Some commonly used distances $d(\boldsymbol{x_1}, \boldsymbol{x_2})$ are:

$$\left(\sum_j \left|x_{1j} - x_{2j}\right|^1\right)^{\frac{1}{1}}$$

ℓ_1 distance

$$\sum_j \left|x_{1j} - x_{2j}\right|$$

$$\left(\sum_j \left|x_{1j} - x_{2j}\right|^2\right)^{\frac{1}{2}}$$

ℓ_2 distance

Also, "Euclidean" distance

$$\left(\sum_j \left|x_{1j} - x_{2j}\right|^{\to\infty}\right)^{\to 0}$$

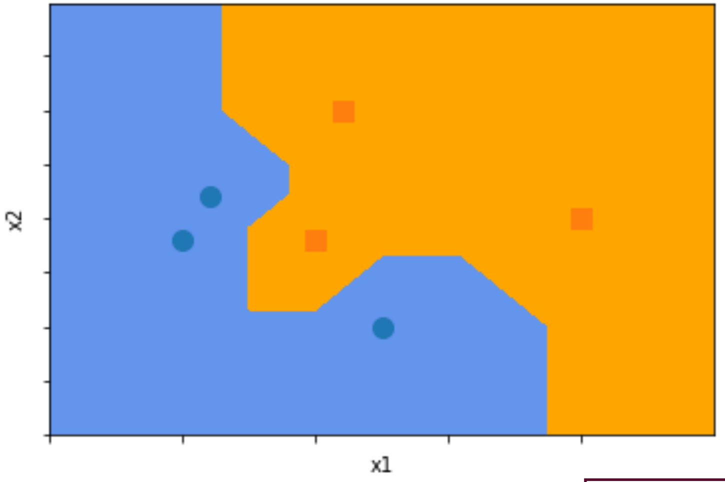
ℓ_∞ distance

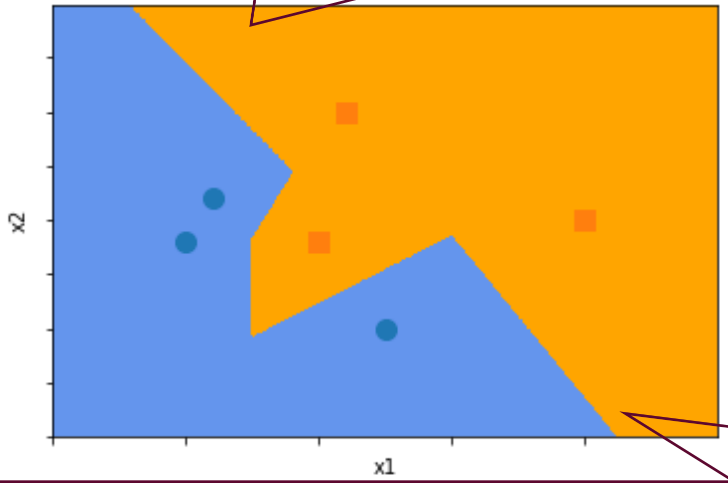
$$\max_j \left(\left|x_{1j} - x_{2j}\right|\right)$$

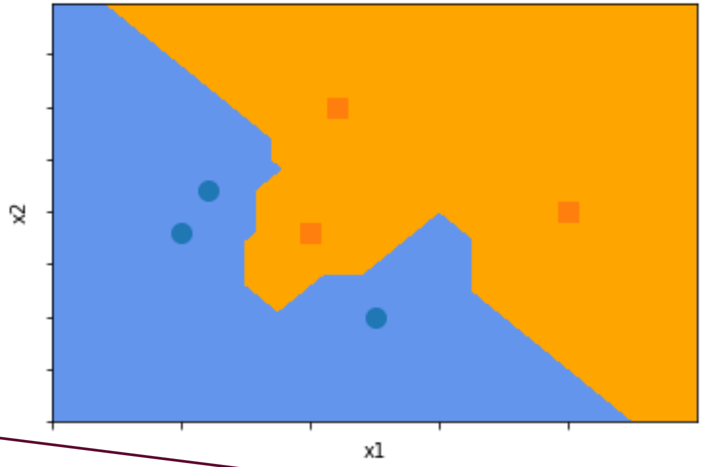
Different distances produce different outcomes

Fix k =1 neighbors

"Decision boundary" plots similar to those we have seen in logistic regression: show what class would be assigned at *every point* $\boldsymbol{x}$







Predictions are usually less reliable when the nearest neighbors are far away …

ℓ_1 distance

$$\sum_j |x_{1j} - x_{2j}|$$

ℓ_2 distance

Also, "Euclidean" distance

ℓ_∞ distance

$$\max_j \left(|x_{1j} - x_{2j}|\right)$$

What Do We Need to Make Predictions

- **Q:** In linear regression / logistic regression / neural networks, what do we need to make predictions?
- **A:** "parameters", or "weights".
  - E.g. for linear regression, we need parameters β so that you can predict $\hat{y} = \beta^T x$

This can be thought of as the definition of "parameters" in ML: they are what we need to make predictions.

$$\boxed{\text{Model class + parameters + new input } \boldsymbol{x} \rightarrow \text{predicted } y}$$

Where Are The "Parameters" in K-NN?

Model class + parameters + new input $\boldsymbol{x} \rightarrow$ predicted y

"kNN classifier" ??

A: The full training dataset!

Funnily, methods like these where the parameters are either the training data itself, or instead grow in size "automatically" with the training data, are called "**<u>non-parametric</u>**" machine learning approaches.

When Is The Training Phase in kNN?

There is no explicit “training” phase!*

The moment we have the dataset, we are ready to produce predictions for new input data!

* caveat: some “approximate nearest neighbors” involve a dataset preprocessing phase that may be thought of as training.

Where Are The Hyperparameters in KNN?

- Choice of distance function
  - Most often an $\ell_{p=2\ or\ 1\ or\ \infty}$ distance
  - Sometimes an ℓ_p distance after transforming inputs x to some $f(x)$: a little bit like basis feature expansion or standardization, more on this later

- Choice of k, the number of nearest neighbors
  - Small values easily affected by noisy data.
  - Large values make it difficult to model sharp changes in the true function.
  - For binary classification, usually an odd number to avoid ties.

What Is The Hypothesis Space in K-NN?

Q: What functions can k-NN classifiers / regressors represent? Or perhaps easier to think about: what functions can k-NN models *not* represent?

A: k-NN models are only limited in expressivity by the training data, so with the right training dataset, k-NN models can represent *any* function.

Harder Question (Dinesh's office hours question)

Exercise:

You are given that $k = 1$ and distance function ℓ_2 for a binary kNN classifier.

You are also given a training dataset of samples $\{x_i\}_{i=1}^{N}$ without their corresponding binary classification labels $\{y_i\}_{i=1}^{N}$.

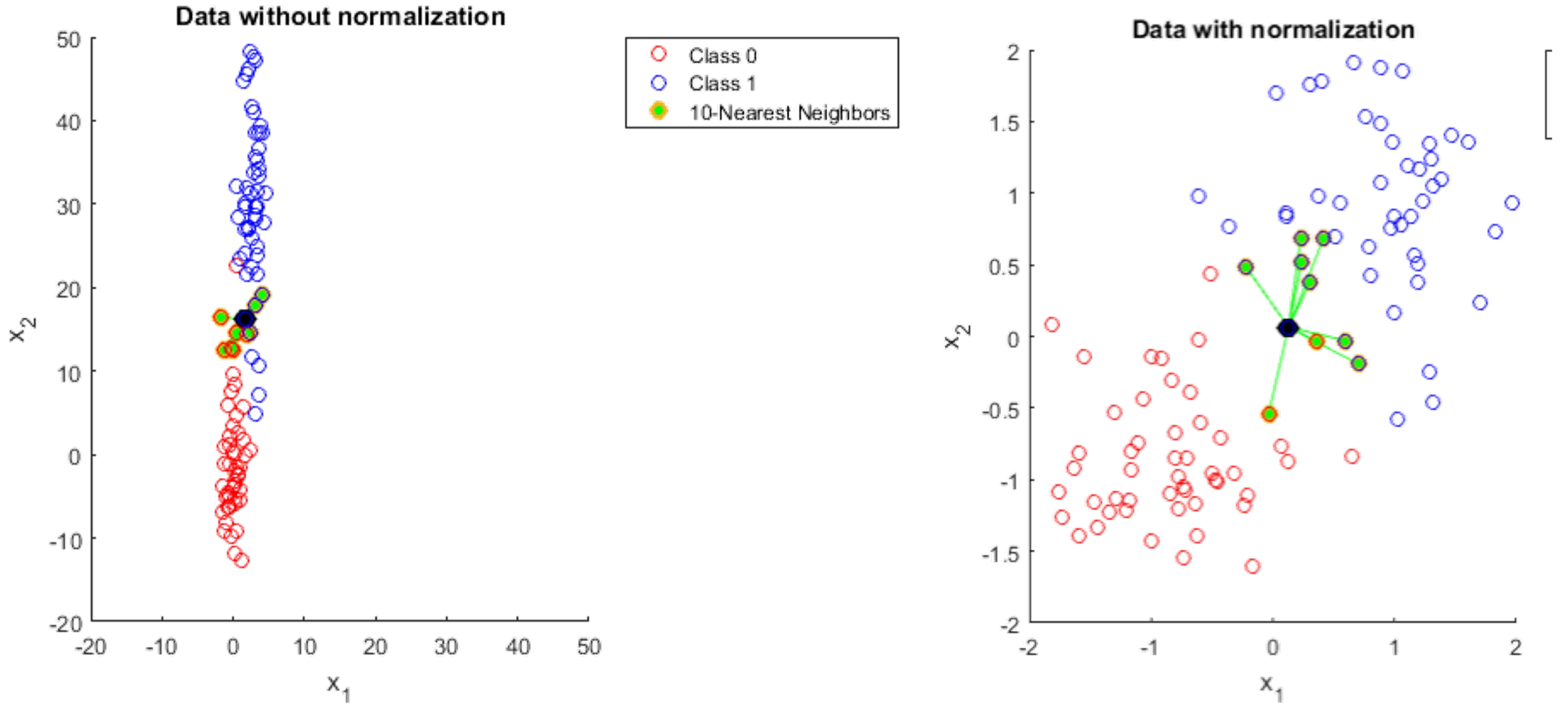
What is the space of all functions that your kNN classifier might eventually produce?

An Excellent First Algorithm To Try

- Recall that an important step in model evaluation is comparing to a "simple baseline."
- For many problems, k-Nearest Neighbors is a good choice of a first "simple baseline".
  - Very easy to write in code.
  - Versatile, does not impose specific restrictions on the learned function, such as "linearity".
  - Very easy to interpret the outcomes because of the direct connection to training data.
- Often works surprisingly well!
- kNN is not without its problems, of course. But more on that later.

Aside: Scale Invariance in kNN

- kNN approaches are not inherently invariant to feature scaling.
  - E.g. if distance measure is L_2, and one feature in the data is scaled 100x, it suddenly plays a much bigger role than before in determining what neighbors are "nearest".
- Same solution works as before: feature standardization / "normalization".



https://stats.stackexchange.com/questions/287425/why-do-you-need-to-scale-data-in-knn

Aside: kNN Distance Functions for String Data Types

Hamming distance (number of characters that are different)

ABCDE vs AGDDF → 3

Edit distance (number of character inserts/replacements/deletes to go from one to the other)

ROBOT vs BOT → 2

Jaccard distance between sets $\frac{|A \cap B|}{|A \cup B|}$

between **n-grams** (n-character substrings of the strings, with (n-1) character padding)

$$SSROBOTSS vs SSBOTSS \rightarrow \overbrace{|\{BOT,OT\$,T\$\$\}|}^{3} / \overbrace{|\{\$\$R,\$RO,ROB,OBO,\$\$B,\$BO,BOT,OT\$,T\$\$\}|}^{9}$$

Aside: Probabilistic Predictions From kNN Classifiers

- Easy to extend to produce probabilistic predictions too.
- One example: for a multi-class classification problem:
  - Find k nearest neighbors
  - Set P(class i) = $1/k$ * number of instances of class i among the neighbors.

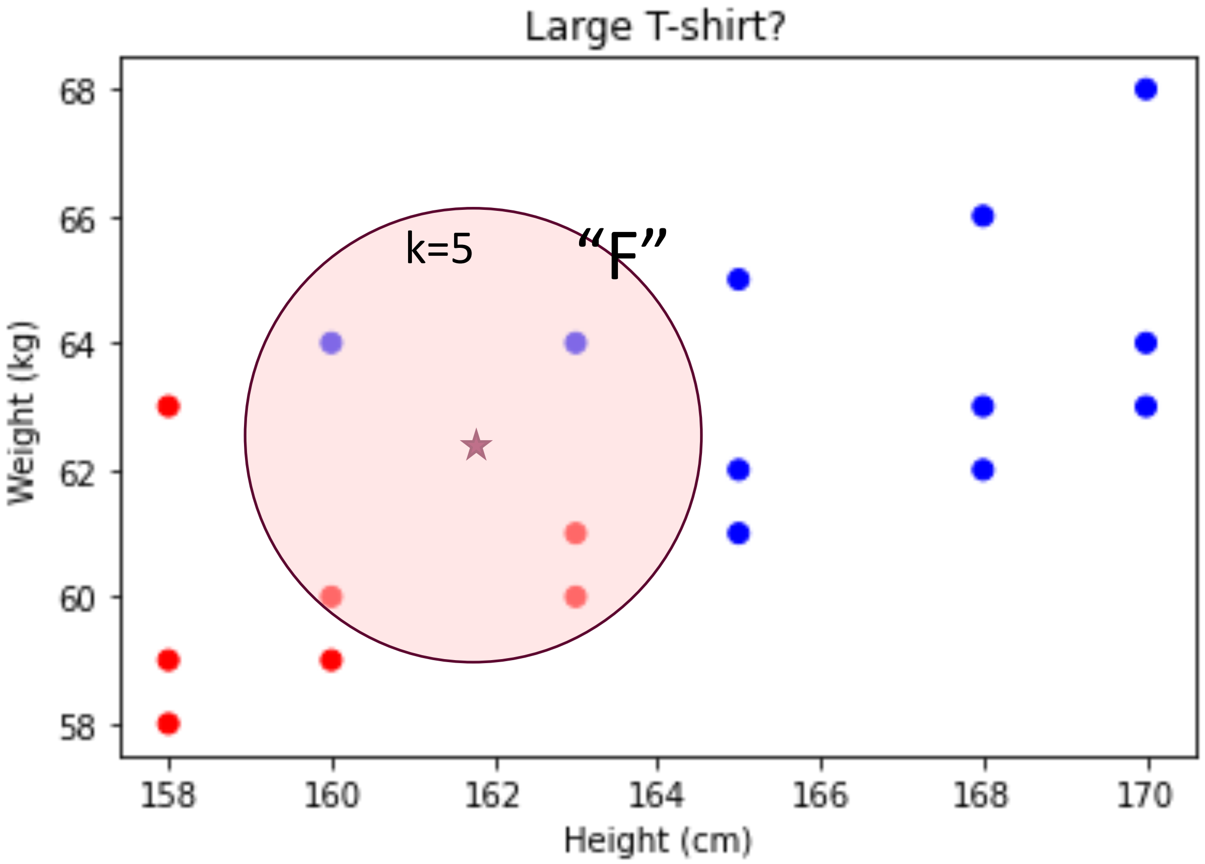
- More sophisticated approaches are possible, e.g., by sorting the k neighbors by distance, and assigning most importance to the closest neighbors.

Summary So Far: K-Nearest Neighbors

kNN Classification: To predict category label y of a new point $\boldsymbol{x}$:

Find k nearest neighbors

Assign the majority label



- Easy to implement
- Versatile in terms of modeling many functions
- Interpretable in terms of data

Based on data from https://www.listendata.com/2017/12/k-nearest-neighbor-step-by-step-tutorial.html

Scaling Issues with kNNs

- Irrelevant features: distances become unreliable.
- Too many features: “curse of dimensionality”
- Large datasets (high N or D): computationally inefficient to make predictions!

Problem 1: Irrelevant Features

- Let's say we want to predict y = t-shirt size for a person.
- What if my input features are:
  - x_1 = height
  - x_2 = weight
  - x_3 = hair length
  - x_4 = age
  - x_5 = body temperature
  - x_6 = what they ate for breakfast this morning

  …

Common distance functions (such as ℓ_2) value all input features equally.

As you add more irrelevant variables, distances get dominated by those irrelevant dimensions in $\boldsymbol{x}$.

i.e., your kNN model might make decisions about t-shirt size more based on hair length, age, breakfast than on the height and weight!

Problem 2: "Curse of Dimensionality"

- Adding more dimensions makes lots of things weird and counterintuitive
  - For example, the percentage of the volume of a D-dimensional sphere with radius r, that lies beyond $\ell_2$distance $0.99r$ from the center is:
    - 3% at $D = 3$
    - 63% at $D = 100$
    - 99.99% at $D = 1000$

- Specifically for k-NN, the space is now so large that all points in any finite dataset are likely to be very far apart.
  - "Closest points" are almost as far away as the farthest away points. When "nearest neighbors" are far away, predictions are poor.

Problem 3: Computationally Expensive

- High N, D also makes it computationally expensive to compute neighbors.

- Naively, must compute N distances between D-dimensional data pairs to compute neighbors before classifying a single new point.

- $O(ND)$ for each new sample

Scaling kNN to high D and N? An Overview

Beyond our scope, but a quick overview:

Indexing

- Use kd-trees and other multidimensional indices to capture the training data. Each lookup operation (finding nearest nbrs) is O(log n) rather than O(n)

Parallelism (e.g., PANDA, LBL)

- Use multiple cores / processors, and either compare against in-memory data or kd trees

Approximation

- https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbor-algorithms
- Libraries like FLANN: "Fast Library for Approximate Nearest Neighbors"
- For example, subsample the training dataset cleverly so that kNN mostly returns the same outputs
- See, e.g., https://www.kaggle.com/code/pawanbhandarkar/knn-vs-approximate-knn-what-s-the-difference/notebook

Still Commonly Used In Practice!

- Often in concern with other methods such as neural nets:
  - E.g. you can train a 4-layer neural network to classify your data, then use the third layer activations $f(x)$ as the inputs to your k-NN classifier.
    - Could mitigate scaling issues, because the activations $f(x)$ could be much smaller-dimensionality than the inputs x.

  - Advantages: Interpretability, and sometimes even better performance! Some references:
    - Sridhar et al, “Memory-Consistent Neural Networks for Imitation Learning”, ICLR 2024.
    - Pari et al, “The Surprising Effectiveness of Representation Learning for Visual Imitation”, CORL 2021.

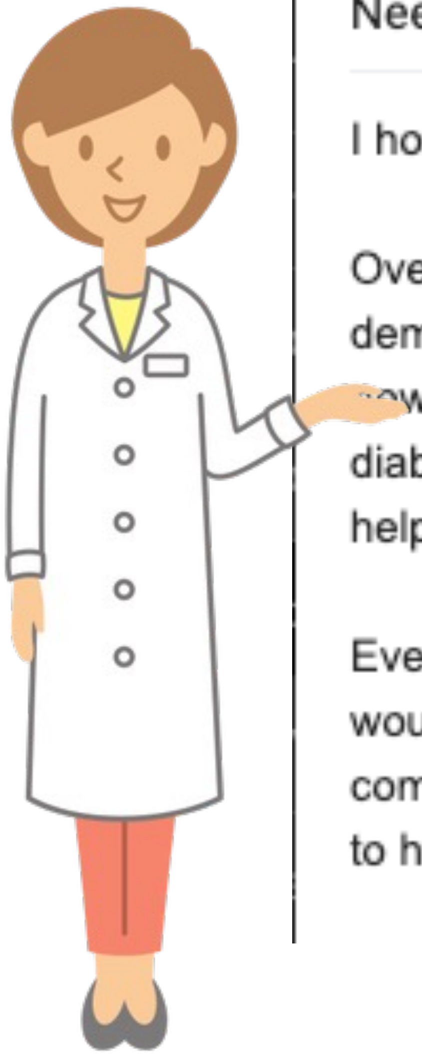
KNNs summary

- A simple and versatile ML approach, tied directly to the data.
- No training phase. Ready to make predictions the moment you have the dataset.
- “Non-parametric”. For KNNs, the data *are* the parameters.
- Scaling troubles, but still almost always worthwhile as your first algorithm for a new problem.

