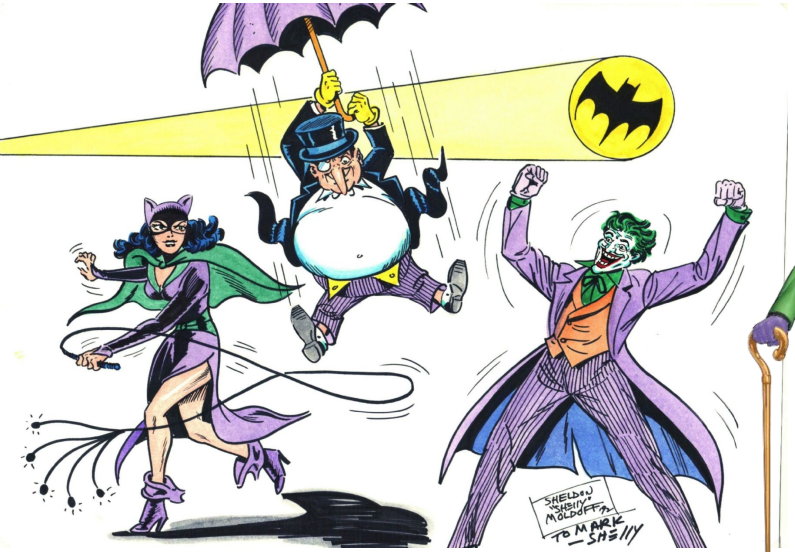
Decision: Are these comic book characters good or evil?



BATMAN ROBIN BATGIRL ALFRED PENNYWORTH

Problem: Is your date good or evil?

	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?

Training data

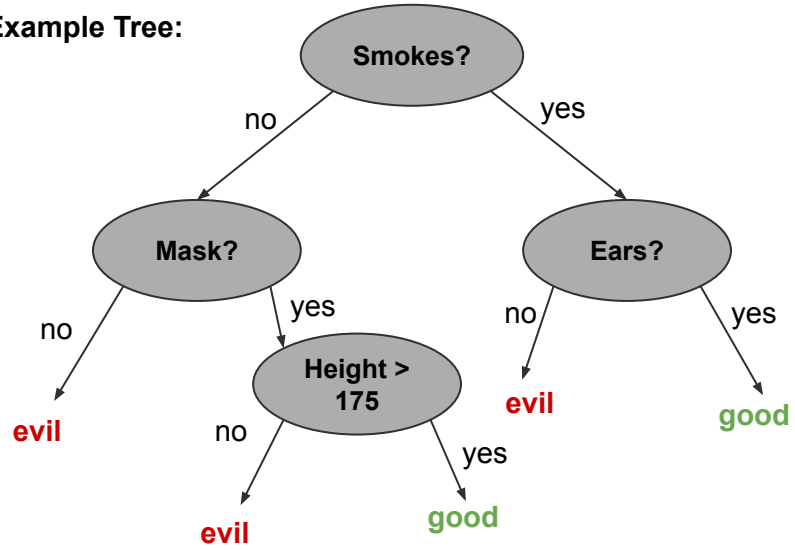
Test Data

Dataset and example from Dr. Kilian Weinberger @Cornell

Decision Tree

	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?

Example Tree:



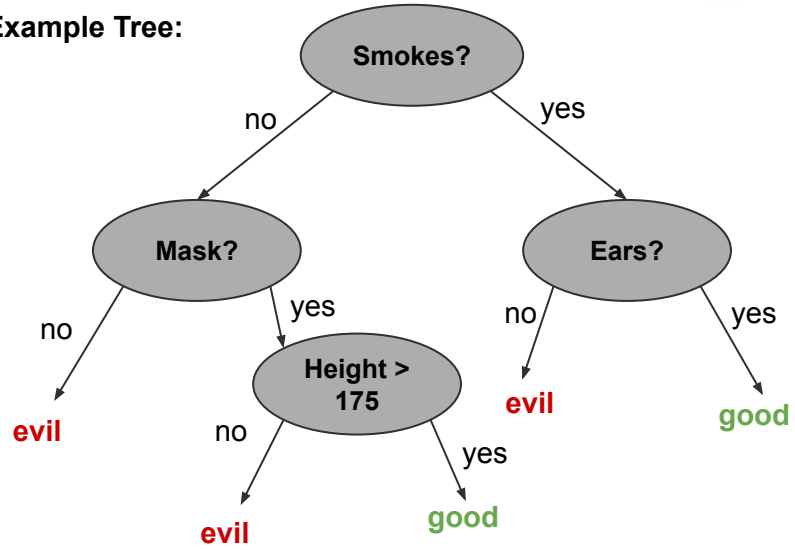
Question: Is this a good tree?

Is this tree **consistent:** would it classify everyone correctly?

Decision Tree

	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?

Example Tree:

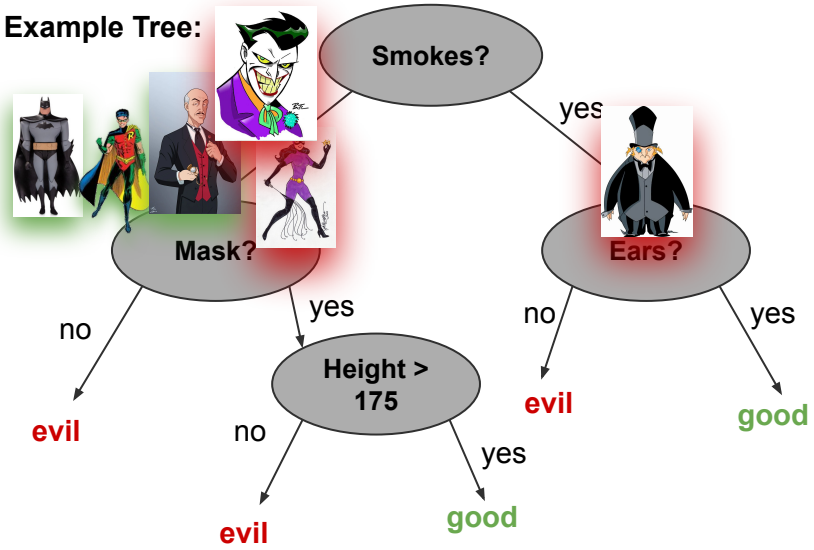


Question: Is this a good tree?

Is this tree **consistent:** would it classify everyone correctly?

Decision Tree

	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?

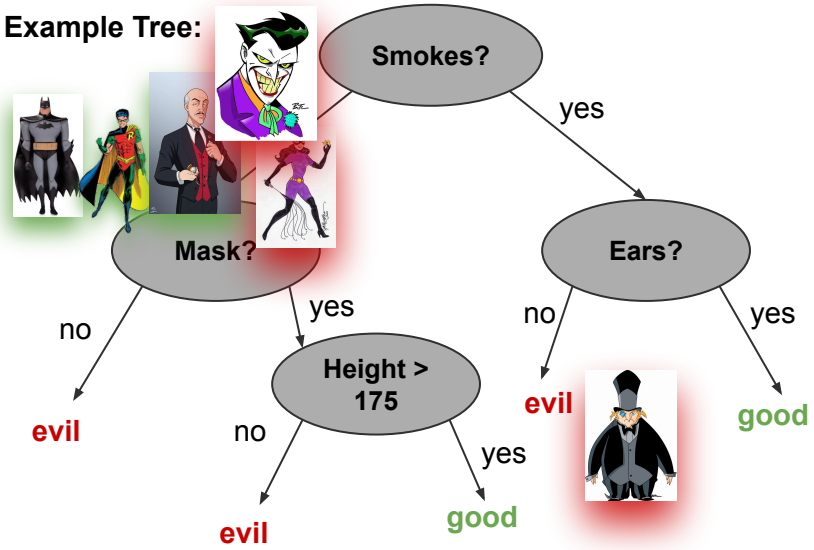


Question: Is this a good tree?

Is this tree **consistent:** would it classify everyone correctly?

Decision Tree

	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?



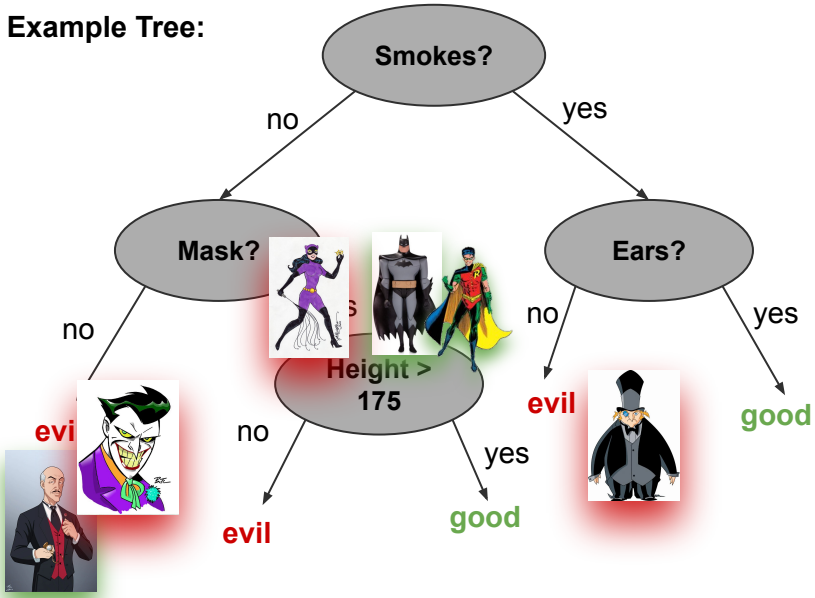
Question: Is this a good tree?

Is this tree **consistent:** would it classify everyone correctly?