End of K-NN

CIS 4190/5190: Lec 09 Mon Sep 30, 2024. Part 2.

Decision Trees



Need help modeling diabetes risks!

I hope you are doing well in these weird times.

Over the years, I've collected data from lots of patients, recording their physical information, their demographic information, habits, and done their lab work to diagnose diabetes. I'm wondering now: from all this data, could I model the risk of other people with similar characteristics having diabetes given all this other information about them? And would your applied ML class be able to help? I've attached the data here for you to take a look.

Eventually, we'll want to explain our findings to patients, and point out any behavioral changes that would mitigate their risk for diabetes. Even if the risk factors we find are non-modifiable, insurance companies would be interested in understanding and estimating this risk. Either way, it'd be great to have something that we can understand and interpret well!

Diabetes Data

data matrix X

labels

ID	AGE RIDAGEYR	WAIST	HEIGHT BMX	CHOLESTEROL	UPPER LEG LENGTH MXLEG	WEIGHT	BMI BMXBMI	RACE	HIGH BP BPQ	ALCOHOL USE	EDUCATION DMDEDUC2	GENDER	FAMILY INCOME RATIO INDFM	GLYCOHAEMOGLOBIN	TIC
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

sample x_i

y_i

Data from NHANES 2013/14 survey

Diabetes Data

ID	AGE	WAIST	HEIGHT	CHOLESTEROL	UPPER LEG LENGTH	WEIGHT	BMI	RACE	HIGH BP	ALCOHOL USE	EDUCATION	GENDER	FAMILY INCOME RATIO	GLYCOHAEMOGLOBIN	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								yes	2.0	college graduate or above	female	5.0	5.5	no
73566	56.0								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		...c 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-Hispa...	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	[illegible]	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								...Multi-Racial	no	0.0	college graduate or above	male	5.0	5.0	no
73585								...Multi-Racial	no	4.0	some college or AA degree	male	2.26	5.0	no
73589								...panic White	no	2.0	high school graduate / GED	male	1.74	5.5	no
73595								...spanic	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

Columns X_j denote features

Patient number: should this really be a feature?

Data from NHANES 2013/14 survey

Feature Types

numeric | nominal | ordinal | binary

ID	AGE (RIDAGEYR)	WAIST	HEIGHT	CHOLESTEROL	UPPER LEG LENGTH (MXLEG)	WEIGHT	BMI (BMXBMI)	RACE	HIGH BP (BPQ0)	ALCOHOL USE	EDUCATION (DMDEDUC2)	GENDER	FAMILY INCOME RATIO (INDFM)	GLYCOHAEMOGLOBIN	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	[illegible]	[illegible]	[illegible]	35.3	[illegible]	[illegible]	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	[illegible]	[illegible]	[illegible]	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

This column seems binary, but also has “refused to answer” and “don’t know” categories

Data from NHANES 2013/14 survey

Data Dictionary

- Data sets are often accompanied by a data dictionary that describes each feature
- It is critical to understand the data!
- The dictionary for our data: https://wwwn.cdc.gov/nchs/nhanes/Default.aspx

ID (SEQN)	AGE (RIDAGEYR)	WAIST_CIRCUM (BMXWAIST)	HEIGHT (BMXHT)	CHOLESTEROL (LBXTC)	UPPER_LEG_LEN (BMXLEG)	WEIGHT (BMXWT)	BMI (BMXBMI)	RACE (RIDRETH1)	HIGH_BP (BPQ020)	ALCOHOL_USE (ALQ120Q)	EDUCATION (DMDEDUC2)	GENDER (RIAGENDR)	FAMILY_INCOME_RATIO (INDFMPIR)	GLYCOHEMOGLOBIN (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									2.0	college graduate or above	female	5.0	5.5	no
73566	56.0									1.0	high school graduate / GED	female	0.48	5.4	no
73567	65.0									4.0	9th-11th grade	male	1.2	5.2	no
73568	26.0									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	[illegible]	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

777 = refused; 999 = don't know

Data from NHANES 2013/14 survey

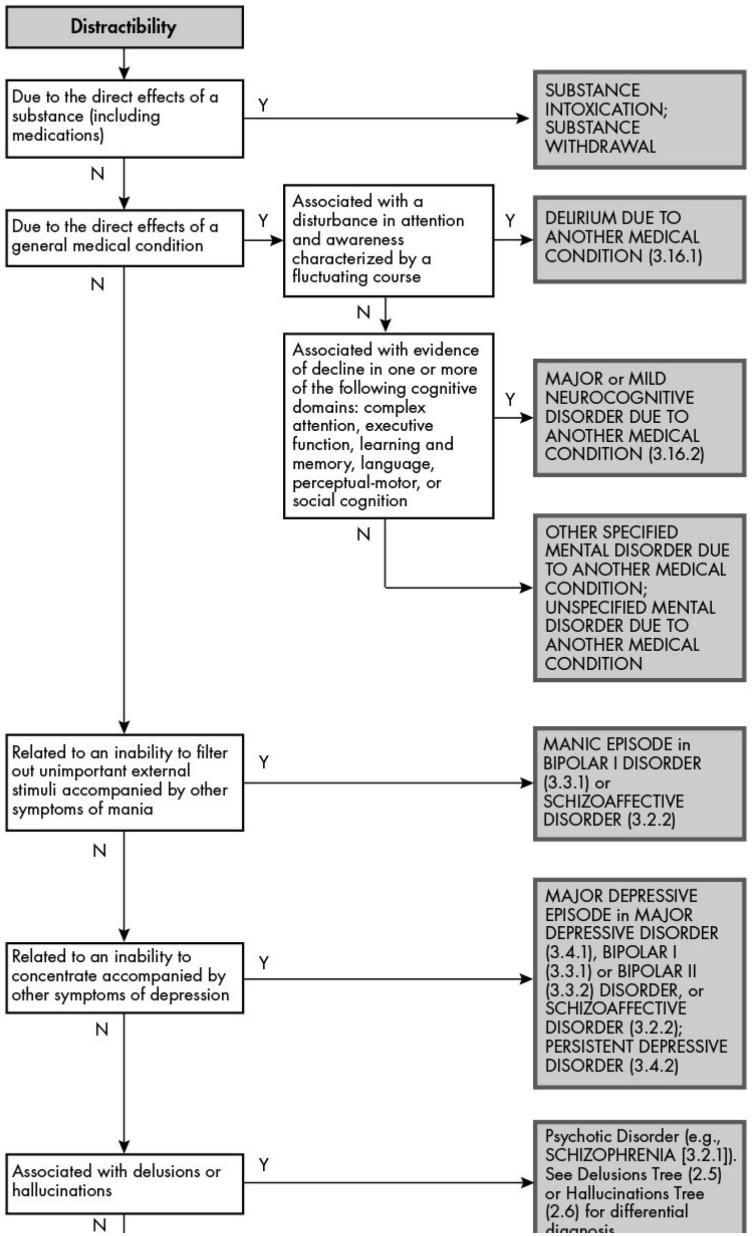
A First Look At Decision Trees
(Outside the Context of ML)

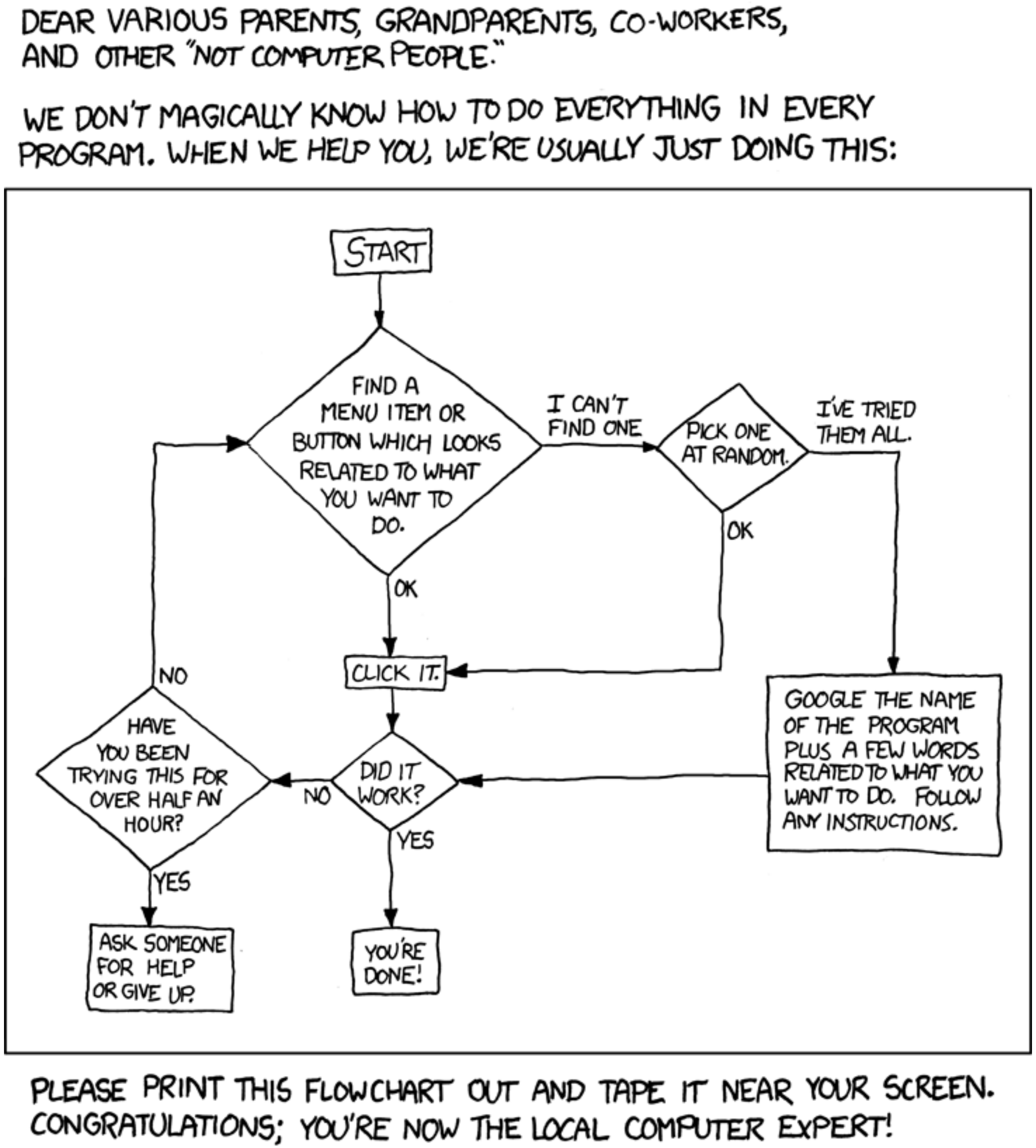
Decision Trees for People

APA DSM Library

How do we train a human to make a diagnosis?

- Often, a kind of flowchart based on tests! "Decision Tree"
  - e.g., how we train psychiatrists to make diagnoses? →
- "Explainable" in a clear way, easy to evaluate




DEAR VARIOUS PARENTS, GRANDPARENTS, CO-WORKERS, AND OTHER "NOT COMPUTER PEOPLE."
WE DON'T MAGICALLY KNOW HOW TO DO EVERYTHING IN EVERY PROGRAM. WHEN WE HELP YOU, WE'RE USUALLY JUST DOING THIS:
START
FIND A MENU ITEM OR BUTTON WHICH LOOKS RELATED TO WHAT YOU WANT TO DO.
I CAN'T FIND ONE
PICK ONE AT RANDOM.
I'VE TRIED THEM ALL.
OK
OK
CLICK IT.
GOOGLE THE NAME OF THE PROGRAM PLUS A FEW WORDS RELATED TO WHAT YOU WANT TO DO. FOLLOW ANY INSTRUCTIONS.
NO
HAVE YOU BEEN TRYING THIS FOR OVER HALF AN HOUR?
NO
DID IT WORK?
YES
YES
ASK SOMEONE FOR HELP OR GIVE UP.
YOU'RE DONE!
PLEASE PRINT THIS FLOWCHART OUT AND TAPE IT NEAR YOUR SCREEN. CONGRATULATIONS; YOU'RE NOW THE LOCAL COMPUTER EXPERT!

Credit: xkcd

Idea: We could create decision trees by looking at example input->output pairs i.e. learning!

But first, let's formalize what we mean by a decision tree…

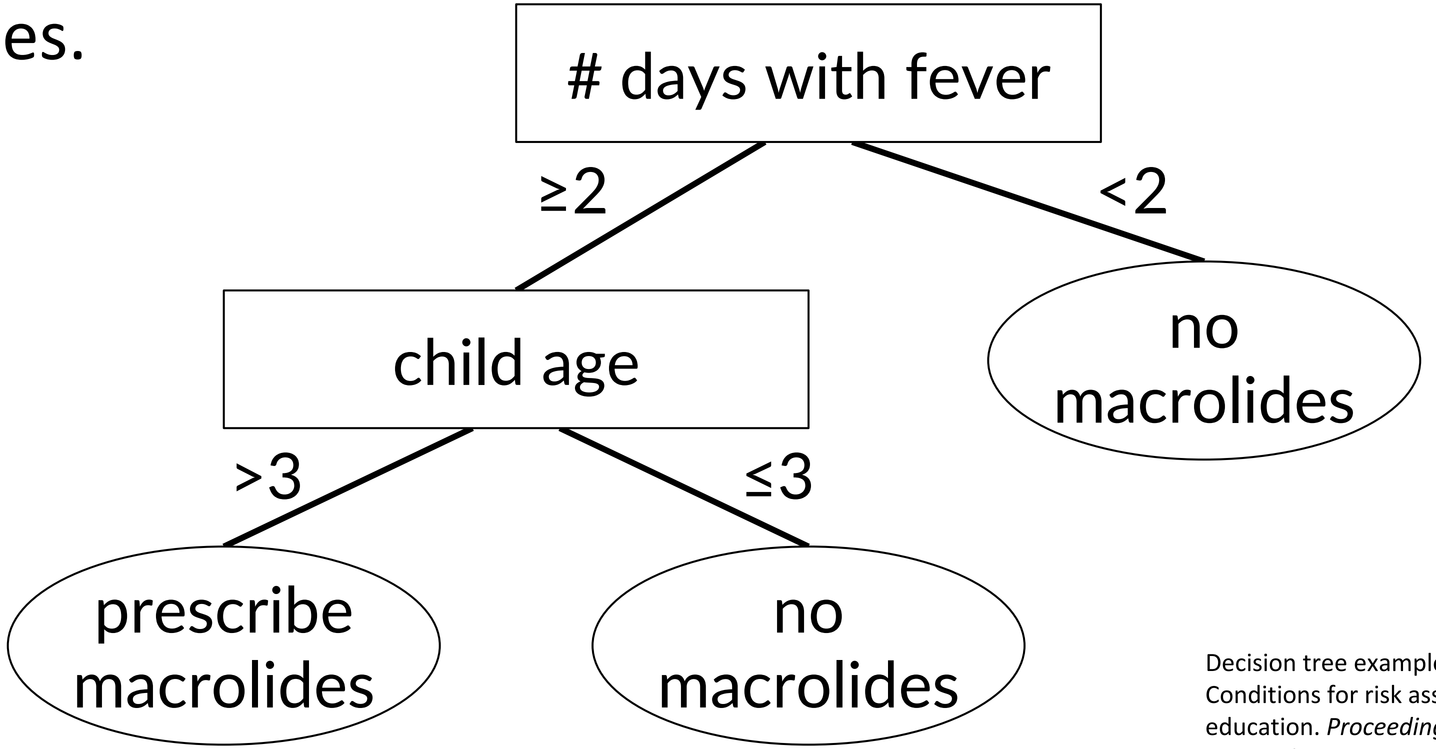
A Decision Tree Based on Boolean Tests

For continuous features, we'll restrict our study to internal nodes that make binary decisions* based on a single feature:

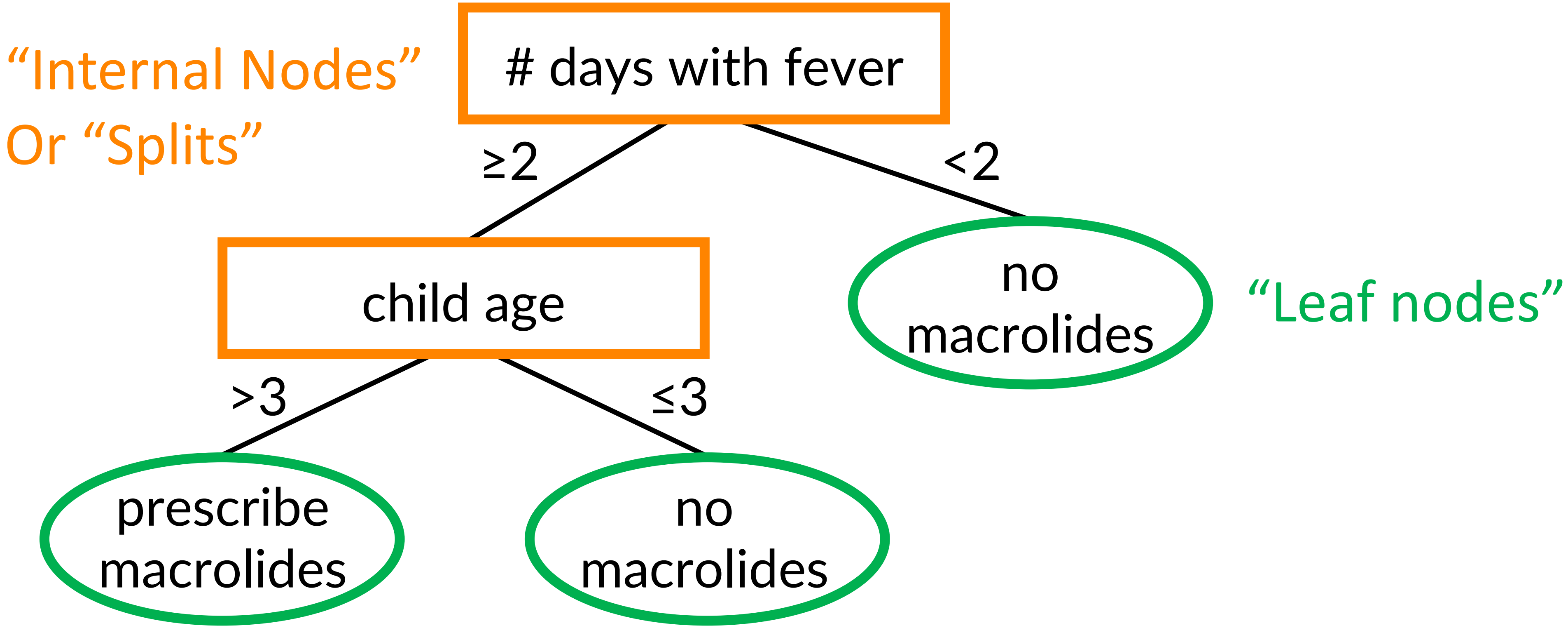
- e.g. is a real-valued feature above or below some threshold?

- e.g. is a binary-valued feature true or false?

* for discrete-valued features we will usually create as many splits as the number of values.



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).