Decision Tree

	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?

Example Tree:



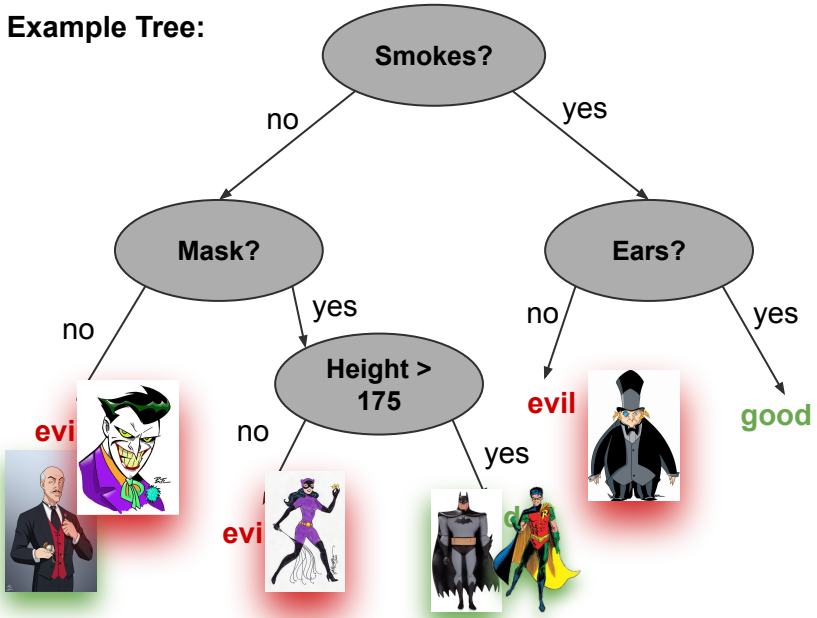
Question: Is this a good tree?

Is this tree **consistent:** would it classify everyone correctly?

Decision Tree

	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?

Example Tree:



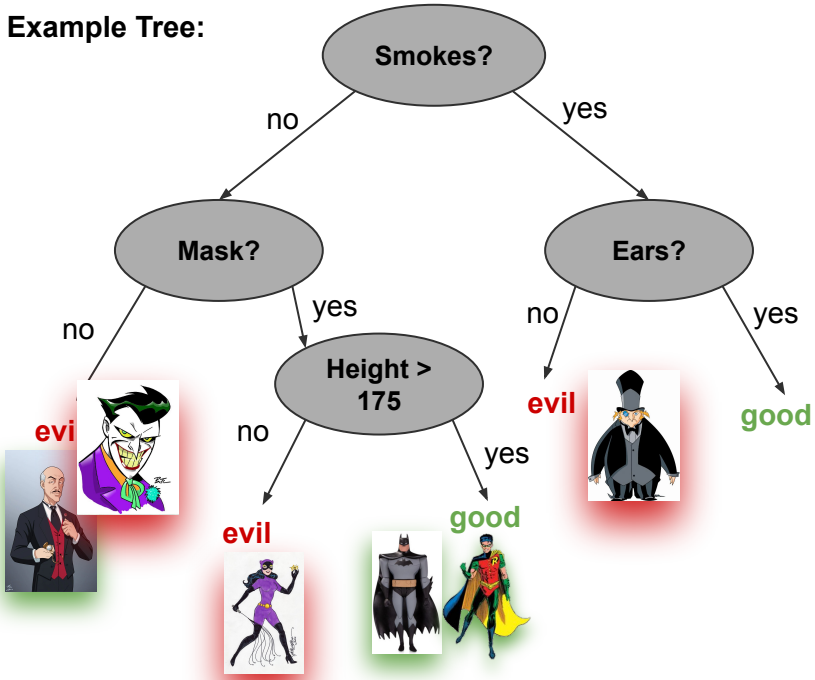
Answer:

- No, it is not consistent. It misclassified Alfred as evil

Decision Tree

	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?

Example Tree:

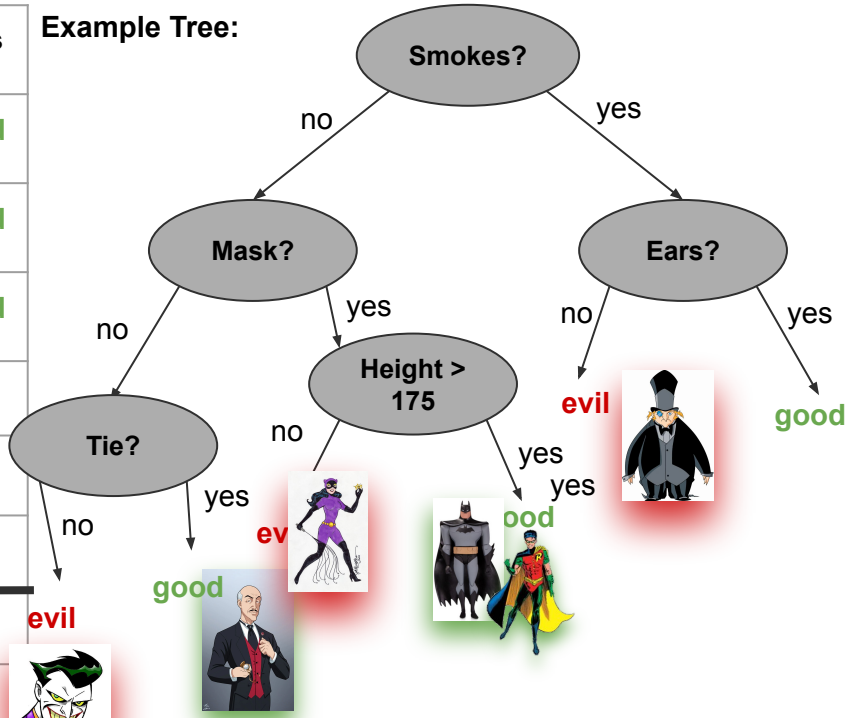


Question: What can we do to make this tree consistent?

Decision Tree

	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?

Example Tree:

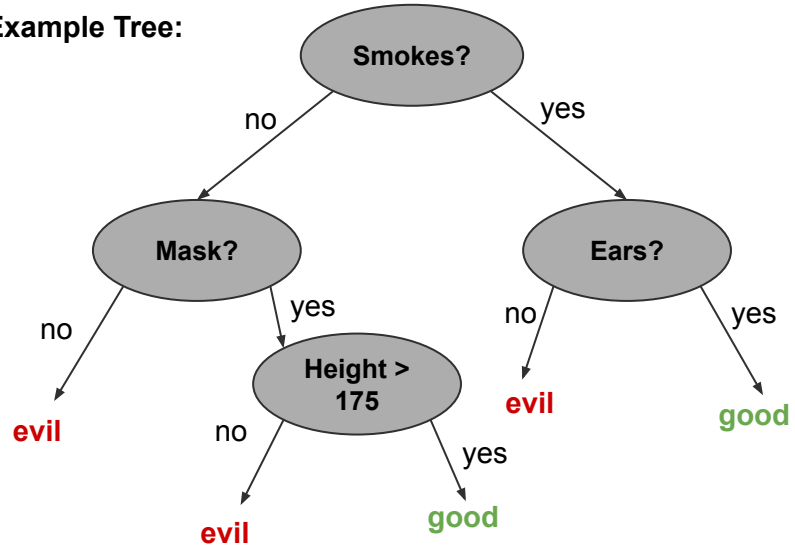


One Possibility: Add tie as an attribute

Activity: What is the smallest consistent tree?

	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?

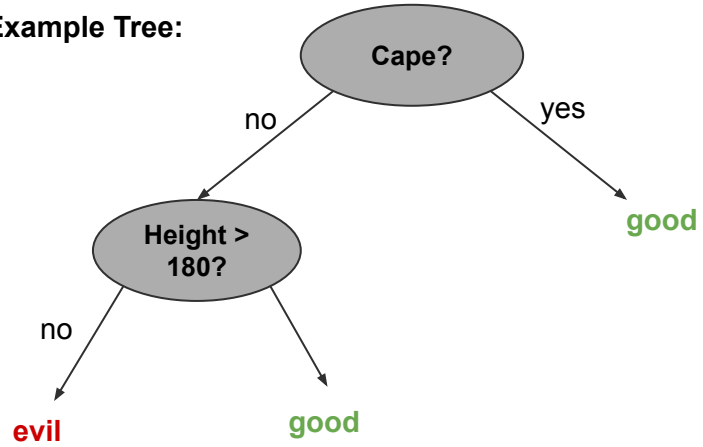
Example Tree:



Decision Tree

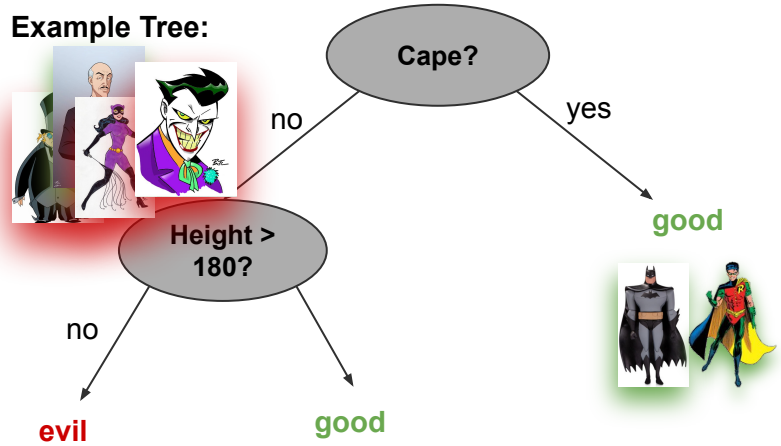
	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?