"Internal Nodes"
Or "Splits"
days with fever
≥2
<2
child age
no macrolides
"Leaf nodes"
>3
≤3
prescribe macrolides
no macrolides

Each Internal Tree Node "Splits" Training Data

(an internal node from a decision tree to classify horse breeds)

ColorOfCoat	TypeOfHorse
black	thoroughbred
bay	Arabian
black	thoroughbred
chestnut	quarter
black	Arabian

N=5; 3 classes

ColorOfCoat == 'black'

ColorOfCoat	TypeOfHorse
black	thoroughbred
black	thoroughbred
black	Arabian

N=3; 2 classes

ColorOfCoat	TypeOfHorse
bay	Arabian
chestnut	quarter

N=2; 2 classes

Each Leaf Node Behaves Like a K-NN Neighborhood

(leaf nodes from a decision tree to classify horse breeds)

TypeOfHorse
thoroughbred
thoroughbred
Arabian

N=3; 2 classes

"classify as thoroughbred"

TypeOfHorse
Arabian
quarter

N=2; 2 classes

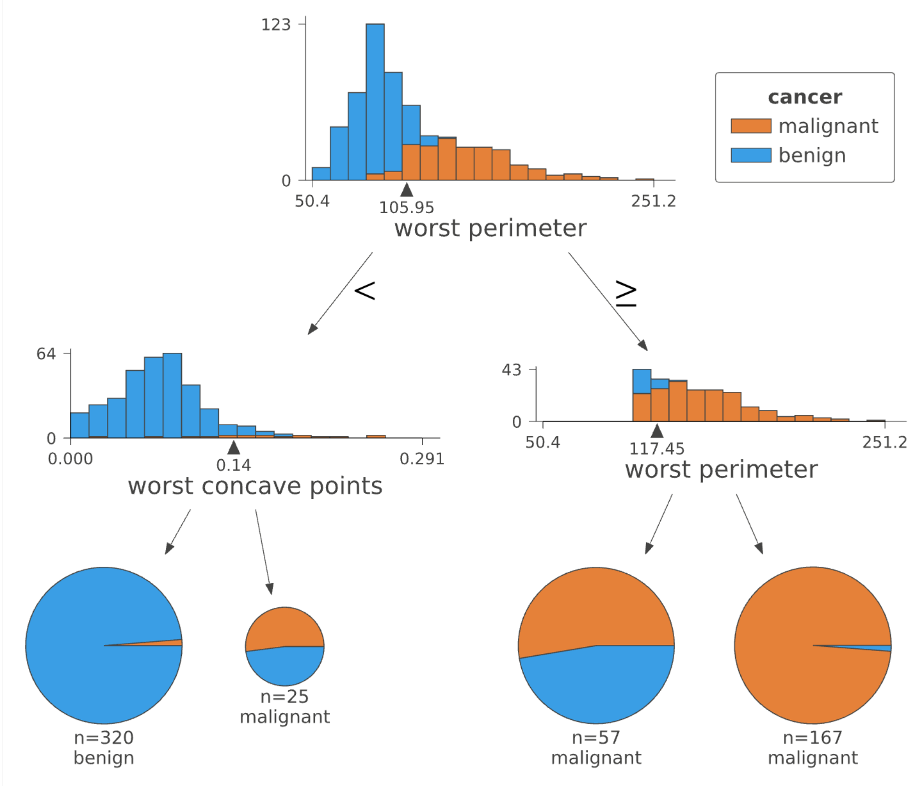
"classify as Arabian / quarter"

Representing Decision Trees

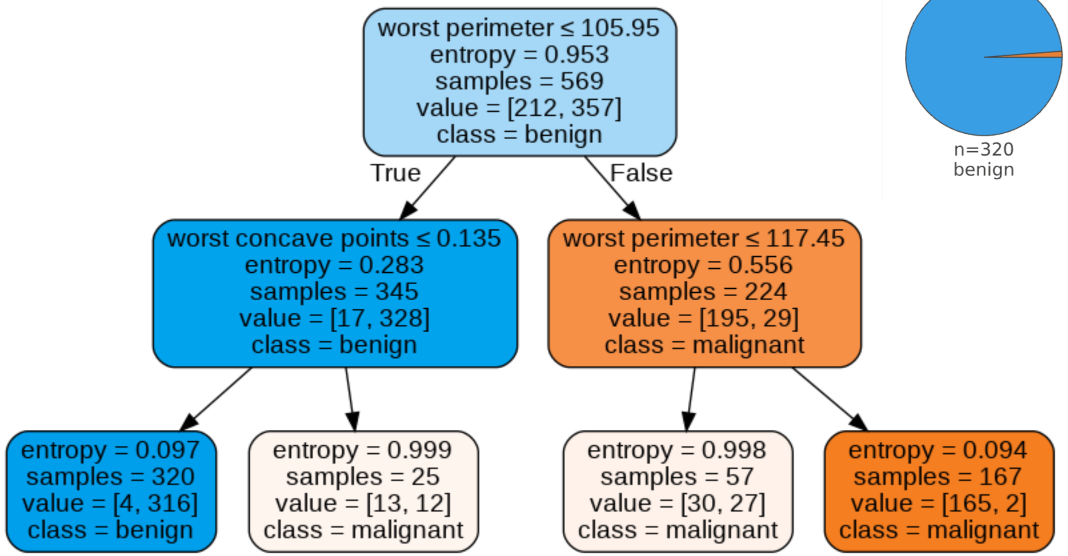
sklearn text

```
|--- worst perimeter <= 105.95
|   |--- worst concave points <= 0.135
|   |   |--- class: benign
|   |--- worst concave points > 0.135
|   |   |--- class: malignant
|--- worst perimeter > 105.95
|   |--- worst perimeter <= 117.45
|   |   |--- class: malignant
|   |--- worst perimeter > 117.45
|   |   |--- class: malignant
```

dtreeviz



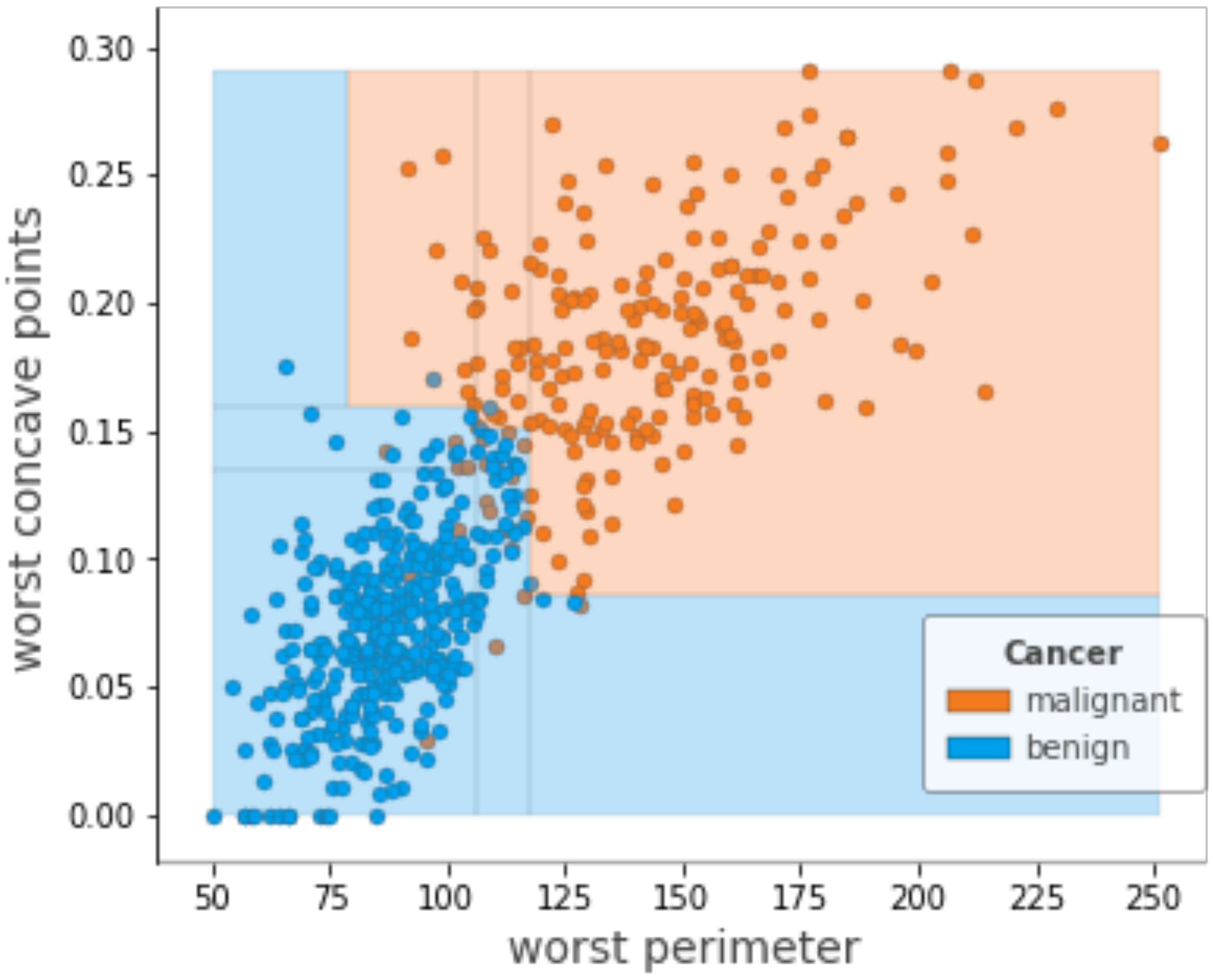
sklearn graphviz



Decisions trees generated on Wisconsin Breast Cancer dataset in sklearn

Decision Tree – Induced Partition

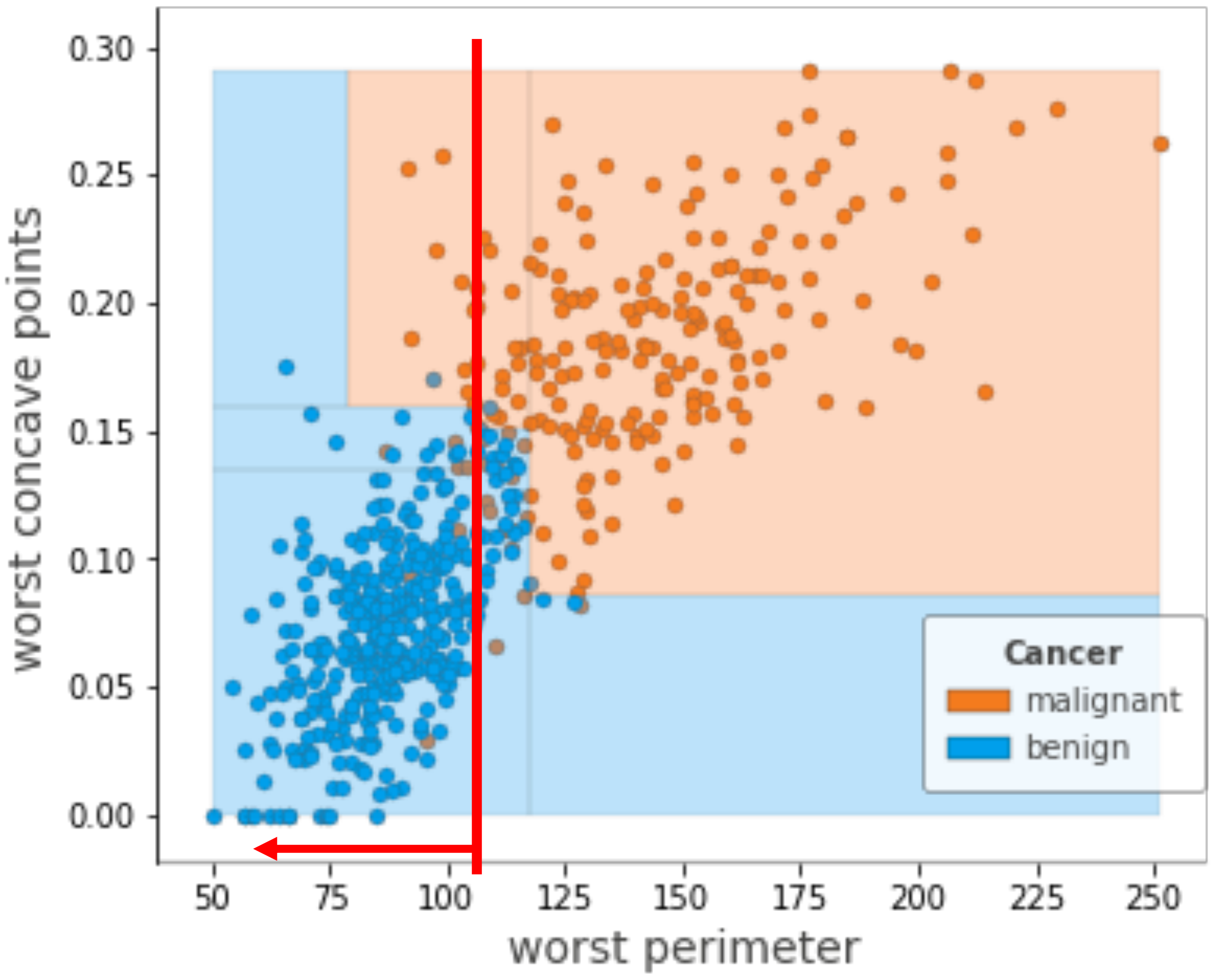
```
|--- worst perimeter <= 105.95
| |--- worst concave points <= 0.135
| | |--- class: benign
| |--- worst concave points > 0.135
| | |--- worst concave points < 0.16
| | | |--- class: benign
| | |--- worst concave points > 0.16
| | | | --- worst perimeter > 80
| | | | | --- class: malignant
| | | |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



Decisions trees generated on Wisconsin Breast Cancer dataset in sklearn

Decision Tree – Induced Partition

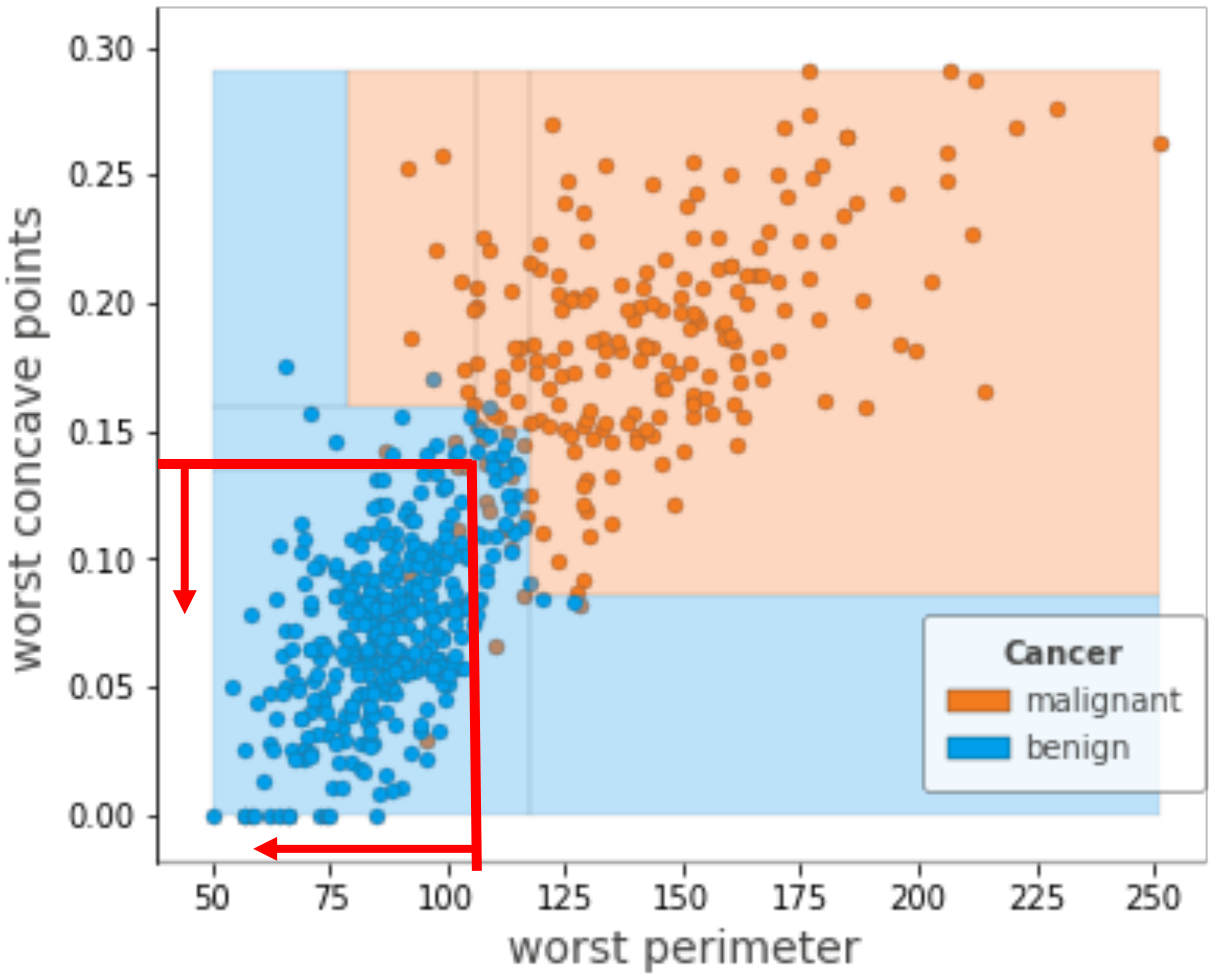
```
|--- worst perimeter <= 105.95
|  |--- worst concave points <= 0.135
|  |  |--- class: benign
|  |--- worst concave points > 0.135
|  |  |--- worst concave points < 0.16
|  |  |  |--- class: benign
|  |  |--- worst concave points > 0.16
|  |  |  | --- worst perimeter > 80
|  |  |  |  | --- class: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



Decisions trees generated on Wisconsin Breast Cancer dataset in sklearn

Decision Tree – Induced Partition

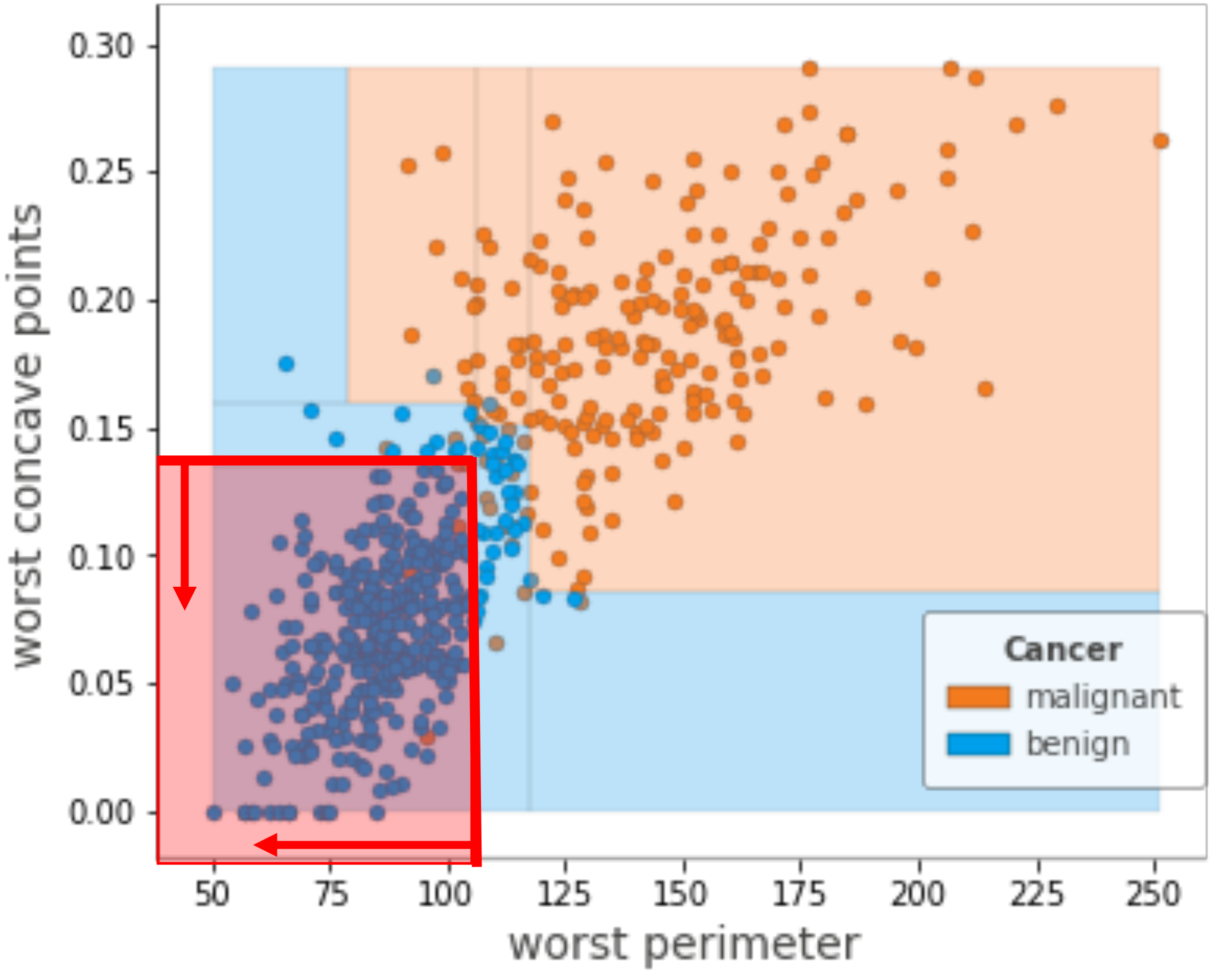
```
|--- worst perimeter <= 105.95
|  |--- worst concave points <= 0.135
|  |  |--- class: benign
|  |--- worst concave points > 0.135
|  |  |--- worst concave points < 0.16
|  |  |  |--- class: benign
|  |  |--- worst concave points > 0.16
|  |  |  | --- worst perimeter > 80
|  |  |  |  | --- class: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



Decisions trees generated on Wisconsin Breast Cancer dataset in sklearn

Decision Tree – Induced Partition