Example Tree:



Decision Tree

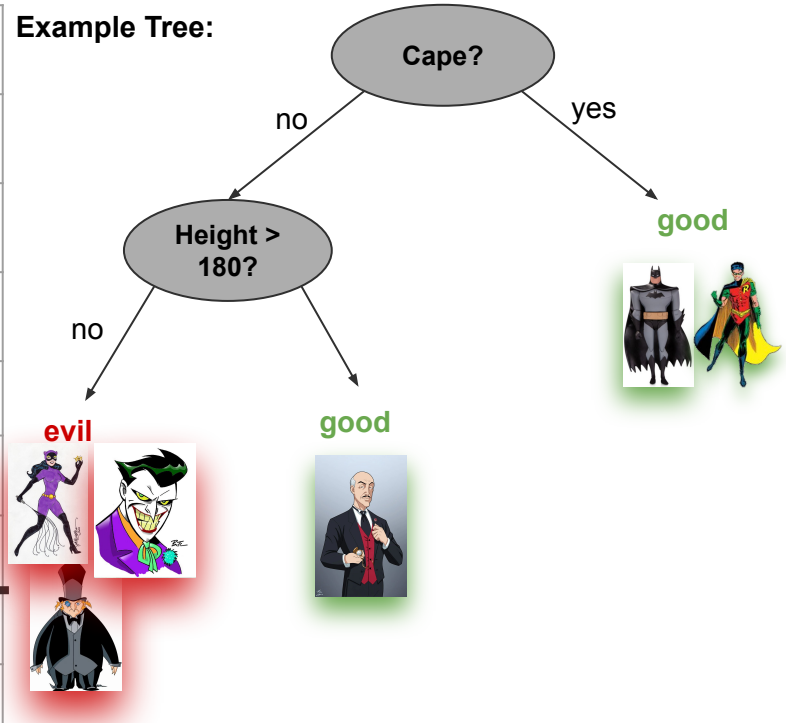
	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?



Decision Tree

	mask	cape	tie	ears	smokes	height	class
Batman	y	y	n	y	n	180	good
Robin	y	y	n	n	n	176	good
Alfred	n	n	y	n	n	185	good
Penguin	n	n	y	n	y	140	evil
Catwoman	y	n	n	y	n	170	evil
Joker	n	n	n	n	n	179	evil
Batgirl	y	y	n	y	n	165	?
Riddler	y	n	n	n	n	182	?
Your Date	n	y	y	y	y	181	?

Example Tree:



Another Example:

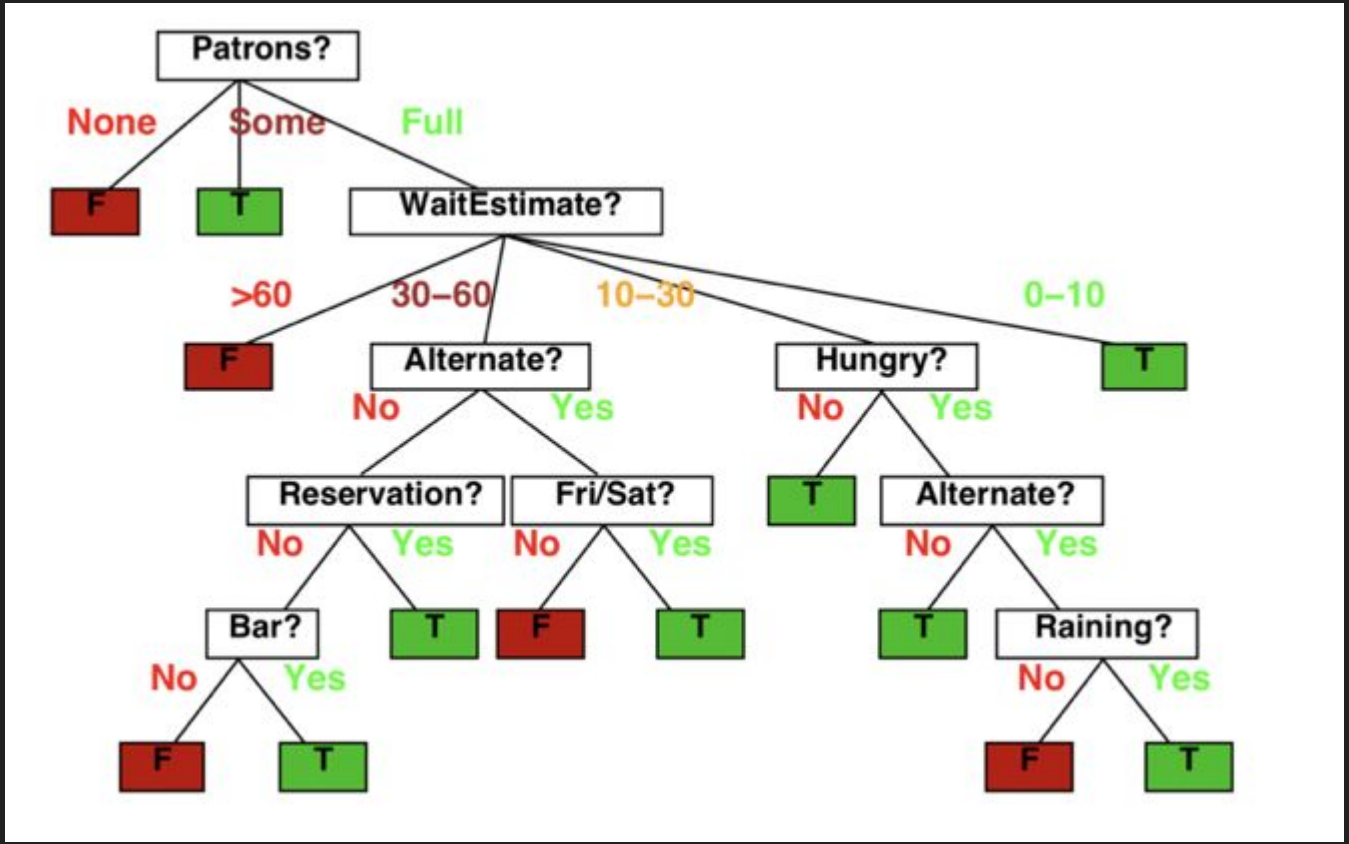
Ex	*Alt*	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	Wait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

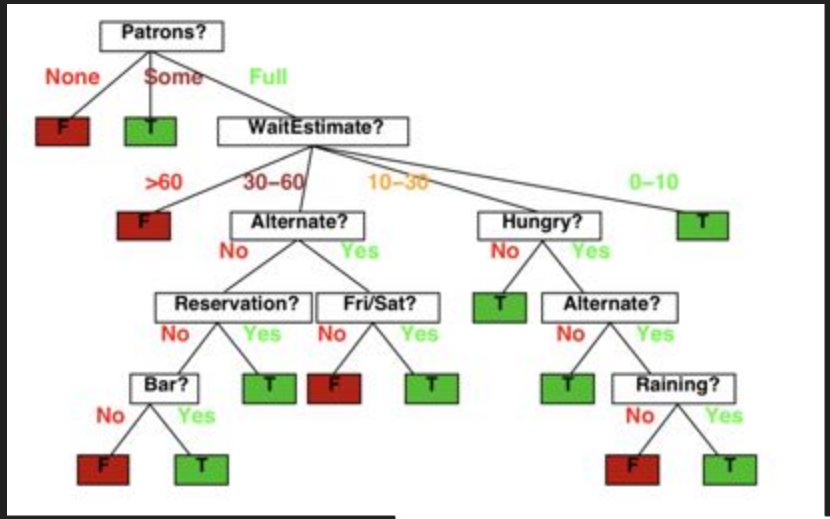
Attributes:

Target is whether or not a person will stay at a restaurant (T, F) with the following features:

1. Alternate: whether there is a suitable alternative restaurant nearby.
2. Bar: whether the restaurant has a comfortable bar area to wait in.
3. Fri/Sat: true on Fridays and Saturdays.
4. Hungry: whether we are hungry.
5. Patrons: how many people are in the restaurant (values are None, Some, and Full).
6. Price: the restaurant’s price range (one, two, or three $'s)
7. Raining: whether it is raining outside.
8. Reservation: whether we made a reservation.
9. Type: the kind of restaurant (French, Italian, Thai, or burger).
10. Est: the wait estimated by the host (0–10 minutes, 10–30, 30–60, or >60).

Example Tree:

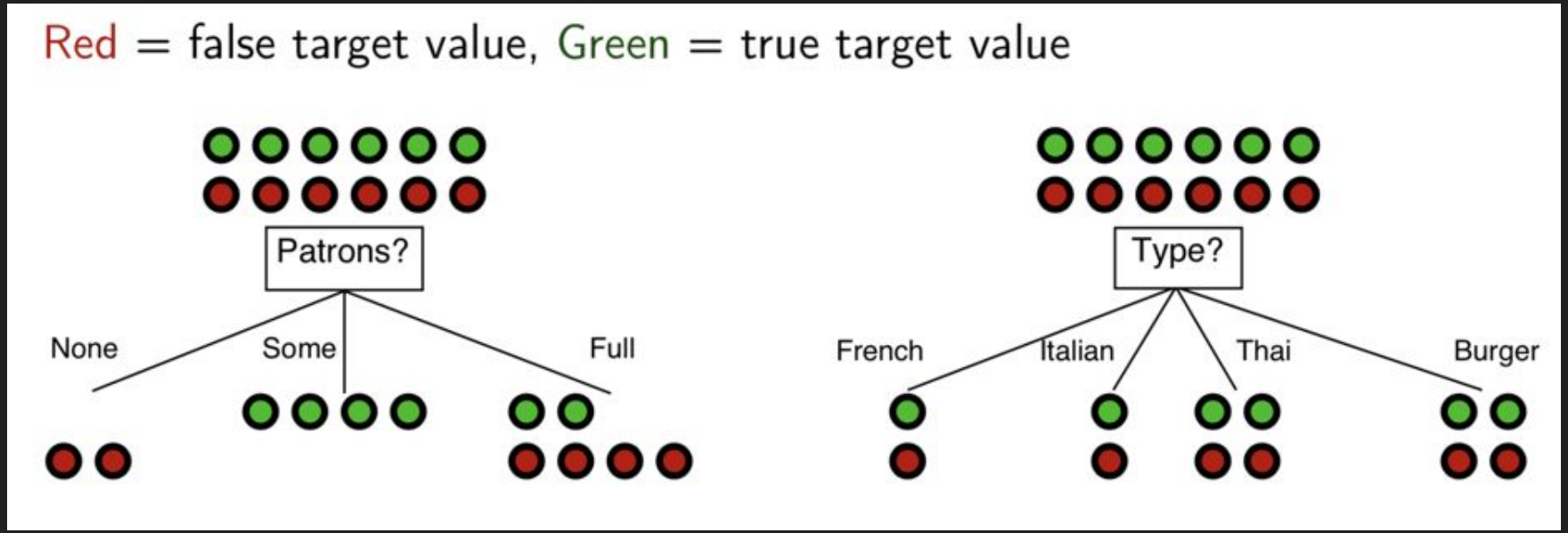




Ex	*Alt*	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	Wait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Choosing a Feature

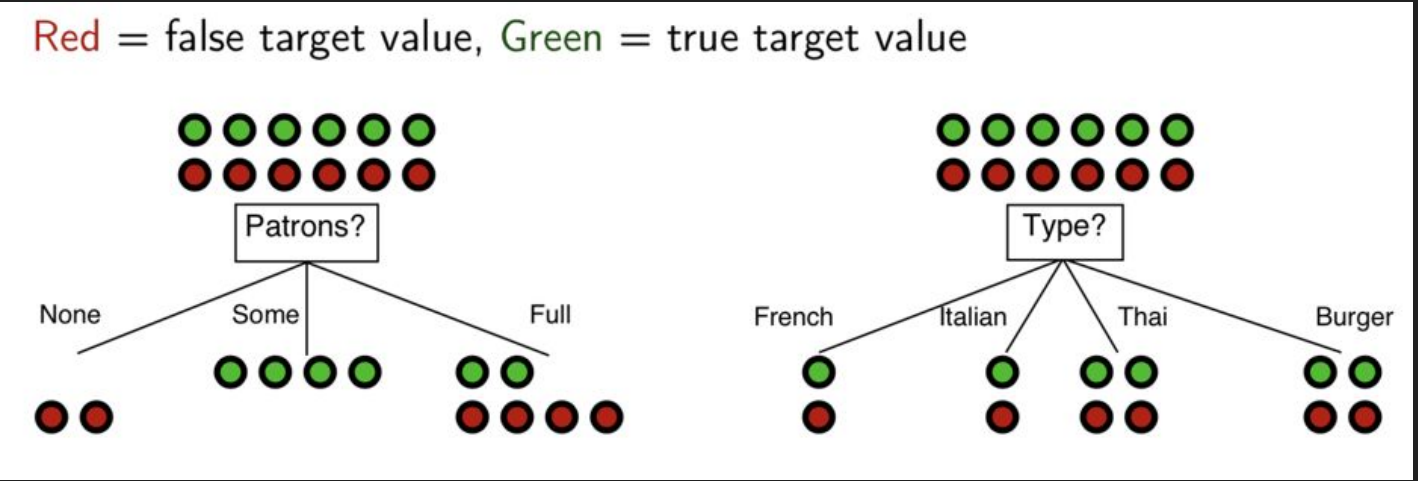
Which of these features do you think is a better choice for putting at the root of the decision tree?



Choosing a Feature

Idea: a good feature splits the examples into **subsets that are as pure as possible** (ideally) “all positive” or “all negative”

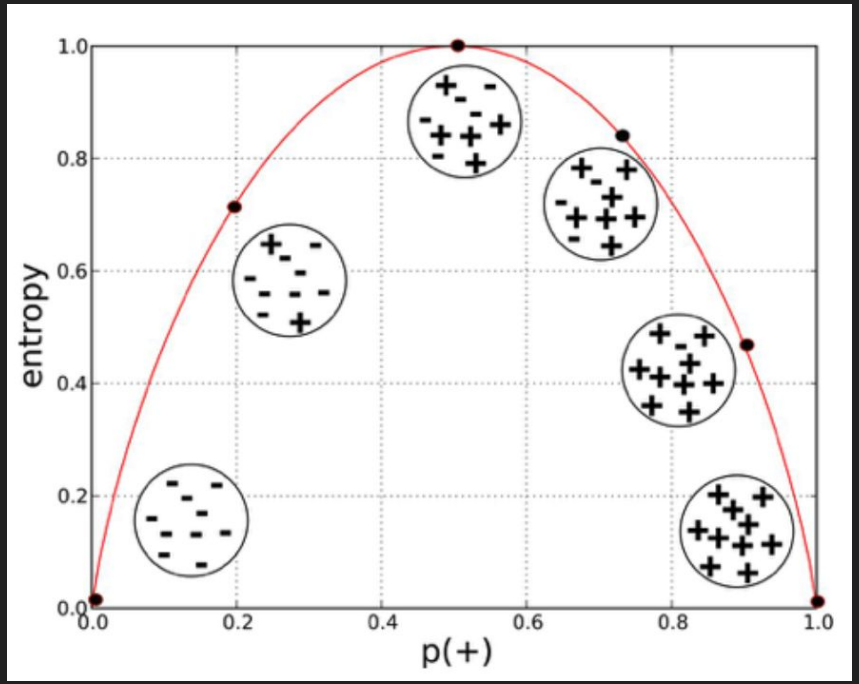
- *Patrons is a better choice--it gives more information about the classification*



Entropy

Entropy: measure of impurity

- **High entropy**: more evenly split classes - highly unpredictable
- **Low entropy**: mostly one class - highly predictable



Calculating Entropy Prior

Prior Probability: aka the ‘prior’

- the split of the examples
- If I have 9 positive examples and 5 negative examples my prior is:

  $\langle 9/14, 5/14 \rangle \approx \langle 0.64, 0.36 \rangle$

Calculating Entropy

Calculating the entropy when prior is ⟨P1,...,Pc⟩ is:

$$Entropy(\langle P_1, \ldots, P_c \rangle) = \sum_{i=1}^{c} -P_i \log_2 P_i$$

Calculating Entropy

Calculating the entropy when prior is ⟨P1,...,Pc⟩ is:

$$Entropy(\langle P_1, \ldots, P_c \rangle) = \sum_{i=1}^{c} -P_i \log_2 P_i$$

- entropy of prior ⟨0.5, 0.5⟩ is $-0.5 \log_2 (0.5) - 0.5 \log_2 (0.5) = 1$
- entropy of prior ⟨0.9, 0.1⟩ is $-0.9 \log_2 (0.9) - 0.1 \log_2 (0.1) \approx 0.47$
- entropy of prior ⟨0.64, 0.36⟩ is $-0.64 \log_2 (0.64) - 0.36 \log_2 (0.36) \approx 0.94$
- entropy of prior ⟨0.25, 0.25, 0.5⟩ is $-0.25 \log_2 (0.25) - 0.25 \log_2 (0.25) - 0.5 \log_2 (0.5) = 1.5$

The maximum entropy is **$\log_2(k)$**, where **k** is the number of categories. It is not always bounded by 0 and 1.

Calculating Entropy

Calculating the entropy when prior is ⟨P1,...,Pc⟩ is:

$$Entropy(\langle P_1, \ldots, P_c \rangle) = \sum_{i=1}^{c} -P_i \log_2 P_i$$

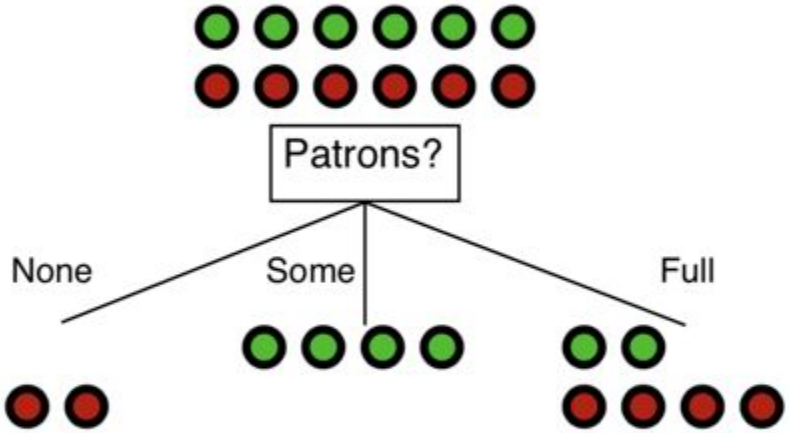
- entropy of prior ⟨0.5, 0.5⟩ is $-0.5 \log_2 (0.5) - 0.5 \log_2 (0.5) = 1$
- entropy of prior ⟨0.9, 0.1⟩ is $-0.9 \log_2 (0.9) - 0.1 \log_2 (0.1) \approx 0.47$
- entropy of prior ⟨0.64, 0.36⟩ is $-0.64 \log_2 (0.64) - 0.36 \log_2 (0.36) \approx 0.94$
- entropy of prior ⟨0.25, 0.25, 0.5⟩ is $-0.25 \log_2 (0.25) - 0.25 \log_2 (0.25) - 0.5 \log_2 (0.5) = 1.5$

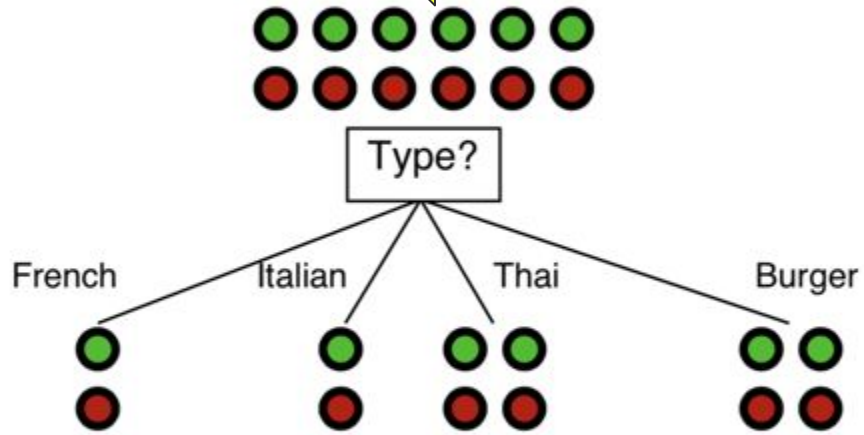
The maximum entropy is **$\log_2(k)$**, where **k** is the number of categories. It is not always bounded by 0 and 1.