```
|--- 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
                    …
                    …
```



So let's ask our usual question: what is the hypothesis class expressed by a DT?

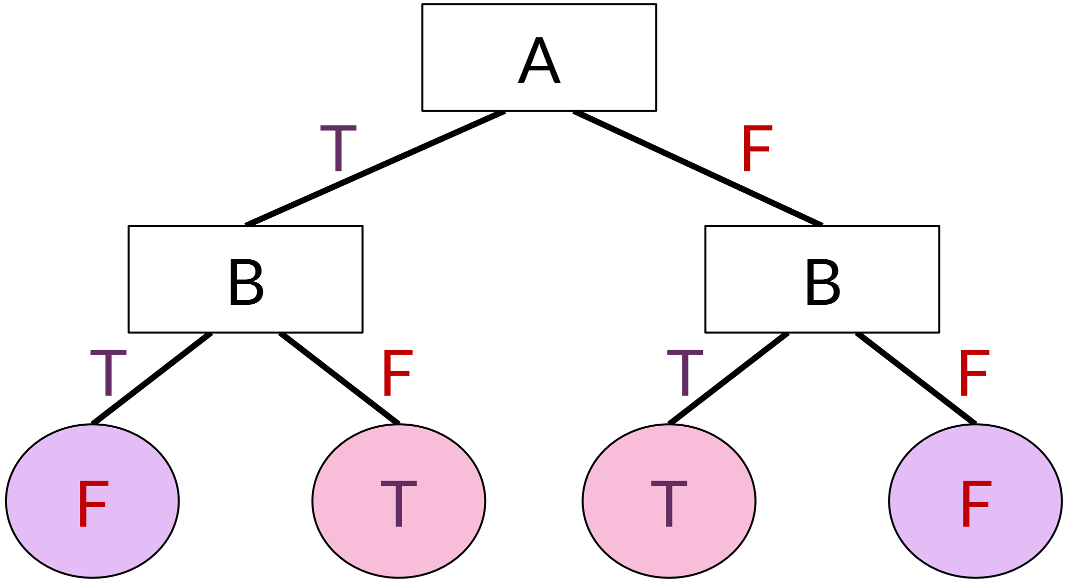
Decision trees divide the feature space into axis-aligned "hyperrectangles"

Decision Trees with Boolean Variables

Decision Trees and Boolean Functions

- Decision trees can represent any Boolean function of the features

A	B	A xor B
T	T	F
T	F	T
F	T	T
F	F	F

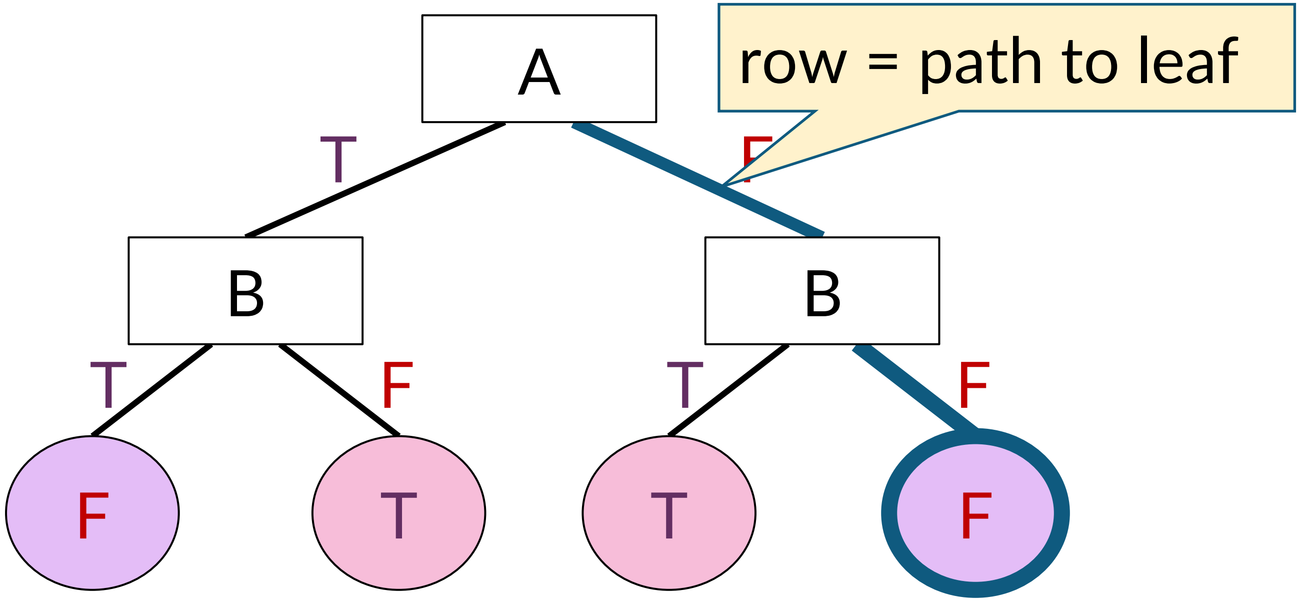


- In the worst case, the tree will require exponentially many nodes

Decision Trees and Boolean Functions

- Decision trees can represent any boolean function of the features

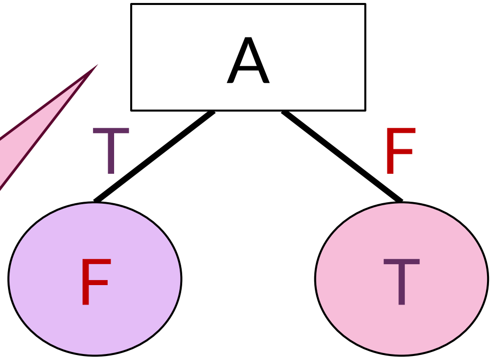
A	B	A xor B
T	T	F
T	F	T
F	T	T
F	F	F

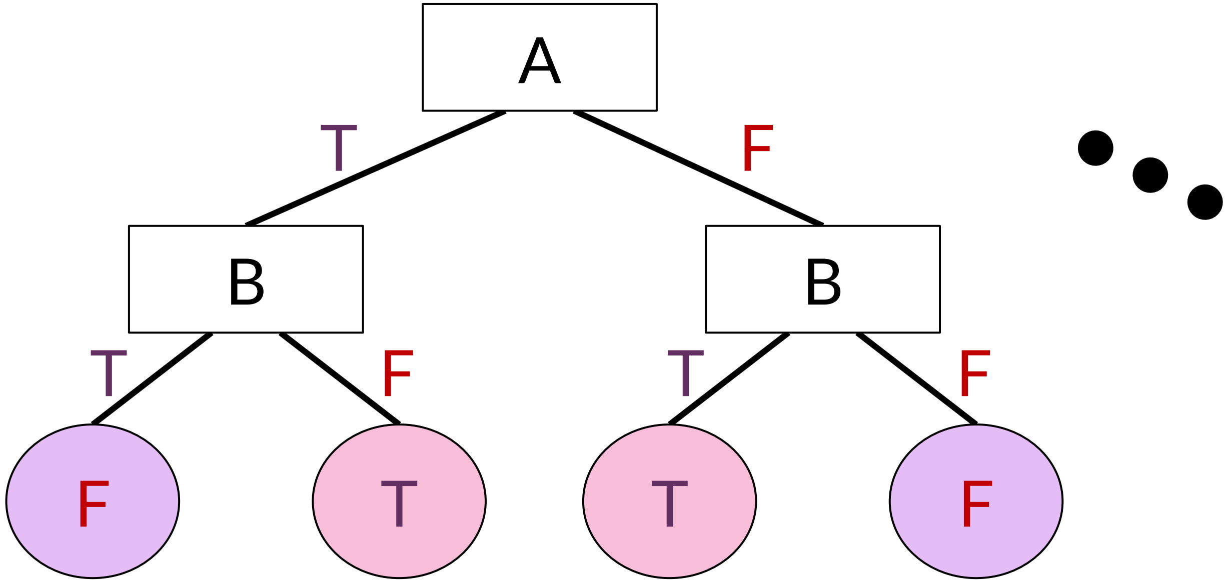


Decision Trees and Boolean Functions

- DTs have a variable-sized hypothesis space based on their depth
  - Depth 1: any boolean function based on one feature
  - Depth 2: any boolean function based on two features
  - . . .

DTs of depth 1 are also called decision stumps



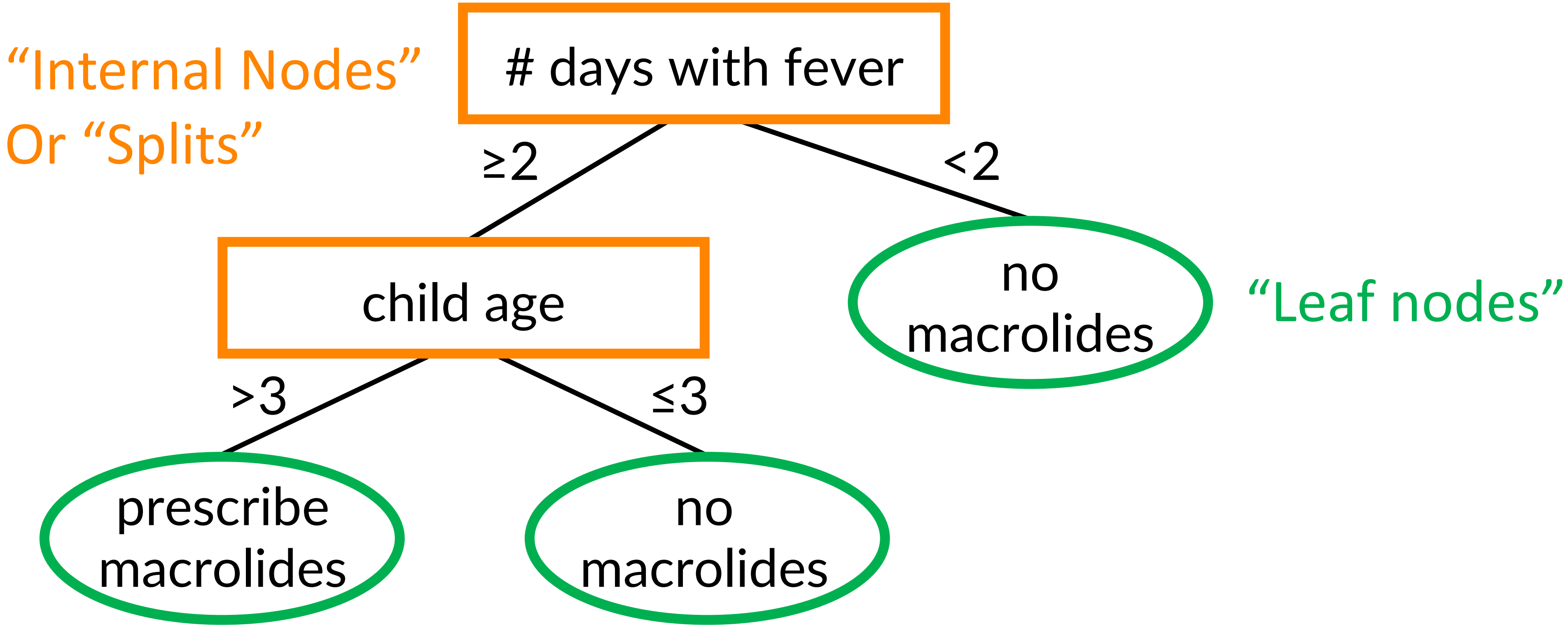


Announcements

CIS 4190/5190: Lec 10 Wed Oct 2, 2024.

Decision Trees (part 2 / 2)

Recap: Decision Trees



Training Decision Trees

Decision Tree Classifier = "20-Questions"

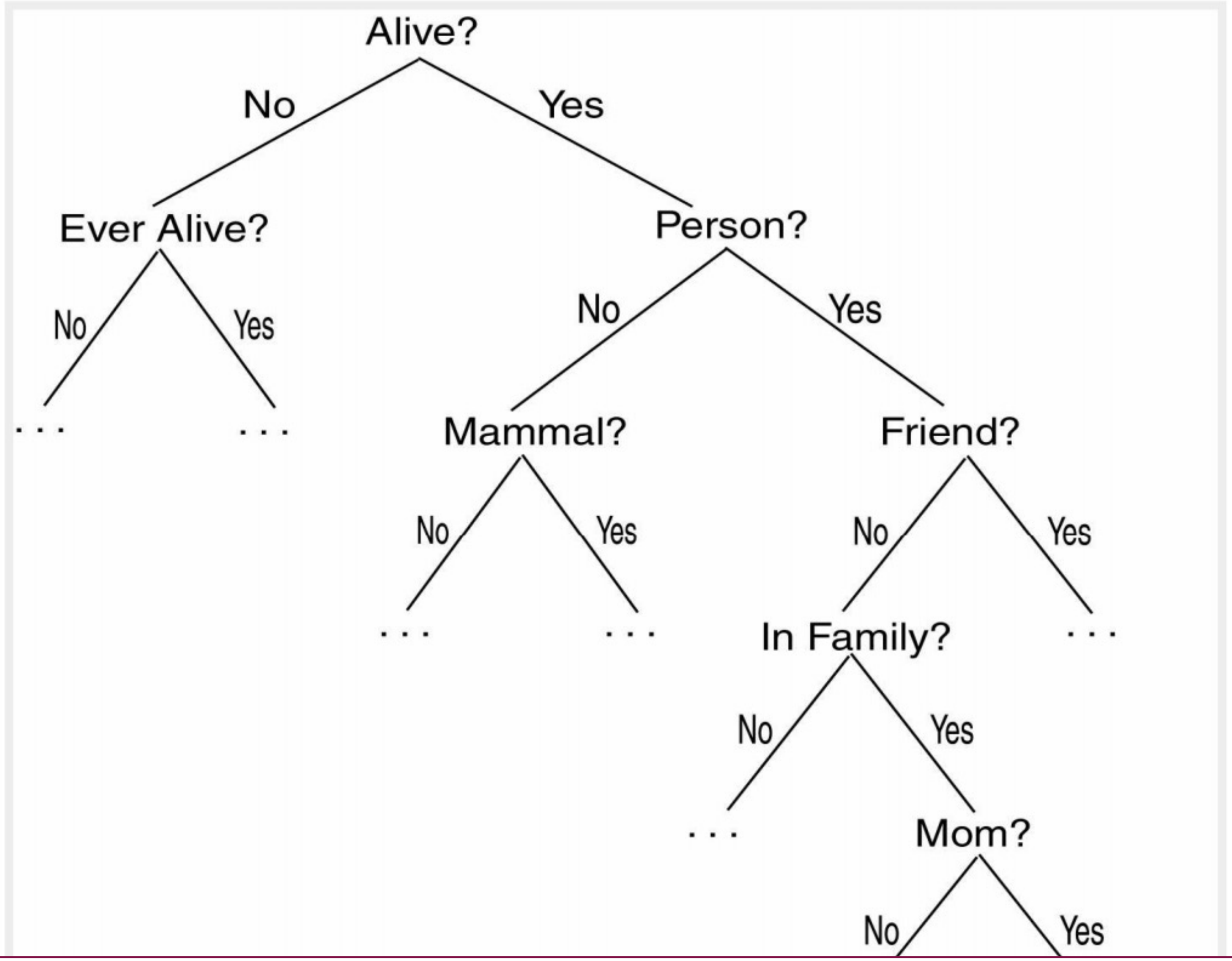
Alice has an object / person in mind

Bob can ask her up to 20 yes/no questions, must guess as quickly as possible

Questions ≈ Decision Tree nodes

Number of questions ≈ depth of tree

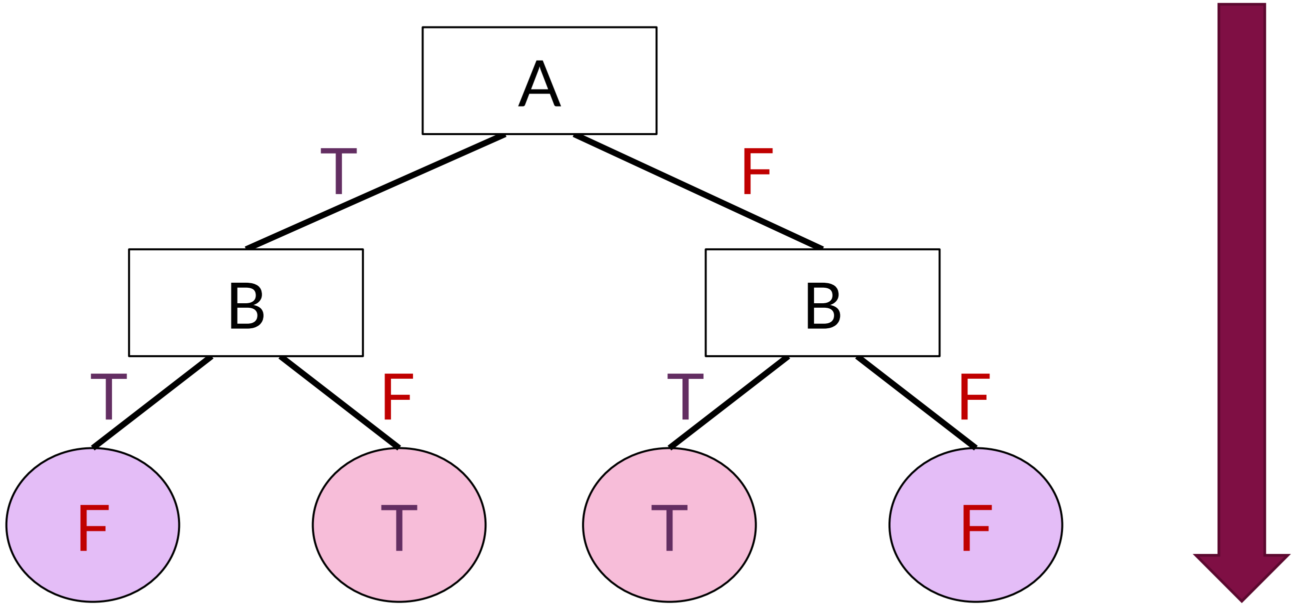
Identity ≈ Category Label



Intuitively, must ask questions such that we expect the answers to:

- "rule out as many category options as possible"
- "reveal as much information about the label as possible"

Top-Down Decision Tree Training – Grow top down



Top-Down Decision Tree Induction

[ID3 (1986), C4.5(1993) by Quinlan]

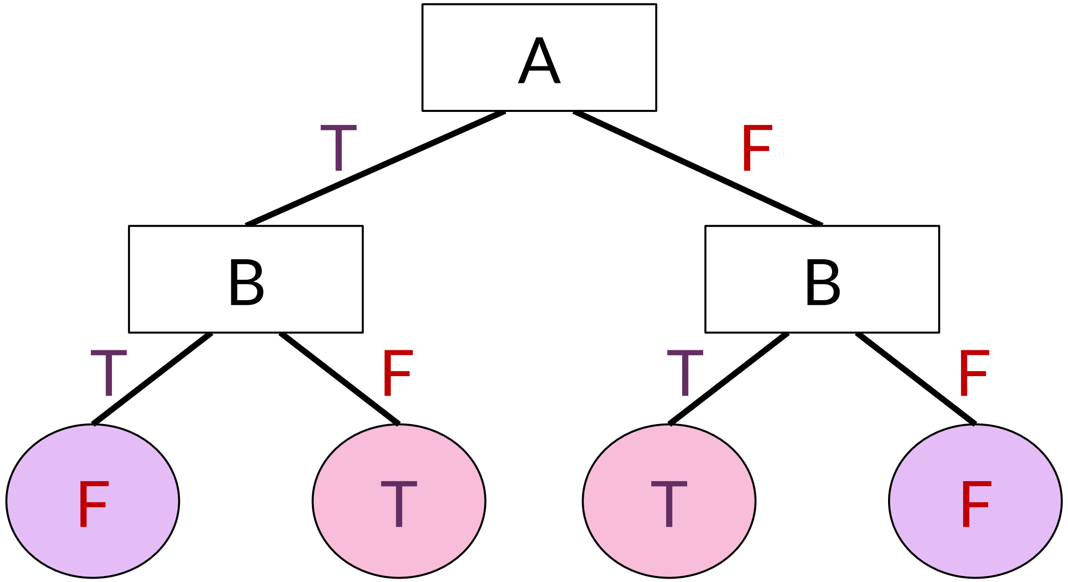
Let $\mathcal{D}$ be a set of labeled instances; $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N} = [X_{N\times D}, \boldsymbol{y}_{N\times 1}]$

Let $\mathcal{D}[X_j = v]$ be the subset of $\mathcal{D}$ where feature X_j has value v

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

1. If data $\mathcal{D}$ all have the same label y, return new `leaf_node`(y)
2. Pick the "best" feature X_j to partition $\mathcal{D}$
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`

Top-Down Decision Tree Training



Top-Down Decision Tree Induction

[ID3, C4.5 by Quinlan]

Let $\mathcal{D}$ be a set of labeled instances; initially $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N} = [X_{N \times D}, \boldsymbol{y}_{N \times 1}]$

Let $\mathcal{D}[X_j = v]$ be the subset of $\mathcal{D}$ where feature X_j has value v

How do we choose which feature is best?

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