$\log_2 (3) = 1.584962501$

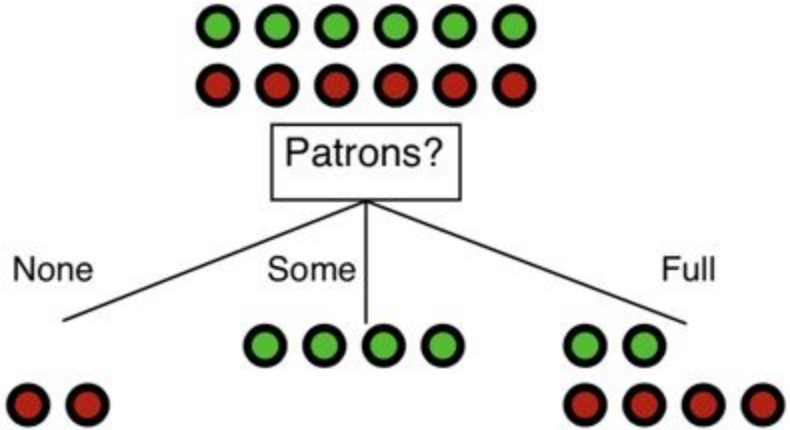
Example:

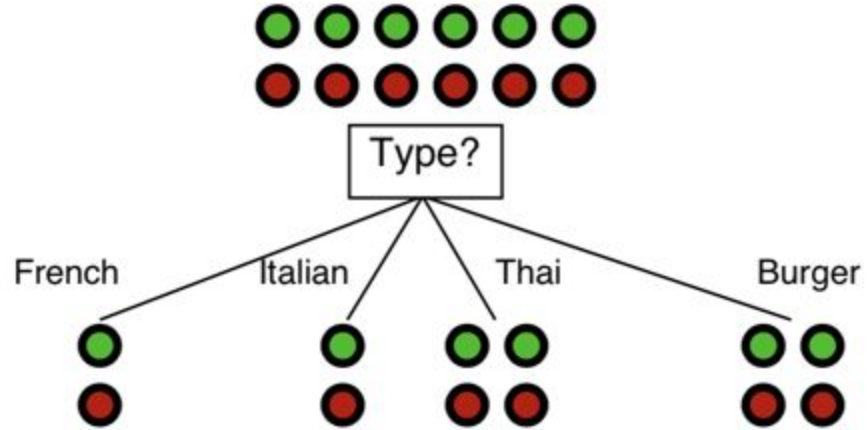
Entropy of the examples before picking a feature: 1





Exercise: compute the entropy of each group after sorting





So, the entropy for the three sets after sorting according to *Patrons* is

$$-\frac{0}{2}\log_2\frac{0}{2}-\frac{2}{2}\log_2\frac{2}{2}=0,$$

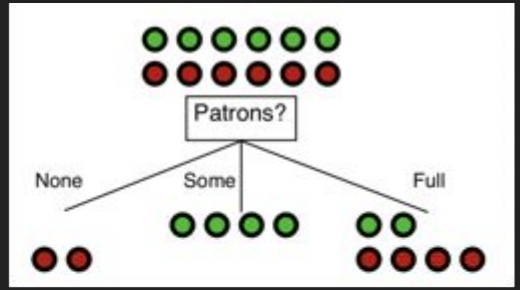
$$-\frac{4}{4}\log_2\frac{4}{4}-\frac{0}{2}\log_2\frac{0}{2}=0,$$

and $$-\frac{2}{6}\log_2\frac{2}{6}-\frac{4}{6}\log_2\frac{4}{6}\approx 0.918$$

Code for Entropy

Expected Entropy

The expected entropy for a feature is defined as the weighted sum of entropies multiplied by the fraction of samples that belong to each set.



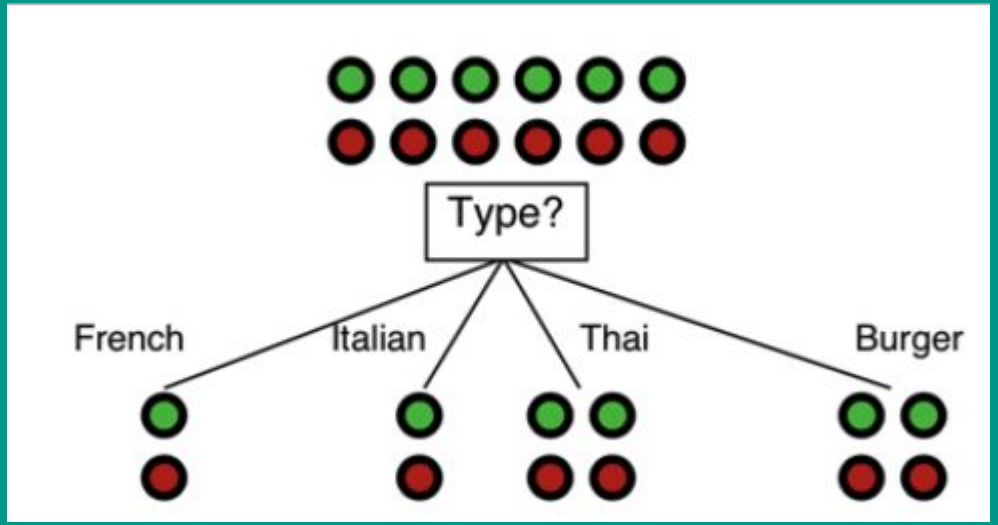
Then, the *expected entropy* remaining after testing the *Patrons* is

$$\approx \underbrace{\frac{2}{12}}_{\text{None}} \cdot 0 + \underbrace{\frac{4}{12}}_{\text{Some}} \cdot 0 + \underbrace{\frac{6}{12}}_{\text{Full}} \cdot 0.918 \approx 0.459$$

Exercise #1

What is the expected entropy for Type?

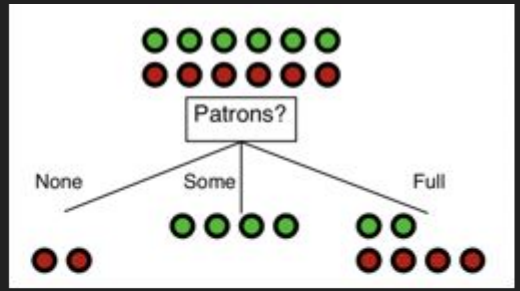
Can you say without doing the math?



Expected Entropy

The expected entropy for a feature is defined as the weighted sum of entropies multiplied by the fraction of samples that belong to each set.

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i)$$



Then, the *expected entropy* remaining after testing the *Patrons* is

$$\approx \underbrace{\frac{2}{12}}_{\text{None}} \cdot 0 + \underbrace{\frac{4}{12}}_{\text{Some}} \cdot 0 + \underbrace{\frac{6}{12}}_{\text{Full}} \cdot 0.918 \approx 0.459$$

Information Gain

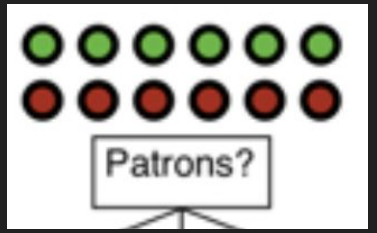
The *difference* between the entropy before the test and the expected entropy after the test is the expected **information gain.**

Gain() = Entropy (before) - Expected Entropy (after)

Gain(Patrons) = 1.0 - 0.459 = 0.541

Calculating Information Gain:

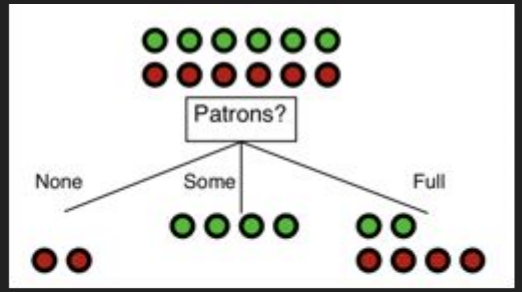
Entropy_before = 1



1. Calculate the entropy of the distribution of the classes before the node you're testing. This is the **entropy_before**

Calculating Information Gain:

1. Calculate the entropy of the distribution of the classes before the node you're testing. This is the **entropy_before**
2. Calculate the **expected entropy**
   a. The weighted sum of the entropy of each split of the data

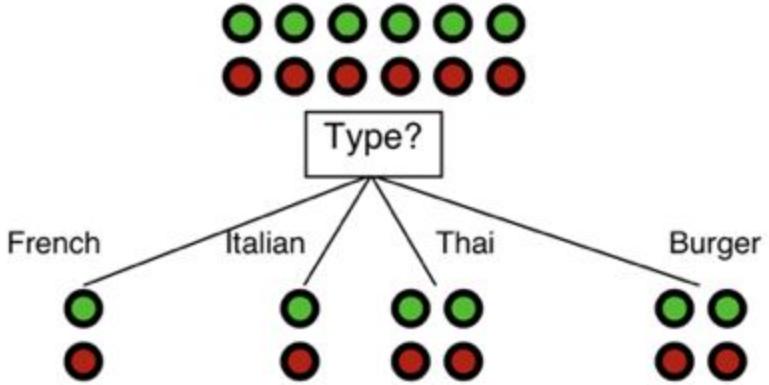


Then, the *expected entropy* remaining after testing the *Patrons* is

$$\approx \frac{2}{12} \cdot 0 + \frac{4}{12} \cdot 0 + \frac{6}{12} \cdot 0.918 \approx 0.459$$

Calculating Information Gain:

1. Calculate the entropy of the distribution of the classes before the node you're testing. This is the **entropy_before**
2. Calculate the **expected entropy**
   a. The weighted sum of the entropy of each split of the data
3. Find the difference between the **entropy_before** and **expected_entropy.**
   a. **Information Gain** = **entropy_before** - **expected_entropy**



Note that the expected entropy for the *Type* feature is

$$\frac{2}{12} \cdot \mathit{Entropy}\left(\left\langle \frac{1}{2}, \frac{1}{2} \right\rangle\right) + \frac{2}{12} \cdot \mathit{Entropy}\left(\left\langle \frac{1}{2}, \frac{1}{2} \right\rangle\right)$$

$$+\frac{4}{12} \cdot \mathit{Entropy}\left(\left\langle \frac{2}{4}, \frac{2}{4} \right\rangle\right) + \frac{4}{12} \cdot \mathit{Entropy}\left(\left\langle \frac{2}{4}, \frac{2}{4} \right\rangle\right)$$

$$= \frac{2}{12} \cdot 1 + \frac{2}{12} \cdot 1 + \frac{4}{12} \cdot 1 + \frac{4}{12} \cdot 1 = 1$$

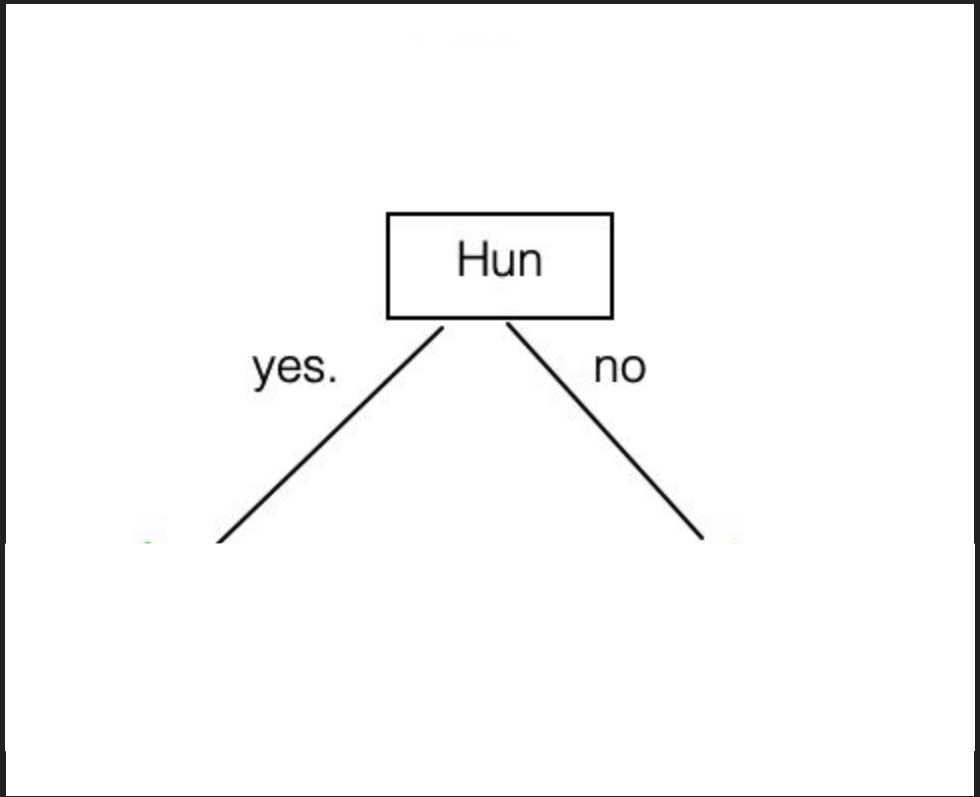
So,

$$\mathit{Gain}(\mathit{Type}) = 1 - 1 = 0$$

Exercise #2

Calculate the Information Gain for *Hun* and *Est*:

Ex	*Alt*	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	Wait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T


Hun
yes.
no

Exercise #3

Ex	*Alt*	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	Wait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

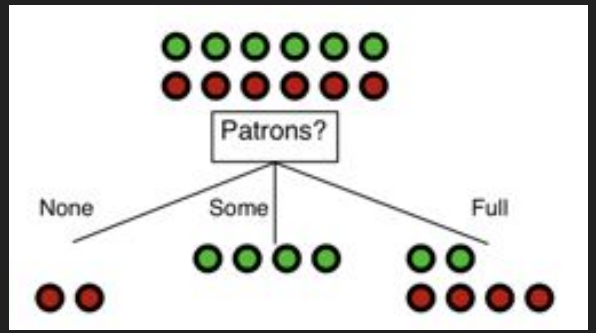
Exercise: Compute the information gain for the rest of the features.

Information gain results:

- Gain(Alt) = 1 - 1 = 0
- Gain (Bar) = 1 - 1 = 0
- Gain(Fri) = 1 - 0.979 = 0.021
- Gain(Hun) = 1 - 0.804 = 0.196
- Gain(Pat) = 1 - 0.459 = 0.541
- Gain(Price) = 1 - 0.804 = 0.196
- Gain(Rain) = 1 - 1 = 0
- Gain(Res) = 1 - 0.979 = 0.021
- Gain(Type) = 1 - 1 = 0
- Gain(Est) = 1 - 0.792 = 0.208

Information gain results:

- Gain(Alt) = 1 - 1 = 0
- Gain (Bar) = 1 - 1 = 0
- Gain(Fri) = 1 - 0.979 = 0.021
- Gain(Hun) = 1 - 8.04 = 0.196
- **Gain(Pat) = 1 - 0.459 = 0.541**
- Gain(Price) = 1 - 0.804 = 0.196
- Gain(Rain) = 1 - 1 = 0
- Gain(Res) = 1 - 0.979 = 0.021
- Gain(Type) = 1 - 1 = 0
- Gain(Est) = 1 - 0.792 = 0.208



Exercise #1

Use the ID3 Decision Tree algorithm to classify whether a member of congress is aligned with Republican or Democrat (Party is the target column).

We have voting records showing whether they votes yes, no, or didn't vote (?) on a variety of bills (shown as column B1-B5 here - they're predictory features).

Assume we've been running the algorithm and at some point, we're left with these five examples at some node in the middle of the tree, and we need to select the best feature to use at this node.

What is the information gain for B3?

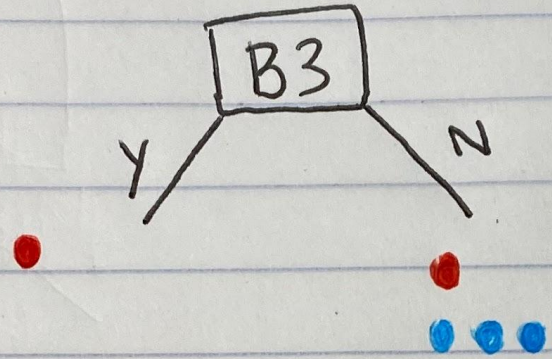
Party	B1	B2	B3	B4	B5
republican	n	y	y	?	y
republican	n	y	n	n	y
democrat	?	y	n	y	y
democrat	n	y	n	y	n
democrat	y	y	n	y	y

What is the information gain for B3?