1. If data $\mathcal{D}$ all have the same label y, return new `leaf_node(`y`)`
2. Pick the "best" feature X_j to partition $\mathcal{D}$
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`

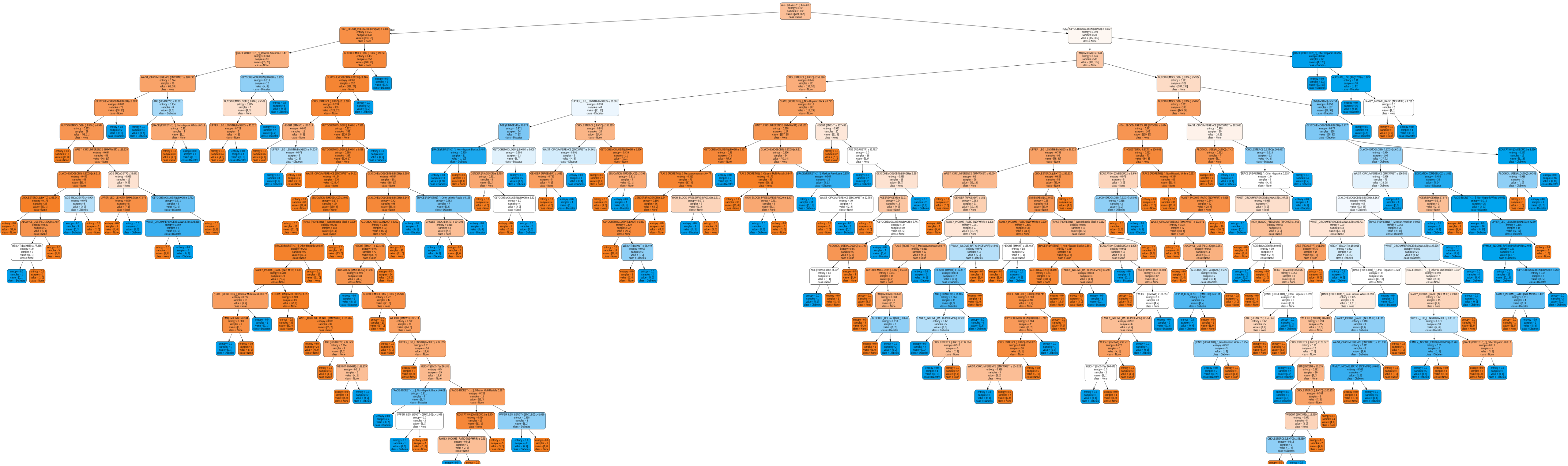
Choosing the "Best Feature"

Key problem: how should we choose which feature to split the data?

Possibilities:

Random
Choose any feature at random?

Diabetes DT – Random Features



So much for interpretability!

Would this even fit the training data?

Is this really the best way to choose splits?

Choosing the Best Feature

Key problem: how should we choose which feature to split the data?

Possibilities:

Random	?
Choose any feature at random	

Choosing the Best Feature

Key problem: how should we choose which feature to split the data?

Possibilities:

Random	**Info-Gain**
Choose any feature at random	Choose the feature with the largest expected *information gain*

i.e., the feature that is expected to result in the shortest subtree

Learning Smaller Models

Learning bias: Occam's Razor

Principle stated by William of Ockham (1285-1347)

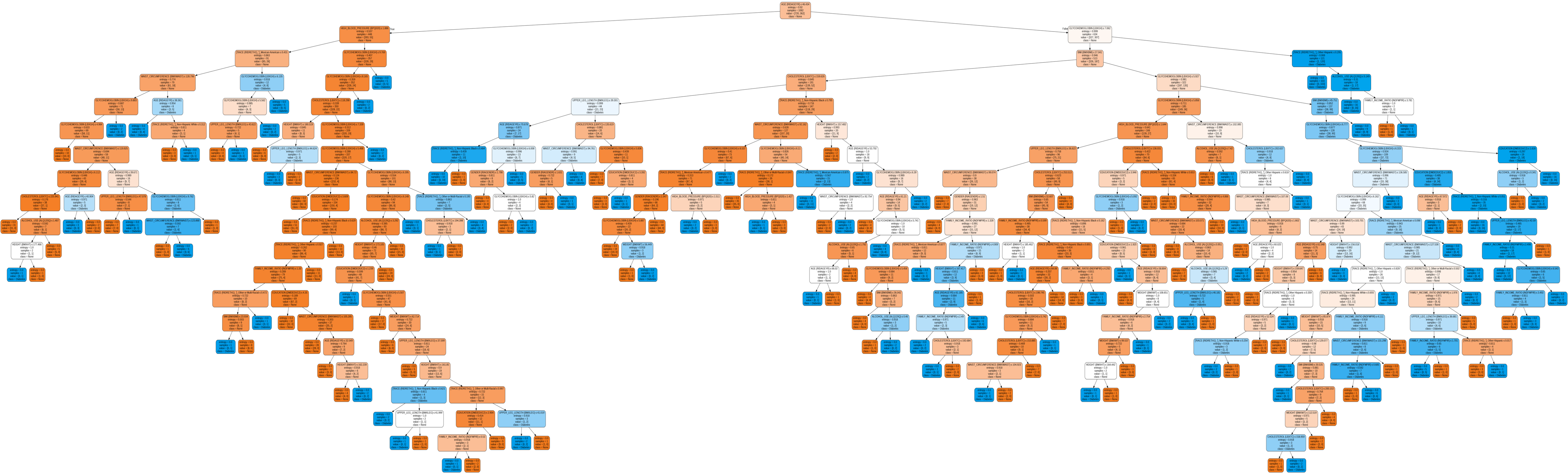
- "non sunt multiplicanda entia praeter necessitatem"
- entities are not to be multiplied beyond necessity
- also called Ockham's Razor, Law of Economy, or Law of Parsimony

Key Idea: The simplest consistent explanation is the best

(Recall: this is also why we have studied bias-variance tradeoffs, regularization, feature selection etc.)

https://en.wikipedia.org/wiki/William_of_Ockham

DT with random features



How could we make smaller trees (and make Occam happy)?

Recap: ID3 learning approach

Top-Down Decision Tree Induction

[ID3 (1986), C4.5(1993) by Quinlan]

Let $\mathcal{D}$ be a set of labeled instances; $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N} = [X_{N\times D}, \boldsymbol{y}_{N\times 1}]$

Let $\mathcal{D}[X_j = v]$ be the subset of $\mathcal{D}$ where feature X_j has value v

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

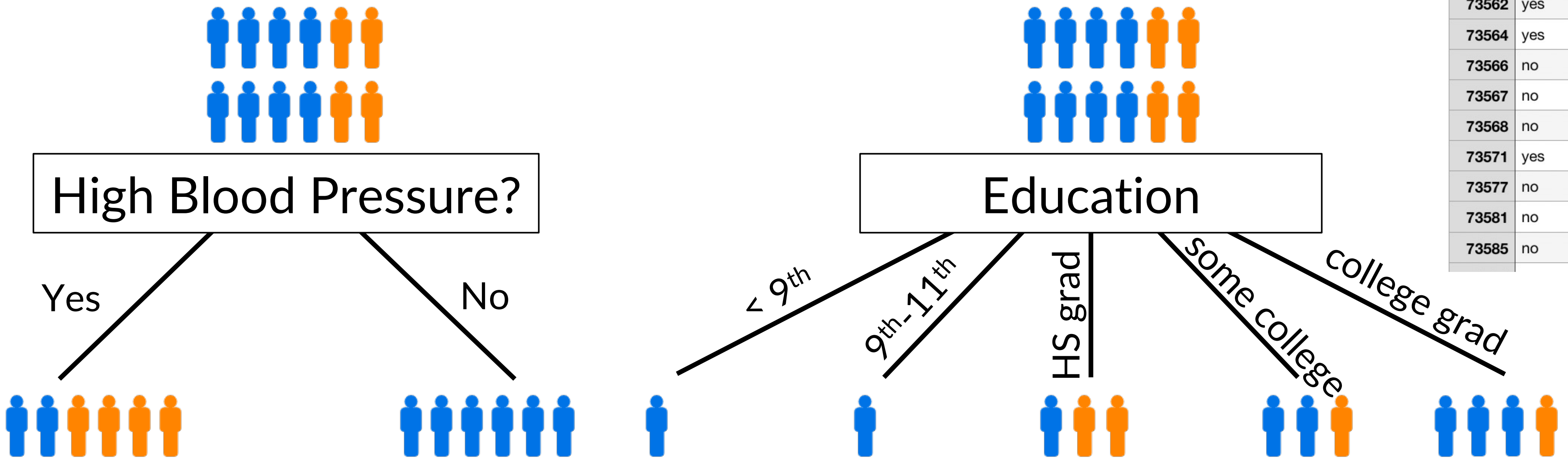
1. If data $\mathcal{D}$ all have the same label y, return new `leaf_node`(y)
2. Pick the "best" feature X_j to partition $\mathcal{D}$
3. Set `node = new decision_node`(X_j)
4. For each value v that X_j can take
   - Recursively create a new child `train_tree`($\mathcal{D}[X_j = v]$) of `node`
5. Return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

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

A Measure of Impurity

Choosing Features for Short Decision Trees

Key Idea: good features ideally partition the data into subsets that are either “all positive” (blue) or “all negative” (orange)



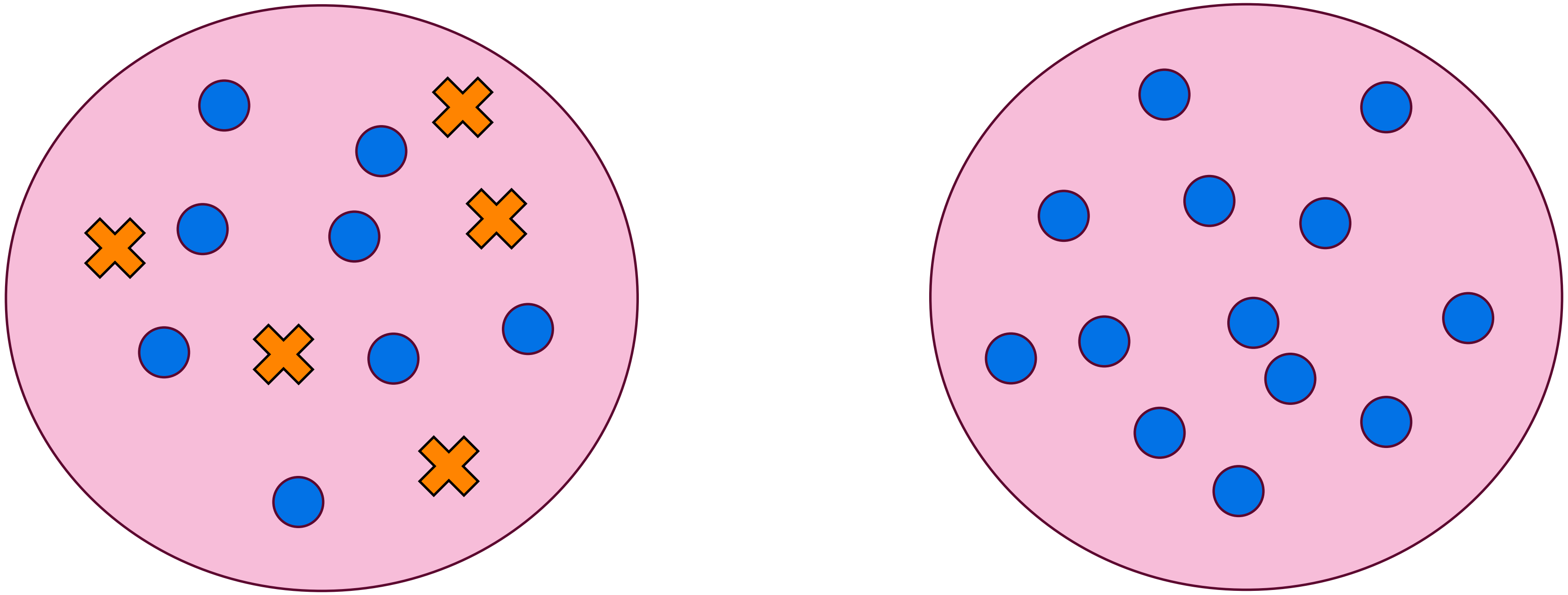
Subset of Data

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

Which split is more informative?

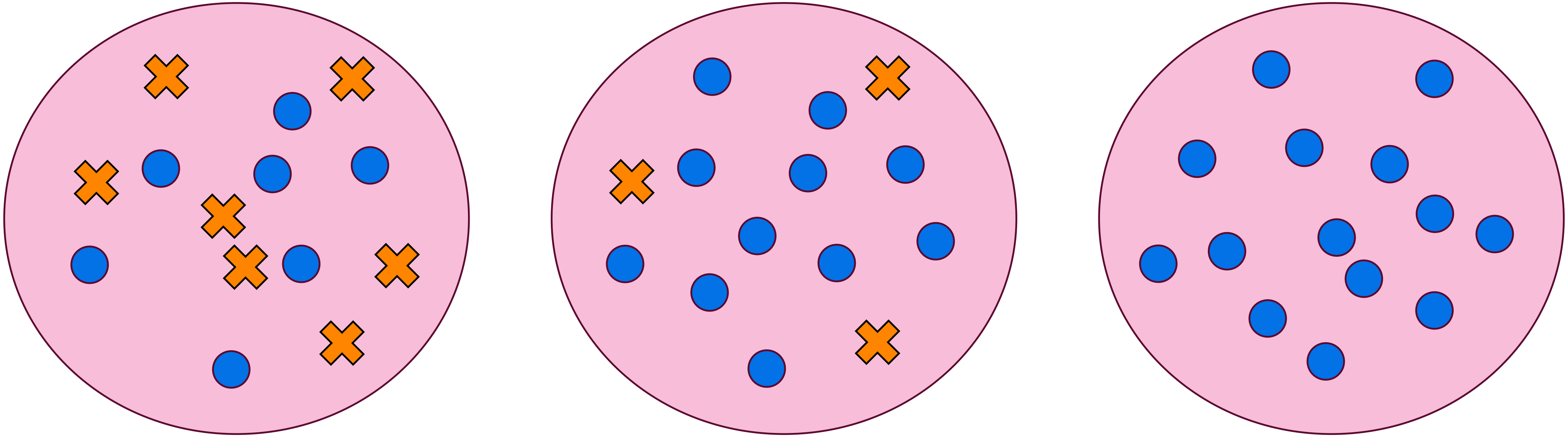
Impurity

- Measures the level of impurity/homogeneity in a group of samples