Party	B1	B2	B3	B4	B5
republican	n	y	y	?	y
republican	n	y	n	n	y
democrat	?	y	n	y	y
democrat	n	y	n	y	n
democrat	y	y	n	y	y

```
entropy(0.00,1.00):  0.000
entropy(0.05,0.95):  0.286
entropy(0.10,0.90):  0.469
entropy(0.15,0.85):  0.610
entropy(0.20,0.80):  0.722
entropy(0.25,0.75):  0.811
entropy(0.30,0.70):  0.881
entropy(0.35,0.65):  0.934
entropy(0.40,0.60):  0.971
entropy(0.45,0.55):  0.993
entropy(0.50,0.50):  1.000
entropy(0.55,0.45):  0.993
entropy(0.60,0.40):  0.971
entropy(0.65,0.35):  0.934
entropy(0.70,0.30):  0.881
entropy(0.75,0.25):  0.811
entropy(0.80,0.20):  0.722
entropy(0.85,0.15):  0.610
entropy(0.90,0.10):  0.469
entropy(0.95,0.05):  0.286
entropy(1.00,0.00):  0.000
```

I made a mistake here… What was it?



$$EE = \frac{1}{5} \cdot E(\langle 1,0 \rangle) + \frac{4}{5} \cdot E\left(\left(\frac{1}{4}, \frac{3}{4}\right)\right) = 0.64$$

Info Gain: $1 - 0.64 = 0.351$

Numeric Features

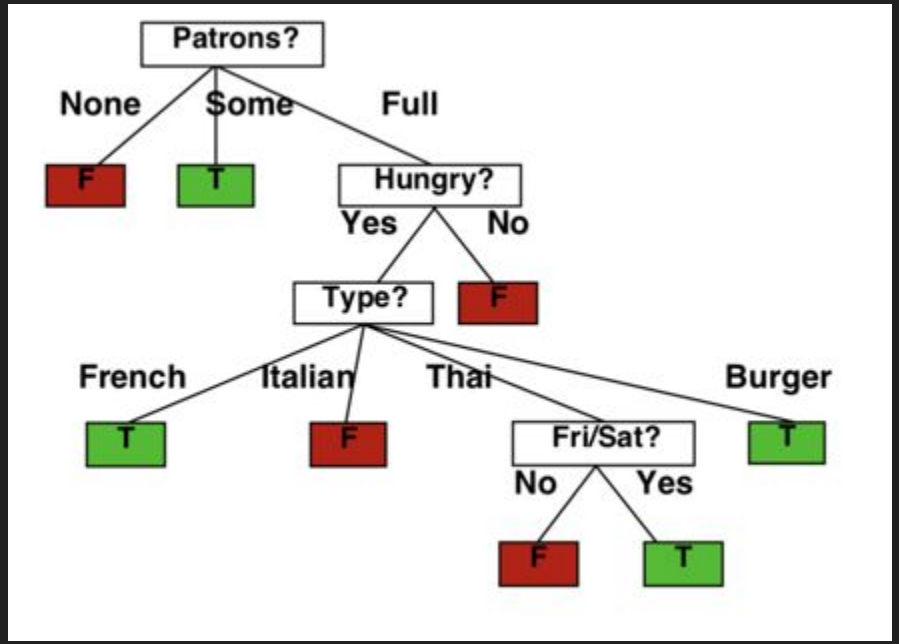
What do we do if we have numeric (even continuous-valued) features like age from the titanic dataset or petal length from the iris dataset?

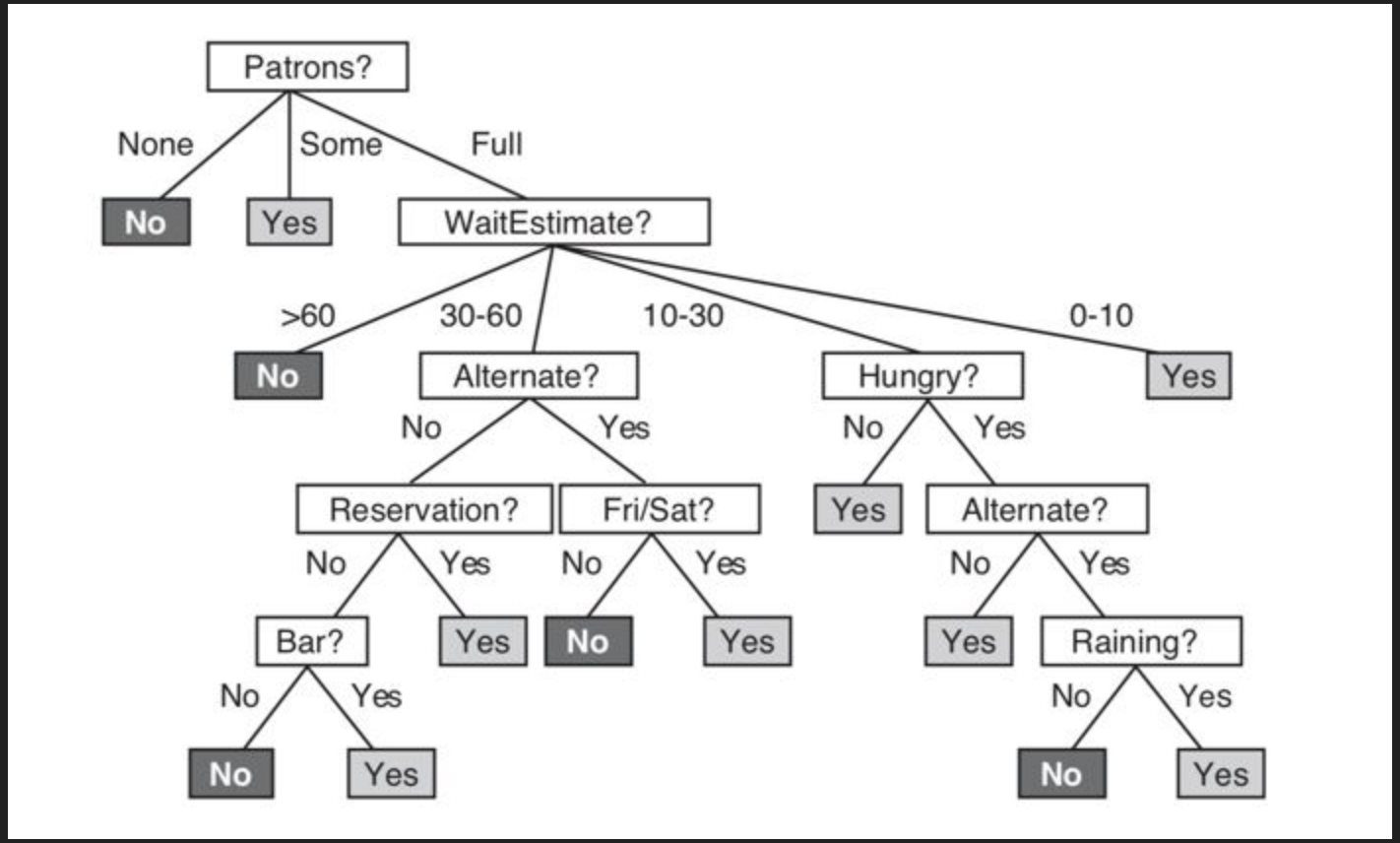
Idea: Decision Tree thresholds: if age > 70

Unfortunate annoying thing: Even though decision tree algorithms work well with categorical data, the Python library we will work with still wants all predictor features converted to a number, so we will have to work with numbers no matter what.

Tree Size Discussion

Decision tree learned from the 12 examples:




Patrons?
None
Some
Full
No
Yes
WaitEstimate?
>60
30-60
10-30
0-10
No
Alternate?
Hungry?
Yes
No
Yes
No
Yes
Reservation?
Fri/Sat?
Yes
Alternate?
No
Yes
No
Yes
No
Yes
Bar?
Yes
No
Yes
Yes
Raining?
No
Yes
No
Yes
No
Yes
No
Yes

Tree Size Discussion

Many different consistent trees possible:

- What quality is preferably?
  - More nodes v fewer nodes?
- What are the consequences of having a deep tree with many nodes?

Inductive Bias of ID3

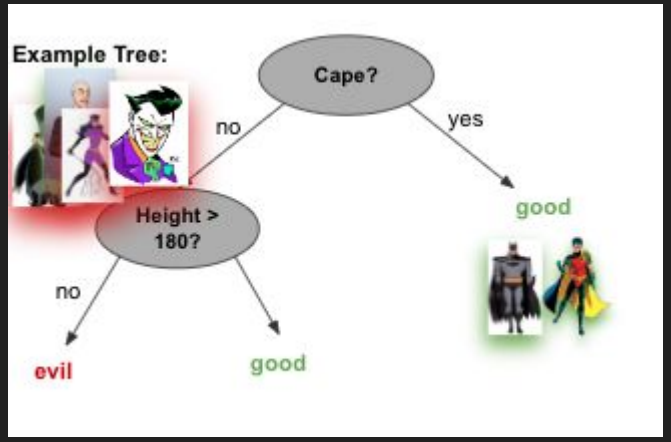
Shorter trees are preferred in ID3, trees with **high-information features closer to the root** are preferred

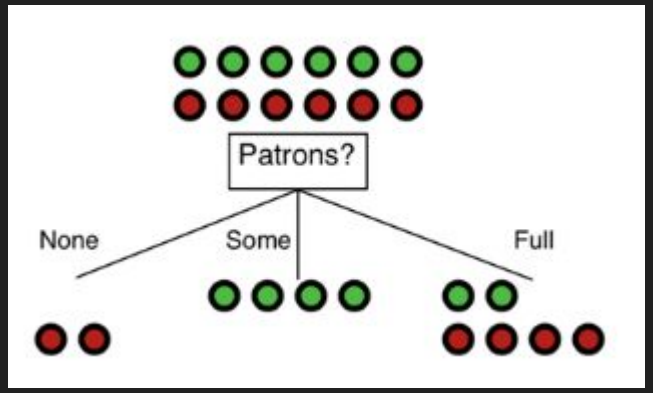
Baises allow us to learn, but you should understand what your algorithm’s bias is.

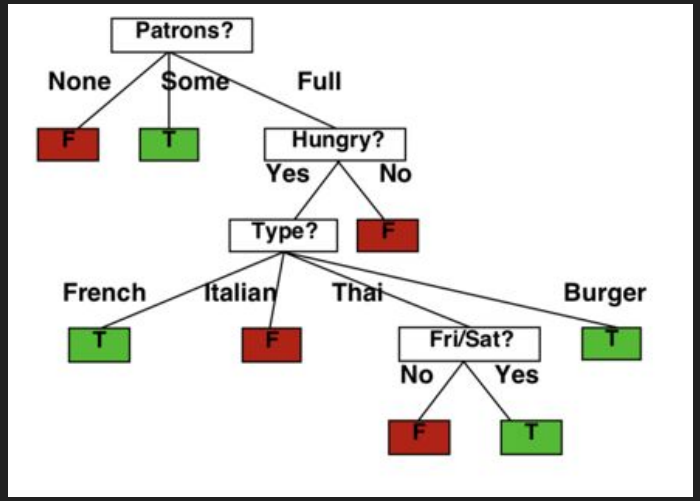
Leaf Nodes

We don’t have to always create a tree that is consistent with the training data.