Impurity

- Measures the level of impurity/homogeneity in a group of samples



maximally impure ⟷ minimally impure

Note: All x's is also "pure"

Could we come up with an "impurity function" of a set of samples?

A Candidate For An "Impurity Function": Entropy

Shannon

- Let Y be any discrete random variable that can take on n values
- The entropy of Y is given by

$$H(Y) = -\sum_{c=1}^{n} P(Y = c) \log_2 P(Y = c)$$

Strictly, the entropy $H(Y)$ maps from a probability distribution (over the class label random variable Y) to an impurity score

$\Updownarrow$

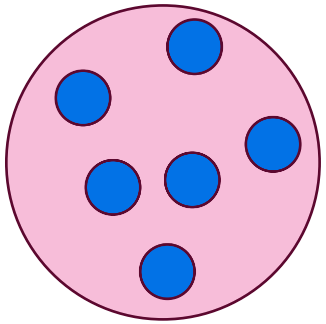
We'll denote $H(\mathcal{D})$ to map from a data subset $\mathcal{D}$ to the impurity score, by setting probability distribution $\approx$ empirical distribution of labels Y in $\mathcal{D}$

Entropy of Binary Classes

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

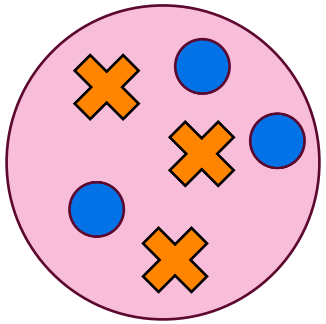
All instances in same class



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

Max Impurity

Instances split evenly among classes



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

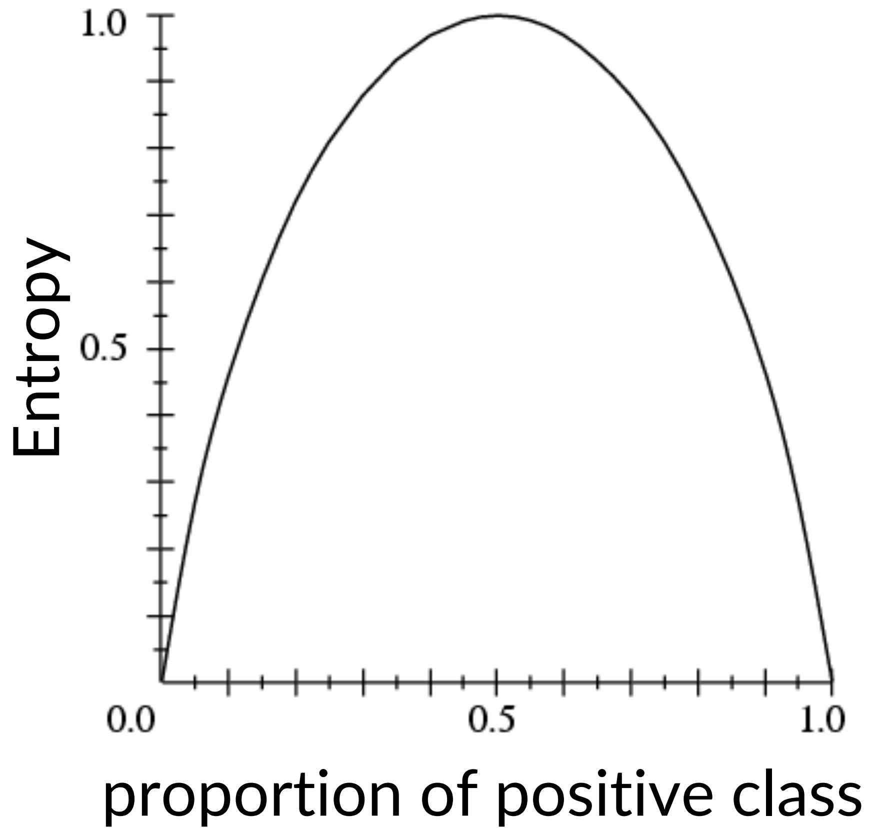
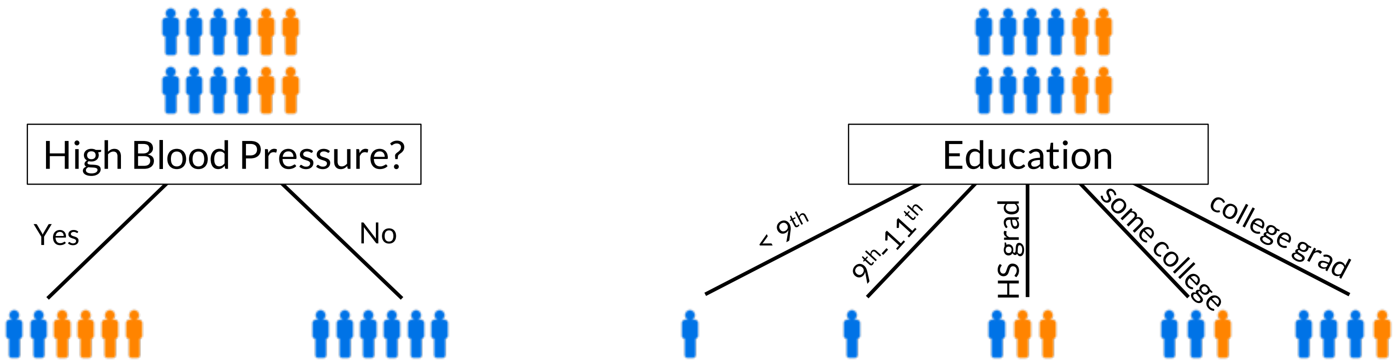


Image: Tom Mitchell

Choosing Features for Short Decision Trees



Recall: Ask questions such that the answers will reduce impurity in child nodes

When considering splitting on attribute / feature X_j,

- Need to estimate the "**expected** drop in impurity" after "getting the answer"/partitioning the data
- "Information Gain" based on our entropy function:

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

Information Gain

Entropy $H(\mathcal{D}) = -\sum_c P(Y = c) \log_2 P(Y = c)$,
where different $c's$ correspond to different class labels

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

- The second term is sometimes called the "conditional entropy":

$$H(\mathcal{D}|X_j) = \sum_v H(\mathcal{D}[X_j = v])P(X_j = v)$$

$E[?]$

- The information gain may then also be written as:

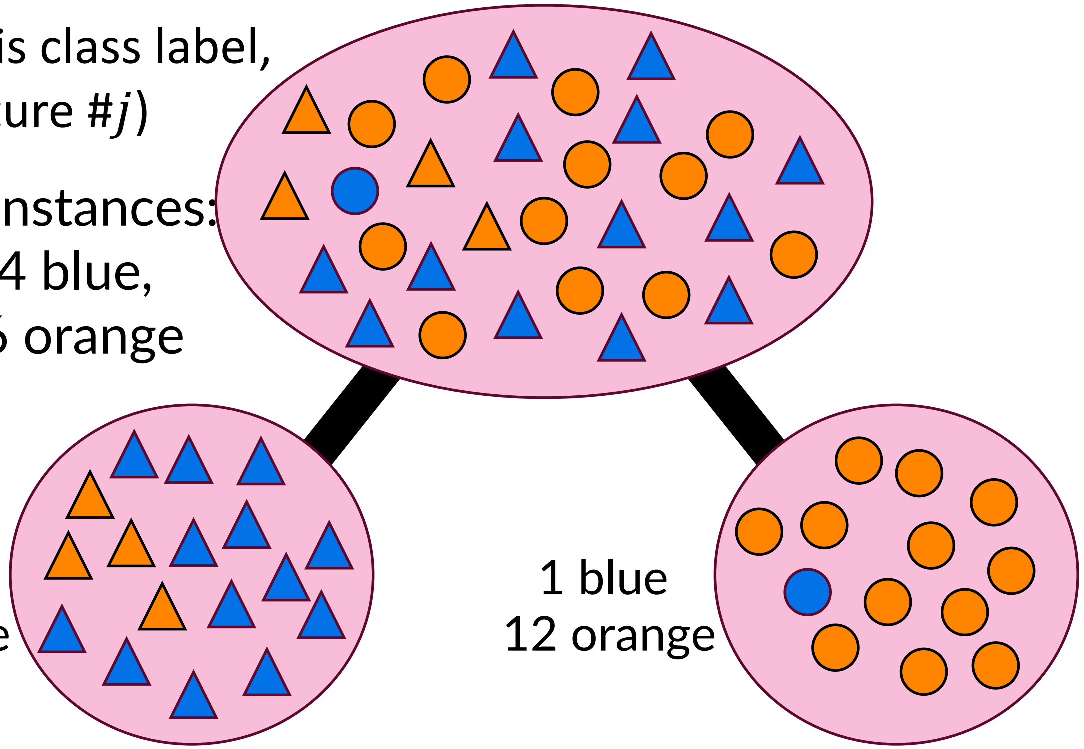
$$IG(\mathcal{D}, X_j) = H(\mathcal{D}) - H(\mathcal{D}|X_j)$$

Example IG Calculation

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

(here: color is class label, shape is feature $\#j$)

30 instances:
14 blue,
16 orange



13 blue
4 orange

1 blue
12 orange

H(child)) =

$$-\left(\frac{13}{17}\log_2\frac{13}{17}\right) - \left(\frac{4}{17}\log_2\frac{4}{17}\right)$$
$$= 0.787$$

H(child)) =

$$-\left(\frac{1}{13}\log_2\frac{1}{13}\right) - \left(\frac{12}{13}\log_2\frac{12}{13}\right)$$
$$= 0.391$$

H(parent) =

$$-\left(\frac{14}{30}\log_2\frac{14}{30}\right) - \left(\frac{16}{30}\log_2\frac{16}{30}\right)$$
$$= 0.996$$

weighted_mean(H(children)) =

$$\frac{17}{30}\cdot 0.787 + \frac{13}{30}\cdot 0.391$$
$$= 0.615$$

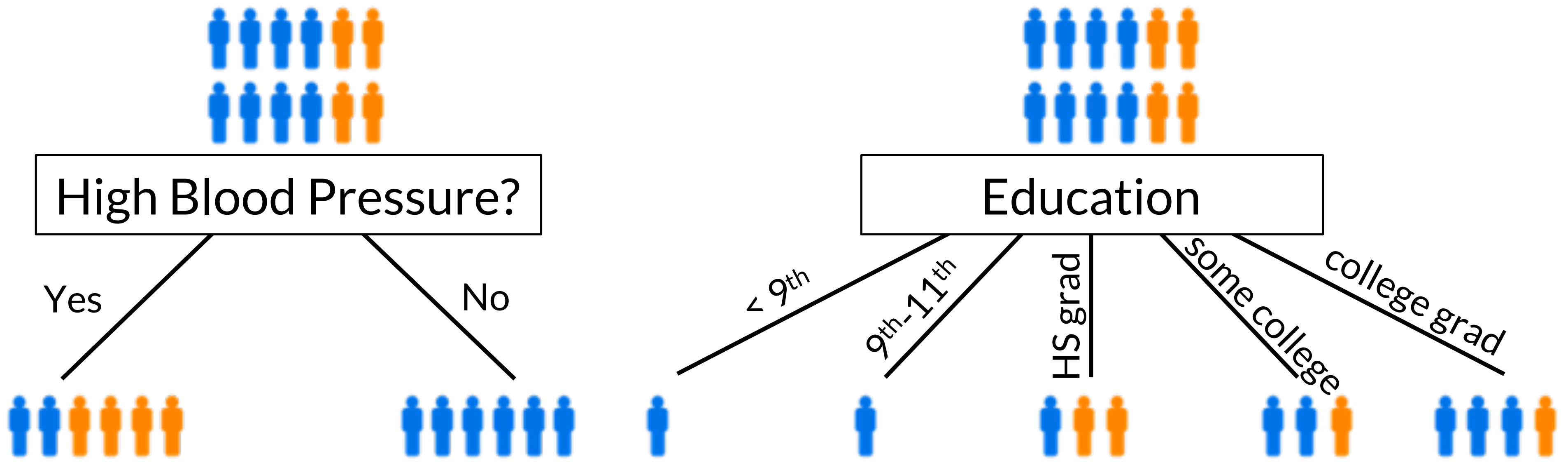
IG = 0.996 – 0.615 = $\boxed{0.381}$

Based on example by Pedro Domingos

Revisiting Our 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

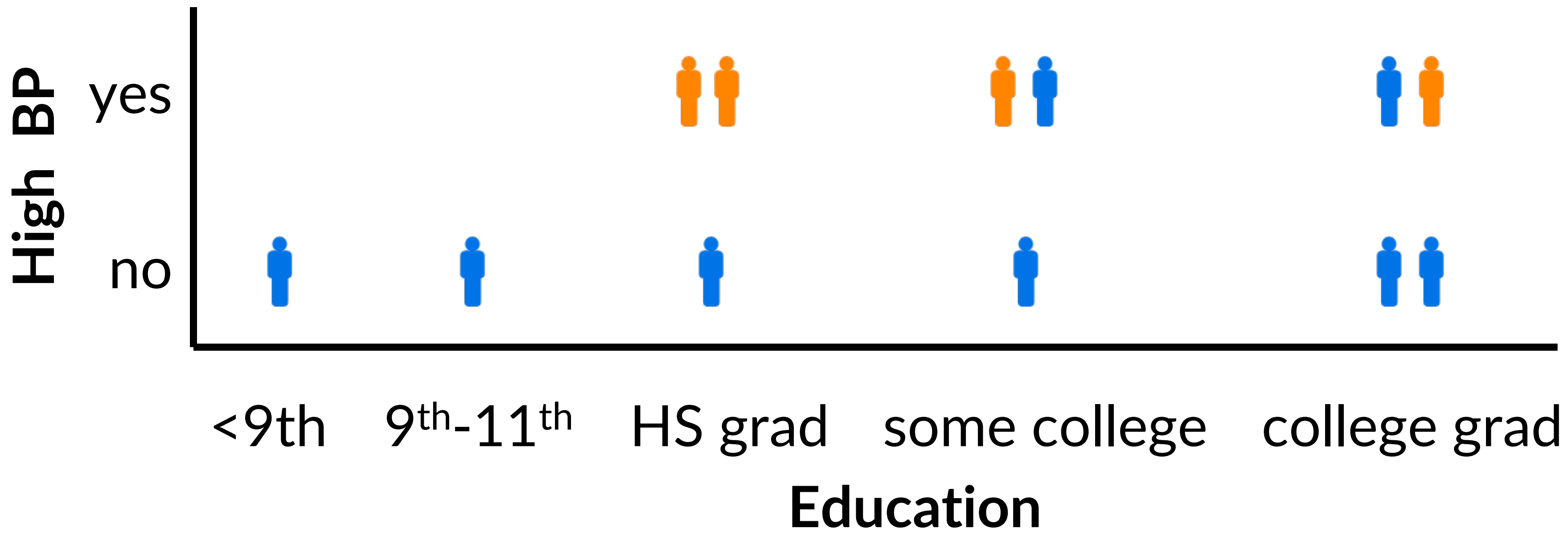
Which split is more informative?



Now we can solve it computationally via information gain

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



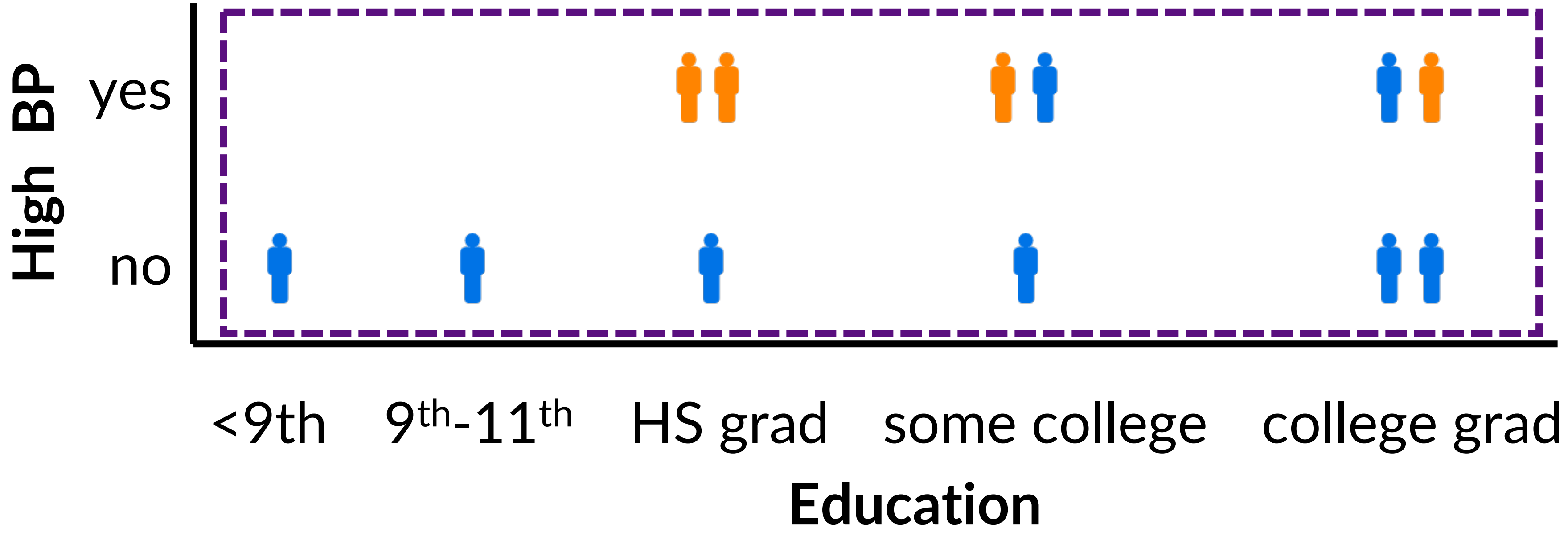
Need to compute:

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

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

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:

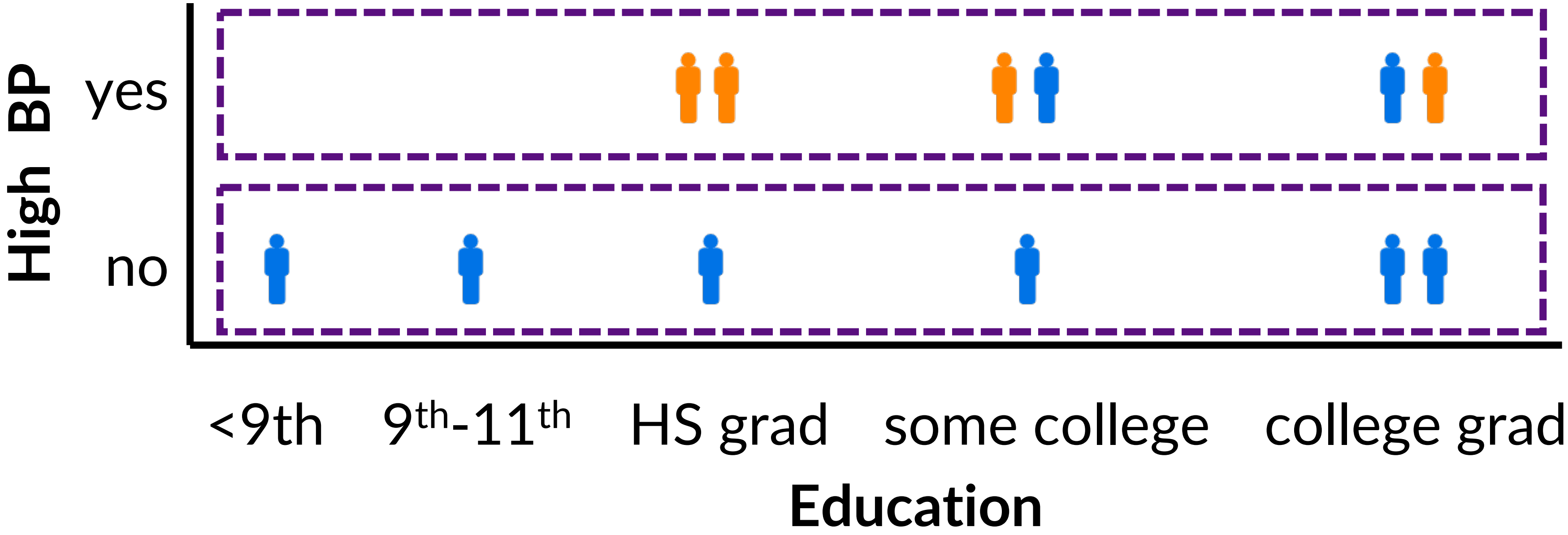
$$IG(\mathcal{D}, High\ BP) = H(\mathcal{D}) - H(\mathcal{D}\,|High\ BP)$$

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

$$H(\mathcal{D}) = -\ 4/12 \lg 4/12 \\ -\ 8/12 \lg 8/12 \\ = 0.918$$

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:

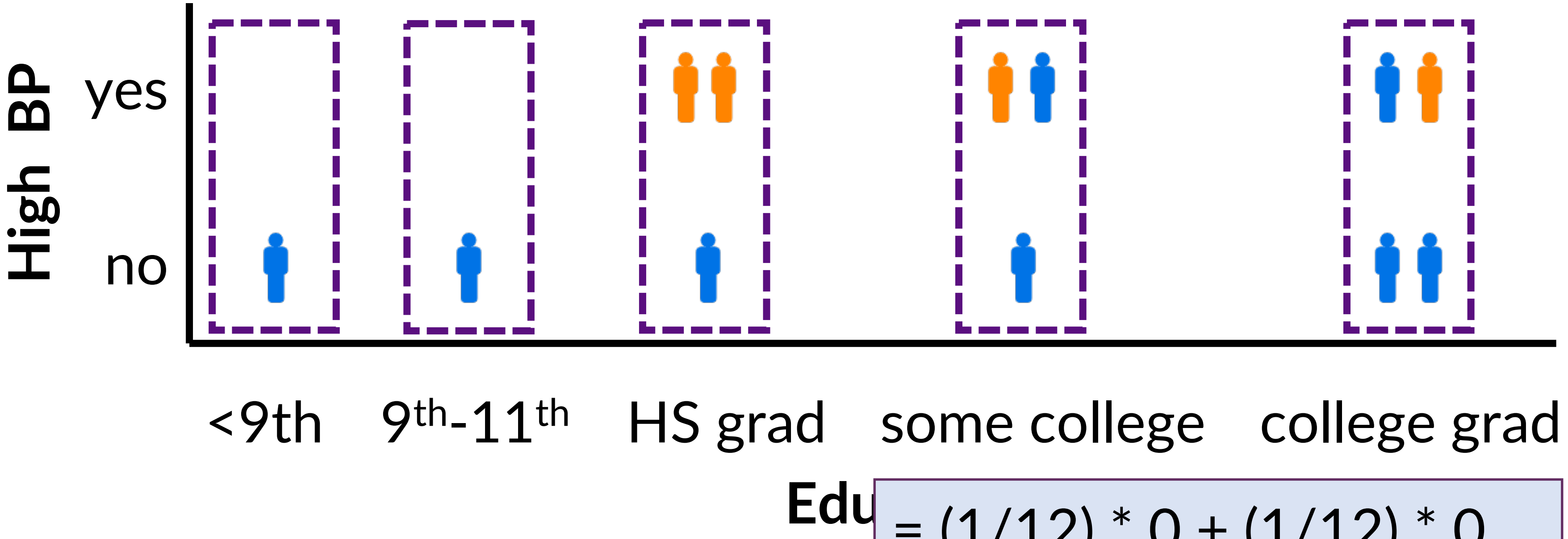
$$IG(\mathcal{D}, High\ BP) = H(\mathcal{D}) - H(\mathcal{D}\,|High\ BP)$$

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

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

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



$$= (1/12) * 0 + (1/12) * 0 + (3/12) * (-1/3 \lg 1/3 - 2/3 \lg 2/3) + (3/12) * (-2/3 \lg 2/3 - 1/3 \lg 1/3) + (4/12) * (-3/4 \lg 3/4 - 1/4 \lg 1/4) = 0.730$$

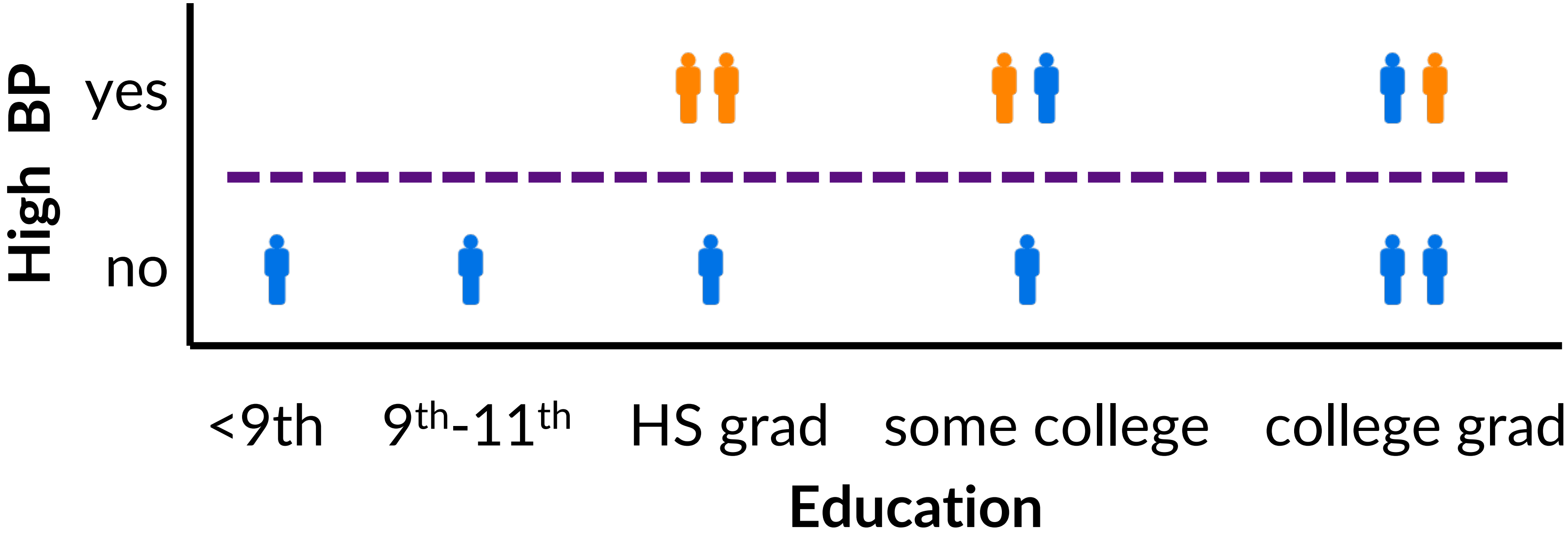
Need to compute:

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

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

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:

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

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

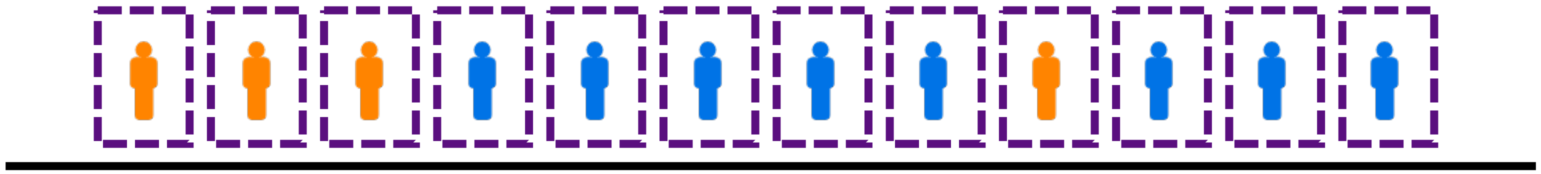
A Problem with Information Gain

- 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

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

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

This scales by the entropy of the split, ignoring classes

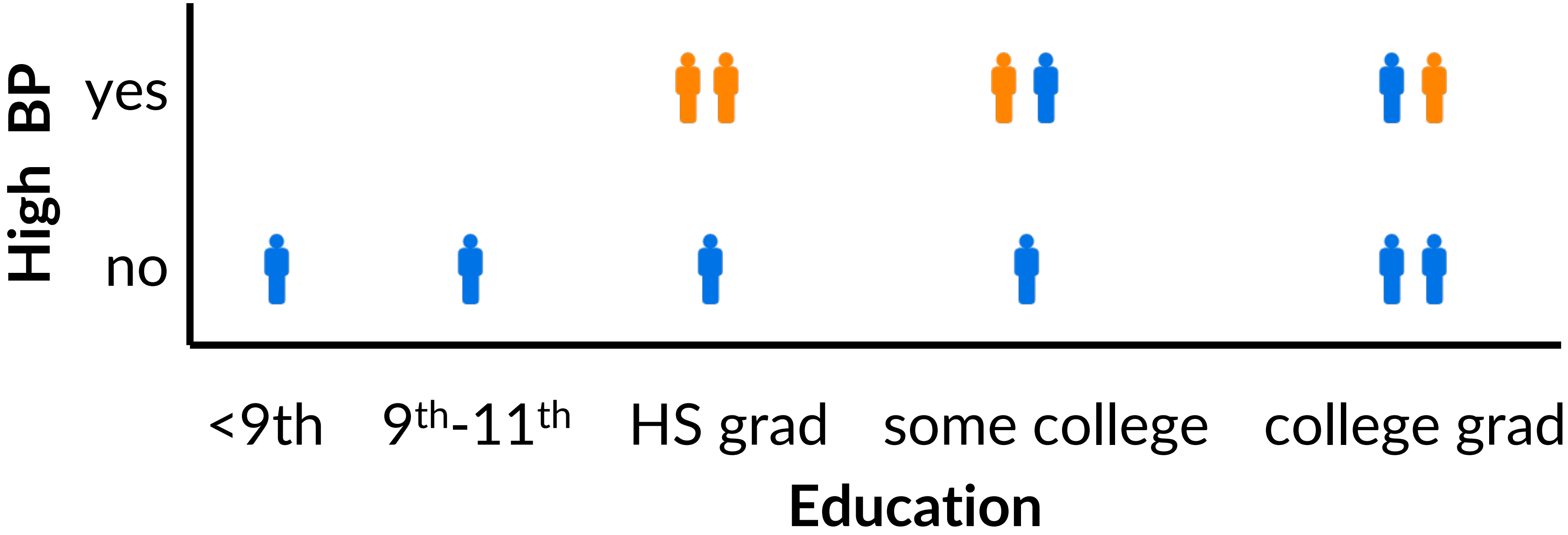
Split information measures the intrinsic information in the feature, not specific to the task --- it doesn't account for the class labels in any way.

Higher split information =>

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

Gain Ratio Example

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



Need to compute:

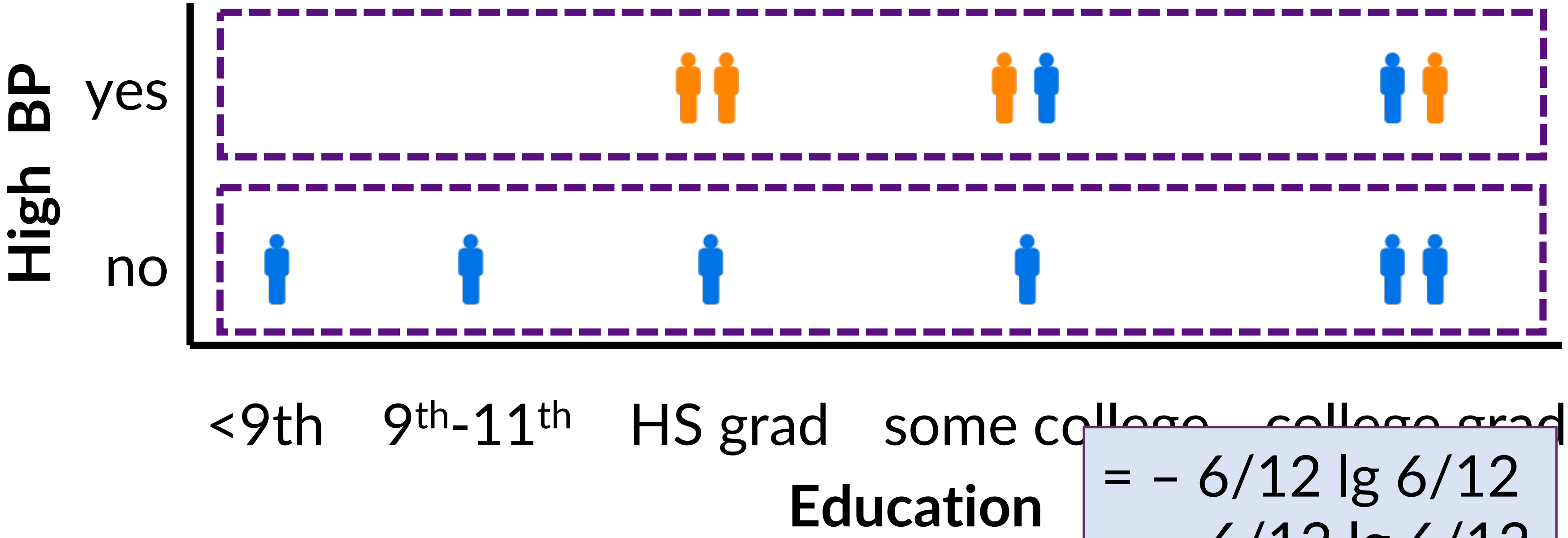
$$\text{GainRatio}(\mathcal{D}\ \text{High BP}) = IG(\mathcal{D}, \text{High BP}) / \text{SplitInfo}(\mathcal{D}, \text{High BP})$$

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

Gain Ratio Example

Already Computed:

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



$= -\,6/12 \lg 6/12$
$\quad -\,6/12 \lg 6/12$
$= 1$

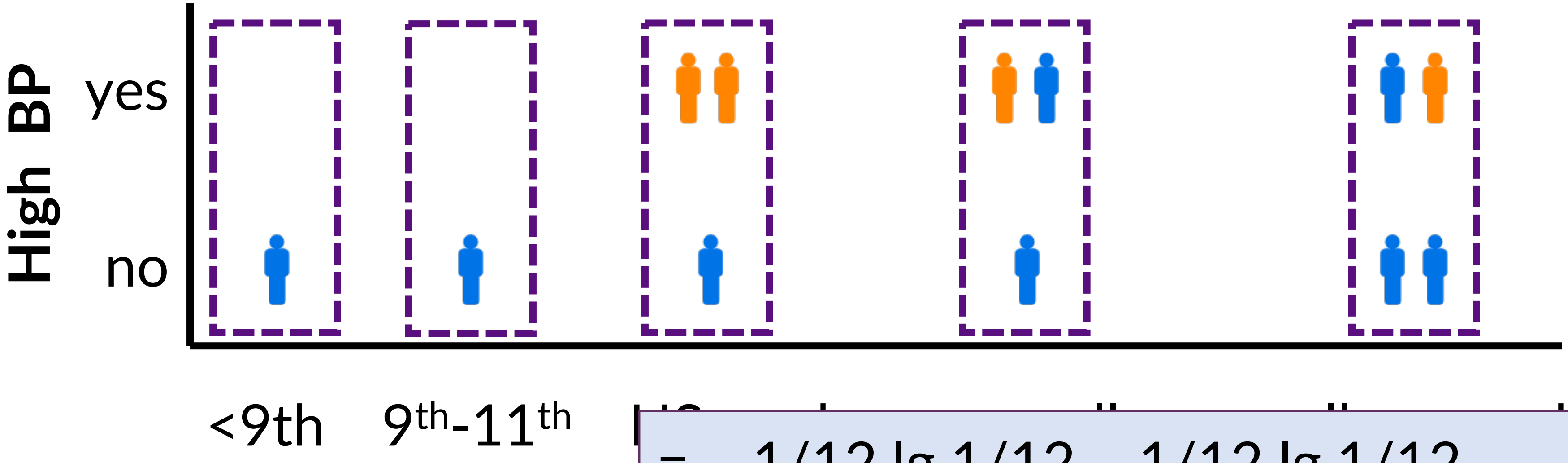
Need to compute:

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

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

Gain Ratio Example

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