Maybe a leaf node that classifies a majority of the cases accurately will *generalize* better than a very deep tree.


Example Tree:
Cape?
no
yes
good
Height > 180?
no
evil
good


Patrons?
None
Some
Full


Patrons?
None
Some
Full
F
T
Hungry?
Yes
No
Type?
F
French
Italian
Thai
Burger
T
F
Fri/Sat?
T
No
Yes
F
T

What about Noisy Data?

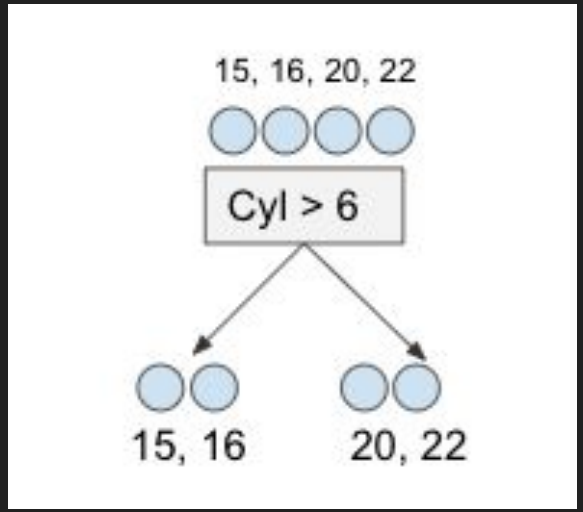
Ex	*Alt*	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	Wait
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T
X_{13}	T	T	T	T	Full	$	F	F	Burger	30–60	F

Regression with Decision Trees

Example: MPG target values

Which group of examples is more pure?

- Group A: 20, 19, 17, 19, 18
- Group B: 14, 17, 25, 7, 30
  - If you had to make a prediction at a leaf node with these groups what would you predict? Which one do you feel better about your prediction and why?
  - Hint: Is there some statistic we've already talked about this semester that can evaluate the purity of a group of numeric target values?

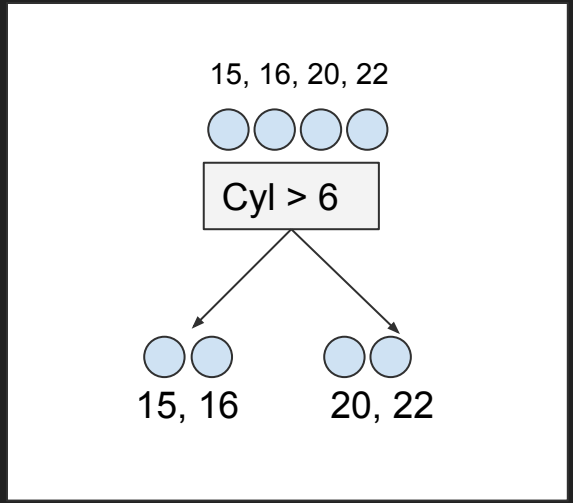

15, 16, 20, 22
Cyl > 6
15, 16
20, 22

Regression with Decision Trees

Example: MPG target values

Which group of examples is more pure?

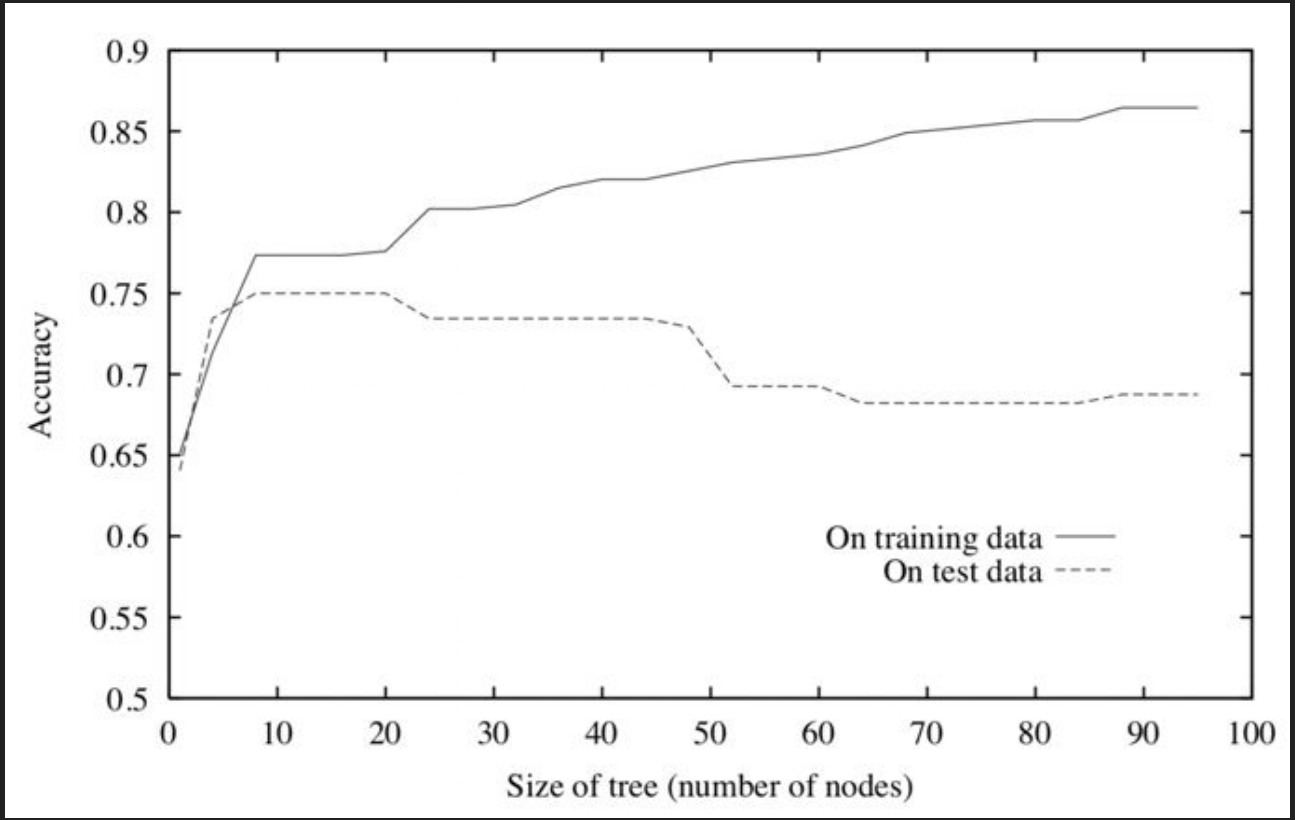
- Group A: 20, 19, 17, 19, 18
- Group B: 14, 17, 25, 7, 30
  - If you had to make a prediction at a leaf node with these groups what would you predict?
    - Hard to calculate entropy without categorical values
  - Hint: Is there some statistic we've already talked about this semester that can evaluate the purity of a group of numeric target values?
  - **Standard Deviation**



Overfitting

Big idea: You **overfit** if you do well on the training set, but not so well on the testing set.

Overfitting



Avoiding Overfitting

Make the tree smaller.

Some ideas on avoiding overly complex trees:

- Stop growing when data split is not statistically significant
- Grow full tree, then post-prune

Discussion Question

What are the benefits of decision trees compared to kNN?

- Disadvantages?

When would you use one over the other?

Discussion Question

What are the benefits of decision trees compared to kNN?

- Disadvantages?

When would you use one over the other?

If one column highly predicts the target variable→

If lots of predictors have similar weight in decision →

If you must be able to interpret the data clearly →

Discussion Question

What are the benefits of decision trees compared to kNN?

- Disadvantages?

When would you use one over the other?

If one column highly predicts the target variable→ decision tree

If lots of predictors have similar weight in decision → kNN

If you must be able to interpret the data clearly → decision tree

Recall

ID3 Decision Tree Learning Algorithm

Main ID3 Loop:

1. A ← the “best” decision feature for next node
2. Assign A as decision feature for node
3. For each possible attribute of A, create new descendant of node
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

But… what does ‘best’ mean? How would we go about deciding which node is the ‘best?

Exercise #2: Create the entire decision tree from this data

Umbrella?	Umbrella Yesterday?	Cloudy?	Temp > 70	Barometer	Humid?
No	Y	Partial	N	Falling	Y
No	N	Complete	N	Rising	N
No	Y	Complete	N	Rising	Y
Yes	N	Complete	N	Neither	Y
Yes	N	Partial	Y	Falling	Y

```
entropy(0.00,1.00):  0.000
entropy(0.05,0.95):  0.286
entropy(0.10,0.90):  0.469
entropy(0.15,0.85):  0.610
entropy(0.20,0.80):  0.722
entropy(0.25,0.75):  0.811
entropy(0.30,0.70):  0.881
entropy(0.35,0.65):  0.934
entropy(0.40,0.60):  0.971
entropy(0.45,0.55):  0.993
entropy(0.50,0.50):  1.000
entropy(0.55,0.45):  0.993
entropy(0.60,0.40):  0.971
entropy(0.65,0.35):  0.934
entropy(0.70,0.30):  0.881
entropy(0.75,0.25):  0.811
entropy(0.80,0.20):  0.722
entropy(0.85,0.15):  0.610
entropy(0.90,0.10):  0.469
entropy(0.95,0.05):  0.286
entropy(1.00,0.00):  0.000
```