$$= -\tfrac{1}{12}\lg\tfrac{1}{12} - \tfrac{1}{12}\lg\tfrac{1}{12} - \tfrac{3}{12}\lg\tfrac{3}{12} - \tfrac{3}{12}\lg\tfrac{3}{12} - \tfrac{4}{12}\lg\tfrac{4}{12} = 2.1258$$

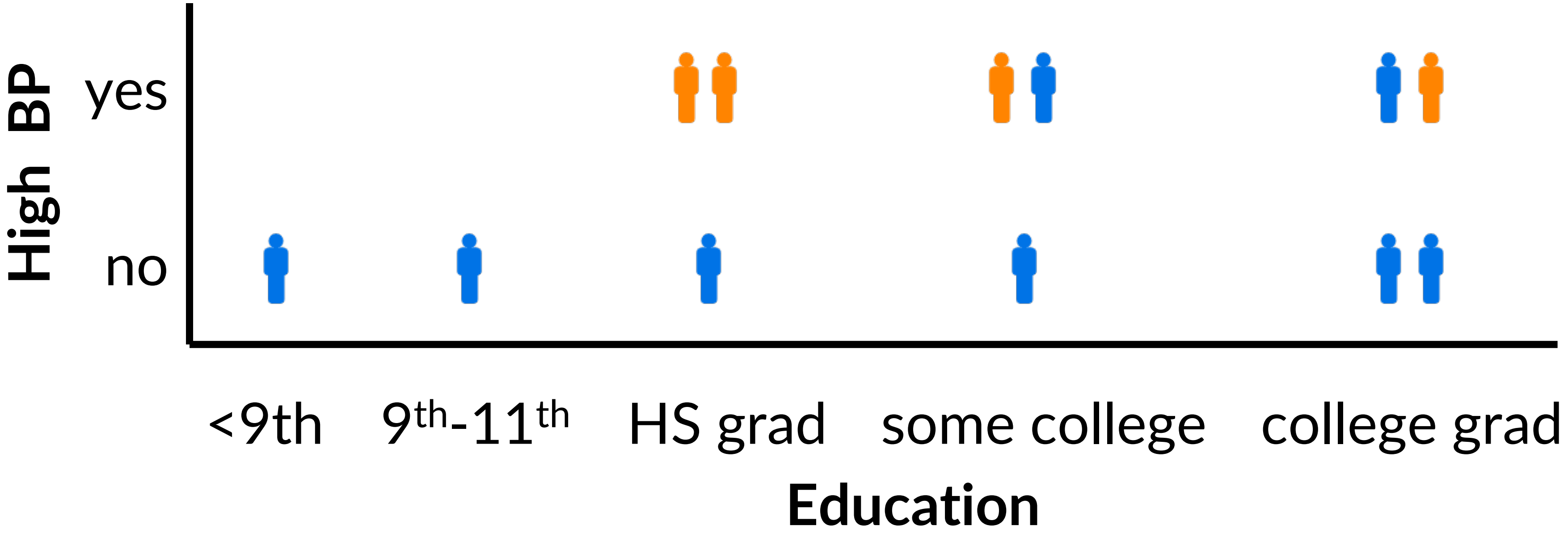
Need to compute:

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

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

Gain Ratio Example

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



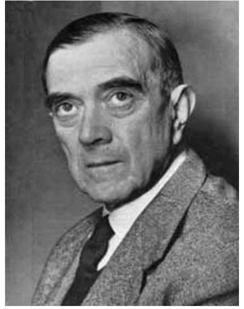
Need to compute:

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

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

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.

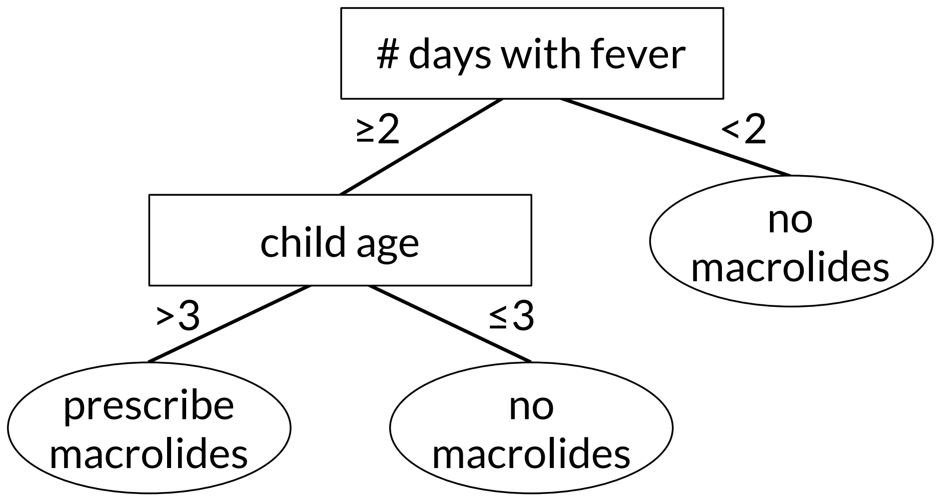
- Recall how we compute Information Gain = Entropy Reduction:
  - Entropy $H(\mathcal{D}) = \sum_c P(Y = c)(-\log_2 P(Y = c))$
  - Information Gain = Entropy of parent – Weighted Average Entropy of Children
- Analogously, Gini Index Reduction:
  - Gini index $\text{Gini}(\mathcal{D}) = \sum_c P(Y = c)\,(1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

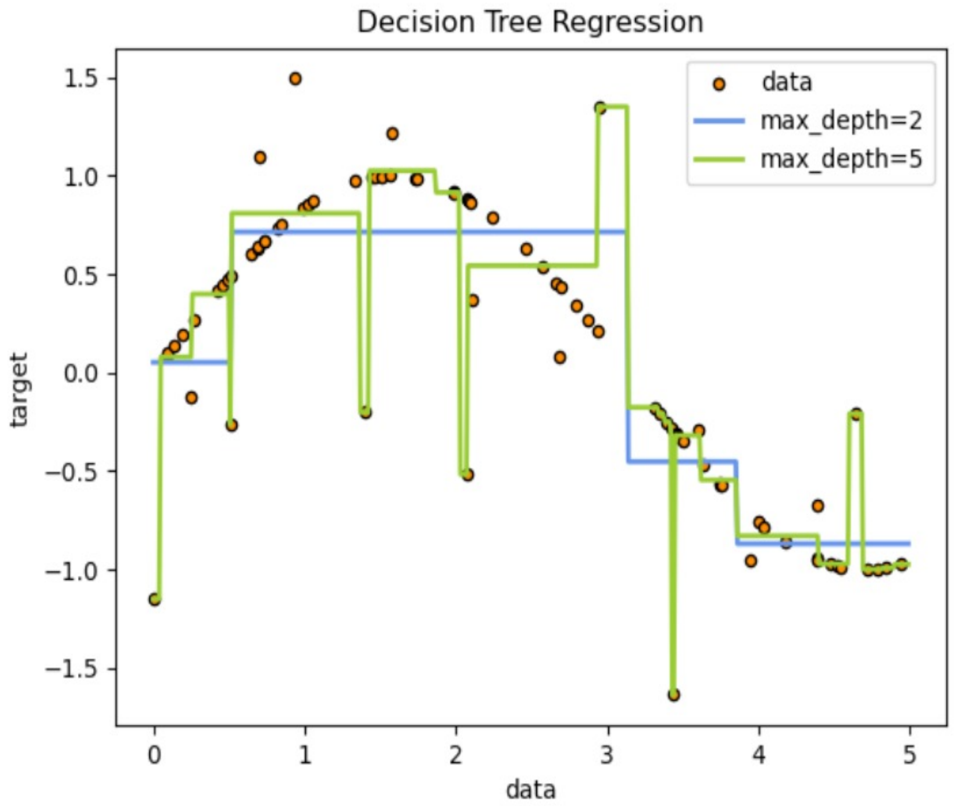
  - Then, choose among splits by IG or GR

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

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

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

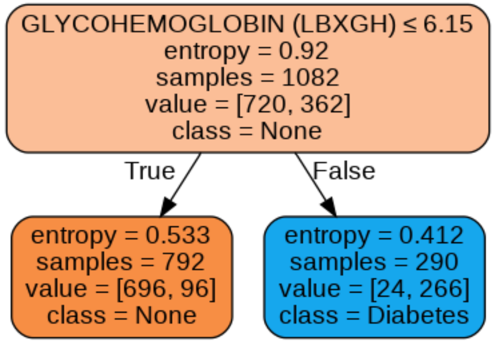
SEQN	RIDAGEYR	BMXWAIST	BMXHT	LBXTC	BMXLEG	BMXWT	BMXBMI	RIDRETH1	BPQ020	ALQ120Q	DMDEDUC2	RIAGENDR	INDFMPIR	LBXGH	DIABETIC
73557	69.0	100.0	171.3	167.0	39.2	78.3	26.7	Non-Hispanic Black	yes	1.0	high school graduate / GED	male	0.84	13.9	yes
73558	54.0	107.6	176.8	170.0	40.0	89.5	28.6	Non-Hispanic White	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-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

$\boldsymbol{X}_1$ $\boldsymbol{X}_2$ . . . $\boldsymbol{X}_{14}$

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

$\boldsymbol{X}_{14}$ (LBXGH) ≤ 6.15 has the highest IG



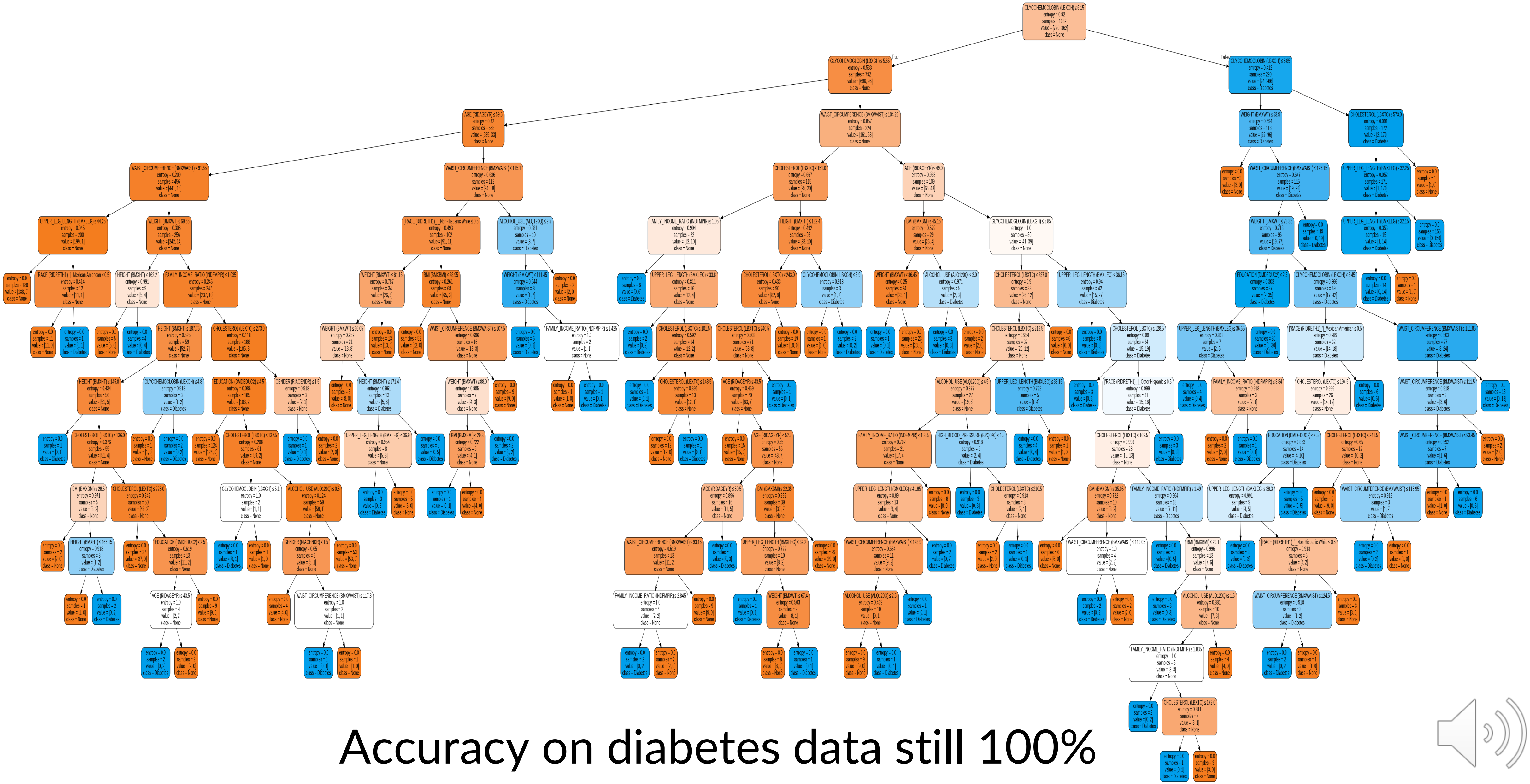
Dataset partition $\mathcal{D}$[LBXGH ≤ 6.15]

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

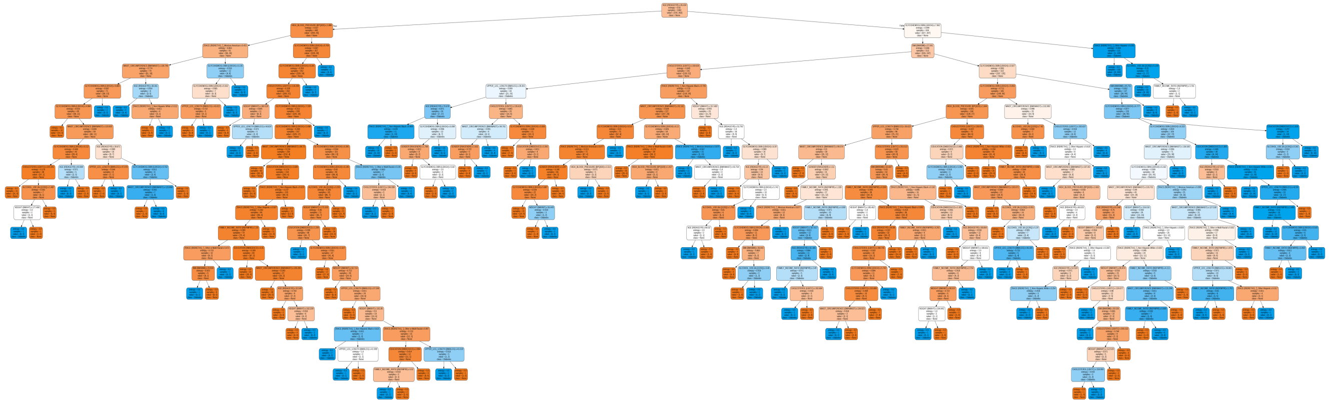
Decision Tree (Version 1)



Accuracy on diabetes data still 100%

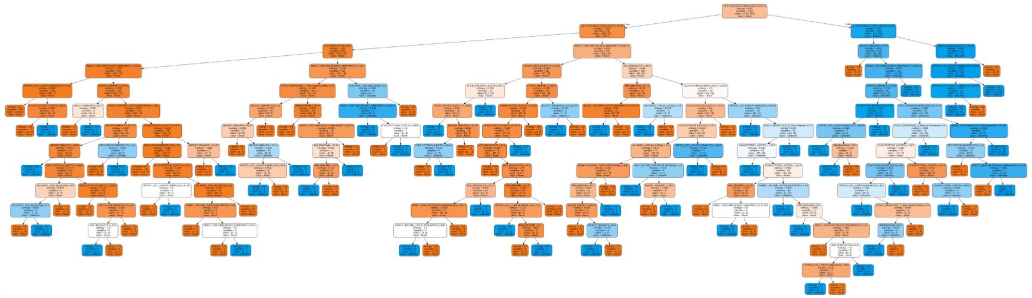
Diabetes DT – Random vs IG Features

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 …

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

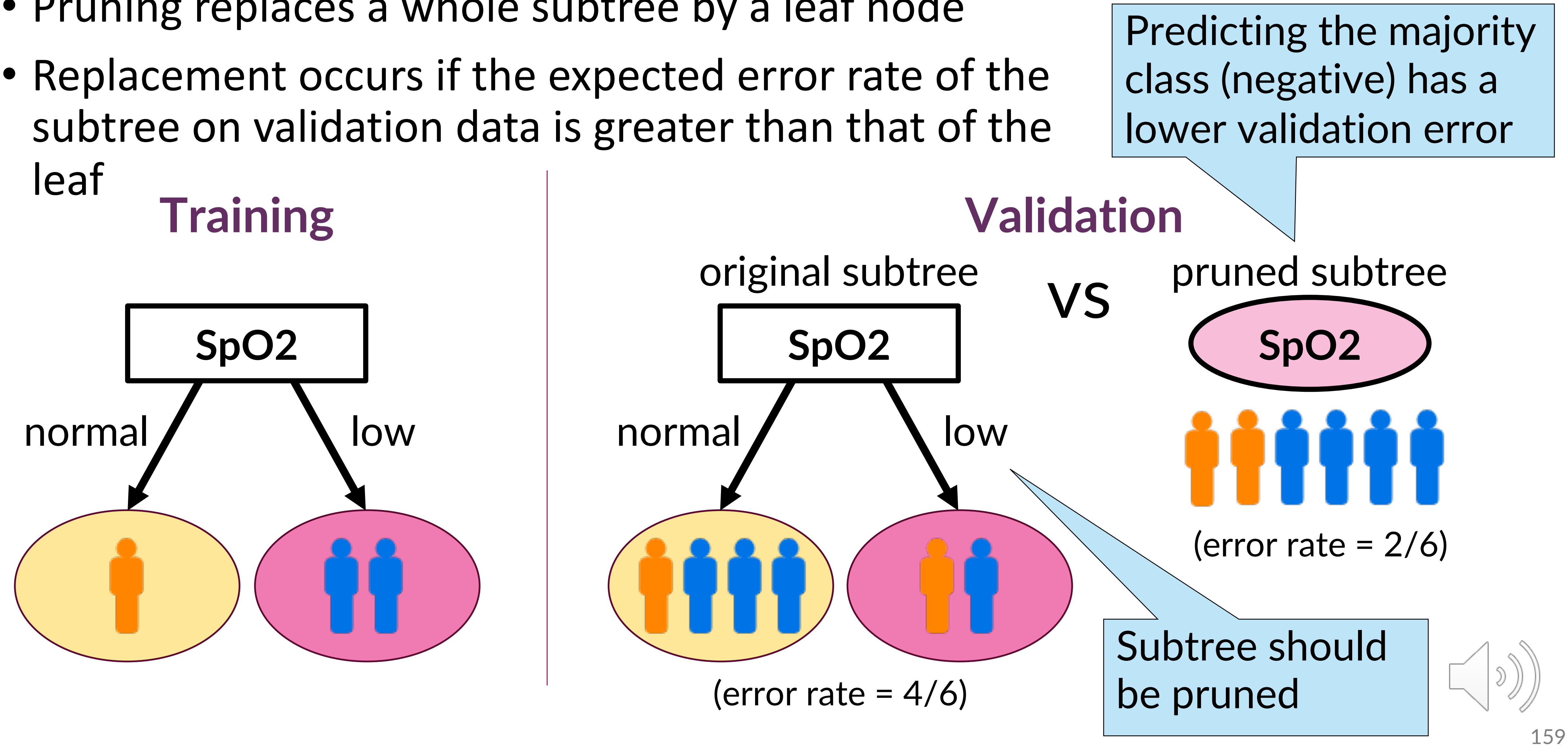
- Grow the decision tree based on the training set

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

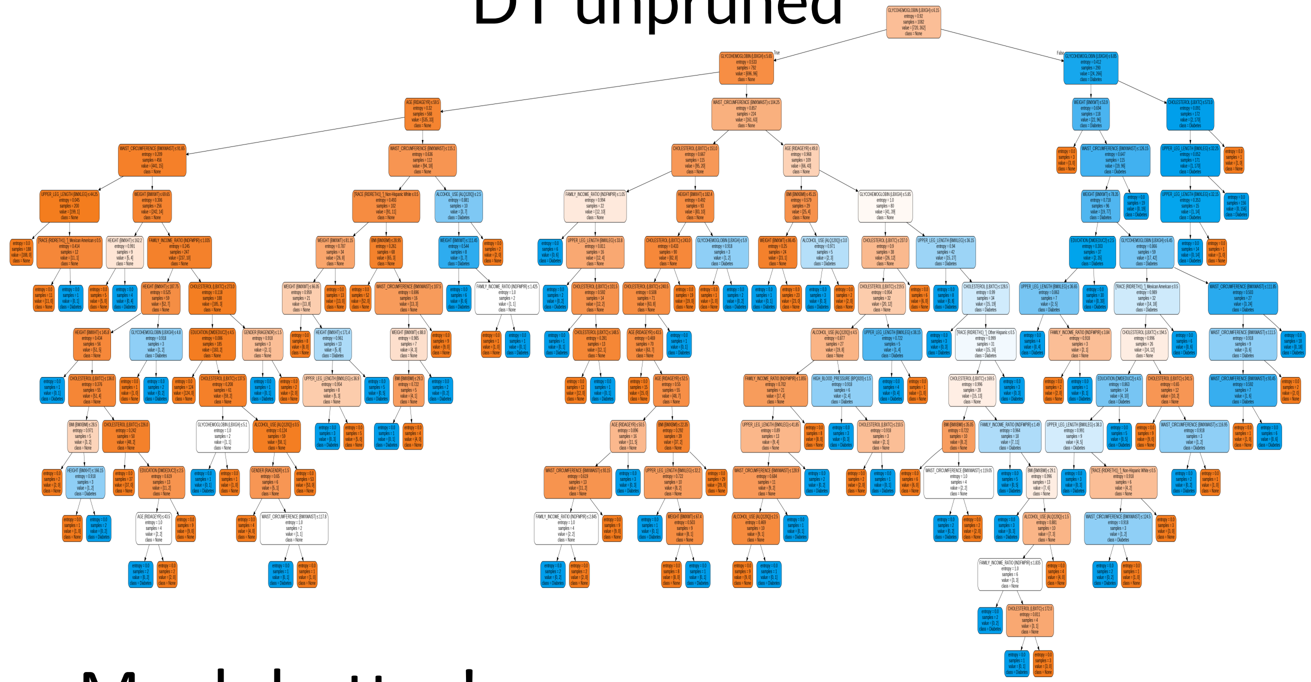
Overview: Reduced-Error Pruning

- 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

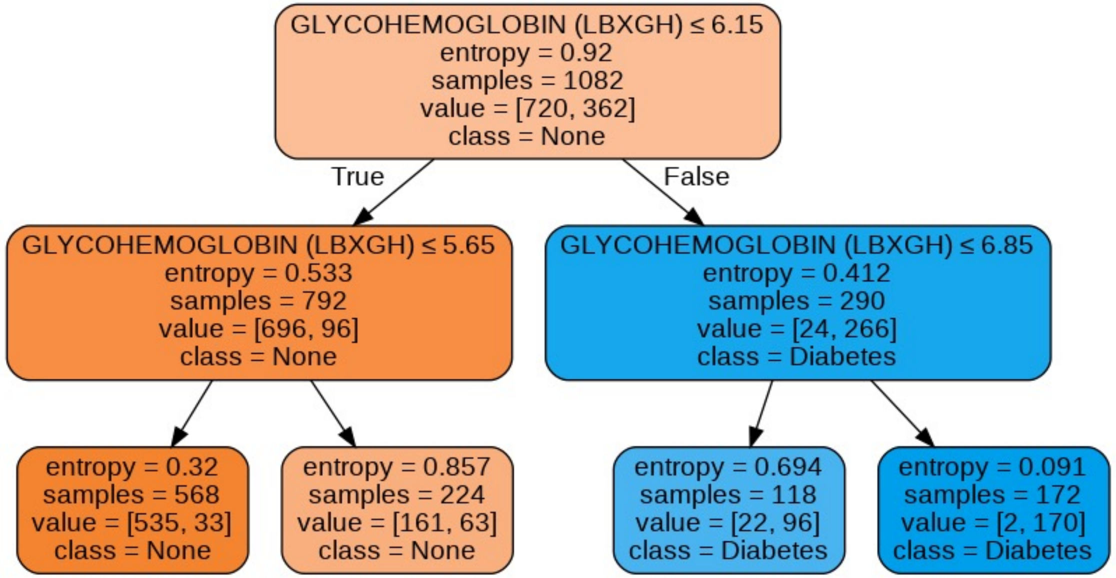


Reduced-Error Pruning on the Diabetes DT

DT unpruned



DT pruned

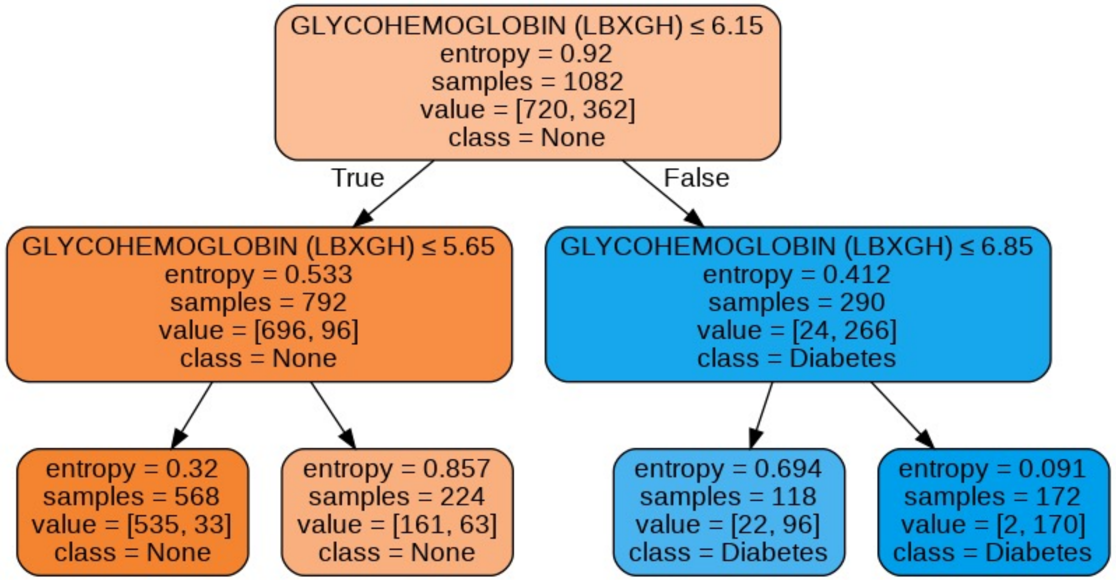


Much better!

	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 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).